

**SKRIPSI**

**RANCANG BANGUN SISTEM INFORMASI GIZILA  
UNTUK KEBUTUHAN GIZI HARIAN BERBASIS WEB  
MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT (RAD)***



**Oleh:**

**ULFA HIDAYANTI  
21250011**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS NAHDLATUL ULAMA INDONESIA  
JAKARTA  
2025**

**RANCANG BANGUN SISTEM INFORMASI GIZILA  
UNTUK KEBUTUHAN GIZI HARIAN BERBASIS WEB  
MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT (RAD)***

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk memperoleh  
Gelar Sarjana Komputer

**Oleh:**

**ULFA HIDAYANTI  
21250011**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS NAHDLATUL ULAMA INDONESIA  
JAKARTA  
2025**

## **HALAMAN PERSETUJUAN**

Nama : Ulfa Hidayanti  
NIM : 21250011  
Program Studi : Sistem Informasi  
Fakultas : Teknik dan Ilmu Komputer  
Jenjang Pendidikan : Strata 1  
Judul Skripsi : RANCANG BANGUN SISTEM INFORMASI GIZILA UNTUK KEBUTUHAN GIZI HARIAN BERBASIS WEB MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT (RAD)*

Skripsi ini telah diperiksa dan disetujui untuk diajukan ke sidang akhir.

Jakarta, 14 Juli 2025

Menyetujui,  
Dosen Pembimbing



**Handy Fernandy, S.T., M.M.S.I.**  
NIDN. 0309029202

Mengetahui,  
Ketua Program Studi  
Sistem Informasi



**Handy Fernandy, S.T., M.M.S.I.**  
NIDN. 0309029202

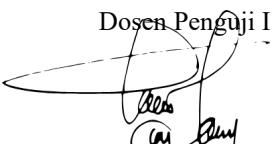
## HALAMAN PENGESAHAN

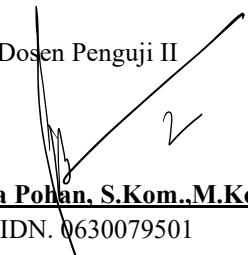
Nama : Ulfa Hidayanti  
NIM : 21250011  
Program Studi : Sistem Informasi  
Fakultas : Teknik dan Ilmu Komputer  
Jenjang Pendidikan : Strata 1  
Judul Skripsi : RANCANG BANGUN SISTEM INFORMASI GIZILA UNTUK KEBUTUHAN GIZI HARIAN BERBASIS WEB MENGGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT (RAD)*

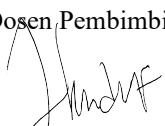
Skripsi ini telah dipertahankan dihadapan dewan penguji ujian skripsi Fakultas Teknik dan Ilmu Komputer, Program Studi Sistem Informasi dan dinyatakan **LULUS**.

Jakarta, 2 September 2025

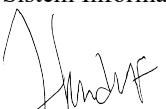
Menyetujui,

Dosen Penguji I  
  
Arifin A. Abd. Karim, S.Kom., M.M.S.I.  
NIDN. 0306099002

Dosen Penguji II  
  
Sry Dhina Pohan, S.Kom.,M.Kom.  
NIDN. 0630079501

Dosen Pembimbing  
  
Handy Fernandy, S.T., M.M.S.I.  
NIDN. 0309029202

Mengetahui,  
Ketua Program Studi  
Sistem Informasi

  
Handy Fernandy, S.T., M.M.S.I.  
NIDN. 0309029202

## **HALAMAN MOTTO**

*"Last but not least, I want to thank me. I want to thank me for believing in me. I want to thank me for doing all this hard work. I want to thank me for not having days off. I want to thank me for never quitting. I want to thank me for always being a giver and trying to give more than I receive. I want to thank me for trying to do more right than wrong. I want to thank me for just being me at all times". Lov u all*

## **HALAMAN PERSEMPAHAN**

Penulis ingin menyampaikan rasa terima kasih kepada sosok yang selama ini tak kenal lelah berjuang, seorang wanita sederhana dengan mimpi besar, yang sering kali sulit dimengerti baik pikirannya maupun hatinya. Ucapan ini ditujukan kepada penulis skripsi ini sendiri, Ulfa Hidayanti, terima kasih atas kehadiranmu di dunia ini, atas kekuatanmu bertahan sejauh ini, dan keberanianmu untuk terus melangkah menghadapi tantangan hidup. Terima kasih telah setia menjadi dirimu sendiri. Aku bangga pada setiap langkah kecil yang kau ambil dan setiap pencapaianmu, meski tak selalu diakui atau dirayakan orang lain. Walaupun harapanmu tak selalu sejalan dengan apa yang diberikan semesta, belajarlah untuk menerima dan mensyukuri setiap anugerah yang datang. Tetaplah bahagia di mana pun kamu berada. Rayakan dirimu sendiri dan jadikan kehadiranmu bercahaya di setiap tempat yang kamu pijaki. Semoga langkahmu selalu diberkahi, selalu diberi kekuatan, dikelilingi oleh orang-orang baik, dan mimpi-mimpimu satu per satu terwujud. Aamiin.

Skripsi ini kupersembahkan untuk orang-orang tercinta terutama kedua orang tua saya yang selalu mendukung dan tidak pernah berhenti untuk mendo'akan. Terima kasih juga kuucapkan untuk keluarga besar PT GUNTARA semoga dapat menjadi salah satu perusahaan maju di Indonesia yang bermanfaat bagi bangsa, negara dan agama. Kupersembahkan secara lebih dekat untuk keluarga saya terutama kepada kedua orang tua tercinta, kakak, paman, dan bibi, saya ucapkan terima kasih yang senantiasa menyediakan bantuan dalam bentuk motivasi, moral, maupun materinya.

## PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini :

Nama : Ulfa Hidayanti

NIM : 21250011

Program Studi : Sistem Informasi

Fakultas : Teknik dan Ilmu Komputer

Jenjang Pendidikan : Strata 1

Menyatakan bahwa skripsi yang berjudul "**RANCANG BANGUN SISTEM INFORMASI GIZILA UNTUK KEBUTUHAN GIZI HARIAN BERBASIS WEB MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT**" ini adalah karya asli saya, dan isi skripsi ini tidak mencakup karya yang pernah diajukan oleh individu lain untuk memperoleh gelar akademis di institusi pendidikan tinggi. Selama pengetahuan saya, tidak ada karya atau pendapat yang pernah ditulis dan/atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam naskah ini dan dicantumkan dalam daftar pustaka. Saya membuat pernyataan ini dengan sepenuh hati, jika di kemudian hari terdapat penyimpangan atau ketidakbenaran dalam pernyataan ini, saya bersedia menerima SANKSI AKADEMIK, termasuk pencabutan gelar yang telah diperoleh, serta sanksi lain sesuai norma yang berlaku di Perguruan Tinggi.

Dinyatakan di Jakarta

2 September 2025



Ulfa Hidayanti

21250011

## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT atas limpahan rahmat, ridho, dan hidayah-Nya, sehingga penulis dapat menuntaskan penyusunan skripsi yang berjudul **“Rancang Bangun Sistem Informasi Gizila Untuk Kebutuhan Gizi Harian Berbasis Web Menggunakan Metode Rapid Application Development”** ini. Tujuan penulisan skripsi ini adalah untuk memenuhi syarat kelulusan Program Studi Strata I Sistem Informasi di Universitas Nahdlatul Ulama Indonesia.

Penulis menyadari bahwa dalam penyusunan laporan ini terdapat sejumlah kesulitan dan hambatan. Selain itu, penulis menyadari bahwa laporan ini masih jauh dari sempurna dan memiliki banyak kekurangan. Oleh karena itu, penulis mengharapkan saran dan kritik yang konstruktif dari semua pihak. Juga penulis ingin mengungkapkan rasa terima kasih yang mendalam kepada semua pihak yang telah memberikan bantuan dalam penyusunan laporan ini:

1. Tuhan Yang Maha Esa, Allah SWT yang telah memberikan kesehatan, pertolongan, kekuatan serta petunjuk-Nya sehingga penulis dapat menyelesaikan skripsi ini.
2. Bapak dr. Syahrizal Syarif, MPH, Ph.D., yang menjabat sebagai Pelaksana Tugas atau PLT Rektor di Universitas Nahdlatul Ulama Indonesia.
3. Ibu Adrinoviarini, M.Sc., selaku Dekan atau Pimpinan Fakultas Teknik dan Ilmu Komputer di Universitas Nahdlatul Ulama Indonesia
4. Bapak Handy Fernandy, S.T., M.M.S.I., selaku Kaprodi atau Pimpinan Program Studi Sistem Informasi di Universitas Nahdlatul Ulama Indonesia. Juga selaku Dosen Pembimbing yang terus menerus mendorong penulis untuk menyelesaikan penelitian skripsi ini.

Akhir kata, semoga Allah SWT senantiasa melimpahkan karunia-Nya serta membala setiap amal, budi, dan kebaikan semua pihak yang telah memberikan bantuan dalam penyusunan laporan ini. Penulis juga berharap karya ini dapat memberikan manfaat bagi setiap pihak yang membutuhkannya.

Jakarta, 14 Juli 2025



Ulfa Hidayanti

## DAFTAR ISI

HALAMAN PERSETUJUAN .....	i
HALAMAN PENGESAHAN .....	ii
HALAMAN MOTTO .....	iii
HALAMAN PERSEMBAHAN.....	iv
PERNYATAAN KEASLIAN .....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	vii
DAFTAR TABEL .....	xii
DAFTAR GAMBAR .....	xiii
ABSTRAK.....	xv
<i>ABSTRACT.....</i>	xvi
BAB I PENDAHULUAN .....	1
1.1.    Latar Belakang .....	1
1.2.    Rumusan Masalah .....	5
1.3.    Batasan Masalah.....	5
1.4.    Tujuan Penelitian.....	5
1.5.    Manfaat Penelitian.....	6
1.6.    Metode Penelitian.....	7
1.7.    Sistematika Penulisan.....	10
BAB II TINJAUAN PUSTAKA .....	12
2.1.    Penelitian Terdahulu.....	12
2.2.    Landasan Teori .....	16
2.2.1.    Rancang Bangun.....	16
2.2.2.    Figma.....	16
2.2.3.    Internet .....	16
2.2.4.    Website .....	17
2.2.5.    Sistem Informasi.....	17
2.2.6. <i>Unified Modelling Language (UML)</i> .....	18
2.2.7.    Gizi .....	28
2.2.8.    Perhitungan Rumus <i>Harris Benedict</i> .....	30
2.2.9.    Kalori.....	31
2.2.10.    Gizila.....	32
2.2.11.    Profil PT Graha Usaha Nusantara (GUNTARA).....	33

2.2.12.	<i>PHP</i> .....	34
2.2.13.	<i>HTML</i> .....	35
2.2.14.	<i>Tailwind CSS</i> .....	36
2.2.15.	<i>Visual Studio Code</i> .....	37
2.2.16.	<i>User Interface</i> dan <i>User Experience</i> .....	38
2.2.17.	<i>System Usability Scale (SUS)</i> .....	39
2.2.18.	<i>User Acceptance Testing (UAT)</i> .....	40
2.2.19.	<i>Black Box Testing</i> .....	41
2.2.20.	<i>XAMPP</i> .....	41
2.2.21.	<i>MySQL</i> .....	42
2.2.22.	<i>Framework Laravel</i> .....	42
2.2.23.	<i>Livewire</i> .....	43
2.2.24.	<i>Alpine.js</i> .....	44
2.2.25.	<i>Metode Rapid Application Development (RAD)</i> .....	44
	BAB III ANALISIS DAN PERANCANGAN .....	46
3.1.	Analisis Sistem yang Berjalan.....	46
3.1.1.	Analisis Proses Bisnis.....	46
3.1.2.	Analisis Sistem Informasi.....	48
3.1.2.1.	Kelebihan Sistem Informasi yang Berjalan .....	49
3.1.2.2.	Kekurangan Sistem Informasi yang Berjalan .....	50
3.2.	Analisis Sistem yang Diusulkan .....	51
3.2.1.	Kebutuhan Fungsional .....	53
3.2.2.	Kebutuhan Non Fungsional .....	54
3.3.	Desain Sistem yang Diusulkan .....	55
3.3.1.	<i>Use Case Diagram</i> .....	55
3.3.1.1.	<i>Use Case Login</i> .....	58
3.3.1.2.	<i>Use Case Akses Dashboard admin Gizila</i> .....	59
3.3.1.3.	<i>Use Case Kelola Dashboard admin Gizila</i> .....	60
3.3.1.4.	<i>Use Case Kelola Fitur CRUD Artikel di Dashboard</i> .....	60
3.3.1.5.	<i>Use Case Kelola Profil</i> .....	61
3.3.1.6.	<i>Use Case Kelola Hak Akses Admin Dashboard</i> .....	62
3.3.1.7.	<i>Use Case Kelola Website Gizila</i> .....	63
3.3.1.8.	<i>Use Case Akses Website Gizila</i> .....	64
3.3.1.9.	<i>Use Case Lihat Fitur-fitur Website Gizila</i> .....	64
3.3.1.10.	<i>Use Case Menghitung Gizi Harian</i> .....	65
3.3.1.11.	<i>Use Case Lihat Artikel Edukasi</i> .....	66

3.3.1.12.	<i>Use Case Mendapat Info Edukasi di Website Gizila</i> .....	66
3.3.2.	<i>Activity Diagram</i> .....	67
3.3.2.1.	<i>Activity Diagram Login Admin</i> .....	67
3.3.2.2.	<i>Activity Diagram Akses Dashboard Admin Gizila</i> .....	68
3.3.2.3.	<i>Activity Diagram Kelola Dashboard admin Gizila</i> .....	69
3.3.2.4.	<i>Activity Diagram Kelola Fitur CRUD Artikel di Dashboard</i> .....	70
3.3.2.5.	<i>Activity Diagram Kelola Profil</i> .....	71
3.3.2.6.	<i>Activity Diagram Kelola Hak Akses Admin Dashboard</i> .....	72
3.3.2.7.	<i>Activity Diagram Kelola Website Gizila</i> .....	73
3.3.2.8.	<i>Activity Diagram Akses Website Gizila</i> .....	74
3.3.2.9.	<i>Activity Diagram Lihat Fitur-fitur Website Gizila</i> .....	75
3.3.2.10.	<i>Activity Diagram Menghitung Gizi Harian</i> .....	76
3.3.2.11.	<i>Activity Diagram Lihat Artikel Edukasi</i> .....	76
3.3.2.12.	<i>Activity Diagram Mendapat Info Edukasi di Website Gizila</i> .....	77
3.3.3.	<i>Sequence Diagram</i> .....	78
3.3.3.1.	<i>Sequence Diagram Login Admin</i> .....	78
3.3.3.2.	<i>Sequence Diagram Akses Dashboard admin Gizila</i> .....	79
3.3.3.3.	<i>Sequence Diagram Kelola Dashboard admin Gizila</i> .....	80
3.3.3.4.	<i>Sequence Diagram Kelola Fitur CRUD Artikel di Dashboard</i> .....	81
3.3.3.5.	<i>Sequence Diagram Kelola Profil</i> .....	84
3.3.3.6.	<i>Sequence Diagram Kelola Hak Akses Admin Dashboard</i> .....	85
3.3.3.7.	<i>Sequence Diagram Kelola Website Gizila</i> .....	86
3.3.3.8.	<i>Sequence Diagram Akses Website Gizila</i> .....	86
3.3.3.9.	<i>Sequence Diagram Lihat Fitur-fitur Website Gizila</i> .....	87
3.3.3.10.	<i>Sequence Diagram Menghitung Gizi Harian</i> .....	88
3.3.3.11.	<i>Sequence Diagram Lihat Artikel Edukasi</i> .....	89
3.3.3.12.	<i>Sequence Diagram Mendapat Info Edukasi di Website Gizila</i> .....	89
3.3.4.	<i>Class Diagram</i> .....	90
3.3.5.	Desain Antarmuka Pengguna .....	91
3.3.5.1	Desain Halaman <i>Login Admin</i> .....	91
3.3.5.2.	Desain Halaman Akses <i>Dashboard</i> admin Gizila .....	91
3.3.5.3.	Desain Halaman Kelola <i>Dashboard</i> admin Gizila.....	92
3.3.5.4.	Desain Halaman Kelola Fitur CRUD Artikel di <i>Dashboard</i> .....	92
3.3.5.5.	Desain Halaman Kelola <i>Website Gizila</i> .....	93
3.3.5.6.	Desain Halaman Akses <i>Website Gizila</i> .....	93
3.3.5.7.	Desain Halaman Lihat Fitur-fitur <i>Website Gizila</i> .....	94

3.3.5.8. Desain Halaman Menghitung Gizi Harian.....	94
3.3.5.9. Desain Halaman Lihat Artikel Edukasi .....	96
3.3.5.10. Desain Halaman Mendapat Info Edukasi di <i>Website</i> Gizila .....	96
3.3.5.11. Desain Halaman <i>About</i> .....	97
<b>BAB IV IMPLEMENTASI DAN PENGUJIAN.....</b>	<b>98</b>
4.1. Lingkungan Pengembangan .....	98
4.1.1. Perangkat Keras.....	98
4.1.2. Perangkat Lunak.....	98
4.2. Implementasi Antarmuka Pengguna.....	99
4.2.1. Implementasi Halaman <i>Login Admin</i> .....	99
4.2.2. Implementasi Halaman Akses <i>Dashboard</i> admin Gizila .....	100
4.2.3. Implementasi Halaman Kelola <i>Dashboard</i> admin Gizila .....	100
4.2.4. Implementasi Halaman Kelola Fitur CRUD Artikel di <i>Dashboard</i> .....	101
4.2.5. Implementasi Halaman Kelola Profil di <i>Dashboard</i> .....	101
4.2.6. Implementasi Halaman Kelola Hak Akses di <i>Dashboard</i> .....	102
4.2.7. Implementasi Halaman Kelola <i>Website</i> Gizila .....	102
4.2.8. Implementasi Halaman Akses <i>Website</i> Gizila .....	103
4.2.9. Implementasi Halaman Lihat Fitur-fitur <i>Website</i> Gizila.....	103
4.2.10. Implementasi Halaman Menghitung Gizi Harian.....	104
4.2.11. Implementasi Halaman Lihat Artikel Edukasi .....	107
4.2.12. Implementasi Halaman Mendapat Info Edukasi di <i>Website</i> Gizila .....	108
4.2.13. Implementasi Halaman <i>About</i> .....	109
4.3. Implementasi Sistem Informasi Antarmuka Admin .....	109
4.3.1. Pengujian Halaman <i>Login Admin</i> .....	109
4.3.2. Pengujian Halaman <i>Dashboard</i> Artikel.....	110
4.3.3. Pengujian Halaman <i>Dashboard</i> Kategori Artikel.....	111
4.3.4. Pengujian Halaman <i>Dashboard</i> Profil.....	112
4.3.5. Halaman <i>Dashboard</i> Admin Gizila .....	113
4.3.6. Halaman Manajemen Artikel .....	113
4.3.7. Halaman Kategori Artikel .....	114
4.3.8. Halaman Profil Admin .....	114
4.3.9. Halaman Hak Akses .....	115
4.4. <i>Black Box Testing</i> .....	115
4.4.1. Pengujian <i>Black Box</i> - <i>Login Admin</i> .....	115
4.4.2. Pengujian <i>Black Box</i> - Kelola <i>Dashboard</i> admin Gizila.....	116
4.4.3. Pengujian <i>Black Box</i> - Kelola Manajemen Artikel .....	116

4.4.4.	Pengujian <i>Black Box</i> – Halaman Manajemen Kategori .....	117
4.4.5.	Pengujian <i>Black Box</i> – Halaman <i>Logout</i> .....	118
4.4.6.	Pengujian <i>Black Box</i> – Halaman Profil .....	119
4.4.7.	Pengujian <i>Black Box</i> – Halaman <i>Website Gizila</i> .....	120
4.5.	<i>User Acceptance Test (UAT)</i> .....	123
4.5.1.	<i>User Acceptance Test Admin</i> .....	123
4.5.2.	<i>User Acceptance Test Publik</i> .....	124
4.6.	Tinjauan Hasil Uji <i>System Usability Scale (SUS)</i> .....	124
4.6.1.	Tahapan Perolehan Skor SUS .....	126
BAB V	PENUTUP .....	128
5.1.	Kesimpulan.....	128
5.2.	Saran.....	128
DAFTAR PUSTAKA.....		130
LAMPIRAN-LAMPIRAN .....		134

## DAFTAR TABEL

<b>Tabel 2.1</b> Perbandingan Penelitian Terdahulu.....	12
<b>Tabel 2.2</b> Kelebihan dan Kekurangan Penelitian Terdahulu .....	14
<b>Tabel 2.3</b> <i>Use Case Diagram</i> .....	21
<b>Tabel 2.4</b> <i>Activity Diagram</i> .....	23
<b>Tabel 2.5</b> <i>Class Diagram</i> .....	26
<b>Tabel 2.6</b> <i>Sequence diagram</i> .....	28
<b>Tabel 2.7</b> Faktor Aktivitas Kebutuhan Energi Pada Orang Sehat .....	31
<b>Tabel 2.8</b> Bobot Penilaian SUS.....	39
<b>Tabel 3.1</b> Kebutuhan Non Fungsional.....	54
<b>Tabel 3.2</b> Penjelasan <i>Use Case Diagram</i> .....	58
<b>Tabel 3.3</b> <i>Use Case Login Admin</i> .....	59
<b>Tabel 3.4</b> <i>Use Case Akses Dashboard</i> .....	59
<b>Tabel 3.5</b> <i>Use Case Kelola Dashboard</i> .....	60
<b>Tabel 3.6</b> <i>Use Case Kelola Fitur CRUD Artikel</i> .....	61
<b>Tabel 3.7</b> <i>Use Case Kelola Profil</i> .....	62
<b>Tabel 3.8</b> <i>Use Case Kelola Hak Akses Admin</i> .....	63
<b>Tabel 3.9</b> <i>Use Case Kelola Website Gizila</i> .....	64
<b>Tabel 3.10</b> <i>Use Case Akses Website Gizila</i> .....	64
<b>Tabel 3.11</b> <i>Use Case Lihat Fitur-fitur Website Gizila</i> .....	65
<b>Tabel 3.12</b> <i>Use Case Menghitung Gizi Harian</i> .....	65
<b>Tabel 3.13</b> <i>Use Case Lihat Artikel Edukasi</i> .....	66
<b>Tabel 3.14</b> <i>Use Case Mendapat Info Edukasi di Website Gizila</i> .....	66
<b>Tabel 4.1</b> Spesifikas Perangkat Keras yang Digunakan .....	98
<b>Tabel 4.2</b> Spesifikasi Perangkat Lunak yang Digunakan .....	99
<b>Tabel 4.3</b> Pengujian Halaman <i>Login</i> .....	109
<b>Tabel 4.4</b> Pengujian Halaman <i>Dashboard Artikel</i> .....	110
<b>Tabel 4.5</b> Pengujian Halaman <i>Dashboard Kategori Artikel</i> .....	111
<b>Tabel 4.6</b> Pengujian Halaman <i>Dashboard Profil</i> .....	112
<b>Tabel 4.7</b> <i>Black Box Testing Login Admin</i> .....	115
<b>Tabel 4.8</b> <i>Black Box Testing Kelola Dashboard admin Gizila</i> .....	116
<b>Tabel 4.9</b> <i>Black Box Testing Kelola Manajemen Artikel</i> .....	116
<b>Tabel 4.10</b> <i>Black Box Testing Halaman Manajemen Kategori</i> .....	117
<b>Tabel 4.11</b> <i>Black Box Testing Halaman Logout</i> .....	118
<b>Tabel 4.12</b> <i>Black Box Testing Halaman Profil</i> .....	119
<b>Tabel 4.13</b> <i>Black Box Testing Halaman Website Gizila</i> .....	120
<b>Tabel 4.14</b> Pengujian UAT Admin .....	123
<b>Tabel 4.15</b> Pengujian UAT Publik .....	124
<b>Tabel 4.16</b> Hasil Penilaian Responden <i>Usability Testing User Publik</i> .....	125
<b>Tabel 4.17</b> Hasil Penilaian Responden <i>Usability Testing User Admin</i> .....	126

## DAFTAR GAMBAR

<b>Gambar 2.14</b> Diagram Pengguna <i>Framework PHP</i> Pada Projek Web .....	34
<b>Gambar 2.15</b> Siklus RAD .....	44
<b>Gambar 3.1</b> Identifikasi Proses Bisnis yang Berjalan .....	47
<b>Gambar 3.2</b> Identifikasi Sistem yang Diusulkan.....	52
<b>Gambar 3.3</b> <i>Use Case Diagram</i> .....	56
<b>Gambar 3.4</b> <i>Activity Diagram Login Admin</i> .....	67
<b>Gambar 3.5</b> <i>Activity Diagram Akses Dashboard Admin Gizila</i> .....	68
<b>Gambar 3.6</b> <i>Activity Diagram Kelola Dashboard Admin Gizila</i> .....	69
<b>Gambar 3.7</b> <i>Activity Diagram Kelola Fitur CRUD Artikel di Dashboard</i> .....	70
<b>Gambar 3.8</b> <i>Activity Diagram Kelola Profil</i> .....	71
<b>Gambar 3.9</b> <i>Activity Diagram Kelola Hak Akses</i> .....	72
<b>Gambar 3.10</b> <i>Activity Diagram Kelola Website Gizila</i> .....	73
<b>Gambar 3.11</b> <i>Activity Diagram Akses Website Gizila</i> .....	74
<b>Gambar 3.12</b> <i>Activity Diagram Lihat Fitur-fitur Website Gizila</i> .....	75
<b>Gambar 3.13</b> <i>Activity Diagram Menghitung Gizi Harian</i> .....	76
<b>Gambar 3.14</b> <i>Activity Diagram Lihat Artikel Edukasi</i> .....	77
<b>Gambar 3.15</b> <i>Activity Diagram Mendapat Info Edukasi di Website Gizila</i> .....	78
<b>Gambar 3.16</b> <i>Sequence Diagram Login Admin</i> .....	79
<b>Gambar 3.17</b> <i>Sequence Diagram Akses Dashboard admin Gizila</i> .....	80
<b>Gambar 3.18</b> <i>Sequence Diagram Kelola Dashboard admin Gizila</i> .....	80
<b>Gambar 3.19</b> <i>Sequence Diagram Create</i> .....	81
<b>Gambar 3.20</b> <i>Sequence Diagram Read</i> .....	82
<b>Gambar 3.21</b> <i>Sequence Diagram Update</i> .....	83
<b>Gambar 3.22</b> <i>Sequence Diagram Delete</i> .....	83
<b>Gambar 3.23</b> <i>Sequence Diagram Kelola Profil</i> .....	84
<b>Gambar 3.24</b> <i>Sequence Diagram Kelola Hak Akses Admin</i> .....	85
<b>Gambar 3.25</b> <i>Sequence Diagram Kelola Website Gizila</i> .....	86
<b>Gambar 3.26</b> <i>Sequence Diagram Akses Website Gizila</i> .....	86
<b>Gambar 3.27</b> <i>Sequence Diagram Lihat Fitur-fitur Website Gizila</i> .....	87
<b>Gambar 3.28</b> <i>Sequence Diagram Menghitung Gizi Harian</i> .....	88
<b>Gambar 3.29</b> <i>Sequence Diagram Lihat Artikel Edukasi</i> .....	89
<b>Gambar 3.30</b> <i>Sequence Diagram Mendapat Info Edukasi di Website Gizila</i> .....	89
<b>Gambar 3.31</b> <i>Class Diagram</i> .....	90
<b>Gambar 3.32</b> Desain Halaman <i>Login Admin</i> .....	91
<b>Gambar 3.33</b> Desain Halaman Akses <i>Dashboard</i> admin Gizila .....	91
<b>Gambar 3.34</b> Desain Halaman Kelola <i>Dashboard</i> admin Gizila .....	92
<b>Gambar 3.35</b> Desain Halaman Kelola Fitur CRUD Artikel di <i>Dashboard</i> .....	92
<b>Gambar 3.36</b> Desain Halaman Kelola <i>Website Gizila</i> .....	93

<b>Gambar 3.37</b> Desain Halaman Akses <i>Website</i> Gizila .....	93
<b>Gambar 3.38</b> Desain Halaman Lihat Fitur-fitur <i>Website</i> Gizila .....	94
<b>Gambar 3.39</b> Desain Halaman Fitur Hitung Gizi Harian 1 .....	94
<b>Gambar 3.40</b> Desain Halaman Fitur Hitung Gizi Harian 2 .....	95
<b>Gambar 3.41</b> Desain Halaman Fitur Hitung Gizi Harian 3 .....	95
<b>Gambar 3.42</b> Desain Halaman Lihat Artikel Edukasi .....	96
<b>Gambar 3.43</b> Desain Halaman Mendapat Info Edukasi di <i>Website</i> Gizila .....	97
<b>Gambar 3.44</b> Desain Halaman <i>About</i> .....	97
<b>Gambar 4.1</b> Implementasi Halaman <i>Login</i> Admin .....	99
<b>Gambar 4.2</b> Implementasi Halaman Akses <i>Dashboard</i> admin Gizila.....	100
<b>Gambar 4.3</b> Implementasi Halaman Kelola <i>Dashboard</i> admin Gizila .....	100
<b>Gambar 4.4</b> Implementasi Halaman Kelola Fitur CRUD Artikel di <i>Dashboard</i> .....	101
<b>Gambar 4.5</b> Implementasi Halaman Kelola Profil di <i>Dashboard</i> .....	101
<b>Gambar 4.6</b> Implementasi Halaman Kelola Hak Akses di <i>Dashboard</i> .....	102
<b>Gambar 4.7</b> Implementasi Halaman Kelola <i>Website</i> Gizila .....	102
<b>Gambar 4.8</b> Implementasi Halaman Akses <i>Website</i> Gizila.....	103
<b>Gambar 4.9</b> Implementasi Halaman Fitur-fitur <i>Website</i> Gizila.....	103
<b>Gambar 4.10</b> Implementasi Halaman Fitur Hitung Gizi Harian 1.....	104
<b>Gambar 4.11</b> Implementasi Halaman Fitur Hitung Gizi Harian 2.....	104
<b>Gambar 4.12</b> Implementasi Halaman Fitur Hitung Gizi Harian 3.....	105
<b>Gambar 4.13</b> Implementasi Halaman Fitur Hitung Gizi Harian 4.....	106
<b>Gambar 4.14</b> Implementasi Halaman Fitur Hitung Gizi Harian 5.....	106
<b>Gambar 4.15</b> Implementasi Halaman Lihat Artikel Edukasi .....	107
<b>Gambar 4.16</b> Implementasi Halaman Mendapat Info Edukasi di <i>Website</i> Gizila .....	108
<b>Gambar 4.17</b> Halaman <i>About</i> .....	109
<b>Gambar 4.18</b> Halaman <i>Home Dashboard</i> Admin Gizila .....	113
<b>Gambar 4.19</b> Halaman <i>Dashboard</i> Manajemen Artikel.....	113
<b>Gambar 4.20</b> Halaman <i>Dashboard</i> Manajemen Kategori Artikel.....	114
<b>Gambar 4.21</b> Halaman <i>Dashboard</i> Profil Admin .....	114
<b>Gambar 4.22</b> Halaman <i>Dashboard</i> Hak Akses Admin .....	115
<b>Gambar 4.23</b> Standar Perolehan Skor SUS .....	126

## ABSTRAK

Hal yang melatarbelakangi Penelitian ini bertujuan untuk memberikan kontribusi yang signifikan dalam pengembangan sistem informasi gizila berbasis web untuk kebutuhan bisnis Gizila yang bergerak dibidang *creative agency*, yang dapat diakses secara langsung oleh public untuk pemantauan informasi edukasi tentang gizi yang mudah diakses, dan perhitungan gizi harian bagi pengguna. Metode pengembangan sistem yang digunakan adalah *Rapid Application Development* (RAD) karena sifatnya yang iteratif dan cepat. Sistem informasi Gizila ini dikembangkan dengan memanfaatkan MySQL sebagai sistem manajemen basis data, bahasa pemrograman PHP dengan framework Laravel versi 11.45.1, dengan perancangan sistem menggunakan diagram *Unified Modeling Language* (UML) yang mencakup *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*. Angka Metabolisme Basal (AMB) dan kebutuhan kalori harian dihitung dengan metode rumus *Harris-Benedict* untuk menghitung kebutuhan gizi pada sistem ini. Tujuan penelitian ini untuk kebutuhan bisnis Gizila, sebuah *creative agency* di bawah naungan PT Guntara, guna menyediakan layanan edukasi dan perhitungan gizi yang dapat diakses publik secara *real-time* juga untuk kebutuhan media informasi masyarakat yang mudah diakses untuk menghitung gizi harian mengingat metode manual yang ada saat ini di era serba terotomatisasi oleh sistem kurang efisien untuk diterapkan. Berdasarkan hasil penilaian kuesioner dengan metode *System Usability Scale* (SUS) pada web Gizila, diperoleh nilai rata-rata 62,1 dari sisi *user* publik dan diperoleh nilai rata-rata 63,9 dari sisi *user* admin. Meskipun penilaian ini menunjukkan kinerja yang tergolong standar, mayoritas aspek web masih dapat dianggap cukup memadai untuk digunakan.

**Kata Kunci:** Sistem Informasi Gizi, Rumus *Harris-Benedict*, *Rapid Application Development*, *Laravel*, Gizi Harian.

## ***ABSTRACT***

*This research aims to provide a significant contribution to the development of the Gizila web-based information system to support Gizila's business operations as a creative agency. The system is designed to be directly accessible to the public, enabling easy access to educational information on nutrition and facilitating the calculation of users' daily nutritional needs. The development method applied is Rapid Application Development (RAD), chosen for its iterative and fast-paced characteristics. The Gizila information system was developed using MySQL as the database management system, the PHP programming language with the Laravel framework version 11.45.1, and system design based on Unified Modeling Language (UML), which includes use case diagrams, activity diagrams, sequence diagrams, and class diagrams. Basal Metabolic Rate (BMR) and daily calorie requirements are calculated using the Harris-Benedict formula to determine nutritional needs within the system. The objective of this research is to fulfill Gizila's business requirements as a creative agency under PT Guntara, by providing educational and nutritional calculation services that can be accessed publicly in real-time, as well as serving as an information medium for the community to easily calculate daily nutrition. This is particularly important since manual methods are less efficient in today's highly automated era. Based on the evaluation results using the System Usability Scale (SUS), the Gizila website obtained an average score of 62.1 from public users and 63.9 from admin users. Although these results indicate a performance level considered standard, most aspects of the system can still be regarded as sufficiently adequate for use.*

**Keywords:** *Nutrition Information System, Harris-Benedict Formula, Rapid Application Development, Laravel, Daily Nutrition.*

## **BAB I**

### **PENDAHULUAN**

#### **1.1. Latar Belakang**

Manusia dalam kehidupan sehari-hari selalu mengonsumsi makanan untuk memenuhi kebutuhan gizi tubuh. Setiap individu menyesuaikan kebutuhan gizinya dengan usia, berat badan, dan faktor lainnya. Banyak orang makan hanya untuk menghilangkan rasa lapar tanpa memeriksa apakah kebutuhan gizinya sudah terpenuhi atau belum. Padahal, manusia perlu memenuhi kebutuhan tubuhnya dengan mengonsumsi makanan yang bergizi seimbang (Sam'ani dkk., 2023).

Gizi atau nutrisi merupakan komponen utama dalam makanan yang berfungsi sebagai sumber energi, mendukung pertumbuhan tubuh, memelihara dan memperbaiki jaringan tubuh, mengatur metabolisme, dan berperan dalam mekanisme pertahanan tubuh. Komposisi zat gizi setiap jenis makanan berbeda-beda, oleh karena itu pemilihan jenis makanan sesuai dengan kebutuhan sangatlah penting agar tidak melebihi ataupun kurang dari kebutuhan tubuh, jika salah dalam pemilihan makanan yang dikonsumsi, maka dapat menyebabkan timbulnya penyakit baru, seperti kekurangan gizi, obesitas, hipertensi, diabetes, dan penyakit lainnya. Ada banyak metode dan cara untuk menghitung kebutuhan gizi, protein, karbohidrat, dan lemak harian, baik untuk orang sehat maupun sakit, tetapi hal ini tidaklah mudah. Metode *Harris Benedict* adalah salah satu metode yang digunakan untuk menilai kondisi kesehatan gizi seseorang. Metode *Harris Benedict* digunakan untuk memperkirakan Angka Metabolisme Basal (AMB) dan kebutuhan kalori harian individu. Nilai perkiraan AMB ini kemudian dikalikan dengan faktor aktivitas dan faktor stres individu. Selanjutnya, kebutuhan protein, lemak, dan karbohidrat dibagi berdasarkan jumlah kebutuhan energi tersebut, yaitu 15% untuk protein, 20% untuk lemak, dan 65% untuk karbohidrat. Metode ini dipilih karena perhitungannya sudah mencakup perhitungan kebutuhan kalori sementara, perhitungan indeks massa tubuh, perhitungan aktivitas energi, dan memberikan saran tentang jumlah dan jenis makanan serta minuman yang akan dikonsumsi untuk mencapai tubuh yang ideal (Qamaruzzaman dkk., 2022).

Untuk membantu masyarakat memahami cara menghitung kebutuhan gizi dan mendapatkan panduan pola makan gizi yang seimbang, diperlukan sebuah media informasi yang dapat diakses kapan saja dan dimana saja. Saat ini, baik anak-anak, orang dewasa, maupun orang tua semakin akrab dengan penggunaan *smartphone* atau tablet, lewat aplikasi berbasis *website*. *Website* atau situs merupakan kumpulan halaman yang menyajikan informasi dalam bentuk, seperti teks, gambar diam maupun bergerak, animasi, video, suara, atau

kombinasi dari elemen-elemen tersebut. Informasi yang ditampilkan dapat bersifat statis maupun dinamis, dan tersusun dalam struktur yang saling terhubung serta dapat diakses melalui jaringan halaman yang terhubung satu sama lain. *Website* memiliki kelebihan bisa diakses kapan saja dan dimana saja dengan tampilan fitur-fitur yang menarik. Berdasarkan penjelasan mengenai *website*, dapat disimpulkan bahwa *website* adalah sebuah aplikasi berbasis web yang dapat digunakan untuk beragam tujuan, seperti kegiatan pembelajaran, bisnis, promosi penjualan, dan lainnya (Wiarso dan Anwar, 2024).

Berbagai aplikasi mengenai kebutuhan gizi telah banyak dikembangkan dan dihasilkan antara lain: penelitian pertama dilakukan oleh Excel dkk. (2024) dengan judul “Rancang Bangun Sistem Informasi Monitoring Gizi Ibu Hamil dan Anak Berbasis *Website*” penelitian berikutnya dilakukan oleh Yuliana dkk. (2024) dengan judul “Sistem Informasi Kebutuhan Gizi pada Ibu Hamil dengan Metode *Cooper* pada Puskesmas Ulak Karang Berbasis Web” adapun penelitian ketiga dilakukan oleh Azhariyah dkk. (2023) dengan judul “Rancang Bangun Aplikasi Pendataan Status Gizi pada Balita Berbasis Web di Desa Rawalele”.

Penelitian pertama yang dilakukan oleh Excel dkk. (2024), yang mengusung inovasi sistem informasi berbasis web yang difokuskan pada monitoring gizi ibu hamil dan anak dengan pendekatan digitalisasi pencatatan riwayat kesehatan dan perkembangan berat badan secara mandiri melalui *website*. Sistem ini memfasilitasi pengguna seperti ibu hamil, petugas kesehatan, dan admin untuk mencatat, mengakses, serta memantau perkembangan kesehatan secara interaktif dan sistematis. Kebaruan yang diusung penelitian ini terletak pada fokus segmentasi gizi ibu hamil dan anak secara bersamaan, serta penggunaan antarmuka web berbasis peran (*multi-user*), yaitu admin, petugas, dan pasien dengan fitur visualisasi data gizi. Penelitian ini memiliki kelebihan pada aspek kemudahan akses informasi kesehatan dan pemantauan mandiri yang dapat dilakukan oleh pengguna tanpa bergantung pada tenaga medis secara langsung. Dengan integrasi fitur pencatatan riwayat kesehatan, grafik berat badan, serta usia kehamilan, pengguna dapat memperoleh gambaran perkembangan kesehatan secara visual dan aktual. Selain itu, struktur sistem yang dirancang melalui pendekatan UML secara menyeluruh memungkinkan pengembangan sistem yang sistematis dan terdokumentasi dengan baik. Namun demikian, kekurangan dari penelitian ini adalah belum adanya penggunaan standar medis perhitungan gizi seperti metode *Cooper* atau *Harris-Benedict* dan lainnya, sehingga validitas ilmiah dari kalkulasi status gizi belum terverifikasi secara klinis. Selain itu, sistem ini masih terbatas pada pencatatan dan pemantauan, belum menyediakan fitur interaktif seperti konsultasi daring, pengingat otomatis, atau personalisasi rekomendasi gizi berdasarkan data individu, yang dapat memperkaya pengalaman pengguna. Sistem juga tidak dijelaskan telah melalui pengujian *usability* formal seperti SUS atau UEQ, sehingga

aspek *user experience* belum terukur secara kuantitatif. Metode pengembangan sistem yang digunakan dalam penelitian adalah metodologi *Waterfall*. Metode ini diterapkan secara bertahap mulai dari analisis kebutuhan, perancangan sistem dengan UML, implementasi, hingga pengujian. Pendekatan *Waterfall* dipilih karena memberikan struktur kerja yang terarah dan sistematis dalam setiap tahap pengembangan, dengan hasil akhir berupa *website* yang telah diimplementasikan dan diuji secara internal. Penggunaan metode *Waterfall* memberikan kelebihan dalam hal dokumentasi yang lengkap, mudahnya kontrol pengembangan proyek, dan cocok untuk sistem yang kebutuhan fungsionalnya telah didefinisikan sejak awal. Namun, kekurangannya adalah kurangnya fleksibilitas dalam menerima perubahan selama proses berlangsung, serta tidak cocok jika sistem memerlukan iterasi atau umpan balik dari pengguna secara *real-time*, yang justru penting dalam sistem yang menyentuh aspek kesehatan masyarakat.

Kemudian penelitian kedua oleh Yuliana dkk. (2024), yang mengusung sistem informasi kebutuhan gizi pada ibu hamil yaitu menggunakan pendekatan metode *waterfall*, metode ini merupakan pengembangan perangkat lunak yang dilakukan secara bertahap dan sistematis, mulai dari tahap analisis, desain, pengkodean, implementasi, hingga pemeliharaan. Penelitian ini memiliki kelebihan pada sisi akurasi medis karena mengadopsi metode *cooper* sebagai dasar perhitungan kebutuhan energi, serta pengembangan sistem yang tertata rapi melalui model *waterfall*. Namun, kekurangannya adalah minimnya fitur interaktif dan tidak adanya evaluasi terkait pengalaman pengguna (*usability* dan UX). Sasaran dan fitur yang dikembangkan ditujukan untuk membantu ibu hamil dalam menghitung kebutuhan kalori dan gizi menggunakan metode *cooper*, yang menghitung kebutuhan energi berdasarkan berat badan ideal, usia kehamilan, jam tidur, dan aktivitas fisik.

Adapun penelitian ketiga oleh Azhariyah dkk. (2023), yang mengusung rancang bangun aplikasi pendataan status gizi pada balita berbasis web yaitu menggunakan pendekatan metode *prototype* cepat, kemudian diperbaiki berdasarkan masukan pengguna hingga menjadi sistem akhir. Penelitian ini memiliki kelebihan dalam kemudahan penggunaan, sangat sesuai untuk diterapkan di wilayah pedesaan yang masih menggunakan pencatatan manual. Sistem ini sangat membantu kader posyandu dalam mempercepat proses pencatatan dan pengiriman data ke puskesmas. Namun, kekurangannya terletak pada tidak adanya fitur konsultasi gizi, metode perhitungan status gizi yang eksplisit, serta belum dilakukan pengujian *usability* terhadap pengguna akhir. Sasaran dan fitur yang dikembangkan ditujukan pada pendataan status gizi balita di desa, dengan sistem yang sederhana, praktis, dan mudah digunakan oleh kader posyandu, namun tanpa fitur konsultasi atau kalkulasi gizi mendalam.

Sementara itu, metode *prototype* yang digunakan dalam penelitian Azhariyah dkk. (2023), meskipun metode ini memberikan kemudahan dalam hal pengembangan sistem yang cepat dan memungkinkan revisi berdasarkan umpan balik pengguna, pendekatan ini cenderung menghasilkan dokumentasi teknis yang kurang lengkap dan struktur sistem yang tidak matang. Prototipe yang dibangun untuk tujuan demonstrasi atau evaluasi awal belum tentu dapat langsung diimplementasikan sebagai sistem final tanpa perombakan besar. Di samping itu, pengguna sering kali menganggap prototipe sebagai versi akhir dari sistem, padahal prototipe tersebut mungkin masih memiliki banyak keterbatasan. Metode ini juga kurang cocok untuk pengembangan sistem berskala besar atau yang membutuhkan keamanan dan stabilitas tinggi sejak awal, karena sifatnya yang eksperimental dan cepat berubah.

Dengan demikian, masing-masing metode pengembangan sistem memiliki keterbatasan yang perlu dipertimbangkan secara matang. Dalam analisis terhadap tiga penelitian yang telah dijelaskan diatas, yang berfokus pada sistem informasi gizi, terdapat kesamaan bahwa ketiganya sama-sama mengembangkan aplikasi gizi yang bertujuan untuk membantu masyarakat atau tenaga kesehatan dalam memantau, menghitung, mengelola status atau kebutuhan gizi, membantu peningkatan pelayanan gizi dan pemantauan status gizi untuk kelompok masyarakat tertentu baik itu pada ibu hamil maupun balita. Namun, terdapat perbedaan mendasar dalam pendekatan metode pengembangan sistem (SDLC) yang digunakan, juga terdapat perbedaan sasaran dan fitur yang dikembangkan.

