

TUGAS AKHIR

**PENGEMBANGAN *E-LEARNING* BERBASIS WEB MENGGUNAKAN
TEKNOLOGI MERN GUNA Mendukung Pembelajaran
*FRONTEND DEVELOPMENT***



Zuleriqhbal Hendri

193510245

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS ISLAM RIAU
PEKANBARU
2023**

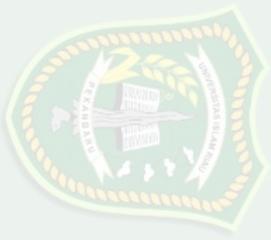
**UNIVERSITAS
ISLAM RIAU**

DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU

Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin



KATA PENGANTAR

Puji syukur penulis panjatkan pada Allah SWT atas segala rahmat dan karunia-Nya, karena atas izin Allah SWT penulis dapat mengerjakan proposal skripsi yang berjudul **"Pengembangan *E-Learning* Berbasis WEB Menggunakan Teknologi MERN Guna Mendukung Pembelajaran *Frontend development*"** ini dengan tepat waktu.

Dalam mengerjakan skripsi ini penulis menyadari bahwa banyak mendapatkan tantangan dan hambatan. Oleh karena itu penulis ingin mengucapkan rasa terima kasih kepada pihak yang sudah membantu, mengajari dan memberikan arahan dalam pembuatan skripsi ini.

Terlepas dari skripsi yang penulis kerjakan, penulis menyadari bahwa skripsi yang dibuat masih jauh dari kata kesempurnaan baik dari segi penulisan dan segi materi. Berbagai kritik yang bersifat konstruktif dari pembaca sangat penulis harapkan untuk dapat membantu menyempurnakan proposal skripsi ini. Sekian dari saya semoga skripsi yang penulis kerjakan dapat memberikan ilmu dan manfaat yang sebesar-besarnya buat pembaca.

Pekanbaru, Agustus 2023

Zuleriqhbal Hendri

**UNIVERSITAS
ISLAM RIAU**

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI.....	ii
DAFTAR GAMBAR	v
DAFTAR TABEL.....	ix
ABSTRAK.....	x
SUMMARY.....	xi
BAB I.....	1
1.1 Latar Belakang	1
Perkembangan teknologi saat ini berkembang dengan sangat pesat.....	1
1.2 Identifikasi Masalah	3
1.3 Rumusan Masalah	3
1.4 Batasan Masalah.....	4
1.5 Tujuan Penelitian.....	4
1.6 Manfaat Penelitian.....	5
BAB II.....	6
2.1 Tinjauan Pustaka	6
2.2 Dasar Teori.....	8
2.2.1 Frontend.....	8
2.2.2 Engineer.....	8
2.2.3 Web.....	8
2.2.4 <i>E-Learning</i>	9
2.2.5 Javascript.....	9
2.2.6 MongoDB	9
2.2.7 ExpressJS.....	11
2.2.8 ReactJS	11





Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin

UNIVERSITAS ISLAM RIAU

PERPUSTAKAAN SOEMAN HS

- 2.2.9 NodeJS 12
- 2.2.10 TailwindCSS 14
- 2.2.11 DFD (*Data Flow Diagram*) 14
- 2.3 Kerangka Pemikiran 15
- BAB III 17
- 3.1 Metode Penelitian 17
 - 3.1.1 Metode Pengumpulan *Data* 17
 - 3.1.2 Objek Penelitian 18
 - 3.1.3 Metode Pengembangan Sistem 19
- 3.2 Support 21
 - 3.2.1 Perangkat Keras (*Hardware*) 21
 - 3.2.2 Perangkat Lunak (*Software*) 21
- 3.3 Analisis Kebutuhan Sistem 22
- 3.4 Pengembangan dan Perancangan *System* 24
 - 3.4.1 Analisis Sistem 25
 - 3.4.2 Konteks Diagram 26
 - 3.4.3 *Hierarchy Chart* 28
 - 3.4.4 Perancangan *RESTful API* 36
 - 3.4.5 Fungsi tiap *Endpoint API* pada Sistem *E-Learning* 36
 - 3.4.6 Rancangan *Database* 43
 - 3.4.7 Desain antar muka 47
 - 3.4.8 *Flowchart* 56
- BAB IV 61
- 4.1 Hasil Penelitian 61
- 4.2 Pengujian *Blacbox* 61
 - 4.2.1 Pengujian Halaman *Register* 61



4.2.2 Pengujian Halaman *Login*65

4.2.3 Pengujian Halaman Lupa *Password*..... 71

4.2.4 Pengujian Halaman *Dashboard*.....74

4.2.5 Pengujian *Edit Data User*..... 103

4.2.6 Pengujian *Download Materi*..... 110

4.3 Hasil Pengujian *Blackbox*..... 112

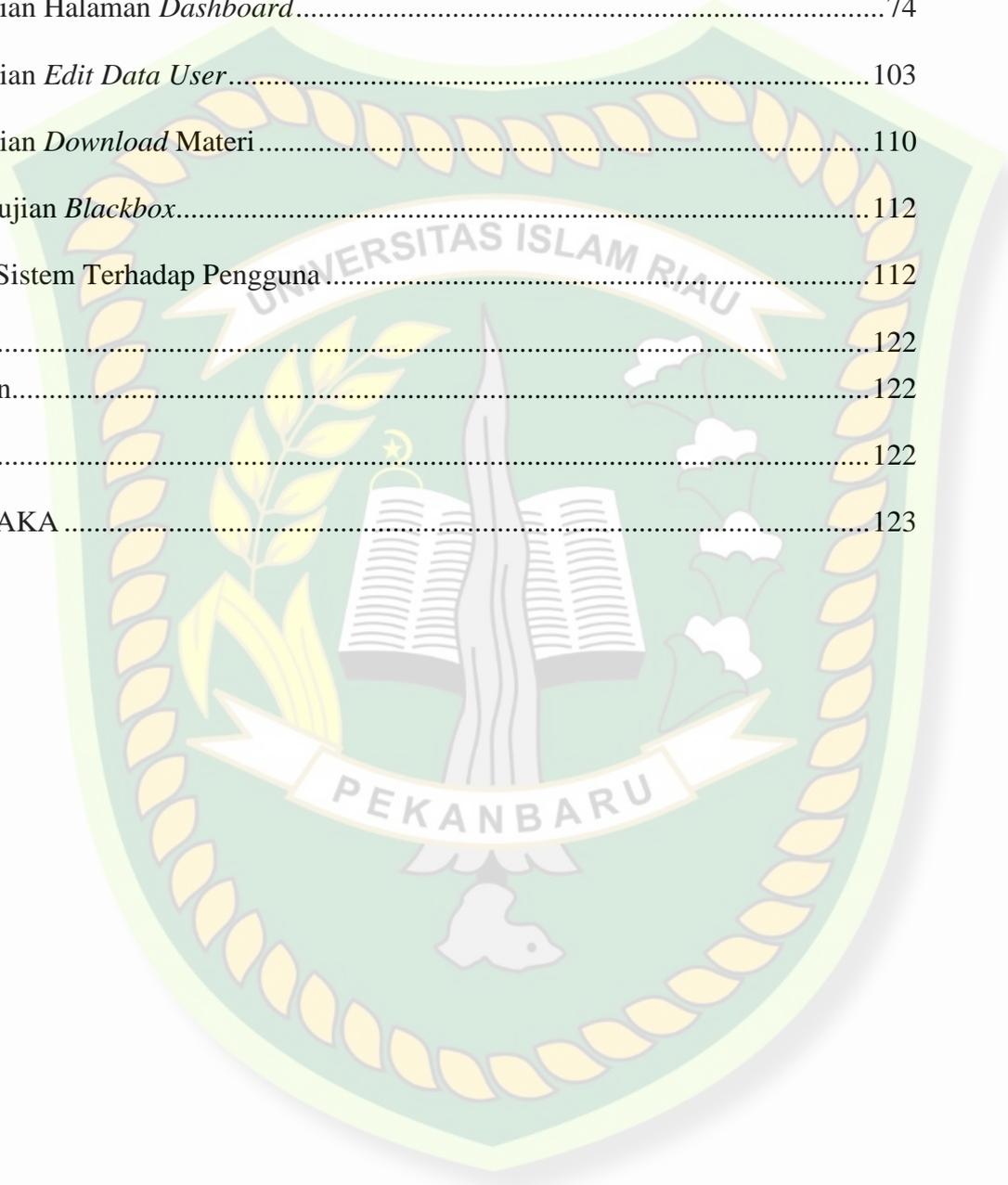
4.4 Pengujian Sistem Terhadap Pengguna 112

BAB V 122

5.1 Kesimpulan..... 122

5.2 Saran..... 122

DAFTAR PUSTAKA 123



UNIVERSITAS ISLAM RIAU

Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin

UNIVERSITAS ISLAM RIAU
PERPUSTAKAAN SOEMAN HS
BUKUMEN INI ADALAH ARSIP MILIK :



DAFTAR GAMBAR

Gambar 2.1 Perbedaan <i>Database</i> Relasi Dengan Orientasi Dokumen	10
Gambar 2.2 Kerangka Pemikiran.....	16
Gambar 3.1 grafik peminat bidang <i>frontend enginner</i>	18
Gambar 3.2 grafik angkatan yang berminat.....	19
Gambar 3.3 <i>Waterfall Model</i>	19
Gambar 3.4 <i>Data</i> Kesulitan Pekerjaan Setelah Lulus.....	22
Gambar 3.5 Analisis <i>Data</i> Kesulitan Sumber Materi	23
Gambar 3.6 <i>use case diagram</i>	26
Gambar 3.7 Konteks Diagram	27
Gambar 3.8 <i>Hierarchy Chart</i>	29
Gambar 3.9 DFD Level 1	30
Gambar 3.10 DFD level 2 proses 1.....	32
Gambar 3.11 DFD level 2 proses 2.....	33
Gambar 3.12 DFD level 2 proses 3.....	34
Gambar 3.13 DFD Level 2 Proses 4	35
Gambar 3.14 Cara Kerja <i>RESTful API</i>	36
Gambar 3.15 Konsep Penerapan LDM.....	43
Gambar 3.16 <i>Document Admin</i>	44
Gambar 3.17 <i>Document</i> Pelajar	45
Gambar 3.18 <i>Document</i> Pengajar	45
Gambar 3.19 <i>Document</i> Materi	46
Gambar 3.20 <i>Document File</i> Tugas Akhir Pelajar	47
Gambar 3.21 <i>Document</i> Nilai Siswa.....	47
Gambar 3.22 <i>Login Page</i>	48
Gambar 3. 23 <i>Register Page</i>	48
Gambar 3.24 <i>forget password page</i>	49
Gambar 3.25 <i>Dashboard Admin</i>	50
Gambar 3.26 Desain <i>Dashboard Pengajar</i>	51
Gambar 3.27 <i>Dashboard</i> Pelajar	51
Gambar 3.28 Antarmuka Ubah <i>Data</i>	52
Gambar 3.29 Antarmuka Ubah Materi	53



Gambar 3.30	Antarmuka Materi Pembelajaran	53
Gambar 3.31	Antarmuka <i>Detail Student</i>	54
Gambar 3.32	<i>Login Group Chat</i>	55
Gambar 3.33	Tampilan <i>group chat</i>	55
Gambar 3.34	<i>Flowchart Admin</i>	57
Gambar 3.35	<i>Flowchart Pengajar</i>	59
Gambar 3.36	<i>Flowchart Student</i>	60
Gambar 4.1	<i>Register Page</i>	62
Gambar 4.2	Peringatan <i>Error Fill Out</i>	62
Gambar 4.3	<i>Error Not Mail UIR</i>	63
Gambar 4.4	Halaman <i>Home Pelajar</i>	64
Gambar 4.5	<i>login page</i>	66
Gambar 4.6	<i>error form</i>	66
Gambar 4.7	<i>error message data not found</i>	67
Gambar 4.8	<i>Dashboard Pelajar</i>	68
Gambar 4.9	<i>Dashboard Pengajar</i>	69
Gambar 4.10	<i>Dashboard Admin</i>	69
Gambar 4.11	Halaman <i>Lupa password</i>	72
Gambar 4.12	<i>success message</i>	72
Gambar 4.13	<i>send to mail</i>	73
Gambar 4.14	<i>Send Data to Email</i>	73
Gambar 4.15	Lihat <i>Profile User</i>	75
Gambar 4.16	<i>Profile Pengajar</i>	75
Gambar 4.17	<i>Detail Pelajar Belum Dinilai</i>	76
Gambar 4.18	nilai pelajar dibawah kriteria.....	77
Gambar 4.19	<i>Profile Pelajar Memenuhi Kriteria</i>	77
Gambar 4.20	<i>Pop up</i> memberikan sertifikat	78
Gambar 4.21	Sukses kirim sertifikat pelajar	79
Gambar 4.22	<i>Remove User Pengajar</i>	80
Gambar 4.23	Hapus <i>User Pelajar</i>	80
Gambar 4.24	Tombol Tambah <i>Register</i>	81
Gambar 4.25	<i>Register Pengajar</i>	81
Gambar 4.26	<i>Error Message</i>	82
Gambar 4.27	Mengisi Semua <i>Field</i>	82



Gambar 4.28	Berhasil Mendaftarkan Akun	83
Gambar 4. 29	<i>Error Message Email</i> Sudah Digunakan.....	83
Gambar 4.30	<i>Dashboard</i> Pengajar	86
Gambar 4.31	<i>Error Messages</i>	87
Gambar 4.32	<i>Field</i> Terisi Semua	88
Gambar 4.33	Materi Baru di <i>Menu</i>	88
Gambar 4.34	<i>Field Edit</i> Materi	89
Gambar 4.35	Ubah Deskripsi.....	89
Gambar 4.36	Berhasil Ubah Dekripsi Materi	90
Gambar 4.37	<i>Remove</i> Materi	91
Gambar 4. 38	Success Remove Materi	91
Gambar 4.39	Halaman <i>Detail</i> Pelajar	92
Gambar 4.40	Halaman Detail Pelajar Mengerjakan <i>Task</i>	93
Gambar 4.41	Pesan Peringatan <i>Form</i>	93
Gambar 4.42	<i>Form</i> Berhasil <i>Input</i> Nilai	94
Gambar 4.43	<i>Form</i> Soal Tugas Akhir.....	95
Gambar 4.44	<i>Field</i> Tugas Akhir Tidak Terisi.....	95
Gambar 4.45	<i>Field</i> Tugas Akhir Terisi.....	96
Gambar 4.46	<i>Menu</i> Tugas Akhir.....	96
Gambar 4.47	<i>Dashboard</i> Pelajar.....	99
Gambar 4.48	Pesan <i>Warning Field Link Github</i>	100
Gambar 4.49	Pesan <i>Warning Field File</i>	100
Gambar 4.50	Tombol Cetak Sertifikat.....	101
Gambar 4.51	Berhasil <i>Download</i> Sertifikat.....	102
Gambar 4.52	Tombol <i>Edit Data User</i> Pelajar.....	104
Gambar 4.53	Tampilan Ubah <i>Data User</i>	104
Gambar 4.54	Pesan <i>Warning Edit Data User</i>	105
Gambar 4. 55	Mengisi Semua <i>Field</i>	105
Gambar 4.56	Sukses Mengubah <i>Data</i>	106
Gambar 4.57	Tampilan Ubah <i>Data</i> Pengajar.....	107
Gambar 4.58	Tampilan Ubah <i>Data</i> Pengajar.....	107
Gambar 4.59	Pesan <i>Warning Field</i>	108
Gambar 4.60	<i>Field</i> Terisi Ubah <i>Data</i>	108
Gambar 4.61	Tampilan Berhasil Ubah <i>Data</i>	109



Gambar 4.62 Tombol *Download Materi*..... 111

Gambar 4.63 Hasil Sukses *Download*..... 111



UNIVERSITAS ISLAM RIAU

DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



DAFTAR TABEL

Tabel 2.1	Definisi Komponen DFD	14
Tabel 3.1	Perangkat Keras (<i>Hardware</i>)	21
Tabel 3.2	Perangkat Lunak (<i>Software</i>).....	21
Tabel 3.3	Fungsi Endpoint API pada Sistem <i>E-Learning</i>	36
Tabel 4.1	Pengujian Halaman <i>Register</i> Pelajar	64
Tabel 4.2	Pengujian Halaman <i>Login</i>	70
Tabel 4.3	pengujian halaman lupa <i>password</i>	73
Tabel 4.4	Pengujian Halaman <i>Dashboard Admin</i>	84
Tabel 4.5	Pengujian Halaman <i>Dashboard</i> Pengajar	97
Tabel 4.6	Pengujian <i>Form</i> Tugas Akhir.....	102
Tabel 4.7	Rincian Pengujian <i>Edit Data User</i>	109
Tabel 4.8	Pengujian <i>Download</i> Materi	111

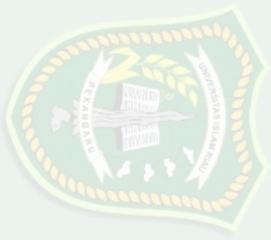
**UNIVERSITAS
ISLAM RIAU**

ABSTRAK

Kesulitan yang dialami oleh banyak pelajar khususnya yang berada di prodi teknik informatika atau ilmu komputer dalam mempelajari ilmu dibidang *frontend development* membuat mereka tidak benar-benar bisa memahami cara menjadi ahli *tech engineer* yang berperan sebagai *frontend development*. Untuk itu sangat diperlukan suatu sistem *E-Learning* yang bisa membantu para pelajar supaya meraih tujuan karir dibidang *frontend development*, sistem ini nantinya akan memiliki berbagai fitur yang dapat membantu proses pembelajaran yang efektif, salah satunya ialah fitur *chat group* yang memungkinkan antar *user* baik pelajar dan pengajar dapat berinteraksi untuk membahas mengenai modul pembelajaran yang ada di sistem *E-Learning*. Dalam pengembangan sistem ini tentunya akan diuji dengan konsep *blackbox* supaya sistem benar-benar teruji dan bisa dipergunakan, selain itu dalam pengembangannya akan menggunakan metode *waterfall*. Sistem *E-Learning* yang dibangun dapat memberikan informasi bahan pembelajaran yang mudah dimengerti oleh pemula, selain itu sistem ini juga bisa diakses secara *asnychronus* sehingga pelajar bisa mengaksesnya kapanpun dan dimanapun berada asalkan tetap terhubung dengan internet, untuk membuktikan efisiensinya, sistem ini telah diuji oleh beberapa pelajar yang ada di universitas islam riau prodi teknik informatika serta diolah dengan menggunakan metode *5 likert*, hasil dari pengolahan datanya menghasilkan persentase 85% yang artinya sistem ini layak dan sangat diminati para pelajar.

Kata Kunci: *frontend development, waterfall, E-Learning, blackbox.*

UNIVERSITAS
ISLAM RIAU



SUMMARY

The difficulties experienced by many students, especially those in information engineering or computer science study programs, in learning knowledge in the field of frontend development mean that they cannot really understand how to become expert tech engineers who play a role in frontend development. For this reason, it is very necessary to have an E-Learning system that can help students achieve career goals in the field of frontend development. This system will have various features that can help with an effective learning process, one of which is a chat group feature that allows users, both students and Teachers can interact to discuss learning modules in the E-Learning system. In developing this system, of course it will be tested using the black box concept so that the system is truly tested and can be used. Apart from that, the development will use the waterfall method. The E-Learning system that was built can provide learning material information that is easy for beginners to understand. Apart from that, this system can also be accessed asynchronously so that students can access it whenever and wherever they are as long as they remain connected to the internet. To prove its efficiency, this system has been tested by several students at the Islamic University of Riau informatics engineering study program and processed using the 5 Likert method, the results of the data processing produced a percentage of 85%, which means this system is feasible and very popular with students.

Keyword: frontend development, waterfall, E-Learning, blackbox

UNIVERSITAS
ISLAM RIAU



BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi saat ini berkembang dengan sangat pesat, hal ini dapat diikutsertakan dengan maju-nya dunia pendidikan. Salah satu bentuk majunya teknologi dalam bidang pendidikan ialah hadirnya metode pembelajaran secara *online* atau *E-Learning*. *E-Learning* adalah sebagai kegiatan belajar *asynchronous* melalui perangkat elektronik komputer yang memperoleh bahan belajar yang sesuai dengan kebutuhannya, dong (dalam kamarga, 2002). Model pembelajaran *E-Learning* terbukti mengalami peningkatan terhadap hasil belajar siswa dengan cara tenaga pendidik lebih kreatif dan inovatif dalam penggunaan media *E-Learning* apa saja agar peserta didik tidak cepat merasa bosan, serta penggunaan media pembelajaran *E-Learning* dapat digunakan kapan dan dimanapun berada (Rudi Haryadi & hanifa Nuraini Al Kansaa, 2021).

Hadirnya *E-Learning* saat ini bisa memberikan tingkat kemandirian para pelajar dalam hal teknis maupun pengalaman menggunakannya. Sistem *E-Learning* pada dasarnya diperuntukkan untuk menyebarkan informasi pembelajaran secara *online*, dengan hadirnya *E-Learning* informasi-informasi yang berkaitan dengan pembelajaran bisa diakses dengan mudah dan hal ini akan menjadi suatu keuntungan dari banyak kalangan dan pihak-pihak yang terlibat.



Dalam mempelajari ilmu dibidang *frontend engineer* tidaklah cukup apabila hanya mengandalkan buku dan materi yang disampaikan di perkuliahan, diperlukan suatu alat media elektronik agar pelajar bisa lebih spesifik mengetahui dan memahami secara terstruktur. *E-Learning* adalah salah satu media yang cocok sebagai pelengkap alat media pembelajaran untuk mendukung pemahaman terkait *frontend development*, namun seringkali *E-Learning* yang ada saat ini masih belum cukup untuk mendukung kegiatan belajar mahasiswa. Mahasiswa sering kali mendapati kendala terkait alur dari pembelajaran frontend yang masih tidak terstruktur selain itu belum didukung adanya mentor atau pelajar lainnya didalam suatu media *E-Learning* yang bisa memberikan arahan dalam mencapai ilmu yang diinginkan. Jika pun ada *E-Learning* diluar sana harus mengeluarkan dana yang tidak bisa dikatakan murah belum lagi mereka hanya akan diberi keterbatasan waktu sesuai jumlah dan tingkat pembayaran, karena hal tersebut ilmu serta pengetahuan yang didapat mungkin bisa saja tidak sepenuhnya dipahami. Oleh karena itu fenomena ini lah yang mendorong penulis untuk mencoba membuat suatu sistem yang dapat menutupi kekurangan-kekurangan tersebut maka dari-nya penulis akan memberikan judul pada penelitian dengan **“PENGEMBANGAN *E-LEARNING* BERBASIS WEB MENGGUNAKAN TEKNOLOGI MERN GUNA MENDUKUNG PEMBELAJARAN *FRONTEND DEVELOPMENT*”**.

Dalam proses perancangan akan digunakan teknologi MERN dalam pengimplementasian *E-Learning*, dimana dengan teknologi tersebut sistem yang dikembangkan hanya menggunakan bahasa pemrograman javascript,



selain itu diharapkan juga dapat menutupi kekurangan *web* yang sudah ada seperti keterbatasan dalam bahasa indonesia, komunikasi *chat group* dan penerapan test kemampuan sih pelajar yang nantinya akan dijadikan acuan pemahaman dan pengetahuannya dibidang *frontend development*. Setelah mahasiswa dan mahasiswi telah menyelesaikan *course* di *E-Learning* yang dibangun semoga dapat membantu mahasiswa sekalian bisa berkarir menjadi seorang *frontend development*.

1.2 Identifikasi Masalah

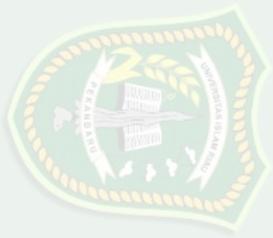
Berdasarkan latar belakang yang sudah penulis uraikan diatas maka terdapat permasalahan-permasalahan yang ter-identifikasi oleh si penulis yakni:

1. Sistem *E-Learning* yang ada belum memiliki fitur *chat group* yang dapat membantu para pelajar.
2. Masih minimnya web *E-Learning* yang terstruktur.
3. Banyaknya website *E-Learning* yang masih berbayar dan mahal sehingga membuat akses pembelajaran terbatas.

