

**RANCANG BANGUN SISTEM PEMBELAJARAN BEREMPATI  
DI MEDIA SOSIAL BERBASIS *LEARNING MANAGEMENT SYSTEM*  
MENGUNAKAN *FRAMEWORK RUBY ON RAILS***

Skripsi Ini Diajukan Sebagai Salah Satu Syarat  
Mendapatkan Gelar Sarjana Komputer (S.Kom)



Oleh:

**ABDUL HAKIM GHANIY**  
**NIM: SIM17030020**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS TEKNIK DAN ILMU KOMPUTER  
UNIVERSITAS NAHDLATUL ULAMA INDONESIA  
JAKARTA  
2021**

## LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Abdul Hakim Ghaniy  
NIM : SIM17030020  
Program Studi : Sistem Informasi  
Fakultas : Teknik dan Ilmu Komputer  
Jenjang : Strata 1 (S1)

Menyatakan bahwa skripsi yang saya buat dengan judul:

### **RANCANG BANGUN SISTEM PEMBELAJARAN BEREMPATI DI MEDIA SOSIAL MENGGUNAKAN *FRAMEWORK RUBY ON RAILS***

1. Merupakan hasil karya tulis yang saya buat sendiri, tidak pernah diajukan untuk penelitian sebelumnya, tidak pernah diajukan untuk memperoleh gelar akademik oleh pihak lain, dan bukan merupakan hasil plagiat.
2. Saya diijinkan untuk mengerjakan penelitian ini oleh UNIVERSITAS NAHDLATUL ULAMA INDONESIA sesuai dengan norma-norma agama dan norma-norma hukum yang berlaku di negara Republik Indonesia.

Pernyataan ini saya buat dengan sejujurnya, penuh kesadaran dan tanggung jawab. Apabila pada kemudian hari terdapat ketidaksesuaian, maka saya siap untuk menerima konsekuensi sesuai aturan yang berlaku.

Jakarta, 1 Desember 2021



Abdul Hakim Ghaniy  
SIM17030020

## LEMBAR PERSETUJUAN

Nama : Abdul Hakim Ghaniy  
NIM : SIM17030020  
Program Studi : Sistem Informasi  
Fakultas : Teknik dan Ilmu Komputer  
Jenjang : Strata 1 (S1)  
Judul Skripsi : **RANCANG BANGUN SISTEM  
PEMBELAJARAN BEREMPATI DI MEDIA  
SOSIAL MENGGUNAKAN *FRAMEWORK RUBY  
ON RAILS***

Skripsi ini telah diperiksa dan disetujui

Jakarta, 1 Desember 2021



**Irchan Ali, S.Kom., M.Kom.**

Pembimbing

Mengetahui,



**Handy Fernandy, S.T., MMSI**

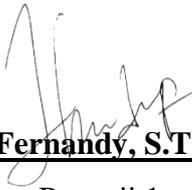
Kaprodi Sistem Informasi

## LEMBAR PENGESAHAN

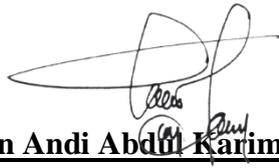
Nama : Abdul Hakim Ghaniy  
NIM : SIM17030020  
Program Studi : Sistem Informasi  
Fakultas : Teknik dan Ilmu Komputer  
Jenjang : Strata 1 (S1)  
Judul Skripsi : **RANCANG BANGUN SISTEM  
PEMBELAJARAN BEREMPATI DI MEDIA  
SOSIAL MENGGUNAKAN *FRAMEWORK RUBY  
ON RAILS***

Skripsi ini telah dipertahankan dihadapan para dewan penguji sidang skripsi Fakultas Teknik dan Ilmu Komputer, program studi Sistem Informasi dan dinyatakan LULUS.

Jakarta, 2 Desember 2021

  
**Handy Fernandy, S.T., MMSI**

Penguji 1

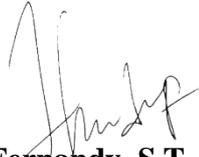
  
**Arifin Andi Abdul Karim, MMSI**

Penguji 2

  
**Ircham Ali, S.Kom., M.Kom.**

Pembimbing

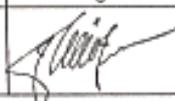
Mengetahui,

  
**Handy Fernandy, S.T., MMSI**

Kaprodi Sistem Informasi

## LEMBAR BIMBINGAN

**Nama** : Abdul Hakim Ghaniy  
Rancang Bangun Sistem Pembelajaran  
**Judul** : Berempati di Media Sosial Berbasis Learning  
Management System Menggunakan  
Framework Ruby on Rails  
**Pembimbing** : Irham Ali, S.Kom., M.Kom.

Tanggal	Perbaikan	Paraf Pembimbing
20/11/2021	Menentukan Tema dan Judul skripsi	
22/11/2021	Penulisan BAB 1 dan Revisi terkait sistematika penulisan pada bagian latar belakang	
23/11/2021	Penulisan BAB 2 pada bagian Kajian Terdahulu dan Kerangka Perancangan Sistem dan Revisi terkait sistematika penulisan	
25/11/2021	Penulisan BAB 3 secara keseluruhan dan Revisi terkait sistematika penulisan	
29/11/2021	Review penulisan BAB 3 dan Penulisan pada BAB 4 serta tinjauan terkait hasil penelitian	
30/11/2021	Review penulisan BAB 4 dan Penulisan pada BAB 5	
1/12/2021	Review keseluruhan penulisan dan perbaikan beberapa salah ketik pada penulisan skripsi	

Pembimbing,



Irham Ali, S.Kom., M.Kom

## **KATA PENGANTAR**

Segala puji bagi Allah, Tuhan Semesta Alam. Yang telah memberikan kepada kita nikmat Iman dan Islam, serta sehat wal ‘afiat, sehingga kita semua masih dapat hidup di dunia ini untuk berserah diri padanya dan bekerja serta menuntut ilmu yang telah Allah luaskan untuk kita semua. Sholawat serta Salam tak lupa selalu kita curahkan untuk Baginda Nabi Muhammad SAW, yang telah membawa umat manusia dari zaman kegelapan, hingga zaman yang terang benderang dengan ilmu pengetahuan seperti saat ini.

Izinkan saya sebagai penulis menuliskan beribu ucapan terimakasih kepada kedua orangtua saya, dan seluruh pihak yang telah membantu saya, baik doa, ide-ide cemerlang, serta kekuatan moril dan materil. Tanpa itu semua, mungkin skripsi ini jauh dari kata selesai. Maka dari itu, dengan rasa hormat, penulis sampaikan ucapan terimakasih kepada:

1. Allah SWT, yang telah memberikan saya kesehatan dan ilmu sehingga saya dapat menyelesaikan skripsi ini tepat pada waktu yang ditentukan
2. Kedua orangtua saya, Ibu Rumiasih dan Bapak Nur Wahyudin, yang selalu mendengar keluh kesah saya, mendoakan saya, dan memberikan segala dukungan untuk kelancaran semasa perkuliahan
3. Ibu Dr. Charyna Ayu R., S.Psi., M.A. yang telah mengizinkan saya menggunakan projek pengembangan aplikasi Modul Digital Berempati di Media Sosial ini sebagai tema penelitian Rancang Bangun Sistem, serta membantu saya dalam pemilihan jurnal-jurnal tentang Psikologi dan Empati

4. Ibu Adrinoviarini, M.Sc selaku Dekan Fakultas Teknik dan Ilmu Komputer Universitas Nahdlatul Ulama Indonesia
5. Bapak Handy Fernandy, S.T., MMSI selaku Kepala Program Studi Sistem Informasi Universitas Nahdlatul Ulama Indonesia yang telah membantu saya dalam proses pelengkapan administrasi ujian akhir skripsi
6. Bapak Ircham Ali, S.Kom., M.Kom. selaku Dosen Pembimbing saya yang telah senantiasa membantu terus menerus dari sisi penulisan, ide-ide, dan masukan yang sangat bermanfaat bagi penulisan skripsi saya
7. Keluarga Husein Family dan Asmaredja Family yang selalu memberikan semangat dan doanya
8. Rekan kerja STIEBI yang telah memberikan support
9. Kawan-kawan Fakultas Teknik dan Ilmu Komputer, terutama teman-teman dari Angkatan Tahun 2017 yang selalu memberikan semangat
10. Mega Apriyani Tarwiyah, S.Pd yang selalu memberikan doa, semangat dan selalu mendengar curhatan saya
11. Serta semua pihak yang tidak dapat saya sebutkan satu persatu, namun tidak mengurangi rasa takzim dan terimakasih saya kepada mereka semua

Semoga Allah SWT membalas segala kebaikan seluruh pihak yang telah membantu dengan ganjaran yang berlimpah, Aamiin, Yaa Robbal ‘Aalamin.

Jakarta, 1 Desember 2021

Penulis

## ABSTRAK

Masifnya perkembangan internet dan pemanfaatan teknologi mendukung terjadinya percepatan pertukaran informasi, baik untuk kebutuhan bisnis, pendidikan, maupun media sosial. Berdasarkan survey APJII, Indonesia menjadi salah satu negara yang mengalami percepatan pertumbuhan internet. Namun, masifnya penggunaan internet juga memberikan dampak pada meningkatnya kasus perundungan maya, ujaran kebencian, *hoax*, kekerasan verbal dan perundungan yang terjadi semasa sekolah. Permasalahan ini semakin diperparah dengan adanya media sosial yang dapat menjadi sarana untuk para pelaku perundungan melakukan hal tersebut secara anonim. Hal ini adalah salah satu bukti rendahnya tingkat empati yang dimiliki oleh para remaja dan pengguna media sosial.

Penelitian ini menggunakan metode kualitatif untuk merancang bangun sebuah sistem pembelajaran tentang berempati di media sosial dengan *framework Ruby on Rails* sebagai kerangka kerja sistem, metode pengembangan *Rapid Application Development*, dan metode pengujian menggunakan teknik *Equivalent Partitioning* pada *Blackbox Testing*. Penelitian ini bekerjasama dengan Komunitas Empati untuk memberikan materi-materi terstruktur dan penyampaian secara detail tentang empati yang dikemas secara interaktif menggunakan konsep LMS.

Sistem pembelajaran ini diharapkan menjadi salah satu oasis bagi anak-anak, remaja, dan pengguna media sosial untuk memahami bagaimana pentingnya mengamalkan nilai-nilai empati, baik pada kehidupan nyata maupun dalam bermedia sosial dengan harapan terciptanya kehidupan harmonis dan saling memahami sesama manusia.

Kata kunci: Empati, *Learning Management System*, *Ruby on Rails*.

## DAFTAR ISI

<b>LEMBAR PERNYATAAN</b> .....	<b>i</b>
<b>LEMBAR PERSETUJUAN</b> .....	<b>ii</b>
<b>LEMBAR PENGESAHAN</b> .....	<b>iii</b>
<b>LEMBAR BIMBINGAN</b> .....	<b>iv</b>
<b>KATA PENGANTAR</b> .....	<b>v</b>
<b>ABSTRAK</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>viii</b>
<b>DAFTAR GAMBAR</b> .....	<b>x</b>
<b>DAFTAR TABEL</b> .....	<b>xii</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang Penelitian.....	1
1.2 Rumusan Penelitian .....	4
1.3 Batasan Masalah.....	4
1.4 Pertanyaan Penelitian .....	4
1.5 Tujuan Penelitian.....	5
1.6 Manfaat Penelitian.....	5
1.7 Sistematika Penulisan.....	6
<b>BAB II KAJIAN TEORI</b> .....	<b>8</b>
2.1 Deskripsi Teori .....	8
2.1.1 Rancang Bangun .....	8
2.1.2 Empati .....	8
2.1.3 <i>Learning Management System</i> .....	9
2.1.4 Bahasa Pemrograman Ruby .....	10
2.1.5 <i>Framework Ruby on Rails</i> .....	12
2.1.6 PostgreSQL .....	23
2.1.7 Figma.....	25
2.2 Kerangka Perancangan Sistem .....	26
2.3 Tinjauan Penelitian Terdahulu .....	27
<b>BAB III METODOLOGI PENELITIAN</b> .....	<b>31</b>
3.1 Metode Penelitian.....	31

3.2	Metode Pengembangan Sistem.....	32
3.3	Metode Perancangan Sistem.....	34
3.3.1	<i>UseCase Diagram</i> .....	35
3.3.2	<i>Activity Diagram</i> .....	36
3.3.3	<i>Sequence Diagram</i> .....	43
3.3.4	<i>Class Diagram</i> .....	46
3.4	Metode Pengujian Sistem.....	51
3.5	Prosedur dan Teknik Pengujian Sistem.....	51
3.6	Waktu dan Lokasi Penelitian.....	52
3.7	Bahan Penelitian.....	52
3.8	Peralatan Penelitian .....	52
<b>BAB IV HASIL DAN PEMBAHASAN PENELITIAN.....</b>		<b>54</b>
4.1	Hasil Penelitian.....	54
4.1.1	Implementasi Halaman Beranda .....	54
4.1.2	Implementasi Halaman Login .....	55
4.1.3	Implementasi Halaman <i>Forgot Password</i> .....	57
4.1.4	Implementasi Halaman <i>Form Register</i> .....	57
4.1.5	Implementasi Halaman <i>Course</i> .....	59
4.1.6	Implementasi Halaman Kuis .....	61
4.2	Pembahasan .....	62
4.2.1	Pengujian Fitur <i>Login</i> .....	63
4.2.2	Pengujian Fitur <i>Register</i> .....	65
4.2.3	Pengujian Fitur <i>Forgot Password</i> .....	68
4.2.4	Pengujian <i>Fetch API</i> .....	70
<b>BAB V PENUTUP .....</b>		<b>77</b>
5.1	Kesimpulan.....	77
5.2	Saran .....	78
<b>DAFTAR PUSTAKA .....</b>		<b>79</b>
<b>LAMPIRAN.....</b>		<b>84</b>

## DAFTAR GAMBAR

Gambar 2.1 OOP Natural pada Ruby.....	11
Gambar 2.2 Teknik <i>Monkey Patching</i> Pada Bahasa Ruby .....	12
Gambar 2.3 Penjelasan Konsep MVC Pada <i>Ruby on Rails</i> .....	14
Gambar 2.4 Konsep Webpack Via <a href="http://webpack.js.org">webpack.js.org</a> .....	19
Gambar 2.5 Kerangka Perancangan Sistem.....	26
Gambar 3.1 RAD <i>Flow</i> via codebots .....	32
Gambar 3.2 <i>UseCase Diagram</i> Pada LMS Empati.....	35
Gambar 3.3 <i>Activity Diagram</i> Pada <i>Session Process</i> .....	36
Gambar 3.4 <i>Activity Diagram</i> Pada <i>Registration Process</i> .....	37
Gambar 3.5 <i>Activity Diagram</i> Pada <i>Course Process</i> .....	38
Gambar 3.6 Penjelasan Validasi <i>Course</i> Menggunakan Program Ruby.....	40
Gambar 3.7 <i>Activity Diagram</i> Pada <i>Quiz Process</i> .....	42
Gambar 3.8 <i>Sequence Diagram</i> Pada <i>Auth Process</i> .....	43
Gambar 3.9 <i>Sequence Diagram</i> Pada <i>Course Process</i> .....	45
Gambar 3.10 <i>Class Diagram</i> Pada LMS Berempati di Media Sosial.....	47
Gambar 3.11 Penetapan <i>helper method associations</i> .....	50
Gambar 4.1 Halaman Beranda .....	54
Gambar 4.2 Halaman Rujukan: Buku Rujukan Tambahan.....	55
Gambar 4.3 Halaman Rujukan: Buku Rujukan Utama.....	55
Gambar 4.4 Halaman Login.....	56
Gambar 4.5 Halaman <i>Forgot Password</i> .....	57
Gambar 4.6 Halaman <i>Form Register</i> .....	57

Gambar 4.7 Halaman <i>Course</i> .....	59
Gambar 4.8 <i>Play button</i> dan <i>to previous course button</i> .....	59
Gambar 4.9 <i>Replay button</i> dan <i>to previous – to next course</i> .....	60
Gambar 4.10 <i>Pause button</i> pada video .....	60
Gambar 4.11 Halaman Kuis dan Contoh Pertanyaan.....	61
Gambar 4.12 <i>Popup</i> Jawaban Benar .....	61
Gambar 4.13 <i>Popup</i> Jawaban Salah.....	62

## DAFTAR TABEL

Tabel 3.1 Atribut pada Entitas <i>Course</i> .....	49
Tabel 3.2 Atribut pada Entitas <i>User</i> .....	49
Tabel 3.3 Atribut pada Entitas <i>UserCourse</i> .....	49
Tabel 3.4 Lokasi Penelitian.....	52
Tabel 4.1 Pengujian Validasi Pada Sistem <i>Login</i> .....	63
Tabel 4.2 Pengujian Integritas Data User Pada Sistem <i>Login</i> .....	64
Tabel 4.3 Pengujian Validasi Pada Sistem <i>Register</i> .....	66
Tabel 4.4 Pengujian Integritas Data <i>User</i> Pada Sistem <i>Register</i> .....	67
Tabel 4.5 Pengujian Validasi Pada Sistem <i>Forgot Password</i> .....	69
Tabel 4.6 Pengujian Integritas Data User Pada Sistem <i>Forgot Password</i> .....	69
Tabel 4.7 Pengujian Integritas Data User via API .....	70
Tabel 4.8 Pengujian Integritas Data <i>Course</i> via API.....	71
Tabel 4.9 Pengujian Integritas Data <i>UserCourse</i> via API .....	72
Tabel 4.10 Pengujian Validasi <i>UserCourse</i> via API.....	74

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang Penelitian

Perkembangan teknologi yang didukung akses internet telah mempercepat pertukaran informasi dan digitalisasi pada semua aspek kehidupan. Survey yang dilakukan APJII menunjukkan bahwa Indonesia telah mengalami percepatan yang sangat signifikan pada kuartal 2019-2020 sebagai salah satu pengguna internet terbesar di Dunia (APJII, 2020).

Masifnya perkembangan internet dan percepatan pertukaran informasi berdampak pada meningkatnya kasus perundungan maya (*cyberbullying*), *hoax*, tingginya ujaran kebencian, kekerasan verbal atau berkomentar secara tidak pantas, baik pada anak-anak, remaja, maupun orang dewasa yang dilakukan di *email*, *chat room*, *website* dan media sosial (Hertz dkk., 2015). Dampak lain dari perundungan maya yang dilakukan oleh pengguna media sosial juga berdampak pada kehidupan seseorang, khususnya para pengguna media sosial sering menyembunyikan identitas diri yang sebenarnya ketika melakukan hal tersebut. Dampaknya, korban perundungan tersebut akan sangat terluka hati karena tidak terhindarkan dari serangan *online* (Smith dkk., 2008)

Kasus-kasus tersebut juga masif terjadi di Indonesia, riset menunjukkan bahwa 84% anak-anak yang ditemukan oleh *The Plan International* dan *International Center for Research on Women* (ICRW) mengalami perundungan, termasuk kekerasan semasa sekolah. Yayasan Semai Jiwa Amini sebagai salah satu Lembaga Swadaya Masyarakat menemukan 67,9% siswa

pada Sekolah Menengah Atas (SMA) dan 66,1% siswa pada Sekolah Menengah Pertama (SMP) diberbagai kota besar seperti Jakarta telah mengalami kasus perundungan (Rizkyanti dkk., 2021).

Hal ini menunjukkan bahwa masih rendahnya tingkat empati yang dimiliki oleh anak-anak dan remaja. Karena empati merupakan suatu kemampuan untuk merasakan apa yang dirasakan orang lain serta mencoba penyelesaian masalah dengan pendekatan emosional, tanpa memberikan penilaian negatif atau iba terhadap orang lain, seakan-akan merasakan perasaan yang sama terhadap diri sendiri (Solekhah dkk., 2018).

Berdasarkan permasalahan yang terjadi saat ini, seperti rendahnya tingkat empati, meningkatnya perundungan maya, masifnya penyebaran *hoax* dan kekerasan verbal maka dibutuhkan sebuah sistem pembelajaran digital terstruktur yang dapat memberikan penjelasan tentang berempati, baik dalam kehidupan nyata maupun dalam bermedia sosial yang dibangun berbasis *Learning Management System (LMS)*. LMS merupakan sebuah *platform* yang mendukung model pembelajaran terstruktur sehingga materi akan tersampaikan dengan baik.

Penelitian terdahulu yang menerapkan LMS sebagai media pembelajaran digital diantaranya penerapan LMS Moodle sebagai media pengembangan pembelajaran digital pada matakuliah Anatomi Fisiologi Manusia (Azis, 2017), perancangan LMS menggunakan *Framework CodeIgniter* pada PT. Rekayasa Industri (Sumarna dkk., 2020), dan perancangan LMS sebagai pendukung Pembelajaran Jarak Jauh (PJJ) pada

Universitas Atma Jaya Makasar (Chyan, 2021). Secara umum diantara penelitian terdahulu tersebut masih terdapat kekurangan seperti penggunaan modul *opensource* Moodle, penerapan *Waterfall* sebagai metode pengembangan sistem, dan penggunaan *Framework CodeIgniter*. Basis pengembangan sistem secara keseluruhan penelitian terkait masih menggunakan bahasa pemrograman PHP sebagai *codebase* dari sistem yang berjalan.