Penelitian sistem informasi Gizila ini menggunakan metode pengembangan sistem model RAD (*Rapid Application Development*). Metodologi ini adalah metode inkremental yang diterapkan dalam pengembangan perangkat lunak, dengan penekanan pada fase kemajuan pengembangan sistem. Metode ini membagi setiap tahap menjadi komponen-komponen kecil, dan setiap prosesnya dibangun dengan pendekatan yang serupa dengan metode air terjun atau *waterfall*. Tujuannya adalah untuk mengembangkan sistem secara bertahap, meskipun perencanaan awal dilakukan secara komprehensif. Model *waterfall* tidak sesuai untuk pengembangan kompleks karena tahapannya tidak dapat diulang, memerlukan waktu pengerjaan yang signifikan, dan biayanya tinggi. Di sisi lain, model RAD sangat sesuai untuk proyek berskala besar karena menerapkan metode iteratif yang dapat mengurangi biaya (Murdiani dan Hermawan, 2022).

Penelitian ini berbeda karena sistem informasi Gizila mengimplementasikan metode perhitungan *Harris Benedict* untuk memperkirakan Angka Metabolisme Basal (AMB) dan kebutuhan kalori harian individu dari berbagai usia. Berdasarkan pemikiran tersebut, penulis berencana untuk melakukan penelitian guna menerapkan metode perhitungan *Harris Benedict* dalam sebuah aplikasi sistem berbasis web yang menyediakan informasi mengenai kebutuhan

gizi harian. Aplikasi berbasis web ini menerapkan metode pengembangan sistem menggunakan model RAD (*Rapid Application Development*) dan berbasis web, sehingga dapat diakses oleh siapa saja, kapan saja, dan di mana saja.

Penelitian ini bertujuan untuk memberikan kontribusi signifikan terhadap pengembangan sistem informasi Gizila berbasis web untuk kebutuhan bisnis Gizila yang beroperasi di bidang agensi kreatif, yang dapat diakses secara *real-time* oleh publik untuk pemantauan informasi edukatif mengenai gizi yang mudah diakses, serta perhitungan gizi harian bagi pengguna, dan menunjukkan penerapan metode RAD secara efektif dalam pengembangan sistem informasi Gizila.

### **1.2. Rumusan Masalah**

Berikut adalah rumusan masalah yang menjadi fokus utama penelitian ini:

1. Bagaimana kebutuhan pengguna dalam perencanaan gizi harian dapat diidentifikasi dan diakomodasi oleh sistem informasi berbasis web?
2. Bagaimana perencanaan gizi harian, termasuk jenis informasi gizi yang diperlukan, fitur yang diinginkan, dan antarmuka pengguna yang mudah digunakan?
3. Bagaimana metode RAD (*Rapid Application Development*) dapat diterapkan secara efektif dalam pengembangan sistem informasi gizila untuk memenuhi kebutuhan pengguna?

### **1.3. Batasan Masalah**

Berdasarkan uraian latar belakang serta identifikasi masalah yang telah dijelaskan sebelumnya, penulis menetapkan sejumlah batasan agar ruang lingkup penelitian menjadi lebih terfokus dan pembahasan tidak meluas. Penelitian ini memiliki batasan sebagai berikut:

1. Penelitian ini tidak mencakup penggunaan oleh pakar kesehatan seperti dokter atau pakar ahli gizi secara professional.
2. Rancang bangun sistem informasi Gizila menggunakan metode pengembangan sistem RAD (*Rapid Application Development*).
3. Bahasa pemrograman yang digunakan adalah PHP versi 8.2.12 dengan *framework* Laravel versi 11.45.1, dan sistem manajemen basis data yang digunakan adalah MySQL versi 10.4.32.

### **1.4. Tujuan Penelitian**

1. Untuk mengkaji bagaimana sistem informasi yang dirancang dapat memberikan rekomendasi gizi harian yang akurat berdasarkan data pengguna, memantau asupan

gizi harian, dan memberikan saran untuk memastikan pengguna mencapai target gizi harian berdasarkan perhitungan rumus *Harris Benedict*.

2. Untuk mengidentifikasi kebutuhan pengguna dalam perencanaan dan pengelolaan gizi harian, serta mengembangkan fitur-fitur yang sesuai untuk memenuhi kebutuhan tersebut. Fitur-fitur ini termasuk perhitungan indeks masa tubuh, perhitungan gizi harian untuk memenuhi kebutuhan gizi berdasarkan data pengguna yang menggunakan perhitungan rumus *Harris Benedict*, serta rekomendasi gizi harian yang berupa edukasi informasi berupa artikel Gizila.
3. Pada penerapan metode RAD dalam pengembangan sistem informasi, ada tahapan RAD seperti perencanaan kebutuhan, desain prototipe, implementasi, pemeliharaan dan evaluasi diterapkan untuk menghasilkan sistem yang efisien.

### **1.5. Manfaat Penelitian**

Melalui penelitian ini, penulis berharap dapat memberikan manfaat kepada beberapa pihak, antara lain:

- 1) Mahasiswa
  - a. Meningkatkan pemahaman mengenai perancangan sistem informasi perhitungan gizi harian berbasis web yang telah banyak digunakan oleh perusahaan untuk keperluan promosi bisnis.
  - b. Memberikan pengalaman dalam merancang bangun sistem informasi perhitungan gizi harian berbasis web yang dapat diakses melalui internet.
  - c. Menyelesaikan tugas akhir perkuliahan pada jenjang pendidikan strata 1 (S-1).
- 2) Universitas Nahdlatul Ulama Indonesia (UNUSIA)
  - a. Penelitian ini dapat menjadi tambahan untuk referensi studi ilmiah bagi universitas di bidang sistem informasi gizi untuk kebutuhan gizi harian.
  - b. Penelitian ini dapat memberikan kontribusi dalam penerapan metode *Rapid Application Development* untuk merancang dan mengembangkan *software*.
  - c. Menjadi bahan ajar terkait sistem informasi gizi bagi mahasiswa.
- 3) PT GUNTARA  
Penelitian dengan judul "Rancang Bangun Sistem Informasi Gizila untuk Kebutuhan Gizi Harian Berbasis Web Menggunakan Metode *Rapid Application Development* (RAD)" ini bertujuan untuk memberikan manfaat yang signifikan bagi PT. GUNTARA khususnya untuk unit bisnis Gizila yang bergerak di bidang *creative agency*.
- 4) Masyarakat atau Publik

Sistem informasi Gizila menyediakan *platform* yang mudah diakses bagi siapa saja dan dimana saja secara *real-time* untuk memperoleh informasi mengenai status gizi (IMT) dan estimasi kebutuhan kalori harian. Masyarakat tidak lagi perlu mencari dan menggunakan rumus-rumus kompleks secara manual, karena sistem ini menyajikannya secara otomatis, gratis, dan instan, sehingga ilmu gizi yang bersifat teknis menjadi lebih mudah dipahami dan dijangkau oleh semua kalangan usia.

#### 1.6. Metode Penelitian

Metode yang digunakan dalam penelitian ini mencakup data yang berkaitan dengan kebutuhan gizi harian, standar nutrisi, dan informasi pengguna yang diperlukan dengan menggunakan perhitungan rumus *Harris Benedict* untuk membuat *logic* pada pengembangan sistem. Data ini diperoleh dari sumber-sumber terpercaya seperti jurnal ilmiah. Metode yang digunakan dalam proses pengumpulan data adalah seperti berikut:

##### 1) Metode Pengumpulan Data

Untuk memperoleh data yang dibutuhkan dalam penelitian ini, digunakan beberapa metode pengumpulan data sebagai berikut:

###### a. Studi Literatur

Metode ini dilakukan dengan mengumpulkan data dan informasi yang relevan dari berbagai sumber tertulis, seperti buku, jurnal ilmiah, artikel, serta referensi lain yang mendukung teori dan landasan penelitian terkait rancang bangun sistem informasi berbasis web menggunakan metode *Rapid Application Development*.

###### b. Observasi

Observasi, atau pengamatan, adalah metode pengumpulan data dengan cara terjun langsung ke lapangan untuk melihat dan mengamati objek penelitian secara langsung. Dalam penelitian ini, penulis melakukan observasi dengan mengamati secara langsung kepuasan pengguna terhadap akun media sosial Gizila yang memberikan informasi edukatif terkait kesehatan dan asupan gizi seimbang untuk kebutuhan sehari-hari.

###### c. Kuesioner

Metode berikutnya adalah kuesioner, yaitu teknik pengumpulan data yang dilakukan dengan memberikan perangkat atau tautan berisi pertanyaan tertulis kepada responden untuk dijawab secara online. Kuesioner ini telah disebarluaskan kepada responden dengan target sampel yaitu *stakeholder* terkait di PT GUNTARA dengan jumlah 7 responden. Jenis angket yang digunakan adalah angket tertutup, di mana jawaban telah disediakan, dan responden hanya perlu memilih salah satu jawaban yang tersedia.

Peneliti sebelumnya sudah melakukan penelitian tentang Gizila yaitu “Penerapan Metode *Pieces* Untuk Menganalisis Tingkat Kepuasan *Followers* Akun Instagram Gizila Berupa *Platform* Informasi Kebutuhan Gizi Harian di PT Graisen Usaha Nusantara”. Berdasarkan hasil perhitungan dari pernyataan yang disebarluaskan kepada responden pengguna sebagai *followers* Instagram Gizila, dengan variabel indikator (*Performance, Information, Economics, Efficiency, Service*), berdasarkan parameter *User Satisfaction* atau kepuasan pengguna diperoleh nilai rata-rata sebesar 3.27 yang termasuk dalam kategori puas. Hal ini menunjukkan bahwa penerapan sistem informasi yang diakses oleh pengguna sebagai *followers* Instagram Gizila, berdasarkan *framework* PIECES telah berjalan dengan baik.

Namun, informasi yang dibagikan kepada publik tetap perlu ditingkatkan lagi kedepannya agar tetap kompetitif dan tidak tertinggal mengingat Gizila masih belum dijangkau luas oleh kalangan publik dan sedang pada tahap berkembang. Oleh karena itu, penulis mengembangkan ide ini dengan membuat rancangan bangun sistem informasi Gizila untuk kebutuhan bisnis Gizila yang bergerak dibidang *Creative Agency* dibawah naungan PT Graisen Usaha Nusantara. Dengan adanya rancangan bangun sistem informasi Gizila ini, bertujuan untuk meningkatkan *audiens* yang lebih banyak, serta tetap konsisten memberikan informasi edukatif tentang kesehatan dan asupan gizi yang seimbang dalam kehidupan sehari-hari untuk mengedukasi masyarakat melalui *platform* website Gizila yang memberikan fitur-fitur menarik diantaranya mencakup kalkulator perhitungan indeks masa tubuh, perhitungan kebutuhan gizi harian berdasarkan data pengguna (seperti usia, jenis kelamin, berat badan, dan tingkat aktivitas), serta rekomendasi asupan gizi harian yang seimbang, dan informasi edukatif mengenai asupan gizi seimbang, hal ini bertujuan agar menarik perhatian *audiens* setia yang bisa diakses secara *realtime*. Sehingga poin-poin diatas yang telah dijelaskan sebelumnya menjawab rumusan-rumusan masalah terkait penelitian ini yaitu untuk memberikan kontribusi yang signifikan dalam pengembangan sistem informasi gizila berbasis web guna kebutuhan bisnis Gizila yang bergerak dibidang *Creative Agency* dan juga untuk perencanaan dan pemantauan gizi harian, serta menunjukkan bagaimana metode RAD dapat diterapkan secara efektif dalam pengembangan sistem informasi Gizila.

## 2) Metode *Rapid Application Development* (RAD)

*Rapid Application Development* (RAD) adalah metode pengembangan perangkat lunak yang fokus pada kecepatan dalam proses pembuatan sistem melalui pendekatan iteratif. Pendekatan ini mengedepankan keterlibatan pengguna secara langsung di setiap tahap pengembangan, dengan tujuan agar sistem yang dihasilkan dapat lebih akurat memenuhi kebutuhan dan harapan pengguna akhir (Nuridza dkk., 2025).

*Rapid Application Development* (RAD) berfokus pada pembangunan yang cepat dan tepat karena ini adalah proses yang sangat penting bagi metodologi ini.

Tahapan-tahapan dalam RAD adalah sebagai berikut:

a. *Business Modelling*:

Proses ini bertujuan untuk memahami kebutuhan pengguna dan mengatur proses bisnis, menentukan sistem yang akan dihasilkan, sumber informasi, serta siapa yang berhak mengelola informasi tersebut

b. *Dokumentasi modelling*:

Proses ini bertujuan untuk mengetahui data yang didapat dan dibutuhkan untuk membuat prototipe yang sesuai dengan keinginan pengguna.

c. *Proses Modelling*:

Proses ini mendefinisikan model data dan memodifikasinya untuk memenuhi kebutuhan pengguna dalam proses bisnis.

d. *Application generation*:

RAD menggunakan teknik generasi ke-4. Proses ini mengutamakan penggunaan komponen alat yang telah ada, dan jika diperlukan, membuat komponen baru. Kebanyakan kasus menggunakan alat bantu otomatis untuk membangun perangkat lunak.

e. *Testing dan turnover*:

Proses ini mengutamakan penggunaan kembali komponen program yang telah diuji untuk mempersingkat waktu pengujian, namun komponen baru diuji secara menyeluruh.

3) Metode Perancangan

Perancangan sistem dilakukan dengan pendekatan pemodelan berorientasi objek menggunakan *Unified Modeling Language* (UML). Pada penelitian ini menggunakan *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram*.

4) Metode Pengujian

Pengujian sistem dalam penelitian ini dilakukan menggunakan metode *black box testing*, yaitu metode pengujian yang berfokus pada fungsionalitas sistem tanpa melihat struktur internal kode program. Pengujian ini bertujuan untuk memastikan bahwa setiap fitur dalam sistem telah berjalan sesuai dengan yang diharapkan berdasarkan kebutuhan pengguna.

## 1.7. Sistematika Penulisan

Sistem penulisan pada tugas akhir ini disusun dalam bentuk karya ilmiah dengan menyesuaikan panduan yang diterbitkan oleh Universitas Nahdlatul Ulama Indonesia. Adapun urutan sistem penulisan pada karya tulis ilmiah ini sebagai berikut.

### BAB I PENDAHULUAN

Bab ini membahas bagian awal dari laporan penelitian yang meliputi latar belakang permasalahan, rumusan masalah, batasan masalah, tujuan penelitian, manfaat yang dihasilkan dari hasil penelitian, metode pengembangan sistem yang digunakan yaitu *Rapid Application Development* (RAD), serta sistematika penulisan. Tujuan dari bab ini adalah memberikan gambaran umum mengenai arah dan ruang lingkup penelitian.

### BAB II LANDASAN TEORI

Bab ini berisi kajian pustaka yang terdiri dari dua bagian utama, yaitu: ada penelitian terdahulu, yang membandingkan penelitian ini dengan studi sebelumnya agar terlihat kontribusi baru yang diberikan. Kemudian ada landasan teori, yang menjelaskan istilah-istilah penting yang ada terkait topik penelitian ini.

### BAB III ANALISIS DAN PERANCANGAN SISTEM

Bab ini berisi analisis kebutuhan pengguna sistem informasi gizi harian berbasis web. Pembahasan mencakup identifikasi kebutuhan fungsional dan non-fungsional, serta perancangan sistem yang meliputi pembuatan diagram UML (*Use Case, Activity, Sequence, dan Class Diagram*), dan desain antarmuka pengguna.

### BAB IV TESTING DAN IMPLEMENTASI

Bab ini berisi tahapan implementasi dari sistem informasi Gizila yang telah dirancang, termasuk penerapan antarmuka dan logika sistem pada *platform* berbasis web. Selain itu, dilakukan pengujian sistem menggunakan metode *black box testing, user acceptance testing* (UAT) dan *system usability scale* (SUS) untuk memastikan bahwa semua fungsi sistem berjalan sesuai dengan kebutuhan pengguna.

### BAB V PENUTUP

Bab ini berisi kesimpulan dari hasil penelitian dan implementasi sistem yang telah dilakukan. Selain itu, disampaikan pula saran yang dapat menjadi acuan untuk pengembangan sistem lebih lanjut di masa depan.

### DAFTAR PUSTAKA

Berisi kumpulan referensi yang digunakan selama proses penyusunan skripsi, baik dari buku, jurnal, maupun artikel ilmiah.

**LAMPIRAN**

Berisi dokumen pendukung seperti tabel, tangkapan layar, foto, serta data lainnya yang mendukung isi penelitian ini.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1. Penelitian Terdahulu

Pada penelitian aplikasi kebutuhan gizi harian telah banyak di analisis pada penelitian sebelumnya. Penulis melampirkan penelitian terdahulu untuk membandingkan dengan penelitian terbaru mengenai rancang bangun sistem informasi Gizila berbasis web dengan metode *Rapid Application Development* (RAD). Penulis menggunakan penelitian terdahulu tersebut sebagai referensi agar penelitian ini bermanfaat bagi peneliti selanjutnya yang meneliti topik serupa. Selain itu, penulis melampirkan penelitian terdahulu guna menghindari terjadinya plagiasi dari penelitian sebelumnya. Berdasarkan hal tersebut, berikut adalah beberapa penelitian yang relevan dengan topik penelitian ini dilampirkan dalam tabel sebagai berikut.

**Tabel 2.1** Perbandingan Penelitian Terdahulu

Judul	Metode	Framework	Kelemahan	Perbedaan
Rancang Bangun Sistem Informasi Gizi Ibu Hamil dan Anak Berbasis Website	<i>Waterfall</i>	Perancangan sistem pada penelitian ini tidak menggunakan <i>framework</i> , melainkan dibangun dengan <i>PHP Native</i>	Sistem yang dirancang belum menerapkan perhitungan kebutuhan gizi harian yang eksplisit tidak menggunakan metode perhitungan tertentu.	Sistem yang dirancang sudah menerapkan fitur perhitungan kebutuhan gizi harian yang jelas dengan menerapkan metode perhitungann menggunakan rumus <i>Harris-Benedict</i> .
Sistem Informasi Kebutuhan Gizi pada Ibu Hamil dengan Metode Cooper pada	<i>Waterfall</i>	Perancangan sistem pada penelitian ini tidak menggunakan <i>framework</i> , melainkan	Sistem hanya terbatas pada perhitungan kebutuhan kalori ibu hamil dengan metode <i>Cooper</i> , tanpa adanya fitur	Sistem yang dirancang sudah menerapkan fitur perhitungan gizi harian serta fitur menu makanan yang dapat disesuaikan

Puskesmas Ulak Karang Berbasis Web		dibangun dengan <i>PHP Native</i>	konsumsi gizi maupun pencatatan menu untuk kebutuhan gizi harian.	dengan kebutuhan pengguna secara detail dengan hasil perhitungan IMT.
Rancang Bangun Aplikasi Pendataan Status Gizi pada Balita Berbasis Web di Desa Rawalele	<i>Prototype</i>	<i>Laravel</i>	Sistem yang dirancang belum menerapkan fitur perhitungan gizi mendalam dan belum menerapkan cakupan penggunaan yang lebih luas di luar Desa Rawalele.	Sistem yang dirancang sudah menerapkan fitur perhitungan gizi mendalam dengan menggunakan rumus perhitungan <i>Harris-Benedict</i> dan sudah menerapkan cakupan penggunaan yang lebih luas yaitu publik yang mengakses web Gizila yang bisa diakses secara <i>realtime</i> .

Berdasarkan tabel perbandingan penelitian terdahulu, dapat dijelaskan bahwa penelitian pertama dilakukan oleh Excel dkk. (2024) dengan judul “Rancang Bangun Sistem Informasi Monitoring Gizi Ibu Hamil dan Anak Berbasis *Website*” penelitian berikutnya dilakukan oleh Yuliana dkk. (2024) dengan judul “Sistem Informasi Kebutuhan Gizi pada Ibu Hamil dengan Metode *Cooper* pada Puskesmas Ulak Karang Berbasis Web” adapun penelitian ketiga dilakukan oleh Azhariyah dkk. (2023) dengan judul “Rancang Bangun Aplikasi Pendataan Status Gizi pada Balita Berbasis Web di Desa Rawalele”.

Berdasarkan kelemahan penelitian terdahulu tersebut, penelitian ini mengatasinya dengan menerapkan perhitungan kebutuhan gizi harian secara detail menggunakan rumus perhitungan *Harris-Benedict* yang terintegrasi dalam sistem. Selain itu, sistem yang dikembangkan sudah mendukung cakupan penggunaan yang lebih luas bagi publik yang megakses web Gizila dengan *platform* berbasis web yang dapat diakses secara *realtime*. Dengan demikian, penelitian ini memiliki nilai tambah dibandingkan penelitian terdahulu,

karena tidak hanya berfungsi sebagai media pendataan gizi, tetapi juga memberikan layanan perhitungan gizi harian yang lebih komprehensif dan praktis baik dari sisi fitur yang dikembangkan maupun dari aspek kemudahan akses pengguna.

Kelemahan dari masing-masing penelitian terdahulu di atas dapat diatasi dengan mengimplementasikan fitur atau fungsi yang belum diterapkan sebelumnya. Misalnya, sistem yang dirancang dapat menambahkan perhitungan kebutuhan gizi harian secara detail menggunakan rumus tertentu seperti perhitungan rumus *Harris-Benedict*, serta memperluas cakupan penggunaan agar tidak terbatas pada wilayah tertentu saja. Selain itu, pada penelitian ini dapat dilengkapi dengan *uji usability* agar dapat memastikan kemudahan penggunaan bagi pengguna akhir.

Dengan demikian, kelebihan dan kekurangan dari masing-masing penelitian terdahulu dicantumkan guna menjadi perbedaan yang jelas antara sistem yang dirancang pada penelitian terdahulu dengan sistem yang dikembangkan dalam penelitian ini dan juga memberikan gambaran mengenai keterbatasan yang masih ada, sekaligus menunjukkan bagaimana penelitian ini dapat menyempurnakan kekurangan tersebut. Hasil sistem yang dirancang dalam penelitian ini dapat memberikan nilai tambah dibandingkan penelitian terdahulu sebelumnya, karena tidak hanya berfungsi sebagai media pendataan, tetapi juga sebagai alat bantu perhitungan kebutuhan gizi harian yang lebih lengkap beserta penggunaan rumus perhitungan dan dapat diakses secara luas oleh publik secara *realtime*. Adapun rincian kelebihan dan kekurangan penelitian terdahulu dapat dilihat pada tabel berikut.

**Tabel 2.2** Kelebihan dan Kekurangan Penelitian Terdahulu

No	Judul	Kelebihan	Kekurangan
1	Rancang Bangun Sistem Informasi <i>Monitoring Gizi Ibu Hamil dan Anak Berbasis Website</i>	Dapat mengembangkan sistem dengan tahapan yang jelas dan terstruktur sehingga memudahkan proses pembangunan sistem.	Melakukan perubahan sistem cukup sulit jika sudah masuk ke tahap implementasi.
		Dapat menyusun dokumentasi secara rapi karena setiap tahap pengembangan tercatat dengan baik.	Melakukan pengembangan membutuhkan waktu lebih lama karena harus menyelesaikan satu tahap sebelum lanjut ke tahap berikutnya.

2	Sistem Informasi Kebutuhan Gizi pada Ibu Hamil dengan Metode <i>Cooper</i> pada Puskesmas Ulak Karang Berbasis Web	Dapat memetakan kebutuhan gizi ibu hamil secara lebih terarah dengan bantuan metode <i>Cooper</i> .	Melakukan perubahan kebutuhan sistem sulit dilakukan karena metode masih kaku.
		Dapat mempermudah pencatatan dan pengolahan data gizi dengan sistem berbasis web.	Melakukan pengembangan dengan <i>PHP Native</i> kurang fleksibel untuk fitur lanjutan dibanding <i>framework</i> modern.
3	Rancang Bangun Aplikasi Pendataan Status Gizi pada Balita Berbasis Web di Desa Rawalele	Dapat mempercepat proses pengembangan dengan melibatkan pengguna sejak tahap awal.	Melakukan pengembangan membutuhkan komunikasi intensif dengan pengguna sehingga memakan waktu jika umpan balik kurang jelas.
		Dapat mempermudah pengelolaan kode program karena <i>framework Laravel</i> mendukung struktur kode yang rapi dan terstandar.	Melakukan iterasi berulang berpotensi menambah biaya dan memperpanjang waktu pengembangan.

Secara keseluruhan, ketiga penelitian terdahulu diatas, memberikan kontribusi penting dalam pengembangan sistem informasi gizi berbasis teknologi, namun dengan pendekatan yang berbeda sesuai dengan kebutuhan dan segmentasi pengguna masing-masing. Meskipun ketiga metode pengembangan sistem yang digunakan dalam masing-masing tiga penelitian terdahulu yang telah dijelaskan diatas memiliki kelebihan dan kontribusi penting dalam pengembangan sistem informasi gizi, masing-masing juga memiliki kekurangan yang perlu diperhatikan sebelum diterapkan dalam penelitian lebih lanjut.

Berdasarkan penelitian terdahulu yang telah diuraikan di atas, penulis mengajukan sebuah sistem informasi Gizila untuk kebutuhan gizi harian berbasis *website* dengan penggunaan bahasa pemrograman PHP, database MySQL dengan server menggunakan XAMPP, serta *framework* Laravel dengan memiliki dukungan arsitektur *Model View Controller* yang memudahkan pengelolaan data. Selain itu, metode pengembangan perangkat lunak yang digunakan pada penelitian ini adalah *Rapid Application Development* (RAD)

karena dinilai lebih efisien dalam proses pengembangan sistem, memungkinkan prototipe cepat dihasilkan, serta lebih fleksibel dalam menghadapi perubahan kebutuhan pengguna.

## 2.2. Landasan Teori

### 2.2.1. Rancang Bangun

Rancang bangun adalah tahapan setelah analisis dalam siklus pengembangan sistem. Tahap ini melibatkan pendefinisian kebutuhan fungsional dan menggambarkan bagaimana sistem akan dibentuk. Proses ini mencakup penggambaran, perencanaan, dan pembuatan sketsa atau pengaturan elemen-elemen yang terpisah menjadi satu kesatuan yang utuh dan berfungsi. Rancang bangun juga mencakup konfigurasi komponen perangkat keras dan perangkat lunak dari suatu sistem (Mulyanto dkk., 2020).

Perancangan adalah aspek penting dalam pembuatan program. Tujuan utamanya adalah memberikan gambaran yang jelas dan lengkap kepada pemrogram dan teknisi yang terlibat. Proses perancangan harus berguna dan mudah dipahami agar dapat digunakan dengan efisien. Perancangan atau desain merupakan serangkaian prosedur untuk menerjemahkan hasil analisis sistem ke dalam bahasa pemrograman, mendeskripsikan secara detail bagaimana komponen-komponen sistem tersebut dapat diimplementasikan (Novitasari dkk., 2021).

### 2.2.2. Figma

Figma adalah aplikasi desain grafis berbasis web yang digunakan untuk merancang antarmuka pengguna (UI) serta pengalaman pengguna (UX) secara interaktif. Aplikasi ini banyak diminati oleh desainer UI/UX karena mendukung kolaborasi tim secara langsung (*real-time*) dan dilengkapi dengan berbagai fitur yang mempermudah proses perancangan. Keunggulan lain dari Figma terletak pada kemudahan akses dan penggunaannya, sebab aplikasi ini dapat dijalankan langsung melalui browser tanpa harus diunduh atau dipasang terlebih dahulu (Surianto dkk., 2023).

Figma merupakan salah satu *design tool* yang digunakan untuk merancang tampilan antarmuka pengguna (*user interface*) pada aplikasi berbasis *website*, *mobile*, desktop, dan *platform* lainnya. Aplikasi ini tersedia dalam versi desktop maupun berbasis daring (Putri dkk., 2023).

### 2.2.3. Internet

Secara teori, *Interconnected Network* atau yang disebut *Internet* merupakan jaringan komputer global yang saling terhubung melalui berbagai node server di seluruh dunia. Berdasarkan definisi yang disampaikan oleh *U.S. Federal Networking Council* pada tahun

1995, *Internet* diartikan sebagai sebuah sistem informasi global yang saling terkoneksi secara logis melalui ruang alamat unik yang berbasis pada *Internet Protocol* (IP) maupun variasinya. Sistem ini mendukung komunikasi menggunakan *Transmission Control Protocol/Internet Protocol* (TCP/IP) beserta pengembangannya, serta menyediakan layanan komunikasi pada lapisan jaringan dan infrastruktur terkait yang dibutuhkan dalam proses tersebut (Noorikhsan dkk., 2023).

#### **2.2.4. Website**

*Website* atau situs merupakan kumpulan halaman yang menyajikan informasi dalam bentuk, seperti teks, gambar diam maupun bergerak, animasi, video, suara, atau kombinasi dari elemen-elemen tersebut. Informasi yang ditampilkan dapat bersifat statis maupun dinamis, dan tersusun dalam struktur yang saling terhubung serta dapat diakses melalui jaringan halaman yang terhubung satu sama lain (Wiarso dan Anwar, 2024).

*Website* adalah istilah yang merujuk pada kumpulan halaman web yang terhubung dalam jaringan internet. Halaman-halaman tersebut umumnya ditulis menggunakan format HTML (*Hyper Text Markup Language*) dan diakses melalui protokol HTTP (*Hypertext Transfer Protocol*). Secara umum, website terbagi menjadi dua jenis, yaitu statis dan dinamis. Website statis menyajikan informasi yang tidak berubah, seperti pada situs profil perusahaan. Sementara itu, website dinamis menyajikan konten yang bersifat interaktif berasal dari dua arah antara pengelola dan pengguna situs, contohnya seperti *Friendster*, *Multiply*, dan sejenisnya (Orisa dkk., 2023)

#### **2.2.5. Sistem Informasi**

Sistem adalah kumpulan objek yang saling berhubungan dan bekerja sama untuk mencapai tujuan yang sama. Informasi adalah hasil dari pengolahan data yang dikumpulkan dan diubah menjadi sesuatu yang lebih bermanfaat bagi penerimanya, baik dalam bentuk teks maupun gambar. Sistem Informasi merupakan suatu sistem dalam organisasi yang mengintegrasikan proses transaksi harian dan mendukung fungsi manajerial, sehingga dapat menghasilkan dan menyajikan laporan tertentu untuk pihak eksternal (Hanny dkk., 2023).

Sistem informasi dapat didefinisikan sebagai sistem yang menghasilkan output berupa informasi yang bermanfaat bagi berbagai tingkatan manajemen. Sistem informasi adalah suatu sistem dalam organisasi yang memenuhi kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategis organisasi. Selain itu, sistem ini juga menyajikan laporan-laporan yang diperlukan oleh pihak eksternal tertentu (Novitasari dkk., 2021).

Sistem informasi merupakan serangkaian metode yang tersusun secara sistematis untuk mengelola, menyimpan, mengontrol, dan menyajikan informasi guna mendukung pencapaian tujuan yang telah ditetapkan oleh suatu organisasi. Pemanfaatan teknologi dalam sistem informasi oleh suatu organisasi dapat meningkatkan kemampuan serta kecepatan dalam proses pengumpulan, analisis, dan penyampaian informasi. Selain itu, sistem ini juga berperan dalam meringankan beban kerja dalam pelaksanaan operasional dan tugas-tugas organisasi. Keberadaan informasi yang akurat dan terpercaya sangat krusial untuk mendukung kelancaran kegiatan administrasi organisasi. Oleh karena itu, secara umum dapat disimpulkan bahwa sistem informasi berkontribusi signifikan dalam menunjang efisiensi dan efektivitas kinerja organisasi. Terdapat tiga kategori sistem informasi berdasarkan kemajuan teknologi saat ini, yaitu: Sistem Informasi Berbasis Desktop, Sistem Informasi Berbasis Web, dan Sistem Informasi Berbasis *Mobile* (Alvrahesta dkk., 2023).

#### **2.2.6. *Unified Modelling Language (UML)***

*Unified Modeling Language (UML)* adalah metode yang digunakan untuk memodelkan sistem perangkat lunak. UML juga dapat diartikan sebagai kumpulan aturan dan simbol notasi yang berfungsi untuk mendeskripsikan spesifikasi suatu sistem. Notasi-notasi ini disusun dalam bentuk elemen-elemen grafis yang digunakan dalam proses pemodelan. Dalam pendekatan *Object Oriented Analysis and Design (OOAD)*, pembangunan aplikasi berbasis objek memandang setiap komponen sebagai objek, dan sistem dipahami sebagai hasil interaksi antarobjek yang digambarkan menggunakan UML. Versi terbaru dari UML mencakup lima belas jenis diagram yang dikelompokkan ke dalam dua kategori utama, yaitu *structure diagram* dan *behaviour diagram*. *Structure diagram* menggambarkan struktur data dan hubungan statis antar komponen dalam sistem, sementara *behaviour diagram* merepresentasikan interaksi dinamis antara objek-objek yang terdapat dalam sistem informasi tersebut (Hidayati dkk., 2023).

*Use case diagram* merupakan salah satu model dalam tahap analisis dan perancangan sistem yang digunakan untuk menggambarkan kebutuhan sistem dari sudut pandang pengguna. Diagram ini menunjukkan bagaimana sistem berinteraksi dengan aktor-aktor eksternal dan digunakan sebagai dasar dalam menggambarkan fungsionalitas sistem secara keseluruhan. *Use case diagram* merepresentasikan bentuk abstraksi dari hubungan antara aktor dan sistem. Oleh karena itu, pemilihan level abstraksi yang tepat sangat diperlukan agar sesuai dengan tujuan perancangan. Diagram ini menggambarkan skenario tipikal interaksi antara pengguna sistem seperti *admin* dan *user*, sehingga mempermudah pemahaman terhadap bagaimana sistem digunakan. Selain memberikan gambaran mengenai fungsionalitas, *use case diagram* juga membantu menjelaskan konteks sistem dan batasan-batasan yang ada. Dua

komponen utama dalam *use case* diagram adalah aktor dan *use case*. Aktor merupakan entitas yang berinteraksi langsung dengan sistem, baik berupa individu dengan peran tertentu maupun sistem eksternal lainnya. Aktor divisualisasikan dengan simbol manusia stik (*stick man*) dan diberi label berupa kata benda yang menjelaskan perannya. Aktor terbagi menjadi dua, yaitu aktor primer yang memulai suatu *use case* dan aktor sekunder yang mendukung pelaksanaannya (Hidayati dkk., 2023).

Sementara itu, *use case* digambarkan dalam bentuk elips yang berisi aktivitas atau fungsi sistem dari perspektif pengguna, biasanya ditulis dengan kata kerja aktif. Seorang aktor dapat terlibat dalam beberapa *use case*, dan satu *use case* pun bisa dijalankan oleh lebih dari satu aktor (Hidayati dkk., 2023).

*Unified Modeling Language* (UML) adalah metode pemodelan visual yang digunakan dalam proses perancangan dan pembangunan perangkat lunak berorientasi objek. UML berfungsi sebagai bahasa standar untuk pemodelan sistem perangkat lunak, dan telah diakui secara luas sebagai cetak biru (*blueprint*) dalam proses pengembangan sistem. UML tidak hanya mencakup rancangan teknis dari sistem, tetapi juga mencakup pemodelan proses bisnis, struktur kelas, serta spesifikasi lainnya dalam bentuk grafis yang mudah dipahami (Ramdany dkk., 2024).

### 1) *Use Case Diagram*

*Use case diagram* berfungsi sebagai media komunikasi tingkat tinggi yang digunakan untuk menggambarkan kebutuhan fungsional suatu sistem. Diagram ini merupakan pendekatan yang mendukung pengembangan berbasis kegunaan (*use-case driven development*) dengan menampilkan interaksi antara pengguna (aktor) maupun entitas luar dengan sistem yang sedang dikembangkan (Hidayati dkk., 2023).

*Use case diagram* merupakan salah satu model dalam tahap analisis dan perancangan sistem yang digunakan untuk menggambarkan kebutuhan sistem dari sudut pandang pengguna. Diagram ini menunjukkan bagaimana sistem berinteraksi dengan aktor-aktor eksternal dan digunakan sebagai dasar dalam menggambarkan fungsionalitas sistem secara keseluruhan. *Use case diagram* merepresentasikan bentuk abstraksi dari hubungan antara aktor dan sistem. Oleh karena itu, pemilihan level abstraksi yang tepat sangat diperlukan agar sesuai dengan tujuan perancangan. Diagram ini menggambarkan skenario tipikal interaksi antara pengguna sistem seperti admin dan *user*, sehingga mempermudah pemahaman terhadap bagaimana sistem digunakan. Selain memberikan gambaran mengenai fungsionalitas, *use case* diagram juga membantu menjelaskan konteks sistem dan batasan-batasan yang ada. Dua komponen utama dalam *use case* diagram adalah aktor dan *use case*. Aktor merupakan entitas yang

berinteraksi langsung dengan sistem, baik berupa individu dengan peran tertentu maupun sistem eksternal lainnya. Aktor divisualisasikan dengan simbol manusia stik (*stick man*) dan diberi label berupa kata benda yang menjelaskan perannya. Aktor terbagi menjadi dua, yaitu aktor primer yang memulai suatu *use case* dan aktor sekunder yang mendukung pelaksanaannya. Sementara itu, *use case* digambarkan dalam bentuk elips yang berisi aktivitas atau fungsi sistem dari perspektif pengguna, biasanya ditulis dengan kata kerja aktif. Seorang aktor dapat terlibat dalam beberapa *use case*, dan satu *use case* pun bisa dijalankan oleh lebih dari satu aktor. *Use case diagram* adalah representasi visual dari fungsi-fungsi utama yang diharapkan terdapat dalam suatu sistem, serta memperlihatkan bagaimana interaksi terjadi antara pengguna (aktor) dengan sistem itu sendiri. Aktor dalam *use case* menggambarkan entitas, baik berupa individu maupun sistem eksternal, yang berperan menjalankan atau berinteraksi dengan fungsionalitas dalam sistem. Dalam *use case diagram*, antar aktor maupun antar *use case* dapat saling terhubung melalui relasi yang memiliki karakteristik tertentu. Sebuah *use case* utama, yang disebut *base use case*, dapat berelasi dengan satu atau lebih *use case* tambahan, yang dikenal sebagai *supplier use case*, melalui relasi *include* dan *extend*. Relasi *extend* menggambarkan bahwa *use case* tambahan dapat memperluas fungsionalitas dari *base use case* apabila dibutuhkan, khususnya saat mengeksekusi alur alternatif dalam skenario dari *base use case*. Sebaliknya, relasi *include* menunjukkan bahwa *base use case* selalu bergantung pada *supplier use case* untuk menyelesaikan proses utamanya, sehingga *supplier use case* selalu dipanggil dalam eksekusi alur utama *base use case*. Perlu dicatat bahwa baik *include* maupun *extend* tidak mengatur urutan pelaksanaan antara *base use case* dengan *supplier use case*, melainkan hanya menekankan hubungan ketergantungan dalam konteks alur yang digambarkan dalam skenario. Selain itu, dalam konteks aktor, dapat pula terbentuk relasi antara aktor induk dan aktor turunan, di mana aktor turunan mewarisi peran dan hak akses dari aktor induk. Hal ini memungkinkan aktor turunan memiliki akses terhadap lebih banyak fungsionalitas sistem dibandingkan aktor induk yang menjadi asalnya (Ramdany dkk., 2024).

*Use case diagram* memiliki beberapa komponen utama, di antaranya aktor, *use case*, asosiasi, dan *stereotype*. *Stereotype* merupakan elemen penamaan dalam *use case* yang disarankan agar sederhana dan mudah dipahami (Ramdany dkk., 2024).

Dua elemen inti dalam diagram ini adalah aktor dan *use case*, masing-masing memiliki peran penting dalam menggambarkan hubungan antara sistem dan lingkungannya. Aktor adalah entitas yang dapat berupa manusia, sistem lain, atau

proses eksternal yang berinteraksi dengan sistem informasi dari luar. Meskipun dilambangkan dengan simbol gambar manusia, aktor tidak selalu merujuk pada individu, tetapi bisa juga berupa sistem eksternal atau perangkat lunak lain. *Use case* menggambarkan fungsi-fungsi atau layanan utama yang disediakan oleh sistem. Setiap *use case* menggambarkan satu unit aktivitas yang dapat terjadi antara sistem dan aktor, biasanya dinyatakan dengan kata kerja pada awal nama *use case* (Ramdany dkk., 2024).

Adapun simbol-simbol yang digunakan dalam *use case diagram* dijelaskan sebagai berikut.

**Tabel 2.3 Use Case Diagram**

Simbol	Nama	Keterangan
	<i>Actor</i>	Menyatakan orang, proses, atau sistem lain yang secara aktif berinteraksi dengan sistem.
	<i>Dependency</i>	Hubungan ini mengindikasikan bahwa ketika suatu elemen mandiri mengalami perubahan, elemen lain yang bergantung padanya, yaitu elemen tidak mandiri, juga akan terpengaruh.
	<i>Generalization</i>	Hubungan ini memungkinkan objek anak ( <i>descendent</i> ) mewarisi perilaku dan struktur data dari objek induk ( <i>ancestor</i> ) yang berada di atasnya.
	<i>Include</i>	Menjelaskan kasus penggunaan sumber secara eksplisit.
	<i>Extend</i>	Menjelaskan bahwa <i>use case</i> target memperluas perilaku <i>use case</i> sumber pada titik tertentu.
	<i>Association</i>	Menjalin hubungan antara satu objek dengan objek lainnya.
	<i>System</i>	Menentukan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Menjelaskan rangkaian tindakan yang ditampilkan oleh sistem untuk menghasilkan hasil yang terukur bagi seorang aktor.

Dalam penyusunannya, terdapat beberapa komponen utama seperti:

- Aktor: Entitas luar sistem yang melakukan interaksi, tidak selalu berupa manusia.
- Use Case*: Fungsionalitas yang disediakan sistem, ditulis menggunakan kata kerja aktif.

- c. Asosiasi: Garis penghubung antara aktor dan *use case* yang menandakan adanya komunikasi atau interaksi.
- d. Relasi: Seperti *include* *use case* tambahan yang selalu dijalankan bersama *use case* utama, *extend* fungsionalitas tambahan opsional, dan *generalization* hubungan turunan antara aktor atau *use case*.

Langkah-langkah dalam membuat *use case* diagram meliputi:

- a. Identifikasi aktor

Langkah awal dalam merancang sistem adalah mengidentifikasi siapa saja aktor yang akan terlibat. Selain itu, perlu ditentukan ruang lingkup sistem agar batasannya menjadi jelas. Pembatasan ini penting untuk mengetahui apakah suatu entitas merupakan aktor dan apakah aktor tersebut berupa individu, perangkat, atau sistem lain. Secara umum, aktor adalah pihak eksternal yang berinteraksi dengan sistem untuk mencapai suatu tujuan.

- b. Menentukan *use case* berdasarkan kebutuhan aktor

Setelah aktor dikenali, langkah berikutnya adalah menentukan *use case* yang relevan. *Use case* merujuk pada fungsionalitas sistem yang memberikan nilai atau manfaat langsung bagi aktor. Setiap *use case* menggambarkan aktivitas atau layanan tertentu yang dibutuhkan oleh aktor dalam menjalankan interaksinya dengan sistem.

- c. Membuat skenario interaksi dari setiap *use case*

Tahap selanjutnya adalah menyusun skenario atau alur proses untuk masing-masing *use case*. Skenario ini menjelaskan langkah-langkah atau proses interaksi antara aktor dan sistem secara rinci, baik untuk alur utama maupun alur alternatif jika ada kondisi tertentu yang mempengaruhi jalannya proses.

Beberapa manfaat dari *use case diagram* antara lain:

- a. Menyediakan sarana untuk mengidentifikasi kebutuhan fungsional sistem.
- b. Membantu memecah ruang lingkup sistem menjadi bagian-bagian yang lebih terstruktur.
- c. Mempermudah komunikasi antara pengembang, pengguna, dan *stakeholder* terkait fungsi sistem.
- d. Menjadi acuan dalam proses identifikasi, pelacakan, dan pengendalian kegiatan pengembangan sistem.
- e. Menyediakan dasar untuk mengestimasi ruang lingkup, usaha, dan waktu pelaksanaan proyek.

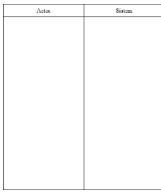
- f. Menjadi referensi dalam menyusun rencana pengujian, termasuk penentuan *test plan* dan *test case*.
- g. Berfungsi sebagai alat pelacak kebutuhan sistem.
- h. Memberikan titik awal dalam mengidentifikasi entitas atau objek data.
- i. Menyediakan kerangka kerja yang dapat diarahkan untuk pengelolaan proyek pengembangan sistem.

2) *Activity Diagram*

*Unified Modeling Language* (UML) merupakan standar bahasa pemodelan yang digunakan untuk menyusun cetak biru (*blueprint*) dalam pengembangan perangkat lunak. Salah satu jenis diagram yang terdapat dalam UML adalah *Activity Diagram*. Dalam konteks ini, simbol-simbol yang digunakan mencerminkan stereotipe dari aktivitas yang digambarkan dalam diagram tersebut. *Activity diagram* biasanya memvisualisasikan aliran proses kerja, di mana suatu proses menerima masukan (*input*) dari sisi kiri dan menghasilkan keluaran (*output*) di sisi kanan. *Activity diagram* berfungsi untuk memodelkan alur aktivitas atau proses dalam sistem informasi secara terstruktur. Diagram ini menjelaskan secara menyeluruh tentang di mana alur kerja dimulai dan berakhir, jenis-jenis aktivitas yang dilakukan selama proses berlangsung, serta urutan kejadian yang terjadi dalam *workflow* tersebut. Selain itu, *activity diagram* juga mendukung pemodelan proses paralel dalam sistem. Untuk pengguna yang sudah akrab dengan pendekatan analisis dan perancangan sistem tradisional, diagram aktivitas mengintegrasikan konsep fundamental dari *data flow diagram* (DFD) dan diagram alir sistem (*flowchart*), sehingga memudahkan pemahaman aliran logika dalam proses yang sedang dimodelkan. *Activity Diagram* adalah representasi visual dari alur kegiatan dalam sistem, menggambarkan urutan proses atau aktivitas yang berlangsung secara logis dalam sistem yang berjalan (Ramday dkk., 2024).