1.3 Rumusan Masalah

Terdapat beberapa rumusan masalah pada penelitian ini yakni:

1. Bagaimana cara mengembangkan web *E-Learning* menggunakan teknologi MERN.
2. Bagaimana cara memberikan kemudahan akses web *E-Learning* yang terstruktur.
3. Bagaimana cara agar si pengajar bisa menilai kemampuan mahasiswa dalam mempelajari *frontend development* melalui *E-Learning*.



4. Bagaimana agar pengajar dan pelajar dapat berkomunikasi didalam web *E-Learning*

1.4 Batasan Masalah

Setiap penelitian tentunya memiliki keterbatasan tak terkecuali pada penelitian ini, mulai dari biaya dan kemampuan yang penulis miliki. maka penulis membuat batasan masalah yakni:

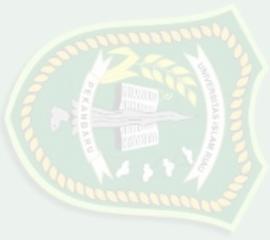
1. Penelitian ini hanya membahas pengembangan *web E-Learning* untuk mempelajari ilmu *frontend development*.
2. Sistem yang akan dibuat hanya dapat diakses oleh mahasiswa dan pengajar dengan menggunakan *email student* bagi mahasiswa.
3. Sistem yang akan dibuat menggunakan teknologi MERN yakni MongoDB, ExpressJS, ReactJS dan NodeJS.
4. Sistem hanya dapat diakses ketika pengguna memiliki jaringan internet.
5. Sistem yang dikembangkan hanya berbasis *web*.
6. Sistem hanya memiliki 3 hak akses yaitu *admin*, pengajar, dan pelajar.

1.5 Tujuan Penelitian

Adapun tujuan yang ingin dicapai oleh penulis dari penelitian ini adalah:

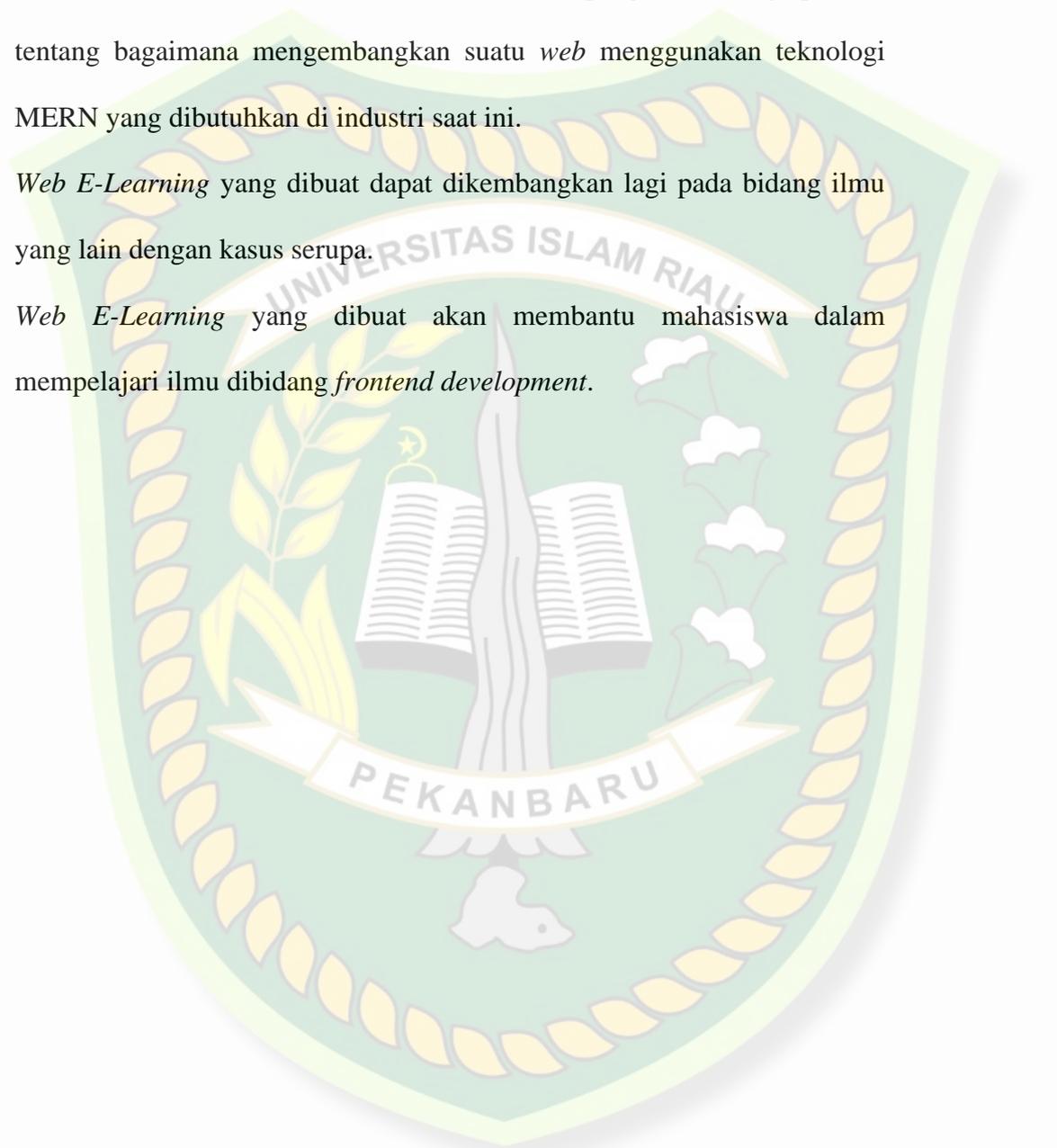
1. Menciptakan suatu *E-Learning* untuk pembelajaran bidang *frontend development*.
2. Menciptakan suatu *E-Learning* yang terstruktur untuk mempelajari ilmu *frontend development*.
3. Menciptakan web *E-Learning* yang dapat berkomunikasi antar pelajar dan mentor.
4. Menciptakan *web E-Learning* yang *friendly user*.





1.6 Manfaat Penelitian

1. Penelitian ini akan memberikan wawasan dan pengalaman bagi penulis tentang bagaimana mengembangkan suatu *web* menggunakan teknologi MERN yang dibutuhkan di industri saat ini.
2. *Web E-Learning* yang dibuat dapat dikembangkan lagi pada bidang ilmu yang lain dengan kasus serupa.
3. *Web E-Learning* yang dibuat akan membantu mahasiswa dalam mempelajari ilmu dibidang *frontend development*.

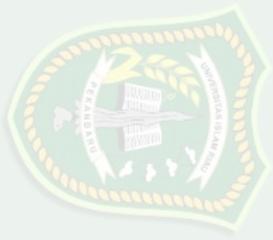


**UNIVERSITAS
ISLAM RIAU**

DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka

Untuk menambah literatur pada penelitian ini, maka penulis perlu untuk mengetahui dan memahami penelitian-penelitian terdahulu sebagai landasan yang akan dilakukan pada penelitian ini.

Ockhy Jey Fhiter Wassalam, dkk (2017) dengan judul “Implementasi dan Pengembangan Sistem *E-Learning* Berbasis *Web* Pada STMIK Muhammadiyah Paguyangan”. Pengembangan sistem *E-Learning* pada STMIK Muhammadiyah Paguyangan merujuk untuk memberikan kemudahan komunikasi antara mahasiswa dan dosen. Dengan sistem yang dikembangkan, dosen dapat memasukkan informasi terkait matakuliah dan modul pembelajaran sehingga mahasiswa dapat dengan mudah mengetahui informasi terbaru atau materi matakuliah, akibatnya mahasiswa akan menguasai bidang ilmu yang akan dikuasainya. Pada penelitian tersebut menggunakan metode penelitian sistem *waterfall*.

Alfath Yauma, dkk (2020) dengan judul “*Learning Management System* (LMS) pada *E-Learning* Menggunakan Metode *Agile* dan *Waterfall* berbasis *Website*”. Pada penelitian ini *system* yang dikembangkan diperuntukkan untuk mengatasi kegiatan belajar mengajar yang terhambat akibat pandemi *covid-19*.

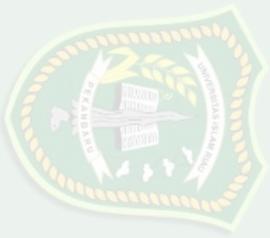
Adanya *system* ini para siswa dan guru dapat melakukan aktivitas belajar mengajar dengan lancar. Pada penelitian ini menggunakan metode *waterfall*

Agar pada saat pengembangan berlangsung mudah di-identifikasi apa saja kebutuhan pada *system* yang dirancang, selain itu juga menggunakan metode *agile* agar *system* yang sudah dirancang jika mengalami perbaikan atau pembaruan bisa lebih cepat dilakukan perubahan.

Agustini, dkk (2019) dengan judul “Sistem *E-Learning* Do’a dan Iqro’ dalam peningkatan proses pembelajaran pada TK Amal Ikhlas”. Penelitian ini berfokus untuk mengubah kegiatan proses belajar mengajar yang konvensional menjadi fleksibel yang bisa dilakukan dimana dan kapan saja. Sistem yang dikembangkan dapat membantu para murid dan orangtua untuk mendapatkan informasi-informasi terbaru dari sekolah melalui media *E-Learning* yang dirancang selain itu para murid bisa mengerjakan tes online dari rumah saja. Pada proses pengembangannya sistem ini menerapkan model SDLC sebagai acuan perancangan sistem *E-Learning*.

Rahmi Darnis, dkk (2022) dengan judul “Perancangan *E-Learning* berbasis *Responsive Web* Studi Kasus di Prodi Teknik Informatika Universitas Selamat Sri Kendal”. Penelitian yang dilakukan bertujuan untuk membantu dan mengatasi sulitnya kegiatan pembelajaran yang masih konvensional di universitas selamat sri kendal, terutama dikalangan mahasiswa yang sedang menjalani kuliah kerja profesi dan kuliah kerja nyata oleh karena-nya maka sih penulis akan menciptakan suatu *system* berbasis *web responsive* supaya dapat dijalankan di berbagai *platform*. Pada penelitian ini menggunakan metode *waterfall* kemudian dalam pengembangannya menggunakan bahasa pemrograman PHP dan MySQL sebagai *databasenya*.





2.2 Dasar Teori

2.2.1 Frontend

Frontend adalah segala sesuatu yang menghubungkan antara *user* dengan sistem *backend* (Arhandi, 2016). *Frontend* adalah sebutan untuk seseorang yang membuat aplikasi dari sisi *client* atau *user*, agar *user* dapat berinteraksi dengan aplikasi *server*. Untuk menjadi seorang *frontend*, seseorang diharuskan untuk menguasai kemampuan dasar *coding* seperti HTML, CSS dan JAVASCRIPT. Setelah menguasai kemampuan dasar *coding* seseorang juga harus bisa menguasai *RESTful API* agar aplikasi yang dibuat bisa berinteraksi dengan *server*.

2.2.2 Engineer

Engineer atau bahasa latinnya *ingenium* adalah julukan profesi yang diperuntukkan untuk seseorang yang bekerja dibidang perteknikan seperti merancang aplikasi atau membangun suatu benda yang didasari dengan ilmu matematika dan *sains*. seorang *engineer* biasanya bekerja dengan penuh ketelitian, ketekunan serta pemikiran yang kreatif agar apa yang mereka rancang bisa berfungsi dengan baik.

2.2.3 Web

Web adalah sejumlah halaman yang memiliki topik saling terkait antar satu halaman dan halaman yang lainnya, yang biasanya ditempatkan pada sebuah *server* web yang dapat diakses melalui jaringan internet maupun jaringan wilayah lokal (LAN) (Yeni Susilowati, 2019).

2.2.4 E-Learning

E-Learning adalah proses pembelajaran jarak jauh dengan menggabungkan prinsip-prinsip dalam proses pembelajaran dengan teknologi (Chandrawati, 2010). *E-Learning* adalah pengajaran dan pembelajaran yang didukung dan dikembangkan melalui teknologi dan media digital, dan juga merupakan salah satu bentuk dari konsep *distance learning* atau belajar jarak jauh (Abdul Barir Hakim, 2016).

2.2.5 Javascript

Javascript adalah sebuah bahasa *script* dinamis yang dapat dipakai untuk membangun interaktifitas pada halaman-halaman HTML statis. Ini dilakukan dengan menambahkan blok-blok kode javascript di hampir semua tempat pada halaman *web* (Siahaan & Rismon, 2020).

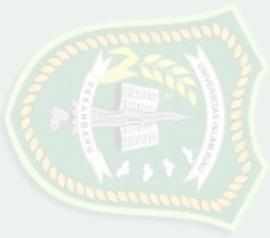
Javascript adalah bahasa *scripting* yang populer di internet dan dapat bekerja di sebagian besar *browser* populer seperti *Internet Explorer* (IE), *Mozilla Firefox*, *Netscape*, dan *opera* (sunnyoto, 2007).

Bahasa javascript juga memiliki beberapa fungsi yakni:

1. Dapat membangun aplikasi dari sisi *client*.
2. Dapat membangun aplikasi dari sisi *server*.

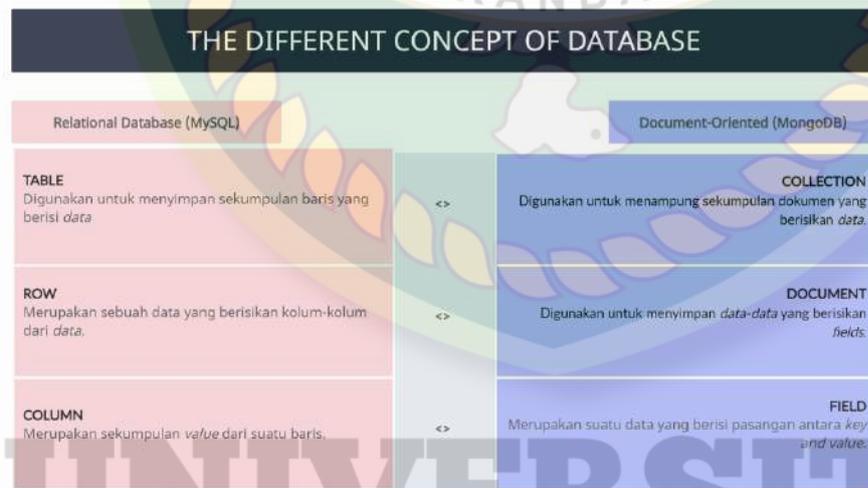
2.2.6 MongoDB

MongoDB (NoSQL) adalah *database open source* yang memiliki *high performance* dengan konsep manajemen *database* berorientasi dokumen (*document-oriented*). MongoDB pertama kali dibuat pada tahun 2007 oleh logen menggunakan bahasa pemrograman C++ dan baru dirilis



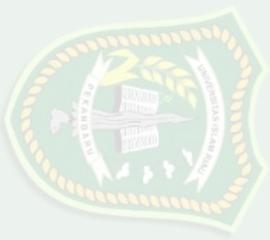
pada tahun 2009. MongoDB sendiri adalah *database* yang memiliki kecepatan yang lebih baik ketimbang SQL seperti MySQL, *document-oriented* yang ada di *database* berfungsi sebagai mengolah, menyimpan, dan mengatur *data* yang ada di MongoDB. MongoDB sangat baik apabila diintegrasikan dengan ReactJS.

Penggunaan pada MongoDB hanya menggunakan 2 *keywords* yakni *collection* yang di asumsikan sebagai folder dan *document* yang di asumsikan sebagai *file*. Penggunaan MongoDB sendiri sebenarnya hampir serupa dengan *relational database* biasa seperti MySQL yang membedakan hanyalah jika di *relational database* biasa kita menggunakan tabel untuk menyimpan beberapa baris, nah tabel yang ada di *relational database* itu serupa dengan *collection* yang ada di MongoDB sedangkan baris yang ada di *relational database* itu serupa dengan *document* yang ada di MongoDB.



Gambar 2.1 Perbedaan Database Relasi Dengan Orientasi Dokumen

MongoDB menggunakan *model* BSON atau *binary json* yang mempunyai konsep *key-value* yang ada di MongoDB dan setiap kali



pembuatan *database*-nya akan dibuatkan secara otomatis indeks *id*-nya secara unik.

2.2.7 ExpressJS

ExpressJS adalah sebuah *framework* atau kerangka kerja yang digunakan dalam pengembangan aplikasi sisi *server*. *expressJS* biasanya diinstall menggunakan *npm* yang ada di *nodeJS*, penggunaan dari *expressJS* sangat memudahkan *developer* dalam mengimplementasikan *http* method seperti *GET*, *POST*, *PUT*, *UPDATE* serta *DELETE*. Kelebihan yang dimiliki oleh *expressJS* yakni:

1. Sangat Fleksibel untuk membangun *request* dan *response* dari *user*.
2. Mudah dipelajari oleh *developer*.
3. Tidak memakan banyak *storage* sehingga dapat menghemat memori dan cepat di-implementasikan pada aplikasi *React*.
4. *Support* diintegrasikan dengan banyak *dependency* yang tersedia pada *NodeJS*.
5. *Support* terhadap banyaknya *middleware*.
6. Apabila terjadi *error* pada program, *expressJS* sangat cocok untuk mengatasi *error* dengan baik dan cepat.

2.2.8 ReactJS

ReactJS adalah sebuah *framework* yang dibuat oleh *facebook*. *React* bukanlah sebuah *framework MVC*. *ReactJS* adalah *library* yang bersifat *composable user interface*, yang artinya kita dapat membuat berbagai *UI* yang bisa kita bagi menjadi beberapa komponen (Musa, 2018).



ReactJS atau React merupakan *open-source javascript library* untuk mengembangkan antarmuka pengguna yang lebih interaktif dan mempermudah *developer* dalam perancangan aplikasi. ReactJS digunakan untuk menangani *view layer* pada aplikasi *single page* dan aplikasi *mobile*. ReactJS dikelola oleh Facebook, Instagram, dan komunitas para *developer* (Khuat, 2018).

ReactJs berusaha untuk memberikan kecepatan, kesederhanaan, dan skalabilitas. Beberapa fitur ReactJS yang biasa dikenal adalah JSX atau javascript XML. JSX adalah *extension* untuk sintaksis ECMAScript. JSX membantu *developer* pada saat mengembangkan UI di dalam javascript, dan juga dapat membantu *developer* ketika sedang melakukan *error debugging* (Khuat, 2018).

Javascript *library* ini sendiri telah digunakan oleh facebook untuk bagian *newsfeed* mereka. Selain itu, banyak situs-situs terkenal yang menggunakan ReactJS ini sebagai salah satu alat yang digunakan dalam mengembangkan situs tersebut, seperti Netflix, Paypal, Vevo, dan masih banyak lagi (M & Sonpatki, 2016).

2.2.9 NodeJS

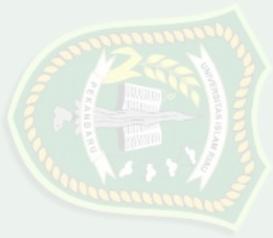
NodeJS adalah sebuah *framework* yang digunakan untuk melengkapi fungsi dari bahasa javascript baik penggunaan *frontend* maupun *backend*. NodeJS bukanlah bahasa pemrograman melainkan suatu *platform* yang berfungsi sebagai *runtime environment* atau interpreter yang dapat



mengeksekusi kode javascript di mana saja tanpa harus bergantung dengan *browser* lagi.

Alasan yang membuat javascript bisa dieksekusi di mana saja karena NodeJS dirancang dengan menggunakan *V8 engine* milik google yang serupa dengan *browser*, selain itu adanya *V8 engine* di nodeJS membuat peng-eksekusian javascript menjadi lebih cepat. NodeJS juga mempunyai *library* sendiri yang membuatnya tidak perlu lagi menggunakan apache ataupun *webserver* NGINX. Mengenai *V8-engine* yang terdapat pada NodeJS mempunyai 3 komponen, yakni:

1. **Compiler**, yang membuat kode javascript yang ditulis langsung diubah kedalam *executable file* yang langsung bisa dijalankan.
2. **Optimizer**, berfungsi membuat sebuah *abstract syntax tree* yang selanjutnya akan diganti menjadi *static single assignment* dan kemudian akan di optimasi kemudian.
3. **Garbage Collector**, adalah sebuah penyimpanan yang ada di *V8 engine*. Nantinya *v8 engine* akan mengubah menjadi 2, penyimpanan lama dan baru. Fungsi dari penyimpanan lama nantinya akan menampung *output* yang berasal dari *compiler*, nah nantinya saat penyimpanan baru sudah penuh *garbage collector* akan memindahkan objek-objek lama ke penyimpanan lama supaya kemampuan nodeJS tetap stabil dan ringan.



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU

UNIVERSITAS
ISLAM RIAU

2.2.10 TailwindCSS

TailwindCSS adalah sebuah *utility-first framework* yang bertujuan untuk membangun design antarmuka dengan lebih cepat. TailwindCSS berbeda dengan bootstrap karena penggunaan bootstrap sudah menyediakan beberapa komponen yang siap pakai, sedangkan tailwindCSS hampir serupa dengan CSS namun gaya penulisannya yang bisa langsung di *attribute tag* HTML. TailwindCSS menyediakan *low-level utility* yang mana penggunaan ini kita bisa membuat desain kita sendiri dengan khusus secara keseluruhan, tanpa harus keluar dari *file* yang kita kerjakan.

TailwindCSS saat ini banyak digunakan pada *website-website modern* karena memiliki kelebihan yang sangat menguntungkan buat para *developer* dan industri.

Penggunaan yang responsif, mudah dimodifikasi, serta *component-friendly*.

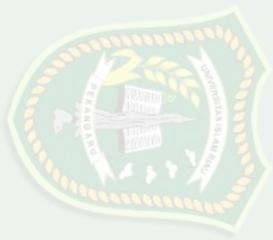
2.2.11 DFD (Data Flow Diagram)

Data flow diagram merupakan suatu diagram yang menggambarkan alir *data* dalam suatu entitas ke sistem atau ke entitas (Saputra, 2018).

Data flow diagram memiliki 2 *level* dalam perancangan atau penggambaran sistem, yaitu *level 0* dan *level 1* (Ummah dkk, 2019).

Tabel 2.1 Definisi Komponen DFD

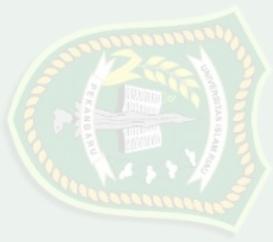
Simbol	Keterangan
	Eksternal <i>Entity</i> merupakan bagian eksternal terluar yang dapat menggambarkan sebuah manusia atau unit sistem yang terkait yang bisa berkomunikasi dengan sistem



	namun bagian luar <i>system</i> , simbol ini dapat memberikan <i>input</i> ke sistem atau menerima <i>output</i> dari sistem.
	Simbol ini adalah proses yang menggambarkan suatu proses pada sistem untuk melakukan pengolahan <i>data</i> , nantinya kegiatan yang dilakukan proses akan menghasilkan transformasi <i>data</i> .
	<i>Data Store</i> / penyimpanan yang digunakan untuk menyimpan <i>data</i> , dokumen, <i>file</i> yang berada pada sistem yang diperlukan.
	Simbol ini adalah sebuah aliran <i>data</i> yang menggambarkan jalannya suatu proses pada sistem.

2.3 Kerangka Pemikiran

Kerangka pemikiran adalah dasar dari sebuah penelitian untuk menyusun beberapa fakta-fakta, observasi dan kajian pustaka. Kerangka pemikiran bertujuan untuk menguraikan pola berpikir dari tujuan dan maksud untuk mengatasi permasalahan yang akan diteliti, misal dengan menguraikan objek-objek penelitian seperti permasalahan yang dihadapi, pendekatan, pengembangan yang dibutuhkan, serta hasil yang diharapkan.