Berdasarkan tinjauan dari penelitian terdahulu maka dalam penelitian ini mengusulkan penerapan bahasa pemrograman Ruby dengan *framework* Rails untuk pengembangan sistem, serta menerapkan metode pengembangan *Rapid Application Development* (RAD). Hal ini dikarenakan bahasa pemrograman Ruby mudah dipahami dan dapat diintegrasikan dengan *framework* Rails yang menggunakan asas *convention over configuration*. Metode RAD digunakan agar dapat meningkatkan kecepatan proses pengembangan aplikasi karena setiap iterasi yang dilakukan akan selalu mendapat masukan dari klien sehingga meminimalisir *bug* dan cacat pada waktu rilis.

Berdasarkan permasalahan yang ada dan berdasarkan tinjauan penelitian terdahulu, penelitian ini mengambil judul “Rancang Bangun Sistem Pembelajaran Berempati di Media Sosial Berbasis *Learning Management System* Menggunakan *Framework Ruby on Rails*” dengan tujuan sebagai media pembelajaran terstruktur untuk menyampaikan nilai-nilai empati yang kian terkikis seiring berkembangnya teknologi dan masifnya penggunaan internet di

Indonesia. Beberapa fitur yang akan diusulkan dalam pengembangan sistem digital ini yaitu menggunakan video animasi interaktif dalam penyampaian materi-materi yang diajarkan, serta pertanyaan berbasis kuis yang dapat dijawab oleh pengguna.

## 1.2 Rumusan Penelitian

Berdasarkan latar belakang dari permasalahan tersebut, penelitian ini merancang bangun sebuah pembelajaran digital berbasis LMS yang diterapkan pada *framework Ruby on Rails* dan dapat memberikan penjelasan secara terstruktur dan interaktif yang dikemas dalam sebuah LMS.

## 1.3 Batasan Masalah

Untuk menghindari pembahasan yang meluas, maka perlu dibuat batasan masalah pada penelitian ini antara lain:

1. Penelitian ini tidak membangun sebuah *dashboard* dengan sistem admin untuk memperbaharui materi-materi pembelajaran secara grafis
2. Penelitian ini hanya fokus pada pengembangan sistem berbasis web

## 1.4 Pertanyaan Penelitian

Dari latar belakang dan rumusan penelitian tersebut, maka dapat diambil beberapa pertanyaan yaitu:

- a. Bagaimana merancang bangun sistem untuk pembelajaran digital berbasis LMS?
- b. Bagaimana menerapkan *Framework Ruby on Rails* pada sistem pembelajaran digital ini?

- c. Bagaimana menyampaikan nilai-nilai Empati pada anak-anak dan remaja secara terstruktur dalam sebuah modul pembelajaran yang dikemas dalam sebuah LMS?

### **1.5 Tujuan Penelitian**

Dari beberapa pertanyaan penelitian sebelumnya, maka dapat diambil beberapa tujuan yaitu:

- a. Merancang bangun sebuah sistem pembelajaran digital menggunakan metode pengembangan sistem RAD berbasis LMS.
- b. Penerapan *Ruby on Rails* sebagai kerangka kerja sistem dengan bahasa Ruby, basis data PostgreSQL, dan *web server* Nginx.
- c. Mengemas setiap materi yang diajarkan menggunakan video animasi interaktif serta kuis yang berkaitan dengan materi yang diajarkan.

### **1.6 Manfaat Penelitian**

Manfaat yang dapat diambil dari penelitian ini diharapkan mendapatkan dampak positif bagi semua pihak yang berkaitan, antara lain:

1. Bagi Universitas Nahdlatul Ulama Indonesia Jakarta

Universitas Nahdlatul Ulama Indonesia sebagai salah satu universitas di Jakarta dapat memberikan kontribusi dan turut serta dalam pengembangan karakter dan moral bangsa melalui pembuatan sistem pembelajaran digital empati, dan turut serta dalam penggabungan berbagai rumpun ilmu untuk kemajuan teknologi.

2. Anak-anak, remaja, mahasiswa dan pengguna Media Sosial

Diharapkan dengan dibangunnya sistem pembelajaran digital empati ini dapat mencegah degradasi moral dan empati yang semakin sering terjadi di era kemajuan teknologi dan masifnya penggunaan internet, serta memberikan pemahaman yang baik tentang empati dan penerapannya pada kehidupan nyata dan media sosial.

### 3. Bagi Penulis

Sebagai salah satu syarat kelulusan untuk mendapatkan gelar sarjana komputer, diharapkan bagi penulis selaku mahasiswa dapat memberikan kontribusi yang terbaik berdasarkan Tri Dharma Perguruan Tinggi, Pancasila, dan Agama, mencegah semakin terdegradasinya moral dan empati pada anak dan remaja, serta dapat menyelesaikan Skripsi ini dengan baik sehingga dapat mengamalkan ilmu yang telah diperoleh selama kuliah dan berguna bagi masa depan.

## 1.7 Sistematika Penulisan

Dalam penulisan skripsi dibutuhkan sistematika penulisan yang berisi langkah-langkah penyusunan laporan tugas akhir. Adapun sistematika penulisan yang digunakan adalah:

### **BAB I      PENDAHULUAN**

Pendahuluan berisi Latar Belakang Penelitian, Rumusan Penelitian, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian dan Sistematika Penulisan.

## **BAB II KAJIAN TEORI**

Pada Kajian Teori berisi Kajian Teori, Kerangka Perancangan Sistem dan Tinjauan Penelitian Terdahulu.

Kajian Teori membahas teori yang digunakan, kerangka perancangan sistem untuk memecahkan permasalahan umum pada suatu objek penelitian, sehingga tercipta sub-sub Analisa yang dapat digunakan untuk membuat suatu sistem, serta penelitian-penelitian terdahulu yang digunakan sebagai kaca perbandingan untuk penelitian yang sedang dikerjakan.

## **BAB III METODOLOGI PENELITIAN**

Pada Metodologi Penelitian, hal yang dibahas antara lain Metode Penelitian, Metode Pengembangan Sistem, Metode Perancangan Sistem, Metode Pengujian Sistem, Prosedur dan Teknik Pengujian, Waktu dan Lokasi Penelitian, Bahan Penelitian, dan Peralatan Penelitian.

## **BAB IV HASIL PENELITIAN**

Pada Hasil Penelitian, hal yang disampaikan berupa Hasil Penelitian dan Pembahasan

## **BAB V PENUTUP**

Pada bagian Penutup, terdapat Kesimpulan dan Saran

## **BAB II**

### **KAJIAN TEORI**

#### **2.1 Deskripsi Teori**

##### **2.1.1 Rancang Bangun**

Proses perancangan menggambarkan keterlibatan suatu entitas pada entitas lainnya, seperti pada Basis Data atau Sub-Process, serta mendapatkan gambaran berupa anggaran dan teknologi yang akan digunakan, sehingga aplikasi yang akan dibuat sesuai dengan kebutuhan. Maka dapat dibuat sebuah kesimpulan yakni rancang bangun adalah proses menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang sudah ada “sesuai dengan analisa kebutuhan”, baik keseluruhan ataupun Sebagian sistem (Nuraeni, 2021).

##### **2.1.2 Empati**

Empati, suatu hal yang sudah dimiliki oleh manusia sejak lahir, adalah suatu potensi yang digunakan untuk memberikan respon terhadap apa yang dipikirkan dan dirasakan orang lain (Rizkyanti dkk., 2021). Empati dapat dilihat dari mereka sejak lahir, walaupun begitu, empati harus tetap dibina oleh keluarga agar tetap berkembang sejak dini. Empati dapat menumbuhkan rasa kepedulian, seperti seorang anak yang membela korban ketika terjadi perundungan atau sekelompok orang yang mengumpulkan donasi untuk korban banjir. Mereka tidak merasakan hal yang sama, tapi mereka mengerti dan memahami bagaimana jika mereka merasakan hal tersebut. Karena itu, penting sekali untuk mengaktivasi potensi tersebut dengan mendidik anak-anak

dan remaja agar mampu beretika serta berempati, baik dalam kehidupan nyata atau dalam bermedia sosial, karena masa remaja adalah titik krusial bagi pengembangan empati (Rizkyanti, 2017). Salah satu cara untuk mengaktivasi hal tersebut adalah dengan penyampaian materi pada hal-hal yang disukai oleh anak-anak dan remaja, seperti penggunaan media sosial sebagai media kampanye ataupun pembuatan sistem pembelajaran digital terstruktur menggunakan konsep LMS yang menampilkan video animasi interaktif tentang empati.

### 2.1.3 *Learning Management System*

Sistem merupakan sekumpulan objek yang berelasi dan berinteraksi atau kumpulan dari bagian-bagian yang bekerjasama dan dirancang untuk mencapai satu tujuan (Al Fatta, 2020). Seperti proses *login* pada suatu sistem yang memiliki sub-proses berupa validasi email, apakah email tersebut berupa email sah atau tidak, lalu sub-proses pengecekan kesesuaian email dan kata sandi terdaftar. Jika email terdaftar di sistem dan menggunakan kata sandi yang tepat, maka proses *login* memberikan izin pengguna untuk mengakses sistem. Sedangkan jika email terdaftar di sistem namun kata sandi tidak tepat, proses *login* akan memberikan peringatan kata sandi salah dan tidak diizinkan masuk. Terakhir, jika email dan kata sandi pengguna tidak terdaftar, proses *login* akan memberikan peringatan untuk mendaftar terlebih dahulu di sistem *register*.

LMS merupakan suatu sistem manajemen bagi pembelajaran digital yang dikembangkan untuk mendukung model pembelajaran terstruktur. LMS memungkinkan untuk berkolaborasi antara pengajar dengan murid secara

interaktif seperti halnya pesan singkat (*chat*), kuis, serta dapat menampilkan video-video animasi atau video pembelajaran, dan tidak terbatas hanya pada materi dalam bentuk tekstual (Fitriani, 2020). Fleksibilitas LMS yang dapat diakses melalui berbagai macam gawai, memungkinkan proses penyampaian materi dapat diakses kapan saja, dimana saja dan oleh siapa saja (Yeti dan Ahyanuardi, 2020).

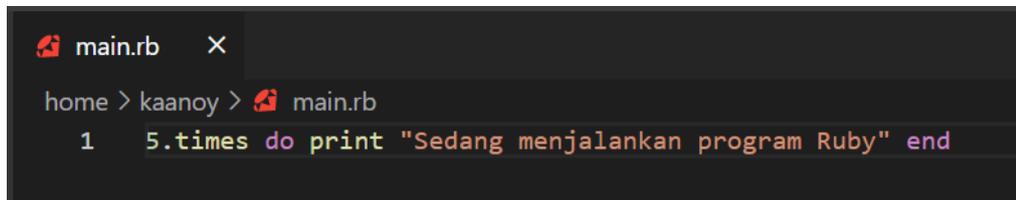
#### 2.1.4 Bahasa Pemrograman Ruby

Ruby adalah sebuah Bahasa Pemrograman Skrip Berorientasi Objek murni yang diciptakan oleh Yukihiro Matsumoto pada tahun 1995, mempunyai sintak sederhana dan diperuntukkan untuk kesenangan *programmer* (RubyLang, 2021).

Ruby termasuk dalam kategori *high-level programming language*, yaitu bahasa tingkat tinggi, dalam arti struktur bahasanya mirip dengan bahasa manusia. Selain itu, Ruby juga termasuk dalam *general-purpose language*, yaitu bahasa yang ditujukan untuk banyak aspek, seperti pengembangan aplikasi web, pengembangan aplikasi berbasis desktop, membuat otomatisasi pada berbagai macam *daemon* (program yang berjalan tanpa intervensi dari pengguna) di *server*, dan lain sebagainya.

Matsumoto, atau “Matz” menciptakan Ruby sebagai bahasa pemrograman yang natural, bukan sederhana. Dengan memperhatikan aspek yang membuatnya mirip seperti bahasa dalam kehidupan nyata, merujuk email Matz pada tahun 2000 di *mail-list Ruby-Talk* (Matsumoto, 2000) “Penampilan Ruby dari luar sangat sederhana, namun sangat kompleks di bagian dalam,

seperti badan manusia kita ini”. Pernyataan tersebut dapat dilihat dari contoh kode berikut:

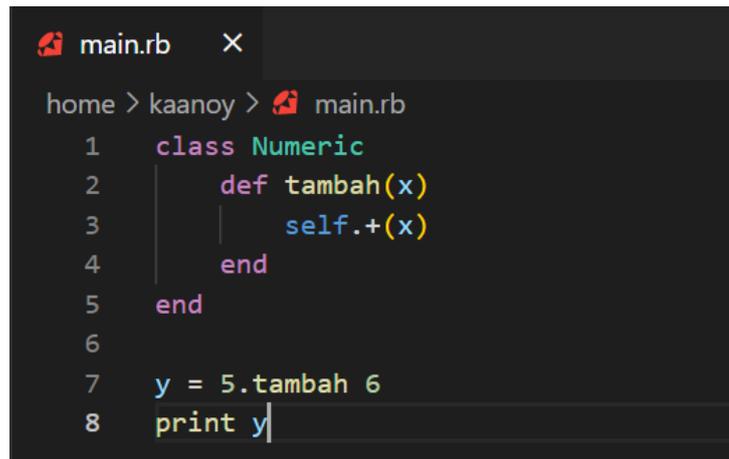


```
main.rb X
home > kaanoy > main.rb
1 5.times do print "Sedang menjalankan program Ruby" end
```

Gambar 2.1 OOP Natural pada Ruby

Kode tersebut akan menampilkan teks berupa “Sedang menjalankan Program Ruby” sebanyak lima kali. Pada bahasa pemrograman selain Ruby, angka (*integer*) atau tipe data primitif bukanlah sebuah objek, sehingga tidak dapat diberikan sebuah fungsi atau method yang mewakili tipe data tersebut. Umumnya, Bahasa selain Ruby terpaut dengan kurung kurawal dan harus menuliskan kurung buka-tutup pada setiap fungsi.

Contoh fleksibilitas lain yang membuat Ruby sengaja dibuat untuk kesenangan *programmer*, yaitu mudahnya mendefinisikan ulang suatu fungsi bawaan yang telah ditetapkan. Contoh dari kemudahan ini adalah dengan menerapkan sebuah teknik *monkey patching* pada fungsi penjumlahan sehingga menggunakan kata “**tambah**” alih-alih menggunakan operator aritmatika bawaan Ruby “+” dengan menambahkan *method* tersebut pada kelas *Numeric* bawaan Ruby.



```

main.rb x
home > kaanoy > main.rb
1  class Numeric
2      def tambah(x)
3          self.+(x)
4      end
5  end
6
7  y = 5.tambah 6
8  print y

```

Gambar 2.2 Teknik *Monkey Patching* Pada Bahasa Ruby

Ruby juga merupakan bahasa yang berlisensi *open source*, yaitu kode sumber yang terbuka untuk umum sehingga *programmer* diseluruh dunia dapat berkontribusi dalam pengembangannya. Bahasa ini menerapkan tipe sistem *Dynamic Typing*, dimana *programmer* tidak perlu mendeklarasikan tipe data setiap membuat variabel atau fungsi. Hal ini menjadi salah satu alasan mengapa program yang dibangun dengan bahasa Ruby akan sangat cepat pada proses *development* (Carlson dan Richardson, 2015). Ruby juga mendukung berbagai macam *plugin* yang dapat digunakan sebagai fungsionalitas tambahan agar program dapat dijalankan dengan baik. Salah satu *plugin* yang dimiliki dan paling banyak digunakan adalah Rails, sebuah *plugin* yang digunakan untuk membangun *framework* aplikasi berbasis web. Ruby menyebut *plugin* sebagai *Gem*.

### 2.1.5 *Framework Ruby on Rails*

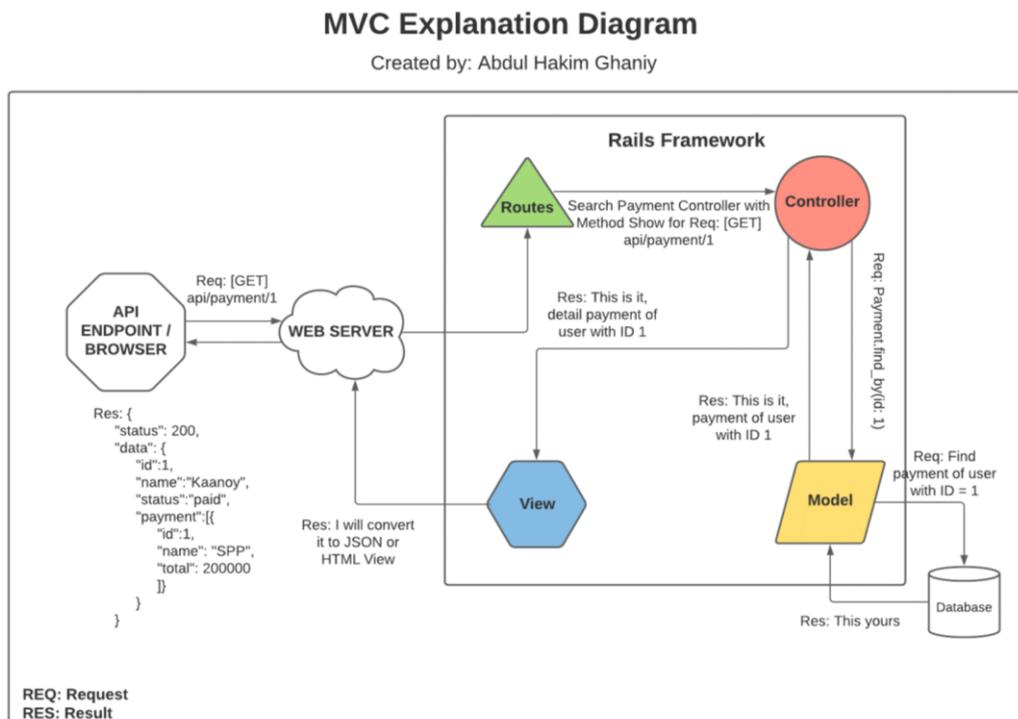
*Framework* atau Kerangka Kerja adalah suatu sistem yang dapat memudahkan *developer* untuk mengerjakan suatu proyek tanpa perlu membuat aplikasi dari nol. *Framework* berisi instruksi dasar yang dikemas dalam

*function* dan *class*. *Developer* hanya perlu memanggil fungsi-fungsi dasar tersebut tanpa perlu membuat kode instruksi yang sama berulang-ulang, sehingga dapat menghemat waktu dan biaya, serta memudahkan *developer* dalam proses *maintenance* (Destiningrum dan Adrian, 2017). Kemudahan yang ditawarkan *framework* menjadi salah satu pilihan dan pertimbangan sebagai langkah awal dalam pengembangan aplikasi, karena telah disediakan file konfigurasi, folder yang terstruktur, *service* atau *sub-process* yang saling terintegrasi, juga telah membuat komponen yang tepat dan saling bekerjasama (Stauffer, 2016). Sedangkan *Ruby on Rails* (atau kebanyakan *programmer* hanya menyebutnya sebagai *Rails*) adalah sebuah *framework* yang dibangun diatas Bahasa Ruby.

*Rails* merupakan *framework* yang berlisensi *opensource* dan menerapkan konsep arsitektur MVC (*Model, View, Controller*) pada *design pattern*-nya, dimana komponen-komponen utama tersebut saling bekerjasama untuk menampilkan dan memproses informasi yang diminta maupun dikirim oleh klien (Anisah dkk., 2015). Proses pada MVC dapat dijelaskan sebagai berikut:

1. Klien melalui peramban atau API (*Application Programming Interface*) mengirimkan *request* ke *server* pada suatu halaman dan ditangkap oleh *WebServer*.
2. *Webserver* mengirimkan data tersebut melalui *routes* yang telah ditentukan.

3. *Controller* melalui *routes* menanggapi *request* tersebut, lalu mengambil data yang diperlukan ke *Model*. *Routes* memilih *controller* dengan *action* yang tepat untuk melakukan pengiriman atau permintaan data.
4. *Model* mencari data di *Database Management System* (DBMS). Setelah mendapatkan hasil yang sesuai, *model* mengirimkan data ke *controller*, lalu *controller* me-render halaman dan mengirimkannya ke *View*. Hasil *render* dapat berupa kode HTML pada Web atau JSON (*Javascript Object Notation Response*) pada RestAPI.
5. *View* menampilkan halaman melalui peramban atau menampilkan *response* yang dikirim melalui *WebServer*.



Gambar 2.3 Penjelasan Konsep MVC Pada *Ruby on Rails*

Kelebihan dari konsep arsitektur MVC adalah pemisahan kode antara logika aplikasi (*Controller*), proses interaksi dari-dan-ke basis data (*Model*) serta proses interaksi ke pengguna (*View*). Sehingga sistem akan lebih mudah di-*maintenance* dan masing-masing divisi dapat mengembangkan proses terkait dengan mudah.

Tiga komponen di *Ruby on Rails* yang merepresentasikan dan bertugas dalam menjalankan fungsi MVC adalah sebagai berikut (Anisah dkk., 2015):

1. **ActionView:** komponen *View* bertugas menerima response dari pengguna kedalam sistem serta merepresentasikan data dari *Model* yang dikirim oleh *Controller*. *ActionView* adalah penamaan untuk komponen *View* di Rails.
2. **ActionController:** *controller* adalah tempat logika aplikasi berada. Komponen ini menjembatani *data flow* antara *View* dengan *Model* dengan menangkap semua *request* yang dikirim melalui *routes*. *ActionController* adalah penamaan untuk komponen *Controller* di Rails.
3. **ActiveRecord:** *model* bertugas mengambil data yang diperlukan ke basis data sesuai instruksi *controller*. Komponen ini dapat merubah kode Ruby yang dijalankan oleh *controller* menjadi perintah SQL yang dapat dimengerti oleh DBMS secara otomatis tanpa perlu campur tangan manusia. *ActiveRecord* adalah penamaan untuk komponen *Model* di Rails.