Adapun simbol-simbol yang digunakan dalam *activity diagram* dijelaskan sebagai berikut.

**Tabel 2.4 Activity Diagram**

Simbol	Nama	Keterangan
	<i>Swimline</i>	Menampilkan siapa yang bertanggung jawab dalam melakukan setiap aktivitas dalam diagram.

	<i>Activity</i>	Menjelaskan langkah demi langkah bagaimana objek dari setiap kelas antarmuka berinteraksi.
	<i>Action</i>	Menjelaskan langkah-langkah dalam sebuah <i>activity</i> mengeksekusi <i>action</i> tertentu saat <i>activity</i> tersebut dimasuki, ditinggalkan, atau ketika merespon <i>event</i> spesifik.
	<i>Decision Node</i>	Mencabangkan alur proses menjadi dua atau lebih jalur berdasarkan kondisi sesuai syarat yang terpenuhi.
	<i>Initial Node</i>	Menjelaskan bagaimana sebuah objek dibentuk atau diawali.
	<i>Activity Final Node</i>	Menjelaskan bagaimana sebuah objek dibentuk dan diakhiri.
	<i>Fork Node</i>	Menjelaskan satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.

*Activity diagram* digunakan untuk menggambarkan aktivitas utama beserta alur hubungan antar aktivitas dalam sebuah proses sistem atau proses bisnis. Diagram ini menekankan pada representasi alur kerja (*workflow*) yang mencerminkan urutan aktivitas dalam sistem, bukan aksi dari aktor yang terlibat (Hidayati dkk., 2023).

Fungsi utama dari *activity diagram* meliputi:

- Mendeskripsikan proses bisnis dengan menampilkan urutan aktivitas yang membentuk alur sistem.
- Menyajikan urutan tampilan pada antarmuka sistem atau pengguna, di mana setiap aktivitas diasumsikan memiliki rancangan UI tersendiri.
- Menjadi dasar rancangan pengujian, di mana tiap aktivitas dapat dikaitkan dengan pengujian tertentu yang harus dirancang kasus ujinya (*test case*).

### 3) Class Diagram

*Class diagram* adalah representasi statis dari struktur sistem yang menunjukkan kelas-kelas beserta hubungan antar kelas yang sifatnya tetap sepanjang waktu. Diagram ini merepresentasikan kelas, termasuk atribut (*state*) dan metode (*behavior*), serta hubungan yang dimiliki antar kelas tersebut (Hidayati dkk., 2023).

*Class Diagram* merupakan representasi visual dari hubungan antara kelas-kelas dalam sebuah sistem serta rincian dari masing-masing kelas. Diagram ini digunakan

untuk memperlihatkan struktur dan tanggung jawab entitas dalam sistem, yang nantinya akan menentukan bagaimana perilaku sistem terbentuk. Dengan kata lain, class diagram menyajikan gambaran struktural dari sistem perangkat lunak melalui berbagai jenis kelas yang dibentuk. Dalam konteks sistem berbasis basis data, *class diagram* menggambarkan alur dan relasi antar komponen yang nantinya menjadi dasar pengembangan sistem. Diagram ini tersusun atas sejumlah kelas dan relasinya. Setiap kelas direpresentasikan dalam bentuk persegi panjang yang terdiri dari tiga bagian: bagian atas berisi nama kelas, bagian tengah berisi atribut atau data yang dimiliki oleh kelas tersebut, dan bagian bawah memuat metode atau fungsi (operasi) yang dimiliki kelas. *Class* dapat diartikan sebagai spesifikasi objek yaitu cetakan atau *blueprint* yang ketika diinstansiasi akan menghasilkan sebuah objek nyata dalam sistem. *Class diagram* adalah bagian inti dalam pendekatan pemrograman berorientasi objek dan banyak digunakan dalam tahap analisis maupun desain sistem. Selama proses analisis, *class diagram* membantu menggambarkan aturan dan tanggung jawab entitas yang menjadi bagian dari sistem. Sedangkan dalam tahap desain, diagram ini menangkap keseluruhan struktur kelas yang akan membentuk arsitektur sistem. Hal ini menjadikan *class diagram* sebagai alat bantu visual yang efektif dalam memetakan struktur sistem secara menyeluruh (Ramdany dkk., 2024).

*Class diagram* secara umum mencakup beberapa komponen utama, yaitu:

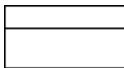
- a. *Class* (*Kelas*): sebagai entitas inti yang memiliki atribut dan operasi.
- b. *Atribut* (*Attributes*): data atau properti yang dimiliki oleh kelas. *Operasi/Metode* (*Operations/Methods*): fungsi atau prosedur yang dapat dijalankan oleh kelas.
- c. *Relasi*: seperti *Association*, *Generalization*, dan *Aggregation*, yang menjelaskan hubungan antar kelas.
- d. *Visibility*: tingkat aksesibilitas terhadap atribut atau metode, misalnya *public*, *private*, atau *protected*.
- e. *Multiplicity/Cardinality*: menunjukkan jumlah minimum dan maksimum objek yang terlibat dalam hubungan antar kelas.

*Class diagram* dapat menampilkan relasi antar kelas secara menyeluruh dan menyediakan pandangan menyeluruh (*global view*) atas sistem yang sedang dikembangkan. Karena kemampuannya dalam menjelaskan hubungan antar objek dan struktur data dalam sistem, class diagram termasuk jenis diagram UML yang paling banyak digunakan dalam pemodelan perangkat lunak modern. *Class diagram* menggambarkan struktur sistem dari sudut pandang *class*, *package*, dan objek yang saling terhubung. Diagram ini mencakup hubungan seperti pewarisian, asosiasi, dan

relasi lainnya yang menjelaskan bagaimana komponen-komponen dalam sistem saling berinteraksi (Ramdany dkk., 2024).

Adapun simbol-simbol yang digunakan dalam *class diagram* dijelaskan sebagai berikut.

**Tabel 2.5 Class Diagram**

Simbol	Nama	Keterangan
—	<i>Generalization</i>	Hubungan antara objek anak ( <i>descendent</i> ) mewarisi perilaku dan struktur data dari objek induk ( <i>ancestor</i> ).
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari dua objek.
	<i>Class</i>	Himpunan objek membagi atribut dan operasi yang sama.
	<i>Collaboration</i>	Sistem menampilkan deskripsi urutan aksi yang menghasilkan hasil terukur bagi aktor.
	<i>Realization</i>	Objek melakukan operasi secara nyata.
	<i>Dependency</i>	Perubahan pada elemen mandiri mempengaruhi elemen bergantung yang tidak mandiri.
—	<i>Association</i>	Relasi menghubungkan satu objek dengan objek lainnya.

*Class Diagram* terdiri dari tiga bagian utama, yaitu bagian atas, bagian tengah, dan bagian bawah, yang masing-masing memiliki peran penting dalam mendeskripsikan struktur suatu kelas dalam sistem. Ketiga komponen ini merepresentasikan elemen-elemen kunci dari pemrograman berbasis objek, mulai dari nama kelas, atribut, hingga metode atau fungsi.

a. Bagian Tengah

Bagian ini menjadi inti dari *class diagram* karena berisi definisi operasi atau metode yang dimiliki oleh suatu kelas. Komponen ini menggambarkan bagaimana objek dari kelas tersebut berperilaku atau berinteraksi, baik dengan objek lain maupun dengan proses dalam sistem. Di sinilah fungsi-fungsi kelas didefinisikan secara detail.

b. Bagian Atas

Bagian paling atas dalam *class diagram* digunakan untuk mencantumkan nama kelas. Komponen ini menggambarkan identitas utama dari sebuah entitas dalam sistem. Sebuah kelas dapat memiliki relasi hierarki dengan kelas lain, baik sebagai induk (*superclass*) maupun sebagai bagian dari kelas yang lebih besar (*member class*).

c. Bagian Bawah

Komponen ini digunakan untuk menuliskan atribut-atribut atau properti yang dimiliki oleh kelas, yang menunjukkan data atau informasi yang disimpan oleh objek dari kelas tersebut. Kumpulan atribut ini menggambarkan karakteristik objek. Dalam beberapa representasi UML, kelas yang tidak memiliki *subclass* atau paling dasar dalam *hierarki* disebut sebagai *root class*, meskipun istilah "*class root*" umumnya tidak digambarkan sebagai lingkaran bergaris, melainkan tetap menggunakan notasi persegi panjang seperti kelas lainnya.

Struktur *class diagram* mencakup:

- a. Atribut, yaitu variabel-variabel yang dimiliki oleh sebuah kelas.
- b. Atribut tersebut menggambarkan properti atau ciri-ciri dari kelas dalam bentuk teks di dalam kotak kelas.
- c. Metode (operasi), yaitu fungsi-fungsi yang menggambarkan perilaku atau aksi yang dapat dilakukan oleh kelas tersebut.

4) *Sequence Diagram*

*Sequence diagram* menggambarkan alur pesan yang dikirimkan antar objek dalam suatu *use case* seiring dengan berjalanannya waktu. Diagram ini menunjukkan interaksi antar objek yang saling berkomunikasi dalam satu proses atau skenario tertentu. *Sequence diagram* mencakup objek-objek yang terlibat serta urutan pesan yang dikirimkan, yang menunjukkan dinamika interaksi antar objek tersebut. Semakin banyak *use case* yang memiliki alur proses berbeda, maka jumlah *sequence diagram* yang diperlukan pun akan semakin banyak, agar seluruh interaksi dalam sistem dapat terdokumentasi dengan lengkap (Hidayati dkk., 2023).

*Sequence diagram* menampilkan pola interaksi antar objek dalam dan sekitar sistem melalui pengiriman pesan, dengan memperhatikan urutan waktu terjadinya komunikasi tersebut (Ramdany dkk., 2024).

Adapun simbol-simbol yang digunakan dalam *sequence diagram* dijelaskan sebagai berikut.

**Tabel 2.6 Sequence diagram**

<b>Simbol</b>	<b>Nama</b>	<b>Keterangan</b>
	<i>Actor</i>	Mendeskripsikan pengguna yang menjalankan interaksi dengan sistem.
	<i>Control Class</i>	Mengatur aliran informasi dalam sebuah skenario. Objek ini umumnya mengendalikan proses bisnis dan perilaku sistem.
	<i>Entity Class</i>	Menampilkan sebuah penyimpanan data atau informasi.
	<i>Boundary Class</i>	Menampilkan antarmuka ( <i>user interface</i> ) atau alat yang memfasilitasi interaksi dengan sistem lain.
	<i>LifeLine</i>	Objek entity membangun antarmuka yang mewakili sebuah objek saling berinteraksi
	<i>Message</i>	Komunikasi antar objek menyajikan informasi tentang aktivitas yang terjadi.
	<i>Message</i>	Komunikasi antar objek memuat informasi tentang aktivitas yang berlangsung.

### 2.2.7. Gizi

Gizi ideal merupakan kondisi seimbang antara asupan zat gizi dengan kebutuhan tubuh. Setiap individu memiliki kebutuhan gizi yang berbeda-beda, tergantung pada faktor usia, berat badan, dan faktor lainnya. Namun, masih banyak masyarakat yang menganggap bahwa makan hanya sekadar untuk menghilangkan rasa lapar, tanpa memperhatikan apakah makanan yang dikonsumsi telah sesuai dengan kebutuhan gizi tubuh. Oleh karena itu, untuk menjaga kesehatan dan keseimbangan tubuh secara optimal, sangat penting untuk memastikan bahwa asupan gizi yang diterima berasal dari makanan yang bergizi dan seimbang. Karena setiap jenis makanan memiliki kandungan zat gizi yang berbeda, maka pemilihan jenis makanan yang sesuai dengan kebutuhan tubuh menjadi hal yang sangat krusial. Ketidaksesuaian asupan makanan, baik kekurangan maupun kelebihan, dapat menimbulkan berbagai masalah kesehatan. Beberapa di antaranya meliputi gangguan gizi seperti kekurangan gizi, obesitas, hipertensi, diabetes, serta berbagai penyakit lainnya. Selain itu, kekurangan zat gizi juga dapat menyebabkan berbagai gangguan kesehatan seperti busuk lapar, anemia, gondok, dan rabun. Status gizi seseorang dapat diukur melalui parameter Indeks Massa Tubuh (IMT), di mana penilaian ini menggunakan dua variabel input, yaitu berat badan dan tinggi badan, serta satu

variabel output berupa nilai status gizi. Nilai gizi ini ditentukan berdasarkan klasifikasi dari hasil perhitungan IMT (Sam'ani dkk., 2023).

Permasalahan gizi di Indonesia mencakup dua aspek utama, yaitu kekurangan dan kelebihan gizi. Kondisi yang dikenal sebagai *Double Burden of Malnutrition* (DBM) atau beban gizi ganda, menggambarkan situasi di mana kekurangan dan kelebihan gizi, baik makronutrien maupun mikronutrien, terjadi secara bersamaan dalam satu siklus kehidupan. Stunting adalah salah satu bentuk kekurangan gizi kronis yang umum terjadi, yaitu kondisi di mana pertumbuhan anak terhambat sehingga tinggi badannya tidak sesuai dengan usianya. Faktor utama penyebab stunting adalah kurangnya asupan energi dan zat gizi serta adanya infeksi berulang. Sebagai upaya untuk mengatasi masalah tersebut, pemerintah Indonesia meluncurkan program gerakan 1.000 Hari Pertama Kehidupan (1.000 HPK) yang bertujuan untuk memperbaiki status gizi dan kualitas hidup anak-anak sejak masa kehamilan hingga usia dua tahun. Periode 1.000 HPK ini merupakan masa kritis yang sangat menentukan perkembangan anak dan memiliki pengaruh jangka panjang terhadap kualitas sumber daya manusia di masa depan. Ketika kekurangan gizi terjadi secara terus-menerus, khususnya pada balita, maka akan meningkatkan angka kesakitan (morbidity) dalam jangka pendek dan memengaruhi kecerdasan serta fungsi kognitif dalam jangka panjang. Status gizi sendiri merupakan kondisi yang menggambarkan keseimbangan antara kebutuhan tubuh akan zat gizi dengan asupan yang diperoleh dari makanan sehari-hari. Perbedaan status gizi antar individu dapat disebabkan oleh berbagai faktor, seperti jenis kelamin, usia, berat dan tinggi badan, serta tingkat aktivitas harian. Secara umum, status gizi diklasifikasikan menjadi tiga kategori, yaitu gizi kurang, gizi normal, dan gizi lebih. Gizi kurang (*undernutrition*) terjadi akibat pola makan yang tidak seimbang atau asupan makanan yang tidak mencukupi kebutuhan tubuh, sedangkan gizi lebih (*overnutrition*) biasanya disebabkan oleh pola makan yang tidak sehat dan konsumsi energi yang berlebihan (Muhammad dkk., 2023) .

Menurut standar WHO–NCHS, status gizi dibedakan menjadi empat kategori: gizi lebih, meliputi kelebihan berat badan (*overweight*) dan obesitas, gizi baik, atau *well-nourished*, gizi kurang, yaitu *underweight* yang mencakup kondisi ringan hingga sedang, termasuk *Protein Calorie Malnutrition* (PCM), gizi buruk, yaitu *severe PCM*, yang mencakup kondisi *marasmus*, *marasmik-kwashiorkor*, dan *kwashiorkor*. Tiga kondisi malnutrisi protein-energi yang berbeda, namun saling berhubungan. *Marasmus* disebabkan oleh kekurangan kalori dan nutrisi secara keseluruhan, termasuk protein, karbohidrat, dan lemak. *Marasmik-kwashiorkor* adalah gabungan dari keduanya, menunjukkan gejala marasmus dan kwashiorkor, seperti wasting parah dan edema. *Kwashiorkor* terjadi ketika tubuh kekurangan protein secara dominan, meskipun asupan kalori mungkin cukup. Status gizi juga menjadi faktor risiko yang

signifikan terhadap munculnya penyakit maupun kematian, karena sangat berperan dalam menentukan tingkat kesehatan dan kemampuan tubuh dalam proses pemulihan. Oleh sebab itu, pengukuran status gizi menjadi langkah penting untuk menurunkan tingginya prevalensi stunting di Indonesia, dengan prevalensi mencapai sekitar 37% dalam satu dekade terakhir. Stunting memiliki dampak langsung dan jangka panjang, seperti penurunan kemampuan kognitif dan belajar, peningkatan risiko penyakit infeksi dan penyakit tidak menular pada masa dewasa, serta menurunnya produktivitas dan kapasitas ekonomi individu (Muhammad dkk., 2023).

#### **2.2.8. Perhitungan Rumus *Harris Benedict***

Metode perhitungan gizi harian pada pengembangan website Gizila menggunakan perhitungan rumus *Harris Benedict* yang merupakan komponen utama dalam menentukan kebutuhan energi harian seseorang, yang didasarkan pada perhitungan angka metabolisme basal (AMB) dan tingkat aktivitas fisik. AMB sendiri merupakan jumlah energi minimal yang dibutuhkan tubuh untuk menjalani masa tubuh (Rian Muhammad dkk., 2024).

Perhitungan kebutuhan kalori menggunakan metode *Harris Benedict* melibatkan beberapa variabel yaitu berat badan, tinggi badan, dan usia. Adapun rumus yang digunakan dalam metode ini berbeda untuk pria dan wanita, yaitu:

- a. BMR Laki-laki =  $66 + (13,7 \times BB) + (5 \times TB) - (6,8 \times U)$
- b. BMR Perempuan =  $655 + (9,6 \times BB) + (1,8 \times TB) - (4,7 \times U)$

Keterangan:

BMR = *Basal Metabolic Rate*

BB = Berat Badan

TB = Tinggi Badan

U = Umur

Metode yang digunakan untuk memperkirakan individu tingkat angka metabolisme basal (AMB) dan aktivitas fisik. AMB mempertimbangkan faktor-faktor seperti usia, jenis kelamin, berat badan, dan tinggi badan.

Setelah nilai AMB individu diperoleh, nilai tersebut kemudian dikalikan dengan faktor aktivitas yang telah ditentukan. Faktor aktivitas yang digunakan untuk menetapkan kebutuhan energi pada orang sehat dapat dilihat pada tabel berikut.

**Tabel 2.7** Faktor Aktivitas Kebutuhan Energi Pada Orang Sehat

Aktivitas	Faktor	
	Laki-laki	Perempuan
Sangat Ringan	1,30	1,30
Ringan	1,65	1,55
Sedang	1,76	1,70
Berat	2,10	2,00

Sumber: <https://ejurnal.stmik-budidarma.ac.id/index.php/jurikom>

Referensi data gizi diperoleh dari situs *nilaigizi.com*, sedangkan acuan Angka Kecukupan Gizi (AKG) diadaptasi dari *hellosehat.com*, serta referensi kalkulator *Harris-Benedict* yaitu *inchcalculator.com*. Untuk perhitungan kebutuhan energi, penelitian ini menggunakan rumus *Harris-Benedict*, dengan menyesuaikan perhitungan menggunakan referensi tambahan dari jurnal ilmiah STMIK Budidarma yang telah dijelaskan pada tabel 2.7 diatas.

Logika perhitungan pada penelitian ini disusun dengan prinsip bahwa seluruh data gizi dalam *database* pada tabel *foods* dinyatakan dalam satuan per 100 gram. Oleh karena itu, digunakan faktor penyesuaian:

$$\text{Faktor} = \frac{\text{berat makanan}}{100} = 2$$

Sebagai ilustrasi misalnya, apabila 100 gram nasi putih mengandung 130 kkal, maka konsumsi 200 gram nasi putih akan menghasilkan faktor perhitungan:

$$\text{Faktor} = \frac{200}{100} = 2$$

Sehingga total energi yang diperoleh adalah:  $130 \times 2 = 260 \text{ kkal}$

Proses perhitungan ini diterapkan secara berulang pada setiap makanan yang dimasukkan pengguna, kemudian hasilnya dijumlahkan dalam program kode dengan variabel *totalIntake*.

## 2.2.9. Kalori

Kalori adalah unit yang digunakan untuk mengukur energi. Dalam konteks makanan, kalori menggambarkan jumlah energi potensial yang terkandung dalam suatu bahan pangan. Makanan yang dikonsumsi manusia dapat mengandung dua jenis nutrisi, yaitu nutrisi penghasil kalori (nutrien energi) dan nutrisi non-kalori. *Nutrien* energi atau makronutrien meliputi karbohidrat, lemak, protein, dan alkohol. Ketika dikonsumsi, *makronutrien* ini berkontribusi terhadap penambahan total kalori dalam tubuh (Rian dkk., 2024).

Makanan yang tergolong tinggi kalori biasanya mengandung sejumlah besar *makronutrien*, baik secara tunggal maupun dalam kombinasi, dan dapat hadir dalam bentuk cair, padat, maupun semi-padat. Kalori yang berasal dari *makronutrien* ini menyediakan energi yang dibutuhkan tubuh untuk menjalankan berbagai aktivitas metabolismik, fisik, serta fungsi kognitif sehari-hari. Apabila asupan kalori melebihi kebutuhan tubuh, kelebihan kalori tersebut akan disimpan sebagai lemak dalam jaringan adiposa. Seluruh kalori, baik yang berasal dari protein, karbohidrat, lemak, maupun alkohol, berpotensi diubah menjadi lemak jika tidak digunakan. Penumpukan lemak secara bertahap dapat menyebabkan peningkatan berat badan. Dalam jangka panjang, kelebihan kalori yang tidak dibakar melalui proses metabolisme atau aktivitas fisik akan berisiko terhadap kelebihan berat badan atau obesitas (Rian Muhammad dkk., 2024).

### 2.2.10. Gizila

Gizila merupakan sebuah media informasi yang masih tahap berkembang dibawah naungan PT. Graisena Usaha Nusantara (GUNTARA) yang bergerak dibidang *creative agency*. Gizila memberikan berita informasi mengenai asupan gizi harian dengan fokus pada edukasi gizi seimbang dan gaya hidup sehat berbasis bukti referensi artikel ilmiah dan ada sumber dari ahli gizi yaitu Indah Puji Lestari S.Gz. yang berperan sebagai *talent* dengan menyampaikan edukasi tentang kesehatan dan asupan gizi seimbang melalui berbagai konten edukatif di *platform digital*, khususnya Instagram, TikTok, (@gizila.id). Gizila menyajikan informasi yang mudah dipahami oleh masyarakat umum mengenai pentingnya pemenuhan zat gizi makro dan mikro, tips konsumsi makanan sehat, serta panduan gizi seimbang untuk kebutuhan sehari-hari. Tujuan utama Gizila adalah meningkatkan literasi gizi di kalangan masyarakat Indonesia agar mampu mengambil keputusan yang lebih baik dalam pola makan sehari-hari. Dengan pendekatan desain visual yang menarik dan penyampaian yang ringan namun informatif, Gizila menjadi sarana penyebarluasan pengetahuan gizi yang relevan dan aplikatif di era digital saat ini.

Berikut Visi dan Misi Gizila adalah:

1) Visi

Menjadi sumber inspirasi dan edukasi gizi serta kesehatan terpercaya di media sosial yang mengajak masyarakat hidup lebih sehat, cerdas, dan bahagia, melalui konten yang mudah dipahami, aplikatif, dan menyenangkan.

2) Misi

a. Memberikan Edukasi Gizi yang Tepat & Terpercaya

Menyampaikan informasi gizi dan kesehatan berdasarkan ilmu yang valid, dibungkus dengan bahasa yang ringan dan mudah dipahami semua kalangan.

b. Membentuk Kebiasaan Hidup Sehat

Mendorong masyarakat untuk mempraktikkan pola makan seimbang, cukup minum air, olahraga teratur, dan menjaga kesehatan mental.

c. Menginspirasi Melalui Konten Kreatif

Menghadirkan tips, infografis, dan ide sehat yang relevan dengan tren, sehingga audiens termotivasi untuk mencintai gaya hidup sehat.

d. Menjadi Komunitas Positif

Membuat ruang interaksi yang ramah dan suportif, tempat berbagi pengalaman dan saling memotivasi untuk menjaga kesehatan.

e. Mengedepankan Kesehatan yang Menyeluruh

Tidak hanya fokus pada nutrisi, tapi juga aktivitas fisik, istirahat, hidrasi, dan kesehatan mental, demi keseimbangan hidup.

### **2.2.11. Profil PT Graha Usaha Nusantara (GUNTARA)**

PT Graisena Usaha Nusantara (GUNTARA) merupakan perusahaan yang bergerak di bidang *Research and Creative Agency*. Perusahaan ini berfokus pada pelayanan sekaligus partner bagi perusahaan dalam meningkatkan penjualan, *insight* positif, serta kebutuhan perusahaan lainnya dengan cara yang keren, inovatif dan kreatif, karena PT Guntara bisa memproduksi konten, mengirimkannya, dan mengatur ritme sesuai kebutuhan perusahaan.

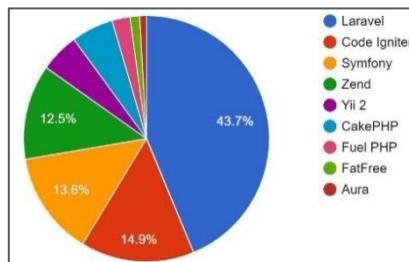
Beberapa pekerjaan yang dilakukan PT Guntara, seperti riset *brand*, periklanan, riset prilaku dan tren pasar, manajemen media sosial (*social media management*), dan pekerjaan jasa konsultan dan *creative agency* lainnya telah menghasilkan perubahan positif untuk para klien-klien PT Guntara. PT Guntara berfokus pada tujuan dalam pekerjaannya, sehingga mampu untuk menghasilkan suatu karya yang tidak hanya bertujuan kepada kepuasan klien, namun juga bisa memuaskan orang lain yang melihatnya. Ada beberapa layanan yang disediakan oleh PT Guntara sebagai perusahaan yang bergerak pada bidang *Research and Creative Agency*:

- a. **Research** mencakup *Research Maketing - Riset Efektivitas Campaign - Koordinator Survey - Manajemen Responden-FGD Organizer*
- b. **Creative & CO** mencakup *Videography-Photography - Graphic Design - Copywriting - Event Organizer*
- c. **Public Relation** mencakup *Copywriting - Artikel - News - Opini Media Sosial - Media Placement - Media Release.*
- d. **Social Media** mencakup *Rencana Konten - Management Media Sosial - Advertisements - KOL Management*
- e. **Kawal Opini** mencakup *Buzzer - Trending Topic – Virality*

## 2.2.12. PHP

PHP atau *Preprocessor Hypertext* adalah bahasa pemrograman *scripting* berbasis *server-side* yang digunakan untuk membuat website dinamis. Salah satu dari keunggulan utama PHP dibandingkan dengan bahasa pemrograman lainnya adalah tingkat kemudahan dalam pembelajaran, kompatibilitas yang tinggi dengan berbagai sistem operasi, serta kinerja yang efisien dan cepat. PHP juga dikenal memiliki banyak fungsi bawaan (*built-in*) yang sangat mendukung dalam proses pengembangan aplikasi web. Konsep *server-side scripting* mengacu pada proses pemrosesan data yang dilakukan di sisi server terlebih dahulu sebelum hasil akhirnya dikirimkan kepada pengguna melalui browser dalam bentuk file HTML, sehingga pengguna hanya melihat tampilan akhir tanpa mengetahui proses yang terjadi di baliknya (Wiarso dan Anwar, 2024).

PHP (*Hypertext Preprocessor*) merupakan salah satu bahasa pemrograman berbasis *scripting* yang paling banyak digunakan di dunia karena memiliki tingkat fleksibilitas dan kemudahan yang tinggi. Berdasarkan data dari w3tech.com, sebanyak 77,6% situs web yang menggunakan *server-side programming* memilih PHP sebagai bahasanya. Dalam satu dekade terakhir, berbagai *framework* populer berbasis PHP telah dikembangkan, seperti *CodeIgniter*, *Symfony*, *Phalcon*, dan *Laravel* (Alvrahesta dkk., 2023). Berdasarkan laporan, *Laravel* menjadi *framework* PHP yang paling banyak digunakan, dengan persentase penggunaan mencapai 43,7%, sebagaimana ditunjukkan pada gambar berikut.



**Gambar 2.14** Diagram Pengguna *Framework* PHP Pada Projek Web

Sumber: <https://ejournal3.undip.ac.id/index.php/jtk/article/view/37723/28884>

*Laravel* memudahkan proses pengembangan website menggunakan sintaksis PHP yang ekspresif, sederhana, dan efisien. *Framework* ini dirancang untuk menghasilkan kualitas kode yang baik dengan mengurangi biaya pengembangan awal serta biaya pemeliharaan, sekaligus meningkatkan pengalaman kerja pengembang. Website yang dikembangkan menggunakan *Laravel* memiliki tingkat skalabilitas yang tinggi, sehingga proses pengembangannya menjadi lebih optimal dan efisien (Alvrahesta dkk., 2023).

Sementara itu, melalui situs dev.to, *Bootstrap* dinobatkan sebagai *framework* CSS paling populer di dunia, dengan lebih dari 20 juta situs yang menggunakan. Di platform

*GitHub*, Bootstrap memperoleh lebih dari 165.000 bintang (*Git Star*), menjadikannya *framework CSS* dengan dukungan komunitas terbesar jika dibandingkan dengan *framework* lain seperti *TailwindCSS*, *Materialize CSS*, dan sebagainya. *Bootstrap* merupakan *framework* yang memadukan *HTML*, *CSS*, dan *JavaScript*, yang dirancang untuk mempermudah pembuatan situs web responsif yang tampil baik pada berbagai perangkat, baik desktop maupun mobile. Dengan dukungan komunitas yang luas, *Bootstrap* memungkinkan pengembangan antarmuka pengguna (UI) secara cepat dan konsisten, baik bagi pemula maupun pengembang berpengalaman (Alvrahesta dkk., 2023).

*Bootstrap* merupakan kumpulan paket aplikasi siap pakai yang digunakan dalam pengembangan *front-end* sebuah *website*. *Framework* ini dapat dikategorikan sebagai *template* desain web dengan berbagai fitur tambahan yang mempermudah proses pembuatan tampilan. *Bootstrap* dirancang untuk mendukung pengguna dari berbagai tingkat kemampuan, baik pemula maupun yang sudah berpengalaman. Dengan hanya berbekal pengetahuan dasar mengenai *HTML* dan *CSS*, pengguna sudah dapat memanfaatkan *Bootstrap* untuk membangun antarmuka web dengan lebih cepat dan efisien (Nugroho dkk., 2021).

Berdasarkan penjabaran tersebut, penulis merancang sistem informasi Gizila dengan memanfaatkan *Framework Laravel* dan *Bootstrap*. Sistem ini diharapkan mampu mendukung masyarakat untuk proses perhitungan gizi secara lebih efisien serta mempermudah masyarakat akses informasi untuk mengedukasi masyarakat tentang gizi yang diperlukan dalam kebutuhan sehari-hari.

### 2.2.13. HTML

HTML (*HyperText Markup Language*) merupakan bahasa markah standar yang digunakan dalam pembuatan halaman web, yang memungkinkan penyajian berbagai elemen informasi seperti teks, gambar, dan konten lainnya melalui web browser. Seiring dengan pertumbuhan pengguna internet yang semakin pesat, HTML juga mengalami perkembangan yang berkelanjutan guna memenuhi kebutuhan penyajian konten web yang lebih baik dan interaktif. Agar kualitas halaman web dapat terus ditingkatkan, HTML terus diperbarui dan dikembangkan. Proses pengembangan dan standarisasi HTML dilakukan oleh organisasi internasional yang bernama W3C (*World Wide Web Consortium*) (Wiarso dan Anwar, 2024).

Pada dasarnya, HTML (*HyperText Markup Language*) bukanlah bahasa pemrograman, melainkan merupakan bahasa markah (*markup language*) yang terdiri atas kumpulan tag atau penanda. Tag-tag ini berfungsi untuk memberi petunjuk kepada browser mengenai bagian-bagian tertentu dari halaman web yang harus ditampilkan kepada pengguna. Penyusunan HTML dilakukan melalui penulisan kode atau simbol tertentu dalam sebuah dokumen, dengan tujuan untuk membentuk struktur dasar dari sebuah halaman web. Dengan demikian, HTML

memungkinkan halaman web dapat divisualisasikan pada layar komputer dan dapat diinterpretasikan secara mudah oleh pengguna (Sinlae dkk., 2024).

Sementara itu, CSS (*Cascading Style Sheets*) merupakan seperangkat aturan yang digunakan untuk mengatur tata letak dan tampilan visual dari halaman web. CSS memungkinkan pengembang untuk menyesuaikan elemen-elemen seperti jenis dan ukuran huruf, warna teks, latar belakang halaman, serta pengaturan visual lainnya, sehingga tampilan web menjadi lebih terstruktur, seragam, dan menarik. Selain mempercepat pemuatan halaman, CSS juga memudahkan dalam pengelolaan kode HTML, memungkinkan tampilan yang responsif pada berbagai ukuran layar, serta mendukung variasi desain yang lebih fleksibel sesuai kebutuhan (Sinlae dkk., 2024).

CSS (*Cascading Style Sheets*) merupakan bahasa yang digunakan untuk mengatur tampilan visual halaman web, termasuk pengaturan warna, tata letak, dan jenis huruf (font). Dengan memanfaatkan CSS, pengembang web dapat merancang halaman yang responsif, yaitu halaman yang mampu menyesuaikan tampilannya secara optimal pada berbagai ukuran layar perangkat. Secara umum, kode CSS disusun terpisah dari dokumen HTML, meskipun dalam kondisi tertentu dapat juga disisipkan langsung ke dalam file HTML. Pemisahan antara CSS dan HTML bertujuan untuk memudahkan proses pengelolaan dan pemeliharaan desain, terutama pada situs web yang memiliki banyak halaman dengan tata letak visual yang seragam (Wiarso dan Anwar, 2024).

Adapun *JavaScript* merupakan bahasa scripting sisi klien (*client-side*), di mana kode yang dituliskan akan diproses langsung oleh perangkat pengguna. *JavaScript* sering dimanfaatkan untuk menciptakan elemen-elemen interaktif dan dinamis pada halaman web, seperti animasi, validasi *form*, dan berbagai interaksi lainnya. Tersedianya berbagai pustaka (*library*) *JavaScript* memberikan kemudahan bagi pengembang dalam menciptakan pengalaman pengguna yang lebih responsif dan menarik. Untuk menjalankan *JavaScript*, diperlukan browser yang memiliki dukungan terhadap bahasa tersebut (Sinlae dkk., 2024).

#### **2.2.14. Tailwind CSS**

Salah satu aspek penting yang harus diperhatikan dalam proses pengembangan *website* adalah bagian *frontend*. *Frontend* merujuk pada komponen sistem yang bertanggung jawab dalam menampilkan tampilan visual kepada pengguna, serta membangun elemen antarmuka pengguna (*user interface*) dari suatu sistem. Salah satu *framework* yang dapat digunakan untuk meningkatkan kualitas tampilan *frontend* adalah *TailwindCSS*, yaitu sebuah kerangka kerja CSS yang digunakan dalam proses pengembangan web. *TailwindCSS* mempermudah implementasi desain antarmuka dengan memungkinkan pengembang untuk mendefinisikan

gaya visual setiap elemen melalui atribut *class* yang telah disediakan (Wiarso dan Anwar, 2024).

Rancang bangun sistem informasi gizila ini dapat dioptimalkan dengan memanfaatkan berbagai teknologi modern yang tersedia saat ini. Salah satu pendekatan yang efektif adalah dengan menggunakan *framework Laravel* sebagai kerangka kerja utama dan *Tailwind CSS* sebagai *framework* untuk pengembangan antarmuka pengguna. *Laravel* menawarkan keunggulan dalam membangun aplikasi web yang stabil, terstruktur, dan terukur, karena dilengkapi dengan berbagai fitur dan fungsi yang mampu mempercepat serta meningkatkan efisiensi proses pengembangan, apabila dibandingkan dengan penggunaan PHP native secara langsung. *Laravel* menerapkan konsep *Model-View-Controller* (MVC), yakni paradigma pemrograman yang membagi aplikasi menjadi tiga komponen utama. *Model* berperan dalam pengelolaan dan interaksi dengan basis data, *View* menangani aspek visual serta logika tampilan, sedangkan *Controller* bertugas mengatur alur komunikasi antara *Model* dan *View*. Sementara itu, *Tailwind CSS* menawarkan pendekatan desain yang fleksibel dan responsif. *Framework* ini memungkinkan efisiensi dalam proses penulisan *style CSS*, serta memberikan keleluasaan bagi pengembang untuk merancang tampilan sesuai kebutuhan, berkat ketersediaan fitur-fitur yang memudahkan proses *styling*. *Tailwind CSS* juga mendukung pembuatan komponen secara bebas tanpa ketergantungan terhadap gaya bawaan dari *framework* lain. Efektivitas penggunaan *Tailwind CSS* dapat semakin meningkat dengan dukungan dari *Flowbite*, yaitu pustaka komponen antarmuka pengguna (UI) yang interaktif dan dibangun menggunakan *Tailwind CSS*. *Flowbite* membantu mempercepat proses pembuatan elemen UI dan meningkatkan efisiensi dalam pengembangan tampilan situs web (Mardiana dkk., 2024).

Dengan mengombinasikan *Laravel* dan *Tailwind CSS*, pengembangan sistem informasi Gizila dapat menghasilkan aplikasi berbasis web yang tidak hanya fungsional dan efisien, tetapi juga memiliki tampilan antarmuka yang menarik serta mudah digunakan oleh pengguna.

Dalam rancang bangun website Gizila ini, digunakan metode *Rapid Application Development* (RAD), yang menekankan pada kecepatan dalam proses pengembangan dengan tetap mempertahankan kualitas sistem. Untuk pengujian sistem, digunakan metode *blackbox testing*, guna memastikan bahwa fungsionalitas sistem berjalan sesuai dengan spesifikasi dan kebutuhan pengguna.

### **2.2.15. Visual Studio Code**

*Visual Studio Code (VSCode)* merupakan salah satu perangkat lunak yang populer di kalangan *programmer*. Aplikasi ini adalah editor teks yang ringan namun memiliki performa tinggi, dikembangkan oleh *Microsoft* dan dapat dijalankan di berbagai sistem operasi seperti

*Windows, macOS, dan Linux.* *VSCode* secara *default* mendukung bahasa pemrograman seperti *JavaScript, TypeScript, dan Node.js*, serta mendukung banyak bahasa pemrograman lainnya misalnya *C++, C#, Python, Go, Java* melalui plugin tambahan yang tersedia di marketplace bawaan. Beragam fitur telah disematkan dalam *VSCode*, seperti *Intellisense* untuk melengkapi kode secara otomatis, integrasi *Git* untuk pengelolaan versi, *debugging* untuk mendeteksi kesalahan kode, serta dukungan ekstensi yang dapat memperluas fungsionalitas editor ini. Seiring waktu, fitur-fitur tersebut terus ditingkatkan dan ditambah sejalan dengan pembaruan versi yang dirilis oleh pengembangnya (Ardianto dan Widiyatmoko 2024).

*Visual Studio Code* berfungsi sebagai perangkat lunak editor kode sumber yang ringan sekaligus kuat untuk pengembangan aplikasi di desktop. Editor ini memiliki dukungan bawaan terhadap *JavaScript* dan *Node.js*, serta memungkinkan pengguna menambahkan ekstensi untuk berbagai bahasa pemrograman lain, seperti *C++, C#, Python*, maupun *PHP* (Nugroho dkk., 2021).

#### **2.2.16. User Interface dan User Experience**

*User Interface (UI)* dan *User Experience (UX)* merupakan aspek penting dalam pengembangan aplikasi, baik berbasis web maupun mobile. UI menggambarkan cara suatu program berinteraksi dengan pengguna dan menjadi bagian dari sistem informasi yang memerlukan interaksi pengguna untuk menghasilkan *input* dan *output*. UI yang baik bersifat *user friendly*, yaitu mudah digunakan serta memiliki fitur yang membuat pengguna merasa nyaman saat mengoperasikannya. Kenyamanan tersebut menjadi bagian dari UX, yang mencakup tidak hanya cara kerja produk atau layanan, tetapi juga pengalaman pengguna saat berinteraksi dengannya. UX menilai sejauh mana penggunaan produk mudah, sederhana, mudah dipahami, serta efektif dan efisien dalam mendukung interaksi (Rahmawati dkk., 2023).

*User interface (UI)* dan *user experience (UX)* merupakan dua komponen yang tidak dapat dipisahkan dalam proses perancangan sebuah produk, baik aplikasi maupun sistem. UI berfokus pada tampilan visual yang berinteraksi langsung dengan pengguna, seperti desain antarmuka, estetika, serta pemilihan warna yang tepat. Sementara itu, UX berkaitan dengan persepsi serta pengalaman pengguna dalam menggunakan produk, meliputi kemudahan pemahaman alur kerja, kenyamanan saat berinteraksi dengan sistem, dan pencapaian tujuan pengguna. UX pada dasarnya tidak dapat dirancang secara langsung, namun perancang UI/UX dapat mengembangkan sistem atau aplikasi yang mampu menciptakan pengalaman pengguna yang optimal (Putri dkk., 2023).

### 2.2.17. System Usability Scale (SUS)

Proses perhitungan dalam metode SUS tergolong mudah dipahami dan tidak rumit. Analisis data dilakukan secara kuantitatif menggunakan metode *System Usability Scale* (SUS) yang dikembangkan oleh *Brooke* pada tahun 1986. SUS dikenal sebagai skala pengukuran kegunaan yang handal, populer, efektif, dan ekonomis (Putro dkk., 2023).

Metode ini digunakan untuk menguji tingkat kegunaan web *Gizila* melalui kuesioner, dalam penelitian ini ada dua aspek yang diuji yaitu dari sisi *user admin* yang mengakses fitur *dashboard* *admin* *Gizila* dan *user publik* yang mengakses web *Gizila*, masing-masing terdiri dari 14 pertanyaan. Instrumen penelitian disusun oleh peneliti untuk mengumpulkan data. Setiap pernyataan dalam kuesioner dinilai dengan skala 1 sampai 5, di mana terdiri dari sangat tidak setuju, tidak setuju, kurang setuju, setuju, dan sangat setuju. Adapun lima alternatif tersebut akan dipilih oleh responden dengan ketentuan bobot penilaian sebagaimana yang dijelaskan pada tabel berikut.

**Tabel 2.8** Bobot Penilaian SUS

Bobot	Keterangan
1	Sangat Tidak Setuju
2	Tidak Setuju
3	Kurang Setuju
4	Setuju
5	Sangat Setuju

Sampel yang diambil dalam penelitian ini mencakup *stakeholder* terkait di PT GUNTARA dengan jumlah 7 responden. Setelah instrumen selesai dibuat, kuesioner disebarluaskan dalam bentuk *Google Form* untuk mengumpulkan data yang akan dianalisis.

Berikut rumus perhitungan metode *System Usability Scale* (SUS):

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan:

$\bar{x}$  = Skor rata-rata.

$\sum x$  = Jumlah skor SUS.

$n$  = Jumlah responden.

Skor SUS = ((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1)) x 2.5).

Perhitungan hasil pengujian *website* Gizila menggunakan instrumen SUS dilakukan dengan mengikuti aturan tertentu. Setiap butir pernyataan memiliki skor kontribusi dengan rentang nilai skala 1 sampai 5. Untuk pernyataan bernomor ganjil, skor diperoleh dari hasil pengurangan skala jawaban responden dengan angka 1. Sedangkan untuk pernyataan bernomor genap, skor dihitung dengan mengurangkan skala jawaban responden dari angka 5. Nilai keseluruhan *System Usability* diperoleh dengan menjumlahkan seluruh skor kontribusi, kemudian mengalikannya dengan faktor 2,5.

Pelaksanaan analisis data dilakukan melalui beberapa tahap. Pertama, tahap validasi data untuk memastikan data yang dikumpulkan sesuai dengan standar penelitian. Kedua, tahap pengeditan data, yaitu memodifikasi data mentah, misalnya menghapus data ganda. Selanjutnya, data dari kuesioner diolah menggunakan perhitungan yang sesuai dengan rumus metode SUS. Skor SUS yang diperoleh kemudian diinterpretasikan untuk memberikan rekomendasi yang dapat meningkatkan tingkat kegunaan (*usability*) dari *website* yang diuji.

### **2.2.18. User Acceptance Testing (UAT)**

*User Acceptance Testing* (UAT) adalah tahap akhir dalam proses pengujian perangkat lunak yang dilakukan langsung oleh pengguna akhir yang akan menggunakan sistem. Tujuan dari pengujian ini adalah untuk memverifikasi bahwa seluruh fitur dan fungsi sistem telah berjalan sesuai dengan kebutuhan nyata pengguna, bukan sekadar mengikuti spesifikasi teknis yang telah ditentukan. UAT menggunakan metode *black box testing*, di mana pengguna menguji sistem berdasarkan hasil yang diharapkan tanpa memperhatikan struktur internal kode. Melalui proses ini, pengguna memberikan penilaian terhadap kinerja sistem secara keseluruhan dan memastikan bahwa sistem benar-benar dapat mendukung aktivitas atau pekerjaan yang akan dilakukan. Hasil dari UAT dapat dijadikan dasar bahwa sistem siap untuk digunakan secara penuh, karena telah memenuhi kebutuhan serta harapan pengguna. Pengujian ini juga dikenal dengan istilah pengujian beta, pengujian aplikasi, atau pengujian oleh pengguna akhir dan menjadi salah satu langkah penting sebelum perangkat lunak diluncurkan ke lingkungan produksi atau pasar (Wahyudi dkk., 2023).