Gambar 2.2 Kerangka Pemikiran

UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU

BAB III

METODOLOGI PENELITIAN

3.1 Metode Penelitian

Agar penulis bisa mencapai tujuan penelitian maka penulis harus melalui tahapan metode penelitian agar penulis memperoleh *data*, gambaran, serta kebutuhan yang dapat membantu mengatasi permasalahan, maka dari itu penulis telah menyusun metode penelitian diantaranya:

3.1.1 Metode Pengumpulan *Data*

Metode pengumpulan *data* bertujuan untuk memperoleh *data* dan informasi sesuai yang dibutuhkan oleh pengguna sistem, supaya peneliti bisa mengembangkan perangkat lunak sesuai dengan spesifikasi dan kebutuhannya. Beberapa teknik yang penulis terapkan dalam pengumpulan *data* berupa:

a. *Data Primer*

Dalam memperoleh *data primer* penelitian ini akan menggunakan hasil dari kuisioner yang akan disebar ke sejumlah mahasiswa teknik informatika uir nantinya hasil jawaban dari responden akan diolah menggunakan metode skala *likert 5 point*, yaitu: Sangat Tidak Setuju (1), Tidak Setuju (2), Ragu-ragu (3), Setuju (4), dan Sangat Setuju (5).

b. Studi Pustaka

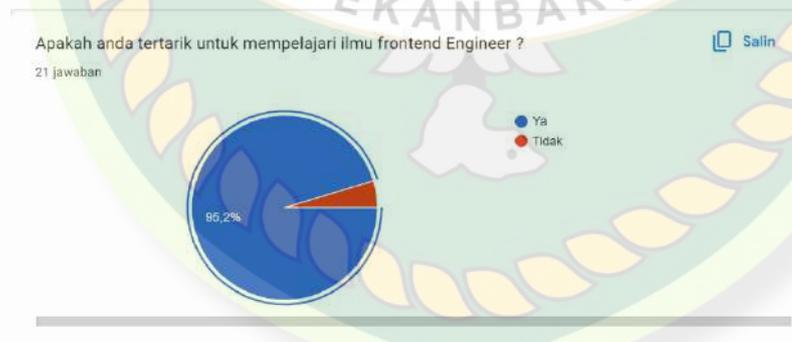
Studi pustaka adalah teknik pengumpulan *data* dengan mempelajari karya ilmiah, buku, dan laporan penelitian orang lain supaya



mendapatkan gambaran solusi dari topik permasalahan yang sedang diteliti

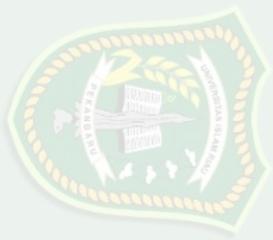
3.1.2 Objek Penelitian

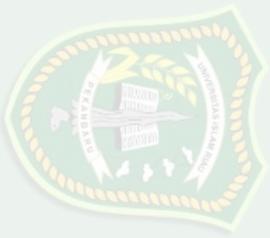
Objek Penelitian ini akan melibatkan 21 sampel mahasiswa prodi teknik informatika UIR dimana 5 mahasiswi (23 persen) dan 16 mahasiswa (77 persen) dengan hampir 92, 5 persennya menjawab “YA” hal ini didapatkan berdasarkan dari hasil kuisioner yang saya berikan mengenai pertanyaan terkait apakah pelajar tertarik dalam mempelajari ilmu dibidang *frontend development*, dan dari responden yang menjawab ternyata didapatkan hampir 98 persen-nya adalah angkatan 2019 dan 2 persen-nya berasal dari angkatan 2020. *Data* tersebut nantinya akan dipergunakan untuk mengukur serta mengidentifikasi kebutuhan penelitian yang dilakukan pada penelitian ini.



Gambar 3.1 grafik peminat bidang *frontend enginner*

UNIVERSITAS
ISLAM RIAU





Masukkan Angkatan Tahun berapa ?

21 jawaban

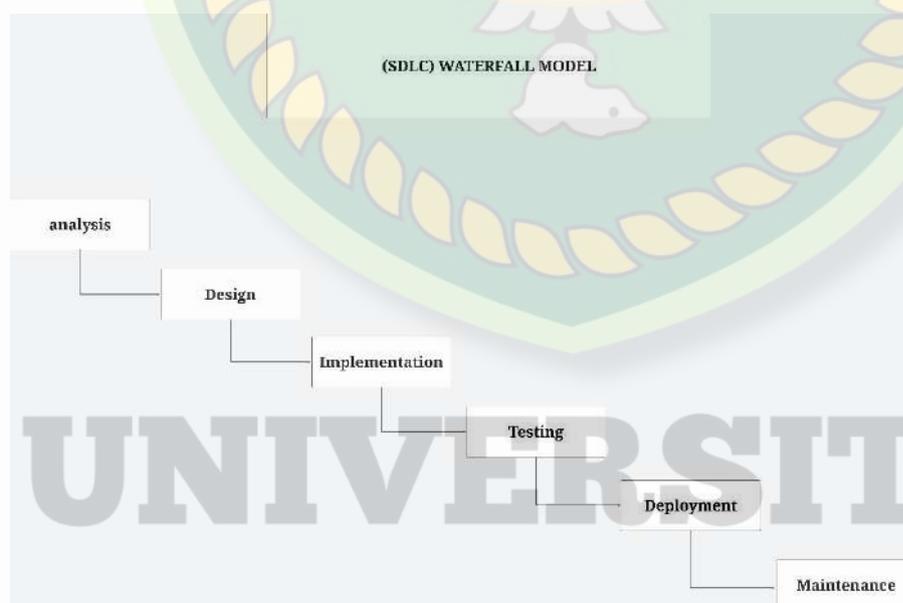
Salin



Gambar 3.2 grafik angkatan yang berminat

3.1.3 Metode Pengembangan Sistem

Pada pengembangan sistem *E-Learning* penulis akan menerapkan pendekatan *Software Development Management System* (SDLC) dengan *model waterfall* sebagai metodologi dalam pembuatan sistem *E-Learning*, ini karena *model waterfall* sangat efisien dan berurutan untuk setiap proses pengembangan sistem. Urutan dalam metode *waterfall* bersifat *serial* yang di mulai dari proses perencanaan, analisa, desain, dan implementasi pada sistem (Purnia, Rifai, & Rahmatullah, 2019; sasmito, 2017; wau, 2022).



Gambar 3.3 Waterfall Model

Berikut Penjelasan langkah-langkah penerapan *model waterfall* dalam proses pengembangan sistem *E-Learning*:

a. *Requirements Analysis and Definition*

Pada tahapan akan dilakukan analisa, perencanaan dan menentukan kebutuhan yang diperlukan untuk kemudian akan didefinisikan sebagai spesifikasi sistem yang dikembangkan pada *E-Learning*.

b. *System and Software Design*

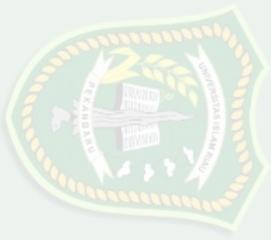
Kemudian setelah melakukan tahap analisis dan kebutuhan yang diperlukan pada sistem *E-Learning*, selanjutnya akan dilakukan perancangan dengan menentukan bahasa pemrograman, dan *tools* yang cocok untuk mendukung pengembangan sistem *E-Learning*.

c. *Implementation and Unit Testing*

Pada tahap ini, perancangan sistem *E-Learning* masuk ke dalam proses implementasi pengembangan dan melakukan pengujian untuk dapat benar-benar bisa di-integrasikan dan melangkah ke tahap berikutnya.

d. *Integration and System Testing*

Pada tahap ini setelah sistem sudah dikembangkan maka akan dilakukannya suatu *quality assurance* terkait sistem yang dibangun, guna memastikan apakah sistem yang dikembangkan layak dipergunakan dan sudah bisa



dioperasikan sesuai *requirement*, apabila layak maka sistem siap diluncurkan.

e. *Operation and Maintenance*

Pada tahap ini sudah dipastikan sistem bisa bekerja namun tetap akan dilakukannya pemantauan dan *fixing bug* bila ditemukan pada sistem.

3.2 Support

3.2.1 Perangkat Keras (*Hardware*)

Perangkat keras (*Hardware*) yang digunakan dalam pengembangan sistem *E-Learning* adalah sebagai berikut:

Tabel 3.1 Perangkat Keras (*Hardware*)

No	Spesifikasi	Keterangan
1	Tipe Laptop	Acer Swift 3
2	CPU	AMD Ryzen 5 3500U
3	SSD	512 GB
4	RAM	12000 MB

3.2.2 Perangkat Lunak (*Software*)

Perangkat lunak dan bahasa pemrograman yang penulis gunakan untuk proses pengembangan sistem *E-Learning* dalam penelitian ini adalah sebagai berikut:

Tabel 3.2 Perangkat lunak (*software*)

no	Spesifikasi	Keterangan
----	-------------	------------



1	Sistem Operasi	<i>Windows 11 64-bit</i>
2	<i>Framework</i>	<i>ReactJS, ExpressJS, TailwindCSS</i>
3	Bahasa Pemrograman	<i>Javascript</i>
4	<i>Database NoSQL</i>	<i>MongoDB</i>
5	<i>Design</i>	<i>Draw.io, Lucidchart</i>

3.3 Analisis Kebutuhan Sistem

Pada saat ini para pelajar tidak sedikit mengalami kebingungan dalam mengambil keputusan *job path* mereka setelah lulus kuliah nanti di jurusan teknik informatika hal ini dapat dibuktikan dengan data yang penulis dapatkan melalui kuesioner dari beberapa orang mahasiswa di universitas islam riau jurusan teknik informatika,



Gambar 3.4 Data Kesulitan Pekerjaan Setelah Lulus

Pada gambar 3.4 menunjukkan grafik bahwa 11 dari 11 orang pelajar yang dimintai jawaban mengatakan “YA” hal ini mungkin bisa disebabkan oleh beberapa faktor salah satunya mungkin belum adanya daya minat belajar siswa dan belum adanya sistem yang bisa membantu mahasiswa tersebut untuk menemukan peran sebagai seorang ahli IT.

banyak dari mereka tidak menemukan sistem yang bisa membantu mereka dalam mempelajari ilmu dibidang *frontend engineer*. Hal ini bisa diperkuat

dengan jawaban para mahasiswa melalui pertanyaan yang ada di kuesioner seperti pada gambar dibawah

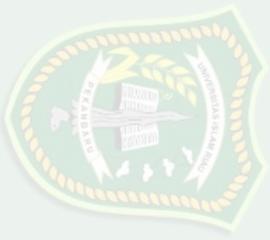


Gambar 3.5 Analisis Data Kesulitan Sumber Materi

Pada gambar 3.5 menunjukkan bahwa ada 10 orang pelajar yang mengatakan sulit untuk bisa mempelajari ilmu dibidang *frontend development*, meskipun pada dasarnya di era saat ini yang sudah serba digital tetapi masih belum adanya sarana yang benar-benar bisa menunjang kegiatan pembelajaran secara interaktif.

Selain itu, materi atau pembelajaran yang diberikan oleh *E-Learning* diluar sana juga diberi batasan waktu yang membuat pelajar harus buru-buru menyelesaikan materinya. Dan kadang tidak dapat mengakses materi berikutnya apabila batasan waktunya sudah habis, Ini yang dapat membuat para pelajar harus mengeluarkan uang lagi untuk dapat membuka dan mengaksesnya kembali.

Dari permasalahan tersebut akhirnya penulis akan mengembangkan sebuah sistem *E-Learning* Berbasis Web Menggunakan Teknologi MERN Stack untuk membantu para pelajar dalam mempelajari ilmu dibidang *frontend development*



secara terstruktur dan tanpa batasan akses. Sistem akan dibangun dengan se-ramah mungkin agar para mahasiswa bisa memahami materi yang disampaikan.

Kebutuhan fungsional berisikan proses-proses yang akan diberikan oleh sistem *E-Learning*. Nantinya sistem ini hanya memiliki 3 hak akses yakni admin, pengajar, dan para *user* sebagai pelajar *frontend development*. Berikut penjelasan mengenai 3 hak akses tersebut:

a. *Admin*

Sistem ini hanya memiliki 1 *admin* yang berfungsi sebagai menambah pengajar, menginputkan sertifikat serta menghapus para pelajar.

b. Pengajar

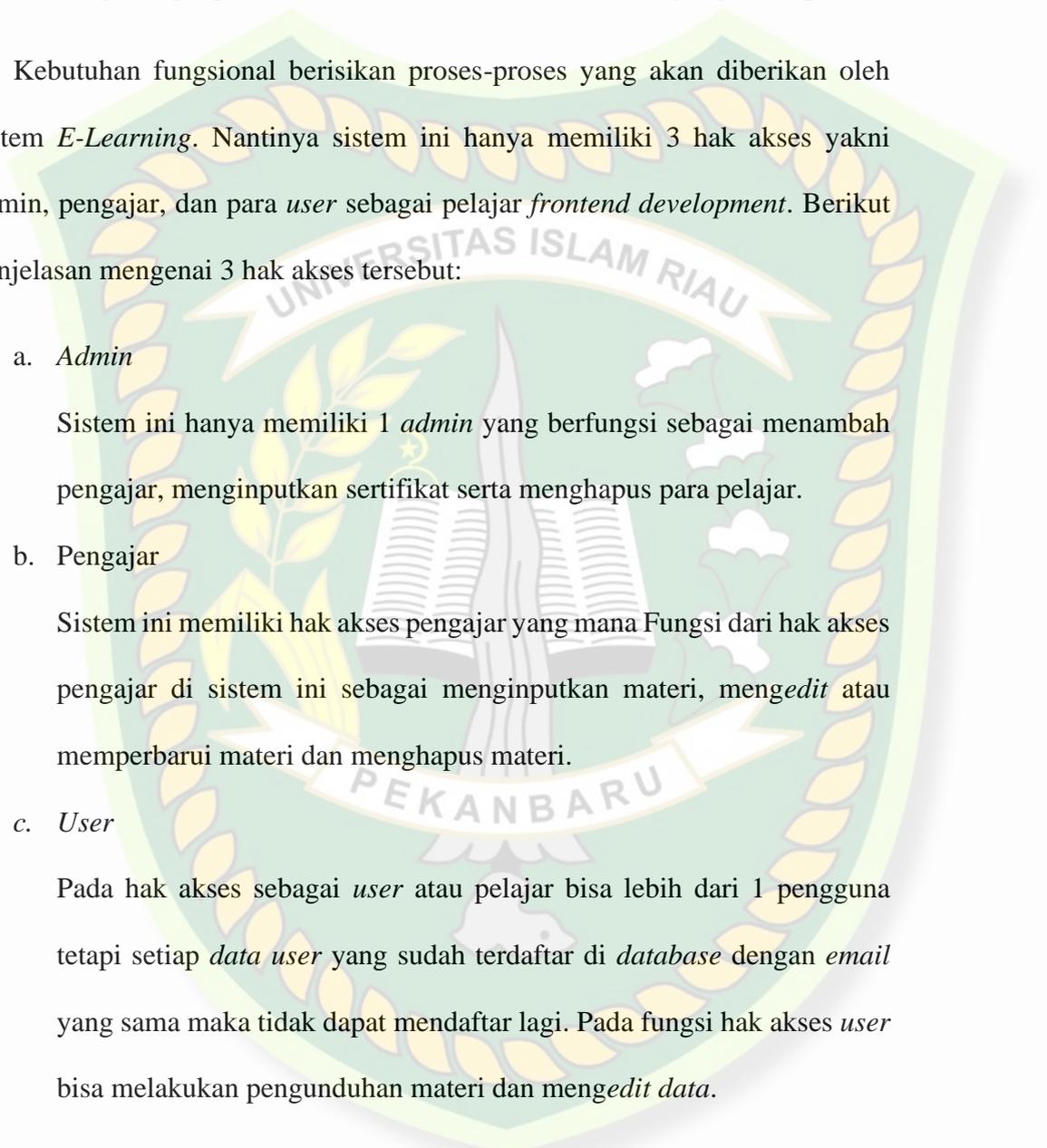
Sistem ini memiliki hak akses pengajar yang mana Fungsi dari hak akses pengajar di sistem ini sebagai menginputkan materi, *mengedit* atau memperbarui materi dan menghapus materi.

c. *User*

Pada hak akses sebagai *user* atau pelajar bisa lebih dari 1 pengguna tetapi setiap *data user* yang sudah terdaftar di *database* dengan *email* yang sama maka tidak dapat mendaftar lagi. Pada fungsi hak akses *user* bisa melakukan pengunduhan materi dan *mengedit data*.

3.4 Pengembangan dan Perancangan System

Supaya sistem yang dikembangkan bisa berjalan dengan semestinya tentu harus diperlukan perancangan terlebih dahulu. Perancangan mengenai sistem yang akan dikembangkan akan dibahas pada sub bab dibawah ini:



UNIVERSITAS
ISLAM RIAU

DOKUMEN INI ADALAH ARSIP MILIK :

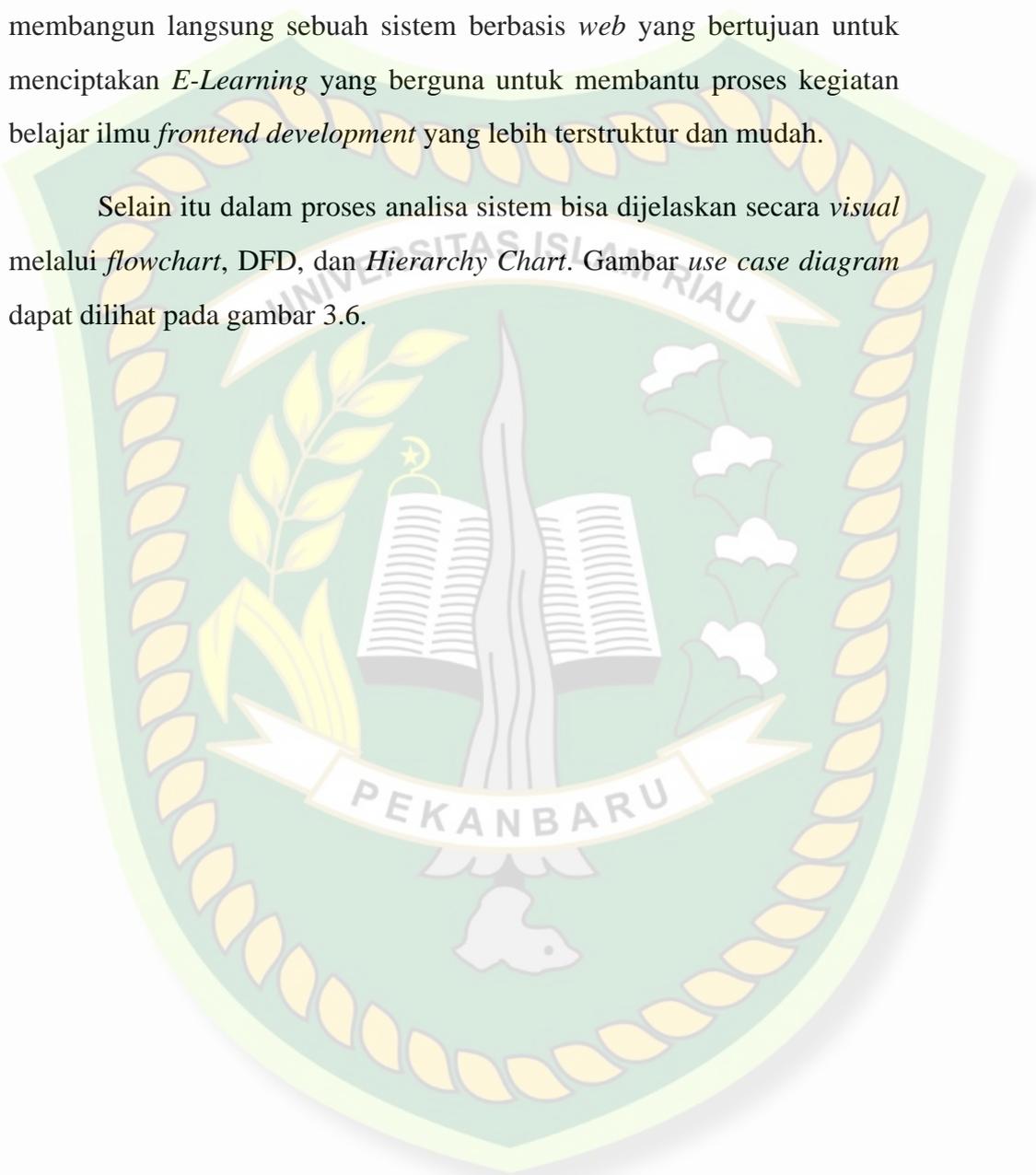
PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU

3.4.1 Analisis Sistem

Pada tahap ini akan melakukan sebuah analisis, *planning* dan membangun langsung sebuah sistem berbasis *web* yang bertujuan untuk menciptakan *E-Learning* yang berguna untuk membantu proses kegiatan belajar ilmu *frontend development* yang lebih terstruktur dan mudah.

Selain itu dalam proses analisa sistem bisa dijelaskan secara *visual* melalui *flowchart*, DFD, dan *Hierarchy Chart*. Gambar *use case diagram* dapat dilihat pada gambar 3.6.