David Heinemer Hanson, sebagai pembuat dan pengembang *Rails Framework* menulis Sembilan Doktrin yang digunakan dan sebaiknya diikuti oleh para pengembang aplikasi berbasis *Rails*. Salah satu doktrin yang terkenal adalah *Convention Over Configuration* (Konvensi “lebih utama” dari Konfigurasi). David menjelaskan makna tersebut di website resmi *Ruby on Rails*:

Transfer Konfigurasi ke Konvensi tidak hanya membebaskan pertimbangan, tetapi juga menyediakan bidang yang lebih baik untuk menumbuhkan abstraksi mendalam. Jika kita dapat bergantung pada pemetaan “kelas *Person*” ke “table *people*”, kita dapat menggunakan infleksi-- Infleksi adalah **perubahan bentuk kata**, “terlebih pada akhir kata” sesuai dengan kaidah *grammatical* pada kalimat tersebut (Dictionary, 2021) --yang sama untuk memetakan asosiasi “pada relasi *database*” sebagai *has\_many :people* untuk mencari kelas *Person*. Kekuatan konvensi yang baik adalah mereka mencakup keseluruhan spektrum pengguna yang luas. Tidak mungkin jika kerangka kerja Anda hanyalah buku teks tebal dan aplikasi baru Anda hanyalah lembaran-lembaran kertas kosong. Dibutuhkan upaya besar untuk mencari tahu dimana dan bagaimana memulainya. Setengah dari perjuangan untuk memulai adalah “menemukan benang” untuk ditarik (Hanson, 2016).

Adapun teknologi utama yang digunakan bersamaan dengan Rails adalah sebagai berikut:

### 1. ERB

ERB atau *Embedded Ruby* adalah sebuah *plugin* yang dipasang secara *default* dengan *Rails*. ERB memungkinkan penulisan kode *Hypertext Markup Language* (HTML) bersamaan dengan kode Ruby. Kode Ruby yang disematkan dalam kode HTML biasanya digunakan untuk menjalankan logika; baik perulangan atau percabangan, ataupun hanya menampilkan *response* tampilan yang di-*render* dari *controller*.

### 2. TailwindCSS

*Cascading Style Sheet* (CSS) adalah berkas yang digunakan untuk membuat tampilan dari HTML menjadi lebih menarik. CSS dapat merubah *behaviour* serta *style* dari kode HTML menggunakan *selector* seperti *class*, *id* atau *tag* yang berada di HTML. *Property* pada CSS menentukan *style* yang diterapkan pada *selector* tertentu. HTML dan CSS bukanlah sebuah Bahasa Pemrograman (Pratama, 2017). Sedangkan Tailwind adalah salah satu CSS *framework* yang digunakan untuk mempermudah pendefinisian *style* pada tiap-tiap elemen HTML, hanya dengan menambahkan *style* tersebut pada atribut *class*.

### 3. JavaScript

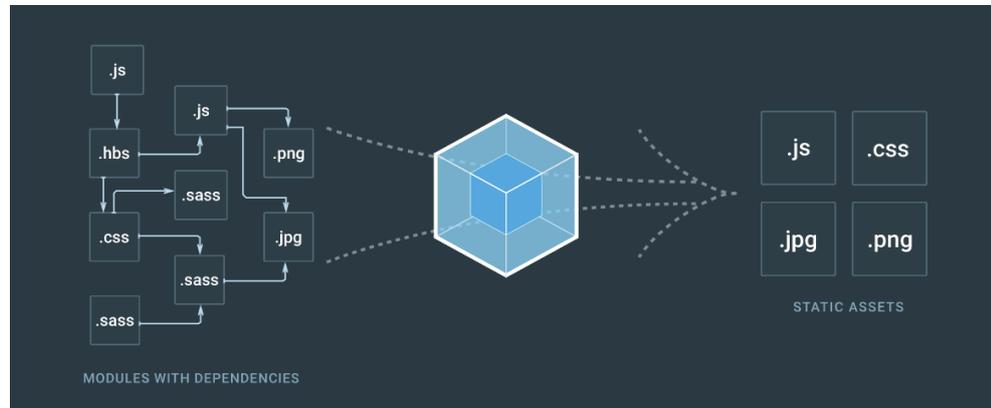
Javascript adalah sebuah bahasa pemrograman yang digunakan untuk memanipulasi HTML dan CSS sehingga menambahkan aspek “pemrograman” pada halaman *website* (Pratama, 2017). Contoh dari penggunaan Javascript sebagai *manipulator* HTML adalah fungsionalitas *Document Object Model* (DOM) yang merupakan API pada JavaScript untuk menambahkan tag *<li>* sesuai dengan data yang diperoleh dari fungsi Ajax, atau merubah *style* CSS dari *display: block;* menjadi *display: none;* ketika *eventlistener* pada *element* HTML di klik.

### 4. Webpack

*Webpack* adalah sebuah modul bundel statik untuk aplikasi yang dibangun dengan *Javascript* modern. *Webpack* membangun sebuah pohon grafik dari modul-modul yang dibutuhkan pada suatu projek aplikasi, sehingga menciptakan satu atau beberapa bundel kecil berkas berformat *.js*. Berkas yang bergantung dengan berkas lainnya akan diperlakukan oleh *webpack* sebagai modul ketergantungan (*dependency module*). Modul ketergantungan tersebut dapat berupa *font*, aset non-kode seperti berkas dengan format *.txt* atau berkas gambar (Webpack, 2021).

Namun tidak hanya merubah ketergantungan tersebut menjadi berkas dengan ekstensi *.js*, *plugin* yang disediakan oleh komunitas

*webpack* juga dapat merubah berkas *CSS Preprocessor* seperti SASS, LESS dan Stylus menjadi berkas CSS murni yang dapat



Gambar 2.4 Konsep Webpack Via [webpack.js.org](http://webpack.js.org)

dibaca oleh peramban. *Webpack* juga dapat digunakan pada *frontend framework* seperti ReactJS, VueJS atau sejenisnya.

##### 5. *Asynchronous JavaScript and XMLHttpRequest (AJAX)*

Ajax bukanlah suatu bahasa pemrograman atau sebuah *framework*, melainkan suatu mekanisme untuk mengirim dan menerima data dari-dan-ke server secara *asynchronous*. *Asynchronous* berarti data yang dikirim dan diterima tidak sampai me-*load* seluruh halaman, melainkan hanya mengganti sebagian elemen yang berubah. Hal ini berbeda dengan mekanisme web tradisional yang akan me-*load* seluruh halaman dan me-*refresh* jika terjadi perubahan data (Sunyoto, 2010).

AJAX digunakan untuk membuat pengalaman pengguna saat mengakses halaman web menjadi lebih *user-friendly*, tidak terganggu dengan menunggu halaman selesai dimuat ulang jika terjadi perubahan atau permintaan data baru. AJAX dapat

digunakan untuk melakukan *API Call* seperti melakukan perubahan status pada tiap-tiap materi pelajaran dari “*in-progress*” menjadi “*completed*” jika video materi telah selesai ditonton.

## 6. API

API adalah sebuah software yang memungkinkan komunikasi dua atau lebih sistem yang berbeda untuk mengintegrasikan data sehingga *developer* tidak membuat suatu fungsi yang sama secara berulang (Anwar dkk., 2017). Seperti komunikasi antara *driver* perangkat keras dengan *kernel* yang dimiliki oleh sistem operasi, atau komunikasi antara Bahasa Ruby dengan DBMS Postgresql menggunakan *library libpq-dev*, *ruby-pg* dan *gem pg*.

REST atau RESTful API merupakan implementasi dari API. REST (*Representational State Transfer*) adalah sebuah arsitektur komunikasi yang berjalan diatas HTTP *protocol*, sama seperti Web. Perbedaan mendasar antara RestAPI dengan Web adalah penerapan *cookies* dimana Web dapat menyimpan informasi terakhir pengguna, seperti status login, atau halaman terakhir dibaca. Sedangkan pada RestAPI tidak diperbolehkan adanya penyimpanan informasi apapun, selain dari data yang dikirim atau diterima menggunakan HTTP *methods*. Konsep ini disebut sebagai *Stateless*.

RESTful API mempunyai 4 komponen penting, yaitu URL (*Uniform Resource Locator*) Design, HTTP Verbs/Methods, HTTP Response Code, dan Format Response (Thoriqul Falahi, 2019).

- a. **URL Design**, adalah penamaan secara konsisten pada suatu URL sehingga menghasilkan API yang *reliable* dan mudah dimengerti oleh *developer*. Contoh penerapan URL design adalah penggunaan kata benda atau nama untuk menunjuk suatu resource yang dimiliki *server*, bukan “kata kerja”, seperti: *example.com/api/v1/student/1* untuk menampilkan data Siswa dengan ID 1, bukan *example.com/api/v1/get-students/id/1*.
- b. **HTTP Verbs**, merupakan *request method* yang dilakukan oleh klien terhadap *server* sehingga menampilkan *response* yang diinginkan. HTTP verbs yang sering digunakan antara lain *GET, POST, DELETE, PUT*.
  - **GET** merupakan *verb* yang digunakan untuk menampilkan *resources* di *server*.
  - **POST** merupakan *verb* yang digunakan untuk menyimpan *resources* dimana “sebelumnya” tidak ada di *server*. Seperti proses registrasi pengguna.
  - **PUT** merupakan *verb* yang digunakan untuk memperbarui lalu menyimpan *resources* yang “sudah tersimpan” di *server*. Seperti merubah email atau *username* pengguna.

- *DELETE* merupakan *verb* yang digunakan untuk menghapus suatu *resources* yang tersimpan di *server*.
- c. **HTTP Response Code**, merupakan suatu *identifier* yang digunakan sebagai informasi status *request* kepada klien. Terdapat lima kelompok *Response Code*, yaitu **1xx**, **2xx**, **3xx**, **4xx**, dan **5xx**.
- **1xx**, merupakan kelompok Tanggapan Informasi. Seperti 100 dan 101. Namun menurut standar HTTP/1.0, kode respon 1xx tidak boleh ditampilkan ke pengguna. Kelompok kode ini hanya dapat dibaca oleh mesin klien dan *server*.
  - **2xx**, merupakan kelompok Permintaan Berhasil “diterima-dipahami-diterima”, baik oleh *server* dan klien. Seperti kode 200 pada *method GET* merupakan *response* dari *resource* yang sesuai telah diterima dengan baik.
  - **3xx**, merupakan kelompok Pengalihan. Biasanya pengalihan dari suatu URL ke URL lain. Seperti kode 301 yang berarti URL telah dialihkan secara permanen dari *api/v1/payments* dialihkan ke *api/v2/payments* atau dari protokol HTTP ke HTTPS (*Hypertext Transfer Protocol Secure*).
  - **4xx dan 5xx** merupakan kelompok yang disebabkan adanya galat. 4xx mengindikasikan galat yang terjadi di sisi

klien, seperti kode 400 merupakan kesalahan parameter pada URL *Endpoint* atau 401 merupakan kesalahan otorisasi pengguna. Sedangkan 5xx merupakan galat yang terjadi di sisi *server*. Seperti kode 500 yang berarti *Internal Server Error*.

- d. **Format Response**, adalah format yang dikirim sebagai hasil dari permintaan klien. Pada Arsitektur REST, format yang dikirim berupa JSON.

## 7. Database

*Database* adalah kumpulan data sistematis yang disimpan pada komputer, dapat diperiksa oleh berbagai macam jenis program komputer sehingga menampilkan informasi dari data yang disimpan (Andaru, 2018). Database terbagi menjadi 2 jenis, SQL Database dan NoSQL Database. SQL Database menggunakan *query syntax* SQL untuk mengakses data. Sedangkan NoSQL menggunakan JSON. NoSQL merupakan akronim dari *Not Only SQL*.

### 2.1.6 PostgreSQL

PostgreSQL (dibaca sebagai *post-gress-Q-L*) adalah salah satu DBMS *opensource* yang telah lebih dari 30 tahun secara aktif dikembangkan dan digunakan serta telah mendapatkan reputasi baik dari segi *reliability*, integritas data, dan keakuratan data. PostgreSQL dibuat pada tahun 1986 oleh professor ilmu komputer di UCB, Michael Stonebraker dan tim nya (Lockhart, 2009).

PostgreSQL dapat berjalan lintas sistem operasi, seperti keluarga UNIX, keluarga Linux, Windows, dan MacOS serta dapat dikoneksikan dengan berbagai macam bahasa pemrograman seperti C/C++, Java, Perl, Python, Ruby, TCL, PHP, dan ODBC (*Open Database Connectivity*). Menurut Simon Riggs dalam bukunya "*PostgreSQL Administration Cookbook*" (Riggs dkk., 2017), PostgreSQL memiliki beberapa fitur utama yaitu:

- a. Sangat mengikut standar SQL sampai SQL:2011
- b. *Client-Server Architecture*, yaitu konsep arsitektur perangkat lunak yang menghubungkan antara Sistem Klien dan Sistem *Server* melalui jaringan komputer, atau komputer yang sama.
- c. Design yang sangat konkurensi, dimana proses penulisan dan pembacaan data akan dilakukan bersamaan dan tidak akan saling memblokir.
- d. Sangat mudah dikonfigurasi dan dengan mudah digunakan oleh berbagai macam aplikasi dan sistem operasi.
- e. Skalabilitas dan performa mumpuni dengan fitur penyetelan yang baik.
- f. Mendukung berbagai macam jenis model data seperti *relational*, *post-relational* (larik / *Array*), model dokumen seperti JSON / XML dan model *key:value*.

Masih pada buku yang sama, Simon Riggs menjelaskan beberapa kelebihan dari PostgreSQL yaitu:

- a. **Robustness**, yaitu kekokohan pada perangkat lunaknya, didukung dengan pengetesan otomatis pada fitur dan sistem konkurensinya. Seluruh transaksi data pada DBMS ini dilindungi oleh *Log Transaction* yang akan melakukan pemulihan otomatis jika terjadi kegagalan *software*.
- b. **Security**, yaitu PostgreSQL dapat dikontrol melalui aturan akses berbasis Host, secara *default* mendukung jaringan SSL terenkripsi, serta komunitas kuat yang akan melaporkan dan menangani berbagai macam bentuk *exploit* pada database.
- c. **SQL and NoSQL**, yaitu dukungan penuh dan sangat mengikuti aturan standar sintak SQL. PostgreSQL juga sangat memungkinkan untuk menggunakan *database* berbasis dokumen seperti PostgreSQL *text*, XML, dan berkas JSONB (JSON *binary*)

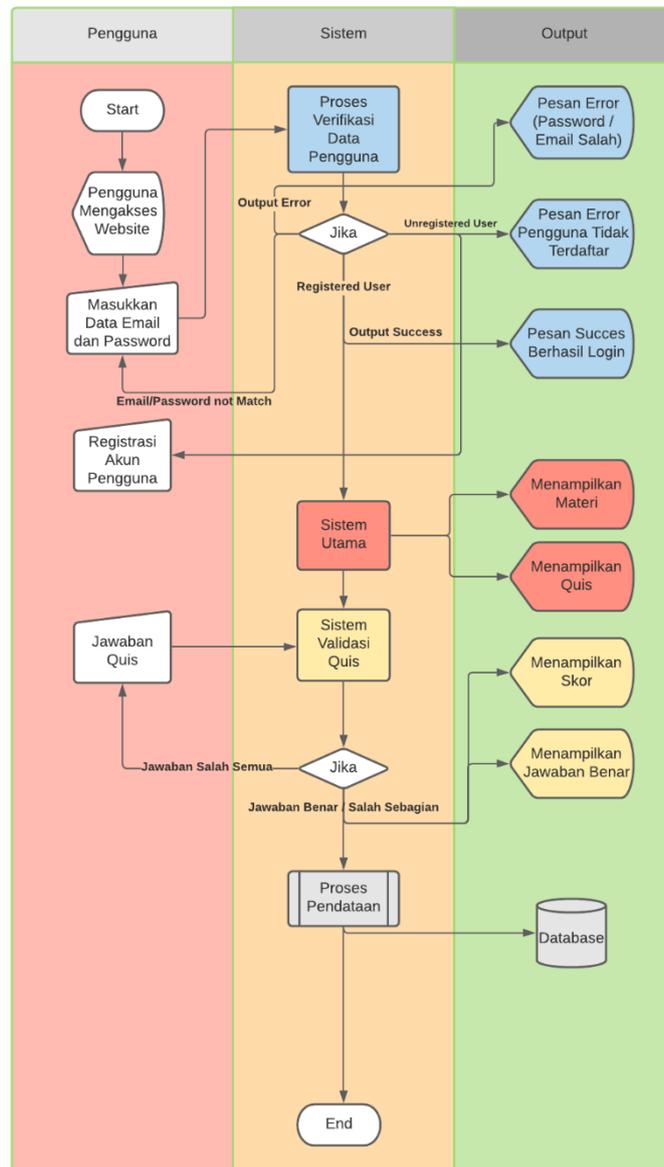
#### 2.1.7 Figma

Figma merupakan sebuah tools yang dapat digunakan untuk membuat desain *User Interface/User Experience* (UI/UX) berupa *wireframe* dan *mockup* pada suatu proyek perancangan aplikasi. Figma dapat berjalan diberbagai macam OS seperti Windows, Linux, Mac, bahkan peramban pada *Smartphone* (Muhyidin dkk., 2020).

## 2.2 Kerangka Perancangan Sistem

### Inovasi Modul Digital "Ayo Berempati di Media Sosial" Flow Diagram

Created By: Abdul Hakim Ghaniy



Gambar 2.5 Kerangka Perancangan Sistem

Pada *flowchart* berikut, terdapat tiga bagian utama, yaitu: Pengguna sebagai *input actor*, Sistem sebagai pemroses data, dan *Output* yang menampilkan *response* atau hasil dari data yang diproses. Pada bagian awal, pengguna mengakses halaman web dan memasukkan data email serta kata

sandi yang sebelumnya didaftarkan. Jika pengguna belum terdaftar, maka sistem akan menampilkan respon bahwa pengguna belum terdaftar, dan diarahkan untuk mendaftar terlebih dahulu. Jika pengguna salah memasukkan email atau kata sandi, maka sistem akan menampilkan respon bahwa masukan pengguna salah. Jika email dan kata sandi benar, maka akan diarahkan langsung ke halaman materi.

Pada halaman materi, akan ditampilkan berbagai macam video animasi yang berisi penjelasan-penjelasan tentang Empati, jika pengguna memulai video, JavaScript *event handler* akan menggunakan respon tersebut untuk mengaktifkan *asynchronous function* yang berguna untuk mendata status dari materi yang saat ini sedang ditonton ke database menggunakan *API Call*. Hal ini juga berlaku ketika pengguna telah sampai pada bagian kuis.

Pada bagian kuis, setiap jawaban akan berada pada status *wait* oleh *event handler*, setiap *event handler* ini akan digunakan untuk menampilkan *response* berupa hasil dari jawaban pengguna. Jika pengguna mencoba keluar sesi (*logout*) sebelum semua materi berakhir, maka pengguna akan diarahkan kembali ke materi terakhir sebelum keluar sesi.

### **2.3 Tinjauan Penelitian Terdahulu**

Beberapa penelitian terdahulu digunakan sebagai tolak ukur pengembangan penelitian saat ini serta digunakan untuk menghindari anggapan plagiasi. Dalam Kajian Teori, penelitian-penelitian tersebut dicantumkan dengan keterangan berupa penelitian yang dilakukan:

- A. Penelitian dengan tema penerapan LMS Moodle sebagai media pengembangan pembelajaran digital pada matakuliah Anatomi Fisiologi Manusia (Azis, 2017). Penelitian ini bertujuan untuk menghasilkan media pembelajaran digital yang mampu meningkatkan kualitas perkuliahan. Kekurangan dari penelitian ini terletak pada penggunaan LMS Moodle yang berupa aplikasi *opensource*, dimana penggunaan LMS ini banyak dibutuhkan penyesuaian disisi pengguna, sulit di-*customize* pada sisi aliran data dan desain tampilan antarmuka, serta banyaknya tampilan yang belum mendukung i18n (*internationalization*), sehingga beberapa tampilan masih menggunakan bahasa Inggris sebagai bahasa asalnya.
- B. Penelitian dengan tema perancangan LMS menggunakan *framework* CodeIgniter pada PT. Rekayasa Industri (Sumarna dkk., 2020). Penelitian ini menggunakan *framework* CodeIgniter sebagai kerangka kerja dari aplikasi yang dibuat. Penelitian ini bertujuan untuk meningkatkan kualitas pekerja pada perusahaan tersebut dengan memberikan pelatihan-pelatihan yang dapat diakses secara daring, serta dapat disesuaikan dengan gaya belajar masing-masing karyawan. Kelemahan dari penelitian ini terletak pada penggunaan kerangka kerja CodeIgniter yang tergolong dapat dengan mudah dirusak konsep MVC nya dengan menyisipkan perintah SQL dan PHP langsung pada *View*, atau mem-*bypass controller* langsung ke *model*. Jika hal ini terjadi, aplikasi akan sangat sulit untuk dilakukan *maintenance* dan pengembangan aplikasi selanjutnya akan terhambat, serta berpotensi menjadi *spaghetti code*. *Spaghetti Code* adalah keadaan dimana kode

sumber yang berada didalam aplikasi sudah tidak mungkin untuk diurai secara singkat.