*User Acceptance Testing* (UAT) dilakukan oleh pengguna akhir atau klien sebagai langkah untuk menilai apakah sistem yang dikembangkan telah sejalan dengan harapan dan memberikan manfaat yang diinginkan sebelum diterapkan dalam lingkungan operasional. UAT terdiri atas beberapa jenis pengujian, di antaranya adalah *Alpha Testing* dan *Beta Testing*, yang bertujuan untuk mengevaluasi apakah fungsionalitas perangkat lunak telah berjalan dengan baik. *Alpha Testing* merupakan pengujian internal yang dilakukan oleh tim pengembang guna mendeteksi dan memperbaiki kesalahan sejak tahap awal pengembangan. Pengujian ini biasanya menggunakan pendekatan *Black Box Testing* untuk menilai

fungsionalitas sistem secara menyeluruh. Sementara itu, *Beta Testing* dilakukan oleh pengguna eksternal atau calon pengguna sistem guna memperoleh umpan balik berdasarkan pengalaman penggunaan yang sebenarnya. Penilaian dalam tahap ini umumnya dilakukan melalui penyebaran kuesioner untuk menilai sejauh mana sistem diakui dan dianggap bermanfaat oleh pengguna (Yakub dkk., 2024).

### **2.2.19. Black Box Testing**

*Black box testing* adalah metode pengujian perangkat lunak yang difokuskan pada pemeriksaan keselarasan sistem dengan kebutuhan dan spesifikasi pengguna yang telah ditetapkan sejak tahap perancangan awal. Tujuan utama dari pengujian ini adalah untuk memastikan bahwa fungsi, masukan (*input*), dan keluaran (*output*) dari sistem sesuai dengan persyaratan yang diharapkan. Pengujian dilakukan berdasarkan item uji yang telah dirancang sebelumnya, dan dilaksanakan menggunakan pendekatan *black box*, tanpa mempertimbangkan struktur internal dari perangkat lunak. Hasil dari pengujian ini menunjukkan bahwa seluruh proses dalam sistem berjalan dengan baik dan sesuai fungsinya. *Black box testing* menitikberatkan pada proses verifikasi terhadap fungsi sistem, guna memastikan bahwa setiap komponen bekerja sesuai dengan spesifikasi yang diinginkan. Dalam pelaksanaannya, penguji melakukan interpretasi terhadap berbagai kemungkinan kondisi masukan, dan menjalankan pengujian terhadap fungsi-fungsi tertentu dalam sistem. Tujuan akhir dari pengujian ini adalah untuk mengidentifikasi serta memperbaiki kesalahan (error), sehingga sistem dapat dinyatakan layak untuk digunakan secara optimal (Wiarso dan Anwar, 2024).

*Black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada evaluasi fungsi sistem tanpa memperhatikan struktur internal atau kode sumber yang membangunnya. Pengujian ini bertujuan untuk menemukan kesalahan pada aspek fungsional, tampilan antarmuka, model data, serta akses ke sumber data eksternal, dengan menguji berbagai kombinasi *input* dan hasil *output* yang diharapkan. Dalam pelaksanaannya, penguji hanya menilai apakah sistem bekerja sesuai dengan spesifikasi atau kebutuhan yang telah ditentukan, tanpa memahami bagaimana proses internalnya berlangsung. Salah satu pendekatan yang umum digunakan dalam metode ini adalah *Equivalence Partitioning*, yaitu dengan mengelompokkan data masukan berdasarkan karakteristik yang serupa guna memastikan pengujian dilakukan secara menyeluruh dan efisien (Nuridza dkk., 2025).

### **2.2.20.XAMPP**

XAMPP menghadirkan satu paket installer yang mencakup *Apache* sebagai web server yang berfungsi menyimpan file-file *website*, dan *phpMyAdmin* sebagai aplikasi yang digunakan untuk melakukan perancangan *database MySQL*. Database berfungsi sebagai

media penyimpanan data dalam sebuah program yang terdiri atas tabel, *field*, dan *record*, yang dikelola oleh DBMS (*Database Management System*). DBMS menggunakan struktur *query* yang disebut SQL untuk mengolah data, dan dalam pembuatannya, pengguna biasanya memanfaatkan perangkat lunak bernama MySQL (Nugroho dkk., 2021). Menurut Sadeli dalam Supriyanta (2015), Apache adalah web server yang mampu beroperasi di berbagai sistem operasi, termasuk *Unix*, *BSD*, *Linux*, *Microsoft Windows*, dan *Novell Netware*, serta berperan penting dalam memberikan layanan dan mengaktifkan fungsi situs web (Nugroho dkk., 2021).

### **2.2.21. MySQL**

*Structured Query Language* (SQL) merupakan bahasa khusus yang digunakan dalam domain sistem basis data relasional, berfungsi untuk mengelola dan memanipulasi data pada sistem *Relational Database Management System* (RDBMS). Salah satu aplikasi RDBMS yang paling umum digunakan oleh pengembang web dalam pengelolaan basis data adalah *MySQL*. Dalam praktik pengembangan web, berbagai fungsi dalam bahasa pemrograman PHP sering dimanfaatkan untuk melakukan operasi pembuatan, pembacaan, pembaruan, dan penghapusan (CRUD) data dalam *database SQL*, yang hasilnya kemudian dapat ditampilkan pada halaman web. PHP (*Hypertext Preprocessor*) adalah bahasa pemrograman sisi server (*server-side*) yang memungkinkan situs web berinteraksi dengan basis data dan menghasilkan konten yang bersifat dinamis. PHP merupakan bahasa *scripting* yang terintegrasi dengan HTML, dan seluruh sintaksisnya dieksekusi di sisi server. Dengan demikian, hanya hasil eksekusi yang akan dikirimkan dan ditampilkan pada browser pengguna. Sementara itu, SQL berperan sebagai sistem manajemen basis data relasional yang menyimpan data terstruktur dari sebuah situs web. Dengan menggabungkan kemampuan PHP dan *MySQL*, pengembang dapat membangun situs web yang mampu menyimpan dan menyajikan data secara sistematis, seperti data, profil pengguna, atau artikel berita, serta mendukung interaksi aktif antara sistem dan pengguna (Sinlae dkk., 2024).

### **2.2.22. Framework Laravel**

*Laravel* merupakan salah satu *framework* berbasis PHP yang dikembangkan dengan menerapkan arsitektur *Model-View-Controller* (MVC). *Framework* ini memiliki ekosistem komunitas yang luas dan aktif, sehingga memudahkan pengembang dalam memperoleh solusi saat menghadapi kendala dalam proses pengembangan aplikasi web. Penggunaan *laravel* dalam pengembangan *website* akan terciptanya aplikasi yang memiliki tingkat skalabilitas tinggi, sehingga proses pengembangannya menjadi lebih efektif dan efisien. *Laravel* dirancang agar pengembangan aplikasi web dapat dilakukan menggunakan sintaks PHP yang jelas,

ekspresif, dan terstruktur, dengan tujuan untuk menghasilkan kualitas kode yang baik, mengurangi biaya pada tahap awal pengembangan, meminimalkan biaya pemeliharaan, serta menciptakan pengalaman kerja yang lebih optimal bagi para pengembang (Alvrahesta dkk., 2023).

*Laravel* merupakan *framework open-source* berbasis PHP yang dirancang untuk menyederhanakan dan mempercepat proses pengembangan aplikasi web. *Framework* ini menyediakan berbagai fitur unggulan seperti sistem *routing*, *templating engine*, serta *Eloquent* ORM untuk pengelolaan basis data secara efisien. *Laravel* mengadopsi arsitektur *Model-View-Controller* (MVC) yang membantu meningkatkan keteraturan struktur kode dan mempermudah pemeliharaan aplikasi. Selain itu, *Laravel* juga mendukung integrasi dengan berbagai pustaka eksternal dan layanan pihak ketiga, serta dilengkapi dengan fitur keamanan yang andal, sehingga banyak digunakan oleh pengembang dalam membangun sistem informasi yang *scalable* dan mudah dikelola (Nuridza dkk., 2025).

*Laravel* adalah *framework* web berbasis bahasa pemrograman PHP yang bersifat *open source* dan dapat digunakan secara gratis. *Framework* ini dikembangkan oleh *Taylor Otwell* dan dirancang khusus untuk membangun aplikasi web dengan menerapkan arsitektur *Model-View-Controller* (MVC). Meskipun mengadopsi pola dasar MVC, struktur implementasi MVC dalam *Laravel* memiliki perbedaan dibandingkan dengan pola MVC konvensional. Salah satu ciri khas *Laravel* adalah sistem *routing*-nya, yang berfungsi sebagai penghubung antara *request* dari pengguna dengan *controller*. Dengan adanya mekanisme *routing* ini, *controller* tidak menerima *request* secara langsung, melainkan melalui pemetaan yang telah ditentukan dalam konfigurasi *routing* *Laravel* (Wiarso dan Anwar, 2024).

### **2.2.23. Livewire**

*Livewire* merupakan salah satu *framework full-stack* pada *Laravel* yang memungkinkan pembuatan tampilan antarmuka yang interaktif tanpa perlu menulis kode *JavaScript* secara langsung. Melalui pendekatan berbasis komponen, *Livewire* memungkinkan interaksi pengguna dikendalikan dari sisi server menggunakan *Laravel*, dan tampilan halaman dapat diperbarui secara langsung melalui mekanisme *AJAX* tanpa harus melakukan pemuatan ulang halaman. Teknologi ini juga dimanfaatkan dalam sistem manajemen proyek guna meningkatkan efektivitas komunikasi dan akses data secara *real-time*, serta menyederhanakan proses pengelolaan proyek berbasis web melalui fitur reaktif yang mampu merespons perubahan data secara instan (Nuridza dkk., 2025).

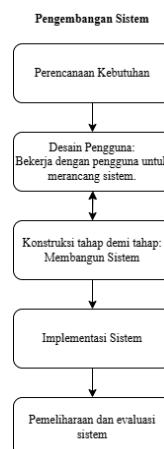
### 2.2.24. *Alpine.js*

*Alpine.js* merupakan *framework JavaScript* yang ringan dan dirancang untuk memberikan interaktivitas pada elemen HTML secara deklaratif dan reaktif. Dengan ukuran hanya sekitar 7kB dalam bentuk *gzip*, *Alpine.js* sangat efisien dan tidak memerlukan konfigurasi yang rumit, sehingga menjadi pilihan tepat untuk aplikasi dengan kebutuhan interaksi sederhana tanpa harus menggunakan *framework* besar seperti *Vue.js* atau *React*. *Framework* ini memanfaatkan sejumlah direktif seperti *x-data* untuk mendefinisikan *state*, *x-bind* untuk pengikatan atribut, *x-on* untuk penanganan *event*, serta *x-show* dan *x-for* yang digunakan dalam pengaturan tampilan berdasarkan kondisi maupun iterasi data (Nuridza dkk., 2025).

### 2.2.25. Metode *Rapid Application Development* (RAD)

Metodologi ini merupakan metode *incremental* yang diterapkan dalam pengembangan sistem perangkat lunak, dengan fokus pada tahapan kemajuan pengembangan sistem. Metode ini memecah setiap tahap menjadi bagian-bagian kecil dan setiap prosesnya dibangun menggunakan metode yang sama dengan metode air terjun (*waterfall*). Tujuannya adalah untuk mengembangkan sistem secara bertahap, meskipun perencanaan awal dilakukan secara menyeluruh (Murdiani dan Hermawan, 2022). Model *waterfall* tidak cocok untuk pengembangan yang kompleks karena tahapannya tidak dapat diulang, memerlukan waktu pengerjaan yang cukup lama, dan biayanya mahal. Di sisi lain, model RAD sangat cocok untuk proyek berskala besar karena menggunakan metode iteratif (berulang) yang dapat menghemat biaya.

Fase Siklus Hidup *Rapid Application Development* (RAD) dalam RAD adalah sebagai berikut.



Gambar 2.15 Siklus RAD

Siklus hidup RAD bertujuan untuk mengembangkan sistem yang sesuai dengan kebutuhan pengguna. Siklus ini terdiri dari beberapa tahapan sebagai berikut:

a. Perencanaan kebutuhan (*Requirements Planning*)

Pada fase ini, beberapa elemen perencanaan sistem digabungkan dengan fase analisis SLDC. Pengguna dan pengembang mendiskusikan kebutuhan bisnis dan persyaratan sistem. Fase ini berakhir ketika pengguna dan pengembang menyepakati masalah-masalah utama dan memberikan izin untuk melanjutkan pengembangan.

b. Desain pengguna (*User Design*)

Pada fase ini, pengguna berinteraksi dengan pengembang untuk membuat dan mengembangkan prototipe. Tim menggunakan teknik *Joint Application Design* (J.A.D) dan alat *Computer Aided System Engineering* (C.A.S.E) untuk menerjemahkan keinginan pengguna. Desain dilakukan secara berkelanjutan sehingga memudahkan pengguna untuk memahami dan memodifikasi sesuai kebutuhan.

c. Konstruksi tahap demi tahap (*Construction*)

Pada fase ini, programmer fokus pada pengembangan sistem aplikasi. Pengguna sangat berperan aktif dalam pengembangan ini.

d. Peralihan dan evaluasi sistem

Fase ini melibatkan peralihan, pengujian, pergantian ke sistem baru, dan pelatihan untuk pengguna baru dan melakukan evaluasi. Tahap evaluasi merupakan proses untuk memastikan bahwa sistem yang dikembangkan dapat berjalan dengan baik serta sesuai dengan kebutuhan yang telah dirancang sebelumnya. Pada fase ini, programmer mulai mengimplementasikan rancangan sistem yang telah disetujui pada tahap perancangan sebelumnya. Sebelum sistem dioperasikan secara penuh, dilakukan terlebih dahulu proses pengujian untuk mendeteksi kemungkinan adanya kesalahan atau bug dalam sistem. Pada tahap ini pula, pengguna atau pihak terkait biasanya memberikan masukan terhadap sistem yang telah dibuat, sekaligus memberikan persetujuan apakah sistem tersebut layak untuk digunakan (Ardhana, 2024).

## **BAB III**

### **ANALISIS DAN PERANCANGAN**

Bab ini menguraikan analisis terhadap sistem yang berjalan, usulan sistem baru, serta perancangan sistem informasi Gizila. Analisis mencakup kebutuhan fungsional dan non-fungsional, sementara perancangan sistem diilustrasikan menggunakan diagram UML yang terdiri dari *Use Case Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*.

#### **3.1. Analisis Sistem yang Berjalan**

Analisis merupakan suatu proses yang dilakukan untuk menguraikan permasalahan menjadi bagian-bagian yang lebih kecil agar dapat dipahami dan dievaluasi secara menyeluruh. Tujuan dari kegiatan analisis adalah untuk memperoleh pemahaman yang lebih mendalam terhadap suatu persoalan, membantu dalam pengambilan keputusan yang tepat, serta menjadi dasar dalam menghasilkan solusi dan inovasi baru yang sesuai dengan kebutuhan.

Dalam penelitian ini, pengelolaan kebutuhan gizi harian masyarakat masih banyak dilakukan secara manual dan tidak terintegrasi secara digital. Masyarakat umumnya kesulitan dalam memantau asupan gizi harian karena tidak adanya sistem yang dapat menghitung, mencatat, dan memberikan rekomendasi gizi secara otomatis. Selain itu, keterbatasan akses terhadap informasi gizi yang akurat dan personalisasi menu harian menjadi hambatan dalam menjaga pola makan yang sehat.

Oleh karena itu, dibutuhkan sebuah sistem informasi gizi berbasis web yang mampu membantu pengguna dalam menghitung dan memantau asupan gizi harian secara mudah dan efisien. Sistem ini dirancang menggunakan pendekatan Rapid Application Development (RAD) agar dapat dikembangkan secara cepat dan fleksibel sesuai dengan kebutuhan pengguna, serta mendukung gaya hidup sehat berbasis data yang lebih akurat dan praktis.

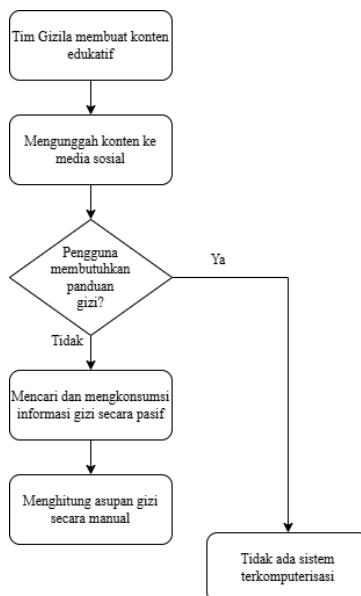
Saat ini, Gizila beroperasi sebagai media informasi yang bergerak dibidang *Creative Agency* di bawah naungan PT. Graisena Usaha Nusantara (GUNTARA). Aktivitas utama Gizila berfokus pada penyebarluasan edukasi gizi seimbang dan gaya hidup sehat melalui *platform* media sosial seperti Instagram dan TikTok (@gizila.id).

##### **3.1.1. Analisis Proses Bisnis**

Proses bisnis yang berjalan saat ini dilakukan yaitu dengan membuat konten edukatif berupa artikel, *short video*, dan infografis, lalu mengunggahnya ke *platform* media sosial oleh tim *Creative Agency* Gizila, dengan dukungan ahli gizi yaitu Indah Puji Lestari S.Gz., sebagai *talent* dalam konten edukatif Gizila. Pengguna masyarakat umum mengonsumsi informasi

tersebut secara pasif, tidak ada sistem terkomputerisasi yang memungkinkan pengguna untuk melakukan perhitungan kebutuhan gizi harian secara personal dan otomatis.

Umumnya, pengguna mencari informasi gizi dari berbagai sumber secara terpisah tanpa adanya panduan yang terpusat, sehingga sulit untuk memantau asupan gizi harian secara terstruktur. Penghitungan nilai gizi makanan, pencatatan menu harian, hingga rekomendasi kebutuhan kalori sering kali dilakukan secara manual, yang rawan menimbulkan kesalahan dan tidak efisien. Sebagai contoh, seseorang harus mencatat sendiri makanan yang dikonsumsi dan menghitung total kalori tanpa bantuan sistem otomatis, sehingga tidak mendapatkan gambaran yang akurat tentang kecukupan gizinya. Oleh karena itu, diperlukan perancangan sistem informasi berbasis web yang dapat membantu dalam menghitung, mencatat, dan merekomendasikan kebutuhan gizi harian pengguna secara praktis dan terintegrasi. Berikut adalah gambaran dari analisis proses bisnis yang berjalan saat ini.



**Gambar 3.1** Identifikasi Proses Bisnis yang Berjalan

Proses bisnis sistem informasi GIZILA yang ditunjukkan pada Gambar diawali oleh tim *Creative Agency GIZILA* yang bertugas membuat konten edukatif mengenai gizi. Konten ini dapat berupa artikel, video pendek, maupun infografis. Setelah proses pembuatan konten selesai, langkah berikutnya adalah mengunggah konten tersebut ke media sosial agar dapat diakses oleh masyarakat umum. Setelah konten dipublikasikan, pengguna atau masyarakat akan mengonsumsi informasi tersebut secara pasif. Di titik ini, terdapat dua kemungkinan kondisi:

- a. Jika pengguna membutuhkan panduan gizi lebih lanjut, pengguna harus mencari sumber informasi lain karena sistem belum menyediakan fitur interaktif atau otomatisasi.
- b. Jika pengguna tidak secara aktif mencari panduan, maka pengguna akan sekadar menyerap konten yang tersedia tanpa tindakan lanjutan.

Dalam kondisi tersebut, pengguna mencari informasi gizi dari berbagai sumber secara terpisah, baik dari internet, media sosial, maupun referensi lain. Informasi yang diperoleh kemudian digunakan untuk menghitung asupan gizi secara manual, seperti menghitung kalori, protein, lemak, dan karbohidrat dari makanan yang dikonsumsi. Proses manual ini sangat rentan terhadap kesalahan, kurang efisien, dan sulit dilakukan secara konsisten oleh pengguna.

Seluruh proses tersebut belum didukung oleh sistem yang terkomputerisasi, sehingga tidak ada aplikasi atau platform yang secara otomatis menghitung, menyimpan, atau merekomendasikan kebutuhan gizi harian pengguna. Hal ini ditunjukkan oleh tahapan akhir pada diagram, yaitu tidak ada sistem terkomputerisasi, yang menandakan keterbatasan dari proses bisnis saat ini.

Dengan demikian, proses bisnis yang sedang berjalan belum mampu menyediakan solusi digital yang personal, otomatis, dan akurat dalam pemenuhan kebutuhan gizi harian. Hal inilah yang mendasari perlunya pengembangan Sistem Informasi Gizila berbasis web yang akan menyediakan perhitungan gizi harian secara otomatis, pencatatan menu makan, hingga rekomendasi sesuai dengan kebutuhan pengguna masing-masing.

### **3.1.2. Analisis Sistem Informasi**

Sistem informasi gizi yang berjalan saat ini pada platform Gizila masih berupa penyebarluasan informasi gizi dilakukan melalui media sosial dalam bentuk konten edukatif yang dibuat oleh tim kreatif dan didukung oleh tenaga ahli gizi. Proses ini mencakup pembuatan konten informasi berupa artikel, video pendek, atau infografis yang kemudian diunggah ke media sosial agar dapat diakses oleh masyarakat umum.

Namun, dari sisi sistem informasi, proses ini memiliki keterbatasan yang cukup signifikan. Informasi gizi yang disediakan hanya bersifat satu arah, artinya pengguna hanya dapat menerima informasi tanpa adanya interaksi langsung dengan sistem. Tidak ada mekanisme untuk mencatat data individu pengguna, menghitung kebutuhan gizi harian, atau memberikan saran spesifik berdasarkan kondisi masing-masing individu.

Hal ini menjadikan sistem informasi Gizila masih bersifat pasif dan informatif, bukan transaksional atau personal.

Selain itu, pengguna yang membutuhkan panduan gizi secara personal masih harus mencari informasi dari berbagai sumber lain secara manual. Proses pencarian informasi ini

tidak terstandar, tidak terstruktur, dan rawan menghasilkan informasi yang tidak akurat atau bahkan menyesatkan. Bahkan untuk hal teknis seperti menghitung kebutuhan gizi harian, pengguna harus melakukan perhitungan sendiri tanpa adanya bantuan alat digital atau sistem otomatis, yang tentu saja menyulitkan dan tidak efisien.

Akibatnya, sistem yang berjalan saat ini:

- a. Tidak menyimpan data pengguna atau riwayat konsumsi makanan.
- b. Tidak memberikan *feedback* atau rekomendasi gizi berdasarkan kondisi fisik pengguna (seperti usia, berat badan, tinggi badan, atau aktivitas fisik).
- c. Tidak mendukung proses personalisasi dan pengambilan keputusan berbasis data.
- d. Belum mampu menciptakan interaksi dua arah antara pengguna dan sistem.

Dengan kata lain, tidak ada sistem informasi terkomputerisasi yang mendukung kebutuhan gizi secara menyeluruh dan otomatis. Hal ini menyebabkan ketidakefisiensian informasi, baik dari segi pencatatan, pemantauan, maupun pengambilan keputusan gizi harian oleh masyarakat umum.

Kondisi ini menimbulkan urgensi untuk membangun sistem informasi Gizila berbasis web yang mampu menghimpun, memproses, dan menyajikan informasi gizi secara otomatis, terstruktur, dan responsif terhadap kebutuhan individual pengguna. Sistem ini akan mempermudah masyarakat dalam menghitung kebutuhan gizi harian, input menu harian, serta menerima saran dan rekomendasi gizi harian.

Belum ada sistem informasi terkomputerisasi yang spesifik untuk perhitungan gizi personal. Sistem yang ada terbatas pada *platform* media sosial yang berfungsi sebagai akses informasi saja. Meskipun informasi yang disajikan dianggap bermanfaat dan mudah dipahami, pengguna belum memiliki *tools* atau sistem otomatisasi untuk menerapkan pengetahuan tersebut secara personal dan terukur.

### **3.1.2.1. Kelebihan Sistem Informasi yang Berjalan**

Sistem informasi gizi yang saat ini dijalankan oleh tim Gizila masih berbasis manual dan dilakukan melalui pendekatan sosial-edukatif di platform media sosial. Meskipun belum menggunakan sistem terkomputerisasi, terdapat beberapa kelebihan yang masih bisa diperoleh dari sistem ini, antara lain:

- a. Proses penyebaran informasi gizi melalui media sosial dapat diakses oleh siapa saja tanpa memerlukan aplikasi khusus atau login ke sistem tertentu. Masyarakat hanya perlu menggunakan media sosial yang umumnya sudah digunakan sehari-hari seperti *Instagram*, *TikTok*. Hal ini membuat informasi gizi dapat diterima secara luas tanpa batasan perangkat atau kebutuhan koneksi yang stabil seperti pada sistem digital terpusat.

- b. Menghemat biaya operasional dan teknologi, karena sistem yang berjalan masih berbasis konten manual dan tidak melibatkan infrastruktur sistem informasi berbasis web, maka tidak diperlukan biaya untuk membangun, memelihara, atau mengembangkan aplikasi. Biaya seperti *domain*, *hosting*, *sertifikat keamanan*, serta pengembangan *software* tidak menjadi beban, sehingga sistem ini cukup efisien secara finansial dalam tahap awal edukasi.
- c. Minim risiko kegagalan teknis, sistem yang tidak berbasis perangkat lunak (*software*) dan perangkat keras (*hardware*) tertentu memiliki keunggulan dalam hal ketahanan terhadap gangguan teknis. Karena tidak ada sistem informasi yang dijalankan secara otomatis, maka tidak ada risiko seperti *server down*, *crash*, terkena virus, atau *bug program*. Proses berjalan secara manual dan bebas dari ketergantungan teknologi tinggi.
- d. Mengandalkan sumber daya manusia sebagai penggerak utama, seluruh proses dijalankan oleh tim kreatif dan ahli gizi, sehingga pendekatan yang digunakan bersifat *human-centered*. Hal ini memungkinkan materi informasi menjadi lebih adaptif, menarik, dan berbasis komunikasi langsung antar manusia, yang dapat menciptakan empati lebih tinggi terhadap *audiens* atau pengguna.

Meskipun sistem manual ini memiliki beberapa kelebihan seperti yang dijelaskan di atas, namun perlu digaris bawahi bahwa sistem ini belum mampu memenuhi kebutuhan personalisasi, otomatisasi, dan interaktivitas yang dibutuhkan oleh masyarakat saat ini dalam memantau kebutuhan gizi harian secara akurat dan efisien. Maka dari itu, pengembangan sistem informasi Gizila berbasis web sangat diperlukan sebagai solusi digital yang lebih komprehensif.

### **3.1.2.2. Kekurangan Sistem Informasi yang Berjalan**

Meskipun penyampaian informasi gizi melalui media sosial memiliki jangkauan yang luas, sistem informasi yang saat ini digunakan oleh Gizila masih bersifat manual dan pasif, sehingga menimbulkan sejumlah kekurangan sebagai berikut:

- a. Proses *monitoring* gizi tidak efisien dan rentan kesalahan, karena belum adanya sistem komputasi yang dapat menghitung kebutuhan gizi harian secara otomatis, pengguna harus mengelola dan menghitung sendiri asupan makanan. Proses ini tidak hanya memerlukan waktu yang lama, tetapi juga rentan terhadap kesalahan perhitungan dan pencatatan, terutama bagi pengguna yang tidak memiliki pemahaman mendalam tentang ilmu gizi.

- b. Tidak adanya sistem pencatatan dan rekapitulasi data secara otomatis, sistem saat ini tidak memiliki fitur untuk mencatat riwayat konsumsi makanan, kebutuhan kalori, atau hasil perhitungan gizi harian.
- c. Informasi gizi bersifat satu arah dan tidak interaktif, konten yang dibagikan oleh Gizila hanya dapat dikonsumsi secara pasif. Tidak tersedia fitur yang memungkinkan pengguna berinteraksi, memasukkan data, atau menerima rekomendasi spesifik. Hal ini mengakibatkan informasi yang diterima oleh pengguna bersifat umum dan tidak selalu relevan dengan kondisi individu, sehingga kurang efektif untuk penerapan secara personal.
- d. Pengguna harus mencari sendiri informasi dari berbagai sumber, karena belum tersedia sistem yang menyediakan informasi gizi terpusat, pengguna harus mencari berbagai referensi sendiri secara manual, seperti kalkulator gizi online, artikel berbeda, atau video edukatif dari sumber lain. Proses pencarian ini memakan waktu, tidak terstandar, dan berpotensi menyebabkan kebingungan jika informasi yang ditemukan tidak saling konsisten.
- e. Tidak mendukung pengambilan keputusan yang cepat dan berbasis data, tanpa adanya sistem otomatis, pengguna tidak bisa langsung mengetahui apakah asupan gizi harian sudah mencukupi atau tidak. Ini membuat proses pengambilan keputusan terkait pengaturan pola makan, pemilihan makanan, dan aktivitas fisik menjadi kurang tepat sasaran dan cenderung subjektif.

### 3.2. Analisis Sistem yang Diusulkan

Berdasarkan berbagai permasalahan yang ditemukan pada sistem informasi yang berjalan saat ini, seperti kurangnya interaktivitas, tidak adanya pencatatan data, serta keterbatasan dalam menghitung kebutuhan gizi secara otomatis, maka diusulkan sebuah sistem informasi baru. Sistem ini dirancang berbasis web dan ditujukan untuk mempermudah masyarakat dalam memantau, menghitung, dan mengelola asupan gizi harian secara digital, personal, dan efisien.

Sistem yang diusulkan adalah sebuah rancangan bangun sistem informasi berbasis web bernama Gizila yang mampu memberikan informasi kebutuhan gizi harian secara personal, interaktif, dan otomatis. Sistem ini akan menjadi perpanjangan dari media edukasi yang sudah ada, dengan menambahkan fungsionalitas utama berupa kalkulator gizi. Perbedaan utama dengan sistem lama adalah transisi dari penyedia informasi umum menjadi platform layanan personal. Jika sebelumnya pengguna hanya bisa membaca tentang gizi, kini pengguna dapat memasukkan data diri dan langsung mendapatkan hasil perhitungan spesifik untuk tubuhnya. Sistem informasi ini berbasis web yang dapat diakses melalui berbagai jenis perangkat seperti

komputer, laptop, tablet, maupun *smartphone*, selama terhubung dengan internet dan menggunakan browser. Sistem ini akan dibangun menggunakan bahasa pemrograman PHP dengan *framework Laravel* versi 11.45.1 dan menggunakan MySQL sebagai basis data. Sistem ini menggunakan metode *Rapid Application Development* (RAD) untuk pengembangan yang cepat dan iteratif serta perhitungan gizi dengan perhitungan rumus *Harris Benedict* yang akurat untuk memperkirakan Angka Metabolisme Basal (AMB) dan kebutuhan gizi harian.



**Gambar 3.2** Identifikasi Sistem yang Diusulkan

### 3.2.1. Kebutuhan Fungsional

Kebutuhan fungsional dalam sistem informasi Gizila merupakan fitur utama yang harus tersedia agar pengguna dan admin dapat menjalankan aktivitasnya sesuai dengan tujuan sistem. Adapun kebutuhan fungsional sistem informasi Gizila terbagi menjadi dua pihak pengguna, yaitu admin dan pengguna umum atau publik.

#### 1) Admin

- a. Admin dapat mengakses halaman *dashboard* sistem Gizila dengan terlebih dahulu melakukan *login* menggunakan data akun admin yang telah terdaftar.
- b. Admin memiliki akses untuk mengelola halaman *dashboard* admin gizila yaitu dengan membuat konten artikel edukatif Gizila, dimana didalamnya memuat fitur CRUD (*Create, Read, Update, Delete*) terhadap artikel yang berkaitan dengan gizi dan kesehatan.
- c. Admin dapat menambahkan atau membuat artikel baru untuk *dipublish* pada halaman *website* Gizila, membuat kategori artikel, melihat detail artikel, mengubah isi artikel, dan menghapus artikel melalui antarmuka dashboard admin Gizila.
- d. Admin juga memiliki kontrol penuh terhadap sistem informasi yang berjalan, memastikan seluruh proses publikasi konten edukasi dilakukan secara terstruktur, sistematis, dan terverifikasi dari sumber terpercaya berdasarkan referensi yang ada.
- e. Fitur ini dirancang untuk membantu admin dalam mengakses fitur CRUD (*Create, Read, Update, Delete*) serta menyampaikan informasi gizi kepada masyarakat secara efektif melalui media berbasis web yang sudah terkomputerisasi.

#### 2) Pengguna Umum atau Publik

- a. Pengguna dapat mengakses *website* Gizila secara langsung tanpa perlu login, untuk menjelajahi konten-konten edukatif yang telah tersedia secara publik.
- b. Pengguna dapat menggunakan fitur perhitungan kebutuhan gizi harian, di mana pengguna perlu mengisi data personal seperti usia, berat badan, tinggi badan, jenis kelamin, dan tingkat aktivitas berdasarkan perhitungan rumus *Harris Benedict*.
- c. Setelah mengisi data, pengguna akan menerima informasi berupa hasil dari perhitungan gizi harian, kalkulasi kalori, protein, lemak, dan karbohidrat harian.
- d. Pengguna dapat melakukan pencatatan menu makanan harian, yang akan diproses oleh sistem untuk menghitung asupan gizi aktual berdasarkan database makanan.
- e. Sistem akan memberikan rekomendasi menu harian untuk mendukung kecukupan gizi, berdasarkan pencatatan yang dilakukan pengguna.
- f. Pengguna juga bisa mengakses fitur informasi artikel yang edukatif, seperti artikel gizi, tips kesehatan, dan informasi gaya hidup sehat tanpa harus login ke sistem.

- g. Sistem Gizila memberikan akses penuh terhadap informasi penting bagi masyarakat umum tanpa menghambat pengguna dengan proses login kecuali untuk fungsi spesifik tertentu.
- 3) Kalkulator Gizi
- a. Sistem dapat menerima input data dari pengguna, mencakup: berat badan (kg), tinggi badan (cm), usia (tahun), jenis kelamin, dan tingkat aktivitas harian.
  - b. Sistem dapat menghitung Indeks Massa Tubuh (IMT) untuk mengetahui status gizi pengguna.
  - c. Sistem dapat menghitung Angka Metabolisme Basal (AMB) menggunakan rumus *Harris Benedict*.
  - d. Sistem dapat menghitung total kebutuhan kalori harian dengan mengalikan AMB dan faktor aktivitas berdasarkan rumus *harris benedict* yang sudah dijelaskan pada bab sebelumnya.
  - e. Sistem dapat menampilkan hasil dari perhitungan (IMT, AMB, dan total kalori) kepada pengguna.

### **3.2.2. Kebutuhan Non Fungsional**

Kebutuhan non-fungsional berfungsi untuk menunjang performa sistem dan menjamin kelayakan penggunaan sistem informasi Gizila. Aspek ini meliputi kebutuhan teknis baik dari sisi pengguna maupun pengembang.

Untuk mendukung pengembangan dan operasional sistem informasi Gizila secara optimal, diperlukan beberapa perangkat pendukung yang terbagi ke dalam dua kategori, yaitu perangkat lunak (*software*) dan perangkat keras (*hardware*). Perangkat ini berperan penting dalam memastikan sistem dapat berjalan secara efisien dan diakses dengan mudah oleh pengguna maupun pengelola sistem.

**Tabel 3.1 Kebutuhan Non Fungsional**

No.	Perangkat Keras	No.	Perangkat Lunak
1	Komputer, tablet, atau smartphone yang dapat menjalankan browser.	1	Browser seperti <i>Google Chrome</i> , atau <i>Microsoft Edge</i> untuk mengakses website Gizila.
2	RAM minimal 2GB untuk memastikan performa browser tidak terhambat saat menggunakan fitur interaktif sistem.	2	Hosting yang berfungsi sebagai media penyimpanan file-file sistem Gizila.

3	Perangkat jaringan nirkabel seperti Wi-Fi atau <i>router</i> diperlukan agar perangkat pengguna dapat terhubung ke internet dan mengakses sistem Gizila secara daring ( <i>online</i> ).	3	Domain yang digunakan adalah <i>biz.id</i> yang digunakan untuk mengakses sistem informasi Gizila di internet.
---	--	---	--

### 3.3. Desain Sistem yang Diusulkan

Pada tahapan ini, proses perancangan sistem mencakup pembuatan *use case diagram*, *activity diagram*, *sequence diagram*, dan *class diagram* dengan perancangan struktur basis data, serta desain antarmuka pengguna. Seluruh elemen tersebut dirancang untuk mendukung fungsi-fungsi utama dari sistem yang telah dijabarkan pada tahap analisis sebelumnya.

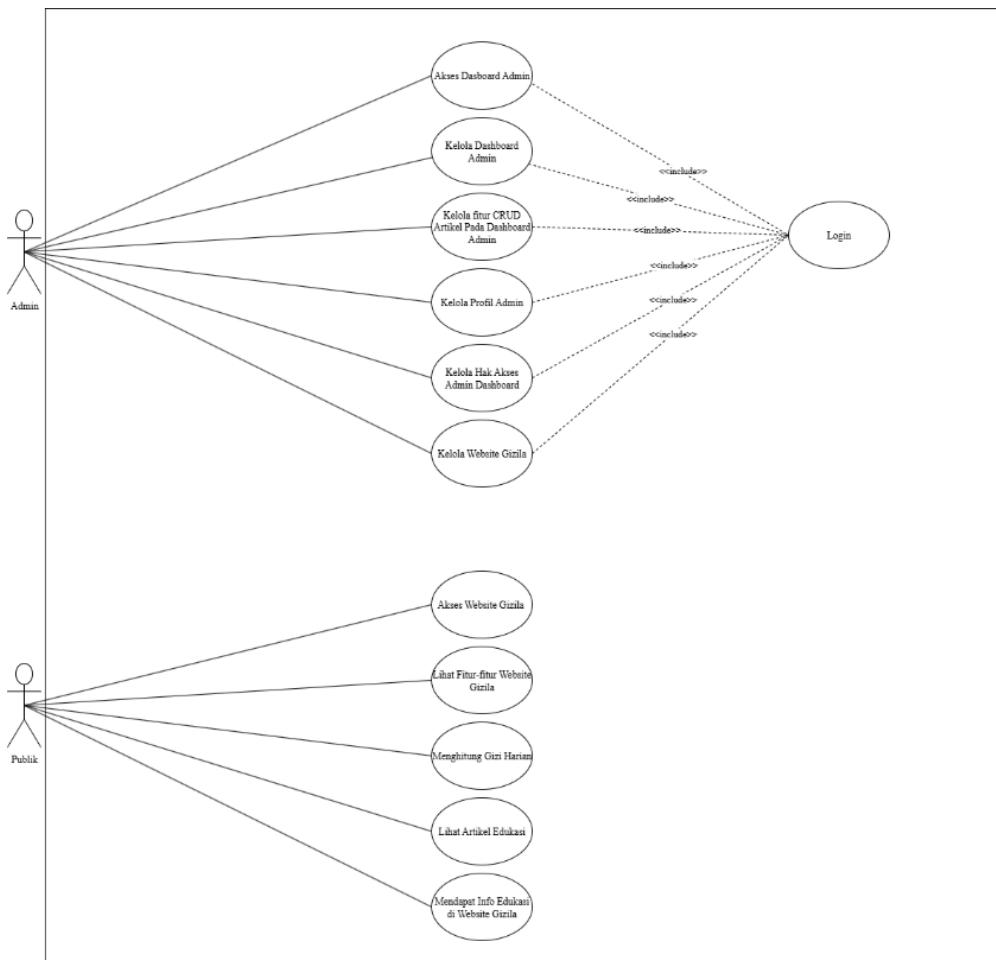
#### 3.3.1. Use Case Diagram

*Use case diagram* berfungsi untuk menggambarkan secara umum bagaimana interaksi antara pengguna dengan sistem informasi Gizila yang dirancang. Diagram ini membantu pengembang memahami dan menentukan hak akses pengguna, baik sebagai admin maupun pengguna umum, terhadap berbagai fitur dan layanan yang tersedia dalam sistem.

Melalui *use case diagram*, proses-proses utama yang dapat dilakukan oleh masing-masing aktor dalam sistem dapat dijelaskan secara sistematis, seperti akses terhadap perhitungan kebutuhan gizi harian, pencatatan menu makanan, serta pengelolaan konten edukasi oleh admin.

Selain itu, *use case diagram* juga berperan dalam menjelaskan skenario penggunaan sistem, batasan interaksi, serta alur proses secara keseluruhan. Hal ini sangat penting dalam memastikan bahwa perancangan sistem dilakukan secara terstruktur, terarah, dan sesuai dengan kebutuhan pengguna.

Berikut adalah *use case diagram* yang menggambarkan rancangan interaksi pengguna dengan sistem informasi Gizila dalam proses perancangannya.



**Gambar 3.3 Use Case Diagram**

*Use case diagram* di atas menggambarkan interaksi antara aktor atau pengguna sistem dengan fitur-fitur atau fungsi utama yang tersedia dalam rancangan Sistem Informasi Gizila berbasis web yang dirancang untuk membantu masyarakat dalam menghitung dan memantau kebutuhan gizi harian secara digital dan berbasis web. Diagram ini disusun untuk memudahkan pemahaman terhadap ruang lingkup sistem yang akan dikembangkan serta memperlihatkan peran masing-masing aktor dalam menjalankan fungsinya terhadap sistem yang dibangun. Diagram diatas menggambarkan dua jenis aktor, yaitu Admin dan Publik (Pengguna Umum), serta fungsi-fungsi utama yang dapat diakses oleh masing-masing aktor berdasarkan hak aksesnya.

1) Aktor: admin

Admin memiliki peran pengelola sistem, yaitu pihak internal (admin) yang bertanggung jawab terhadap pengelolaan tampilan web untuk membuat konten edukasi dalam

bentuk artikel, serta pengelolaan akses ke *dashboard* Gizila. Aktivitas yang dapat dilakukan oleh admin dalam sistem ini meliputi:

a. *Login*

Admin harus terlebih dahulu masuk ke sistem dengan akun yang valid untuk mendapatkan akses penuh ke dalam *dashboard* pengelolaan konten. *Login* merupakan pintu awal agar admin bisa masuk ke sistem. Fungsi *login* bersifat wajib agar terhubung (<<includes>>) ke semua fitur *dashboard* admin Gizila.

b. Akses *Dashboard*

Setelah berhasil *login*, admin diarahkan ke halaman *dashboard* sistem untuk mengelola berbagai fungsi administratif.

c. Kelola *Dashboard* Admin Gizila

Admin dapat memantau dan mengelola berbagai komponen sistem yang ada dalam *dashboard*.

d. Kelola Fitur CRUD Artikel pada *Dashboard* Admin Gizila

Admin dapat melakukan operasi CRUD (*Create*, *Read*, *Update*, dan *Delete*) terhadap artikel edukasi yang akan ditampilkan kepada pengguna publik di *website* Gizila.

e. Kelola Profil Admin

Admin dapat mengelola data profil melalui *dashboard* dengan cara memperbarui atau mengubah informasi sesuai kebutuhan.

f. Kelola Hak Akses Admin *Dashboard*

Admin dapat mengelola dan mengatur hak akses di *dashboard* dengan menentukan peran dan izin akses yang sesuai.

g. Kelola *Website* Gizila

Admin memiliki kontrol terhadap pengelolaan konten dan tampilan pada halaman utama *website* Gizila.

2) Aktor: Publik (Pengguna Umum)

Publik dapat menggunakan fitur-fitur utama Gizila secara bebas tanpa perlu *login* terlebih dahulu. Berikut adalah fungsionalitas yang dapat diakses oleh publik:

a. Akses *Website* Gizila

Pengguna dapat membuka dan menjelajahi halaman website Gizila menggunakan perangkat seperti smartphone, tablet, maupun laptop.

b. Lihat Fitur-Fitur *Website* Gizila

Pengguna bisa melihat semua fitur yang tersedia, termasuk kalkulasi gizi dan artikel edukasi.

c. Menghitung Gizi Harian

Pada fitur ini, pengguna dapat memanfaatkan fitur ini untuk menghitung kebutuhan gizi berdasarkan data seperti usia, tinggi badan, berat badan, dan aktivitas fisik. Perhitungan dilakukan dengan menggunakan rumus *Harris-Benedict*.

d. Lihat Artikel Edukasi

Pengguna dapat melihat daftar artikel yang tersedia untuk meningkatkan pemahaman mengenai gizi dan kesehatan.

e. Mendapat Info Edukasi di *Website* Gizila

Informasi yang telah ditulis oleh admin Gizila akan ditampilkan dalam bentuk artikel edukatif yang bisa diakses bebas oleh pengguna publik.

**Tabel 3.2** Penjelasan *Use Case Diagram*

No	Aktor Sistem	Layanan Sistem
1.	Admin	<i>Login</i>
		Akses <i>dashboard</i> admin Gizila
		Kelola <i>dashboard</i> admin Gizila
		Kelola fitur CRUD artikel pada <i>dashboard</i> admin Gizila
		Kelola <i>website</i> Gizila
2.	Pengguna publik	Akses <i>website</i> Gizila
		Lihat fitur-fitur <i>website</i> Gizila
		Menghitung gizi harian
		Lihat artikel edukasi
		Mendapat info edukasi di <i>website</i> Gizila

### 3.3.1.1. *Use Case Login*

<i>Use Case</i>	: <i>Login</i>
Nama Aktor	: Admin
Keterangan	: Proses admin melakukan akses <i>login</i> ke sistem
Output	: Admin berhasil <i>login</i> ke sistem

**Tabel 3.3 Use Case Login Admin**

Use Case Utama		
No	Pengguna	Sistem
1.	Admin mengakses halaman <i>login</i>	Sistem menampilkan halaman login yang berupa form login yang terdiri dari <i>email</i> dan <i>password</i>
Opsi tambahan: pengguna <i>input email</i> dan <i>password</i>		
2.	Admin input data <i>email</i> dan <i>password</i> pada <i>form login</i> . Data yang diinput, data yang telah terdaftar di halaman hak akses <i>dashboard</i> admin Gizila	Sistem melakukan proses verifikasi untuk mencocokkan data yang diinputkan dengan data yang tersimpan dalam <i>database</i> .