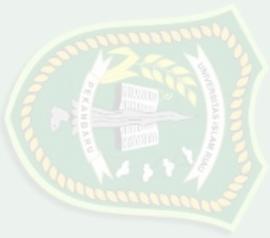


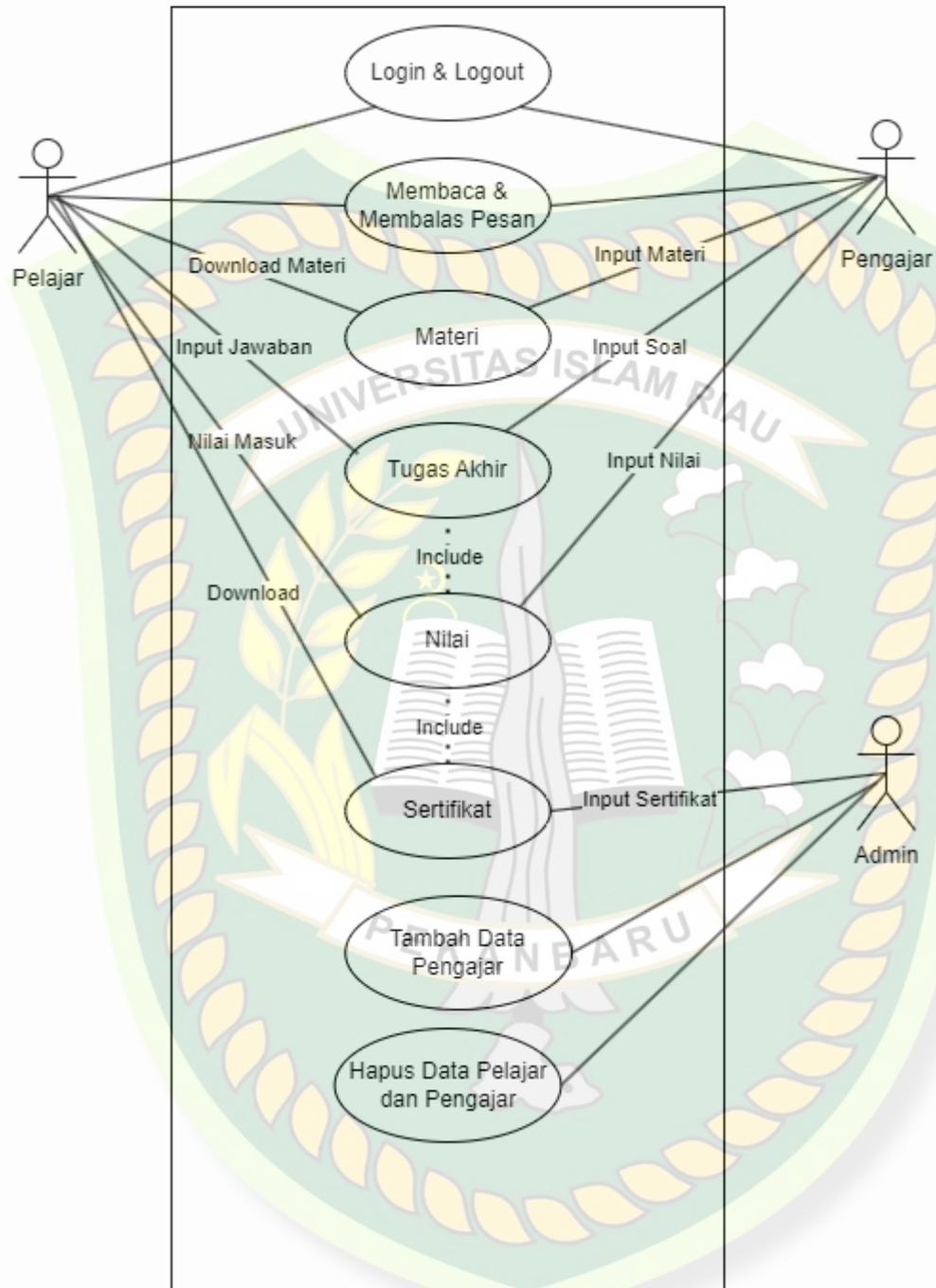
**UNIVERSITAS
ISLAM RIAU**

DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU





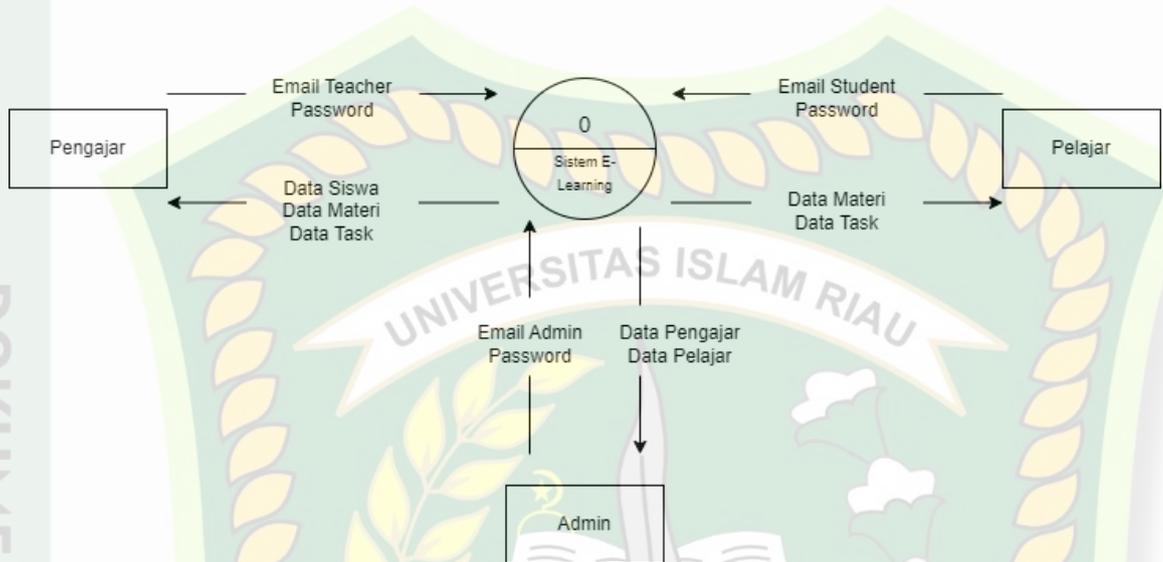
Gambar 3.6 use case diagram

3.4.2 Konteks Diagram

Konteks diagram merupakan sebuah *data flow* dengan tingkat paling tinggi dari sebuah *data flow diagram* (DFD), ini bertujuan untuk memberikan batasan-batasan konteks dan menetapkan sebuah sistem di



dalam suatu pemodelan. Penerapan konteks diagram di sistem *E-Learning* yang dikembangkan adalah sebagai berikut:



Gambar 3.7 Konteks Diagram

Adapun Penjelasan dari konteks diagram pada gambar 3.7 adalah sebagai berikut:

a. Admin

Berfungsi sebagai menambahkan *data* pengajar serta dapat melakukan penghapusan *data* pengajar dan pelajar. *Use case admin* juga dapat melakukan penginputan sertifikat apabila pelajar telah mendapatkan nilai yang diinputkan oleh si pengajar. Nantinya *use case admin* juga dapat melakukan pengiriman pesan dan membaca pesan.

b. Pengajar

Berfungsi sebagai memasukkan, merubah dan menghapus materi selain itu pengajar juga dapat melakukan perubahan atas *datanya* sendiri seperti nama dan *password*. Pengajar selain



fungsi yang sudah disebutkan diatas dapat melakukan penginputan nilai terhadap pelajar yang sudah menyelesaikan tugas akhir materi pelajar dan dapat menghapus jawaban pelajar serta melakukan pengiriman, membaca pesan.

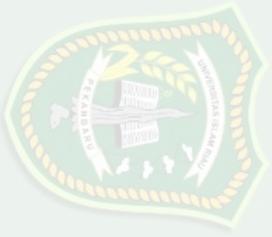
c. Pelajar

Use case pelajar berperan di sistem *E-Learning* untuk melakukan pengupload-an jawaban tugas akhir, mengubah *data* pribadi seperti nama dan password serta melihat dan mengunduh seluruh materi yang di *upload* oleh pengajar. Pelajar juga dapat mengirimkan dan membaca pesan.

3.4.3 *Hierarchy Chart*

Hierarchy Chart adalah sebuah konsep atau gambaran sistem dan sub-sub sistem yang menjelaskan setiap prosesnya dan setiap sub-sub sistem yang berada didalam sistem utama akan saling berelasi satu dengan yang lainnya, pembedanya hanyalah dari level prosesnya saja.

**UNIVERSITAS
ISLAM RIAU**



DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU



Dilarang mengemukakan dan memperbanyak sebagian atau seluruh karya tulis ini dalam bentuk apapun tanpa izin

DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU

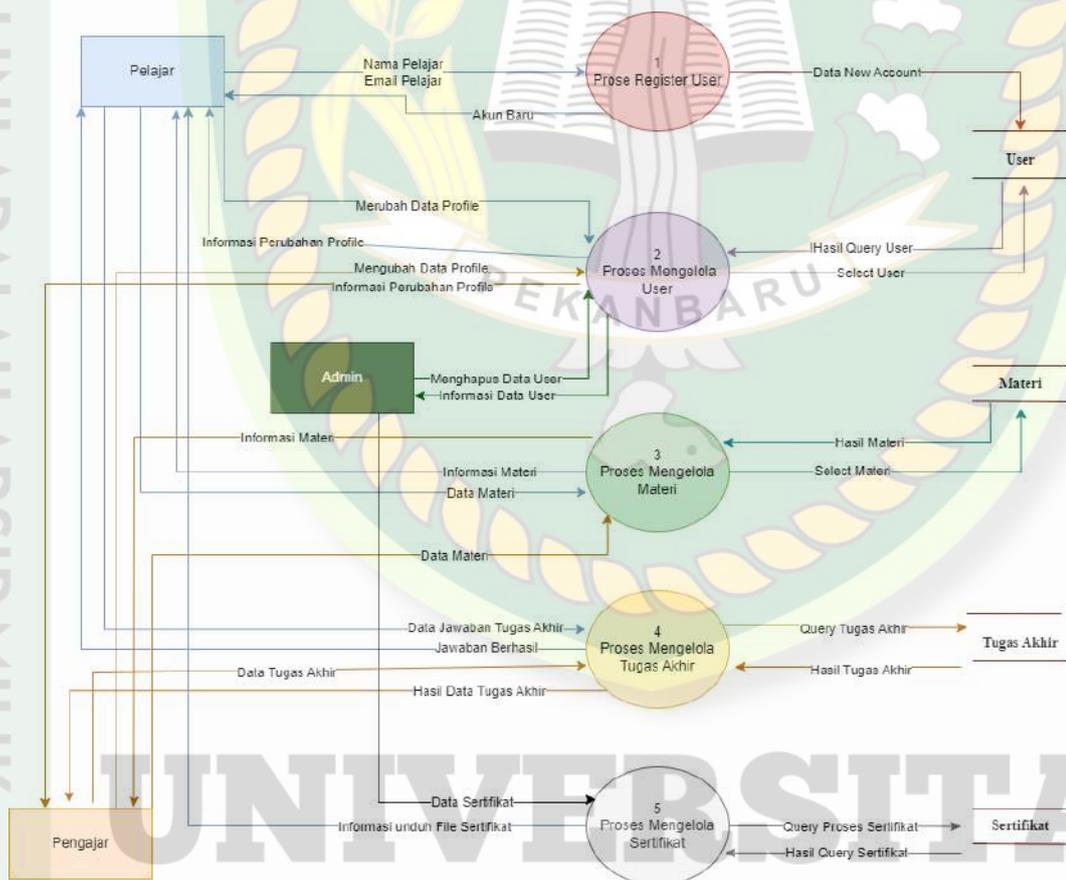


Gambar 3.8 Hierarchy Chart

UNIVERSITAS ISLAM RIAU

Pada gambar 3.8 terdapat 2 sub sistem pada sistem *E-Learning* yakni sub sistem *level 1* dan *level 2*. Pada sub sistem *level 1* memiliki 4 komponen yaitu *register user*, *mengelola user*, *mengelola materi* dan *mengelola tugas akhir*. Pada sub sistem *level 2* memiliki 10 komponen yaitu menambahkan *user* pengajar, *mengedit user*, *menghapus user*, *menambah materi*, *mengunduh materi*, *mengedit materi*, *menghapus materi*, *menambah tugas akhir*, *mengunduh tugas akhir*, *menghapus jawaban tugas akhir*. DFD (*Data Flow Diagram*)

3.4.3.1 DFD Level 1



Gambar 3.9 DFD Level 1

DFD *level 1* adalah proses keseluruhan yang terdapat pada sistem. Proses yang digambarkan dalam DFD hanya berupa simbol-

simbol tertentu. DFD *level* 1 yang terdapat pada sistem akan dibuat dan dapat dilihat seperti pada gambar 3.9.

Gambar 3.9 Merupakan DFD *level* 1 dari sistem *E-Learning* yang akan dikembangkan. Pada DFD tersebut memiliki 3 *terminator* atau hak akses yaitu *admin*, pengajar dan pelajar. Pada dasarnya sistem ini memiliki 5 proses utama, yakni:

- a. Proses *register user* yaitu merupakan sebuah proses ketika seorang pelajar mendaftar sebuah akun baru.
- b. Proses mengelola *user* yaitu merupakan sebuah proses dimana pelajar dan pengajar ingin melakukan perubahan akun dan bagi *terminator admin* proses ini digunakan untuk melakukan penghapusan akun pengajar dan pelajar.
- c. Proses mengelola materi yaitu merupakan sebuah proses ketika pengajar memasukkan, *mengedit*, menghapus sebuah materi dan pelajar dapat membaca materi serta mengunduh materi yang tersedia.
- d. Proses mengelola tugas akhir merupakan sebuah proses yang mana *terminator* pelajar dapat memasukkan *file* jawaban dari hasil pengerjaan soal tugas akhir, sedangkan bagi *terminator* pengajar dapat memasukkan soal tugas akhir.
- e. Proses mengelola sertifikat merupakan sebuah proses yang mana *admin* bisa menginputkan sertifikat yang diberikan kepada para pelajar yang sudah menyelesaikan materi dan



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

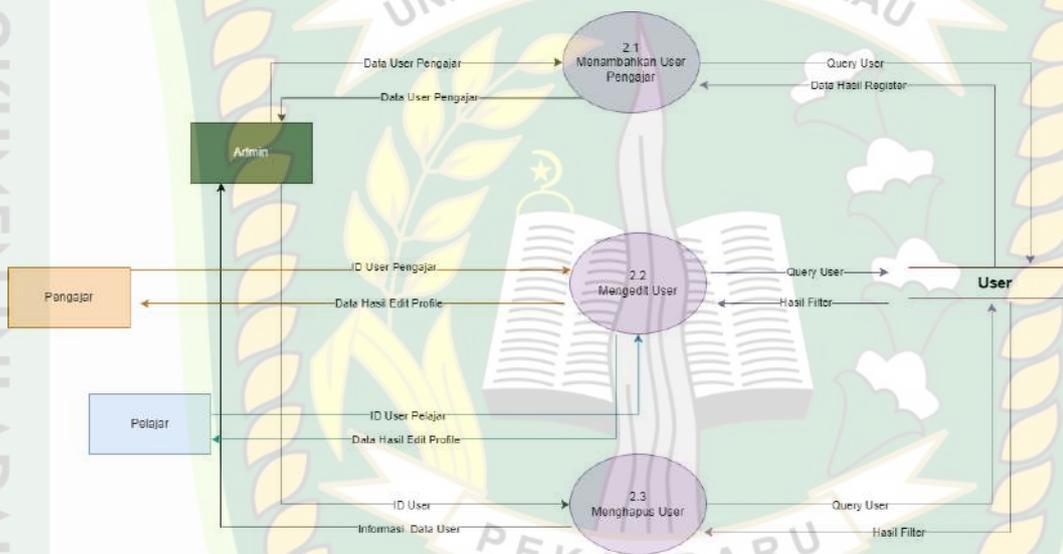
UNIVERSITAS ISLAM RIAU

UNIVERSITAS
ISLAM RIAU

mengerjakan soal tugas akhir, dan pelajar bisa mengunduh *file* sertifikat tersebut.

3.4.3.2 DFD Level 2 (Proses 1)

DFD level 2 proses 1 merupakan suatu proses yang memiliki 3 sub proses yaitu menambah materi, mengunduh materi, menghapus materi. *Data flow* diagramnya terdapat pada gambar 3.8 dibawah ini.

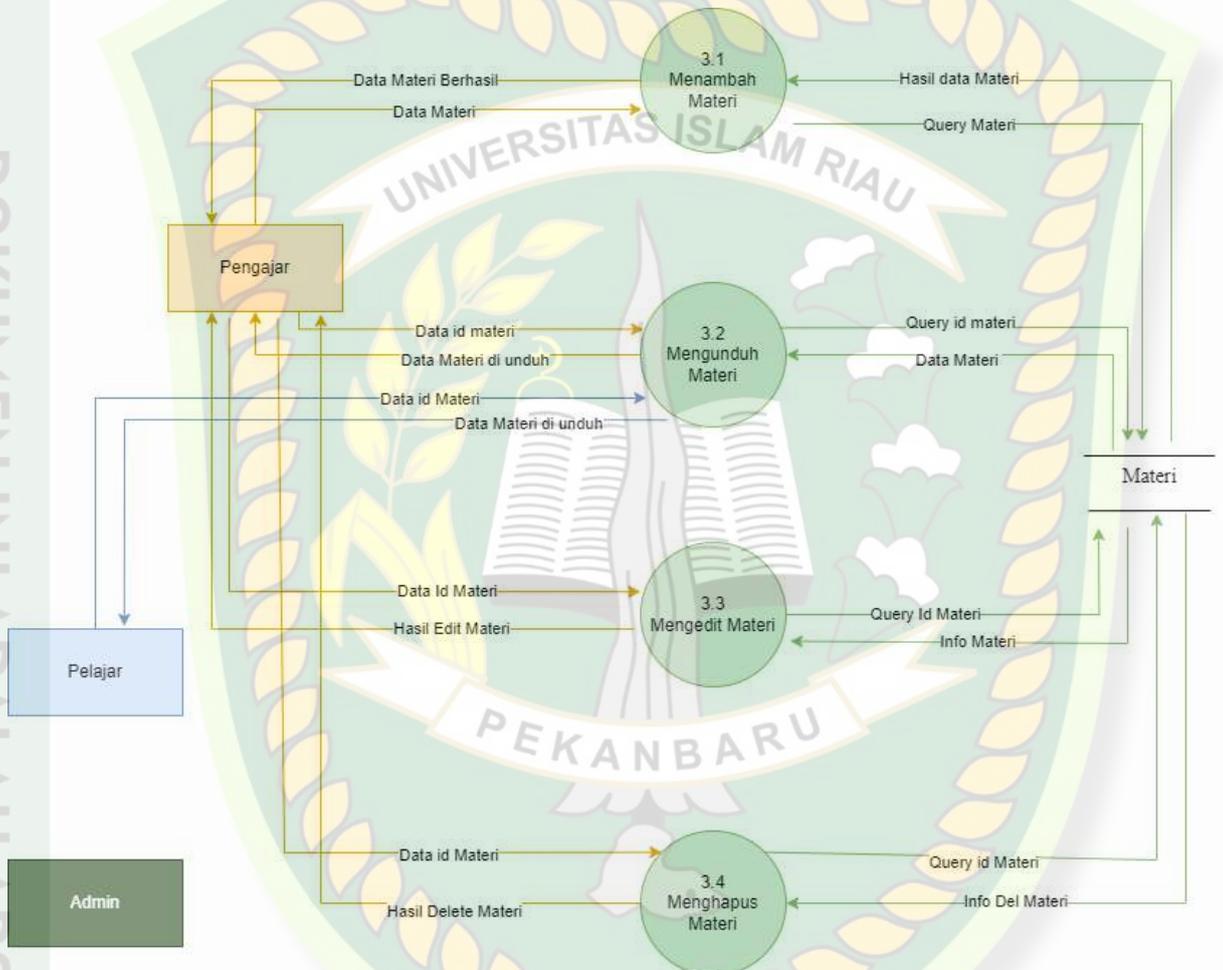


Gambar 3.10 DFD level 2 proses 1

Pada gambar 3.10 DFD level 2 pada proses 1, menjelaskan sebuah proses mengelola *user*. Sub proses tersebut dilakukan dan di proses oleh 3 entitas *eksternal* atau hak akses yaitu *admin* pengajar dan pelajar. Pengajar dan pelajar dapat melakukan 1 sub proses yakni *mengedit user*, sedangkan *admin* hanya mampu melakukan 2 sub proses yakni mendaftarkan *user* pengajar dan menghapus *user* pelajar dan pengajar. Nantinya hasil dari 2 sub proses tersebut akan disimpan pada *collection user* yang ada di *mongoddb*.

3.4.3.3 DFD Level 2 (Proses 2)

DFD *level 2* (proses 2) adalah sebuah sub proses yang beroperasi pada proses mengelola *profile* dan terdapat 2 sub proses yakni *mengedit* nama *profile* dan *mengedit password profile*. *Data flow* diagramnya dapat dilihat pada gambar 3.10.



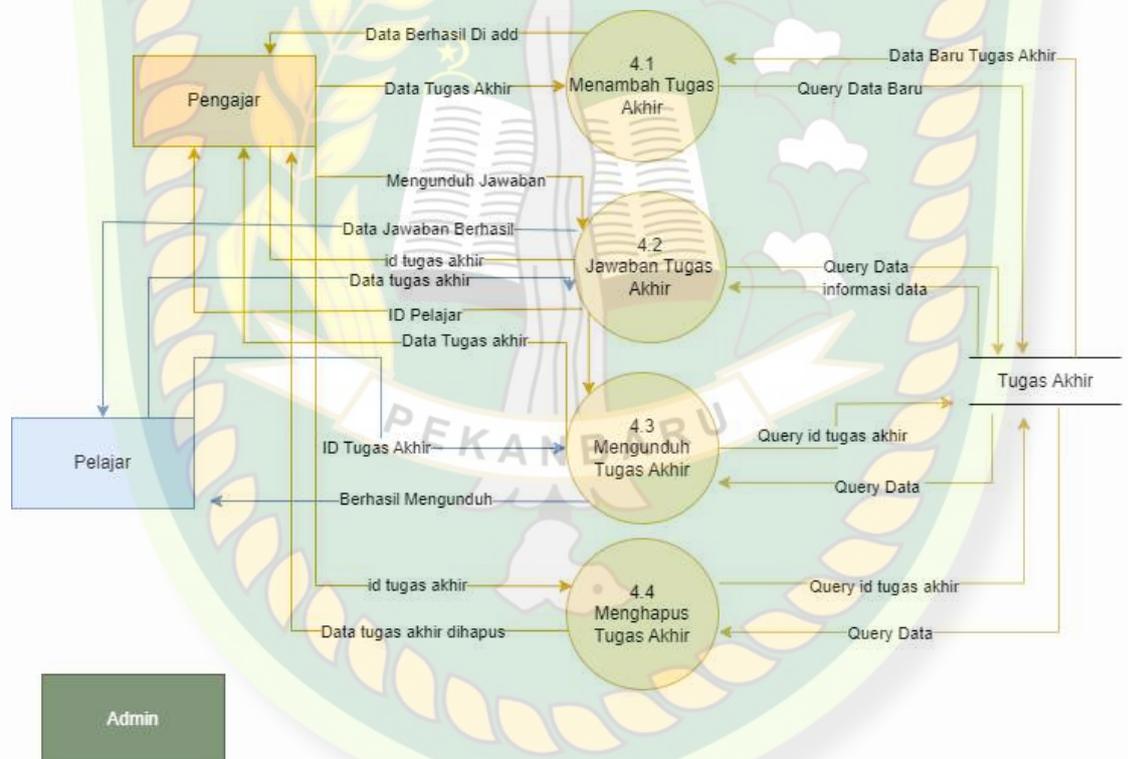
Gambar 3.11 DFD level 2 proses 2

Pada Gambar 3.11 DFD level 2 proses 2, menjelaskan sebuah proses materi yang ada pada sistem *E-Learning*. Sub proses ini dapat digunakan oleh 2 entitas *eksternal* atau hak akses, dimana hak akses pengajar dapat melakukan 4 sub proses yakni dapat melakukan menambah materi, mengunduh materi, mengedit materi dan menghapus materi. Sedangkan untuk hak akses pelajar hanya mampu

melakukan 1 sub proses yakni hanya dapat mengunduh materi yang tersedia.

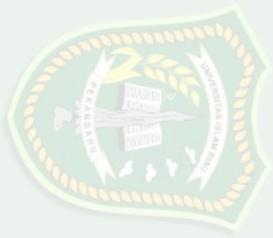
3.4.3.4 DFD Level 2 (Proses 3)

DFD *level 2* proses 3 merupakan sebuah sub proses yang berjalan di proses mengelola tugas akhir dan memiliki 3 sub proses yakni mengupload tugas akhir, mengunduh tugas akhir dan menghapus tugas akhir, untuk penjelasan lebih lanjut ada pada gambar 3.10 dibawah ini:



Gambar 3.12 DFD level 2 proses 3

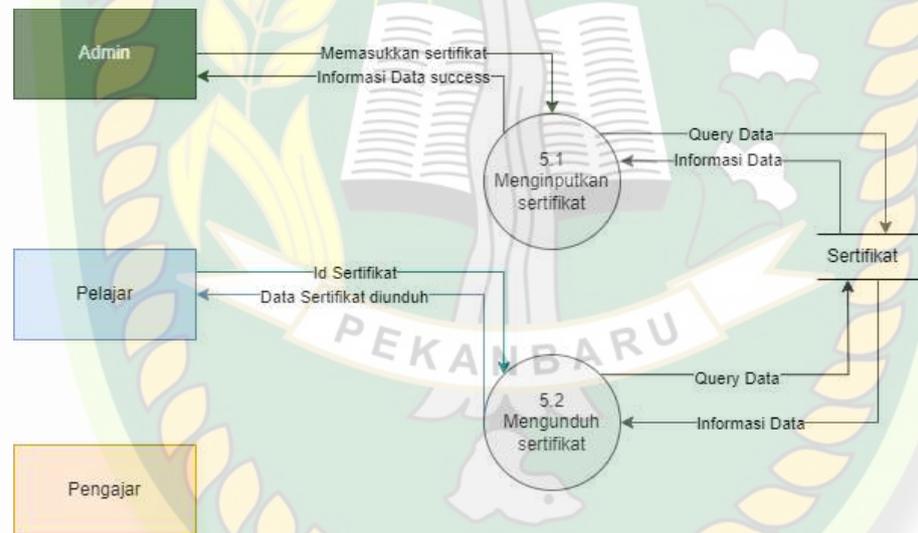
Dari gambar 3.12 DFD *level 2* proses 3 terdapat 2 entitas *eksternal* yang mana pengajar bisa memberikan sebuah *file* soal tugas akhir di dalam sistem dan bisa melihat jawaban yang sudah diinputkan oleh pelajar tidak hanya itu pengajar juga dapat mengunduh dan menghapus *file* tugas akhir yang ada di sistem. Entitas eksternal



pelajar hanya mampu melakukan 2 sub proses saja yakni mengunduh *file* soal tugas akhir dan menginputkan *file* jawaban tugas akhir Hasil dari kedua proses tersebut nantinya akan disimpan ke *document* tugas akhir.

3.4.3.5 DFD Level 2 (Proses 4)

DFD level 2 proses 4 adalah sebuah sub proses yang berjalan di sub proses mengelola sertifikat dan memiliki 2 sub proses yakni menginputkan sertifikat dan mengunduh sertifikat untuk penjelasan lebih lanjut dapat dipahami pada gambar 3.13.