- C. Penelitian dengan tema perancangan LMS sebagai pendukung PJJ pada Universitas Atma Jaya Makassar (Chyan, 2021). Penelitian ini menggunakan *SDLC (System Development Life Cycle) Waterfall Method* sebagai metode pengembangan aplikasinya yang ditujukan untuk mendukung Pendidikan Jarak Jauh di Universitas Atma Jaya Makassar sehingga akses pendidikan dapat tercapai dimana saja, kapan saja. Hal ini dilakukan demi tercapainya angka partisipasi nasional sebesar 50% pada tahun 2024, yang mana pada tahun 2018 tingkat partisipasi kasar baru tercapai 33,3%. Kekurangan dari penelitian ini adalah menggunakan metode *Waterfall*, dimana metode pengembangan ini sudah sangat lama. Pengembangan aplikasi menggunakan metode ini sangat *documentation-oriented* (pengembangan aplikasi harus berdasarkan dokumentasi yang sebelumnya sudah dibuat), sehingga jika terjadi fitur yang tidak sesuai keinginan klien ketika aplikasi selesai dibuat, dibutuhkan waktu yang lama untuk proses *rebuild*-nya.

Penelitian saat ini adalah membangun sebuah sistem pembelajaran digital berbasis LMS menggunakan *framework Ruby on Rails*. Hasil penelitian ini diharapkan dapat membantu seluruh anak-anak, remaja, mahasiswa dan seluruh pengguna media sosial untuk selalu menerapkan empati, baik dalam kehidupan nyata maupun dalam bermedia sosial. Karena empati adalah suatu

potensi yang sudah dimiliki oleh manusia sejak lahir dan harus tetap dijaga dan dikembangkan demi terwujudnya lingkungan yang harmonis.

Perbedaan penelitian saat ini dengan penelitian sebelumnya adalah menggunakan bahasa Ruby dengan *framework* Rails, dimana sintak bahasa Ruby sangat mudah dipahami. Fitur lainnya seperti *scaffolding* yang dapat mempercepat proses *development*, serta konsep *convention over configuration*, dimana penamaan pada tiap-tiap *class* di *Controller*, *Model*, *View*, dan pada *Migration File* seperti penamaan tabel serta *Helper Method* sudah ditetapkan sejak awal.

Konsep ini tidak menghilangkan fleksibilitas dalam penerapannya, konsep ini justru dapat mempermudah *developer* untuk memberikan penamaan, sehingga menghindari kemungkinan terjadinya *bug* karena penamaan yang tidak konsisten. Sistem ini juga menggunakan *database* PostgreSQL serta menggunakan metode RAD pada analisis dan perancangan aplikasi, sehingga tercipta aplikasi yang mumpuni dan sesuai kebutuhan karena mengandalkan *feedback* berulang dari klien, dan berdampak pada kecepatan proses pengembangannya.

## **BAB III**

### **METODOLOGI PENELITIAN**

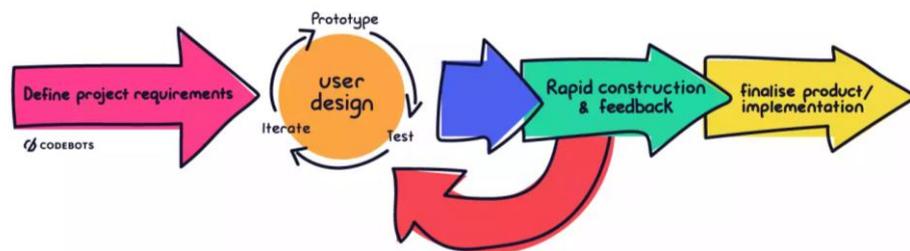
#### **3.1 Metode Penelitian**

Metodologi dapat didefinisikan sebagai “analisis pada prinsip-prinsip suatu metode, aturan-aturan, dan suatu asumsi postulat yang didasarkan pada suatu disiplin, atau studi sistematis tentang metode yang digunakan, mungkin, atau sudah digunakan pada suatu disiplin ilmu, atau prosedur tertentu atau seperangkat prosedur”. Metodologi Penelitian digunakan untuk menemukan, mengembangkan, dan menguji kebenaran dalam penelitian. Metodologi juga dapat mempertahankan penelitian dengan nilai ilmiah yang tinggi menggunakan syarat-syarat tertentu (Winarno, 2013).

Penelitian ini menggunakan metode kualitatif. Pada penelitian yang berlandaskan metode kualitatif, terdapat beberapa hal yang dapat diperhatikan seperti peneliti yang terlibat dalam interaksi dengan penelitian yang ditelitinya, sangat memperhatikan proses, peristiwa dan otentitas, Penelitian yang berlandaskan metode kualitatif juga cenderung berada dalam subjek yang sedikit, sehingga hal yang umum dilakukan adalah berkuat dengan analisa tematik. Tidak dapat dipungkiri, pada penelitian ini akan disajikan berbagai macam data mulai dari analisa, perancangan, pembuatan sistem hingga menghasilkan sebuah aplikasi.

### 3.2 Metode Pengembangan Sistem

Penelitian ini menggunakan metode pengembangan aplikasi *Rapid Application Development (RAD)*. RAD merupakan model pengembangan aplikasi yang menggunakan pengulangan-pengulangan secara linear, serta *feedback* dari klien secara berkala, sehingga dapat mempercepat proses pengembangan aplikasi serta meminimalisir “ketidakpuasan” klien terhadap aplikasi yang sudah rampung. Dengan metode RAD, proyek yang minimal dapat memakan waktu selama 180 hari dapat diselesaikan dengan waktu 30 sampai 90 hari, tergantung dari besar/kompleksitas aplikasi tersebut (Fadli, 2018).



Gambar 3.1 RAD Flow via codebots

Metode ini pertama kali dipaparkan oleh seorang konsultan teknologi dan penulis, James Martin pada tahun 1991. RAD terdiri dari 4 tahap, yaitu: mendefinisikan kebutuhan proyek, *prototyping*, *rapid construction and feedback gathering*, serta implementasi atau penyelesaian produk. Penjelasan masing-masing tahap adalah sebagai berikut:

#### A. Mendefinisikan Kebutuhan Proyek

Pada tahap ini seluruh *stakeholder* yang terdiri dari pengembang aplikasi, klien, dan *software user*, saling berkomunikasi untuk menentukan kebutuhan proyek, baik berupa *timeline*, anggaran, serta kebutuhan sistem dan alur bisnis yang akan direncanakan. Tahap ini sama seperti tahap yang digunakan pada model pengembangan aplikasi tradisional. Tahapannya singkat, namun sangat berimplikasi besar terhadap keberhasilan akhir proyek.

#### B. *Prototyping*

Tahap *prototyping* dapat dilaksanakan jika tahap pertama telah rampung. Tujuan utama tahap ini adalah menghasilkan *prototype* secara cepat yang akan didemonstrasikan pada klien. Mereka saling bekerjasama dengan klien hingga produk akhir siap dan sesuai permintaan. Sifat rilis yang cepat akan sangat berimplikasi pada kesalahan yang mungkin ditemukan lebih awal. Hal ini dapat mengarah pada pengurangan kesalahan dan *debugging*. Pada tahap ini juga, *developer* dapat dengan cepat menilai kompleksitas aplikasi sehingga aplikasi akan lebih terstruktur, lebih kuat, tidak rawan kesalahan, dan dapat dengan mudah menambahkan fitur baru dimasa mendatang.

#### C. *Rapid Construction and Feedback Gathering*

Pada tahap ini, pengkodean aplikasi, pengujian sistem, dan integrasi antar unit terjadi, mengubah *prototype* dan sistem beta kedalam model kerja. Karena seluruh perubahan dan masukan dari klien banyak terjadi di fase *prototyping*, hal ini mengakibatkan fase ke 3 dari RAD *cycle* menjadi lebih

cepat, walaupun tidak dapat dipungkiri klien meminta suatu fitur untuk diubah dan fitur tersebut masuk kembali ke fase *prototyping*. Selama fase ini, seluruh sistem diuji kembali secara menyeluruh untuk memastikan hasil akhir sesuai dengan harapan klien. Jika klien memberikan masukan positif saat pengujian menyeluruh maka aplikasi siap untuk memasuki tahap akhir.

#### D. Implementasi / Penyelesaian Produk

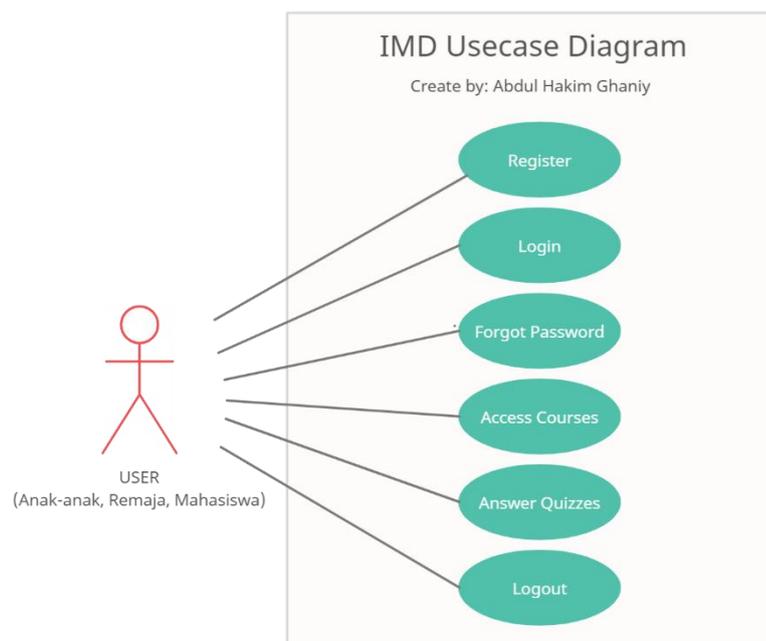
Pada fase akhir ini, para *developer* menyelesaikan beberapa “hutang teknis” (fitur yang belum sempurna) pada fase *prototyping*, atau permasalahan kecil lain akibat dari integrasi antar unit, serta meningkatkan stabilitas produk dengan memasukkan sistem tersebut ke *production environment* atau kedalam lingkungan pengujian penuh untuk mengidentifikasi *bug* produk. Pada fase ini juga, seluruh tim menulis dokumentasi menyeluruh serta menyelesaikan tugas pemeliharaan menyeluruh hingga akhirnya produk lengkap tersebut diserahkan ke klien.

### 3.3 Metode Perancangan Sistem

Kerangka perancangan sangat dibutuhkan ketika memulai suatu penelitian rancang bangun sistem. Adapun desain perancangan yang digunakan adalah *Unified Modeling Language* (UML). UML merupakan alat bantu untuk mengembangkan suatu sistem berorientasi objek, yang standar pemodelannya dikontrol oleh konsorsium terbuka *Object Management Group* (OMG), sebuah konsorsium yang terdiri dari berbagai perusahaan yang fokus dalam pengembangan standar-standar pendukung interoperabilitas (Mahdiana, 2011).

Penelitian ini menggunakan beberapa pemodelan UML seperti *UseCase Diagram*, *Activity Diagram*, *Sequence Diagram*, dan *Class Diagram*, adapun penjelasan kerangka perancangan sistem dari masing-masing diagram sebagai berikut:

### 3.3.1 *UseCase Diagram*



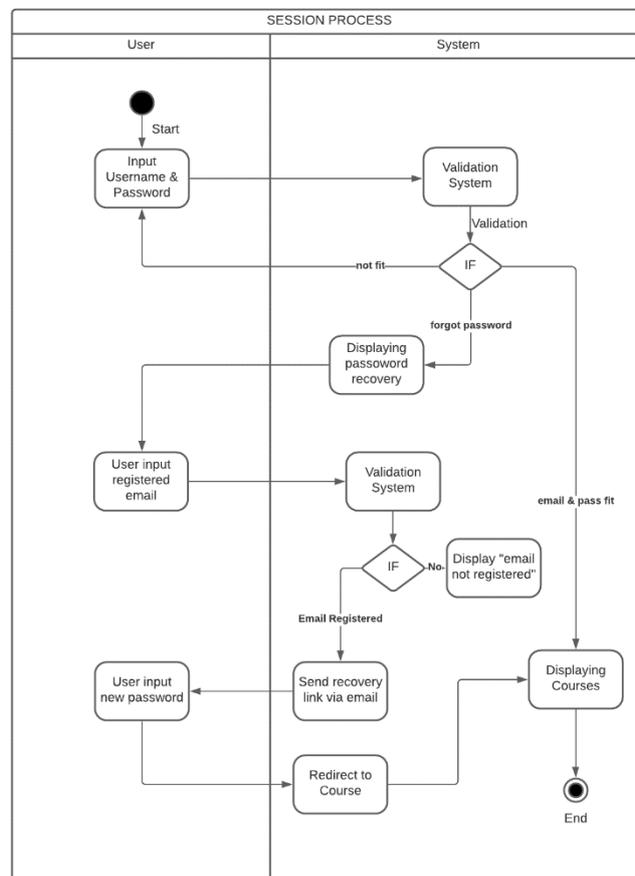
Gambar 3.2 *UseCase Diagram* Pada LMS Empati

Pada diagram ini dijelaskan seorang aktor yang dapat langsung berinteraksi dengan sistem dan memiliki akses pada proses *Register*, *Login*, *Forgot Password*, *Access Course*, *Answer Quizzes* dan *Logout*. Masing-masing *case* yang terdapat pada diagram tersebut akan dijelaskan pada *Activity Diagram*.

### 3.3.2 Activity Diagram

## IMD Empathy Activity Diagram

Created by: Abdul Hakim Ghaniy



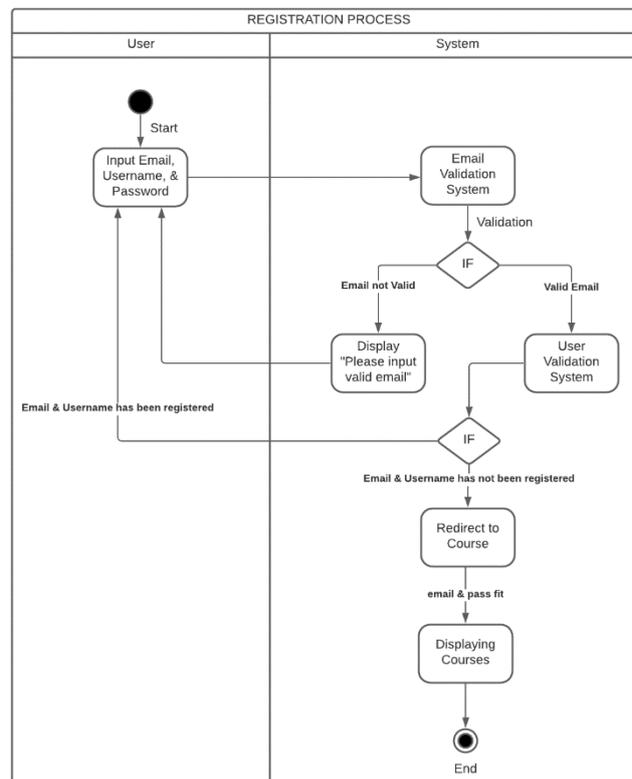
Gambar 3.3 Activity Diagram Pada Session Process

Diagram ini menjelaskan bagaimana alur dari proses masuk hingga sampai pada proses ditampilkannya materi pembelajaran. Lingkaran hitam adalah suatu keadaan dimana pengguna memulai aksi. Aksi yang dilakukan yaitu memasukkan data berupa email dan kata sandi. Jika data tersebut cocok, maka akan ditampilkan materi-materi pembelajaran. Jika pengguna lupa kata sandi yang dimiliki, pengguna diarahkan untuk masuk ke halaman lupa kata sandi, lalu sistem akan mengecek data tersebut. Jika pengguna terdaftar di

sistem, langkah selanjutnya adalah pengiriman email berupa tautan *recovery password* yang dapat digunakan sebagai jalan menuju halaman penggantian kata sandi baru. Bagi pengguna yang baru mendaftar, telah disediakan halaman pendaftaran tersendiri. Adapun *Activity Diagram* pada proses pendaftaran sebagai berikut:

### IMD Empathy Activity Diagram

Created by: Abdul Hakim Ghaniy

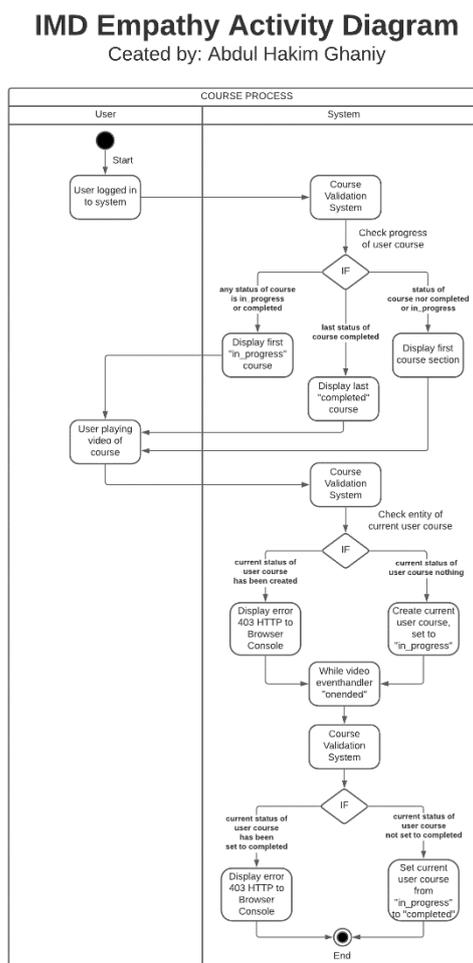


Gambar 3.4 Activity Diagram Pada Registration Process

Diagram ini menjelaskan bagaimana alur dari proses registrasi pengguna. Registrasi dibutuhkan untuk mendata rekam jejak materi yang akan dipelajari. Pada proses registrasi, aksi yang pertama kali dilakukan adalah memasukkan email, nama pengguna, dan kata sandi baru. Jika email atau nama pengguna belum pernah terdaftar sebelumnya, maka data tersebut akan dibuat

dan pengguna akan langsung diarahkan ke materi pertama pada sistem pembelajaran empati, namun jika sudah terdaftar, maka sistem akan memberikan notifikasi bahwa data tersebut sudah digunakan sebelumnya.

Pengguna yang berhasil *login* atau berhasil mendaftar akan diarahkan ke halaman materi pembelajaran. Materi-materi tersebut disajikan sesuai status pembelajaran sebelumnya. *Activity Diagram* pada proses penyajian materi sebagai berikut:



Gambar 3.5 Activity Diagram Pada Course Process

Diagram ini menjelaskan adanya *Course Validation System* yang berguna sebagai sistem validasi status terkini dan pendataan status dari tiap-

tiap materi. Sistem ini aktif ketika pengguna masuk kedalam sistem, baik melalui metode *login* atau *register*.

ID pengguna digunakan untuk mengidentifikasi tabel status pembelajaran. Jika terdapat materi yang status pembelajarannya "*in\_progress*" dan terdapat materi yang berstatus "*completed*", sistem validasi akan mengarahkan pengguna langsung ke materi dimana masih dalam status "*in\_progress*".

Jika terdapat banyak materi yang masih berstatus "*in\_progress*", maka materi berstatus "*in\_progress*" dengan ID lebih awal yang akan ditampilkan. Jika tidak terdapat materi yang berstatus "*in\_progress*" (semua materi pembelajaran yang diakses dalam status "*completed*"), maka materi berstatus "*completed*" dengan ID paling akhir yang akan ditampilkan. Jika status pembelajaran dengan ID pengguna saat ini tidak ada di sistem (pengguna baru mendaftar dan belum pernah mengikuti materi), maka materi dengan ID = 1 yang akan ditampilkan.

Penjelasan lebih detail tentang validasi untuk pengguna yang sudah mengikuti materi pembelajaran dengan status pembelajaran "*in\_progress*" dan "*completed*" akan ditampilkan dalam bentuk program sederhana menggunakan bahasa Ruby sebagai berikut:

```

1 user_courses = {
2   "1": {
3     course_status: [
4       { "pembukaan": "Completed",
5         "definisi": "in progress",
6         "faktor-penyebab-rusaknya-empati": "in progress"
7     }
8   ],
9   "2": {
10    course_status: []
11  }
12 }
13
14 user_courses.each do |user_id, course_status|
15   course_status.each do |key, data|
16     # Jika user belum pernah mengikuti course,
17     # maka data course_status = Empty!
18     if data == []
19       puts "ID: #{user_id}, CourseStatus = Empty!"
20     end
21
22     # Mengambil semua data course status,
23     # baik yang completed atau yang in progress
24     data.each_with_index do |details, index|
25       details.each do |name, status|
26         # Validasi dimulai
27         # hanya hanya mengambil course yang sedang "in progress"
28         if status == "in progress"
29           # index dimulai dari 0
30           # Tambahkan 1 untuk menampilkan index seakan akan ID yang
31           # dimulai dari 1
32           puts "ID: #{user_id}, Course: #{name}, Status: #{status},
33             ID course status: #{index+1}"
34         end
35       end
36     end
37   end
38 end

```

```

[Running] ruby /home/kaany/integrated-daarurrahman-app/tmpCodeRunnerFile.ruby
ID: 1, Course: definisi, Status: in progress, ID Course Status: 2
ID: 1, Course: faktor-penyebab-rusaknya-empati, Status: in progress, ID Course Status: 3
ID: 2, CourseStatus = Empty!

[Done] exited with code=0 in 0.211 seconds

```

Gambar 3.6 Penjelasan Validasi *Course* Menggunakan Program Ruby

Setelah *Course Validation System* berhasil menampilkan materi dan video pembelajaran yang sesuai, langkah selanjutnya adalah pembuatan Status Pembelajaran Terkini ketika pengguna menekan tombol *play*.