### 3.3.1.2. Use Case Akses Dashboard admin Gizila

Use Case : Akses *Dashboard*  
 Nama Aktor : Admin  
 Keterangan : Proses admin melakukan akses *halaman dashboard* admin Gizila  
 Output : Admin berhasil mengakses halaman *dashboard* admin Gizila

**Tabel 3.4 Use Case Akses Dashboard**

Use Case Utama		
No	Admin	Sistem
1.	Admin melakukan <i>login</i> untuk mengakses halaman <i>dashboard</i> Gizila	Sistem menampilkan halaman utama <i>dashboard</i> admin setelah data admin di verifikasi dengan data yang tersimpan dalam <i>database</i> .
Opsi tambahan: Akses data dan informasi		
2.	Admin mengakses data yang ada di <i>dashboard</i> admin Gizila	Sistem menampilkan informasi yang ada di halaman <i>dashboard</i>

### 3.3.1.3. *Use Case Kelola Dashboard admin Gizila*

Use Case : Kelola *Dashboard*  
 Nama Aktor : Admin  
 Keterangan : Proses admin mengelola *halaman dashboard* admin Gizila  
 Output : Admin berhasil mengelola data dan informasi di *dashboard*

**Tabel 3.5 Use Case Kelola Dashboard**

Use Case Utama		
No	Admin	Sistem
1.	Admin melakukan <i>login</i> untuk mengakses halaman <i>dashboard</i> Gizila	Sistem menampilkan halaman utama <i>dashboard</i> admin setelah data admin di verifikasi dengan data yang tersimpan dalam <i>database</i> .
Opsi tambahan: Kelola data dan informasi		
2.	Admin mengakses data yang ada di <i>dashboard</i> admin Gizila untuk manajemen artikel	Sistem menampilkan informasi yang ada di halaman <i>dashboard</i> untuk manajemen artikel

### 3.3.1.4. *Use Case Kelola Fitur CRUD Artikel di Dashboard*

Use Case : Kelola Fitur CRUD Artikel  
 Nama Aktor : Admin  
 Keterangan : Proses admin mengelola fitur CRUD artikel di *dashboard*  
 Output : Admin berhasil mengelola fitur CRUD artikel di *dashboard*

**Tabel 3.6 Use Case Kelola Fitur CRUD Artikel**

Use Case Utama		
No	Admin	Sistem
1.	Admin memilih fitur manajemen artikel	Sistem menampilkan daftar artikel yang dapat ditambahkan, diubah, dibaca, atau dihapus
Opsi 1: <i>Create</i> artikel		
2.	Admin membuat data artikel baru yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data artikel sesuai aksi yang dipilih ke dalam <i>database</i>
Opsi 2: <i>Read</i> artikel		
3.	Admin membaca data artikel yang ada di <i>dashboard</i> admin Gizila untuk manajemen artikel	Sistem menampilkan informasi yang ada di halaman <i>dashboard</i> untuk manajemen artikel
Opsi 3: <i>Update</i> artikel		
4.	Admin memperbarui data artikel yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data artikel sesuai aksi yang dipilih ke dalam <i>database</i>
Opsi 4: <i>Delete</i> artikel		
5.	Admin menghapus data artikel yang tidak relevan yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data artikel sesuai aksi yang dipilih ke dalam <i>database</i>

### 3.3.1.5. Use Case Kelola Profil

- Use Case : Kelola Profil  
 Nama Aktor : Admin  
 Keterangan : Proses admin mengelola profil admin di *dashboard* admin  
 Output : Admin berhasil mengelola profil admin di *dashboard* admin

**Tabel 3.7 Use Case Kelola Profil**

Use Case Utama		
No	Admin	Sistem
1.	Admin memilih fitur profil	Sistem menampilkan data profil yang dapat ditambahkan, dan diperbarui.
Opsi 1: <i>Create</i>		
2.	Admin membuat data profil baru seperti memperbarui foto profil, nama, email dan password, yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data profil sesuai aksi yang dipilih ke dalam <i>database</i>
Opsi 2: <i>Read</i>		
3.	Admin membaca data profil yang ada di <i>dashboard</i> admin Gizila untuk manajemen profil	Sistem menampilkan informasi yang ada di halaman <i>dashboard</i> untuk manajemen profil
Opsi 3: <i>Update</i>		
4.	Admin memperbarui data profil yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data profil sesuai aksi yang dipilih ke dalam <i>database</i>

### 3.3.1.6. *Use Case Kelola Hak Akses Admin Dashboard*

Use Case	: Kelola Hak Akses Admin <i>Dashboard</i>
Nama Aktor	: Admin
Keterangan	: Proses admin mengelola hak akses admin <i>dashboard</i>
Output	: Admin berhasil mengelola hak akses admin <i>dashboard</i>

**Tabel 3.8 Use Case Kelola Hak Akses Admin**

Use Case Utama		
No	Admin	Sistem
1.	Admin memilih fitur hak akses	Sistem menampilkan daftar hak akses admin yang dapat ditambahkan, diubah, dibaca, atau dihapus
Opsi 1: <i>Create</i>		
2.	Admin membuat hak akses baru yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data artikel sesuai aksi yang dipilih ke dalam <i>database</i>
Opsi 2: <i>Read</i>		
3.	Admin membaca data hak akses yang ada di <i>dashboard</i> admin Gizila untuk manajemen hak akses admin <i>dashboard</i>	Sistem menampilkan informasi yang ada di halaman hak akses untuk manajemen hak akses admin <i>dashboard</i>
Opsi 3: <i>Update</i>		
4.	Admin memperbarui data hak akses yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data hak akses sesuai aksi yang dipilih ke dalam <i>database</i>
Opsi 4: <i>Delete</i>		
5.	Admin menghapus data hak akses yang tidak relevan yang ada di <i>dashboard</i> admin Gizila	Sistem menyimpan atau memperbarui data hak akses sesuai aksi yang dipilih ke dalam <i>database</i>

### 3.3.1.7. Use Case Kelola Website Gizila

Use Case	: Kelola Website Gizila
Nama Aktor	: Admin
Keterangan	: Proses admin mengelola website Gizila
Output	: Admin berhasil mengelola website Gizila

**Tabel 3.9 Use Case Kelola Website Gizila**

Use Case Utama		
No	Admin	Sistem
1.	Admin mengakses halaman <i>website</i> Gizila melalui browser	Sistem menampilkan halaman utama <i>website</i> Gizila
Opsi 1: Akses web tanpa login		
2.	Admin dapat langsung menjelajahi isi situs web Gizila tanpa <i>login</i>	Sistem menampilkan konten publik yang dapat diakses oleh siapa saja secara terbuka
Opsi 2: Meninjau isi konten		
3.	Admin meninjau tampilan konten yang muncul di <i>website</i> Gizila untuk memastikan kesesuaian isi dengan tujuan edukatif	Sistem menampilkan data konten sesuai yang telah diinput pada <i>dashboard</i> admin, agar dapat dievaluasi oleh admin

**3.3.1.8. Use Case Akses Website Gizila**

Use Case : Akses *Website* Gizila  
 Nama Aktor : Publik (Pengguna Umum)  
 Keterangan : Proses *user* mengakses *website* Gizila  
 Output : *User* berhasil mengakses *website* Gizila

**Tabel 3.10 Use Case Akses Website Gizila**

Use Case Utama		
No	User	Sistem
1.	<i>User</i> mengakses halaman <i>website</i> Gizila melalui browser	Sistem menampilkan halaman utama <i>website</i> Gizila
Opsi tambahan: Akses data dan informasi		
2.	<i>User</i> mengakses data yang ada di <i>website</i> Gizila	Sistem menampilkan informasi yang ada di <i>website</i> Gizila

**3.3.1.9. Use Case Lihat Fitur-fitur Website Gizila**

Use Case : Lihat Fitur-fitur *Website* Gizila  
 Nama Aktor : Publik (Pengguna Umum)  
 Keterangan : Proses *user* melihat fitur-fitur *website* Gizila

Output : Admin berhasil melihat fitur-fitur *website* Gizila

**Tabel 3.11 Use Case Lihat Fitur-fitur Website Gizila**

Use Case Utama		
No	User	Sistem
1.	<i>User</i> melihat fitur-fitur <i>website</i> Gizila melalui browser	Sistem menampilkan halaman fitur-fitur <i>website</i> Gizila
Opsi tambahan: Pilih fitur-fitur <i>website</i> Gizila		
2.	<i>User</i> memilih fitur tertentu di <i>website</i> Gizila	Sistem menampilkan informasi halaman fitur yang dipilih oleh <i>user</i> secara dinamis

### 3.3.1.10. Use Case Menghitung Gizi Harian

Use Case : Menghitung Gizi Harian  
 Nama Aktor : Publik (Pengguna Umum)  
 Keterangan : Proses *user* melakukan perhitungan gizi harian  
 Output : *User* berhasil melakukan perhitungan gizi harian

**Tabel 3.12 Use Case Menghitung Gizi Harian**

Use Case Utama		
No	User	Sistem
1.	<i>User</i> melakukan perhitungan gizi harian di <i>website</i> Gizila melalui browser	Sistem menampilkan formulir input data diri (usia, berat, tinggi, aktivitas)
Opsi tambahan: Mengisi dan mengirim data		
2.	<i>User</i> mengelola asupan gizi harian dengan mengisi form fitur gizi harian	Sistem memproses perhitungan dan menampilkan informasi hasil perhitungan kebutuhan gizi harian berdasarkan data yang diinput <i>user</i>

### 3.3.1.11. Use Case Lihat Artikel Edukasi

Use Case	: Lihat Artikel Edukasi
Nama Aktor	: Publik (Pengguna Umum)
Keterangan	: Proses <i>user</i> melihat artikel edukasi
Output	: <i>User</i> berhasil melihat artikel edukasi

**Tabel 3.13 Use Case Lihat Artikel Edukasi**

Use Case Utama		
No	User	Sistem
1.	<i>User</i> memilih fitur artikel di website Gizila melalui browser	Sistem menampilkan daftar artikel di website Gizila
Opsi tambahan: Akses melalui laman utama atau menu paginasi		
2.	<i>User</i> membuka salah satu judul artikel	Sistem menampilkan konten artikel secara lengkap dalam slug halaman artikel yang dipilih

### 3.3.1.12. Use Case Mendapat Info Edukasi di Website Gizila

Use Case	: Mendapat Info Edukasi di Website Gizila
Nama Aktor	: Publik (Pengguna Umum)
Keterangan	: Proses <i>user</i> mencari informasi artikel edukasi
Output	: <i>User</i> berhasil mendapat informasi artikel edukasi

**Tabel 3.14 Use Case Mendapat Info Edukasi di Website Gizila**

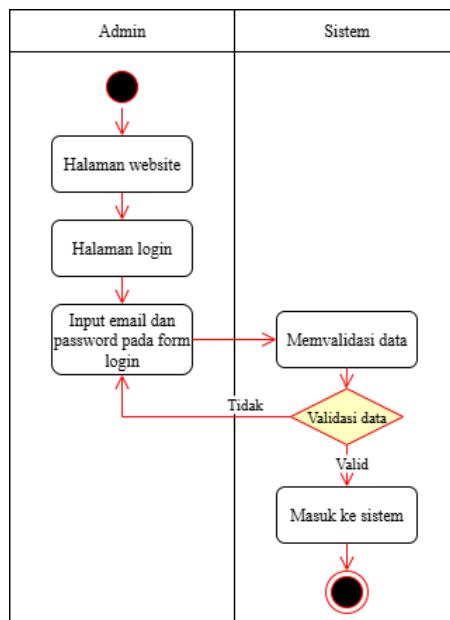
Use Case Utama		
No	User	Sistem
1.	<i>User</i> mencari informasi gizi melalui artikel di website Gizila melalui browser	Sistem menampilkan hasil pencarian atau rekomendasi berdasarkan kategori
Opsi tambahan: Pencarian manual seperti scroll halaman artikel		
2.	<i>User</i> membaca isi informasi artikel	Sistem menampilkan informasi berdasarkan artikel edukasi yang dipilih

### 3.3.2. Activity Diagram

*Activity diagram* berfungsi untuk memberikan visualisasi alur proses dalam sistem secara menyeluruh, khususnya pada pengembangan sistem informasi Gizila untuk pemenuhan kebutuhan gizi harian berbasis web. Dengan adanya diagram aktivitas ini, proses kerja dalam sistem menjadi lebih mudah dipahami, mulai dari langkah awal hingga akhir. Diagram ini juga berguna untuk mengidentifikasi kondisi pengambilan keputusan, proses paralel, serta urutan aktivitas yang harus dilakukan agar sistem berjalan secara optimal dan sesuai dengan kebutuhan pengguna. Berikut ini merupakan *activity diagram* yang dirancang dalam tahapan perancangan sistem informasi Gizila pada penelitian ini.

#### 3.3.2.1. Activity Diagram Login Admin

Berikut diagram aktivitas yang mendeskripsikan proses *login* admin:



Gambar 3.4 Activity Diagram Login Admin

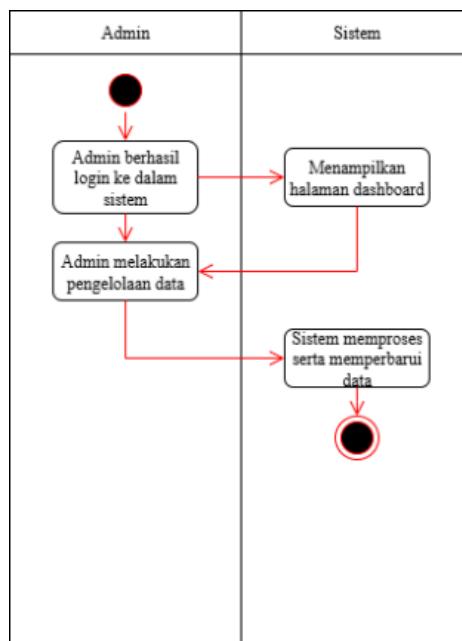
Diagram aktivitas *login* admin berdasarkan gambar 3.4 di atas menggambarkan alur proses masuk ke dalam sistem, berikut penjelasannya:

- a. Admin mengakses halaman *login* website Gizila melalui browser. Proses ini menandai awal dari interaksi antara admin dan sistem.
- b. Di halaman ini tersedia formulir yang harus diisi untuk dapat masuk ke dalam sistem yaitu admin memasukkan *email* dan *password* pada formulir *login*.
- c. Data ini kemudian dikirim ke sistem untuk dilakukan proses validasi.

- d. Sistem menerima input dari admin dan langsung menjalankan proses validasi data dengan mencocokkan data yang dimasukkan dengan data yang tersimpan di dalam basis data (*database*).
- e. Setelah proses validasi selesai, jika data yang dimasukkan tidak valid (salah *email* atau *password*), maka sistem mengembalikan proses ke *form login* dan admin diminta untuk menginput ulang data yang benar. Jika data yang dimasukkan valid, maka sistem akan memberikan akses dan mengarahkan admin masuk ke dalam sistem, yaitu ke halaman *dashboard* utama.
- f. Proses *login* dianggap berhasil jika admin berhasil masuk ke sistem dan dapat mulai menggunakan fitur-fitur yang tersedia di *dashboard* admin Gizila.

### 3.3.2.2. *Activity Diagram Akses Dashboard Admin Gizila*

Berikut diagram aktivitas yang mendeskripsikan proses akses *dashboard* admin Gizila:



**Gambar 3.5 Activity Diagram Akses Dashboard Admin Gizila**

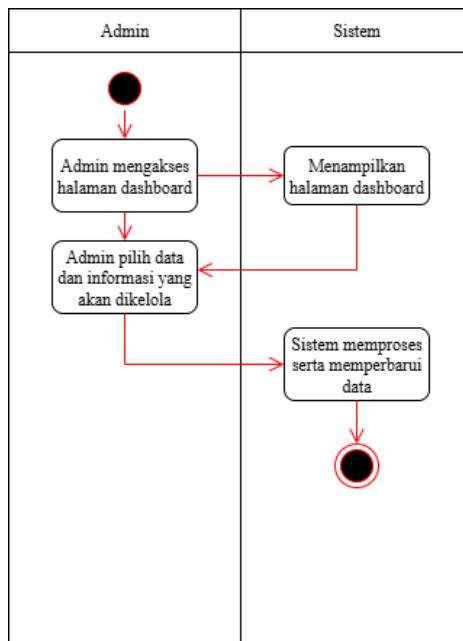
Diagram aktivitas akses *dashboard* admin Gizila berdasarkan gambar 3.5 di atas menggambarkan alur proses akses *dashboard* oleh admin, berikut penjelasannya:

- a. Admin berhasil *login* ke dalam sistem. Tahap ini menandai bahwa admin sudah melalui proses validasi data sebelumnya.
- b. Sistem menampilkan halaman *dashboard*.

- c. Setelah *login* berhasil, sistem secara otomatis mengarahkan admin ke halaman *dashboard* utama. Halaman ini menjadi akses untuk semua fitur yang tersedia bagi admin, termasuk pengelolaan data dan informasi di *dashboard* admin Gizila.
- d. Admin dapat melakukan berbagai aktivitas pengelolaan data seperti menambah, mengubah, membaca, atau menghapus data yang berkaitan dengan sistem.
- e. Sistem memproses serta memperbarui data.

### 3.3.2.3. *Activity Diagram Kelola Dashboard* admin Gizila

Berikut diagram aktivitas yang mendeskripsikan proses kelola *dashboard* admin Gizila:



**Gambar 3.6** *Activity Diagram Kelola Dashboard* Admin Gizila

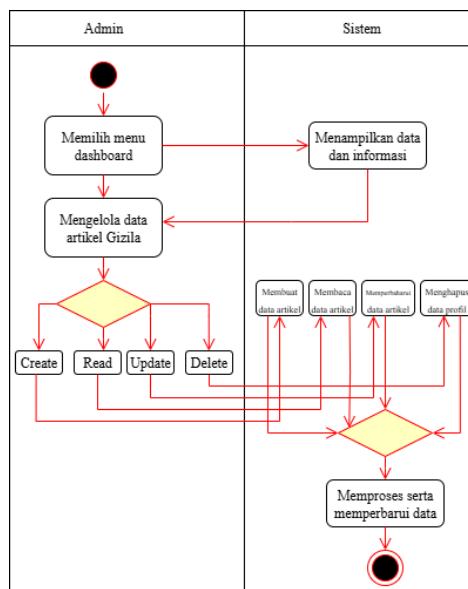
Diagram aktivitas kelola *dashboard* admin Gizila berdasarkan gambar 3.6 di atas menggambarkan alur proses aktivitas ketika admin melakukan pengelolaan data dan informasi pada *dashboard* admin Gizila, berikut penjelasannya:

- a. Admin mengakses halaman *dashboard*.
- b. Admin berhasil masuk ke sistem dan mengelola halaman *dashboard*. Halaman ini merupakan tampilan sistem yang menyediakan berbagai fitur manajemen data dan informasi yang tersedia dalam *dashboard* admin Gizila.
- c. Sistem menampilkan halaman *dashboard*.

- d. Selanjutnya, admin memilih salah satu data atau fitur informasi yang tersedia di halaman dashboard. Pilihan ini bergantung pada kebutuhan manajemen data saat itu.
- e. Setelah admin melakukan tindakan terhadap data, sistem akan memproses perintah tersebut. Misalnya, sistem menyimpan data baru, memperbarui data lama, atau menghapus data yang sudah tidak relevan. Proses ini juga memastikan bahwa semua perubahan tersimpan dengan benar di dalam *database*.

#### **3.3.2.4. Activity Diagram Kelola Fitur CRUD Artikel di Dashboard**

Berikut diagram aktivitas yang mendeskripsikan proses kelola fitur CRUD artikel di *dashboard* admin Gizila:



**Gambar 3.7 Activity Diagram Kelola Fitur CRUD Artikel di Dashboard**

Diagram aktivitas kelola fitur CRUD artikel di *dashboard* admin Gizila berdasarkan gambar 3.7 di atas menggambarkan alur proses yang dilakukan oleh admin dalam mengelola artikel di dalam *dashboard*. Proses ini menggunakan fitur CRUD (*Create, Read, Update, Delete*), yang merupakan proses utama dalam manajemen konten berbasis web, berikut penjelasannya:

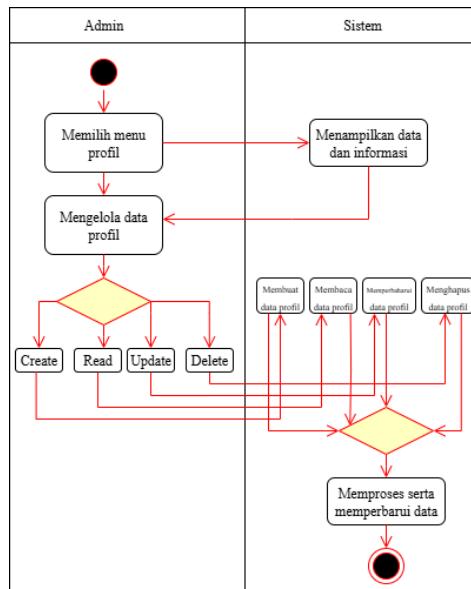
- a. admin mengakses sistem dan memilih menu *dashboard*, sistem kemudian merespon dengan menampilkan data dan informasi yang tersedia pada *dashboard*, termasuk daftar artikel yang sudah ada.
- b. admin akan menentukan tindakan apa yang akan dilakukan terhadap artikel, yang selanjutnya masuk kelola fitur CRUD artikel yaitu *create* untuk admin

memilih opsi *create* untuk menambahkan artikel baru maupun kategori artikel baru. Kemudian sistem menampilkan formulir entri artikel, kemudian menyimpan data artikel ke dalam sistem. *Read* untuk admin melihat artikel yang sudah tersedia. Kemudian sistem menampilkan data artikel yang dipilih agar dapat ditinjau atau dibaca. Pada bagian *update* admin memilih opsi *update* untuk memperbarui data artikel. Kemudian sistem menyimpan pembaruan artikel setelah admin selesai melakukan modifikasi. *Delete* untuk admin memilih opsi *delete* untuk menghapus artikel yang tidak relevan, kemudian sistem menghapus data artikel.

- c. Setelah salah satu aksi (CRUD) dipilih dan dijalankan, sistem akan masuk ke tahap pemrosesan dan pembaruan data. Proses ini dilakukan di *backend* sistem untuk memastikan bahwa semua perubahan terekam dan tersimpan dengan benar dalam basis data (*database*).

### 3.3.2.5. Activity Diagram Kelola Profil

Berikut diagram aktivitas yang mendeskripsikan proses kelola *website* Gizila:



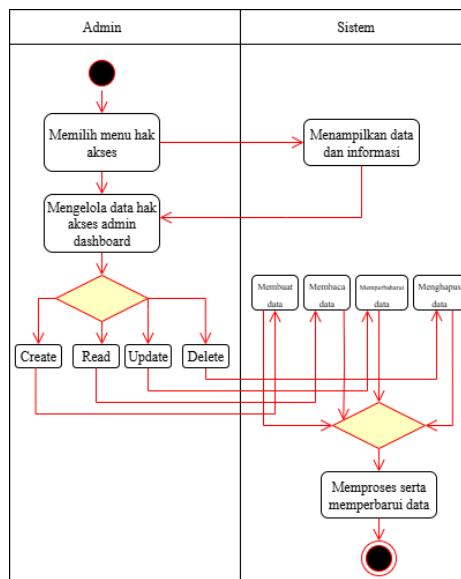
Gambar 3.8 *Activity Diagram Kelola Profil*

Diagram aktivitas kelola fitur profil pada *dashboard* admin Gizila berdasarkan gambar 3.8 di atas menggambarkan alur proses yang dilakukan oleh admin dalam mengelola data profil. Proses ini menggunakan fitur CRUD (*Create, Read, Update, Delete*), yang merupakan inti dari manajemen data berbasis web. Berikut penjelasannya:

- Admin mengakses sistem dan memilih menu profil, kemudian sistem merespons dengan menampilkan data dan informasi profil yang tersedia.
- Admin masuk ke tahap mengelola data profil, di mana admin dapat menentukan tindakan yang akan dilakukan menggunakan fitur CRUD yaitu *create* untuk admin membuat data profil baru. Sistem kemudian menampilkan formulir entri data, dan setelah data diisi, sistem menyimpan data profil ke dalam *database*. *Read* untuk admin melihat data profil yang sudah ada. Sistem kemudian menampilkan data profil agar dapat ditinjau. *Update* untuk admin memperbarui data profil yang sudah tersimpan. *Delete* untuk admin memilih opsi *delete* untuk menghapus hak akses admin yang tidak relevan. Setelah admin melakukan perubahan, sistem menyimpan pembaruan data.
- Setelah salah satu aksi CRUD dijalankan, sistem akan masuk ke tahap pemrosesan dan pembaruan data. Proses ini dilakukan oleh sistem di bagian *backend* untuk memastikan semua perubahan dengan benar dan tersimpan dalam *database*.

### 3.3.2.6. Activity Diagram Kelola Hak Akses Admin Dashboard

Berikut diagram aktivitas yang mendeskripsikan proses kelola *website* Gizila:



**Gambar 3.9** Activity Diagram Kelola Hak Akses

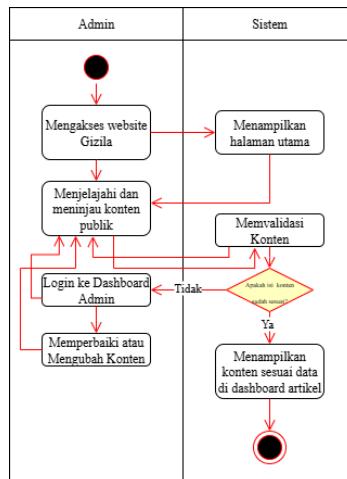
Diagram aktivitas kelola fitur hak akses pada *dashboard* admin Gizila berdasarkan gambar 3.9 di atas menggambarkan alur proses yang dilakukan oleh

admin dalam mengelola data hak akses. Proses ini menggunakan CRUD (*Create, Read, Update, Delete*) yang menjadi inti dari manajemen data berbasis web. Berikut penjelasannya:

- a. Admin mengakses sistem dan memilih menu hak akses, kemudian sistem merespon dengan menampilkan data dan informasi hak akses yang tersedia pada *dashboard*.
- b. Admin masuk ke tahap mengelola data hak akses, di mana admin dapat menentukan tindakan yang akan dilakukan menggunakan fitur CRUD yaitu *create* untuk admin membuat data hak akses baru. Sistem kemudian menampilkan formulir entri data, dan setelah diisi, sistem menyimpan data hak akses tersebut ke dalam basis data. *Read* untuk admin melihat data hak akses yang sudah ada. Sistem kemudian menampilkan data hak akses yang dipilih agar dapat ditinjau. *Update* untuk admin memperbarui data hak akses yang sudah tersimpan. Setelah admin melakukan perubahan, sistem menyimpan pembaruan data tersebut. *Delete* untuk admin menghapus data hak akses yang tidak relevan. Sistem kemudian menghapus data tersebut dari *database*.
- c. Setelah salah satu aksi CRUD dijalankan, sistem akan masuk ke tahap pemrosesan dan pembaruan data. Proses ini dilakukan oleh sistem di bagian *backend* untuk memastikan semua perubahan data tersimpan dengan benar di dalam *database*.

#### **3.3.2.7. Activity Diagram Kelola Website Gizila**

Berikut diagram aktivitas yang mendeskripsikan proses kelola *website* Gizila:



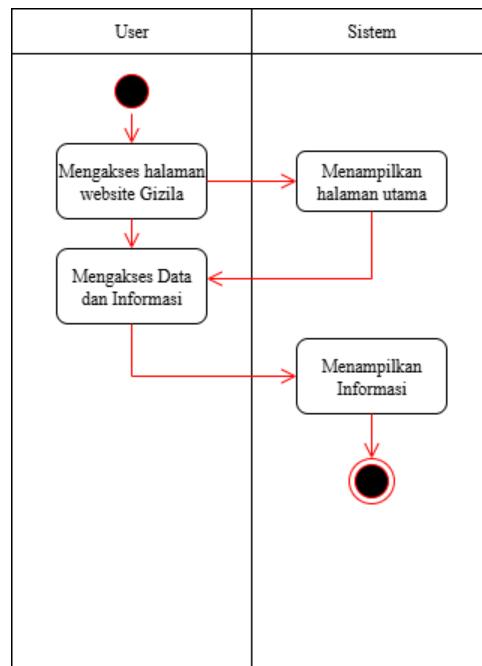
**Gambar 3.10** Activity Diagram Kelola Website Gizila

Diagram aktivitas kelola *website* Gizila berdasarkan gambar 3.10 di atas menggambarkan alur proses aktivitas admin dalam mengelola tampilan konten pada *website* Gizila, berikut penjelasannya:

- a. Admin mengakses *website* Gizila melalui browser.
- b. Sistem merespon dengan menampilkan halaman utama *website*, yang berisi konten informasi umum.
- c. Admin dapat langsung menjelajahi dan meninjau konten publik tanpa harus login terlebih dahulu.
- d. Admin melakukan proses validasi konten yaitu melakukan pemeriksaan: "apakah isi konten sudah sesuai atau perlu perbaikan?". Jika "ya" konten sudah sesuai, maka admin tidak perlu *login* ke *dashboard* admin. Jika "tidak" konten tidak sesuai, maka admin diarahkan untuk memperbaiki atau mengubah konten yaitu dengan *login* ke *dashboard* admin.
- e. Setelah admin selesai mengedit, sistem akan kembali memvalidasi dan menampilkan konten yang sudah diperbarui.

### 3.3.2.8. *Activity Diagram Akses Website Gizila*

Berikut diagram aktivitas yang mendeskripsikan akses *website* Gizila:



**Gambar 3.11** *Activity Diagram Akses Website Gizila*

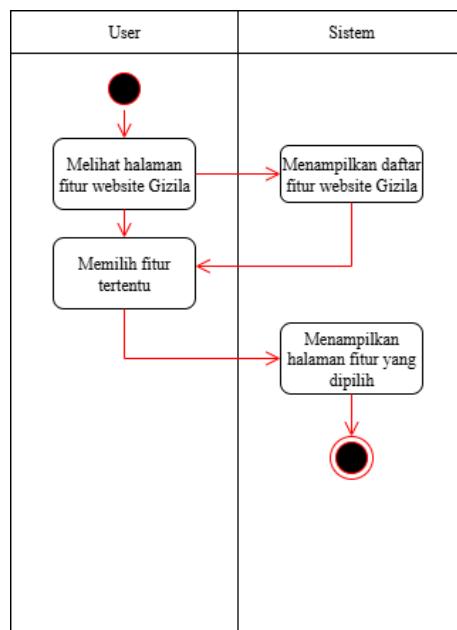
Diagram aktivitas akses *website* Gizila berdasarkan gambar 3.11 di atas menggambarkan alur proses aktivitas pengguna umum (*user*) ketika mengakses web Gizila, berikut penjelasannya:

- a. *User* mengakses halaman *website* Gizila melalui browser.
- b. Sistem menampilkan halaman utama dari *website* Gizila beserta fitur-fitur yang tersedia.
- c. Setelah berada di halaman utama, *user* memilih dan mengakses fitur-fitur web Gizila yang tersedia (misalnya kalkulator gizi, artikel edukasi dan rekomendasi menu harian).

Sistem kemudian menampilkan informasi yang diminta oleh *user* secara dinamis dari *database website*.

### 3.3.2.9. Activity Diagram Lihat Fitur-fitur Website Gizila

Berikut diagram aktivitas yang mendeskripsikan lihat fitur-fitur *website* Gizila:



**Gambar 3.12** Activity Diagram Lihat Fitur-fitur Website Gizila

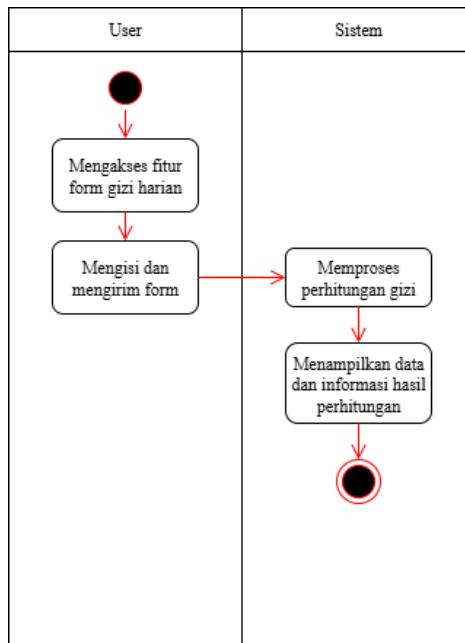
Diagram aktivitas lihat fitur-fitur *website* Gizila berdasarkan gambar 3.12 di atas menggambarkan alur proses aktivitas pengguna umum (*user*) saat melihat dan mengakses fitur-fitur yang tersedia di web Gizila, berikut penjelasannya:

- a. *User* mengakses halaman utama atau navigasi fitur dari *website* Gizila.
- b. Sistem menampilkan daftar fitur *website* Gizila.

- c. *User* memilih salah satu fitur dari daftar tersebut berdasarkan kebutuhannya.
- d. Setelah fitur berhasil ditampilkan, *user* dapat melihat dan menggunakan fitur tersebut yang berupa data maupun informasi yang ada.

### 3.3.2.10. *Activity Diagram Menghitung Gizi Harian*

Berikut diagram aktivitas yang mendeskripsikan fitur menghitung gizi harian:



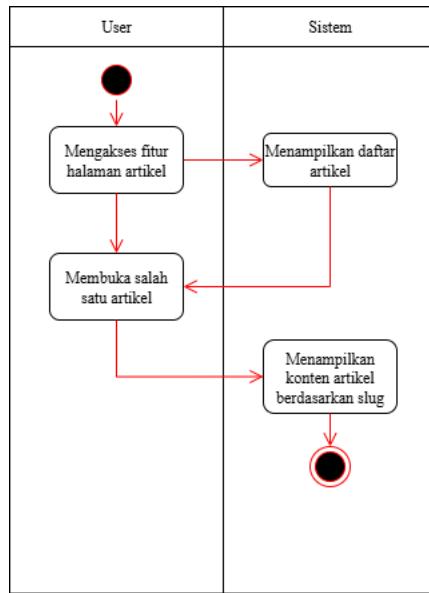
**Gambar 3.13** *Activity Diagram Menghitung Gizi Harian*

Diagram aktivitas menghitung gizi harian berdasarkan gambar 3.13 di atas menggambarkan alur proses interaksi pengguna umum (*user*) dalam menggunakan fitur perhitungan gizi harian di web Gizila, berikut penjelasannya:

- a. *User* membuka halaman kalkulator gizi harian pada website Gizila. Halaman ini berisi formulir untuk mengisi data individu yang dibutuhkan dalam perhitungan kebutuhan gizi, seperti: usia, jenis kelamin, tinggi badan, berat badan, dan tingkat aktivitas harian.
- b. *User* mengisi, dan mengirim *form* dengan mengklik tombol *submit* atau *hitung*.
- c. Sistem menerima data input dan mulai menghitung kebutuhan gizi harian berdasarkan *logic* menggunakan rumus rumus *Harris-Benedict*.
- d. Sistem menampilkan data dan informasi hasil perhitungan

### 3.3.2.11. *Activity Diagram Lihat Artikel Edukasi*

Berikut diagram aktivitas yang mendeskripsikan lihat artikel edukasi:



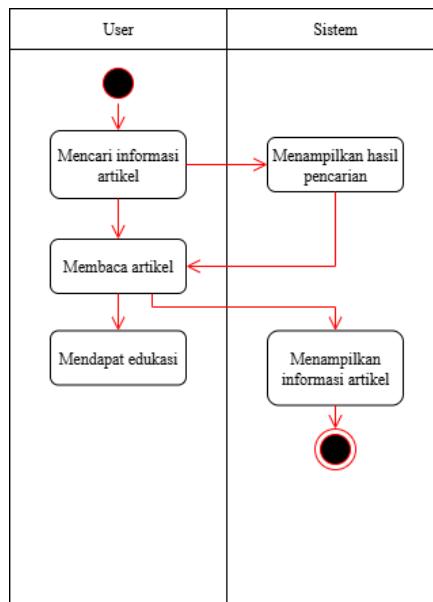
**Gambar 3.14 Activity Diagram Lihat Artikel Edukasi**

Diagram aktivitas menghitung gizi harian berdasarkan gambar 3.14 di atas menggambarkan alur proses interaksi pengguna umum (*user*) ketika mengakses fitur artikel edukasi di web Gizila, berikut penjelasannya:

- User* membuka halaman artikel pada *website* Gizila. Halaman ini menampilkan daftar artikel edukatif yang telah tersedia.
- Sistem menampilkan daftar artikel.
- User* memilih salah satu artikel berdasarkan minat atau kebutuhan informasi. Proses ini dilakukan dengan membuka artikel dengan klik *link* artikel yang diinginkan.
- Sistem menampilkan konten artikel berdasarkan slug.

### 3.3.2.12. *Activity Diagram Mendapat Info Edukasi di Website Gizila*

Berikut diagram aktivitas yang mendeskripsikan *user* mendapat edukasi di *website* Gizila:



**Gambar 3.15** *Activity Diagram* Mendapat Info Edukasi di *Website* *Gizila*

Diagram aktivitas menghitung gizi harian berdasarkan gambar 3.15 di atas menggambarkan alur proses interaksi saat pengguna umum (*user*) mencari informasi edukatif melalui artikel di web *Gizila*, berikut penjelasannya:

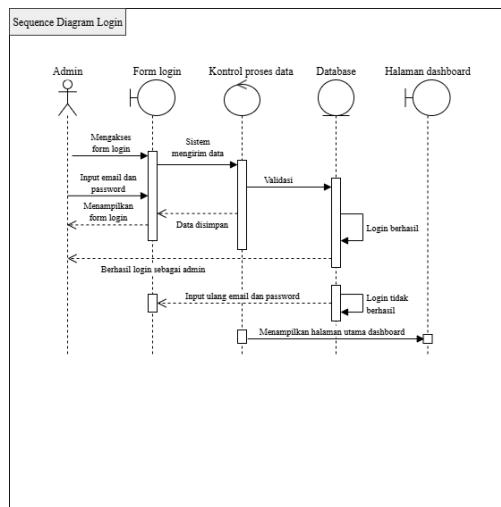
- User* melakukan pencarian informasi, misalnya dengan mengetikkan kata kunci pada kolom pencarian artikel.
- Sistem menampilkan hasil pencarian.
- User* membaca artikel.
- User* mendapatkan edukasi dari artikel yang telah dibaca.

### 3.3.3. *Sequence Diagram*

*Sequence diagram* berfungsi untuk memvisualisasikan interaksi antar objek atau komponen dalam sistem berdasarkan urutan waktu, khususnya pada pengembangan Sistem Informasi *Gizila* untuk pemenuhan kebutuhan gizi harian berbasis web. Melalui diagram ini, proses komunikasi antar entitas, seperti pengguna, sistem, yang dapat digambarkan secara rurut dan sistematis. *Sequence diagram* membantu mengidentifikasi urutan pesan yang dikirim dan diterima dalam menjalankan suatu skenario fungsional, sehingga mempermudah pemahaman alur sistem dari sisi implementasi logika proses. Diagram ini sangat berguna dalam tahap perancangan sistem, karena memberikan gambaran yang jelas tentang bagaimana komponen saling berinteraksi untuk memenuhi kebutuhan pengguna dalam penelitian ini.

#### 3.3.3.1. *Sequence Diagram Login Admin*

Berikut alur *sequence diagram* yang mendeskripsikan *login* admin:



**Gambar 3.16 Sequence Diagram Login Admin**

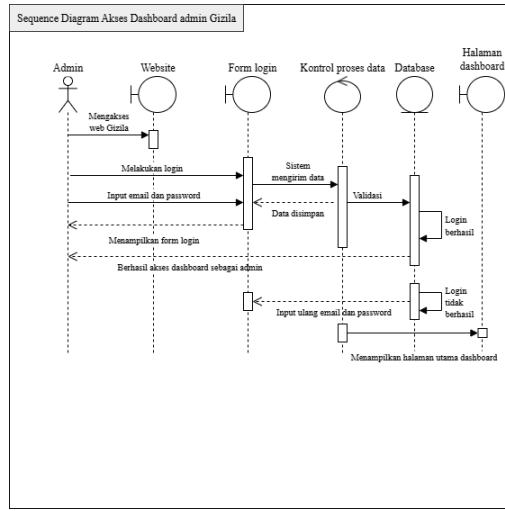
*Sequence diagram login admin berdasarkan gambar 3.16 di atas, proses login dimulai ketika admin mengakses halaman *form login* pada sistem. Admin kemudian memasukkan *email* dan *password* melalui *form login* yang ditampilkan oleh sistem. Data yang telah diinput oleh admin akan dikirimkan oleh *form login* ke kontrol proses data, yang kemudian akan melakukan proses validasi data ke database.*

*Jika hasil validasi yang dikembalikan dari database menyatakan *login* berhasil (*email* dan *password* sesuai dengan data yang tersimpan), maka kontrol proses data akan mengirimkan respon ke *form login*, dan sistem akan menampilkan halaman utama *dashboard* sebagai tanda bahwa admin berhasil *login*. Selanjutnya admin dapat mengakses berbagai fitur yang tersedia pada *dashboard*.*

*Sebaliknya, jika hasil validasi menyatakan *login* gagal karena *email* atau *password* salah atau tidak terdaftar, maka kontrol proses data akan mengirimkan informasi *login* gagal ke *form login* yang menampilkan pesan kesalahan (*error*) di halaman *login*, meminta admin untuk menginput ulang *email* dan *password* yang benar. Proses ini akan terus berulang sampai admin berhasil *login* ke dalam sistem dengan input *email* dan *password* sesuai data yang tersimpan di *database*.*

### 3.3.3.2. Sequence Diagram Akses Dashboard admin Gizila

Berikut alur *sequence diagram* yang mendeskripsikan akses *dashboard* admin Gizila:



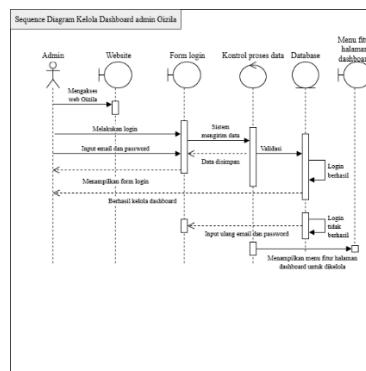
**Gambar 3.17 Sequence Diagram Akses Dashboard admin Gizila**

*Sequence diagram akses dashboard admin berdasarkan gambar 3.17 di atas, proses dimulai ketika admin mengakses website Gizila. Setelah halaman website terbuka, admin melakukan login dengan menginput email dan password pada form login yang tersedia. Website kemudian menampilkan form login dan mengirimkan data yang telah diinput ke bagian kontrol proses data.*

Selanjutnya, kontrol proses data akan meneruskan data tersebut ke *database* untuk divalidasi. Jika hasil validasi menyatakan bahwa data *email* dan *password* sesuai dengan yang tersimpan dalam *database*, maka sistem akan menampilkan halaman utama *dashboard* admin Gizila. Namun, apabila hasil validasi menyatakan bahwa data tidak sesuai atau tidak terdaftar, maka admin diminta untuk menginput ulang *email* dan *password* yang benar.

### 3.3.3.3. Sequence Diagram Kelola Dashboard admin Gizila

Berikut alur *sequence diagram* yang mendeskripsikan kelola *dashboard* admin Gizila:



**Gambar 3.18 Sequence Diagram Kelola Dashboard admin Gizila**

*Sequence diagram* kelola *dashboard* admin berdasarkan gambar 3.18 di atas, proses diawali ketika admin mengakses website Gizila. Setelah halaman website terbuka, admin melakukan *login* dengan menginput *email* dan *password* melalui *form login* yang tersedia.

Website akan menampilkan *form login*, dan data yang diinput oleh admin akan dikirimkan oleh *form login* ke bagian kontrol proses data. Selanjutnya, kontrol proses data akan mengirimkan data tersebut ke *database* untuk dilakukan proses validasi.

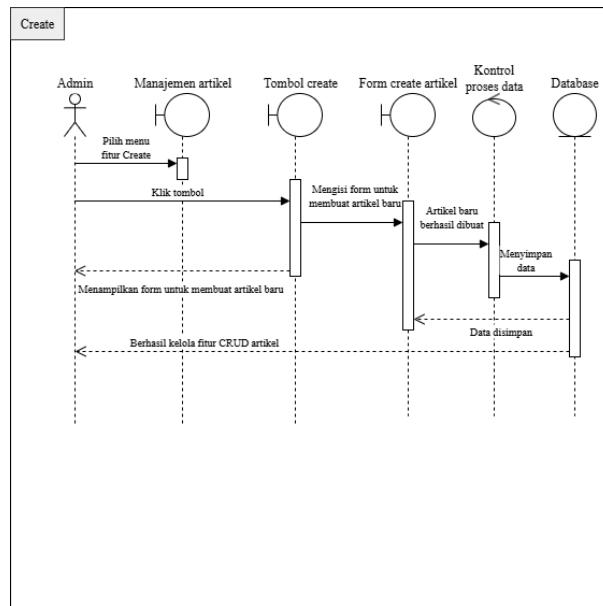
Jika hasil validasi menyatakan bahwa data *email* dan *password* sesuai dengan yang tersimpan di dalam *database*, maka proses *login* dinyatakan berhasil. Sistem kemudian menampilkan tampilan *dashboard* juga menampilkan menu fitur halaman *dashboard* yang dapat dikelola oleh admin.

Namun, apabila hasil validasi menyatakan login tidak berhasil karena email atau password salah, maka sistem akan menampilkan pesan *error* pada *form login* dan meminta admin untuk menginput ulang data login yang benar. Jika berhasil, maka barulah admin bisa mengelola berbagai fitur pada *dashboard* admin Gizila.