Gambar 3.13 DFD Level 2 Proses 4

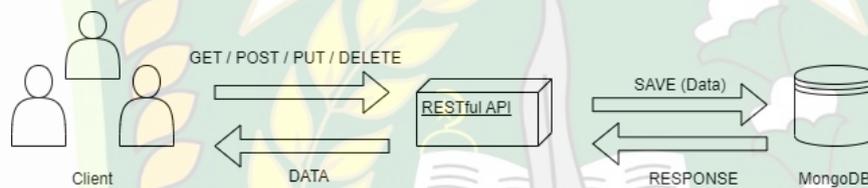
Pada gambar 3.13 terlihat bahwa DFD level 2 proses 4 memiliki 2 entitas yang bekerja yakni *admin* dan pelajar, untuk entitas eksternal *admin* hanya dapat melakukan 1 sub proses saja dimana ia akan menginputkan sertifikat kepada para pelajar, sedangkan pelajar juga hanya bisa melakukan 1 sub proses saja yakni mengunduh sertifikat apabila sudah diinputkan oleh si *admin*.

ISLAM RIAU



3.4.4 Perancangan *RESTful* API

Setiap *data* yang dikirim melalui sisi *client* akan dikirimkan dan diolah dalam sebuah penyimpanan mongoDB. Agar *data* dapat terkirim dan diambil dari mongodb maka diharuskan untuk membangun sebuah *RESTful API*. *RESTful API* sendiri ialah sebuah antarmuka yang menghubungkan sistem *client* dan *server*, ini bertujuan agar dapat saling bertukar informasi secara aman melalui internet. Berikut rancangan sebuah *RESTful API* pada gambar 3.14.



Gambar 3.14 Cara Kerja *RESTful API*

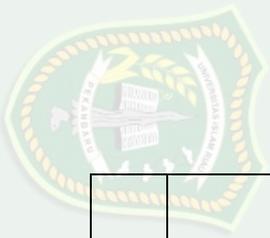
3.4.5 Fungsi tiap *Endpoint* API pada Sistem *E-Learning*

Sistem *E-Learning* yang dibangun menggunakan *RESTful API* untuk mendapatkan dan bertukar *data* dari setiap *user* terhadap sistem *E-Learning*, untuk itu maka dibawah ini adalah penjelasan dari fungsi *endpoint* api yang sudah di desain dan dipergunakan oleh sistem *E-Learning* yang dirancang.

Tabel 3.3 Fungsi *Endpoint* API pada Sistem *E-Learning*

No	Endpoint	Body Request	Response 200 – 201 (Success)	Response 400 – 404 (Error)	Method	Fungsi
1	/api/login	{ "Email", "Password" }	{ "nama", "Email", "password" }(200)	{ "Error": email anda tidak terdaftar }(404)	POST	Fungsi dari <i>endpoint</i> ini digunakan untuk mengirimkan <i>data user</i> yang ingin mengakses

				<pre> “Error”: Email atau password anda salah }(400) </pre>		sistem <i>E-Learning</i> .
2	/api/register	<pre> { “Nama”, “EmailStudent”, “password” } </pre>	<pre> { “nama”, “Email”, “password” }(201) </pre>	<pre> { “error”: error Message }(400) </pre>	POST	<p>Fungsi dari <i>endpoint</i> ini digunakan untuk melakukan pendaftaran akun <i>user student</i> agar dapat mengakses sistem <i>E-Learning</i>, namun saat melakukan pendaftaran harus menggunakan akun <i>email student</i> UIR.</p>
3	/api/sendMail	<pre> { “Email” } </pre>	<pre> { “msg”: “success to send mail” }(201) </pre>		POST	<p>Fungsi dari <i>endpoint</i> ini digunakan untuk mengirimkan <i>password</i> melalui <i>email student</i> yang sudah terdaftar</p>



						di <i>database</i> sistem <i>E-Learning</i>
4	/api/user/student		{ <i>dataStudent</i> }(200)	{ "err": <i>there is no data student</i> }(404)	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mendapatkan keseluruhan <i>data user student</i> yang terdaftar di <i>database E-Learning</i>
5	/api/user/student/{_idUser}		{ <i>dataSpesificStudent</i> }(200)	{ "err": <i>data murid tidak ada</i> }(404)	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mendapatkan <i>data student</i> secara spesifik.
6	/api/user/student/{_idUser}	{ "email", "nama", "Password" }	{ <i>dataStudent</i> }(201)	{ "err": <i>data tidak berhasil diubah</i> }(400)	PATCH	Fungsi dari <i>endpoint</i> ini digunakan untuk merubah atau memperbarui <i>data student</i> .
7	/api/user/admin/{_idUser}		{ <i>dataPengajar</i> }	{ "err": <i>data tidak dapat ditemukan</i> }(400)	GET	Fungsi dari <i>endpoint</i> ini adalah untuk mendapatkan <i>data pengajar</i> secara spesifik

8	/api/user/admin/{_idUser}	{ "email", "nama", "Password" }	{ dataPengajarU pdate (201) }	{ "err": data pengajar tidak dapat diubah (400) }	PATCH	Fungsi dari <i>endpoint</i> ini digunakan untuk merubah atau memperbarui <i>data</i> pengajar secara spesifik.
9	/api/file/material		{ allFile (200) }	{ "err": there is no materi (400) }	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mengambil <i>data</i> materi pelajaran secara keseluruhan yang ada di <i>database</i> .
10	/api/file/upload	{ "titleFile", "descFile", "catergyFile", "file" }	{ newData (201) }	{ "err": tidak dapat menambahm ateri (400) }	POST	Fungsi dari <i>endpoint</i> ini digunakan untuk memasukkan <i>data</i> materi ke dalam <i>database</i> .
11	/api/file/download/{_idMateri}		Download materi(200)	{ "err": does not exist the file (400) }	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mendownload materi pelajaran

12	/api/file/download/{_idMateri}	{ "ChangeTitle", "ChangeDesc", }	{ dataMateriUpdate }(201)	{ "err": data materi tidak dapat diubah }(400)	PATCH	Fungsi dari <i>endpoint</i> ini digunakan untuk merubah <i>data</i> materi.
13	/api/file/material/{_idMateri}		{ Data materi }(200)	{ "err": data does't exist }(400)	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mendapatkan <i>data</i> materi secara spesifik
14	/api/file/material/{_idMateri}	{ "_id" }	{ Msg: "success remove data" }(200)	{ Err: "Materi tidak dapat dihapus" }(400)	DELETE	Fungsi dari <i>endpoint</i> ini digunakan untuk menghapus <i>data</i> materi yang ada di <i>database</i> .
15	/api/file/taskStudent	{ "linkProject", "fileTasl" }	{ Msg: Task Berhasil terkirim }(201)	{ Err: task tidak dapat dikirim, pastikan internet stabil }(400)	POST	Fungsi dari <i>endpoint</i> ini digunakan untuk menginputkan <i>data</i> tugas akhir yang bakal masuk ke <i>database</i> .
16	/api/file/taskStudent		{ dataTaskStudent }	{ Err: "file tidak ada"	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk

			}(200)	}(400)		menampilkan tugas akhir yang berada di <i>database</i> .
17	/api/file/taskStudent/{_idTask}	{ “_id”, }	{ Msg: “data materi terhapus” }(200)	{ Err: “data materi tidak dapat dihapus” }(400)	DELETE	Fungsi dari <i>endpoint</i> ini digunakan untuk menghapus <i>data</i> tugas akhir yang ada di <i>database</i>
18	/api/file/taskStudent/{_idTask}		Download data task(200)	{ Err: “data tidak tersedia” }(400)	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mendownload <i>file</i> tugas akhir yang ada di <i>database</i> .
19	/api/file/fileStudent/{_idUser}	{ “linkGithub”, “fileStudent” }	{ Msg: jawaban TA berhasil dikirim }(200)	{ Err: Tugas akhir tidak dapat terkirim }(400)	POST	Fungsi dari <i>endpoint</i> ini digunakan untuk mengirimkan jawaban tugas akhir dari hasil pengerjaan soal tugas akhir yang diberikan.
20	/api/file/fileStudent/{_idUser}		{ dataFileAnswerStudent }(200)	{ Err: “data jawaban” }	GET	Fungsi dari <i>endpoint</i> ini digunakan untuk mengambil <i>data</i>

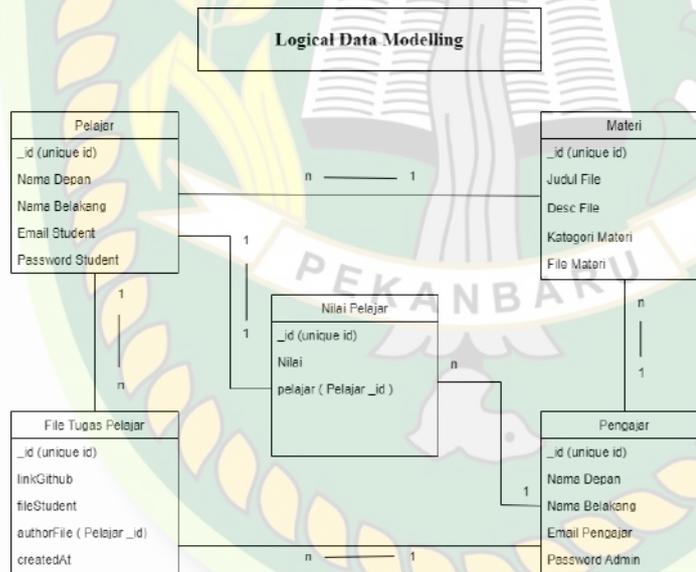
				siswa tidak ada” {(400)}		jawaban dari hasil pengerjaan tugas akhir yang dikerjakan oleh student.
21	Api/file/taskDownload/{_idUser}		Download dataAnswerStudent(200)	{ Err: “data siswa tidak dapat di download” {(400)}	GET	Fungsi dari endpoint ini digunakan untuk mendownload file jawaban dari hasil pengerjaan yang dilakukan oleh student.
22	/api/nilaiStudent/{_idUser}	{ “nilaiStudent”, “alasanNilai” }	{ Msg: “Nilai berhasil diinputkan” {(201)}	{ Err: “form tidak boleh ada yang kosong” {(400)}	POST	Fungsi dari endpoint ini digunakan untuk menginputkan nilai dan memberikan sertifikat kepada student yang lolos dalam menyelesaikan tugas akhir yang diberikan.
23	/api/nilaiStudent/downloadCertificate/{_idUser}		dataCertificate Student(200)	{ Err: “anda belum mengerjakan task akhir”	GET	Fungsi dari endpoint ini digunakan untuk mendownload

				}(400)		sertifikat yang ada di <i>database</i> .
--	--	--	--	--------	--	------------------------------------------

3.4.6 Rancangan Database

3.4.6.1 Logical Data Model

Penerapan *logical data model* adalah suatu cara untuk dapat memvisualisasikan kebutuhan *data* pada *storage system* yang dirancang, tujuannya adalah untuk membuat suatu fondasi pada penerapan *physical data modelling* supaya dapat membedakan setiap atribut dan hubungan dari setiap entitas. Berikut penerapan *Logical Data Modelling* pada *system E-Learning* pada gambar 3.15:



Gambar 3.15 Konsep Penerapan LDM

3.4.6.2 JSON Format Representation

Dalam penggunaan *mongoDB data* disimpan dan berbentuk dokumen-dokumen BSON. BSON adalah representasi dari *format JSON* dan didalam *format JSON* terdiri dari representasi *form* yang

memiliki *field* dan *value* yang saling berpasangan (Rupali Arora & Rinkle Rani Aggarwal, 2013).

Penulisan JSON sendiri sama halnya seperti membuat suatu *object* di bahasa pemrograman javascript yang memerlukan kurung siku atau *square brackets* (“[]”) dan kurung kurawal atau *curly brackets* (“{ }”), dalam mongoDB *data* yang disimpan dalam *format* JSON biasa disebut dengan dokumen. Penerapan *database* yang akan dibuat memiliki 3 dokumen yang berasal dari *collection* yang berbeda, berikut dokumen-dokumen tersebut ialah pelajar, materi, dan pengajar.

a. Dokumen *Admin*

Dokumen *admin* adalah dokumen yang berfungsi untuk menyimpan *data user admin* dan pada dokumen ini memiliki 5 *field* dan *_id* sebagai *field* yang secara otomatis dibuat oleh sistem mongodb selain *field* tersebut ada *field* *nama_depan*, *nama_belakang*, *email_admin* dan *password_admin*, untuk lebih jelasnya dapat melihat pada gambar 3.16 dibawah ini.

```

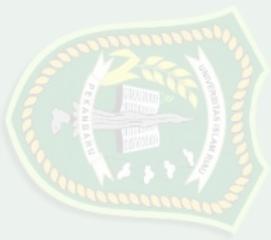
1  "admin": {
2    "_id": "<String Val>",
3    "nama_depan": "<String Val>",
4    "nama_belakang": "<String Val>",
5    "email_admin": "<String Val>",
6    "password_admin": "<String Val>"
7  }

```

Gambar 3.16 Document Admin

b. Dokumen Pelajar

Dokumen pelajar adalah dokumen yang berfungsi untuk menyimpan *data para user student* dan pada dokumen pelajar memiliki 5 *field* dan *_id* sebagai *field unique* yang



dibuat secara otomatis oleh mongoDB selain itu isi dari *field* pelajar ada nama_depan, nama_belakang, *email_student* dan password_student untuk lebih jelasnya dapat dilihat pada gambar 3.17.

```

1  ▾ "Pelajar": {
2    "_id": "<string val>",
3    "nama_depan": "<string val>",
4    "nama_belakang": "<string val>",
5    "email_student": "<string val>",
6    "password_student": "<string val>"
7  }

```

Gambar 3.17 Document Pelajar

c. Dokumen Pengajar

Dokumen Pengajar adalah dokumen yang digunakan untuk menyimpan dan memproses *data* para pengajar. Pada dokumen ini memiliki 5 *field* dan *_id* sebagai *field unique* yang dibuat secara otomatis oleh mongoDB selain *field _id* ada *field* nama_depan, nama_belakang, *email_pengajar*, dan password_pengajar, untuk lebih jelasnya dapat dilihat pada gambar 3.18.

```

1  ▾ "Pengajar": {
2    "_id": "<string val>",
3    "nama_depan": "<string val>",
4    "nama_belakang": "<string val>",
5    "email_pengajar": "<string val>",
6    "password_pengajar": "<string val>"
7  }

```

Gambar 3.18 Document Pengajar

d. Dokumen Materi



Dokumen materi adalah dokumen yang digunakan untuk menampung atau menyimpan *data* materi pelajaran terkait *frontend enginner*, pada dokumen materi memiliki beberapa *field* didalam suatu dokumen yang disimpan di *database* mongodb dan *field* *_id* sebagai *field unique* untuk setiap dokumennya yang dibuat secara otomatis oleh mongoDB, untuk lebih jelasnya dapat dilihat pada gambar 3.19.

```

1  {
2    "_id": "<String Val>",
3    "titleFile": "<String Val>",
4    "descFile": "<String Val>",
5    "categoryFile": "<String Val>",
6    "file": "String Val"
7  }

```

Gambar 3.19 Document Materi

e. Dokumen *File* Tugas Akhir Pelajar

Dokumen *file* tugas akhir pelajar adalah dokumen yang digunakan untuk menyimpan serta mengolah *data file* tugas akhir pelajar pada dokumen ini berisikan *field* terkait *data* yang di-inputkan oleh pelajar sesuai inputan yang ditetapkan dalam menyelesaikan soal tugas akhir materi. *Field-field* tersebut berisikan *linkGithub*, *fileStudent*, dan *authorFile*.

Berikut lebih jelasnya dapat dilihat pada gambar 3.20.



```

1 {
2   "_id": "<String Val>",
3   "linkGithub": "<String Val>",
4   "fileStudent": "<String Val>",
5   "authorFile": "<String Val>"
6 }

```

Gambar 3.20 Document File Tugas Akhir Pelajar

f. Dokumen Nilai Siswa

Dokumen nilai siswa adalah dokumen yang bertujuan untuk menyimpan *data-data* nilai siswa sehabis mengerjakan tugas akhir, pada dokumen ini nantinya akan diinputkan oleh *admin*. Dokumen ini sama seperti dokumen yang lainnya yang mana *field* *_id* akan dibuat otomatis oleh *mongoDB*, berikut untuk lebih jelasnya dapat dilihat pada gambar 3.21.

```

1 {
2   "_id": "<String Val>",
3   "idSiswa": "<String Val>",
4   "nilaiSiswa": "<String Val>"
5 }

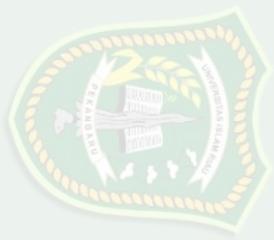
```

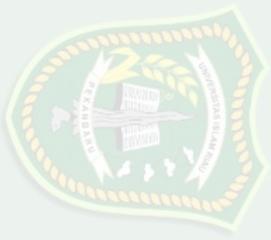
Gambar 3.21 Document Nilai Siswa

3.4.7 Desain antar muka

3.4.7.1 Desain Antar Muka Login

Antarmuka *login* adalah halaman untuk masuk kedalam *system E-Learning* dengan memasukkan *data admin*, pengajar atau pelajar yang sudah terdaftar di *database*, dapat dilihat pada gambar 3.22.





Welcome a new generation of
Frontend Engineer
Make it Your Future

X [Logo]

Email :
x

Password :
x

Login

Belum memiliki akun? daftar.

Lupa Password?

Gambar 3.22 Login Page

3.4.7.2 Desain Antar Muka Register

Antar muka *register* merupakan halaman yang disediakan untuk melakukan pendaftaran akun bagi setiap pelajar supaya bisa mengakses sistem *E-Learning* ini, design antarmuka dapat dilihat pada gambar 3.23.

Register New Account X

Nama Depan Nama Belakang

Email Address Student

Password

Sign Up

Gambar 3. 23 Register Page

Dapat dilihat pada gambar 3.21 merupakan halaman bagi *user* pelajar yang memilih *menu* daftar untuk melakukan *register* akun.

3.4.7.3 Desain antarmuka lupa password

Desain antarmuka lupa *password* merupakan halaman bagi *user* pengajar maupun pelajar apabila lupa terhadap *password* merekatampilan antarmuka lupa *password* dapat dilihat pada gambar 3.24.



Gambar 3.24 *forget password page*

3.4.7.4 Desain Antarmuka *Dashboard Admin*

Desain antarmuka *dashboard* admin adalah tampilan yang akan dijalankan apabila login di sistem *E-Learning* sebagai *user* admin. Halaman website sebagai admin dapat melakukan beberapa aksi seperti melihat pengajar, melihat pelajar dan memberikan sertifikat kepada pelajar serta dapat menghapus pengajar dan pelajar, untuk lebih jelasnya dapat dilihat pada gambar 3.25 dibawah ini.

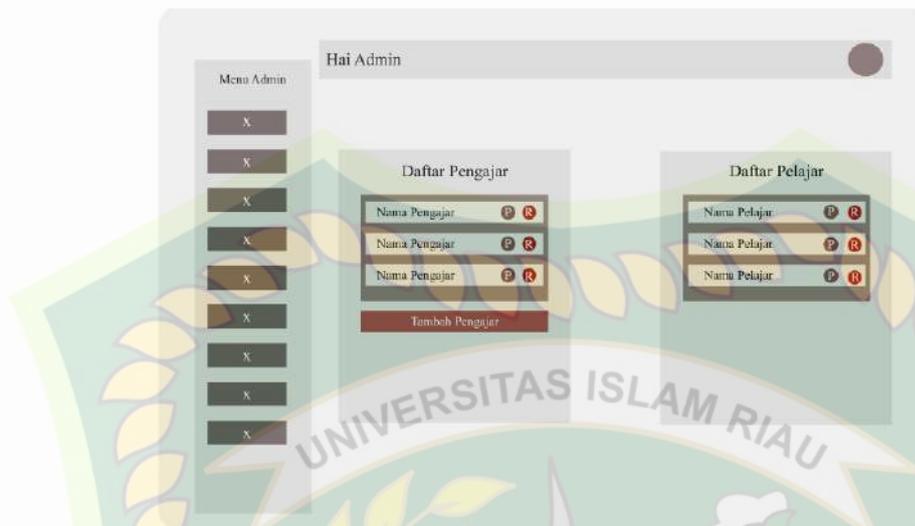
UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



Gambar 3.25 Dashboard Admin

dapat dilihat bahwa pada gambar diatas terdapat huruf P dan R, jika huruf P adalah Profile yang merujuk ke profile *user* tersebut apabila button-nya di klik, sedangkan R adalah Remove yang merujuk untuk menghapus akun *user* tersebut.

3.4.7.5 Desain Antarmuka Dashboard Pengajar

Desain antarmuka *dashboard* pengajar adalah *design* yang menampilkan halaman *website* bagi *user* pengajar yang dapat mengisi materi baru ke sistem *E-Learning* terkait pembelajaran *frontend development*. Selain itu pada *dashboard* pengajar, pengajar juga dapat melihat *data* siswa, menerima dan mengirim pesan. Untuk melihat desain pada *dashboard* pengajar dapat dilihat pada gambar

3.26.

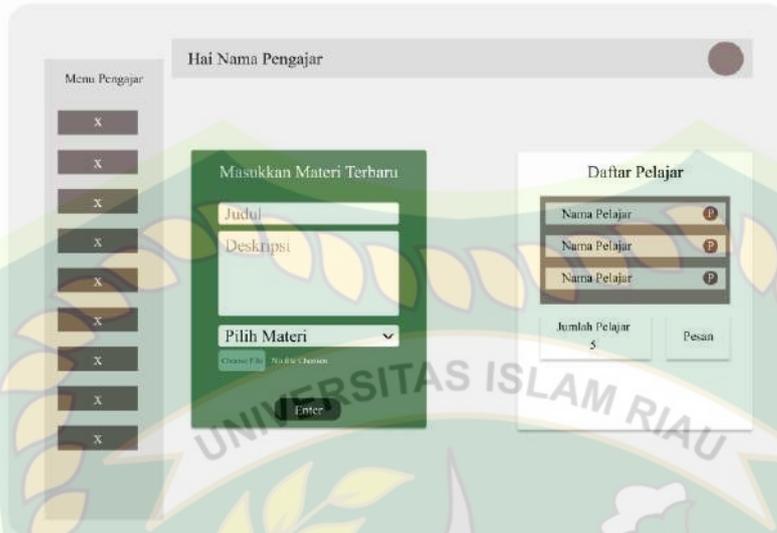
UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



Gambar 3.26 Desain *Dashboard Pengajar*

3.4.7.6 Desain antarmuka *Dashboard Pelajar*

Desain antarmuka pelajar adalah antarmuka yang menampilkan inputan tugas akhir dan melihat *score* yang diraih setelah menyelesaikan *course* dan tugas akhirnya, selain itu pada *dashboard* pelajar juga dapat melihat pesan masuk dan mengirim pesan, untuk lebih tepatnya dapat dilihat pada gambar 3.27.



Gambar 3.27 *Dashboard Pelajar*

3.4.7.7 Desain Antarmuka Ubah *Data User*

Desain antarmuka ubah *user* digunakan untuk merubah *data user* yang telah ada di *database*, setiap inputan yang ada di *form* antarmuka sudah disesuaikan dengan kebutuhan *data* yang telah ditetapkan pada *database* sistem *E-Learning*, untuk lebih jelasnya dapat dilihat pada gambar 3.28.

Gambar 3.28 Antarmuka Ubah Data

3.4.7.8 Desain Antarmuka Ubah Materi

Desain antarmuka ubah materi adalah antarmuka yang dapat membantu *user* pengajar untuk merubah *data* apabila terjadi kesalahan inputan, misalnya kesalahan menginput judul, deskripsi, serta kategori *file* materi yang sudah diinputkan sebelumnya, untuk lebih jelasnya terkait desain antarmuka ubah materi dapat dilihat pada gambar 3.29.