Ada 4 kemungkinan status sebelum pengguna menekan tombol *play*. Pertama, pengguna mengakses materi tersebut karena materi tersebut masih berstatus "*in progress*". Kedua, pengguna mengakses materi tersebut karena berstatus "*completed*" dengan ID paling akhir dari deretan materi berstatus "*completed*" lainnya. Ketiga, pengguna mengakses materi tersebut sebagai pengguna baru, yang mana sudah ditentukan oleh sistem bahwa materi dengan ID = 1 akan ditampilkan bagi pengguna baru yang belum pernah mengikuti materi pembelajaran, dan terakhir pengguna adalah mengakses materi baru yang status pembelajarannya belum pernah dicatat.

Dari 4 kemungkinan tadi, dua kemungkinan pertama akan menyebabkan konflik dan redundansi data jika tidak terdapat sistem validasi.

Maka, fungsionalitas *Course Validation System* diperluas dengan menambahkan filter untuk mencegah kemungkinan Pertama dan Kedua dicatat kembali status pembelajarannya.

Setiap materi yang sudah berada dalam status "*in\_progress*" maupun "*completed*" tidak akan dicatat dan akan menampilkan galat 403 *Forbidden* pada respon API jika tetap dilakukan pencatatan. Respon galat ini hanya ditampilkan pada konsol peramban dan tidak akan ditampilkan pada halaman web serta tidak menyebabkan video gagal dimainkan. Kenyamanan pengguna ketika mengakses aplikasi sebagai pertimbangan akan hal ini.

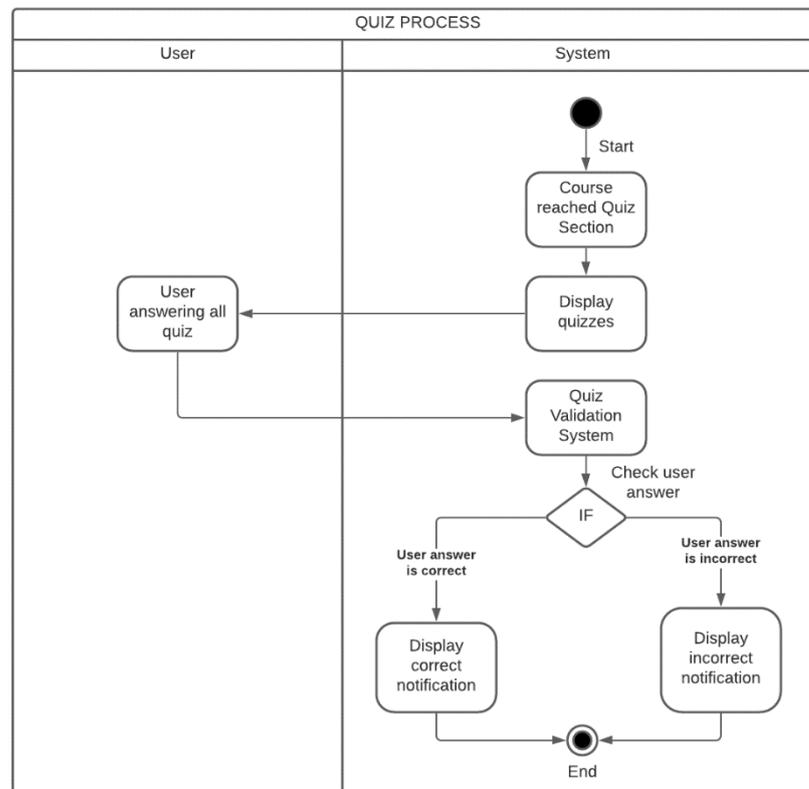
Lain halnya dengan pengguna yang baru pertama kali mengakses materi, sistem validasi tidak akan memblokir pendataan tersebut dan status pembelajarannya akan dibuat dan langsung di-*set* menjadi "*in\_progress*".

Setelah video berakhir, materi pembelajaran terkini akan langsung di-*set* menjadi "*completed*" berdasarkan *event handler onended* pada JavaScript. Materi pembelajaran yang sudah berstatus "*completed*" akan digagalkan pemutakhiran ulang datanya oleh *Course Validation System* demi mencegah redundansi data.

Setelah pengguna selesai mempelajari beberapa materi yang disampaikan, pengguna dapat mengakses kuis yang disediakan pada aplikasi ini. *Activity Diagram* pada penyajian kuis sebagai berikut:

## IMD Empathy Activity Diagram

Created by: Abdul Hakim Ghaniy



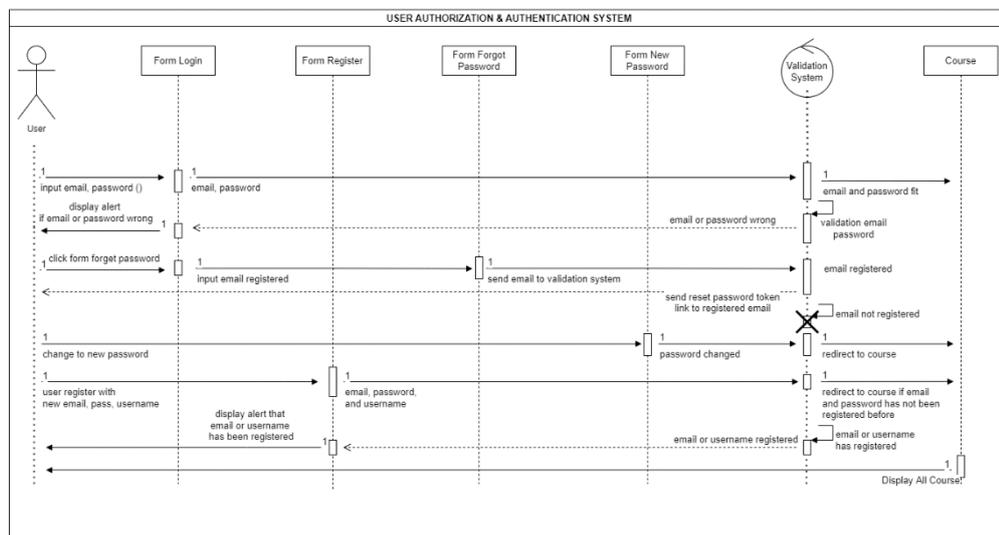
Gambar 3.7 Activity Diagram Pada Quiz Process

Proses pada kuis hanya melakukan validasi jawaban pengguna sesuai dengan materi yang dipelajari. Jawaban dari tiap-tiap soal telah disesuaikan oleh pihak yang kompeten dibidang Psikologi dan Pendidikan Karakter Anak. Setiap jawaban yang dikirim akan divalidasi oleh *Quiz Validation System*, jika jawaban benar akan ditampilkan *popup* jawaban benar, begitu juga sebaliknya.

### 3.3.3 Sequence Diagram

## IMD Empathy Sequence Diagram

Created by: Abdul Hakim Ghaniy



Gambar 3.8 Sequence Diagram Pada Auth Process

Diagram *Sequence* diatas menjelaskan alur proses pada sistem otentikasi dan otorisasi pengguna. Terdapat 5 objek yang ada didalamnya yaitu *Form Login*, *Form Register*, *Form Forgot Password*, *Form New Password* dan *Course*, 1 *Control Flow* yaitu *Validation System* dan 1 aktor yaitu pengguna itu sendiri.

Setiap aksi yang dilakukan pengguna terkait proses otentikasi dan otorisasi akan membawa pesan yang disampaikan dari masing-masing objek ke *Control Flow*, lalu *Control Flow* akan mengatur *output* yang sesuai dengan validasi dan aksi yang dilakukan.

Dimulai dari aksi pengguna yaitu memasukkan email dan kata sandi terdaftar pada halaman *login*, lalu *Control Flow* melakukan validasi pengguna. Jika pengguna terdaftar, *Control Flow* akan mengalihkan halaman *login* ke

halaman *course*. Jika tidak terdaftar, *Control Flow* akan mengembalikan hasilnya untuk men-*trigger* notifikasi kesalahan pada halaman *login*.

Aksi lainnya yakni jika pengguna lupa kata sandi yang terdaftar. Pengguna mengakses halaman *forgot password* yang diakses pertama pada halaman *login*, lalu pengguna memasukkan email terdaftar hanya jika halaman *forgot password* sudah tampil di peramban. *Control Flow* akan memvalidasi email tersebut. Jika sesuai dengan yang terdaftar di sistem, maka *Control Flow* akan mengembalikan hasilnya berupa *Reset Password Link* yang dikirim via email. Pengguna membuka tautan tersebut di peramban dan mengganti kata sandi lama dengan kata sandi baru. Perubahan disimpan oleh sistem dan akan langsung diarahkan ke halaman *course* setelah menekan tombol *Change My Password*.

Apabila email tidak terdaftar, *Control Flow* akan otomatis men-*destroy* aksi tersebut dengan tidak dikirimnya *Reset Password Link* ke email yang dimasukkan pada halaman *forgot password*.

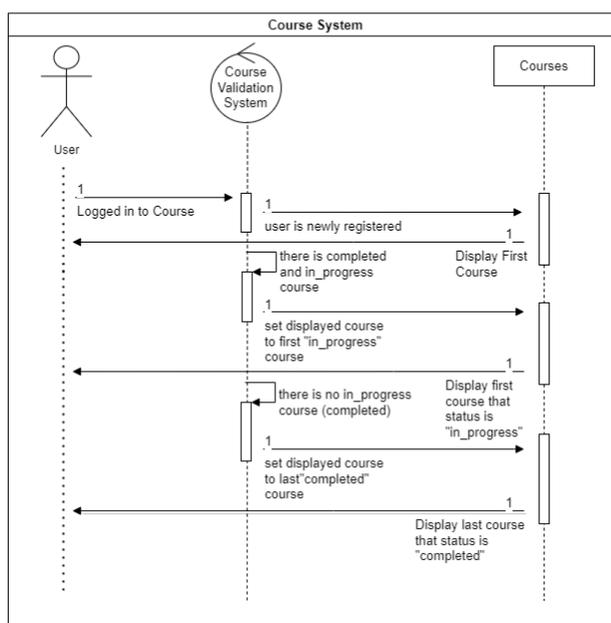
Aksi terakhir jika pengguna melakukan pendaftaran akun pada halaman *register*. Pengguna diminta untuk memasukkan email, nama pengguna, dan kata sandi. *Form Register* mengirim data tersebut ke *Control Flow*. Jika nama pengguna atau email telah terdaftar di sistem, *Control Flow* akan mengembalikan hasilnya untuk men-*trigger* notifikasi kesalahan pada halaman *register*. Jika nama pengguna dan email belum pernah terdaftar sebelumnya, serta kata sandi yang sesuai dengan validasi yang diterapkan sistem, data

disimpan dan pengguna akan langsung diarahkan ke halaman *course* dengan judul materi “Pembukaan” sebagai materi dengan ID = 1.

Setelah pengguna masuk ke halaman *course*, penerapan *control flow* digunakan untuk menampilkan materi yang sesuai dengan status pembelajaran terakhir pengguna. *Sequence Diagram* untuk *course process* sebagai berikut:

## IMD Empathy Sequence Diagram

Created by: Abdul Hakim Ghaniy



Gambar 3.9 *Sequence Diagram* Pada *Course Process*

Diagram ini menjelaskan alur data pada *course process*. Terdapat 1 objek yaitu *courses*, 1 *Control Flow*, dan 1 Aktor. Aktor atau pengguna hanya dapat mengakses halaman *course*. Pada tahap ini, pemegang kendali penuh terhadap data yang disajikan terletak pada *control flow*. Sistem ini memproses data berdasarkan Status Pembelajaran Terakhir yang pengguna dapatkan.

Jika pengguna keluar aplikasi sebelum materi video selesai, maka Status Pembelajaran Terakhir yakni "*in\_progress*". Materi yang ditampilkan saat pengguna *login* kembali yaitu materi pertama dari deretan materi-materi berstatus "*in\_progress*" lainnya.

Jika pengguna menyelesaikan salah satu *course* terakhir dan keluar aplikasi, maka Status Pembelajaran Terakhir yakni "*completed*". Materi yang ditampilkan saat pengguna *login* kembali yaitu materi terakhir dari deretan materi-materi berstatus "*completed*" lainnya.

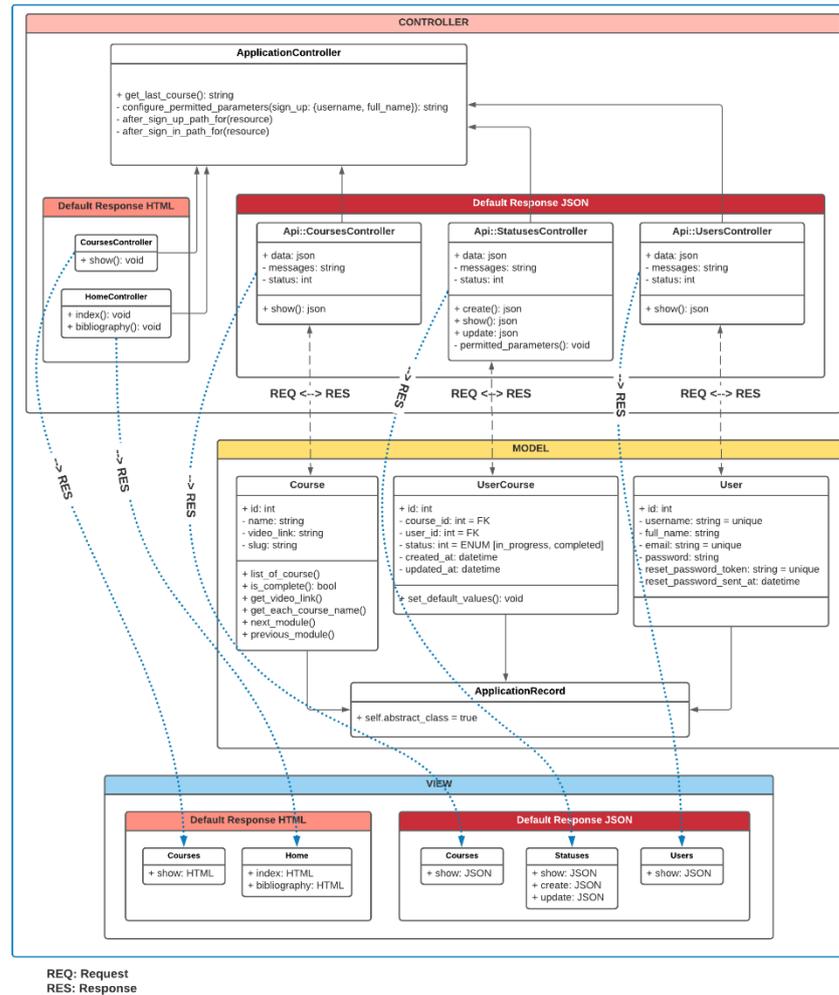
Terakhir, jika pengguna baru mendaftar dan belum pernah memulai materi, atau pengguna yang telah menyelesaikan beberapa materi dan ingin memulai materi baru (belum menekan tombol *play*), maka materi yang disajikan adalah materi pembelajaran dengan ID = 1 (untuk pengguna baru) dan pengalihan pada materi dengan status "*completed*" terakhir (pengguna yang ingin memulai materi).

#### 3.3.4 ***Class Diagram***

*Class Diagram* merupakan representasi dari gabungan antar entitas yang ada di sistem dalam wujud kelas-kelas. Tiap-tiap entitas memiliki tipe data serta fungsi yang dapat digunakan untuk berkomunikasi dengan entitas lainnya. *Class Diagram* pada penelitian ini adalah sebagai berikut:

## IMD Empathy Class Diagram

Created by: Abdul Hakim Ghaniy



Gambar 3.10 *Class Diagram* Pada LMS Berempati di Media Sosial

Dalam arsitektur MVC, setiap bagian yang terdiri dari *Model*, *View* dan *Controller* memiliki *parent class* nya masing-masing. Konsep ini dalam Pemrograman Berorientasi Objek disebut *Inheritance* atau Pewarisan. Setiap *class* yang dibuat mewarisi sifat induknya, mempunyai fungsi tersendiri, tanpa mempengaruhi *parent class*.

Pada diagram diatas, terdapat 3 *container* yang masing-masing mewakili setiap komponen. Pada *container controller*, terdapat 2 *scope* yang

masing-masing memiliki tugas berbeda. *Scope* atau cakupan pertama bertugas untuk menangani permintaan dan pengiriman data dalam bentuk HTML, dan cakupan kedua bertugas untuk menangani permintaan dan pengiriman data dalam bentuk JSON yang ditangani oleh RESTful API.

Cakupan pertama pada *container controller* terdiri dari *class CoursesController* dan *HomeController*. Cakupan kedua terdiri dari *class Api::CoursesController*, *Api::StatusesController*, dan *Api::UsersController*. Setiap *class* pada masing-masing *controller* memiliki *action*. *Action* ini dapat saling terhubung dengan *View*.

Asas *convention over configuration* pada Rails mengharuskan penamaan setiap folder di *View* sesuai dengan nama *Controller* yang ingin dihubungkan, serta nama berkas yang menampilkan halaman web sesuai dengan nama *action* yang dimiliki *controller* tersebut. Misal: kelas *controller HomeController* yang dapat menampilkan halaman beranda web. Hal yang dilakukan adalah membuat sebuah folder dengan nama *Home* dengan 2 berkas yang didalamnya sesuai *action* yang dimiliki: *index.html.erb* untuk *action index()* dan *bibliography.html.erb* untuk *action bibliography()*. Semua *controller* tersebut mewarisi sifat induknya, yakni *ApplicationController* yang merupakan *inheritance* dari *ActionController::Base*.

*Container* selanjutnya adalah *model*. Pada *container* ini terdapat 3 *class* yang merepresentasikan entitas pada basis data: *class Course* dengan tabel bernama *courses*, *class User* dengan tabel bernama *users*, *class UserCourse* dengan tabel bernama *user\_courses*.

Attribut yang dimiliki dari masing-masing entitas dipaparkan pada tabel berikut:

Tabel 3.1 Atribut pada Entitas *Course*

<b><i>Model Name: Course</i></b>		
<b><i>Table Name: courses</i></b>		
<b><i>Attribute</i></b>	<b><i>DataType</i></b>	<b><i>Notes</i></b>
<i>id</i>	<i>Integer</i>	<i>Primary Key</i>
<i>name</i>	<i>String</i>	
<i>video_link</i>	<i>String</i>	
<i>slug</i>	<i>String</i>	

Tabel 3.2 Atribut pada Entitas *User*

<b><i>Model Name: User</i></b>		
<b><i>Table Name: users</i></b>		
<b><i>Attribute</i></b>	<b><i>DataType</i></b>	<b><i>Notes</i></b>
<i>id</i>	<i>Integer</i>	<i>Primary Key</i>
<i>username</i>	<i>String</i>	<i>Unique</i>
<i>email</i>	<i>String</i>	<i>Unique</i>
<i>password</i>	<i>String</i>	<i>Bcrypt Encrypted</i>
<i>full_name</i>	<i>String</i>	
<i>reset_password_token</i>	<i>String</i>	<i>Unique</i>
<i>reset_password_sent_at</i>	<i>Datetime</i>	

Tabel 3.3 Atribut pada Entitas *UserCourse*

<b><i>Model Name: UserCourse</i></b>		
<b><i>Table Name: user_courses</i></b>		
<b><i>Attribute</i></b>	<b><i>DataType</i></b>	<b><i>Notes</i></b>
<i>id</i>	<i>Integer</i>	<i>Primary Key</i>
<i>user_id</i>	<i>Integer</i>	<i>Foreign Key</i>
<i>course_id</i>	<i>Integer</i>	<i>Foreign Key</i>
<i>status</i>	<i>Integer</i>	<i>Enum:[in_progress, completed]</i>
<i>created_at</i>	<i>Datetime</i>	
<i>updated_at</i>	<i>Datetime</i>	

Penentuan *relationship* antar entitas ditetapkan dalam *Model*. Macam-macam relasi antar entitas seperti *one-to-one*, *one-to-many*, dan *many-to-many* ditetapkan menggunakan *built-in helper method*. Untuk relasi *one-to-one*, *helper method* ***has\_one*** ditetapkan pada entitas berderajat kuat, dan ***belongs\_to*** ditetapkan pada entitas berderajat lemah. Pada relasi *one-to-many*, *helper method* ***has\_many*** ditetapkan pada entitas berderajat kuat dan ***belongs\_to*** ditetapkan pada entitas berderajat lemah. Sedangkan pada relasi *many-to-many*, sebuah model baru dibuat sebagai penghubung antar entitas. *Helper method* ***has\_many*** ditetapkan pada entitas yang membutuhkan, dan ***belongs\_to*** ditetapkan pada *model* penghubung tersebut.

Relasi antara *User*, *Course* dan *UserCourse* ditetapkan sebagai relasi *many-to-many*. Karena setiap user dapat mempelajari banyak *Course* yang ditetapkan status pembelajarannya pada *UserCourse*. Begitupun dengan *Course*, entitas ini dapat dimiliki oleh banyak *User* melalui perantara *UserCourse* yang mencatat status pembelajaran dari masing-masing pengguna.