#### 3.3.3.4. *Sequence Diagram* Kelola Fitur CRUD Artikel di *Dashboard*

Berikut *sequence diagram* yang mendeskripsikan proses kelola fitur CRUD artikel di *dashboard* admin Gizila:

##### a. *Create*



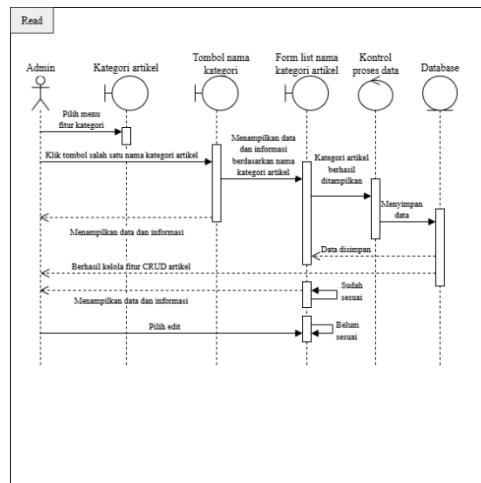
Gambar 3.19 *Sequence Diagram Create*

*Sequence diagram* kelola fitur CRUD artikel bagian *create* berdasarkan gambar 3.19 di atas, proses dimulai ketika admin memilih menu fitur *create* pada halaman manajemen artikel. Setelah itu, admin menekan tombol *create* untuk memulai proses pembuatan artikel baru.

Sistem kemudian menampilkan *form create* artikel yang dapat diisi oleh admin. Admin mengisi *form* tersebut dengan konten artikel yang ingin dibuat, lalu mengirimkannya untuk diproses. *Form create* artikel akan mengirimkan data artikel ke kontrol proses data.

Setelah menerima data, kontrol proses data akan memproses dan menyatakan bahwa artikel baru berhasil dibuat. Kemudian, kontrol proses data menyimpan data artikel tersebut ke dalam *database*. Kemudian *database* menyimpan data yang telah dikirimkan.

#### b. *Read*



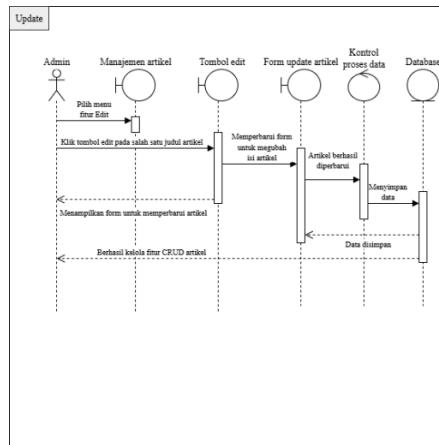
Gambar 3.20 Sequence Diagram *Read*

*Sequence diagram* kelola fitur CRUD artikel bagian *read* berdasarkan gambar 3.20 di atas, proses dimulai saat admin memilih menu fitur kategori artikel pada halaman manajemen artikel. Selanjutnya, admin mengklik salah satu tombol nama kategori artikel yang tersedia.

Setelah itu, sistem akan memproses permintaan dan menampilkan data serta informasi artikel berdasarkan kategori yang dipilih melalui *form list* nama kategori artikel. Kemudian admin meninjau isi data dan infromasi yang tersedia apakah isinya sudah sesuai atau belum. Apabila data artikel sudah sesuai maka tidak perlu melakukan pembaruan data. Namun jika belum sesuai, admin dapat memilih tombol *edit* untuk memperbarui isi artikel tersebut. Kemudian setelah

melakukan pembaruan, data tersebut akan meneruskan permintaan ke kontrol proses data, yang selanjutnya akan disimpan kedalam *database*.

#### c. Update



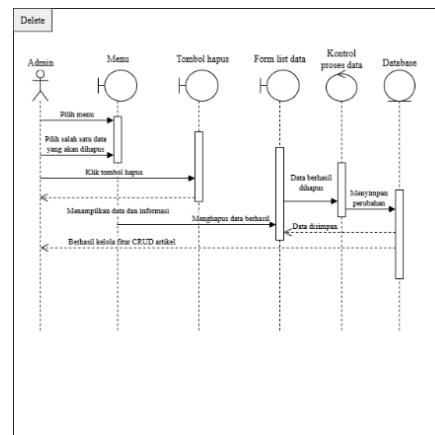
**Gambar 3.21 Sequence Diagram Update**

*Sequence diagram* kelola fitur CRUD artikel bagian *update* berdasarkan gambar 3.21 di atas, proses dimulai ketika admin memilih menu fitur edit pada halaman manajemen artikel. Admin kemudian mengklik tombol edit pada salah satu judul artikel yang ingin diperbarui.

Setelah itu, sistem akan menampilkan *form update* artikel yang dapat diisi ulang oleh admin sesuai dengan perubahan isi artikel yang diinginkan. Admin memperbarui *form* tersebut untuk mengubah konten artikel.

*Form update* artikel kemudian mengirimkan data pembaruan tersebut ke kontrol proses data. Selanjutnya, diproses dan kemudian dilanjutkan dengan menyimpan data pembaruan ke dalam *database*.

#### d. Delete



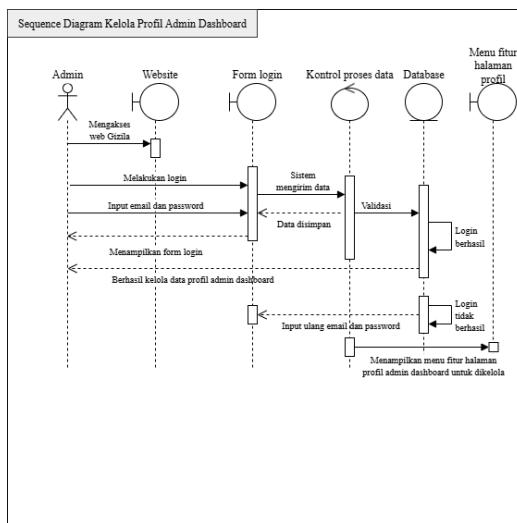
**Gambar 3.22 Sequence Diagram Delete**

*Sequence diagram* kelola fitur CRUD artikel bagian *delete* berdasarkan gambar 3.22 di atas, proses dimulai ketika admin memilih menu untuk menampilkan data artikel yang tersedia. Admin kemudian memilih salah satu data artikel yang akan dihapus dan mengklik tombol hapus.

Setelah itu, sistem meneruskan perintah untuk menghapus data yang dipilih oleh admin yang kemudian dikirimkan ke kontrol proses data untuk diproses penghapusan sehingga data berhasil dihapus. Data yang telah dihapus kemudian dikonfirmasi dan disimpan sebagai perubahan ke dalam *database*.

### 3.3.3.5. Sequence Diagram Kelola Profil

Berikut alur *sequence diagram* yang mendeskripsikan kelola profil:



Gambar 3.23 *Sequence Diagram* Kelola Profil

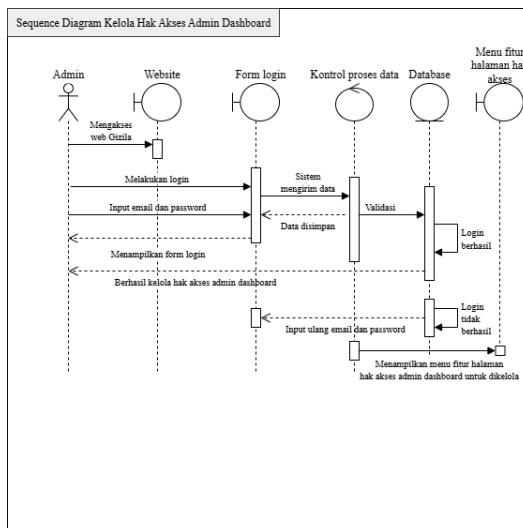
*Sequence diagram* kelola profil admin *dashboard* berdasarkan gambar 3.23 di atas menjelaskan alur proses yang dilakukan oleh admin ketika ingin mengakses dan mengelola data profil pada *dashboard* admin *Gizila*. Proses dimulai ketika admin mengakses *website* *Gizila*. Setelah itu, admin melakukan proses *login* dengan memasukkan *email* dan *password* pada *form login* yang telah disediakan.

Sistem kemudian akan mengirimkan data *login* ke kontrol proses data untuk divalidasi pada *database*. Apabila data *login* sesuai (*valid*), maka proses login dinyatakan berhasil dan sistem akan menampilkan *dashboard* dan admin memilih menu fitur halaman profil yang dapat dikelola oleh admin. Selanjutnya, admin dapat mengelola data profil sesuai kebutuhan.

Namun, apabila data login tidak sesuai (tidak *valid*), maka proses *login* dinyatakan gagal. Sistem akan menampilkan notifikasi error dan meminta admin untuk menginput ulang *email* dan *password* yang benar agar dapat melanjutkan proses *login*.

### 3.3.3.6. Sequence Diagram Kelola Hak Akses Admin Dashboard

Berikut alur *sequence diagram* yang mendeskripsikan kelola hak akses admin *dashboard*:



**Gambar 3.24** Sequence Diagram Kelola Hak Akses Admin

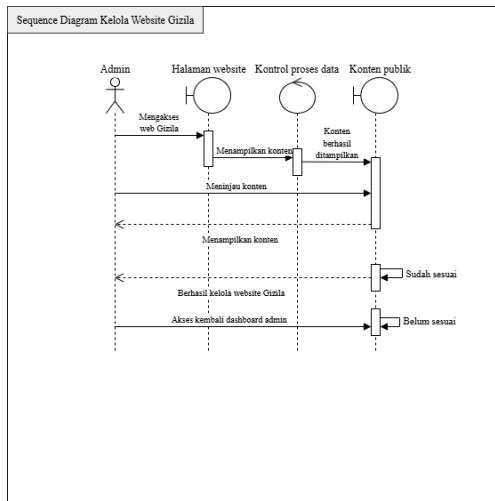
*Sequence diagram* kelola hak akses admin *dashboard* berdasarkan gambar 3.24 di atas, proses dimulai ketika admin mengakses *website* *Gizila*. Setelah itu, admin melakukan *login* dengan menginput *email* dan *password* pada *form login* yang tersedia.

Sistem kemudian akan mengirimkan data *login* ke kontrol proses data untuk diverifikasi pada *database*. Jika data *login* sesuai dan valid, maka proses *login* dinyatakan berhasil. Sistem selanjutnya akan menampilkan *dashboard* dan admin memilih menu fitur halaman hak akses admin yang dapat dikelola oleh admin. Selanjutnya, admin dapat mengelola data hak akses admin sesuai kebutuhan.

Namun, jika data *login* tidak valid, maka sistem menolak *login* dan menampilkan notifikasi error. Pada tahap ini, admin diminta untuk menginput ulang *email* dan *password* yang benar agar dapat mencoba masuk kembali ke sistem.

### 3.3.3.7. Sequence Diagram Kelola Website Gizila

Berikut alur *sequence diagram* yang mendeskripsikan kelola website Gizila:



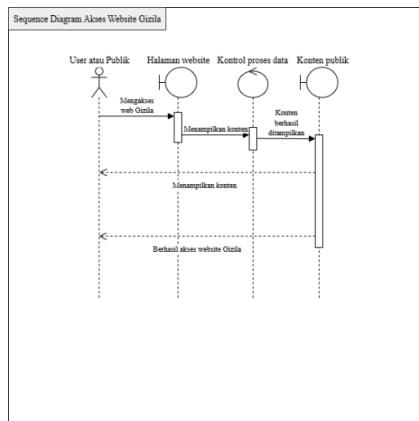
**Gambar 3.25 Sequence Diagram Kelola Website Gizila**

*Sequence diagram* kelola website Gizila berdasarkan gambar 3.25 di atas, proses dimulai saat admin mengakses halaman utama website Gizila. Kemudian sistem menampilkan konten yang tersedia di halaman website.

Admin kemudian melakukan peninjauan terhadap konten artikel yang ditampilkan pada halaman web Gizila. Jika konten sudah sesuai, maka tidak perlu tindakan perubahan di *dashboard*. Namun, jika konten belum sesuai, admin dapat kembali ke *dashboard* untuk melakukan perubahan.

### 3.3.3.8. Sequence Diagram Akses Website Gizila

Berikut alur *sequence diagram* yang mendeskripsikan akses website Gizila:



**Gambar 3.26 Sequence Diagram Akses Website Gizila**

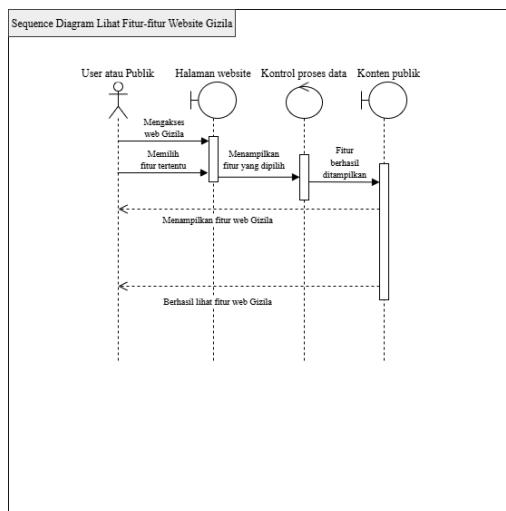
*Sequence diagram* akses website Gizila berdasarkan gambar 3.26 di atas, menjelaskan alur interaksi pengguna atau publik saat mengakses konten website Gizila. Proses diawali ketika pengguna membuka atau mengakses halaman website Gizila.

Kemudian sistem memproses dan meneruskan permintaan tersebut ke kontrol proses data, dan konten berhasil ditampilkan.

Pada proses ini menjelaskan bahwa pengguna tidak perlu *login* untuk mengakses informasi publik yang tersedia di website Gizila, sehingga memudahkan masyarakat dalam memperoleh informasi terkait kebutuhan gizi harian untuk memberikan kemudahan akses informasi secara cepat dan efisien.

### 3.3.3.9. Sequence Diagram Lihat Fitur-fitur Website Gizila

Berikut alur *sequence diagram* yang mendeskripsikan lihat fitur-fitur website Gizila:



Gambar 3.27 *Sequence Diagram* Lihat Fitur-fitur Website Gizila

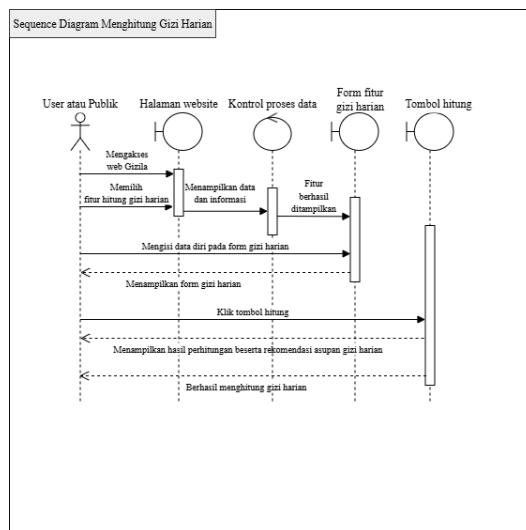
*Sequence diagram* lihat fitur-fitur website Gizila berdasarkan Gambar 3.27 di atas, menggambarkan alur interaksi ketika pengguna atau publik mengakses fitur-fitur yang tersedia di dalam website Gizila. Proses dimulai saat pengguna membuka website Gizila melalui browser.

Setelah halaman website berhasil dimuat, pengguna dapat memilih fitur tertentu yang ingin diakses. Pilihan fitur yang dipilih oleh pengguna kemudian dikirimkan ke bagian sistem, lalu diteruskan ke komponen kontrol proses data untuk memproses permintaan tersebut dan menampilkan data dan informasi pada fitur yang dipilih.

Proses ini menunjukkan bahwa pengguna dapat dengan mudah menavigasi dan memanfaatkan berbagai fitur dalam *website* Gizila tanpa perlu login terlebih dahulu untuk kemudahan akses pengguna.

### 3.3.3.10. Sequence Diagram Menghitung Gizi Harian

Berikut alur *sequence diagram* yang mendeskripsikan menghitung gizi harian:



**Gambar 3.28 Sequence Diagram Menghitung Gizi Harian**

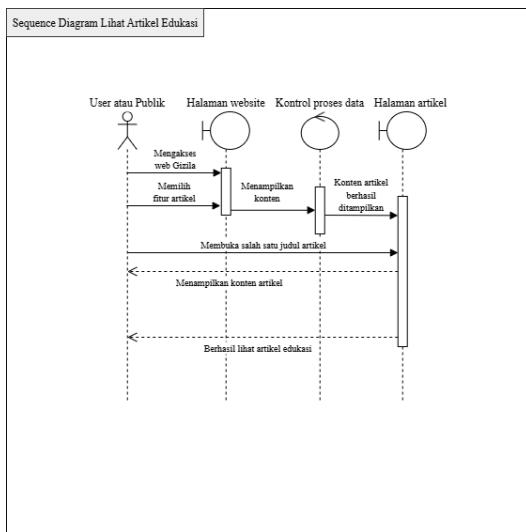
*Sequence diagram* fitur menghitung gizi harian berdasarkan gambar 3.28 di atas, menggambarkan saat pengguna atau publik mengakses halaman *website* Gizila. Setelah tampilan halaman *website* berhasil dimuat, pengguna kemudian memilih fitur perhitungan gizi harian yang tersedia pada menu utama.

Setelah pilihan fitur dipilih, sistem menampilkan data awal serta informasi yang dibutuhkan pada fitur form gizi harian: dimana pengguna melakukan input data diri seperti usia, jenis kelamin, tinggi badan, berat badan, aktivitas fisik. Setelah seluruh data diisi, pengguna menekan tombol hitung yang tersedia. Tombol hitung ini akan menjalankan fungsi pemrosesan perhitungan, yang sudah diatur pada *backend* sistem.

Setelah tombol hitung diklik, sistem akan menjalankan logika perhitungan dan menampilkan hasil akhir perhitungan kebutuhan gizi harian pengguna, disertai rekomendasi asupan nutrisi yang seimbang sesuai dengan data yang telah diinput.

### 3.3.3.11. Sequence Diagram Lihat Artikel Edukasi

Berikut alur *sequence diagram* yang mendeskripsikan lihat artikel edukasi:



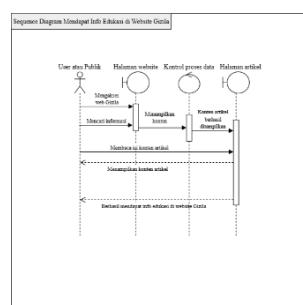
**Gambar 3.29 Sequence Diagram Lihat Artikel Edukasi**

*Sequence diagram* lihat artikel edukasi berdasarkan gambar 3.29 di atas, menggambarkan saat user atau publik mengakses halaman website Gizila. Setelah tampilan halaman *website* berhasil dimuat, pengguna memilih fitur artikel edukasi yang tersedia pada menu atau navigasi. Proses ini dilanjutkan oleh kontrol proses data, yang akan mengambil dan memproses data artikel yang tersimpan di sistem, kemudian menampilkannya pada halaman artikel.

Selanjutnya, pengguna memilih salah satu judul artikel yang ingin dibaca dengan cara membuka tautan atau klik pada judul tersebut. Setelah itu, sistem menampilkan konten lengkap artikel edukasi yang telah dipilih oleh pengguna.

### 3.3.3.12. Sequence Diagram Mendapat Info Edukasi di Website Gizila

Berikut alur *sequence diagram* yang mendeskripsikan *user* mendapat info edukasi di *website* Gizila:



**Gambar 3.30 Sequence Diagram Mendapat Info Edukasi di Website Gizila**

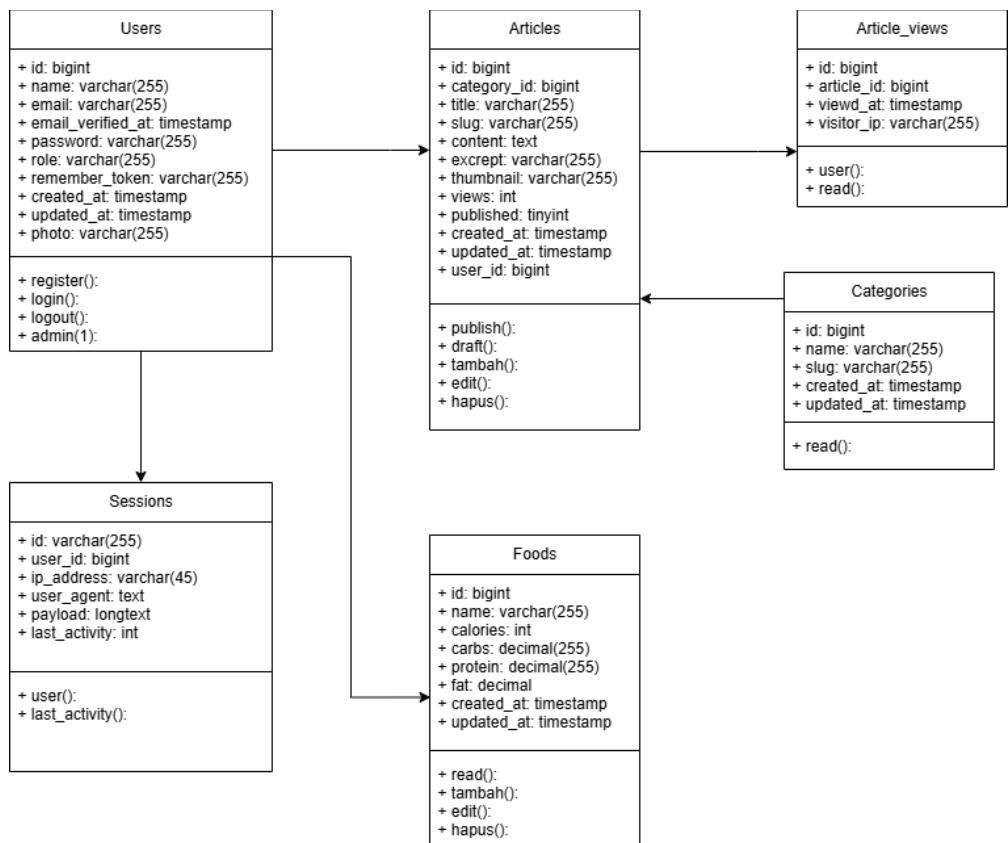
*Sequence diagram user publik mendapat info edukasi di website Gizila berdasarkan gambar 3.30 di atas, menggambarkan saat pengguna atau publik mengakses halaman website Gizila. Setelah itu, pengguna mencari informasi terkait edukasi gizi yang tersedia.*

Permintaan pencarian informasi tersebut kemudian kontrol proses data akan menampilkan konten edukasi yang sesuai dengan permintaan pengguna.

Setelah itu, pengguna dapat mulai membaca isi konten artikel edukasi yang telah dimuat sehingga pengguna telah berhasil mendapatkan informasi edukasi di website Gizila.

### 3.3.4. Class Diagram

Pada *class diagram* berdasarkan gambar 3.31 dibawah ini, diagram tersebut menggambarkan model secara keseluruhan mengenai sistem informasi Gizila yang telah dirancang. *Class diagram* digunakan untuk memberikan gambaran sistem yang dirancang secara lebih terstruktur, serta membantu dalam menyesuaikan kebutuhan antara sistem yang dioperasikan dengan pengguna.



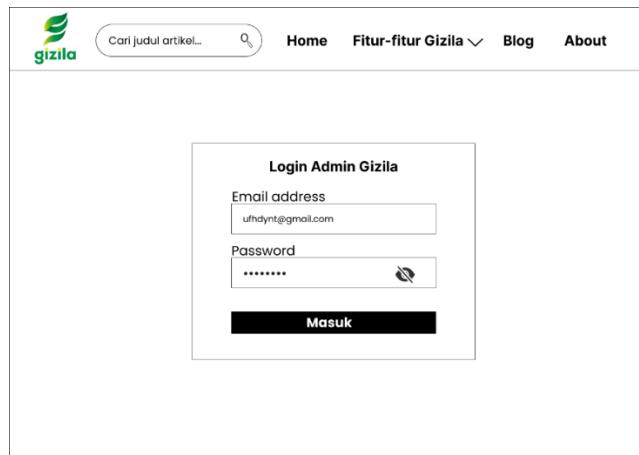
Gambar 3.31 Class Diagram

### 3.3.5. Desain Antarmuka Pengguna

Desain antarmuka pengguna pada sistem informasi Gizila dirancang menggunakan Figma sebagai *tools* utama.

#### 3.3.5.1 Desain Halaman *Login* Admin

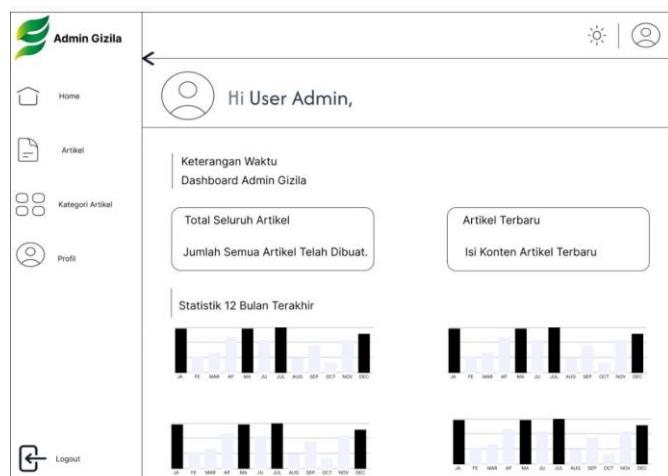
Perancangan halaman *login* admin dapat diakses khusus oleh pihak admin. Halaman ini berfungsi untuk melakukan proses masuk ke halaman *dashboard* admin yang mana data yang dimasukkan harus sesuai dengan data yang tersimpan didalam database sistem yang sudah didaftarkan sebelumnya di dalam hak akses. Sehingga user atau publik tidak bisa mengakses halaman ini.



Gambar 3.32 Desain Halaman *Login* Admin

#### 3.3.5.2. Desain Halaman Akses *Dashboard* admin Gizila

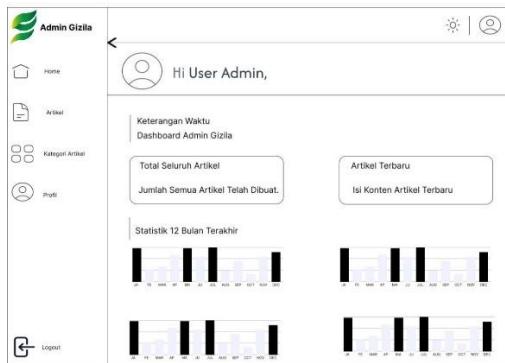
Perancangan halaman *dashboard* admin Gizila dapat diakses khusus oleh pengguna yang memiliki hak akses sebagai admin.



Gambar 3.33 Desain Halaman Akses *Dashboard* admin Gizila

### 3.3.5.3. Desain Halaman Kelola *Dashboard* admin Gizila

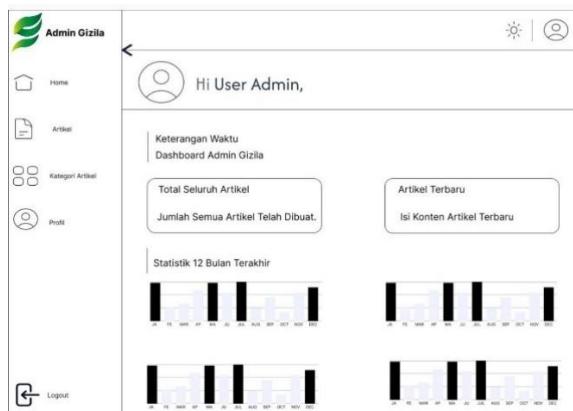
Perancangan halaman kelola *dashboard* admin merupakan bagian dari sistem yang hanya dapat diakses oleh admin Gizila. Halaman ini dirancang untuk memberikan kontrol penuh terkait pengelolaan data dan informasi didalam halaman *dashboard* admin Gizila, sehingga admin dapat mengelola *dashboard* admin Gizila dengan lebih efektif dan efisien. Halaman ini berfungsi untuk mengontrol dan memantau seluruh aktivitas sistem, termasuk pengelolaan artikel, pengelolaan hak akses *dashboard* admin, pengelolaan halaman profil admin, pengelolaan kategori artikel, serta grafik data dan informasi artikel.



Gambar 3.34 Desain Halaman Kelola *Dashboard* admin Gizila

### 3.3.5.4. Desain Halaman Kelola Fitur CRUD Artikel di *Dashboard*

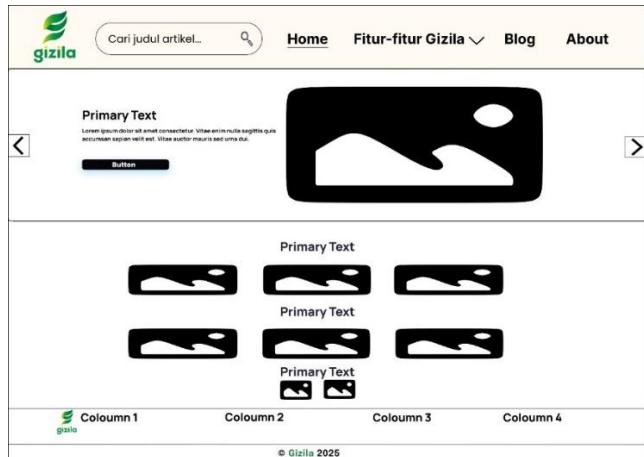
Perancangan halaman kelola artikel dirancang khusus untuk admin. Halaman ini memiliki peran penting untuk mendukung penyediaan data dan informasi, sehingga pengguna dapat mengakses artikel yang relevan sesuai dengan kebutuhan *user*. Dengan adanya fitur CRUD memastikan bahwa informasi yang diberikan kepada pengguna selalu terbaru dengan referensi data yang valid.



Gambar 3.35 Desain Halaman Kelola Fitur CRUD Artikel di *Dashboard*

### 3.3.5.5. Desain Halaman Kelola Website Gizila

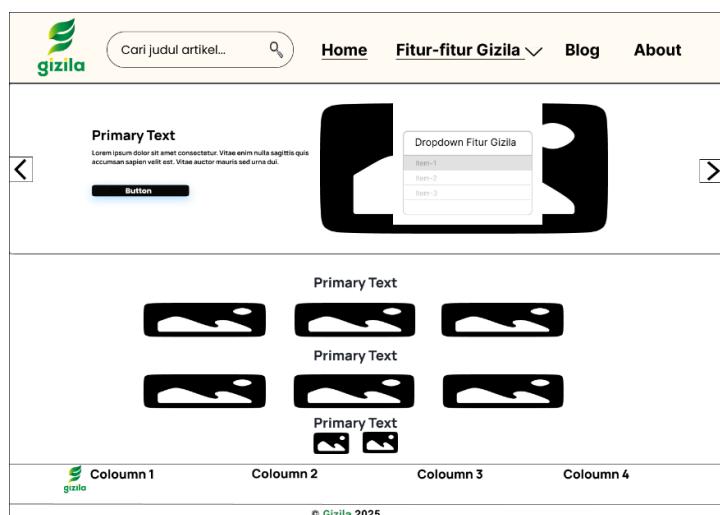
Perancangan halaman ini ditujukan khusus bagi admin sistem informasi Gizila dalam rangka mengelola konten dan tampilan website utama Gizila. Fitur ini penting untuk menjaga agar informasi dan layanan yang diberikan kepada pengguna tetap relevan, informatif, dan mudah diakses.



Gambar 3.36 Desain Halaman Kelola Website Gizila

### 3.3.5.6. Desain Halaman Akses Website Gizila

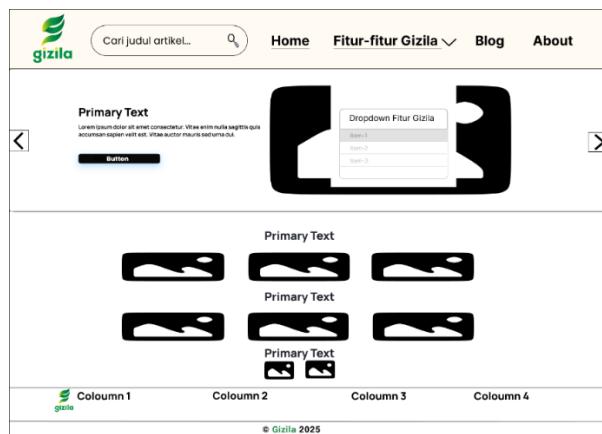
Perancangan halaman ini diperuntukkan bagi pengunjung umum (*user* atau publik) yang mengakses website Gizila tanpa perlu melakukan registrasi akun. Tujuannya adalah untuk memberikan kemudahan akses terhadap fitur-fitur utama Gizila secara cepat dan informatif, terutama yang berkaitan dengan kebutuhan gizi harian.



Gambar 3.37 Desain Halaman Akses Website Gizila

### 3.3.5.7. Desain Halaman Lihat Fitur-fitur Website Gizila

Perancangan halaman fitur-fitur Gizila dapat diakses secara langsung oleh pengguna umum (publik) tanpa harus melakukan *login* atau registrasi. Tujuannya untuk menyediakan akses cepat terhadap fitur utama Gizila yang berkaitan dengan edukasi dan perhitungan gizi harian.



Gambar 3.38 Desain Halaman Lihat Fitur-fitur Website Gizila

### 3.3.5.8. Desain Halaman Menghitung Gizi Harian

Perancangan halaman kalkulator Indeks Massa Tubuh (IMT) pada sistem informasi Gizila dapat diakses langsung oleh pengguna umum (publik) tanpa harus melakukan registrasi akun terlebih dahulu. Halaman ini berfungsi sebagai alat bantu interaktif bagi pengguna untuk menghitung status gizi berdasarkan perhitungan IMT. Fitur ini bertujuan untuk memberikan informasi awal mengenai kondisi tubuh pengguna, apakah termasuk kekurangan gizi, berat badan ideal, atau kelebihan berat badan. Perhitungan gizi harian pada sistem informasi Gizila menggunakan rumus *Harris-Benedict* yang mana sudah dijabarkan pada bab sebelumnya.

 A wireframe screenshot of a form titled 'Formulir Kalkulator Indeks Massa Tubuh (IMT)'. The form has two main sections: 'Berat (kg)' and 'Tinggi (cm)'. Each section contains an input field with a dropdown arrow. Below these are 'Usia (tahun)', 'Jenis Kelamin', and 'Tingkat Aktivitas:' sections. The 'Jenis Kelamin' section shows icons for 'Laki-laki' and 'Perempuan'. The 'Tingkat Aktivitas:' section shows four icons with corresponding activity levels: 'Sedikit atau tidak ada olahraga', 'Olahraga ringan 1-3x/minggu', 'Olahraga sedang 3-5x/minggu', and 'Latihan fisik berat 6-7x/minggu'. At the bottom, there's a 'Button' and a footer with four columns labeled 'Column 1', 'Column 2', 'Column 3', and 'Column 4'.

Gambar 3.39 Desain Halaman Fitur Hitung Gizi Harian 1

Perancangan halaman berdasarkan gambar 3.40 diatas ditujukan untuk memberikan kemudahan kepada pengguna umum dalam mengetahui status gizi berdasarkan Indeks Massa Tubuh (IMT). Halaman ini dapat diakses oleh publik secara langsung tanpa harus melakukan registrasi akun, sehingga memudahkan pengguna dalam memperoleh informasi dasar mengenai kondisi tubuh dari sisi gizi berdasarkan data pengguna pada *form* gizi harian.

**Gambar 3.40** Desain Halaman Fitur Hitung Gizi Harian 2

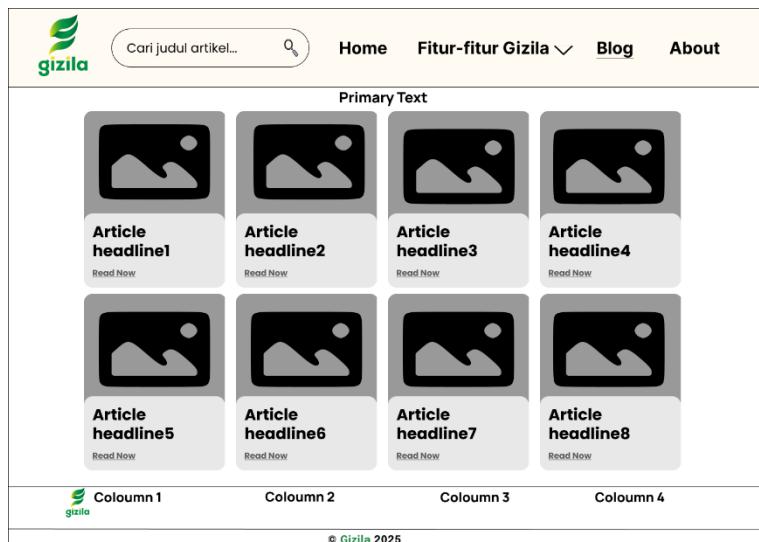
Perancangan halaman berdasarkan gambar 3.41 diatas merupakan salah satu bagian dari sistem informasi Gizila yang dapat diakses secara langsung tanpa harus melakukan *login* akun. Fitur ini dirancang untuk membantu pengguna dalam menghitung kebutuhan gizi harian secara cepat dan interaktif berbasis web. Pada halaman ini, pengguna dapat memilih jenis makanan atau minuman yang dikonsumsi pada waktu sarapan. Sistem menyediakan kolom pencarian menu makanan dan kolom input untuk jumlah porsi dalam gram sebagai data awal perhitungan. Selain itu, pengguna diberikan opsi untuk mengunggah gambar atau mengambil foto menu makanan yang dikonsumsi, guna memperjelas identifikasi menu yang dimaksud. Tombol "+Tambah Menu Lainnya" memungkinkan pengguna menambahkan lebih dari satu jenis makanan yang dikonsumsi pada waktu sarapan, makan siang dan makan malam.

**Gambar 3.41** Desain Halaman Fitur Hitung Gizi Harian 3

Perancangan halaman berdasarkan gambar 3.42 merupakan bagian dari proses validasi dalam sistem informasi Gizila yang berfungsi untuk memastikan bahwa data perhitungan gizi harian pengguna dapat dilakukan secara akurat dan personal, sehingga bagi pengguna yang mengakses fitur ini diarahkan untuk menghitung IMT terlebih dahulu yang ditampilkan dalam bentuk *popup* peringatan yang muncul ketika pengguna belum menghitung Indeks Massa Tubuh (IMT).

### 3.3.5.9. Desain Halaman Lihat Artikel Edukasi

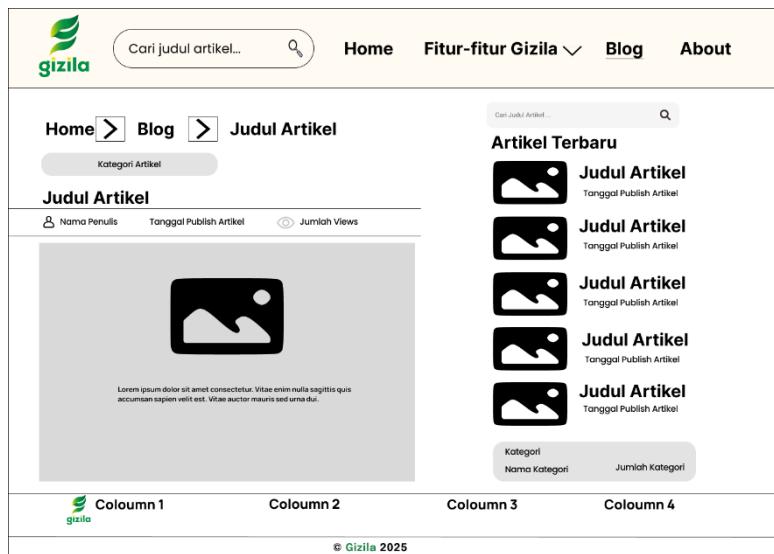
Perancangan halaman ini merupakan bagian dari fitur publik dalam sistem informasi Gizila yang dapat diakses secara bebas oleh seluruh pengunjung *website*. Halaman ini dirancang sebagai media informasi dan edukasi gizi bagi masyarakat umum. Desain halaman ini menampilkan daftar artikel edukasi gizi dalam bentuk artikel yang tersusun secara rapi dan responsif yang bisa diakses secara *real-time*.



Gambar 3.42 Desain Halaman Lihat Artikel Edukasi

### 3.3.5.10. Desain Halaman Mendapat Info Edukasi di *Website* Gizila

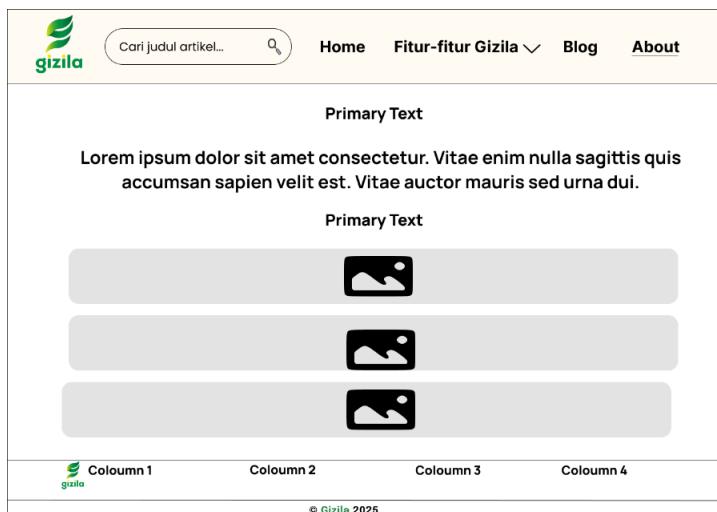
Perancangan halaman slug artikel ini merupakan fitur publik pada sistem informasi Gizila yang dirancang untuk menyajikan konten edukasi gizi secara terstruktur. Pada gambar 3.43 dibawah ini, artikel yang ditampilkan berjudul "Peran Serat dalam Menjaga Kesehatan Pencernaan" dan artikel termasuk dalam kategori edukasi gizi. Hal ini membantu pengguna untuk meningkatkan literasi gizi, sehingga pengguna dapat lebih sadar akan pentingnya pola makan sehat dan pengaruhnya terhadap kesehatan tubuh.



Gambar 3.43 Desain Halaman Mendapat Info Edukasi di *Website* Gizila

### 3.3.5.11. Desain Halaman *About*

Perancangan halaman *about* ini menjadi bagian dari fitur publik dalam sistem informasi Gizila yang dapat diakses dengan mudah oleh seluruh pengunjung *website*. Halaman ini dirancang untuk memberikan gambaran menyeluruh mengenai tujuan, visi, dan manfaat utama Gizila sebagai media informasi edukasi gizi. Melalui halaman ini, pengguna dapat memahami latar belakang pengembangan Gizila, serta peran pentingnya dalam mendukung pola hidup sehat masyarakat. Desain halaman menampilkan susunan informasi yang ringkas, jelas, dan responsif sehingga memudahkan pengunjung dalam memperoleh pemahaman secara *real-time* mengenai sistem informasi Gizila.



Gambar 3.44 Desain Halaman *About*

## **BAB IV**

### **IMPLEMENTASI DAN PENGUJIAN**

#### **4.1. Lingkungan Pengembangan**

Pengembangan proses rancang bangun sistem informasi Gizila untuk kebutuhan gizi harian ini memerlukan dukungan perangkat yang tepat sebagai alat bantu utama. Perangkat tersebut menjadi komponen utama yang memastikan kelancaran seluruh proses pengembangan. Kebutuhan perangkat dalam penelitian ini dikelompokkan menjadi dua jenis, yakni perangkat keras (*hardware*) dan perangkat lunak (*software*). Rincian spesifikasi kedua jenis perangkat yang digunakan dalam pembangunan sistem berbasis web Gizila yang akan dijabarkan pada tabel di bawah ini.

##### **4.1.1. Perangkat Keras**

Penggunaan perangkat keras dalam penelitian ini bertujuan untuk memfasilitasi kebutuhan sistem dalam hal input, pemrosesan, penyimpanan, dan penyajian data.

Adapun spesifikasi perangkat keras yang digunakan dalam perancangan sistem ini dirincikan pada tabel berikut ini.

**Tabel 4.1 Spesifikas Perangkat Keras yang Digunakan**

No.	Perangkat Keras	Keterangan
1.	Laptop	IdePad 3 14ALC6
2.	<i>Processor</i>	AMD Ryzen 3 5300U with Radeon Graphics
3.	<i>Memory RAM</i>	8.00 GB
4.	<i>Hard Disk Drive (HDD)</i>	500 GB

##### **4.1.2. Perangkat Lunak**

Dalam proses rancang bangun sistem informasi Gizila, serangkaian perangkat lunak (*software*) digunakan untuk mendukung siklus pengembangan. Perangkat lunak ini berfungsi sebagai alat utama untuk merancang arsitektur, mengembangkan fungsionalitas, mengimplementasikan kode, serta memastikan sistem dapat berjalan sesuai dengan tujuan.

Adapun spesifikasi perangkat lunak yang digunakan dalam perancangan sistem ini dirincikan pada tabel berikut ini.