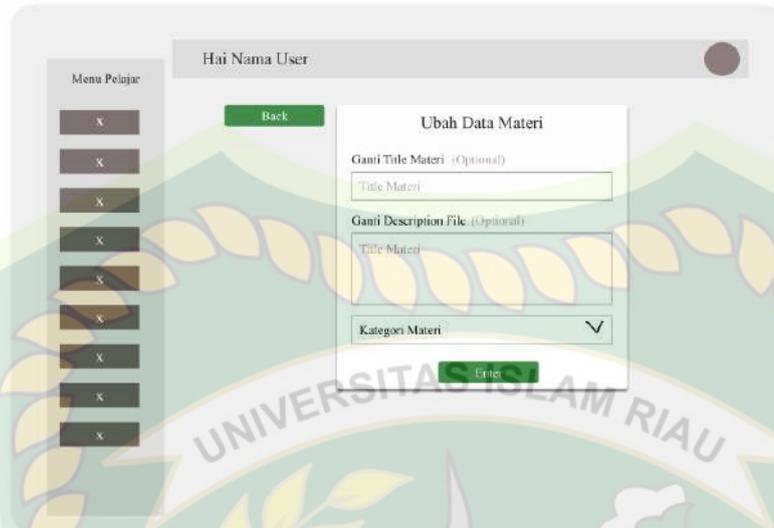
UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



Gambar 3.29 Antarmuka Ubah Materi

3.4.7.9 Desain Antarmuka Materi

Desain antarmuka materi adalah tampilan yang berisikan materi-materi sesuai dengan *menu* yang ada, pada penulisan penelitian ini, penulis hanya mencontohkan tampilan pada menu HTML dasar saja sebagai gambarannya, untuk lebih jelasnya dapat dilihat pada gambar 3.30.



Gambar 3.30 Antarmuka Materi Pembelajaran

3.4.7.10 Desain Antarmuka *Detail Student*

Desain antarmuka *detail student* adalah desain yang mendeskripsikan setiap pelajar baik itu dari nama, *email* serta tugas apa saja yang sudah dikumpulkan. Untuk lebih jelasnya dapat dilihat pada gambar 3.31.

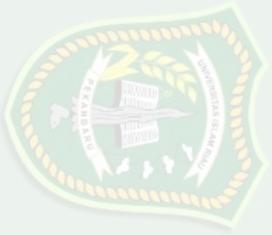


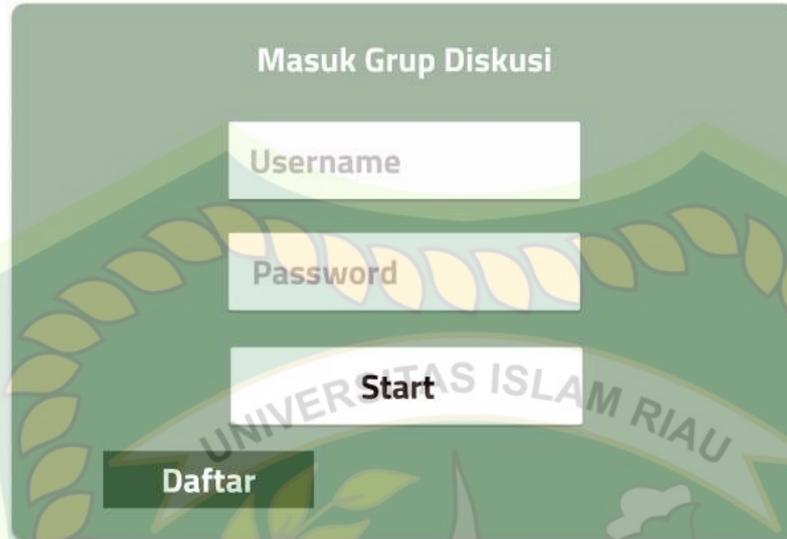
Gambar 3.31 Antarmuka *Detail Student*

3.4.7.11 Desain Tampilan *Login Group Chat*

Desain antarmuka login group *chat* adalah desain antarmuka untuk dapat masuk ke halaman grup *chat* dengan memasukkan *username* dan *password* yang sudah didaftarkan oleh pengajar, berikut lebih jelasnya dapat dilihat pada gambar 3.32.

UNIVERSITAS
ISLAM RIAU

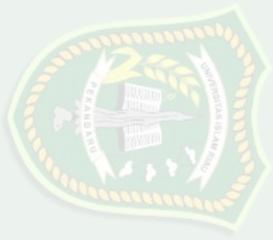




Gambar 3.32 Login Group Chat

3.4.7.12 Desain Halaman *Group Chat*

Pada desain halaman *group chat* terdapat tampilan yang berisi sekumpulan para pengajar dan pelajar. Halaman ini pengajar dan pelajar dapat membuat grup diskusi, mengirimkan gambar, mengirimkan file, untuk lebih jelasnya dapat dilihat pada gambar 3.33.

Gambar 3.33 Tampilan *group chat*

3.4.8 Flowchart

3.4.8.1 Flowchart Admin



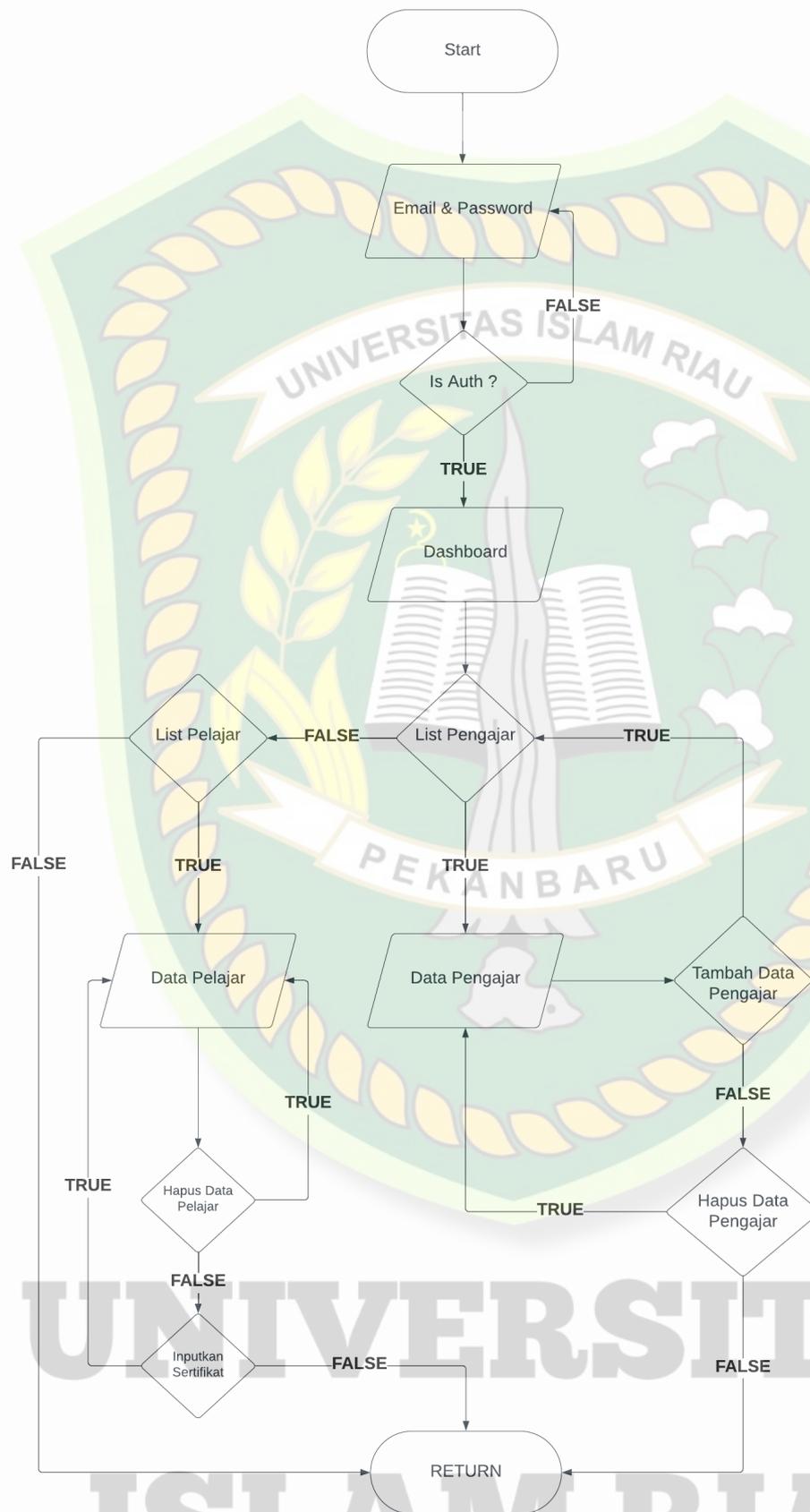
**UNIVERSITAS
ISLAM RIAU**



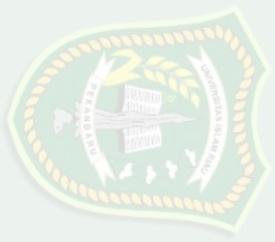
DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



Gambar 3.34 Flowchart Admin

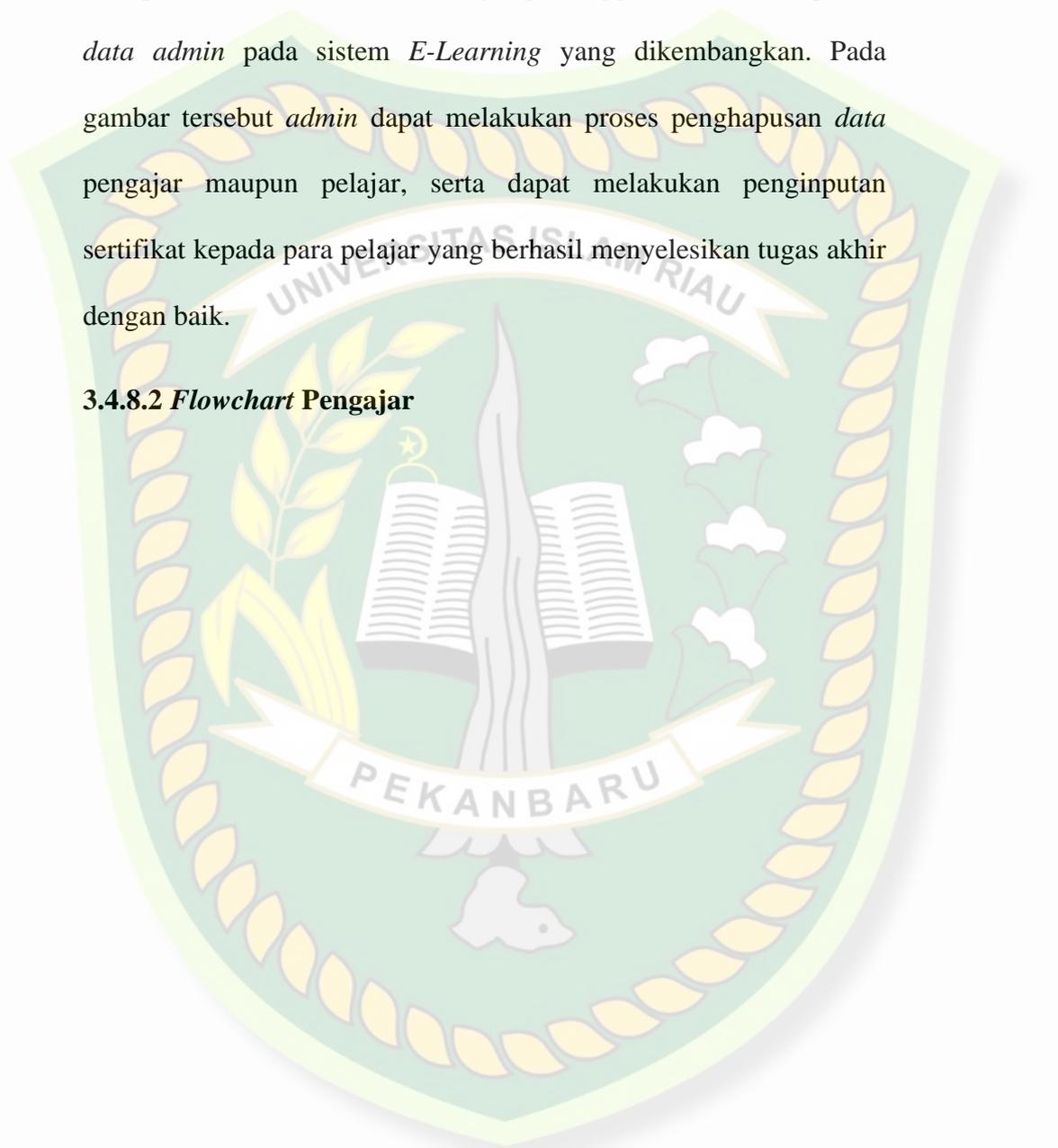


DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU

UNIVERSITAS
ISLAM RIAU

Flowchar Admin yang terdapat pada gambar 3.34 diatas merupakan sebuah aliran sistem yang menggambarkan alur proses *data admin* pada sistem *E-Learning* yang dikembangkan. Pada gambar tersebut *admin* dapat melakukan proses penghapusan *data* pengajar maupun pelajar, serta dapat melakukan penginputan sertifikat kepada para pelajar yang berhasil menyelesaikan tugas akhir dengan baik.

3.4.8.2 *Flowchart* Pengajar

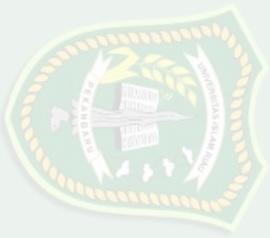


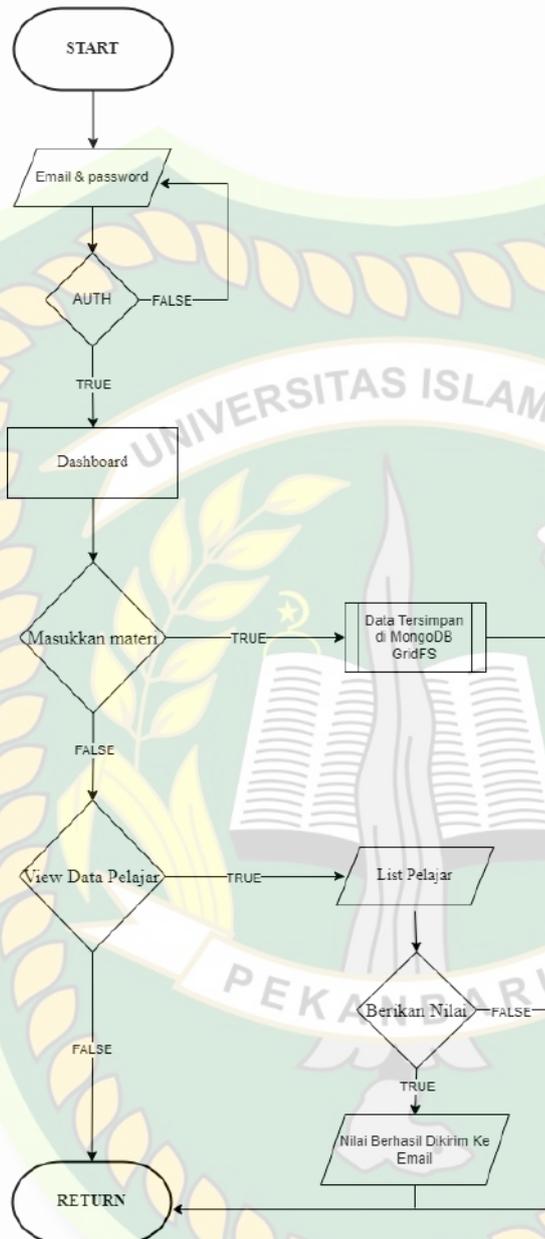
**UNIVERSITAS
ISLAM RIAU**

DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



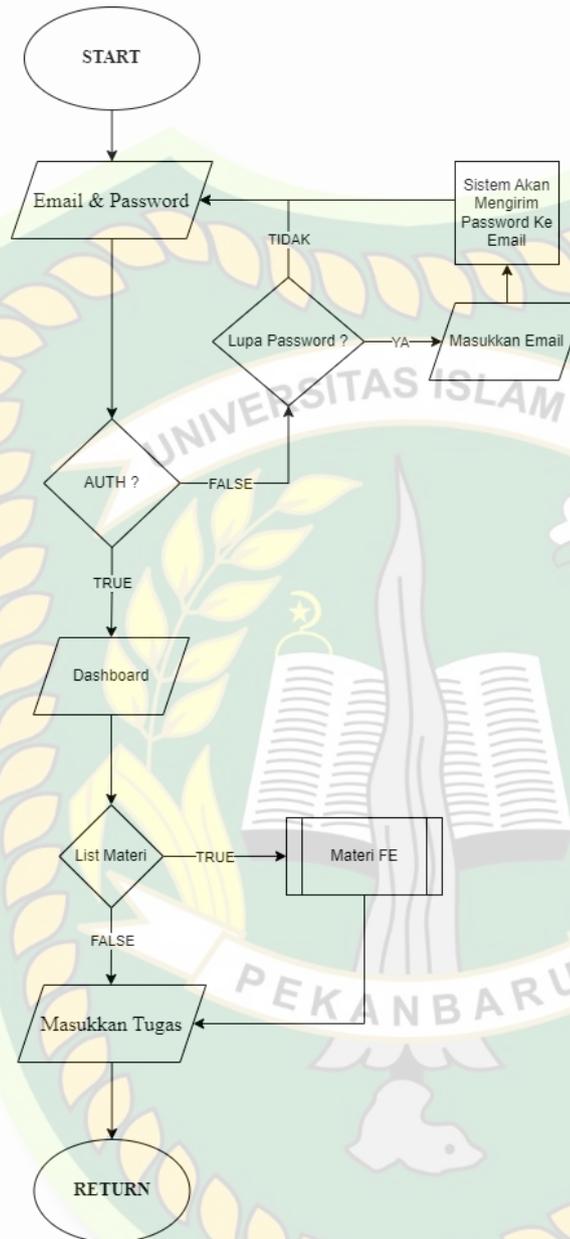


Gambar 3.35 Flowchart Pengajar

Flowchart pengajar yang ada pada gambar 3.35 adalah suatu aliran sistem yang mendeskripsikan alur proses *data* pengajar pada sistem *E-Learning* yang dikembangkan, pengajar selain aktivitas login yang dilakukan namun juga bisa melakukan penginputan materi dan memberikan nilai terhadap pelajar.

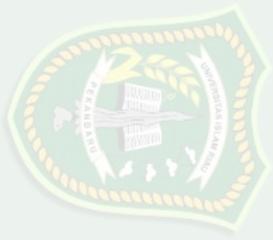
3.4.8.3 Flowchart Pelajar





Gambar 3.36 Flowchart Student

Flowchart Pelajar pada gambar 3.36 diatas adalah alur proses yang menggambarkan diagram alir pelajar mulai dari login kemudian dapat melihat *list* materi yang tersedia, pelajar juga dapat melakukan penginputan jawaban tugas akhir berupa *file* pdf dan link proyek yang dikerjakan.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Penelitian

Sebelum sistem yang sudah dikembangkan benar-benar digunakan oleh para pengajar dan pelajar ada beberapa tahap lagi yang harus dilakukan, hal ini supaya bertujuan untuk meminimalisir ditemukannya kesalahan atau *bug* pada sistem. Dalam pengujian pada sistem nantinya akan dilakukan yang namanya *testing* dengan menggunakan metode *Blacbox*.

4.2 Pengujian *Blacbox*

Pengujian *Blacbox* atau *Blacbox testing* adalah sebuah teknik pengujian pada suatu sistem yang bertujuan untuk menemukan *bug* pada sistem dan menemukan kesalahan yang terjadi di sistem. *Blacbox* sangat penting pada pengujian khususnya saat proses terjadinya inputan yang masuk dan apakah output yang keluar sesuai dengan yang diharapkan atau tidak. Dalam pengujian *Blacbox* di sistem *E-Learning* pembelajaran *frontend development* ini, berikut hal-hal yang sudah dilakukan:

4.2.1 Pengujian Halaman *Register*

Tampilan pada halaman *register* merupakan tempat dimana siswa akan melakukan pendaftaran sebagai pelajar yang ingin mempelajari ilmu *frontend development*. Dengan mengisi beberapa *form* yang telah disediakan seperti: nama depan, nama belakang, *email student* dan *password* maka secara otomatis *data* siswa akan tersimpan didalam suatu

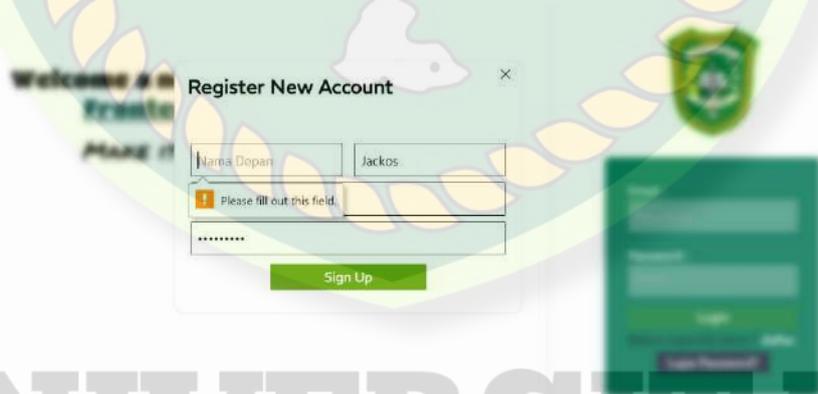


database yang telah disediakan *developer*. Halaman *register* dapat dilihat pada gambar 4.1.



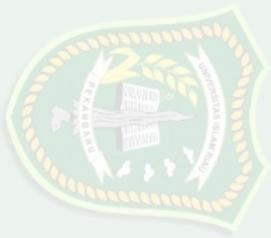
Gambar 4.1 Register Page

Setelah pelajar berhasil melakukan proses *register* maka secara otomatis pelajar akan langsung masuk kedalam sistem *E-Learning frontend development*, namun apabila pelajar tidak mengisi salah satu *form* maka akan terdapat pesan *error* seperti pada gambar 4.2.



Gambar 4.2 Peringatan Error Fill Out

UNIVERSITAS
ISLAM RIAU



Namun apabila pelajar mendaftar dengan tidak menggunakan *email student* uir maka akan terdapat pesan *error* dan tidak dapat melakukan proses registrasi akun, seperti pada gambar 4.3.



Gambar 4.3 Error Not Mail UIR

Pada gambar 4.3 merupakan notifikasi yang menandakan bahwa pelajar tidak menginputkan *email student* uir sebagai *email* registrasinya, jika pelajar tidak mendaftar dengan menggunakan *email student* maka sistem akan menolak registrasi akun sih pelajar. Namun jika *student* mendaftar sesuai arahan yang diberikan maka *student* akan langsung diarahkan ke halaman *dashboard*, seperti pada gambar 4.4

**UNIVERSITAS
ISLAM RIAU**



DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU



Gambar 4.4 Halaman *Home* Pelajar

Kemudian dari hasil pengujian halaman register yang telah dilakukan, rinciannya bisa dilihat pada tabel 4.1.

Tabel 4.1 Pengujian Halaman *Register* Pelajar

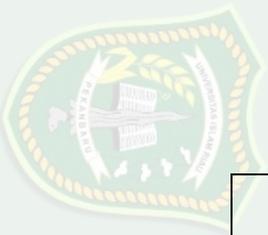
NO	Komponen Yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Halaman Register	Mengosongkan semua <i>field</i> yang ada pada <i>form register</i> seperti nama depan, nama belakang, <i>email student</i> , dan <i>password</i> , lalu mengklik <i>login</i> .	Sistem akan menolak dan memberikan pesan <i>error</i> yang berisi " <i>please field out this field</i> ".	Berhasil
		Mengosongkan salah satu <i>field</i> yang ada pada <i>form register</i> misal: nama depan: zul nama belakang: hendri <i>email</i> : - <i>password</i> : *****	Sistem akan menolak melakukan register dan akan memberikan <i>error message</i> yang berisi " <i>please field out this field</i> ".	Berhasil

<p>Mengisi semua <i>field</i> tetapi pada pengisian <i>email student</i> seperti: Nama depan: zul Nama belakang: hendri Email: eri@gmail.com Password: *****</p>	<p>Sistem akan menolak melakukan register dan akan memberikan <i>error message</i> yang berisi “Harus Menggunakan <i>Email Student UIR</i>”</p>	<p>Berhasil</p>
<p>Mengisi semua <i>field</i> sesuai arahan yang diberikan dan memasukkan <i>email</i> dengan memberikan <i>email student uir</i></p>	<p>Sistem akan menerima <i>data</i> register pelajar dan akan langsung masuk ke dalam sistem web <i>E-Learning</i> pembelajaran <i>frontend development</i></p>	<p>Berhasil</p>

Dari hasil pengujian yang sudah dilakukan pada halaman register maka dapat disimpulkan setiap aspek dan fungsi yang diharapkan telah berhasil.