Penetapan *helper method* pada masing-masing entitas di *model* dapat dilihat dari tangkapan layar berikut:

```

1 class Course < ApplicationRecord
2   extend FriendlyId
3   friendly_id :name, use: :slugged
4
5   has_many :user_courses
6   has_many :users, through:
7     :user_courses
8 end
9
1 class UserCourse < ApplicationRecord
2   belongs_to :user
3   belongs_to :course
4
5   before_save :set_default_values
6
7   enum status: [:not_started, :in_progress,
8     :completed]
9
10 private
11 def set_default_values
12   self.status ||= :not_started
13 end
14 end
15
1 class User < ApplicationRecord
2   # Include default devise modules. Others
3   available are:
4   # :confirmable, :lockable, :timeoutable,
5   # :trackable and :omniauthable
6   devise :database_authenticatable,
7     :recoverable, :rememberable,
8     :validatable
9
10 validates_uniqueness_of :username
11 has_many :user_courses
12 has_many :courses, through: :user_courses
13 end

```

Gambar 3.11 Penetapan *helper method associations*

### 3.4 Metode Pengujian Sistem

Dalam pengembangan aplikasi, pengujian merupakan tahap untuk memastikan bahwa sistem berjalan sesuai dengan yang diharapkan dan menampilkan pesan kesalahan jika terjadi hal yang tidak sesuai (Mustaqbal dkk., 2015). Pada implementasinya, pengujian terdiri dari dua macam, pengujian *blackbox* dan pengujian *whitebox*.

Penelitian ini menggunakan *blackbox* sebagai metode pengujian sistem, fokus utama pengujian berada pada sisi fungsionalitas aplikasi. Sebelum melakukan pengujian, *tester* mendefinisikan terlebih dahulu setiap kondisi masukan dan hasil yang diharapkan dari masukan tersebut, jika aplikasi berhasil menampilkan *output* yang diharapkan, maka pengujian dianggap lolos.

### 3.5 Prosedur dan Teknik Pengujian Sistem

Penelitian ini menggunakan teknik *Equivalent Partitioning* sebagai salah satu metode pengujian *Blackbox*. Pada *equivalent testing*, pengujian dilakukan untuk menguji validasi yang diterapkan sistem, seperti *form input* dan *API response*. Setiap pengujian harus memiliki identitas dan keterhubungan dari masing-masing komponen yang diuji (Maulana dkk., 2020).

Prosedur pengujian pada penelitian ini yakni mempersiapkan komponen yang akan diuji, membuat skenario pengujian, lalu menguji komponen-komponen tersebut berdasarkan model atau skenario yang sudah dibuat (Maulana dkk., 2020).

### 3.6 Waktu dan Lokasi Penelitian

Penelitian ini bekerjasama dengan Komunitas Empati, sebuah komunitas yang berfokus pada penelitian dan pengembangan tentang Empati serta pencegahan dalam perundungan maya. Perancangan dan pengembangan sistem pembelajaran digital ini berada dalam kerjasama *Empathy Project*. Berikut data yang diperoleh:

Tabel 3.4 Lokasi Penelitian

Institusi	<i>Empathy Community</i>
Status	Organisasi Nirlaba
Pimpinan	Dr. Charyna Ayu Rizkyanti, S.Psi., M.A
Alamat	Cinere Estate Blok A, Jl. Mangga No. 203, Depok, Jawa Barat 16514
Naungan Kerjasama	<i>Empathy Project</i>

### 3.7 Bahan Penelitian

Secara garis besar, bahan dalam penelitian ini adalah terdegradasinya rasa empati dikalangan anak-anak, remaja dan pengguna media sosial, sehingga meningkatnya kasus perundungan, baik secara langsung maupun dunia maya, *hoax*, ujaran kebencian, serta kekerasan verbal, sehingga perlunya penyampaian nilai-nilai empati yang dapat diterima oleh semua kalangan, dengan penyampaian terstruktur dan informatif, serta memanfaatkan masifnya perkembangan teknologi dan internet, yaitu dengan pengimplementasian pembelajaran digital berbasis LMS.

### 3.8 Peralatan Penelitian

Dalam menjalankan penelitian ini dibutuhkan beberapa *tools*, baik dari tahap analisis, perancangan, hingga implementasi. Pada tahap analisis, *tools*

yang digunakan yaitu pulpen, kertas HVS dan internet. Pada tahap perancangan, *tools* yang digunakan yaitu:

1. Laptop Lenovo dengan spesifikasi Intel Core i5 10<sup>th</sup> Gen dan RAM 12GB
2. Figma untuk perancangan desain *wireframe* dan *mockup*
3. LucidChart dan diagrams.net untuk perancangan desain UML
4. *Visual Studio Code* (VSCode) untuk penulisan kode program
5. Github untuk menyimpan setiap perubahan pada kode yang ditulis dalam bentuk *repository*
6. Netlify untuk menjalankan *prototype* berupa Halaman Web Statis
7. Cloudinary *Transformation* untuk mereduksi ukuran setiap video yang diunggah ke Cloudinary *Content Delivery Network* (CDN)
8. Google Drive sebagai media penyimpanan aset mentah

Pada tahap implementasi, *tools* yang digunakan yaitu:

1. Domain dengan nama ayoberempati.id
2. CloudFlare sebagai bagian dari proteksi terhadap serangan *Distributed Denial of Service* (DDoS), dan sebagai *DNS Management*
3. *Virtual Private Server* (VPS) dengan spesifikasi VCPU 2 Core dan RAM 4GB
4. Cloudinary CDN
5. *Let's Encrypt* SSL

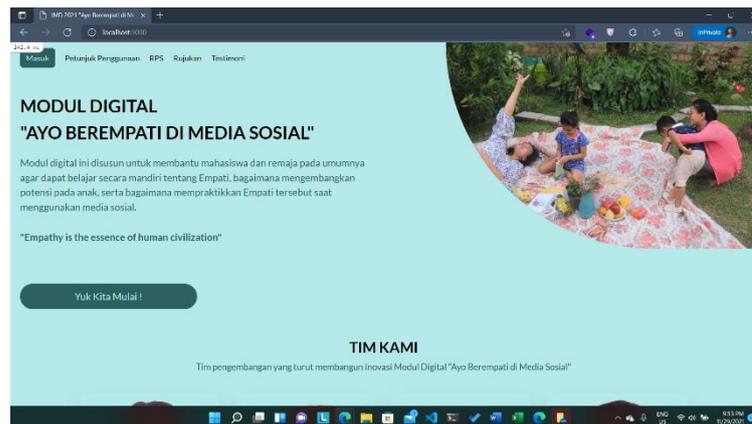
## BAB IV

### HASIL DAN PEMBAHASAN PENELITIAN

#### 4.1 Hasil Penelitian

Penelitian ini telah menghasilkan sebuah aplikasi yang berjalan pada *framework Ruby on Rails* berbasis LMS dengan beberapa modul yang terdapat didalamnya seperti halaman beranda dengan konsep *Landing Page*, halaman Rujukan, *Form Login*, *Form Register*, *Form Forgot Password*, halaman *Course* dan Kuis, Secara rinci spesifikasi yang terdapat di dalamnya akan dipaparkan pada poin-poin berikut:

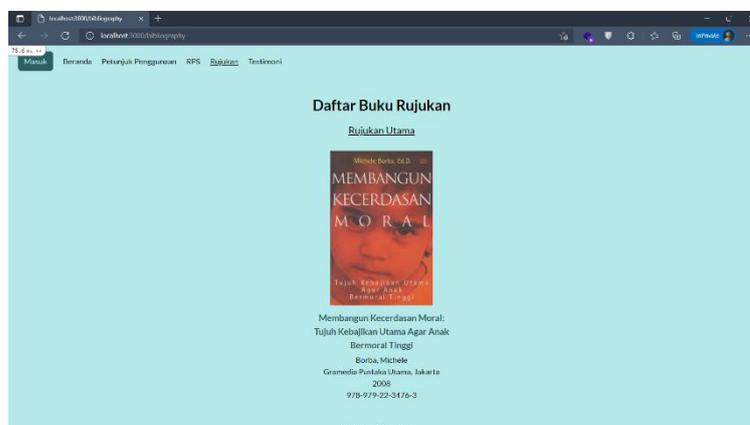
##### 4.1.1 Implementasi Halaman Beranda



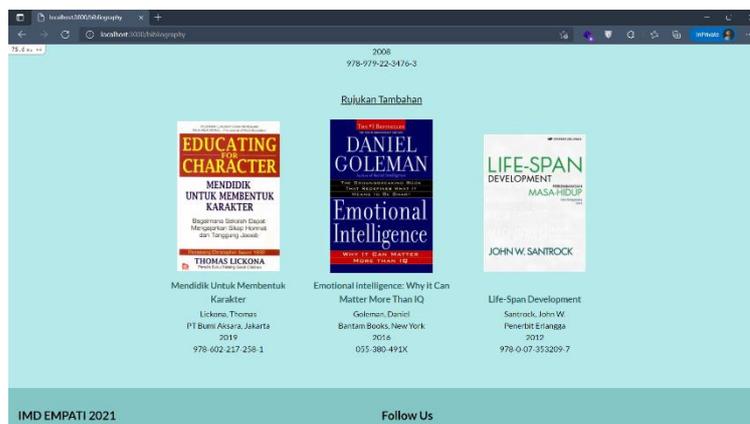
Gambar 4.1 Halaman Beranda

Pada halaman beranda terdapat beberapa tautan yang mengarah pada masing-masing fitur seperti tombol *Masuk* untuk mengarah ke halaman *login*, tautan *Petunjuk Penggunaan*, tautan *Rencana Pembelajaran Semester (RPS)* yang memuat RPS pada matakuliah *Psikologi dan Pengembangan Karakter Anak*, tautan *Rujukan* yang berisi berbagai *Buku Rujukan* pada pengembangan

materi tentang Empati, dan tautan Testimoni. Petunjuk Penggunaan, RPS, dan Testimoni mengarah ke berkas yang disimpan pada Google Drive, sedangkan halaman Rujukan adalah *built-in page* yang terdapat di sistem. Berikut



Gambar 4.2 Halaman Rujukan: Buku Rujukan Tambahan



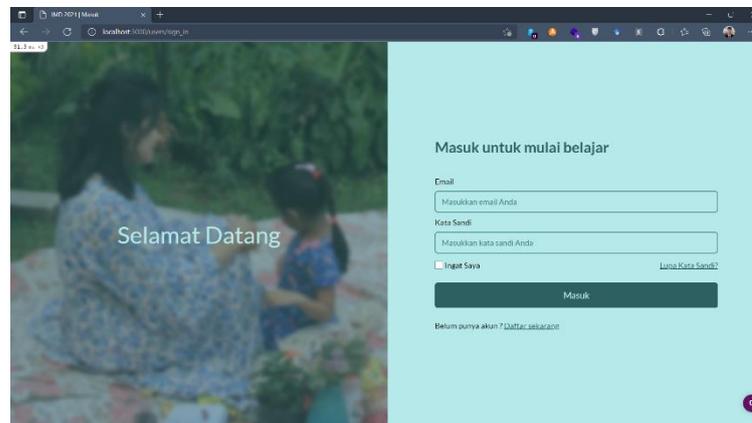
Gambar 4.3 Halaman Rujukan: Buku Rujukan Utama

tampilannya:

#### 4.1.2 Implementasi Halaman Login

Pada halaman *login* terdapat 2 *input form*, yaitu Email dan Kata Sandi, serta 1 *checkbox* Ingat Saya yang digunakan untuk menyimpan *session cookie* pengguna dan 1 tombol Masuk. Pada halaman ini juga terdapat 2 tautan, yaitu Lupa Kata Sandi dan Daftar Sekarang. Tautan Lupa Kata Sandi digunakan

untuk mengarahkan pengguna jika lupa kata sandi yang didaftarkan, dan tautan Daftar Sekarang digunakan untuk mengarahkan pengguna jika belum terdaftar

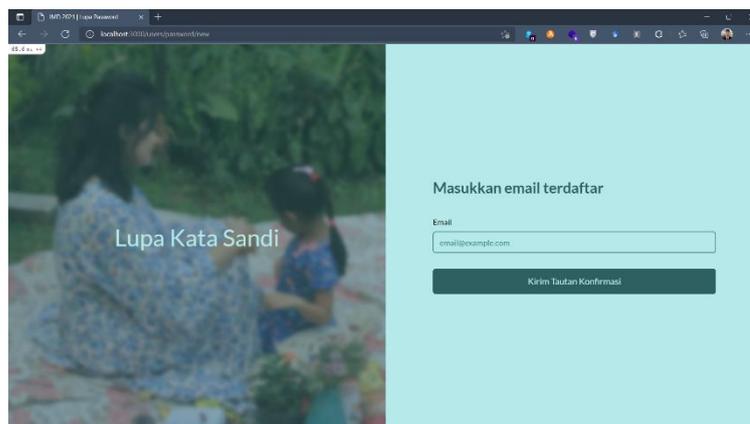


Gambar 4.4 Halaman Login

di sistem.

Pengujian juga dilakukan pada halaman ini menggunakan metode *blackbox* yang bertujuan untuk memvalidasi setiap *behaviour* yang dilakukan pengguna saat mengakses halaman *login*. Pengujian pada masing-masing fitur akan dipaparkan pada sub-bab pembahasan.

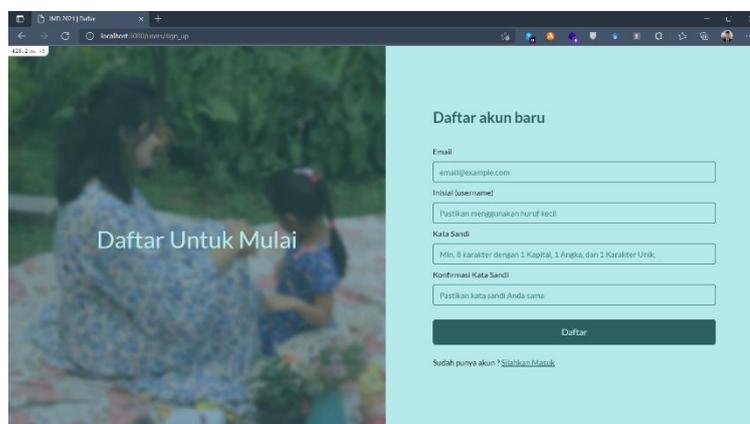
#### 4.1.3 Implementasi Halaman *Forgot Password*



Gambar 4.5 Halaman *Forgot Password*

Halaman ini sebagai sarana sistem untuk mengirim *Reset Password Link*. Sistem akan membaca email yang dimasukkan pada *form* tersebut. Tautan ini dapat digunakan untuk mengganti kata sandi yang dikirim via email, dan hanya email terdaftar yang dapat menerima email perubahan kata sandi. Secara terperinci pengujian untuk halaman ini berada di sub-bab pembahasan.

#### 4.1.4 Implementasi Halaman *Form Register*

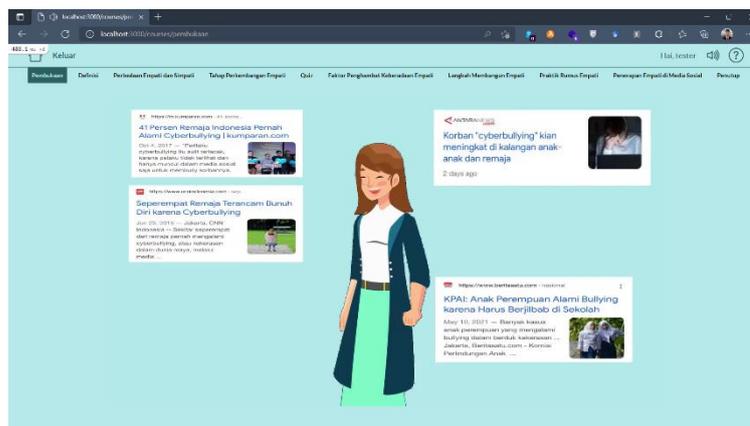


Gambar 4.6 Halaman *Form Register*

Pada halaman *register* terdapat 4 kolom: Email, *Username*, Kata Sandi dan Konfirmasi Kata Sandi dan 1 tombol Daftar. Tiap masing-masing kolom memiliki fungsionalitas tersendiri serta validasi yang akan dibahas pada sub-

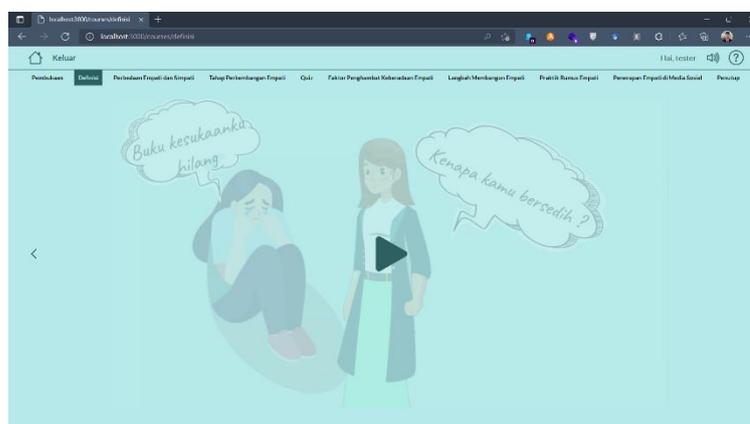
bab pembahasan. Tautan Silahkan Masuk digunakan bagi pengguna yang sudah mendaftar di sistem. Pada beberapa kejadian, pengguna tidak sadar bahwa pernah mendaftarkan email yang digunakan sebelumnya. Ketika *form* ini di-*submit* dan sistem membaca bahwa email yang didaftarkan saat itu pernah digunakan sebelumnya, notifikasi kesalahan akan muncul. Fungsi dari tautan tersebut untuk membantu pengguna menuju halaman *login* tanpa perlu kembali ke halaman beranda jika hal seperti ini terjadi.

#### 4.1.5 Implementasi Halaman *Course*

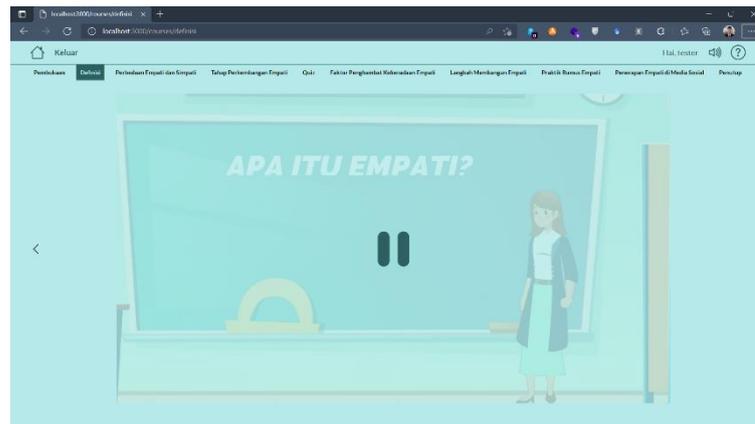


Gambar 4.7 Halaman *Course*

Pada halaman *course*, terdapat beberapa bagian: tombol *Home*, tombol keluar, sapaan ke pengguna, tombol aktif-non-aktifkan suara video, tombol bantuan, *current course navigation* dan video animasi. Pada bagian video animasi, terdapat tombol *play* ketika video belum dimulai, tombol *pause* untuk menghentikan sejenak video, dan tombol *replay* jika ingin mengulang video. Video pembelajaran pada sistem ini tidak dapat dipercepat hingga rentang waktu tertentu. Hal ini dikarenakan dapat merusak konsep yang diciptakan sistem, yaitu pembelajaran terstruktur menggunakan konsep LMS.

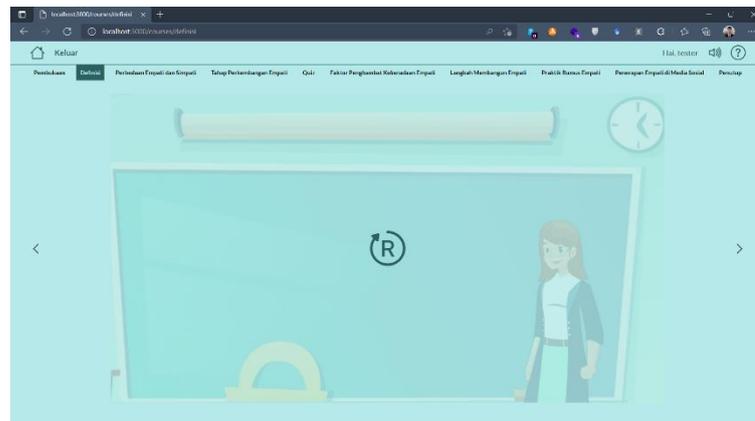


Gambar 4.8 *Play button* dan *to previous course button*



Gambar 4.10 *Pause button* pada video

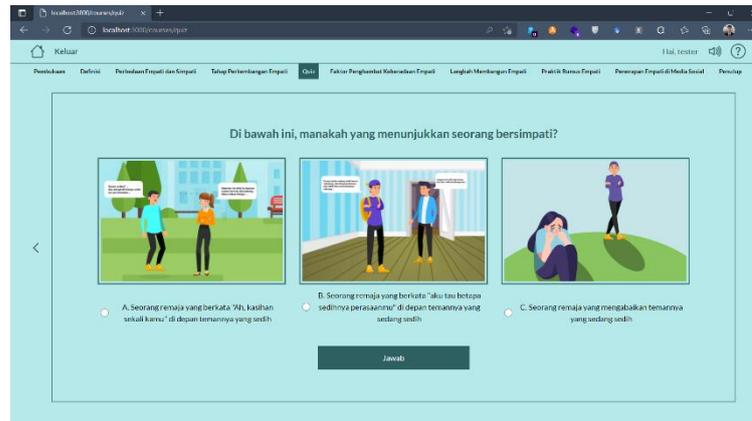
Setiap pengguna mencoba keluar aplikasi sebelum menyelesaikan satu materi setelah menekan tombol *play*, maka pengguna akan dialihkan paksa ke *course* yang belum selesai setelah pengguna *login* kembali ke sistem.