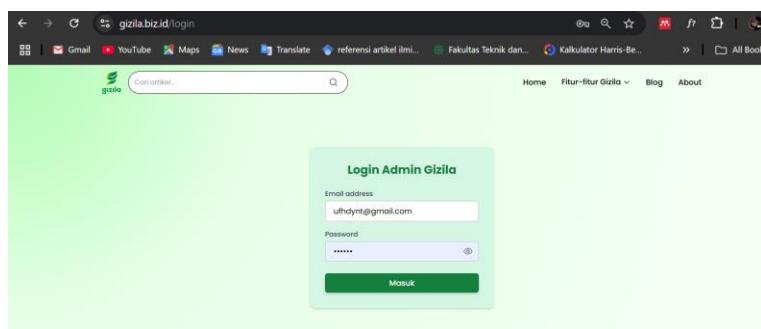
**Tabel 4.2** Spesifikasi Perangkat Lunak yang Digunakan

No.	Perangkat Lunak	Keterangan
1.	<i>Operation System</i>	<i>Windows 11 Home Single Language</i>
2.	<i>Browser</i>	<i>Google Chrome</i>
3.	<i>Text Editor</i>	<i>Visual Studio Code</i>
4.	<i>Framework</i>	Laravel versi 11.45.1
5.	<i>Server</i>	<i>XAMPP</i>
6.	<i>Database</i>	<i>MySQL</i> versi 10.4.32.
7.	Desain UML	<i>Draw.io</i>
8.	Desain UI/UX	Figma

## 4.2. Implementasi Antarmuka Pengguna

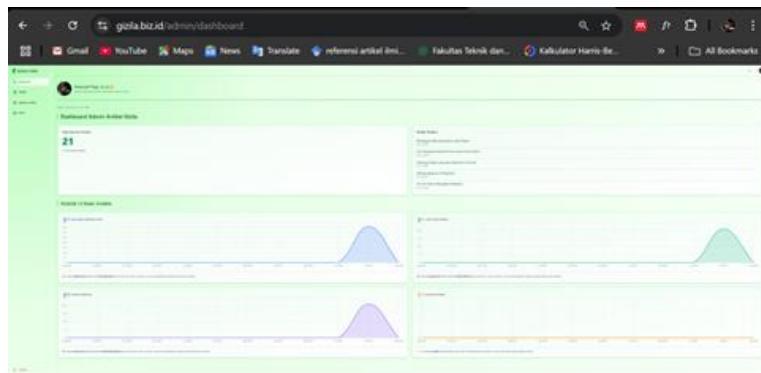
Proses penerapan atau pengembangan antarmuka pengguna dari sistem yang telah dirancang adalah bagian dari sistem yang langsung berinteraksi dengan pengguna akhir. Pada tahap ini, ditampilkan tampilan-tampilan utama dari aplikasi berbasis web yang telah dibuat, seperti halaman beranda, *form input*, tampilan hasil perhitungan, serta navigasi antar halaman. Implementasi ini mencerminkan hasil dari desain antarmuka yang telah dijelaskan pada bab sebelumnya dan memperlihatkan bagaimana sistem dirancang agar mudah digunakan, interaktif, dan responsif terhadap pengguna.

### 4.2.1. Implementasi Halaman *Login* Admin

**Gambar 4.1** Implementasi Halaman *Login* Admin

Halaman *login* admin berfungsi sebagai pintu masuk bagi admin sistem informasi gizila. Pada halaman ini, admin diminta untuk memasukkan alamat email dan kata sandi yang telah terdaftar sebelumnya ke dalam sistem. Tujuan dari halaman ini adalah untuk memberikan akses terbatas hanya kepada pengguna yang memiliki hak administratif agar dapat mengelola data dan informasi, dan komponen lain yang berkaitan dengan manajemen artikel Gizila.

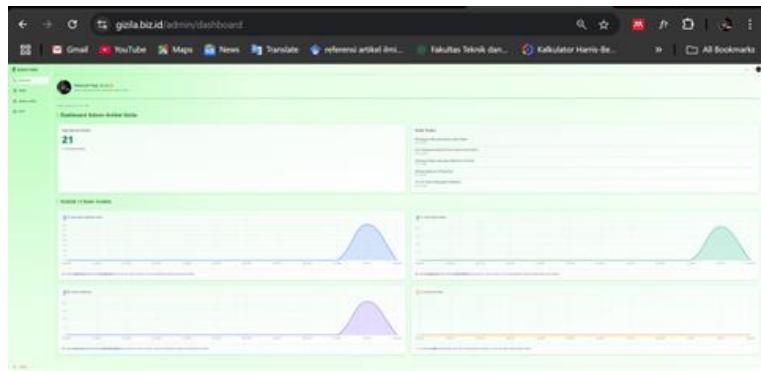
#### 4.2.2. Implementasi Halaman Akses *Dashboard* admin Gizila



**Gambar 4.2** Implementasi Halaman Akses *Dashboard* admin Gizila

Halaman *dashboard* admin pada sistem informasi Gizila berfungsi sebagai pusat kontrol dan *monitoring* utama bagi admin. Halaman ini hanya dapat diakses setelah admin berhasil melakukan proses *login* yang sah. Pada halaman ini, admin dapat melihat ringkasan data sistem secara *real-time*, termasuk jumlah artikel gizi yang telah ditambahkan, grafik statistik interaksi, data dan informasi, serta daftar artikel terbaru yang telah dipublikasikan. Halaman ini memiliki peran strategis karena menjadi titik awal bagi admin untuk mengelola keseluruhan informasi gizi yang ditampilkan kepada publik.

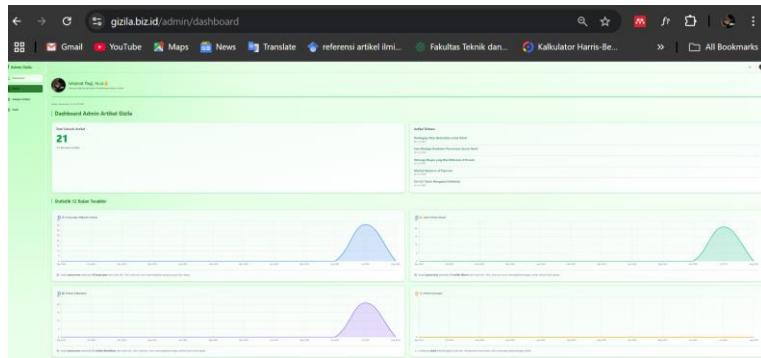
#### 4.2.3. Implementasi Halaman Kelola *Dashboard* admin Gizila



**Gambar 4.3** Implementasi Halaman Kelola *Dashboard* admin Gizila

Halaman *dashboard* pada sistem informasi Gizila tidak hanya berfungsi sebagai pusat informasi, tetapi juga menjadi sarana bagi admin dalam mengelola data dan memantau aktivitas sistem. Halaman ini memungkinkan admin untuk mengakses, memantau, mengelola data dan informasi artikel, statistik pengguna, serta performa sistem secara keseluruhan.

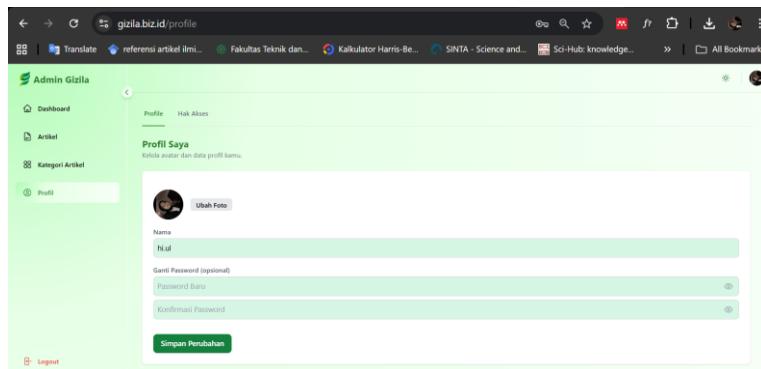
#### 4.2.4. Implementasi Halaman Kelola Fitur CRUD Artikel di Dashboard



**Gambar 4.4** Implementasi Halaman Kelola Fitur CRUD Artikel di *Dashboard*

Halaman kelola artikel pada *dashboard* admin Gizila dirancang untuk memberikan kemudahan bagi admin dalam mengelola konten informasi gizi harian. Fitur ini mencakup proses CRUD (*Create, Read, Update, Delete*) terhadap artikel yang berkaitan dengan topik gizi, sesuai dengan tujuan utama sistem informasi Gizila sebagai media penyedia informasi gizi yang akurat dan bermanfaat. Dengan adanya fitur CRUD artikel, admin memiliki kendali penuh terhadap isi informasi yang disajikan kepada pengguna. Hal ini penting untuk mendukung peran Gizila sebagai sistem informasi gizi harian berbasis web yang terpercaya dan selalu diperbarui.

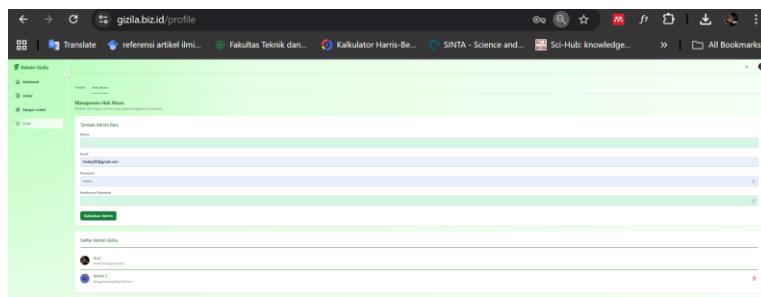
#### 4.2.5. Implementasi Halaman Kelola Profil di Dashboard



**Gambar 4.5** Implementasi Halaman Kelola Profil di *Dashboard*

Halaman profil admin Gizila dirancang untuk memudahkan admin dalam mengelola data profil. Melalui halaman ini, admin dapat melakukan pengelolaan data, seperti mengubah nama, mengganti foto profil, serta memperbarui kata sandi. Fitur ini memberikan fleksibilitas bagi admin untuk menyesuaikan data pribadinya tanpa perlu melakukan konfigurasi manual di *database*.

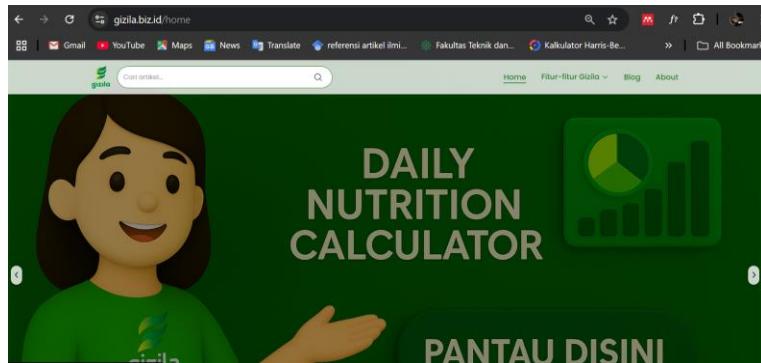
#### 4.2.6. Implementasi Halaman Kelola Hak Akses di *Dashboard*



**Gambar 4.6** Implementasi Halaman Kelola Hak Akses di *Dashboard*

Halaman hak akses admin Gizila dirancang untuk memberikan kendali penuh kepada admin utama dalam mengelola peran dan otorisasi pengguna di *dashboard*. Melalui halaman ini, admin dapat menambahkan akun admin baru dengan mengisi nama, *email*, kata sandi. Selain itu, sistem juga menampilkan daftar admin yang sudah terdaftar, sehingga memudahkan pengelolaan untuk melihat siapa saja yang terdaftar memiliki hak akses sebagai admin. Dengan adanya fitur ini, manajemen hak akses menjadi lebih terorganisir didalam sistem.

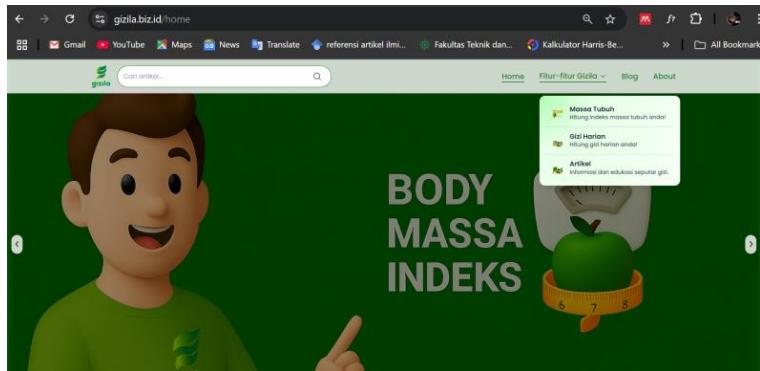
#### 4.2.7. Implementasi Halaman Kelola Website Gizila



**Gambar 4.7** Implementasi Halaman Kelola Website Gizila

Halaman utama *website* Gizila merupakan halaman yang ditujukan bagi publik untuk mengakses informasi terkait kebutuhan gizi harian secara interaktif dan informatif. Dibalik tampilannya, admin memiliki peran penting dalam mengelola tampilan konten dan fitur yang tersedia di halaman *website* ini. Melalui implementasi halaman ini, admin memiliki kontrol terhadap konten dan tampilan yang disajikan kepada pengguna, sehingga informasi gizi yang disampaikan dapat terus diperbarui dan disesuaikan dengan kebutuhan masyarakat.

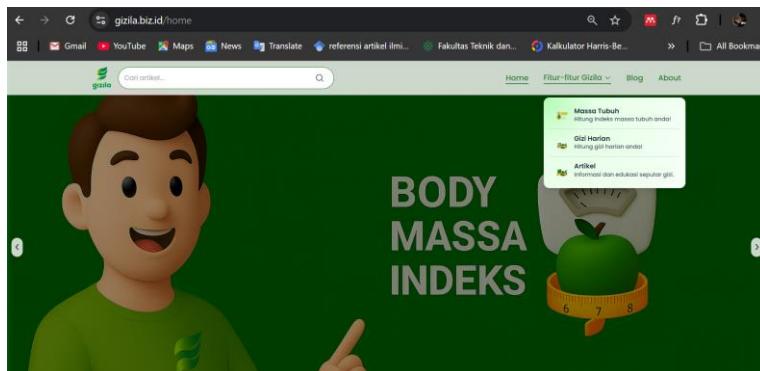
#### 4.2.8. Implementasi Halaman Akses Website Gizila



**Gambar 4.8** Implementasi Halaman Akses Website Gizila

Halaman akses *website* Gizila dirancang untuk memberikan kemudahan bagi publik dalam memanfaatkan fitur-fitur utama sistem informasi gizi. Melalui implementasi halaman ini, Gizila bertujuan untuk mendukung masyarakat dalam mengakses informasi gizi secara cepat dan akurat tanpa perlu mendaftar atau login terlebih dahulu. Dengan demikian, sistem ini tidak hanya berfungsi sebagai alat kalkulasi gizi, namun juga sebagai media edukasi publik terkait pentingnya pemenuhan gizi harian.

#### 4.2.9. Implementasi Halaman Lihat Fitur-fitur Website Gizila



**Gambar 4.9** Implementasi Halaman Fitur-fitur Website Gizila

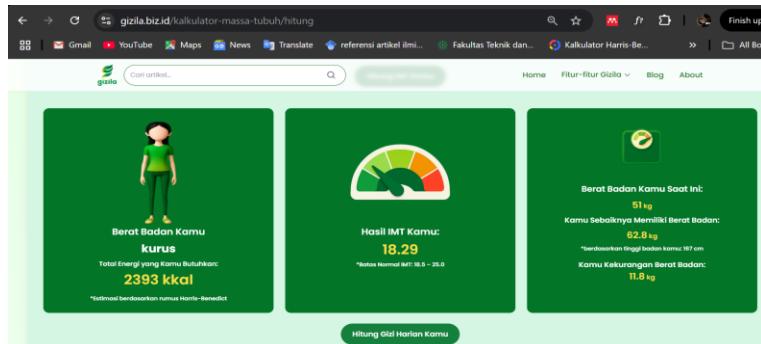
Halaman fitur-fitur pada *website* Gizila merupakan salah satu antarmuka yang dirancang untuk memudahkan publik dalam mengakses layanan utama yang tersedia. Tampilan halaman ini juga dirancang responsif dan mudah digunakan. Pengguna tidak perlu melakukan *login* untuk mengakses fitur-fitur tersebut, sehingga dapat langsung memanfaatkan sistem tanpa hambatan. Hal ini mendukung tujuan utama dari sistem informasi Gizila, yaitu memberikan edukasi dan layanan gizi secara terbuka dan mudah diakses oleh masyarakat luas.

#### 4.2.10. Implementasi Halaman Menghitung Gizi Harian

The screenshot shows a web-based calculator for calculating Body Mass Index (IMT). The interface is clean with a light green header. At the top, there are input fields for 'Berat (kg)', 'Tinggi (cm)', 'Usia (tahun)', and 'Jenis Kelamin' (with options for Laki-laki and Perempuan). Below these are four activity level icons: 'Kurus IMT' (underweight), 'Normal IMT' (normal), 'Overweight IMT' (overweight), and 'Obesitas IMT' (obesity). Each icon has a small explanatory text below it. A large green button at the bottom right says 'Hitung IMT Kamu'.

**Gambar 4.10** Implementasi Halaman Fitur Hitung Gizi Harian 1

Halaman fitur hitung gizi harian 1 merupakan bagian awal dari proses penghitungan kebutuhan gizi harian yang dirancang untuk pengguna pada sistem informasi Gizila. Pengguna diarahkan untuk mengisi form indeks massa tubuh terlebih dahulu yang berisi *form input* data individu yang relevan sebagai dasar perhitungan kebutuhan gizi, seperti: berat badan, tinggi badan, usia, jenis kelamin, tingkat aktivitas fisik.



**Gambar 4.11** Implementasi Halaman Fitur Hitung Gizi Harian 2

Halaman fitur hitung gizi harian 2 merupakan kelanjutan dari proses kalkulasi kebutuhan gizi berdasarkan data yang telah diinput oleh pengguna pada tahap sebelumnya. Sistem menampilkan hasil analisis berupa informasi status gizi dan kebutuhan energi harian pengguna, yang berupa:

- Status Berat Badan dan Kebutuhan Energi

Sistem menampilkan status berat badan pengguna misalnya "kurus" berdasarkan perhitungan Indeks Massa Tubuh (IMT). Di bawahnya, tercantum estimasi kebutuhan energi harian dalam satuan kilokalori (kkal), yang dihitung menggunakan rumus *Harris-Benedict*, disesuaikan dengan data jenis kelamin, usia, tinggi badan, berat badan, dan tingkat aktivitas fisik.

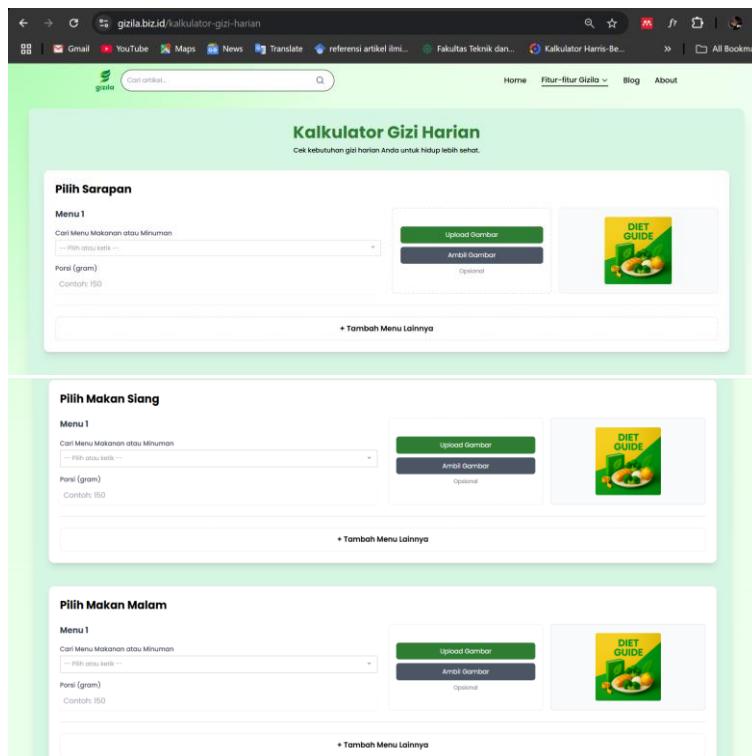
b. Hasil IMT

Sistem menampilkan hasil perhitungan IMT pengguna secara numerik, disertai dengan indikator visual berupa skala warna untuk menunjukkan apakah IMT tersebut berada dalam rentang normal (18.5 – 25.0) atau tidak.

c. Rekomendasi Berat Badan Ideal

Sistem memberikan informasi untuk membantu pengguna memahami sejauh mana kondisi berat badannya dari standar ideal, serta menjadi acuan untuk perencanaan kebutuhan gizi yang lebih tepat.

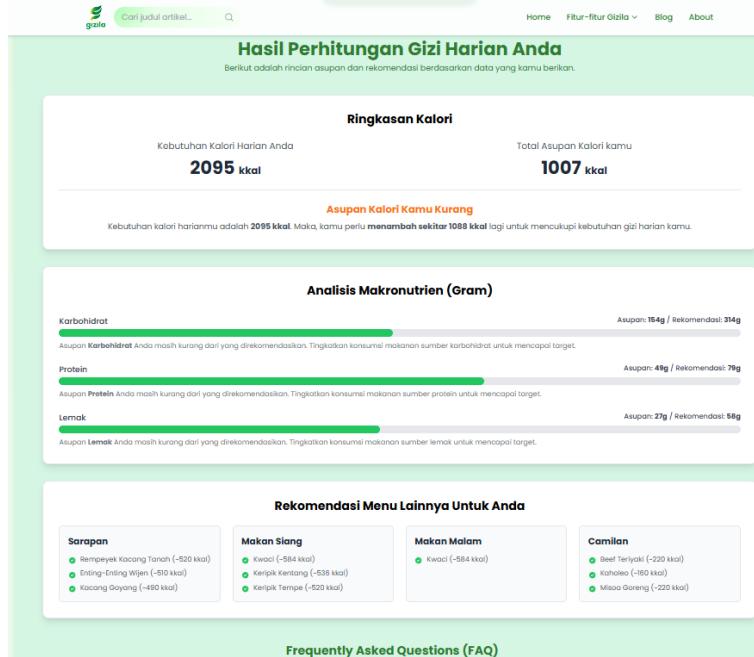
Setelah memperoleh hasil ini, pengguna dapat melanjutkan ke tahap berikutnya dengan mengklik tombol “Hitung Gizi Harian Kamu”, yang akan menampilkan perhitungan kebutuhan gizi makro (karbohidrat, protein, lemak) sesuai estimasi jumlah energi yang telah dihitung.



**Gambar 4.12** Implementasi Halaman Fitur Hitung Gizi Harian 3

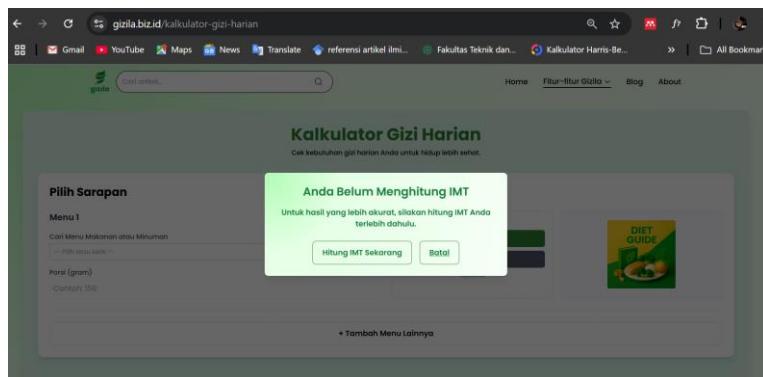
Halaman fitur hitung gizi harian 3 berfungsi untuk membantu pengguna dalam mencatat dan menghitung asupan makanan atau minuman yang dikonsumsi dalam sehari. Pengguna diminta untuk memilih menu makanan atau minuman berdasarkan waktu makan, misalnya waktu sarapan, makan siang, dan makan malam.

Halaman ini memungkinkan pengguna melakukan pelacakan konsumsi makanan secara personal dan harian. Dengan mengisi data ini secara lengkap, sistem nantinya dapat menghitung total asupan kalori serta makronutrien (karbohidrat, protein, dan lemak).



Gambar 4.13 Implementasi Halaman Fitur Hitung Gizi Harian 4

Halaman fitur hitung gizi harian 4 merupakan tampilan akhir yang menampilkan hasil perhitungan gizi harian berdasarkan input data pengguna. Fitur ini dirancang untuk dapat diakses oleh semua pengguna publik.

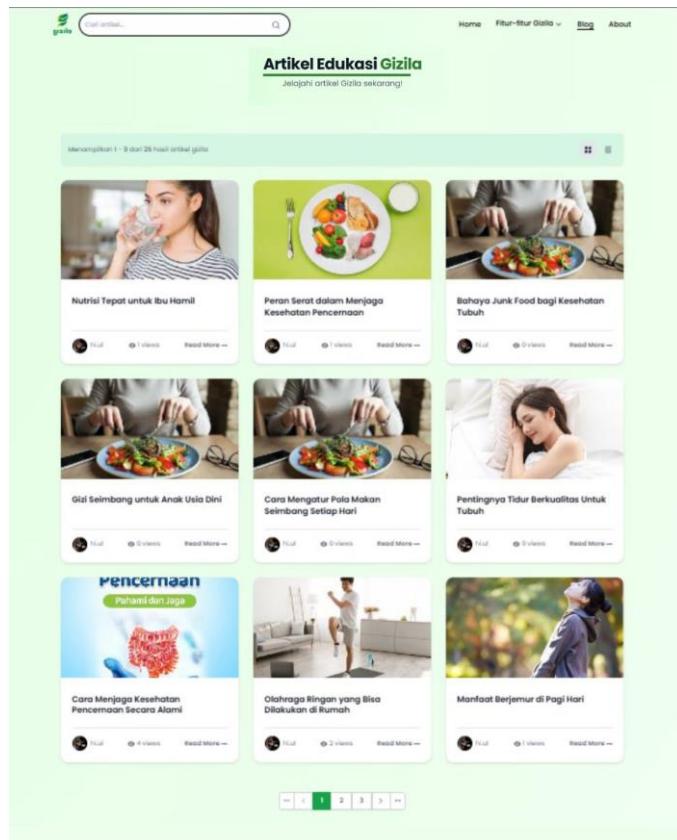


Gambar 4.14 Implementasi Halaman Fitur Hitung Gizi Harian 5

Halaman fitur hitung gizi harian 5 menampilkan sebuah *popup* peringatan interaktif yang muncul ketika pengguna mencoba menggunakan fitur perhitungan gizi harian namun

belum menghitung Indeks Massa Tubuh (IMT). *Popup* ini merupakan bagian dari validasi sistem untuk memastikan akurasi hasil perhitungan gizi.

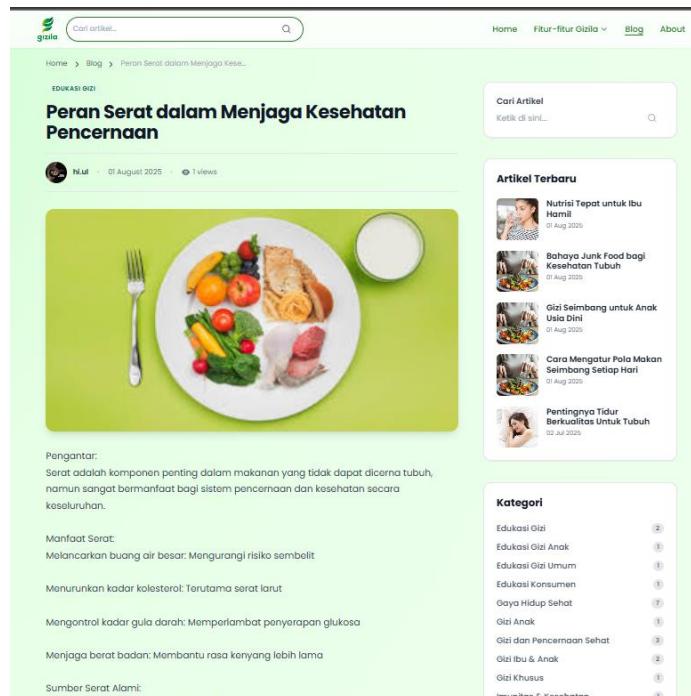
#### 4.2.11. Implementasi Halaman Lihat Artikel Edukasi



**Gambar 4.15** Implementasi Halaman Lihat Artikel Edukasi

Halaman lihat artikel edukasi merupakan salah satu fitur publik pada sistem informasi Gizila yang dirancang untuk menyajikan informasi dan pengetahuan seputar gizi kepada seluruh pengunjung *website*. Halaman ini dapat diakses tanpa perlu *login*, sehingga sangat cocok sebagai media edukasi yang terbuka bagi masyarakat luas.

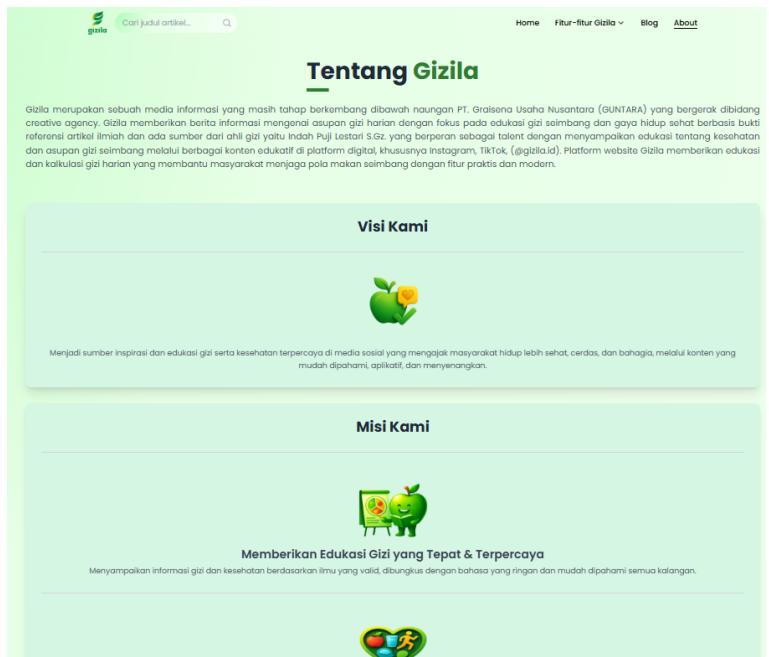
#### 4.2.12. Implementasi Halaman Mendapat Info Edukasi di Website Gizila



**Gambar 4.16** Implementasi Halaman Mendapat Info Edukasi di *Website Gizila*

Halaman mendapat info edukasi merupakan fitur publik dari sistem informasi Gizila yang dirancang sebagai pusat informasi. Halaman ini menyajikan artikel secara lengkap dan detail artikel (slug) kepada pengguna umum tanpa harus melakukan *login* terlebih dahulu. Tujuannya adalah agar edukasi gizi dapat dijangkau oleh lebih banyak masyarakat sebagai bagian dari fungsi literasi digital yang inklusif.

#### 4.2.13. Implementasi Halaman *About*



**Gambar 4.17** Halaman *About*

Halaman *About* berfungsi untuk memberikan informasi mengenai konteks dan tujuan sistem informasi Gizila. Pada halaman ini, ditampilkan visi misi dari Gizila juga informasi kontak yang dapat dihubungi oleh pengguna jika membutuhkan bantuan atau ingin mengetahui lebih lanjut mengenai Gizila dan juga alamat lengkap berupa *Google Maps* yang diintegrasikan untuk menunjukkan lokasi kantor secara interaktif.

#### 4.3. Implementasi Sistem Informasi Antarmuka Admin

##### 4.3.1. Pengujian Halaman *Login* Admin

**Tabel 4.3** Pengujian Halaman *Login*

Aspek yang diuji	Hasil yang diinginkan	Hasil Aktual / Keluaran Sistem
Admin memasukkan <i>email</i> dan <i>password</i> yang benar	Sistem mengarahkan ke halaman <i>dashboard</i> admin	Sistem berhasil menampilkan halaman <i>dashboard</i> admin
Admin memasukkan <i>email</i> atau <i>password</i> yang salah	Sistem menampilkan pesan kesalahan “ <i>email</i> atau <i>password</i> salah”	Sistem menampilkan notifikasi error sesuai yang diharapkan

#### 4.3.2. Pengujian Halaman *Dashboard Artikel*

Tabel 4.4 Pengujian Halaman *Dashboard Artikel*

Aspek yang diuji	Hasil yang diinginkan	Hasil Aktual / Keluaran Sistem
Admin berhasil login ke sistem dan membuka halaman <code>/admin/blog</code>	Sistem menampilkan halaman <i>dashboard</i> artikel berisi daftar artikel yang telah ditambahkan	Sistem berhasil menampilkan halaman <i>dashboard</i> artikel sesuai data pada <i>database</i>
Admin mengklik tombol + Tambah Artikel	Sistem menampilkan <i>form</i> untuk menambahkan artikel baru	Sistem berhasil menampilkan <i>form</i> tambah artikel
Admin mengisi <i>form</i> artikel lalu menekan tombol Simpan	Sistem menyimpan artikel dan menampilkannya di daftar artikel	Sistem berhasil menyimpan dan menampilkan artikel di daftar manajemen artikel
Admin menggunakan fitur pencarian artikel di kolom "Cari artikel..."	Sistem menampilkan artikel sesuai kata kunci yang diketik	Sistem berhasil menampilkan hasil pencarian artikel yang relevan
Admin mengklik ikon edit pada salah satu artikel	Sistem menampilkan <i>form</i> edit artikel	Sistem berhasil menampilkan data artikel dalam <i>form</i> edit
Admin mengklik ikon hapus pada salah satu artikel	Sistem menampilkan konfirmasi penghapusan lalu menghapus artikel	Sistem berhasil menghapus artikel setelah konfirmasi
Admin mengisi <i>form</i> tambah artikel tapi ada data yang kosong misal: judul kosong lalu klik Simpan	Sistem menampilkan pesan validasi bahwa kolom tidak boleh kosong	Sistem menampilkan pesan validasi sesuai harapan
Admin melakukan pencarian dengan kata kunci yang tidak sesuai atau kosong	Sistem menampilkan informasi "Artikel dengan judul .. tidak ditemukan"	Sistem berhasil menampilkan pesan "Artikel dengan judul .. tidak ditemukan" atau hasil kosong

#### 4.3.3. Pengujian Halaman *Dashboard* Kategori Artikel

**Tabel 4.5** Pengujian Halaman *Dashboard* Kategori Artikel

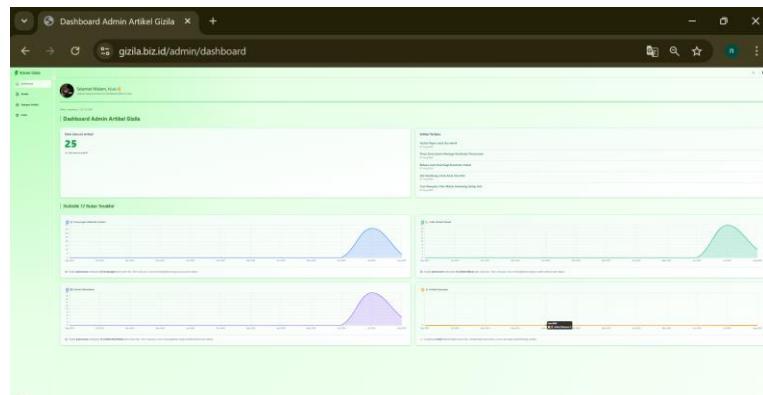
Aspek yang diuji	Hasil yang diinginkan	Hasil Aktual / Keluaran Sistem
Admin memasukkan <i>email</i> dan <i>password</i> yang benar	Sistem mengarahkan ke halaman <i>dashboard</i> admin	Sistem berhasil menampilkan halaman <i>dashboard</i> admin
Admin berhasil <i>login</i> dan memilih menu Kategori Artikel di <i>sidebar dashboard</i>	Sistem menampilkan halaman Manajemen Kategori yang berisi daftar kategori, tombol tambah kategori, serta jumlah artikel tiap kategori	Sistem berhasil menampilkan halaman Manajemen Kategori sesuai harapan
Admin mengklik tombol Tambah Kategori Baru	Sistem menampilkan <i>form</i> untuk input kategori baru	Sistem berhasil menampilkan <i>form</i> tambah kategori
Admin mengklik ikon edit pada salah satu kategori	Sistem menampilkan <i>form</i> edit kategori sesuai data kategori tersebut	Sistem berhasil menampilkan <i>form</i> edit kategori
Admin mengklik ikon hapus pada salah satu kategori	Sistem meminta konfirmasi penghapusan lalu menghapus kategori	Sistem berhasil menghapus kategori setelah konfirmasi
Admin mencoba menambahkan kategori tanpa mengisi nama kategori	Sistem menampilkan pesan validasi bahwa kolom tidak boleh kosong	Sistem menampilkan pesan validasi sesuai harapan
Admin menginput nama kategori yang sudah ada	Sistem menampilkan pesan error "Nama kategori sudah digunakan"	Sistem menampilkan notifikasi error sesuai yang diharapkan

#### 4.3.4. Pengujian Halaman *Dashboard Profil*

**Tabel 4.6** Pengujian Halaman *Dashboard Profil*

Skenario Pengujian	Hasil yang diinginkan	Hasil Aktual / Keluaran Sistem
Admin masuk ke halaman profil dan mengubah data profil (nama, foto profil atau <i>password</i> ) lalu klik “Simpan Perubahan”	Sistem menampilkan pesan bahwa data profil berhasil diperbarui	Sistem menampilkan halaman profil yang telah diperbarui sesuai data yang dimasukkan
Admin masuk ke tab ”Hak Akses”, mengisi <i>form</i> nama, <i>email</i> , <i>password</i> dan klik ”Daftarkan Admin”	Sistem menambahkan admin baru ke dalam daftar admin Gizila	Sistem berhasil menambahkan data admin baru dan menampilkannya dalam daftar admin
Admin mengisi <i>form</i> ubah profil dengan kolom nama kosong lalu klik ”Simpan Perubahan”	Sistem menampilkan pesan error bahwa nama tidak boleh kosong	Sistem menolak penyimpanan data dan menampilkan notifikasi kesalahan
Admin mengisi <i>password</i> baru tetapi tidak mengisi konfirmasi <i>password</i>	Sistem menampilkan pesan error bahwa konfirmasi <i>password</i> wajib diisi	Sistem gagal menyimpan perubahan dan menampilkan notifikasi sesuai validasi
Admin mengisi <i>form</i> tambah admin baru dengan <i>password</i> dan konfirmasi <i>password</i> yang tidak sama	Sistem menampilkan pesan error bahwa <i>password</i> tidak cocok	Sistem tidak menambahkan admin baru dan memberikan notifikasi kesalahan

#### 4.3.5. Halaman *Dashboard* Admin Gizila



**Gambar 4.18** Halaman *Home Dashboard* Admin Gizila

Pada halaman *home dashboard* admin yang digunakan sebagai pusat kontrol utama bagi admin dalam mengelola artikel Gizila. Melalui halaman ini, admin dapat memantau total jumlah artikel yang telah dipublikasikan, melihat artikel-artikel terbaru, serta mengakses statistik performa sistem informasi Gizila dalam 12 bulan terakhir. Seluruh elemen pada halaman ini dirancang untuk memudahkan admin dalam melakukan manajemen data dan informasi secara *real-time*.

#### 4.3.6. Halaman Manajemen Artikel

The screenshot shows the 'Manajemen Artikel' page. It features a table with the following columns: NAMA, KATEGORI, PENULIS, STATUS, TANGGAL, VIEWS, and AKSI. The table lists 12 articles, each with a small thumbnail image, the article title, the category (e.g., Gizi Ibu & Anak, Gizi Sehat, Gizi dan Perencanaan Sehat), the author (e.g., Gita Triputri Sahat, Gita dan Perencanaan Sehat), the status (e.g., Published, Published), the date (e.g., 01 Aug 2023, 01 Jul 2023), the views (e.g., 1, 2, 1, 0, 0, 9, 4, 2, 1), and a set of actions (edit, delete).

**Gambar 4.19** Halaman *Dashboard* Manajemen Artikel

Pada halaman ini dilakukan proses penambahan, pengelolaan, dan pemantauan artikel oleh admin. Halaman ini merupakan pusat manajemen konten artikel yang ditampilkan pada sistem, dan hanya dapat diakses oleh pengguna yang memiliki hak akses sebagai admin berdasarkan data yang ada di *database*.

#### 4.3.7. Halaman Kategori Artikel

Manajemen Kategori			
NAMA KATEGORI	SLUG	JUMLAH ARTIKEL	Aksi
Gizi Anak	gizi-anak	1 Artikel	
Edukasi Gizi	edukasi-gizi	2 Artikel	
Gizi Khusus	gizi-khusus	1 Artikel	
Edukasi Konsumen	edukasi-konsumen	1 Artikel	
Pola Makan & Psikologi	pola-makan-psikologi	1 Artikel	
Edukasi Gizi Umum	edukasi-gizi-umum	1 Artikel	
Imunitas & Kesehatan	imunitas-kesehatan	1 Artikel	
Gizi Ibu & Anak	gizi-ibu-anak	2 Artikel	
Penyuluhan Gizi	penyuluhan-gizi	1 Artikel	
Edukasi Gizi Anak	edukasi-gizi-anak	1 Artikel	

**Gambar 4.20** Halaman Dashboard Manajemen Kategori Artikel

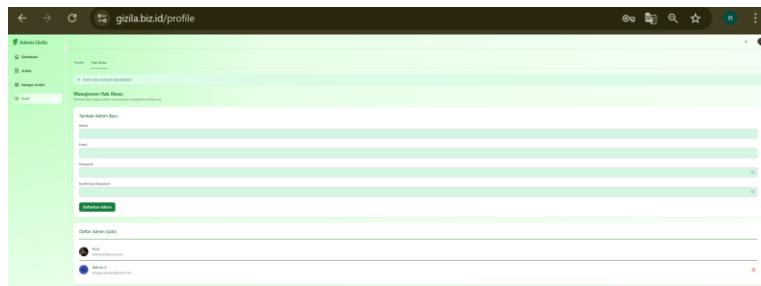
Pada halaman ini, admin dapat mengelola daftar kategori artikel yang tersedia di sistem. Admin dapat menambahkan kategori baru dengan menekan tombol Tambah Kategori Baru, mengubah kategori yang sudah ada melalui ikon edit atau menghapus kategori dengan ikon tempat sampah (*delete*). Halaman ini juga menampilkan informasi seperti nama kategori, slug, dan jumlah artikel yang terkait dengan masing-masing kategori.

#### 4.3.8. Halaman Profil Admin

**Gambar 4.21** Halaman Dashboard Profil Admin

Pada halaman ini, admin dapat mengelola informasi profil, termasuk mengganti foto profil dan memperbarui data seperti nama. Selain itu, admin juga memiliki opsi untuk mengubah kata sandi dengan mengisi kolom "Password Baru" dan "Konfirmasi Password". Setelah melakukan perubahan, admin dapat menyimpannya dengan menekan tombol Simpan Perubahan.

#### 4.3.9. Halaman Hak Akses



**Gambar 4.22** Halaman *Dashboard* Hak Akses Admin

Pada halaman ini, admin dapat mengelola hak akses pengguna yang memiliki peran sebagai admin di sistem. Admin dapat menambahkan admin baru dengan mengisi *form* yang berisi nama, email, kata sandi, dan konfirmasi kata sandi, lalu menekan tombol "Daftarkan Admin". Selain itu, halaman ini juga menampilkan daftar *user* akun admin *dashboard* yang sudah terdaftar.

#### 4.4. *Black Box Testing*

*Black box testing* merupakan metode pengujian perangkat lunak yang berfokus pada fungsi sistem tanpa memeriksa struktur internalnya. Dalam pengembangan sistem informasi GIZILA berbasis web, pengujian ini bertujuan untuk memastikan bahwa setiap fitur berjalan sesuai dengan kebutuhan pengguna. Melalui pendekatan ini, kesalahan atau ketidaksesuaian fungsionalitas dapat diidentifikasi dan diperbaiki agar sistem dapat bekerja sebagaimana mestinya. Hasil dari pengujian *black box* terhadap sistem GIZILA ditampilkan pada tabel berikut.