4.2.2 Pengujian Halaman *Login*

Halaman *login* adalah halaman untuk *user* yang ingin masuk kedalam sistem *E-Learning* pembelajaran *frontend development*, dengan memasukkan beberapa *field* yang disediakan seperti *email* dan *password* dan apabila *data user* sudah terdata di dalam *database* maka secara otomatis *user* akan masuk kedalam sistem *E-Learning*, namun apabila *data email* dan *password* yang dimasukkan *user* tidak terdaftar di *database* maka proses *login* akan gagal. Untuk dapat melihat bagaimana tampilan halaman *login* dapat dilihat pada gambar 4.5.





Welcome a new generation of
Frontend Engineer
MAKE IT YOUR FUTURE



Gambar 4.5 login page

Setelah *user* memasukkan *field email* dan *password* yang sudah terdaftar di dalam *database* maka secara otomatis *user* akan langsung diarahkan ke halaman *home* pada sistem *E-Learning* pembelajaran *frontend development*, namun apabila *user* gagal atau mendapati pesan *error* saat login artinya *user* memasukkan *data* yang salah atau tidak mengisi salah satu *field* maka akan terdapat *error message* seperti pada gambar 4.6.

Welcome a new generation of
Frontend Engineer
MAKE IT YOUR FUTURE



Gambar 4.6 error form

Dapat dilihat pada gambar diatas halaman *login* akan memberikan *error message* jika *user* tidak mengisi salah satu *field* yang disediakan pada

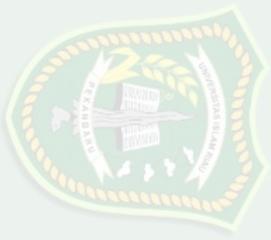
saat menekan tombol *login*. Kemudian apabila *user* mengisi *field* yang tidak terdaftar pada *database* maka akan memunculkan *error message* seperti pada gambar dibawah 4.7.



Gambar 4.7 error message data not found

Dapat dilihat pada gambar 4.7 menunjukkan apabila *data* yang dimasukkan belum terdaftar didalam *database* sistem maka akan memberikan *response error message* yang bertuliskan “Email Anda Belum Terdaftar”. Namun apabila *user* memasukkan *data* yang benar dan sudah terdaftar didalam *database* sistem maka *user* baik pelajar, pengajar, dan *admin* akan langsung diarahkan ke dalam sistem *E-Learning* pembelajaran *frontend development* seperti pada gambar 4.8 hingga gambar 4.10.

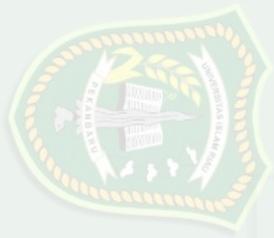
UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU

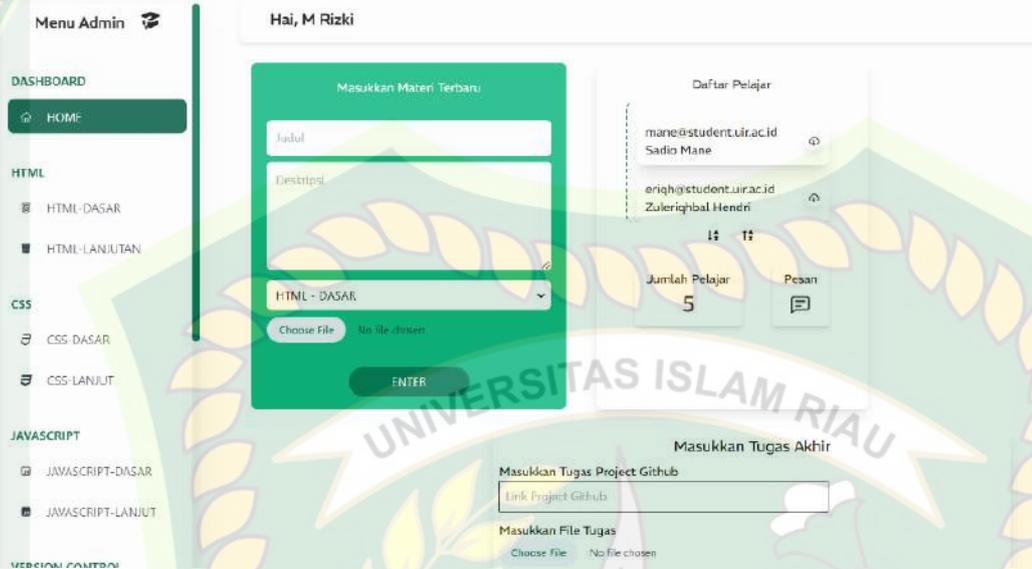
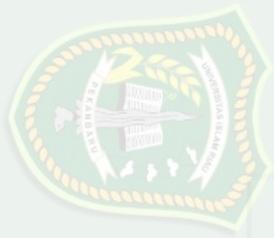


Gambar 4.8 Dashboard Pelajar

Pada gambar 4.8 adalah halaman *dashboard* dari pelajar apabila *user* memasukkan *data* pelajar yang sudah terdaftar di dalam *database*, dan pada halaman *dashboard* pelajar terdapat beberapa *menu* terkait materi pembelajaran seperti:

- HTML Dasar
- HTML Lanjutan
- CSS Dasar
- CSS Lanjutan
- Javascript Dasar
- Javascript Lanjutan
- Git & Github
- Node JS
- React JS
- Vue JS
- Angular JS
- *Final Study*

**UNIVERSITAS
ISLAM RIAU**



Gambar 4.9 Dashboard Pengajar

Pada gambar 4.9 adalah halaman *dashboard user* pengajar, pada halaman *dashboard* pengajar di *menu home* memiliki perbedaan dengan *dashboard* pelajar dimana di *menu* ini *user* pengajar dapat memasukkan materi terbaru, melihat daftar pelajar serta memasukkan soal tugas akhir. Pada *menu* sendiri sisanya kurang lebih sama pada pelajar namun memiliki perbedaan dimana bagi *user* pengajar dapat melakukan perubahan materi dan menghapus materi.



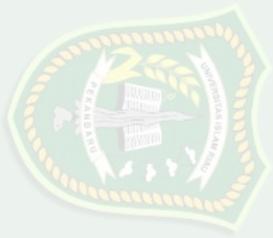
Gambar 4.10 Dashboard Admin

Pada gambar 4.10 adalah halaman *dashboard* dari *user admin* dimana pada halaman *dashobard* admin, *user admin* dapat melakukan beberapa aktivitas seperti melihat *profile pengajar* dan *pelajar*, serta dapat menghapus *data* pengajar dan pelajar selain itu *user admin* dapat menambahkan *user* pengajar dan juga memberikan sertifikat pada pelajar apabila sudah memiliki nilai diatas 75 atau sama dengan 75.

Kemudian dari hasil pengujian halaman login yang telah dilakukan, dapat dilihat rinciannya pada tabel 4.2.

Tabel 4.2 Pengujian Halaman *Login*

No	Komponen Yang Diuji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
1	Halaman <i>Login</i>	Mengosongkan semua <i>field</i> yang disediakan seperti <i>email</i> dan <i>password</i> , kemudian menekan tombol <i>login</i> .	Sistem akan menolak <i>login</i> dan akan memberikan <i>error message</i> seperti “ <i>Form Tidak Boleh Ada Yang Kosong</i> ”.	Berhasil
		Mengosongkan salah satu <i>field</i> yang ada di halaman <i>login</i> , kemudian menekan tombol <i>login</i> .	Sistem akan menolak <i>login</i> dan akan memberikan <i>error message</i> seperti “ <i>Form Tidak Boleh ada Yang Kosong</i> ”.	Berhasil
		Mengisi <i>data user</i> yang salah atau tidak terdaftar di dalam <i>database</i> sistem.	Sistem akan menolak <i>login</i> dan akan memberikan <i>error message</i> seperti “ <i>Email anda belum terdaftar</i> ”.	Berhasil

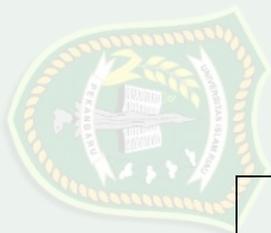


	Mengisi <i>data user</i> dengan <i>email</i> yang benar dan <i>password</i> yang tidak sesuai dengan <i>data user</i> .	Sistem akan menolak <i>login</i> dan akan memberikan <i>error message</i> seperti “ <i>email</i> atau <i>password</i> anda salah”.	Berhasil
	Mengisi <i>data</i> dengan benar dengan <i>data</i> pelajar seperti <i>Email: eri@student.uir.ac.id</i> <i>Password: *****</i>	Maka sistem akan langsung masuk kedalam <i>dashboard</i> pelajar.	Berhasil
	Mengisi <i>data</i> dengan benar dengan <i>data</i> pengajar seperti <i>email: rizky@gmail.com</i> <i>password: *****</i>	Maka sistem akan langsung masuk kedalam <i>dashboard</i> pengajar.	Berhasil
	Mengisi <i>data</i> dengan benar dengan <i>data admin</i> seperti <i>email: admin@admin.com</i> <i>password: *****</i>	Maka sistem akan langsung masuk kedalam <i>dashboard admin</i> .	Berhasil

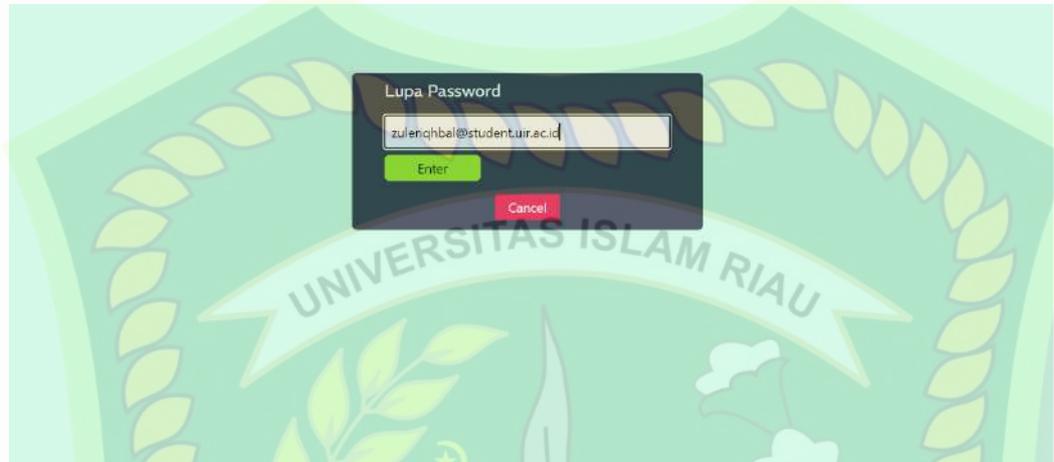
Dari pengujian yang dilakukan pada halaman login, maka dapat disimpulkan bahwa setiap aspek dan fungsinya telah berjalan sesuai harapan.

4.2.3 Pengujian Halaman Lupa *Password*

Halaman lupa *password* adalah halaman yang memberikan kemudahan bagi *user* yang lupa dengan *password*-nya, nantinya harapan dari hasil pengujian di halaman lupa *password* ini adalah *data password*

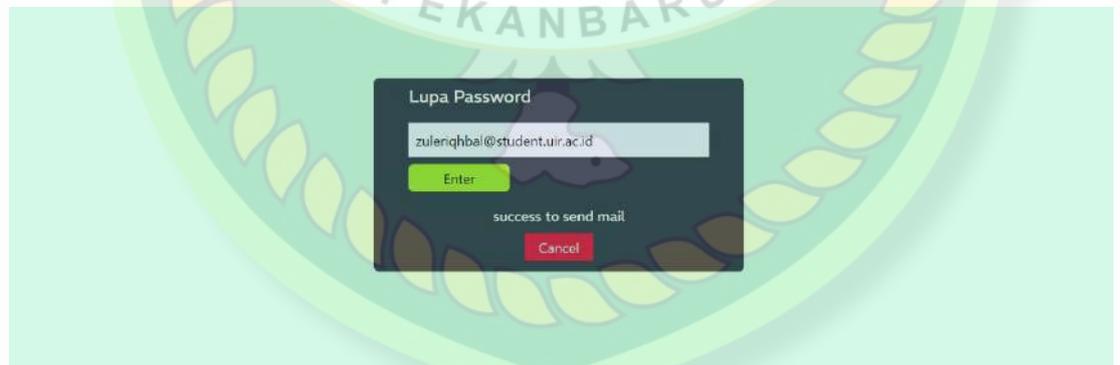


akan terkirim ke *email* secara otomatis setelah *user* mengisi *field email* dan menekan tombol *enter*, seperti pada gambar 4.11.



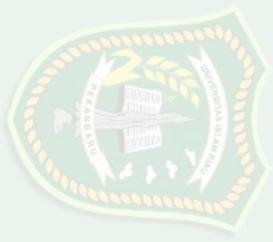
Gambar 4.11 Halaman Lupa password

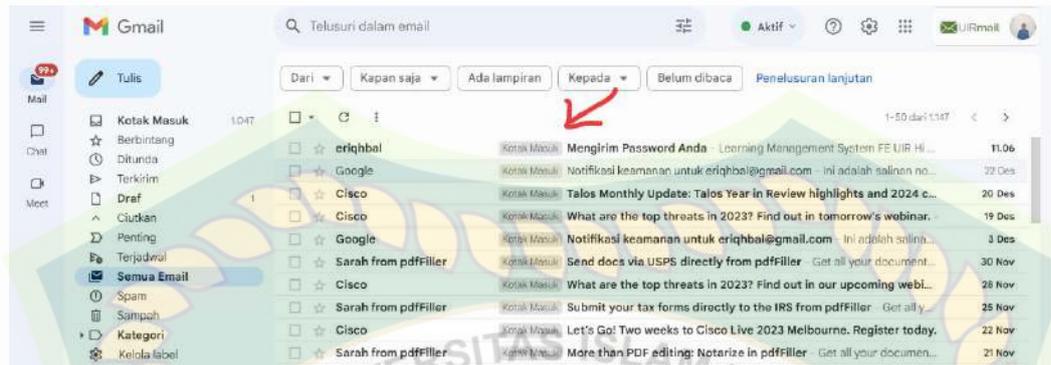
Dapat dilihat pada gambar 4.11 diatas setelah *user* mengisi alamat *email* yang sesuai dengan *data* yang terdaftar sebelumnya, maka *sistem* akan mengirimkan response message success seperti pada gambar 4.12.



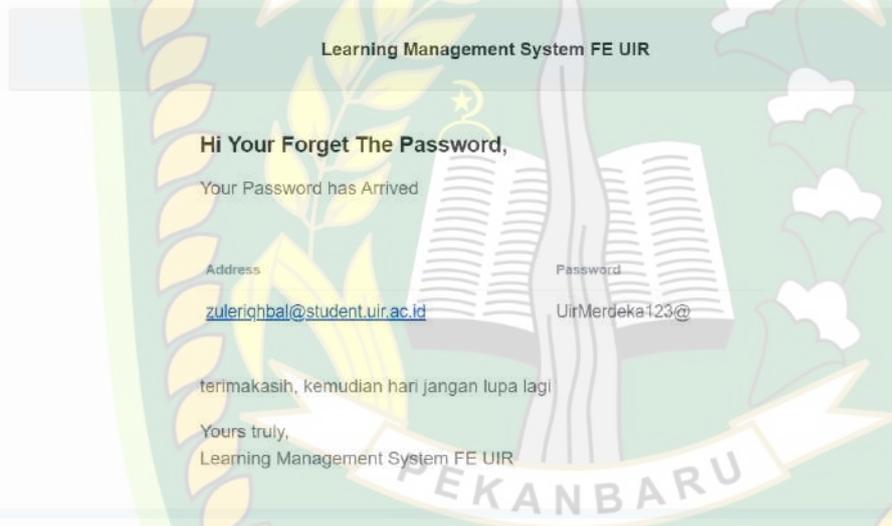
Gambar 4. 12 success message

Kemudian setelah itu *user* dapat melihat pada *email* apakah pesan sudah masuk yang dikirimkan dari sistem berupa *data user* dan passwordnya, untuk memastikannya dapat dilihat pada gambar 4.13.





Gambar 4.13 send to mail



Gambar 4.14 Send Data to Email

Dapat dilihat pada gambar 4.13 dan 4.14 adalah pesan berupa *data user* yang dikirimkan oleh sistem apabila *user* mengisi *data email* dengan benar, namun apabila *user* mengisi *email* dengan yang tidak terdaftar di dalam *database* maka sistem tidak akan merespon

Untuk melihat keseluruhan rincian dari pengujian halaman lupa password ini dapat dilihat pada table 4.3.

Tabel 4.3 pengujian halaman lupa *password*

No	Komponen yang Diuji	Skenario Pengujian	Hasil yang diharapkan	Hasil

1	Halaman Lupa Password	User tidak mengisi <i>field email</i> kemudian menekan tombol <i>enter</i> .	Sistem tidak dapat merespon apapun pada aktivitas <i>user</i> .	Berhasil
		User mengisi alamat <i>email</i> yang salah dan tidak terdaftar di <i>database</i> sistem.	Sistem tidak dapat merespon apapun pada aktivitas <i>user</i> .	Berhasil
		User mengisi alamat <i>email</i> dengan benar dan terdaftar didalam <i>database</i> sistem.	Sistem akan merespon dan mengirimkan <i>data user</i> ke alamat <i>email user</i> .	Berhasil

Dari hasil pengujian yang sudah dilakukan pada halaman lupa *password* maka dapat disimpulkan setiap aspek dan fungsi yang diharapkan telah berhasil.

4.2.4 Pengujian Halaman *Dashboard*

Halaman *dashboard* adalah halaman utama untuk setiap aktivitas *user* baik itu pelajar, pengajar, dan *admin*. Setiap *role* memiliki perannya masing-masing di halaman *dashboard*-nya, berikut lebih untuk lebih jelasnya.

4.2.4.1 Pengujian Halaman *Dashboard Admin*

Pada halaman *dashboard admin*, *user* dapat melakukan beberapa hal seperti melihat *data* pengajar dan pelajar, menambahkan *data* pengajar, menghapus *data* pengajar dan pelajar, serta

menginputkan sertifikat pada *user* pelajar. Berikut untuk lebih jelasnya dilihat pada gambar 4.15.



Gambar 4.15 Lihat Profile User

Dapat dilihat bahwa pada gambar 4.15 dengan arah panah yang ditunjukkan adalah *button* untuk melihat *data* setiap *user*, apabila di tekan *button* tersebut pada *menu* daftar pengajar maka *admin* dapat melihat *user* pengajar, berikut lebih jelasnya dapat dilihat pada gambar 4.16.



Gambar 4.16 Profile Pengajar

Pada gambar 4.16 adalah tampilan dimana ketika *admin* menekan tombol *profile* di daftar pengajar maka terlihat jelas *data* dari pengajar mulai dari nama lengkap, *email* dan *password*-nya.

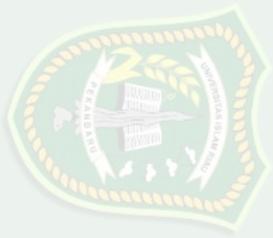
Kemudian apabila *user admin* menekan tombol *profile* di daftar pelajar maka *admin* dapat melihat *profile* pelajar sekaligus pada *menu* tersebut juga dapat memasukkan sertifikat apabila nilai sih pelajar memenuhi kriteria namun apabila tidak memenuhi kriteria maka *admin* tidak dapat menginputkan sertifikat kepada pelajar tersebut, untuk lebih jelasnya dapat dilihat pada gambar 4.17.



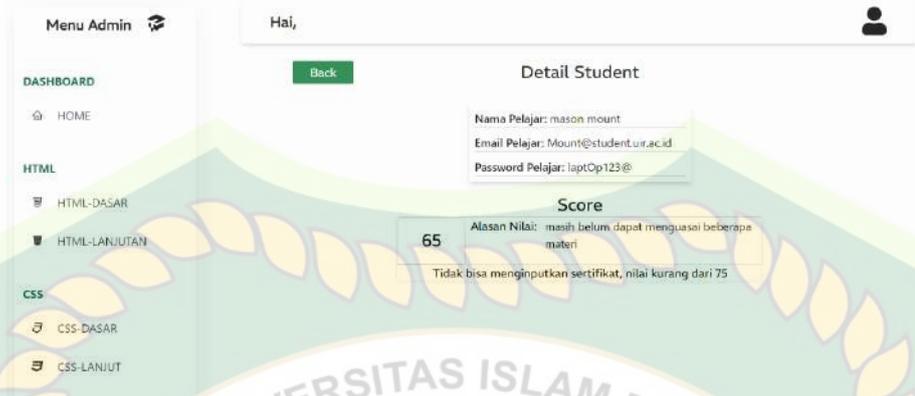
Gambar 4.17 Detail Pelajar Belum Dinilai

Pada gambar 4.17 adalah gambar yang menjelaskan bahwa sih pelajar tersebut belum mengerjakan tugas akhir yang diberikan sih pengajar, sehingga pelajar tersebut tidak memiliki nilai dan *user admin* juga tidak dapat menginputkan sertifikat kepada *user* pelajar tersebut. Namun apabila *user* pelajar memiliki nilai tetapi dibawah atau sama dengan 75 maka akan terdapat *response message* seperti gambar 4.18.

UNIVERSITAS
ISLAM RIAU

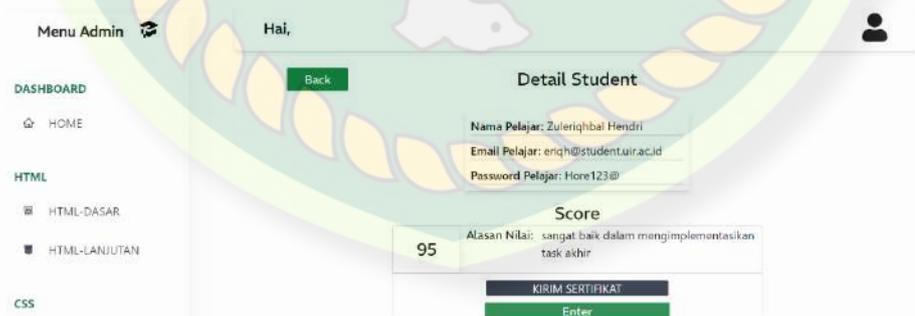


DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU



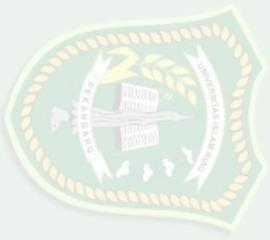
Gambar 4.18 nilai pelajar dibawah kriteria

Pada gambar 4.18 diatas, menjelaskan bahwa *user* pelajar sudah memiliki nilai tetapi nilai yang ada masih rendah dan dibawah kriteria yang ditentukan, sehingga *user admin* tidak dapat melakukan penginputan sertifikat kepada pelajar tersebut, namun bagi *user* pelajar yang memiliki nilai diatas kriteria yang ditentukan maka tampilannya berbeda dengan gambar-gambar sebelumnya dimana *user admin* dapat melakukan penginputan sertifikat, untuk lebih jelasnya dapat dilihat pada gambar 4.19.



Gambar 4.19 Profile Pelajar Memenuhi Kriteria

Pada gambar 4.19 *user* pengajar dapat menginputkan sertifikat terhadap pelajar tersebut karena sudah memiliki nilai yang



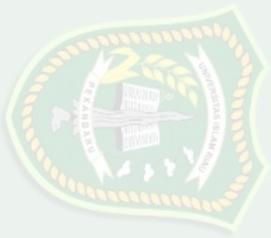
memenuhi kriteria, apabila *user admin* menekan tombol kirim sertifikat kemudian tekan *enter* maka akan terdapat *pop up* yang menunjukkan bahwa penginputan sertifikat akan dilakukan, untuk lebih jelasnya dapat dilihat pada gambar 4.20.