Gambar 4.9 *Replay button* dan *to previous – to next course*

*Replay Button* dan tombol *to next course* akan muncul diakhir video setelah video tersebut selesai ditonton. Tombol *to previous course* akan muncul disetiap materi, kecuali materi Pembukaan.

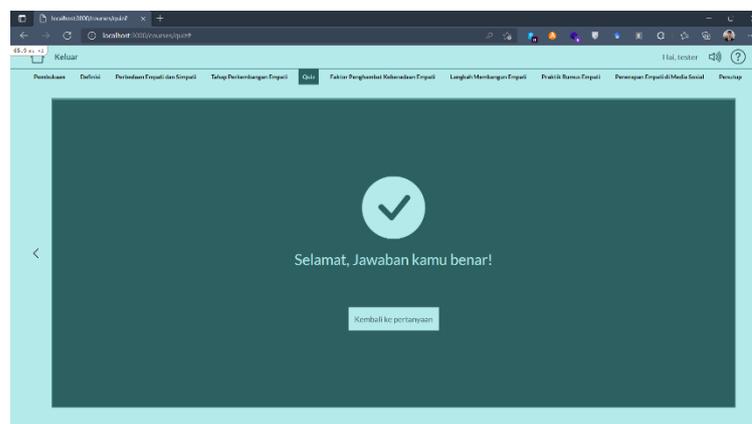
#### 4.1.6 Implementasi Halaman Kuis



Gambar 4.11 Halaman Kuis dan Contoh Pertanyaan

Pada halaman kuis terdapat beberapa pertanyaan. masing-masing pertanyaan dan jawabannya sudah disesuaikan dengan materi dan dibuat oleh pihak yang kompeten dibidang Psikologi dan Pendidikan Karakter Anak.

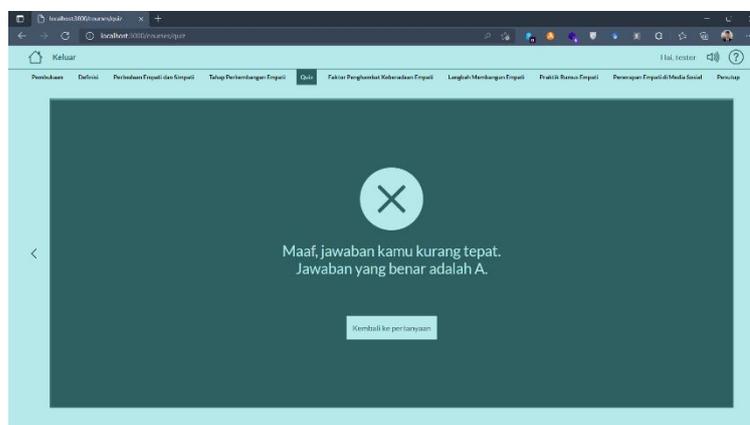
Kuis memiliki 3 jawaban dengan opsi pilihan ganda A, B, atau C. Ketika pengguna menekan *radio button* dari jawaban yang benar, status “*in progress*” pada bagian kuis akan didata sistem. Jika pengguna meng-klik tombol Jawab pada jawaban yang benar, akan muncul *popup* “Selamat, Jawaban Kamu Benar”.



Gambar 4.12 *Popup* Jawaban Benar

Status pada bagian kuis akan di ubah ke “*completed*” sesaat setelah pengguna menekan tombol Kembali Ke Pertanyaan dan tombol *to next course* akan muncul pada sisi kanan layar pengguna.

Hal ini tidak demikian pada jawaban pengguna yang salah, status pembelajaran tidak akan dibuat dan tidak akan diubah statusnya sampai pengguna menjawab pertanyaan tersebut dengan benar. Konsep ini sengaja dibuat agar pengguna dapat memahami pertanyaan-pertanyaan tersebut dengan baik, dan tidak terjadi pemahaman yang salah tentang empati.



Gambar 4.13 *Popup* Jawaban Salah

## 4.2 Pembahasan

Pada penelitian ini, setiap fitur diuji kemampuan dan fungsionalitas serta validasi-validasinya menggunakan metode *blackbox*. Pengujian ini bertujuan untuk menciptakan aplikasi yang *reliable*, *durable* dan layak untuk digunakan secara masif. Secara rinci pengujian pada masing-masing fitur dipaparkan pada poin-poin berikut:

#### 4.2.1 Pengujian Fitur *Login*

Fitur *login* adalah salah satu fitur penting pada sebuah sistem yang digunakan sebagai pembatas antara sebuah sistem dengan manusia. Fitur *login* memberikan hak akses dan sebagai bentuk pembuktian bahwa setiap pengguna yang mengakses sistem adalah pengguna yang sah. Konsep ini disebut *Authentication* dan *Authorization*. Maka, pengujian pada fitur ini adalah salah satu pengujian yang krusial dan harus dilakukan dengan cermat, agar tidak terjadi hal-hal yang tidak terduga seperti peretasan atau aktifitas *login* yang tidak sah. Berikut pengujian untuk fitur ini:

Tabel 4.1 Pengujian Validasi Pada Sistem *Login*

<b>KELOMPOK PENGUJIAN</b>	:	<b>PROSES LOGIN</b>		
<b>Kode</b>	:	<b>ProsesLogin-01</b>		
<b>Jenis Pengujian</b>	:	<b>Pengujian Validasi</b>		
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>		
<b>Teknik Pengujian</b>	:	<i>Equivalent Partitioning</i>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
Email tidak sesuai format regex berikut: $\wedge A [\wedge w + \wedge - . J + @ [ a - z \wedge d \wedge - J + ( \wedge . [ a - z \wedge d \wedge - J + ) * \wedge . [ a - z ] + \wedge z / i$		Menggunakan email dengan bentuk [abduhkimghaniy.com, abduhkimghaniy@lwg]	Menampilkan pesan kesalahan: "Email atau Kata Sandi tidak sah"	Valid
Kolom Email Kosong		Hanya form password yang diisi	Menampilkan pesan kesalahan: "Email atau Kata Sandi tidak sah"	Valid

Kolom Password Kosong	Hanya form email yang diisi	Menampilkan pesan kesalahan: "Email atau Kata Sandi tidak sah"	Valid
Mencoba mengakses <i>course</i> tanpa <i>login</i> via URL	Mencoba masuk ke <i>course</i> via URL: [localhost:3000/course/pembukaan, localhost:3000/course/penutup] tanpa <i>login</i>	Menampilkan pesan kesalahan: "Anda harus mendaftar atau masuk untuk melanjutkan" dan <i>redirect</i> ke halaman <i>login</i>	Valid

Pada tabel 4.1, pengujian validasi dilakukan untuk menguji apakah email yang di-*input* pada *form* email merupakan email yang valid. Terdapat salah satu mekanisme pengecekan, yaitu menggunakan *Regular Expression* (Regex). Regex akan mengembalikan nilai *true* jika masukan sesuai dengan ekspresi karakter yang telah ditentukan, dan akan mengembalikan nilai *false* jika tidak sesuai. Tiap kolom-kolom seperti email dan kata sandi diuji satu persatu untuk memastikan bahwa sistem tidak memberikan hak akses jika hanya salah satu kolom yang terisi, dan kombinasi email serta kata sandi tersebut tidak terdata di sistem.

Tabel 4.2 Pengujian Integritas Data User Pada Sistem *Login*

<b>KELOMPOK PENGUJIAN</b>	:	<b>PROSES LOGIN</b>
<b>Kode</b>	:	<b>ProsesLogin-02</b>
<b>Jenis Pengujian</b>	:	<b>Pengujian Integritas Data User</b>
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>
<b>Metode Pengujian</b>	:	<b><i>Blackbox Testing</i></b>
<b>Teknik Pengujian</b>	:	<b><i>Equivalent Partitioning</i></b>

<b>Data Diuji</b>	<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
User tidak terdaftar (berdasarkan Email dan Password)	Login menggunakan email dan password belum terdaftar	Menampilkan pesan kesalahan: "Email atau Kata Sandi tidak sah"	Valid
Email atau Password salah	Login menggunakan email atau password terdaftar dengan kombinasi salah	Menampilkan pesan kesalahan: "Email atau Kata Sandi tidak sah"	Valid
Email dan Password benar	Login menggunakan email dan password terdaftar dengan kombinasi benar	Berhasil masuk ke halaman <i>Courses</i>	Valid

Pada tabel 4.2, pengujian integritas data *user* dilakukan untuk menguji apakah email dan kata sandi yang dimasukkan pada kolom tersebut dapat diproses dengan baik oleh sistem, seperti menampilkan halaman *Course* hanya kepada *user* yang terdaftar, atau justru tetap memberikan akses masuk pada *user* yang tidak terdaftar. Pengujian juga dilakukan untuk memastikan notifikasi kesalahan hanya muncul jika validasi menemukan email yang tidak sesuai format dan sistem pengecekan menemukan kombinasi email dan kata sandi yang tidak terdaftar.

#### 4.2.2 Pengujian Fitur *Register*

Fitur *register* memungkinkan pengguna untuk menggunakan aplikasi setelah melakukan registrasi. Tujuan fitur ini adalah pemberian ID unik yang disematkan di basis data pada masing-masing pengguna agar pemberian akses ke materi sesuai dengan *progress* yang dicapai. Pengujian juga sangat dibutuhkan pada fitur ini agar tidak terjadi duplikasi data, inkonsistensi data,

dan tercampurnya data pengguna baru dengan pengguna lama. Adapun pengujiannya sebagai berikut:

Tabel 4.3 Pengujian Validasi Pada Sistem *Register*

<b>KELOMPOK PENGUJIAN</b>	:	<b>PROSES REGISTER</b>		
<b>Kode</b>	:	<b>ProsesRegister-01</b>		
<b>Jenis Pengujian</b>	:	<b>Pengujian Validasi</b>		
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>		
<b>Teknik Pengujian</b>	:	<i>Equivalent Partitioning</i>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
Email tidak sesuai format regex berikut: $\wedge A [\wedge w + \wedge - . ] + @ [ a - z \wedge d \wedge - ] + ( \wedge . [ a - z \wedge d \wedge - ] + ) * \wedge . [ a - z ] + \wedge z / i$		Menggunakan email dengan bentuk [abduhakinghaniy.com, abduhakinghaniy@lwg]	Menampilkan pesan kesalahan: "Email tidak sesuai format / Email tidak valid"	Valid
Kolom Email Kosong		Semua form diisi kecuali Form Email	Menampilkan pesan kesalahan: "Kolom Email tidak boleh kosong"	Valid
Kolom Password Kosong		Semua form diisi kecuali Form Password	Menampilkan pesan kesalahan: "Kolom Password tidak boleh kosong"	Valid
Kolom Konfirmasi Password Kosong		Semua form diisi kecuali Form Konfirmasi Password	Menampilkan pesan kesalahan: "Kolom Password tidak boleh kosong"	Valid

Pada tabel 4.3, pengujian dilakukan pada masing-masing *form* untuk menguji validitas data yang akan dicatat oleh sistem. Pengujian validasi

Password tidak sesuai format regex berikut: <code>^(?=.*?[A-Z])(?=.*?[a-z])(?=.*?[0-9])(?=.*?[#?!@\$%^&amp;*]).{8,70}\$</code>	Menggunakan password dengan bentuk [123456, 123456789, Ghaniy, Ghaniy!, ghaniy!98, !@#\$\$%^&*]	Menampilkan pesan kesalahan: "Kata Sandi Anda tidak kuat. Panjang minimal 8 karakter disertai 1 Huruf Kapital, 1 Angka, dan 1 Karakter Unik"	Valid
Password kurang dari 8 karakter	Menggunakan password dengan bentuk [Lwg!98]	Menampilkan pesan kesalahan: "Kata Sandi terlalu pendek"	Valid
Kolom Username kosong	Semua form diisi kecuali Form Username	Menampilkan pesan kesalahan: "Username tidak boleh kosong"	Valid

menggunakan Regex diterapkan untuk kolom email dan kata sandi. Tujuan dari penggunaan Regex pada kedua kolom ini untuk menjaga integritas data yang sesuai dengan standar keamanan sistem. Saat ini, standar keamanan pada kata sandi yaitu minimal panjang karakter 8, dengan minimal 1 huruf kapital, 1 angka, dan 1 karakter unik. Kolom kata sandi hanya menerima bentuk kata sandi yang sesuai standar validasi. Pemberian kata sandi dengan kombinasi sebagai berikut: [Lwg!98, 123456, 123456789, Ghaniy, Ghaniy!, ghaniy!98, !@#\$\$%^&\*] tidak akan diterima oleh sistem dan akan memicu pesan kesalahan "Kata Sandi Anda tidak kuat. Panjang minimal 8 karakter disertai 1 Huruf Kapital, 1 Angka, dan 1 Karakter Unik"

Tabel 4.4 Pengujian Integritas Data *User* Pada Sistem *Register*

<b>KELOMPOK PENGUJIAN</b>	:	<b>PROSES REGISTER</b>		
<b>Kode</b>	:	<b>SistemRegister-02</b>		
<b>Jenis Pengujian</b>	:	<b>Pengujian Integritas Data User</b>		
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>		
<b>Teknik Pengujian</b>	:	<i>Equivalent Partitioning</i>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
Penggunaan Email yang sudah terdaftar		Menggunakan Email [abdulhakimghaniy37@gmail.com] yang sudah terdaftar	Menampilkan pesan kesalahan: "Email sudah digunakan"	Valid
Penggunaan Username yang sudah terdaftar		Menggunakan Username [GhaniyKie] yang sudah terdaftar	Menampilkan pesan kesalahan: "Username sudah digunakan"	Valid
Penggunaan Email dan Username yang belum terdaftar		Menggunakan Username dan Email yang belum terdaftar di sistem	Berhasil mendaftarkan pengguna dan redirect ke halaman <i>course</i>	Valid

Pengujian integritas yang tertera pada tabel 4.4 ditujukan untuk mencegah duplikasi data seperti email atau *username* yang sudah digunakan sebelumnya. Selain menjaga integritas data, hal ini pula dapat meningkatkan kepercayaan pengguna pada keamanan datanya yang diberikan.

#### 4.2.3 Pengujian Fitur *Forgot Password*

Pengujian pada fitur *forgot password* ditujukan untuk memastikan bahwa instruksi perubahan password hanya dapat dikirimkan pada email yang sudah didaftarkan sebelumnya pada proses *register*. Hal ini untuk mencegah kemungkinan hak akses yang tidak diinginkan. Berikut pengujian yang dilakukan:

Tabel 4.5 Pengujian Validasi Pada Sistem *Forgot Password*

<b>KELOMPOK PENGUJIAN</b>	:	<b>PROSES LUPA PASSWORD</b>		
<b>Kode</b>	:	<b>ProsesLupaPassword-01</b>		
<b>Jenis Pengujian</b>	:	<b>Pengujian Validasi</b>		
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>		
<b>Teknik Pengujian</b>	:	<i>Equivalent Partitioning</i>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
Kolom Email Kosong		Mengosongkan form email dan klik Kirim Tautan Konfirmasi	Sistem membuang aksi tersebut dengan tidak menampilkan reaksi	Valid

Tabel 4.6 Pengujian Integritas Data User Pada Sistem *Forgot Password*

<b>KELOMPOK PENGUJIAN</b>	:	<b>PROSES LUPA PASSWORD</b>		
<b>Kode</b>	:	<b>ProsesLupaPassword-02</b>		
<b>Jenis Pengujian</b>	:	<b>Pengujian Integritas Data User</b>		
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>		
<b>Teknik Pengujian</b>	:	<i>Equivalent Partitioning</i>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
Email tidak terdaftar		Menggunakan email yang tidak terdaftar di sistem	Sistem membuang aksi tersebut dan tidak memberikan tautan <i>reset password</i> via email	Valid

Email terdaftar	Menggunakan email yang terdaftar di sistem	Sistem memverifikasi email terdaftar, lalu memberikan tautan <i>reset password</i> via email	Valid
-----------------	--	--	-------

#### 4.2.4 Pengujian *Fetch API*

Pengujian *Fetch API* bertujuan untuk mencegah pengaksesan data yang tidak sah seperti mengakses API tanpa *login*, serta mencegah adanya aktifitas tidak wajar seperti mempercepat materi tanpa menonton secara keseluruhan video pembelajaran yang telah disediakan. Hal ini mereduksi terjadinya kecurangan sehingga materi yang tersampaikan secara utuh dapat diterima oleh pengguna.

Seluruh *control flow* pada halaman *course* diatur oleh API, seperti menampilkan video yang tepat sesuai dengan materi yang sedang berlangsung, menampilkan serta memberikan tautan yang sesuai pada tombol *to next course* dan *to previous course*, menambahkan serta memperbaharui status pembelajaran pengguna, menampilkan materi terakhir yang masih dalam status “*in progress*”, dan lain-lain. Pengujian pada halaman *Course* dan halaman Kuis sudah diwakili oleh pengujian pada *fetch API* sehingga tidak diperlukan pengujian ulang. Berikut pengujian yang sudah dilakukan:

Tabel 4.7 Pengujian Integritas Data User via API

<b>KELOMPOK PENGUJIAN</b>	:	<b>API FETCH</b>
<b>Kode</b>	:	<b>ApiFetch_User-01</b>
<b>Jenis Pengujian</b>	:	<b>Pengujian Integritas Data User via API</b>

<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<b><i>Blackbox Testing</i></b>		
<b>Teknik Pengujian</b>	:	<b><i>Equivalent Partitioning</i></b>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
<i>Fetching /api/users/ tanpa login di web</i>		Pengujian pada API menggunakan <i>method</i> GET untuk <i>fetch /api/users/ tanpa login di web</i>	<i>Response 401 Unauthorized</i> dan tampilkan pesan kesalahan: "Anda harus <i>login</i> dahulu"	Valid
<i>Fetching /api/users/ dengan login di web</i>		Pengujian pada API menggunakan <i>method</i> GET untuk <i>Fetch /api/users/ dengan login di web</i>	<i>Response 200 OK</i> dan menampilkan data user	Valid

Pengujian pada tabel 4.7 dilakukan untuk melihat apakah API akan menampilkan data *user* jika seseorang tidak *login* kedalam sistem. Pengujian ini dilakukan untuk mencegah seseorang melihat data-data yang bukan hak nya.

Tabel 4.8 Pengujian Integritas Data *Course* via API

<b>KELOMPOK PENGUJIAN</b>	:	<b>API FETCH</b>
<b>Kode</b>	:	<b>ApiFetch_Courses-01</b>
<b>Jenis Pengujian</b>	:	<b>Pengujian Integritas Data Course via API</b>
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>

<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>		
<b>Teknik Pengujian</b>	:	<i>Equivalent Partitioning</i>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
<i>Fetching</i> <i>/api/courses/:course_id</i> tanpa <i>login</i> di web		Pengujian pada API menggunakan <i>method GET</i> untuk <i>Fetch</i> <i>/api/courses/pembukaan</i> tanpa <i>login</i> di web	<i>Response 401 Unauthorized</i> dan tampilkan pesan kesalahan: "Anda harus login dahulu"	Valid
<i>Fetching</i> <i>/api/courses/:course_id</i> dengan <i>login</i> di web		Pengujian pada API menggunakan <i>method GET</i> untuk <i>Fetch</i> <i>/api/courses/pembukaan</i> dengan <i>login</i> di web	<i>Response 200 OK</i> dan menampilkan detail data <i>Course</i> berdasarkan <i>FriendlyID</i>	Valid

Pada tabel 4.8, pengujian dilakukan untuk memastikan bahwa data materi yang ada di sistem jika dan hanya jika *user* sedang *login* kedalam sistem. Hal ini untuk memastikan agar data materi tidak digunakan untuk mem-*bypass* status pembelajaran pengguna lain.