##### 4.4.1. Pengujian *Black Box* - Login Admin

**Tabel 4.7** *Black Box Testing* Login Admin

Skenario Uji		<i>Login Admin</i>	
HTTP Method		<u><a href="http://127.0.0.1:8000/login">http://127.0.0.1:8000/login</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin mengakses halaman <i>login</i> kemudian input <i>email</i> dan <i>password</i> yang valid	Sistem menampilkan halaman <i>dashboard</i> admin	Sesuai

2.	Admin mengakses halaman <i>login</i> kemudian input <i>email</i> dan <i>password</i> yang salah	Sistem menampilkan pesan error “ <i>email</i> atau <i>password</i> salah”	Sesuai
----	---	---	--------

#### 4.4.2. Pengujian *Black Box* - Kelola Dashboard admin Gizila

**Tabel 4.8** *Black Box Testing* Kelola Dashboard admin Gizila

<b>Skenario Uji</b>		<b>Kelola Dashboard</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/admin/dashboard">http://127.0.0.1:8000/admin/dashboard</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1.	Admin berhasil <i>login</i> dan mengakses halaman <i>dashboard</i>	Sistem menampilkan halaman <i>dashboard</i> admin	Sesuai
2.	User yang belum <i>login</i> mencoba mengakses halaman <i>dashboard</i>	Sistem mengarahkan ke halaman <i>login</i>	Sesuai

#### 4.4.3. Pengujian *Black Box* - Kelola Manajemen Artikel

**Tabel 4.9** *Black Box Testing* Kelola Manajemen Artikel

<b>Skenario Uji</b>		<b>Tambah Artikel</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/admin/blog/create">http://127.0.0.1:8000/admin/blog/create</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1.	Admin klik tombol tambah artikel kemudian mengisi <i>form</i> tambah artikel	Sistem menampilkan halaman <i>form</i> tambah artikel baru dan menyimpan data	Sesuai
2.	Admin mengisi <i>form</i> tambah artikel, kemudian ada salah satu kolom yang tidak diisi	Sistem menampilkan pesan validasi error pada kolom yang belum diisi	Sesuai
<b>Skenario Uji</b>		<b>Cari Judul Artikel</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/admin/blog?search=joging">http://127.0.0.1:8000/admin/blog?search=joging</a></u>	

No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin input kata kunci pada kolom pencarian	Sistem menampilkan artikel dengan judul mengandung kata "joging"	Sesuai
2.	Admin input kata kunci tidak ditemukan	Sistem menampilkan pesan atau hasil kosong	Sesuai
<b>Skenario Uji</b>		<b>Edit Artikel</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/admin/blog/pentingnya-tidur-berkualitas-untuk-tubuh(slug)/edit">http://127.0.0.1:8000/admin/blog/pentingnya-tidur-berkualitas-untuk-tubuh(slug)/edit</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin klik tombol edit, kemudian memperbarui mengisi form edit	Sistem menampilkan <i>form</i> edit dengan data artikel yang dipilih dan menyimpan data	Sesuai
2.	Admin klik tombol edit, kemudian ada salah satu kolom yang dikosongkan	Sistem menampilkan pesan validasi error pada kolom yang belum diisi	Sesuai
<b>Skenario Uji</b>		<b>Hapus Artikel</b>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1	Admin klik tombol hapus	Sistem menampilkan pesan konfirmasi hapus dan tekan ya kemudian data berhasil terhapus	Sesuai

#### 4.4.4. Pengujian *Black Box* – Halaman Manajemen Kategori

**Tabel 4.10** *Black Box Testing* Halaman Manajemen Kategori

<b>Skenario Uji</b>		<b>Tambah Kategori</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/admin/categories">http://127.0.0.1:8000/admin/categories</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin klik tombol tambah kategori, kemudian mengisi <i>form</i>	Sistem menampilkan halaman <i>form</i> tambah kategori baru dan menyimpan data	Sesuai

2.	Admin tidak mengisi nama kategori lalu klik simpan	Sistem menampilkan pesan validasi bahwa kolom nama kategori wajib diisi	Sesuai
<b>Skenario Uji</b>		<b>Edit Kategori</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/admin/categories/9(id_kategori)/edit">http://127.0.0.1:8000/admin/categories/9(id_kategori)/edit</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin klik tombol edit, kemudian memperbarui mengisi <i>form</i> edit	Sistem menampilkan <i>form</i> edit dengan kategori artikel yang dipilih dan menyimpan data	Sesuai
2.	Admin klik tombol edit, kemudian ada salah satu kolom yang dikosongkan	Sistem menampilkan pesan validasi error pada kolom yang belum diisi	Sesuai
<b>Skenario Uji</b>		<b>Hapus Kategori</b>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin klik tombol hapus	Sistem menampilkan pesan konfirmasi hapus dan tekan ya kemudian data berhasil terhapus	Sesuai

#### 4.4.5. Pengujian Black Box – Halaman Logout

**Tabel 4.11** Black Box Testing Halaman Logout

<b>Skenario Uji</b>		<b>Logout</b>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	Admin klik tombol <i>logout</i>	Sistem menutup sesi dan mengarahkan pengguna ke halaman <i>home website</i> Gizila	Sesuai

#### 4.4.6. Pengujian *Black Box* – Halaman Profil

**Tabel 4.12 Black Box Testing Halaman Profil**

<b>Skenario Uji</b>		<b>Halaman Profile</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/profile">http://127.0.0.1:8000/profile</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1.	Admin mengakses halaman <i>profile</i>	Sistem menampilkan halaman <i>profile</i> bagian tab <i>profile</i> berupa <i>form</i> input nama dan <i>password</i> untuk mengelola perubahan data <i>user admin</i>	Sesuai
2	Admin mengakses halaman <i>profile</i> pada tab <i>profile</i> kemudian ingin mengubah data akun <i>user admin</i> dan input <i>password</i> yang salah	Sistem menampilkan pesan error	Sesuai
<b>Skenario Uji</b>		<b>Halaman Profil Bagian Tab Hak Akses</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/profile">http://127.0.0.1:8000/profile</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1	Admin mengakses halaman <i>profile</i> pada tab hak akses	Sistem menampilkan halaman <i>profile</i> bagian tab hak akses yang berisi <i>form</i> input nama, <i>email</i> dan <i>password</i> untuk manajemen hak akses <i>user admin</i>	Sesuai
2.	Admin mengakses halaman <i>profile</i> pada tab hak akses kemudian ingin menambah akun hak akses <i>user admin</i> dan input <i>email</i> yang sudah digunakan atau <i>password</i> yang salah	Sistem menampilkan pesan error “ <i>email</i> sudah digunakan atau <i>password</i> tidak cocok”	Sesuai

#### 4.4.7. Pengujian *Black Box* – Halaman Website Gizila

**Tabel 4.13** *Black Box Testing* Halaman Website Gizila

<b>Skenario Uji</b>		<b>Halaman Home</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/home">http://127.0.0.1:8000/home</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1.	<i>User</i> mengakses halaman <i>home</i>	Sistem menampilkan halaman <i>home</i> berisi informasi umum yang dapat diakses tanpa login	Sesuai
2.	<i>User</i> input kata kunci artikel tidak ditemukan pada halaman <i>home</i>	Sistem menampilkan pesan atau hasil kosong	Sesuai
<b>Skenario Uji</b>		<b>Hitung IMT</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/kalkulator-massa-tubuh">http://127.0.0.1:8000/kalkulator-massa-tubuh</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1.	<i>User</i> mengakses halaman kalkulator IMT	Sistem menampilkan halaman formulir perhitungan Indeks Massa Tubuh yang berisi input berat badan, tinggi badan, usia, jenis kelamin, dan tingkat aktivitas	Sesuai
2.	<i>User</i> klik tombol hitung IMT, kemudian ada salah satu kolom yang dikosongkan	Sistem menampilkan pesan validasi error pada kolom yang belum diisi	Sesuai
<b>Skenario Uji</b>		<b>Hasil Hitung IMT</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/kalkulator-massa-tubuh/hitung">http://127.0.0.1:8000/kalkulator-massa-tubuh/hitung</a></u>	
<b>No</b>	<b>Aspek yang diuji</b>	<b>Hasil yang diharapkan</b>	<b>Hasil yang diuji</b>
1.	<i>User</i> klik tombol hitung IMT setelah semua kolom pada formulir IMT sudah diisi	Sistem menampilkan hasil perhitungan IMT berdasarkan data yang diinput <i>user</i>	Sesuai
2.	<i>User</i> klik tombol hitung IMT, kemudian ada	Sistem menampilkan pesan validasi error pada kolom yang belum diisi	Sesuai

	kolom pada formulir IMT belum diisi		
<b>Skenario Uji</b>		<b>Hitung Gizi Harian</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/kalkulator-gizi-harian">http://127.0.0.1:8000/kalkulator-gizi-harian</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	<i>User</i> mengakses halaman kalkulator gizi harian	Sistem menampilkan halaman kalkulator yang berisi form pemilihan menu sarapan, makan siang, dan makan malam, beserta kolom porsi dan opsi upload/ambil gambar menu	Sesuai
2.	<i>User</i> memilih menu makanan/minuman, mengisi porsi, dan menambahkan menu lainnya	Sistem mencatat data input menu dan porsi untuk perhitungan gizi harian	Sesuai
3.	<i>User</i> mengunggah atau mengambil gambar menu	Sistem menampilkan gambar menu pada kolom yang tersedia	Sesuai
4.	<i>User</i> menyelesaikan input data untuk semua sesi waktu makan	Sistem menampilkan hasil perhitungan total gizi harian sesuai data yang dimasukkan	Sesuai
5.	<i>User</i> klik tombol hitung gizi harian, kemudian ada salah satu kolom yang dikosongkan	Sistem menampilkan pesan validasi error	Sesuai
6.	<i>User</i> akses halaman kalkulator-gizi-harian, tanpa mengisi formulir halaman perhitungan IMT	Sistem menampilkan pesan <i>popup</i> untuk diarahkan ke halaman IMT	Sesuai
<b>Skenario Uji</b>		<b>Hasil Hitung Gizi Harian</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/kalkulator-gizi-harian/hitung">http://127.0.0.1:8000/kalkulator-gizi-harian/hitung</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	<i>User</i> klik tombol hitung gizi harian setelah semua	Sistem menampilkan hasil perhitungan gizi harian	Sesuai

	kolom pada formulir gizi harian sudah diisi	berdasarkan data yang diinput user	
2.	User klik tombol hitung gizi harian, kemudian ada salah satu kolom yang dikosongkan	Sistem menampilkan pesan validasi error	Sesuai
3.	User akses halaman kalkulator-gizi-harian/hitung, tanpa mengisi formulir halaman perhitungan IMT	Sistem menampilkan pesan <i>pop-up</i> untuk diarahkan ke halaman IMT	Sesuai
<b>Skenario Uji</b>		<b>Halaman Artikel</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/blog">http://127.0.0.1:8000/blog</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	User akses halaman blog	Sistem menampilkan halaman daftar artikel yang berisi informasi edukasi	Sesuai
2.	User klik tombol <i>pagination</i> artikel	Sistem menampilkan halaman artikel berdasarkan nomor halaman atau tombol panah	Sesuai
3.	User input kata kunci artikel tidak ditemukan pada halaman artikel atau slug artikel	Sistem menampilkan pesan atau hasil kosong	Sesuai
<b>Skenario Uji</b>		<b>Halaman Blog</b>	
<b>HTTP Method</b>		<u><a href="http://127.0.0.1:8000/about">http://127.0.0.1:8000/about</a></u>	
No	Aspek yang diuji	Hasil yang diharapkan	Hasil yang diuji
1.	User mengakses halaman <i>about</i>	Sistem menampilkan halaman <i>about</i> berisi informasi kontak, alamat dan terkait tentang Gizila	Sesuai

#### **4.5. User Acceptance Test (UAT)**

*User Acceptance Testing* (UAT) adalah tahap pengujian yang dilakukan oleh pengguna akhir untuk memastikan bahwa sistem telah memenuhi kebutuhan dan harapan pengguna. Pada pengembangan sistem informasi Giza berbasis web, UAT dilakukan untuk memverifikasi bahwa setiap fitur berfungsi dan mendukung proses kerja pengguna secara optimal. Melalui pengujian ini, umpan balik langsung dari pengguna dapat diperoleh untuk mengidentifikasi kekurangan atau perbaikan yang diperlukan sebelum sistem diimplementasikan secara penuh. Hasil pelaksanaan UAT pada sistem informasi Giza tertera pada tabel berikut.

##### **4.5.1. User Acceptance Test Admin**

**Tabel 4.14 Pengujian UAT Admin**

No	Fitur	Proses	Hasil yang diinginkan	Hasil Pengujian
1.	<i>Login</i>	Admin input <i>email</i> dan <i>password</i> yang valid	Admin berhasil <i>login</i>	Berhasil
2.	<i>Dashboard</i>	Admin mengakses halaman <i>dashboard</i> setelah berhasil <i>login</i>	Admin berhasil melihat <i>dashboard</i>	Berhasil
3.	Artikel	Admin menambah, mengedit, atau menghapus data artikel	Admin berhasil memperbarui data artikel	Berhasil
4.	Kategori	Admin menambah, mengedit, atau menghapus kategori artikel	Admin berhasil memperbarui kategori artikel	Berhasil
5.	Profil	Admin melihat dan mengubah, informasi profil	Admin berhasil memperbarui data profil	Berhasil
6.	Hak Akses	Admin melihat, membuat, mengubah, menghapus informasi hak akses admin yang terdaftar di hak akses admin	Admin berhasil memperbarui informasi hak akses admin	Berhasil
7.	<i>Logout</i>	Admin klik tombol <i>logout</i>	Admin berhasil <i>logout</i>	Berhasil

#### 4.5.2. *User Acceptance Test* Publik

**Tabel 4.15** Pengujian UAT Publik

No	Fitur	Proses	Hasil yang diinginkan	Hasil Pengujian
1.	<i>Home</i>	<i>User</i> mengakses halaman <i>home</i>	<i>User</i> berhasil melihat halaman <i>home</i>	Berhasil
2.	Kalkulator Indeks Massa Tubuh (IMT)	<i>User</i> input data pada <i>form</i> perhitungan IMT	Sistem menampilkan hasil perhitungan IMT	Berhasil
3.	Hasil Perhitungan Kalkulator Indeks Massa Tubuh (IMT)	Sistem menghitung dan menampilkan hasil perhitungan IMT berdasarkan input <i>user</i>	<i>User</i> berhasil melihat hasil perhitungan berdasarkan data yang diinput	Berhasil
4.	Kalkulator Gizi Harian	<i>User</i> input data pada <i>form</i> perhitungan gizi harian	Sistem menampilkan hasil perhitungan gizi harian	Berhasil
5.	Hasil Perhitungan Kalkulator Gizi Harian	Sistem menghitung dan menampilkan hasil perhitungan gizi harian berdasarkan input <i>user</i>	<i>User</i> berhasil melihat hasil perhitungan berdasarkan data yang diinput	Berhasil
6.	Artikel	<i>User</i> mengakses halaman artikel	<i>User</i> berhasil melihat halaman artikel	Berhasil
7.	<i>About</i>	<i>User</i> mengakses halaman <i>about</i>	<i>User</i> berhasil melihat halaman <i>about</i>	Berhasil

#### 4.6. Tinjauan Hasil Uji *System Usability Scale* (SUS)

Tahapan pengujian aplikasi memanfaatkan metode *System Usability Scale* (SUS) guna menilai tingkat efektivitas, efisiensi, serta kepuasan pengguna terhadap sistem (Susila dan Arsa, 2023).

Perhitungan hasil pengujian *website* Gizila menggunakan instrumen SUS dilakukan dengan mengikuti aturan tertentu. Setiap butir pernyataan memiliki skor kontribusi dengan rentang nilai skala 1 sampai 5. Untuk pernyataan bernomor ganjil, skor diperoleh dari hasil pengurangan skala jawaban responden dengan angka 1. Sedangkan untuk pernyataan

bernomor genap, skor dihitung dengan mengurangkan skala jawaban responden dari angka 5. Nilai keseluruhan *System Usability* diperoleh dengan menjumlahkan seluruh skor kontribusi, kemudian mengalikannya dengan faktor 2,5.

Adapun hasil penelitian yang mencakup desain dan kuesioner akan dibahas pada bagian ini. Pembahasan difokuskan pada tingkat kegunaan fitur-fitur yang terdapat pada *website*.

$$\bar{x} = \frac{\sum x}{n}$$

Keterangan:

$\chi$  = Skor rata-rata.

$\Sigma x$  = Jumlah skor SUS.

*n* = Jumlah responden.

Skor SUS = ((R1 - 1) + (5 - R2) + (R3 - 1) +(5 - R4) + (R5 - 1) + (5 - R6) +(R7 - 1)) x 2.5).

Dalam penelitian ini ada dua aspek yang diuji yaitu dari sisi user publik yang mengakses web Gizila dan *user admin* yang mengakses fitur *dashboard* admin Gizila, masing-masing terdiri dari 14 pertanyaan. Kuesioner disusun menggunakan *Google Form* agar dapat diakses oleh responden. Sebelum mengisi kuesioner, responden diberikan arahan untuk terlebih dahulu melihat hasil rancangan web Gizila yang dibuat peneliti serta memahami tata cara pengisian. Selanjutnya, responden menjawab setiap pertanyaan sesuai dengan pendapat masing-masing. Berikut hasil penilaian responden dilampirkan pada tabel dibawah ini.

**Tabel 4.16** Hasil Penilaian Responden *Usability Testing User Publik*

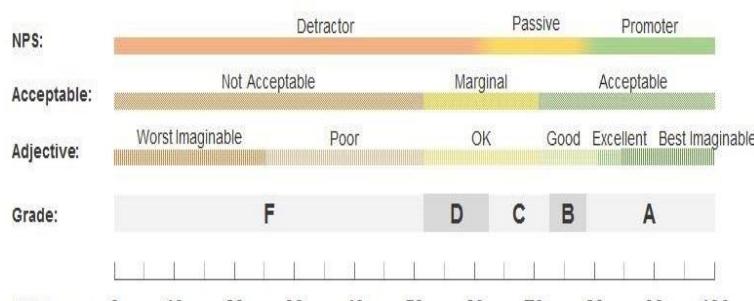
Tabel 4.15 diatas merupakan hasil penilaian responden *usability testing* dari sisi user publik yang mengakses web Gizila dari hasil perhitungan dengan menggunakan rumus skor SUS =  $((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1)) \times 2.5$ .

**Tabel 4.17** Hasil Penilaian Responden *Usability Testing User Admin*

Responden	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12	Q13	Q14	Jumlah	Hasil Penilaian (Jumlah x 2,5)
R1	4	0	4	0	3	0	4	2	1	2	3	0	4	0	27	67,5
R2	3	1	4	1	3	0	3	1	0	3	0	1	4	0	24	60
R3	3	1	3	1	3	1	3	1	1	2	2	1	3	1	26	65
R4	3	1	2	1	3	1	3	1	2	2	1	2	2	2	26	65
R5	3	2	4	2	4	0	3	1	1	2	1	1	3	1	29	72,5
R6	4	0	4	0	4	0	4	0	1	0	1	0	4	0	22	55
R7	4	0	4	0	4	2	4	0	1	1	1	0	4	0	25	62,5
Skor rata-rata (hasil akhir):															63,9	

Tabel 4.16 diatas merupakan hasil penilaian responden *usability testing* dari sisi *user admin* yang mengakses fitur *dashboard* admin Gizila dari hasil perhitungan dengan menggunakan rumus skor SUS =  $((R1 - 1) + (5 - R2) + (R3 - 1) + (5 - R4) + (R5 - 1) + (5 - R6) + (R7 - 1)) \times 2.5$ .

#### 4.6.1. Tahapan Perolehan Skor SUS



**Gambar 4.23** Standar Perolehan Skor SUS

Sumber: <https://www.jurnalitp.web.id/index.php/jitp/article/view/68/31>

Berdasarkan hasil penilaian kuesioner dengan metode *System Usability Scale* (SUS) pada *website* Gizila, diperoleh nilai rata-rata 62,1 dari sisi *user* publik dan diperoleh nilai rata-rata 63,9 dari sisi *user* admin dengan rincian sebagai berikut:

- a. Tingkat *Acceptable* berada pada kategori *Marginal*

Nilai ini menunjukkan bahwa secara umum, penggunaan *website* Gizila masih memerlukan perbaikan yang signifikan agar dapat mencapai tingkat penerimaan yang diharapkan. Beberapa pengguna kemungkinan menghadapi beberapa kendala atau menemukan beberapa aspek yang kurang memuaskan dalam penggunaan *website*.

- b. Tingkat *Grade Scale* berada pada kategori D

Hasil ini mencerminkan kualitas yang belum optimal dari segi fungsionalitas, antarmuka, maupun kinerja, sehingga diperlukan perbaikan agar *website* dapat berfungsi lebih efektif dan efisien dalam memenuhi kebutuhan pengguna.

- c. *Adjective Rating* berada pada kategori Ok

Meskipun penilaian ini menunjukkan kinerja yang tergolong standar, sebagian besar aspek *website* masih dapat dianggap cukup layak digunakan. Namun demikian, perbaikan tetap diperlukan untuk meningkatkan kualitas secara keseluruhan.

Kesimpulannya, hasil pengujian SUS menunjukkan adanya ruang yang cukup besar untuk perbaikan pada *website* Gizila. Upaya peningkatan fungsionalitas, antarmuka, dan performa secara menyeluruh diperlukan agar pengalaman pengguna menjadi lebih memuaskan dan tujuan sistem dapat tercapai dengan lebih efektif.

## **BAB V**

### **PENUTUP**

#### **5.1. Kesimpulan**

Berdasarkan hasil penelitian, analisis, dan implementasi sistem yang telah dilakukan, dapat disimpulkan bahwa perancangan Sistem Informasi Gizila berbasis web telah berhasil mencapai tujuannya. Sistem ini memberikan kemudahan bagi masyarakat umum untuk menghitung kebutuhan gizi harian, memantau status gizi melalui kalkulator Indeks Massa Tubuh (IMT), dan mengakses informasi edukatif seputar gizi secara instan dan *real-time*. Penggunaan metode perhitungan *Harris Benedict* telah berhasil diimplementasikan untuk memberikan estimasi kebutuhan kalori yang dipersonalisasi. Selain itu, penerapan metode *Rapid Application Development* (RAD) terbukti efektif dan efisien, memungkinkan pengembangan sistem dilakukan secara cepat dan fleksibel sesuai dengan kebutuhan yang telah diidentifikasi. Berdasarkan hasil pengujian menggunakan metode *black box testing* dan *User Acceptance Test* (UAT), seluruh fungsionalitas sistem dinyatakan telah berjalan sesuai dengan yang diharapkan dan diterima dengan baik oleh pengguna. Berdasarkan hasil penilaian kuesioner dengan metode *System Usability Scale* (SUS) pada web Gizila, diperoleh nilai rata-rata 62,1 dari sisi *user* publik dan diperoleh nilai rata-rata 63,9 dari sisi *user* admin. Meskipun penilaian ini menunjukkan kinerja yang tergolong standar, sebagian besar aspek web masih dapat dianggap cukup layak digunakan.

Harapan peneliti terkait penelitian ini dapat bermanfaat bagi masyarakat untuk mewujudkan sebuah *platform digital* yang fungsional untuk mendukung kebutuhan bisnis Gizila dalam menyediakan layanan edukasi gizi kepada publik secara lebih luas dan interaktif.

#### **5.2. Saran**

Menyadari bahwa sistem yang dibangun dalam penelitian ini masih memiliki ruang untuk pengembangan, maka peneliti memberikan beberapa saran yang diharapkan dapat menjadi acuan untuk penelitian atau pengembangan selanjutnya. Adapun saran yang dapat diberikan adalah sebagai berikut:

- 1) Menambahkan fitur konsultasi dengan ahli gizi, mengingat penelitian ini tidak mencakup penggunaan oleh pakar kesehatan profesional, disarankan pada pengembangan selanjutnya untuk menambahkan fitur konsultasi *online* dengan ahli gizi. Fitur ini dapat meningkatkan validitas dan kepercayaan pengguna terhadap saran gizi yang diberikan.
- 2) Mengembangkan versi aplikasi *mobile*, untuk meningkatkan jangkauan dan kemudahan akses bagi pengguna, disarankan agar sistem ini dapat dikembangkan lebih lanjut

menjadi aplikasi berbasis *mobile* (*Android/iOS*), sehingga pengguna dapat memantau asupan gizi kapan saja dan di mana saja dengan lebih praktis melalui aplikasi.

- 3) Integrasi dengan perangkat kesehatan, sistem dapat ditingkatkan dengan fitur integrasi data dari perangkat kesehatan digital atau *smartwatch* untuk melacak tingkat aktivitas fisik pengguna secara otomatis. Hal ini akan membuat perhitungan kebutuhan kalori menjadi lebih dinamis dan akurat.
- 4) Memperluas *database* makanan dengan menambahkan variasi makanan yang lebih beragam ke dalam database sistem, terutama makanan khas lokal atau produk makanan kemasan yang umum dijumpai di Indonesia, akan sangat membantu pengguna dalam mencatat asupan harian dengan lebih presisi.

## DAFTAR PUSTAKA

- Alvrahesta, A., Windasari, I. P., & Prasetijo, A. B. (2023). Rancang Bangun Sistem Informasi Penerimaan Beasiswa Sariraya Co. Ltd. Menggunakan Framework Laravel dan Bootstrap. *Jurnal Teknik Komputer*, 2(1), 1–10. <https://doi.org/10.14710/jtk.v2i1.37723>
- Ardhana, V. Y. P. (2024). Perancangan Sistem Informasi Kedai Kopi Menggunakan Metode Rapid Application Development (RAD). *JDMIS: Journal of Data Mining and Information Systems*, 2(1), 43–49. <https://doi.org/10.54259/jdmis.v2i1.2422>
- Ardianto, D., & Widiyatmoko, A. T. (2024). Color Detector in an Image using Python and Computer Vision Library. *Journal of Intelligent Systems and Information Technology*, 1(1), 25–30. <https://doi.org/10.61971/jisit.v1i1.27>
- Azhariyah, S., Rosid, U. A., Abdurrachman, T., & Masriah, W. S. N. (2023). *Jurnal Indonesia : Manajemen Informatika dan Komunikasi Rancang Bangun Aplikasi Pendataan Status Gizi Pada Balita Berbasis Web di Desa Rawalele*. 4(1), 224–232.
- Excel, J. J., Teguh, R., & Sylviana, F. (2024). Rancang Bangun Sistem Informasi Monitoring Gizi Ibu Hamil dan Anak Berbasis Website. *Jsik*, 5(1), 1–7.
- Hanny, Samsugi, S., & Sulistiyawati, A. (2023). Rancang Bangun Sistem Informasi Pendataan Calon Penerima Bantuan Sosial Dan Desa Berbasis Web (Studi Kasus : Desa Cilimus). *Jurnal Teknologi Dan Sistem Informasi*, 4(3), 328–339.
- Hidayati, A. T., Widayantoro, A. E., & Ramadhan, H. J. (2023). Perancangan Sistem Informasi Wirausaha Mahasiswa (Siwirma) Berbasis Web dengan Unified Modelling Languange (UML). *Jurnal Penelitian Rumpun Ilmu Teknik*, 2(4), 86–107. <https://doi.org/10.55606/juprit.v2i4.2906>
- Mardiana, I., Wahyudin, & Junaeti, E. (2024). Pengembangan Learning Management System dengan Framework Laravel dan Tailwind CSS. *Multinetics*, 10(1), 40–49. <https://doi.org/10.32722/multinetics.v10i1.6678>
- Muhammad, S. D. A., Sulistiyowati, Y. E., Karima, T. F., Huda, M. S., Muttaqin, N., & Ghufron. (2023). Perancangan Aplikasi Countional Dalam Mengimplementasi Pengukuran Gizi Sehari-Hari. *Jurnal Transistor Elektro Dan Informatika (TRANSISTOR EI)*, 5(2), 70–76.
- Mulyanto, Y., Karisma, Y., & Maharani, U. (2020). Rancang Bangun Sistem Monitoring Perkembangan Anak Di Tkit Taamasa Meggunakan Metode Spiral. *Jurnal*

- InformatRancang Bangun Sistem Monitoring Perkembangan Anak Di Tkit Taamasa Meggunakan Metode Spiralika, Teknologi Dan Sains, 2(3), 190–195.*  
<https://doi.org/10.51401/jinteks.v2i3.754>
- Murdiani, D., & Hermawan, H. (2022). Perbandingan Metodologi Waterfall Dan Rad (Rapid Application Development) Pada Pengembangan Sistem Informasi. *JUTEKIN (Jurnal Teknik Informatika)*, 10(2), 14–23. <https://doi.org/10.51530/jutekin.v10i2.655>
- Noorikhsan, F. F., Ramdhani, H., Sirait, B. C., & Khoerunisa, N. (2023). Dinamika Internet, Media Sosial, dan Politik di Era Kontemporer: Tinjauan Relasi Negara-Masyarakat. *Journal of Political Issues*, 5(1), 95–109. <https://doi.org/10.33019/jpi.v5i1.131>
- Novitasari, Y. S., Adrian, Q. J., & Kurnia, W. (2021). *Rancang Bangun Sistem Informasi Media Pembelajaran Berbasis Website ( Studi Kasus : Bimbingan Belajar De Potlood )*. 2(3), 136–147.
- Nugroho, F. A., Fadilah, D., Sumitro, C. M., & Saputra, R. A. (2021). Rancang Bangun Sistem Informasi Sebaran Distribusi KIS Provinsi Sulawesi Tenggara Berbasis Web. *Jurnal Teknologi Informasi, Komputer Dan Aplikasinya (JTIKA)*, 4(2), 182–193. <http://jtiika.if.unram.ac.id/index.php/JTIKA/>
- Nuridza, A. B., Fitriastuti, F., & Setiyorini, A. (2025). Implementasi Tall Stack Dalam Pengembangan Sistem Informasi Peminjaman Alat Dengan Menggunakan Rapid Application Development. *Jurnal Informatika Teknologi Dan Sains (Jinteks)*, 7(2), 729–738. <https://doi.org/10.51401/jinteks.v7i2.5660>
- Orisa, M., Faisol, A., & Ashari, M. I. (2023). Perancangan Website Company Profile Menggunakan Design Science Research Methodology (Dsrm). *Jurnal Informatika Teknologi Dan Sains*, 5(1), 160–164. <https://doi.org/10.51401/jinteks.v5i1.2576>
- Putri, R. E., Widya, R., & Yusman, Y. (2023). Prototype Sistem Informasi Bimbingan Dan Konseling Menggunakan Figma. *Jurnal Indonesia : Manajemen Informatika Dan Komunikasi*, 4(2), 540–551. <https://doi.org/10.35870/jimik.v4i2.246>
- Putro, L. H., Aini, A. N., Rahmadina, A., Haq, E. H. H., Prakosa, R. F., Fendari, S. N. L., & Hidayat, A. T. (2023). Evaluasi Sistem Berbasis Website Menggunakan Metode System Usability Scale (SUS). *Jurnal Informatika Dan Teknologi Pendidikan*, 3(2), 54–61. <https://doi.org/10.25008/jitp.v3i2.68>
- Qamaruzzaman, M. H., Sutami, Sam'ani, & Budiman, I. (2022). Penerapan Metode Harris

- Benedict Pada Media Informasi Kebutuhan Gizi Harian Berbasis Android. *JURIKOM (Jurnal Riset Komputer)*, 9(5), 1346. <https://doi.org/10.30865/jurikom.v9i5.4867>
- Rahmawati, E. F., Ayuningtyas, & Sagirani, T. (2023). Penerapan Metode Double Diamond pada Desain User Interface Website. *Jurnal Komunika: Jurnal Komunikasi, Media Dan Informatika*, 11(1), 11–22. <https://doi.org/10.31504/komunika.v11i1.4991>
- Ramdany, S. W., Kaidar, S. A., Aguchino, B., Amelia Alira Putri, C., & Anggie, R. (2024). Penerapan UML Class Diagram dalam Perancangan Sistem Informasi Perpustakaan Berbasis Web. *Journal of Industrial and Engineering System*, 5(1). <https://doi.org/10.31599/2e9afp31>
- Rian, S. M., Sudin, S., & Ibrahim, A. (2024). *Implementasi sistem penentu kalori sehat harian menggunakan persamaan harris benedict berbasis web*. 17(2), 3–5.
- Sam’ani, Haris, F., Ichsan, M., & Qamaruzzaman, M. H. (2023). Sosialisasi Media Informasi Kebutuhan Gizi Harian Pada Unit Pelaksana Teknis (UPT) Puskesmas Mandomai. *Jurnal Pengabdian Masyarakat (Jupemas)*, 4(1), 49–55. <https://doi.org/10.36465/jupemas.v4i1.1021>
- Sinlae, F., Maulana, I., Setiyansyah, F., & Ihsan, M. (2024). Pengenalan Pemrograman Web: Pembuatan Aplikasi Web Sederhana Dengan PHP dan MYSQL. *Jurnal Siber Multi Disiplin (JSMD)*, 2(2)Pengenalan Pemrograman Web: Pembuatan Aplikasi Web Sederhana Dengan PHP dan MYSQL), 68–82. <https://creativecommons.org/licenses/by/4.0/>
- Surianto, D. F., Wahid, M. S. N., Parenreng, J. M., Wahid, A., Zain, S. G., Edy, M. R., & Risal, A. A. N. (2023). PKM Figma Training for Information System Prototype Design. *Vokatek : Jurnal Pengabdian Masyarakat*, 1(2), 57–63.
- Susila, A. A. N. H., & Arsa, D. M. S. (2023). Analisis System Usability Scale (SUS) dan Perancangan Sistem Self Service Pemesanan Menu di Restoran Berbasis Web. *Majalah Ilmiah UNIKOM*, 21(1), 3–8. <https://doi.org/10.34010/miu.v21i1.10683>
- Wahyudi, I., Fahrullah, Alameka, F., & Haerullah. (2023). Analisis Blackbox Testing Dan User Acceptance Testing Terhadap Sistem Informasi SolusimedsoSKU. *Jurnal Teknosains Kodepena |*, 04(01), 1–9.
- Wiarso, R. H., & Anwar, T. (2024). Implementasi Framework Tailwindcss Pada Frontend Website Supply Chain Management Cv. Marvelindo Utama. *JATI (Jurnal Mahasiswa*

- Teknik Informatika), 8(4), 7561–7569. <https://doi.org/10.36040/jati.v8i4.10094>*
- Yakub, H., Daniawan, B., Wijaya, A., & Damayanti, L. (2024). Sistem Informasi E-Commerce Berbasis Website Dengan Metode Pengujian User Acceptance Testing. *JSITIK: Jurnal Sistem Informasi Dan Teknologi Informasi Komputer*, 2(2), 113–127. <https://doi.org/10.53624/jsitik.v2i2.362>
- Yuliana, D., Suwarti, Ayu, F., & Purnomo, N. (2024). *Sistem Informasi Kebutuhan Gizi Pada Ibu Hamil Dengan Metode Cooper Pada Puskesmas Ulak Karang Berbasis Web.* 16(1), 165–170.

## LAMPIRAN-LAMPIRAN

### 1. Surat pengantar/permohonan data



**UNIVERSITAS NAHDLATUL ULAMA INDONESIA**  
FAKULTAS TEKNIK DAN ILMU KOMPUTER

Jln. Taman Amir Hamzah No.5 Jakarta 10320  
021 390 6501 - 021 315 6864  
ftik@unusia.ac.id - [www.unusia.ac.id](http://www.unusia.ac.id)

Nomor : 160/DK.FTIK/200.12.14/VIII/2025  
Lampiran : -  
Perihal : **Surat Pengantar Penelitian**

Kepada Yth.,  
**Bapak Brian Fadli Fahmi**  
**PT. Graisena Usaha Nusantara**

di Tempat

*Assalamu'alaikum Warahmatullahi Wabarakatuh*

Salam silaturahmi teriring doa kami sampaikan, semoga kita semua senantiasa mendapatkan rahmat dan hidayah Allah SWT serta kesehatan dalam menjalankan aktivitas sehari-hari. Amin.

Sehubungan dengan penyusunan Proposal Karya Tulis Ilmiah Mahasiswa Program Studi Teknik Informatika Universitas Nadhlatul Ulama Indonesia, maka dengan ini kami mohon agar berkenan membantu memberikan izin untuk penyusunan Proposal Skripsi bagi Mahasiswa dibawah ini:

Nama	:	Ulfa Hidayanti
NIM	:	21250011
Program Studi	:	Sistem Informasi
Judul Skripsi	:	Rancang Bangun Sistem Informasi Gizila Untuk Kebutuhan Gizi Harian Berbasis Web Menggunakan Metode Rapid Application Development (RAD)

Demikian Surat ini disampaikan untuk mendapatkan proses lebih lanjut, atas perhatian dan kerjasamanya diucapkan terimakasih.

Jakarta, 01 Agustus 2025  
**Dekan Fakultas Teknik dan Ilmu Komputer**

  
**Adrinoviarini, M.Sc.**  
**NIDN: 0375118007**

2. Form Bimbingan

**FORM BIMBINGAN**

Nama : Ulfa Hidayanti  
 Judul Skripsi : Rancang Bangun Sistem Informasi Gizila Untuk Kebutuhan Gizi  
 Harian Berbasis Web Menggunakan Metode *Rapid Application Development* (RAD)  
 Pembimbing : Handy Fernandy, S.T., M.M.S.I.

No	Hari/Tanggal	Perbaikan	Paraf
1	Rabu, 14 Mei 2025	Penambahan fitur-fitur website Gizila.	
2	Senin, 19 Mei 2025	✓ Penambahan fitur: <i>dashboard</i> admin artikel, dan fitur login/register sebagai hak akses masuk ke <i>dashboard</i> .	
3	Sabtu, 14 Juni 2025	✓ Perkembangan <i>project website</i> .	
4	Minggu, 15 Juni 2025	✓ Perkembangan <i>project website</i> .	
5	Kamis, 19 Juni 2025	✓ Perkembangan <i>project website</i> .	
6	Selasa, 24 Juni 2025	✓ Latar belakang diperbarui dengan menambah referensi dan penghubung kalimat yang baik. ✓ Batasan masalah diperbaiki. ✓ Bab 2 di penelitian terdahulu ditambahkan tentang penjelasan detail.	
7	Sabtu, 26 Juli 2025	✓ Pengecekan <i>testing website</i> Gizila	

Jakarta, 26 Juli 2025

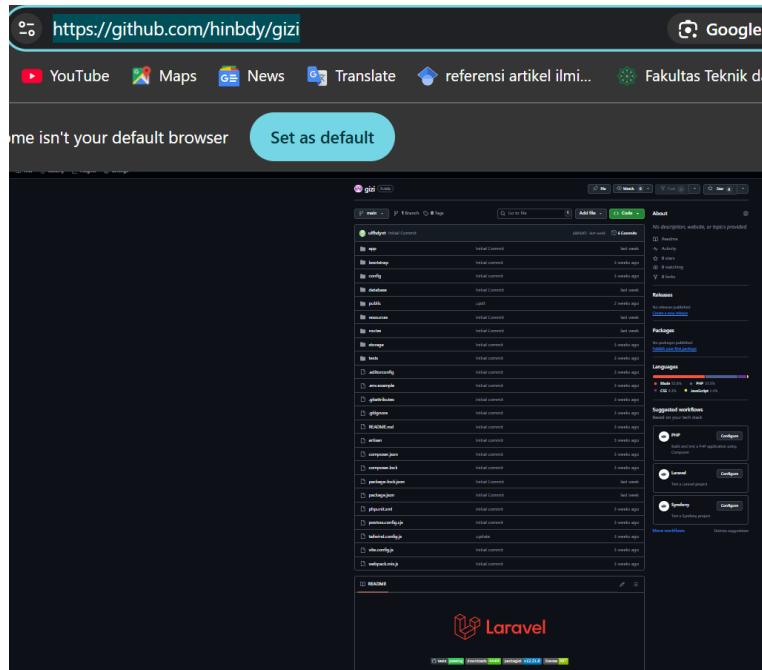
Pembimbing,



**Handy Fernandy, S.T., M.M.S.I.**

NIDN. 0309029202

3. *Source code dan file database (link github)*



<https://github.com/hinbdy/gizi>, <https://gizila.biz.id/>

4. Dokumentasi penelitian (bukti wawancara, observasi)



Bersama ahli gizi di Gizila yaitu Mba Indah Puji Lestari S.Gz.

5. Naskah wawancara atau kuesioner

- ✓ Tabel pernyataan kuesioner dibawah ini dibuat pada saat membuat laporan skripsi.

Tabel Pernyataan *Usability Testing User Publik Web Gizila*

No	Pernyataan <i>User Publik</i>	Keterangan dan Skala				
		SS	S	TS	KS	STS
		5	4	3	2	1
1	Website Gizila memudahkan saya untuk berinteraksi dengan sistem untuk mendapatkan informasi gizi yang saya butuhkan.	3	3	1	0	0
2	Saya dapat memahami alur navigasi dengan mudah untuk menjelajahi berbagai fitur dan konten di website Gizila.	4	2	1	0	0
3	Website Gizila memiliki fitur interaktif seperti kalkulator IMT dan gizi harian yang sangat membantu saya dalam menghitung kebutuhan gizi harian.	4	3	0	0	0
4	Saya mudah mengingat letak menu dan navigasi untuk menjelajahi fitur dan konten di website Gizila.	3	3	1	0	0
5	Istilah atau fitur menu memberikan petunjuk yang jelas tentang fungsi setiap fitur.	4	3	0	0	0
6	Saya dapat mengakses fitur-fitur utama seperti kalkulator gizi dan artikel dengan cepat dan tanpa hambatan.	4	2	1	0	0
7	Saya memahami konten informasi artikel edukasi, hasil perhitungan gizi yang disajikan di website Gizila dengan mudah.	4	3	0	0	0
8	Saya dapat memperoleh informasi yang saya butuhkan, seperti status gizi (IMT), estimasi kebutuhan kalori, dan tips kesehatan dari website Gizila.	4	3	0	0	0
9	Saya merasa kesulitan saat mencoba menemukan fitur atau menu tertentu yang saya cari di website Gizila.	0	1	0	6	0
10	Jika saya melakukan kesalahan misalnya: salah mengisi data di kalkulator, sistem memberikan petunjuk yang jelas untuk memperbaikinya.	1	1	4	1	0
11	Selama menggunakan website, saya menemukan pesan kesalahan (error) atau masalah teknis yang mengganggu.	0	0	1	3	2
12	Secara keseluruhan, saya merasa puas dengan kualitas informasi dan tampilan desain website Gizila.	3	3	1	0	0
13	Desain website Gizila memiliki tata letak (layout) yang rapih, sehingga perpaduan gambar dan teks mudah dibaca dan informatif.	4	3	0	0	0
14	Komposisi warna dan penempatan ikon-ikon pada website Gizila terlihat serasi dan tidak membingungkan.	4	2	1	0	0

Tabel Pernyataan *Usability Testing User Admin Web Gizila*

No	Pernyataan <i>User Admin</i>	Keterangan dan Skala				
		SS	S	TS	KS	STS
		5	4	3	2	1
1	Dashboard admin Gizila memudahkan saya mengelola data dan informasi.	3	4	0	0	0
2	Alur navigasi di dashboard mudah dipahami untuk mengakses berbagai menu dan fitur administrasi.	3	3	1	0	0
3	Fitur interaktif seperti pengelolaan data dan informasi artikel, memantau grafik dashboard, memudahkan saya menjalankan tugas administrasi dengan efektif.	5	1	1	0	0
4	Letak menu dan navigasi di dashboard mudah diingat untuk untuk menjelajahi fitur manajemen artikel di dashboard admin Gizila.	3	3	1	0	0
5	Istilah atau fitur menu memberikan petunjuk yang jelas tentang fungsi setiap fitur.	3	4	0	0	0
6	Saya dapat mengakses fitur-fitur utama seperti manajemen artikel dengan cepat dan tanpa hambatan.	4	2	1	0	0
7	Data, laporan grafik dashboard mudah saya pahami untuk keperluan pengelolaan data dan informasi.	3	4	0	0	0
8	Dashboard admin Gizila menyediakan informasi yang saya butuhkan.	2	4	1	0	0
9	Saya mengalami kesulitan menemukan menu atau fitur tertentu di dashboard.	0	0	1	5	1
10	Jika terjadi kesalahan input data, sistem memberikan petunjuk yang jelas untuk memperbaikinya.	1	1	5	0	0
11	Selama menggunakan dashboard, saya menemukan pesan kesalahan (error) atau masalah teknis yang mengganggu.	0	1	2	3	1
12	Secara keseluruhan, saya merasa puas dengan manajemen informasi dan tampilan desain dashboard admin Gizila.	3	3	1	0	0
13	Desain dashboard admin Gizila memiliki tata letak (layout) yang rapih, sehingga perpaduan gambar dan teks mudah dibaca dan informatif.	4	3	0	0	0
14	Komposisi warna dan penempatan ikon-ikon pada dashboard admin Gizila terlihat serasi dan tidak membingungkan.	4	2	1	0	0

- ✓ Tabel pernyataan kuesioner dibawah ini dibuat pada saat membuat laporan proposal.

Tabel Pernyataan Kepuasan Pengguna Sebagai *Followers* Instagram Gizila

<b>No</b>	<b>Pernyataan</b>	<b>Keterangan</b>			
		<b>SP</b>	<b>P</b>	<b>TP</b>	<b>STP</b>
<b>Performance (Kinerja)</b>					
1	Akun instagram Gizila sangat mudah di akses oleh pengguna	6	8	0	0
2	Akun instagram Gizila dapat berfungsi secara optimal di perangkat yang sedang digunakan.	5	8	0	1
3	Akun instagram Gizila memberikan informasi gizi secara <i>real-time</i> .	6	7	1	0
4	Unggahan konten di akun instagram gizila dilakukan secara konsisten.	5	7	2	0
<b>Information (Informasi)</b>					
1	Informasi yang diberikan oleh akun instagram Gizila mudah dipahami dan dapat dipercaya.	4	10	0	0
2	Akun instagram Gizila memberikan informasi terbaru atau up-to-date.	4	9	1	0
3	Informasi yang dihasilkan oleh akun instagram Gizila sudah sesuai dengan yang dibutuhkan.	6	8	0	0
4	Dengan mengakses akun instagram Gizila Informasi mengenai Gizi menjadi lebih mudah diketahui.	8	6	0	0
<b>Economics (Ekonomi)</b>					
1	Informasi yang diperoleh dari akun Gizila membantu menghemat waktu dalam mencari informasi gizi.	5	8	1	0
2	Penggunaan data internet dalam mengakses akun instagram Gizila terbilang murah.	4	9	1	0
3	Informasi gizi yang diberikan akun Instagram Gizila sudah sesuai dengan yang diinginkan followers Gizila.	4	9	1	0
4	Akun Instagram Gizila memberikan akses yang lebih mudah dalam mencari informasi kebutuhan gizi harian.	5	9	0	0
<b>Efficiency (Efisiensi)</b>					
1	Tampilan informasi yang diberikan akun instagram Gizila mudah dipahami dan mudah dimengerti.	4	10	0	0
2	Akun instagram Gizila dapat mudah diakses oleh semua kalangan masyarakat.	4	10	0	0
3	Konten spesifik seperti informasi gizi, tips harian, resep, atau kebutuhan kalori di akun Gizila dapat mudah ditemukan.	4	10	0	0
4	Informasi yang disajikan akun instagram Gizila meringankan pengguna dari segi biaya dan waktu.	4	10	0	0

<b>Service (Pelayanan)</b>		<b>SP</b>	<b>P</b>	<b>TP</b>	<b>STP</b>
1	Setiap informasi pada akun instagram Gizila dapat diakses dengan mudah.	3	11	0	0
2	Setiap panduan informasi mengenai tips gizi harian yang diberikan pada akun instagram Gizila dapat dipahami dan diikuti dengan baik.	3	11	0	0
3	Pengguna dapat melihat informasi tips gizi harian secara akurat.	4	10	0	0
4	Pengguna dapat melihat informasi tips gizi harian secara <i>up-to-date</i> .	3	10	1	0
<b>User Satisfaction (Kepuasan Pengguna)</b>		<b>SP</b>	<b>P</b>	<b>TP</b>	<b>STP</b>
1	Pengguna menyukai tampilan yang disajikan dalam akun instagram Gizila.	3	11	0	0
2	Pengguna akan merekomendasikan akun instagram Gizila.	4	10	0	0
3	Pengguna puas dengan informasi yang diberikan akun instagram Gizila.	3	11	0	0
4	Pengguna selalu mengakses informasi gizi yang telah disajikan Gizila.	3	11	0	0