Tabel 4.9 Pengujian Integritas Data *UserCourse* via API

<b>KELOMPOK PENGUJIAN</b>	:	<b>API FETCH</b>
<b>Kode</b>	:	<b>ApiFetch_CoursesStatus-01</b>
<b>Jenis Pengujian</b>	:	<b>Pengujian Integritas Data UserCourse via API</b>
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>
<b>Metode Pengujian</b>	:	<i>Blackbox Testing</i>

<b>Teknik Pengujian</b>		: <i>Equivalent Partitioning</i>	
<b>Data Diuji</b>	<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>
<i>Fetching</i> <i>/api/courses/:id/status</i> atases tanpa <i>login</i> di web	Pengujian pada API menggunakan <i>method</i> GET untuk <i>Fetch</i> <i>/api/courses/1/status</i> es tanpa <i>login</i> di web. 1 adalah ID untuk pembukaan	<i>Response</i> 401 <i>Unauthorized</i> dan tampilkan pesan kesalahan: "Anda harus login dahulu"	Valid
<i>Fetching</i> <i>/api/courses/:id/status</i> atases dengan <i>login</i> di web	Pengujian pada API menggunakan <i>method</i> GET untuk <i>Fetch</i> <i>/api/courses/1/status</i> es tanpa <i>login</i> di web. 1 adalah ID untuk pembukaan	<i>Response</i> 200 OK dan menampilkan detail data <i>UserCourse</i> berdasarkan ID yang diberikan pada <i>user</i> yang <i>login</i> saat ini	Valid
<i>Fetching</i> <i>/api/courses/:id/status</i> atases tanpa <i>login</i> di web	Pengujian pada API menggunakan <i>method</i> POST untuk <i>Fetch</i> <i>/api/courses/1/status</i> es tanpa <i>login</i> di web. 1 adalah ID untuk pembukaan	<i>Response</i> 401 <i>Unauthorized</i> dan tampilkan pesan kesalahan: "Anda harus login dahulu"	Valid
<i>Fetching</i> <i>/api/courses/:id/status</i> atases dengan <i>login</i> di web	Pengujian pada API menggunakan <i>method</i> POST untuk <i>Fetch</i> <i>/api/courses/1/status</i> es tanpa <i>login</i> di web. 1 adalah ID untuk pembukaan	<i>Response</i> 403 <i>Forbidden</i> dan tampilkan pesan kesalahan: " <i>Status not saved. Duplicate request for this course detected</i> " Jika sudah dibuat statusnya. Dan <i>Response</i> 200 OK dan tampilkan pesan: " <i>Status of course saved !</i> " jika status belum dibuat	Valid

<i>Fetching</i> <i>/api/courses/:id/statuses/:id</i> tanpa <i>login</i> di web	Pengujian pada API menggunakan <i>method</i> PUT untuk <i>Fetch</i> <i>/api/courses/1/statuses/1</i> tanpa <i>login</i> di web. 1 pertama adalah ID untuk pembukaan, dan 1 kedua adalah ID <i>usercourse</i>	<i>Response</i> 401 <i>Unauthorized</i> dan tampilkan pesan kesalahan: "Anda harus login dahulu"	Valid
<i>Fetching</i> <i>/api/courses/:id/statuses/:id</i> dengan <i>login</i> di web	Pengujian pada API menggunakan <i>method</i> PUT untuk <i>Fetch</i> <i>/api/courses/1/statuses/1</i> dengan <i>login</i> di web. 1 pertama adalah ID untuk pembukaan, dan 1 kedua adalah ID <i>usercourse</i>	<i>Response</i> 403 <i>Forbidden</i> dan tampilkan pesan kesalahan: " <i>Status not saved. Duplicate request for this course detected</i> " Jika sudah dibuat statusnya. Dan <i>Response</i> 200 OK dan tampilkan pesan: " <i>Status of course saved !</i> " jika status belum dibuat	Valid

Tabel 4.10 Pengujian Validasi *UserCourse* via API

<b>KELOMPOK PENGUJIAN</b>	:	<b>API FETCH</b>		
<b>Kode</b>	:	<b>ApiFetch_CoursesStatus-02</b>		
<b>Jenis Pengujian</b>	:	<b>Pengujian Validasi</b>		
<b>Tanggal Pengujian</b>	:	<b>Senin, 29 November 2021</b>		
<b>Metode Pengujian</b>	:	<b><i>Blackbox Testing</i></b>		
<b>Teknik Pengujian</b>	:	<b><i>Equivalent Partitioning</i></b>		
<b>Data Diuji</b>		<b>Skenario Pengujian</b>	<b>Yang Diharapkan</b>	<b>Hasil Pengujian</b>

<i>Fetching</i> /api/courses/:id/statuses menggunakan FriendlyID	Pengujian pada API menggunakan <i>method</i> GET untuk <i>Fetch</i> /api/courses/pembukaan/statuses.	<i>Response</i> 404 <i>Not Found</i> dan tampilkan pesan kesalahan: " <i>Data not found !</i> "	Valid
<i>Fetching</i> /api/courses/:id/statuses dengan pembukaan tidak terdaftar	Pengujian pada API menggunakan <i>method</i> GET untuk <i>Fetch</i> /api/courses/pembukaan-hiyahiya/statuses. Pembukaan-hiyahiya tidak terdaftar di sistem	<i>Response</i> 404 <i>Not Found</i> dan tampilkan pesan kesalahan: " <i>Data not found !</i> "	Valid
<i>Fetching</i> /api/courses/:id/statuses dengan format status tidak sah	Pengujian pada API menggunakan <i>method</i> POST untuk membuat status pada URL /api/courses/1/statuses menggunakan body dengan enum yang tidak didaftarkan ke sistem, yaitu 3, 4, 5. Enum terdaftar: [1,2]	<i>Response</i> 422 <i>Unprocessable Entity</i> dan tampilkan pesan kesalahan: " <i>Status must in_progress</i> "	Valid
<i>Fetching</i> /api/courses/:id/statuses dengan format status tidak sah	Pengujian pada API menggunakan <i>method</i> POST untuk membuat status pada URL /api/courses/1/statuses menggunakan body yang tidak terdapat status	<i>Response</i> 422 <i>Unprocessable Entity</i> dan tampilkan pesan kesalahan: " <i>Status must in_progress</i> "	Valid
<i>Fetching</i> /api/courses/:id/statuses dengan format status tidak sah	Pengujian pada API menggunakan <i>method</i> POST untuk membuat status pada URL /api/courses/1/status	<i>Response</i> 422 <i>Unprocessable Entity</i> dan tampilkan pesan kesalahan: " <i>Invalid authenticity token</i> "	Valid

	es tanpa menggunakan header X-CSRF-Token		
<i>Fetching</i> /api/courses/:id/statuses/:id dengan format status tidak sah	Pengujian pada API menggunakan <i>method</i> PUT untuk membuat status pada URL /api/courses/1/statuses/1 tanpa menggunakan header X-CSRF-Token. 1 pertama adalah ID untuk pembukaan, dan 1 kedua adalah ID usercourse	Response 422 <i>Unprocessable Entity</i> dan tampilkan pesan kesalahan: " <i>Invalid authenticity token</i> "	Valid

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Berdasarkan hasil analisis dari masalah yang dihadapi dan mengacu pada pertanyaan penelitian pada Bab Pendahuluan, serta tinjauan penelitian terdahulu, maka penelitian ini dapat diambil kesimpulan sebagai berikut:

1. Metode pengembangan sistem menggunakan *Rapid Application Development* telah mempercepat proses perancangan dan pembuatan aplikasi, karena *prototype* dapat langsung dilihat oleh klien, sehingga penambahan fitur maupun perubahan dapat dengan mudah dibicarakan, tanpa perlu menunggu aplikasi siap pakai.
2. *Framework Ruby on Rails* memudahkan perancangan dan pengkodean aplikasi karena mudahnya sintaksis bahasa Ruby, serta asas *convention over configuration* yang membuat proses perancangan sistem hanya fokus pada sisi keterbacaan kode, logika, dan keamanan. Tanpa perlu memperhatikan aspek penamaan dan ketergantungan antar *class*.
3. Konsep LMS pada penelitian ini membuat struktur pembelajaran tentang Empati menjadi lebih baik. Ilmu Psikologi kini tidak hanya berkutat dengan literatur-literatur bacaan, namun dapat memanfaatkan masifnya perkembangan teknologi dan internet.

## 5.2 Saran

Penelitian ini masih terdapat beberapa kelemahan yang besar kemungkinan dapat menjadi saran dan masukan untuk penelitian selanjutnya.

Beberapa saran yang dapat disampaikan yaitu:

1. Implementasi *mobile responsive* pada tiap-tiap halaman sehingga aplikasi ini dapat diakses melalui peramban *smartphone*. Seperti telah diketahui sebelumnya, survey APJII mengemukakan bahwa Indonesia mengalami kenaikan jumlah pengguna internet, dan banyak dari mereka yang mengakses internet menggunakan *smartphone* dan paket data (APJII, 2020).
2. Implementasi poin pada kuis tanpa menghilangkan konsep yang telah ada saat ini. Poin ini nantinya dapat digunakan untuk mengukur tingkat efektifitas pembelajaran empati dari materi-materi yang telah disampaikan.
3. Pembuatan *Admin Dashboard* untuk memudahkan pembaharuan materi dan penambahan-penambahan materi lainnya.

## DAFTAR PUSTAKA

- Al Fatta, H. (2020). *Analisis & Perancangan Sistem Informasi Untuk Keunggulan Bersaing Perusahaan & Organisasi Modern* (H. A. Iriyuliana (ed.); 1st ed.). ANDI.
- Andaru, A. (2018). Pengertian database secara umum. *OSF Preprints*, 2.
- Anisah, I., Sufyana, C. M., & Rohmat, B. S. (2015). *Perancangan Sistem Informasi Inventori Barang Menggunakan Ruby on Rails Di Pt . Xxx. Mvc*, 6–8.
- Anwar, K., Fairuzabadi, M., & Sari, M. W. (2017). Rancang Bangun Sistem Manajemen User Hotspot Menggunakan Mikrotik Php Api Berbasis Web Di Pondok Pesantren Al-Luqmaniyyah. *Isbn: 978-602-73690-8-5*, 260–264.
- APJII. (2020). Laporan Survei Internet APJII 2019 – 2020. *Asosiasi Penyelenggara Jasa Internet Indonesia, 2020*, 1–146.  
<https://apjii.or.id/survei>
- Azis, A. A. (2017). PENGEMBANGAN MEDIA E-LEARNING BERBASIS LMS MOODLE PADA MATAKULIAH ANATOMI FISILOGI MANUSIA. *Jurnal Pendidikan Biologi*, 7(1), 1–8.  
<http://journal2.um.ac.id/index.php/jpb/article/view/712>
- Carlson, L., & Richardson, L. (2015). *Ruby Cookbook: Recipes for Object Oriented Scripting*.  
[http://shop.oreilly.com/product/0636920032236.do%5Cnhttp://shop.oreilly.com/product/0636920032236.do?code=WKRURAA&intcmp=il-prog-books-videos-lp-intsrch\\_ruby20140805\\_owo\\_ct](http://shop.oreilly.com/product/0636920032236.do%5Cnhttp://shop.oreilly.com/product/0636920032236.do?code=WKRURAA&intcmp=il-prog-books-videos-lp-intsrch_ruby20140805_owo_ct)

- Chyan, P. (2021). Perancangan Learning Management System Sebagai Pendukung Pembelajaran Jarak Jauh. *Rabit : Jurnal Teknologi Dan Sistem Informasi Univrab*, 6(1), 7–13. <https://doi.org/10.36341/rabit.v6i1.1521>
- Destiningrum, M., & Adrian, Q. J. (2017). Sistem Informasi Penjadwalan Dokter Berbasis Web Dengan Menggunakan Framework Codeigniter (Studi Kasus: Rumah Sakit Yukum Medical Centre). *Jurnal Teknoinfo*, 11(2), 30. <https://doi.org/10.33365/jti.v11i2.24>
- Dictionary, O. (2021). *Oxford Dictionary#Inflection*. [https://www.oxfordlearnersdictionaries.com/definition/american\\_english/inflection](https://www.oxfordlearnersdictionaries.com/definition/american_english/inflection)
- Fadli, S. (2018). Model Rapid Application Development Dalam Pengembangan Sistem Reservasi dan Penyewaan Kamar Hotel. *Jurnal Informatika Dan Rekayasa Elektronik*, 1(1), 57. <https://doi.org/10.36595/jire.v1i1.33>
- Fitriani, Y. (2020). ANALISA PEMANFAATAN LEARNING MANAGEMENT SYSTEM ( LMS ) SEBAGAI MEDIA PEMBELAJARAN ONLINE SELAMA PANDEMI COVID-19 Yuni Fitriani JISICOM ( Journal of Information System , Informatics and Computing ) JISICOM ( Journal of Information System , Informatics and. *Journal of Information System, Informatics and Computing (JISICOM)*, 4(2), 1–8.
- Hanson, D. H. (2016). *The Rails Doctrine*. <https://rubyonrails.org/doctrine/>
- Hertz, Marci Feldman and Everett Jones, Sherry and Barrios, Lisa and David-Ferdon, Corinne and Holt, M. (2015). Association between bullying victimization and health risk behaviors among high school students in the

- United States. *Physiology & Behavior*, 85(1), 833–842.  
<https://doi.org/10.1111/josh.12339>.Association
- Lockhart, T. (2009). *PostgreSQL Tutorial*. 2126.  
<ftp://194.85.164.6/pub/os2/database/pgsql/tutorial.pdf>
- Mahdiana, D. (2011). Pengadaan Barang Dengan Metodologi Berorientasi Obyek : Studi Kasus Pt . Liga Indonesia. *Jurnal TELEMATIKA*, 3(2), 36–43.
- Matsumoto, Y. (2000). *Ruby-Talk Milis, Matz*. Ruby-Talk.  
<http://blade.nagaokaut.ac.jp/cgi-bin/scat.rb/ruby/ruby-talk/2773>
- Maulana, A., Kurniawan, A., Keumala, W., Sukma, V. R., & Saifudin, A. (2020). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Metode Equivalents Partitions (Studi Kasus: PT Arap Store). *Jurnal Teknologi Sistem Informasi Dan Aplikasi*, 3(1), 50.  
<https://doi.org/10.32493/jtsi.v3i1.4307>
- Muhyidin, M. A., Sulhan, M. A., & Sevtiana, A. (2020). Perancangan Ui/Ux Aplikasi My Cic Layanan Informasi Akademik Mahasiswa Menggunakan Aplikasi Figma. *Jurnal Digit*, 10(2), 208.  
<https://doi.org/10.51920/jd.v10i2.171>
- Mustaqbal, M. S., Firdaus, R. F., & Rahmadi, H. (2015). *PENGUJIAN APLIKASI MENGGUNAKAN BLACK BOX TESTING BOUNDARY VALUE ANALYSIS (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN)*. I(3), 31–36.
- Nuraeni, N. (2021). *Rancang Bangun Virtual Reality Pemengalan Tari Daerah di Jawa Barat Pada Sanggar Tari Cineur*. 5(2), 18–25.
- Pratama, A. (2017). *JavaScript Uncover – Panduan Belajar JavaScript untuk*

- Pemula*. 1–3. <https://www.duniaikom.com/javascript-uncover-panduan-belajar-javascript-untuk-pemula/>
- Riggs, S., Ciolli, G., & Bartolini, G. (2017). *PostgreSQL Administration Cookbook, 9.5/9.6 Edition*.  
<https://books.google.cv/books?id=IUIwDwAAQBAJ>
- Rizkyanti, C. A. (2017). *the Role of Empathy on Behavioral and Emotional Strenght*. 4(1), 1–10.
- Rizkyanti, C. A., Wahyuni, C., & Alatas, S. (2021). *Empathy and Defender Role in Bullying at School : Student-Teacher Relationship as Mediator*. 19(54), 227–246.
- RubyLang. (2021). *Tentang Ruby*. <https://www.ruby-lang.org/id/about/>
- Smith, P. K., Mahdavi, J., Carvalho, M., Fisher, S., Russell, S., & Tippett, N. (2008). Cyberbullying: Its nature and impact in secondary school pupils. *Journal of Child Psychology and Psychiatry and Allied Disciplines*, 49(4), 376–385. <https://doi.org/10.1111/j.1469-7610.2007.01846.x>
- Solekhah, A. M., Athikah, T. P., & Istiqomah, M. (2018). Faktor-Faktor yang Mempengaruhi Sikap Empati terhadap Perilaku Prososial pada Anak Sekolah Dasar. *Universitas Negri Semarang, 0291*, 86–90.
- Stauffer, M. E. (2016). *Laravel: Up and Running* (1st ed.). O'Reilly Media, Inc.
- Sumarna, Suhendry, M. R., Riana, E., Riyanto, V., & Nurdin, H. (2020). *Rancang Bangun Learning Management System Menggunakan Framework CodeIgniter Pada PT. Rekayasa Industri*. 7(1), 135–138.  
<https://doi.org/10.31294/jtk.v4i2>

- Sunyoto, A. (2010). Ajax: Asynchronous JavaScript and XML. *Jurnal Dasi*, 31(2), 32–35.
- Thoriqul Falahi, M. (2019). RANCANG BANGUN APLIKASI PAPERLESS OFFICE BERBASIS WEB SEBAGAI SISTEM PENGOLAHAN DAN PENCATATAN DATA MENGGUNAKAN RESTFUL API (Studi Kasus: QIS (Yasasan Quali International Surabaya)). *Jurnal Manajemen Informatika*, 9(2).
- Webpack. (2021). *Webpack#Concepts*. Webpack.Org.  
<https://webpack.js.org/concepts>
- Winarno, B. (2013). Ilmu Sosial Dan Ilmu Politik: Filsafat, Teori Dan Metodologi. *Jurnal Upnyk*, 17(1), 1–20.
- Yetti, W., & Ahyanuardi, A. (2020). Pengembangan Modul E-Learning Berbasis LMS Sebagai Media Interaktif Pada Pelajaran Simulasi Dan Komukasi Digital. *INVOTEK: Jurnal Inovasi Vokasional Dan Teknologi*, 20(3), 81–88.  
<https://doi.org/10.24036/invotek.v20i3.839>

## LAMPIRAN

### Lampiran 1. Surat Paklaring *Empathy Project*



### Empathy Project

Cinere Estate Blok A, Jl. Mangga No. 203, Depok, Jawa Barat  
16514

---

#### SURAT TUGAS

Yang bertanda tangan di bawah ini:

Nama : Dr. Charyna Ayu Rizkyanti, S.Psi., M.A  
Jabatan : Ketua Tim Modul Digital  
Alamat : Cinere Estate Blok A, Jl. Mangga No. 203, Depok, Jawa Barat  
16514

Menerangkan dengan sebenarnya bahwa yang bersangkutan di bawah ini:

Nama : Abdul Hakim Ghaniy  
Alamat : Bekasi  
Jabatan : Full-stack Web Developer

Adalah benar merupakan tim di modul digital Empathy Project, berlokasi di Cinere Estate Blok A, Jl. Mangga No. 203, Depok, Jawa Barat 16514

Terhitung sejak 28 Juli 2021 hingga 24 Desember 2021 dengan jabatan terakhir sebagai Full-stack Web Developer

Saudara Abdul Hakim Ghaniy telah bekerja dengan penuh dedikasi dan loyalitas yang baik kepada tim Inovasi Modul Digital 2021 serta tidak pernah terlibat hal-hal negatif yang membawa pengaruh buruk bagi Empathy Project.

Melalui pernyataan ini, kami berterima kasih atas kinerja yang telah dilakukan selama ini.

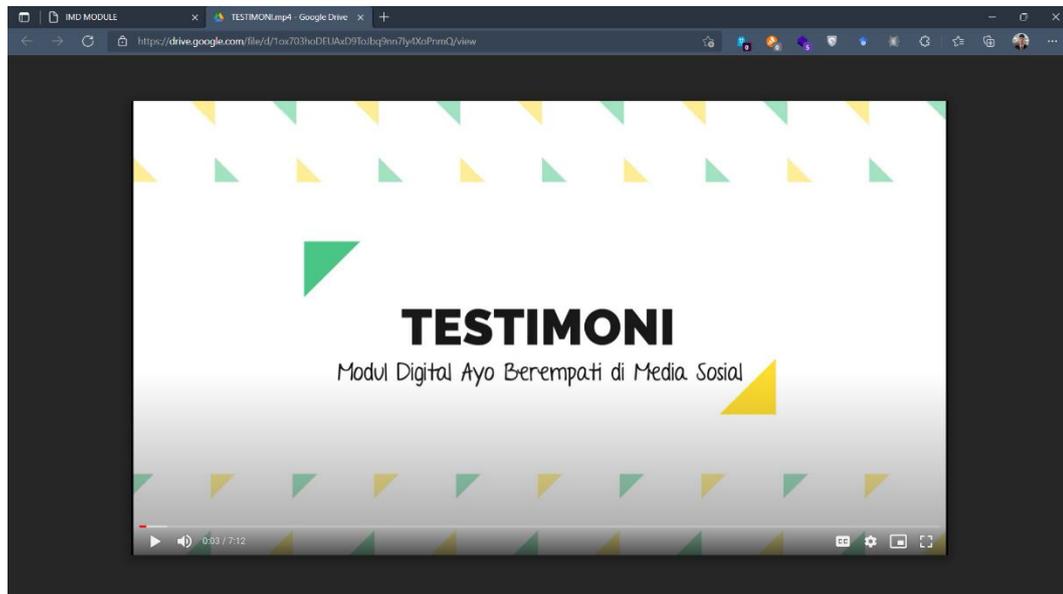
Demikian surat ini dibuat untuk dipergunakan sebagaimana mestinya.

Depok,

Dr. Charyna Ayu Rizkyanti, S.Psi., M.A

Ketua Tim Modul Digital

## Lampiran 2. Testimoni Penggunaan Modul Pembelajaran Empati



Testimoni penggunaan Modul Digital dapat dilihat pada tautan berikut

<https://drive.google.com/file/d/1ox703hoDEUAXD9ToJbq9nn7Iy4XoPnmQ/view>

Lampiran 3. *Source Code* pada Github.

*Source Code* pada penelitian ini merupakan produk dengan lisensi *closed source*. Pemberian tautan hanya sebagai bukti bahwa penelitian ini menggunakan *source* asli yang dibuat oleh peneliti. Kode sumber tidak dapat dilihat secara publik.

<https://github.com/GhaniyKie/IMD-Project-2021>