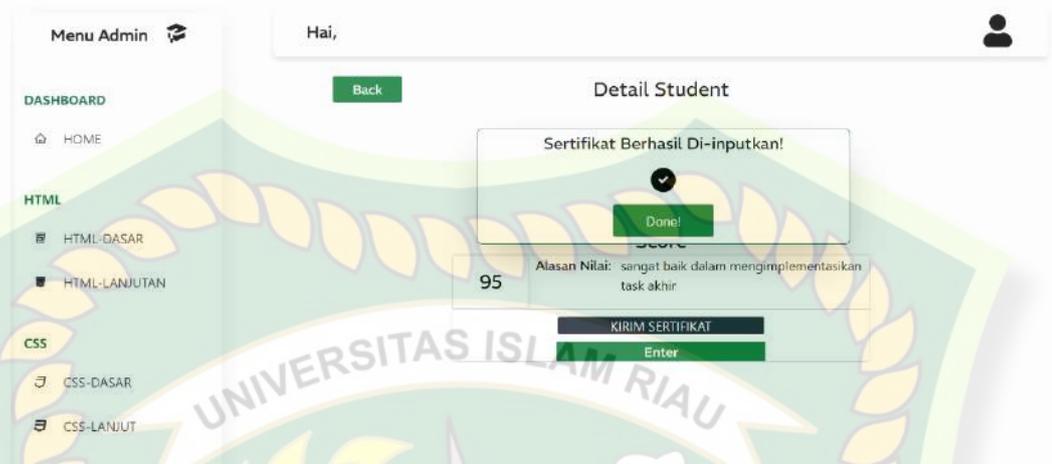
Gambar 4.20 *Pop up* memberikan sertifikat

Setelah *user admin* menekan tombol kirim sertifikat maka akan muncul *pop up* yang membantu untuk mengirimkan sertifikat kepada sih pelajar tersebut, lalu menekan tombol *enter* maka nantinya akan muncul pemberitahuan bahwa sertifikat berhasil dikirim ke pelajar tersebut, untuk lebih jelasnya dapat dilihat pada gambar 4.21.

UNIVERSITAS
ISLAM RIAU



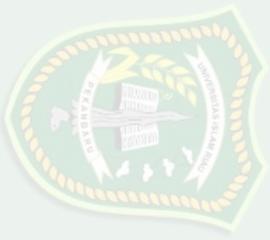
DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU



Gambar 4.21 Sukses kirim sertifikat pelajar

Pada gambar 4.21 diatas menunjukkan pemberitahuan bahwa penginputan sertifikat telah berhasil dilakukan. Selanjutnya pada *dashboard admin* kita juga dapat melakukan beberapa hal lagi seperti menghapus *data user* pengajar dan pelajar, hal ini sangat diperlukan sewaktu-waktu *user* tersebut tidak ada berkepentingan lagi dengan sistem seperti pelajar yang sudah menyelesaikan semua *course* dan benar-benar bisa dibidang *frontend development*, maupun pengajar yang sudah tidak bisa membantu proses pembelajaran lagi, untuk lebih jelasnya dapat dilihat pada gambar

**UNIVERSITAS
ISLAM RIAU**



**DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS**

UNIVERSITAS ISLAM RIAU

4.22.



Gambar 4.22 Remove User Pengajar

Pada gambar 4.22 adalah tombol untuk menghapus *data* pengajar, jika kita mengklik tombol tersebut maka *data* pengajar akan terhapus, misal kita ingin menghapus *data user* pengajar yang bernama aldo brodo maka *data user* aldo brodo akan terhapus.

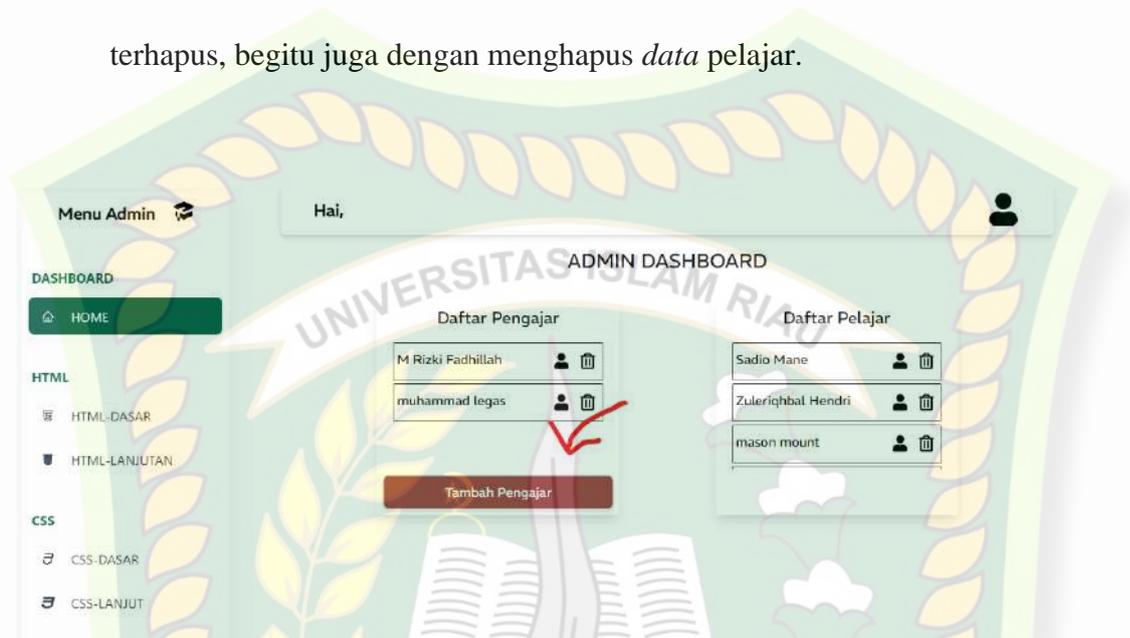


Gambar 4.23 Hapus User Pelajar

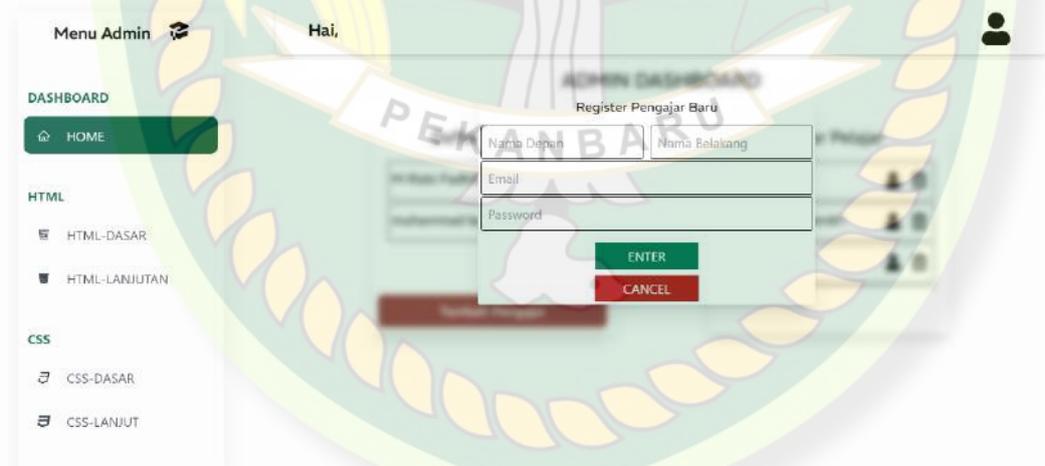
UNIVERSITAS
ISLAM RIAU



Pada gambar 4.23 diatas adalah contoh menghapus *data user*, terlihat akun pengajar yang bernama aldo brodo sudah terhapus, begitu juga dengan menghapus *data* pelajar.

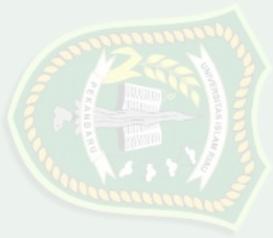


Gambar 4.24 Tombol Tambah Register



Gambar 4.25 Register Pengajar

Pada gambar 4.25 adalah gambar untuk mendaftarkan sebuah akun baru untuk *user* pengajar jika menekan tombol Tambah Pengajar maka akan muncul *pop up* seperti pada gambar 4.25, jika *user admin* mengosongkan semua *field* yang tersedia atau



mengosongkan salah satu *field* maka akan terdapat pesan peringatan “Please fill out this field” seperti pada gambar 4.26.

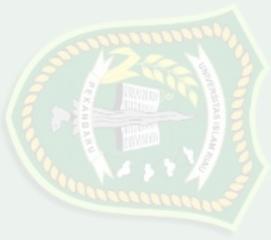
The image shows a registration form titled "Register Pengajar Baru". It contains three input fields: "Nama Depan", "Nama Belakang", and "Password". The "Nama Belakang" field is highlighted with a red border and a yellow warning icon, with a tooltip message that says "Please fill out this field.". Below the fields are two buttons: a green "ENTER" button and a red "CANCEL" button.

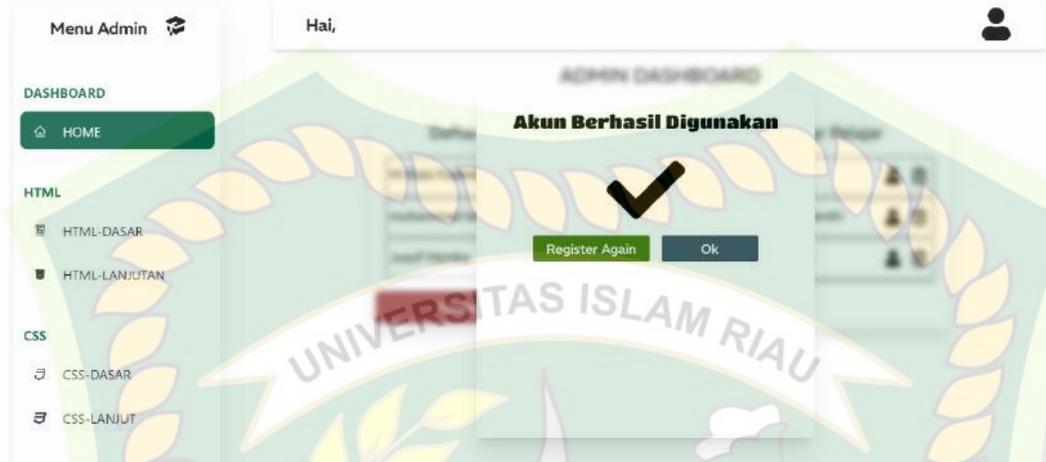
Gambar 4.26 Error Message

Gambar diatas adalah contoh apabila *user admin* tidak mengisi *field* yang disediakan, tetapi jika *user admin* mengisi semua *field* maka akan tampil *pop up* sebagai pemberitahuan bahwa akun berhasil didaftarkan, berikut penjelasannya pada gambar 4.27 hingga 4.28.

The image shows the same registration form as in Gambar 4.26, but now all fields are filled out. The "Nama Depan" field contains "Jusuf", the "Nama Belakang" field contains "Hamka", the "Email" field contains "Jusuf@gmail.com", and the "Password" field contains "Keluarkan123@". The "ENTER" and "CANCEL" buttons are still visible below the fields.

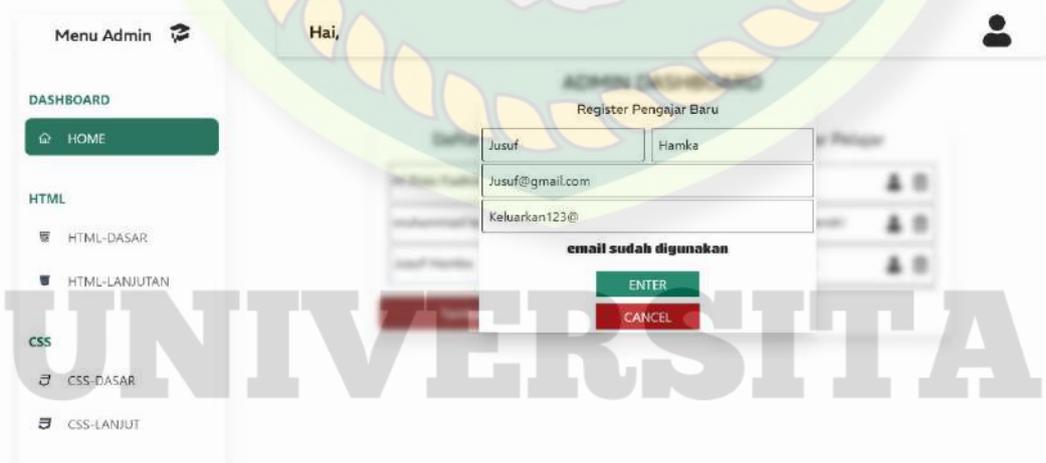
Gambar 4.27 Mengisi Semua *Field*





Gambar 4.28 Berhasil Mendaftarkan Akun

Pada contoh di gambar 4.27 adalah ketika semua *field* terisi kemudian *user admin* akan menekan tombol *enter* dan akan muncul *pop up* seperti pada gambar 4.28 yang menandakan pendaftaran akun pengajar telah berhasil didaftarkan. Tetapi jika *user admin* mengisi dengan *email* yang sama maka akan terdapat pesan “*email sudah digunakan*” seperti pada gambar 4.29.



Gambar 4.29 Error Message Email Sudah Digunakan

Tabel 4.4 Pengujian Halaman *Dashboard Admin*

No	Komponen yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil
1	Melihat <i>Profile User</i> pengajar dan pelajar.	Menekan tombol <i>icon profile user</i> yang ada di daftar <i>user</i> pengajar dan pelajar.	<i>Admin</i> dapat melihat setiap <i>data user</i> secara lengkap.	Berhasil
2	Melihat <i>profile detail user</i> pelajar dan melihat tidak ada <i>score</i> .	Menekan tombol <i>icon profile user</i> yang ada di daftar <i>user</i> pelajar.	Halaman detail pelajar akan menampilkan <i>data</i> pelajar dan menampilkan pesan “Pelajar belum memiliki nilai”	Berhasil
3	Melihat <i>profile detail user</i> pelajar dan melihat <i>score</i> dibawah atau sama dengan 75	Menekan tombol <i>icon profile user</i> yang ada di daftar <i>user</i> pelajar.	Halaman detail pelajar akan menampilkan <i>data</i> pelajar dan menampilkan pesan “Tidak bisa menginputkan sertifikat, nilai kurang dari 75”.	Berhasil
4	Melihat <i>profile detail user</i> pelajar dan melihat <i>score</i> diatas 75	Menekan tombol <i>icon profile user</i> yang ada di daftar <i>user</i> pelajar.	Halaman <i>detail</i> pelajar akan menampilkan <i>data</i> pelajar dan bisa menginputkan	Berhasil

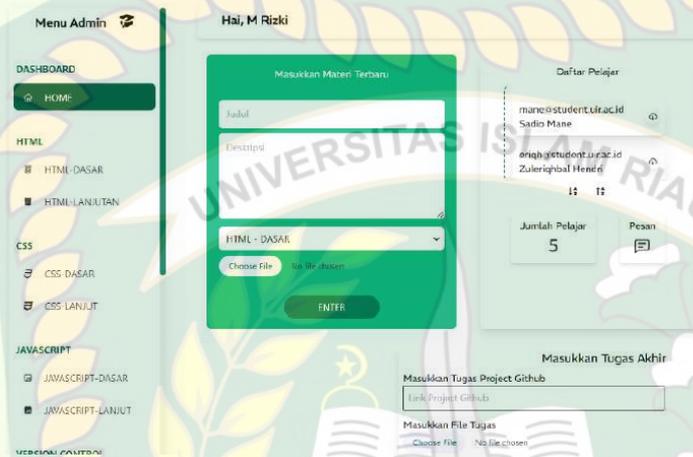


			sertifikat kepada pelajar tersebut.	
5	Menghapus <i>user</i> pengajar dan pelajar	Menekan tombol <i>icon trash</i> yang ada di setiap daftar <i>user</i> baik pengajar maupun pelajar.	Daftar pengajar dan pelajar akan menghilangkan <i>user</i> yang dihapus.	Berhasil
6	Menambahkan <i>user</i> pengajar	Menekan tombol “Tambah Pengajar” dan tidak mengisi semua <i>field</i> yang disediakan atau tidak mengisi salah satu	Sistem akan menolak pendaftaran pengajar tersebut dan akan memberikan <i>warning message</i> ‘ <i>please fill out this field</i> ’.	Berhasil
		Menekan tombol “tambah pengajar” dan mengisi semua <i>field</i> yang disediakan.	Sistem akan menerima <i>data user</i> pengajar baru yang dibuat oleh <i>user admin</i> .	Berhasil

4.2.4.2 Pengujian Halaman *Dashboard* Pengajar

Pengujian pada halaman *dashboard* pengajar bertujuan untuk menguji beberapa fungsi yang ada di halaman *dashboard* pengajar seperti, menambahkan materi, melihat dan menginputkan nilai untuk para pelajar, serta menginputkan atau menambah soal tugas akhir.

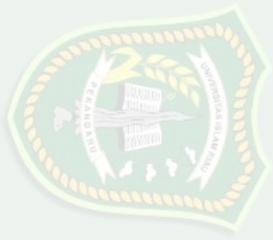
Dalam pengujian ini diharapkan semua fitur-fitur yang dibutuhkan berjalan sesuai perencanaan, untuk melihat semua fitur tersebut dapat dilihat pada gambar 4.30.



Gambar 4.30 Dashboard Pengajar

Pada gambar 4.30 kita bisa melakukan masukkan materi terbaru, dan melihat isi *profile* pelajar dengan menekan *icon* awan, dan juga dapat melakukan penginputan tugas akhir. Untuk memasukkan materi pembelajaran *user* pengajar diwajibkan untuk mengisi semua *field* yang tersedia, namun apabila *user* pengajar tidak mengisi salah satu *field* maka akan muncul pesan *error* seperti pada gambar 4.31.

UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU

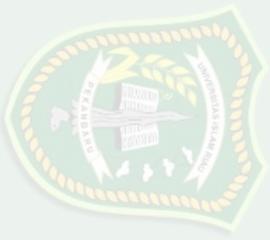
The screenshot shows a web form with the following elements:

- Title:** Masukkan Materi Terbaru
- Input Fields:**
 - CSS Position:** A text input field.
 - Deskripsi:** A large text area for description.
 - CSS - DASA:** A dropdown menu with an error message: "Please fill out this field."
- File Upload:** A button labeled "Choose File" with the text "No file chosen" below it.
- Submit:** A large green button labeled "ENTER".

Gambar 4.31 Error Messages

Terlihat pada gambar 4.31 diatas proses penambahan materi tidak dapat dilakukan dikarenakan setiap *field* yang tersedia belum terisi, namun apabila semua *field* terisi maka *field* akan kosong kembali yang menandakan berhasilnya proses penginputan *data* materi seperti di gambar 4.32 hingga 4.33.

**UNIVERSITAS
ISLAM RIAU**



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU

Masukkan Materi Terbaru

CSS Position

Materi ini akan mempelajari tentang penggunaan position yang akan digunakan

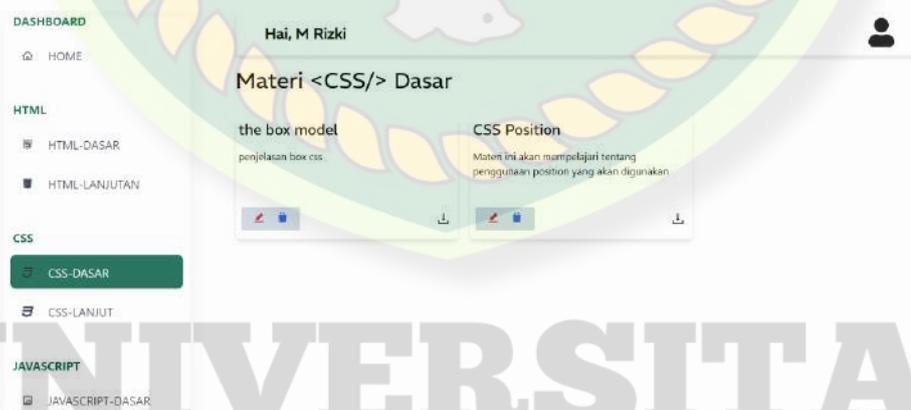
CSS - DASAR

Choose File

CSS Dasar 3.pdf

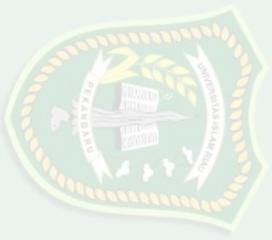
ENTER

Gambar 4.32 Field Terisi Semua



Gambar 4.33 Materi Baru di Menu

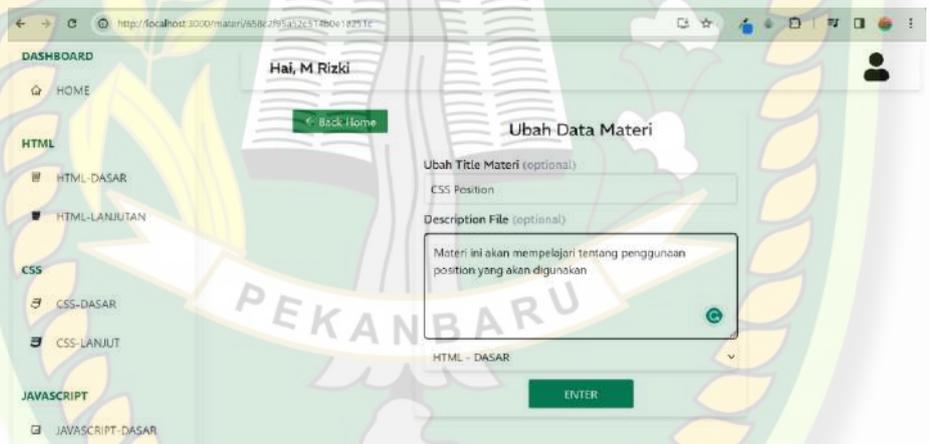
Seperti pada gambar 4.32 *user* pengajar mengisi semua *field* yang ada, kemudian nantinya materi yang dimasukkan akan muncul di *menu*



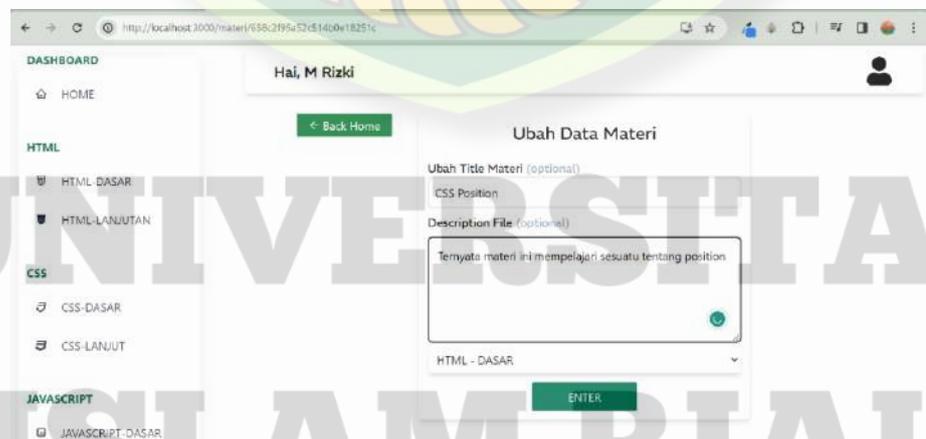
kategori yang sudah ditentukan diawal, pada gambar 4.33 adalah *menu* css-dasar yang berisi materi-materi yang diinputkan tadi, disini *user* pengajar dapat melakukan beberapa hal seperti *mengedit* materi dan menghapus materi, untuk lebih detailnya dapat dilihat pada *sub menu* dibawah.

a. *Mengedit Materi*

Mengedit materi dapat dilakukan apabila *user* pengajar salah dalam penulisan judul, deskripsi materi, atau kategori materi saat penginputan diawal, pada gambar 4.34 adalah gambar halaman *field edit* materi.

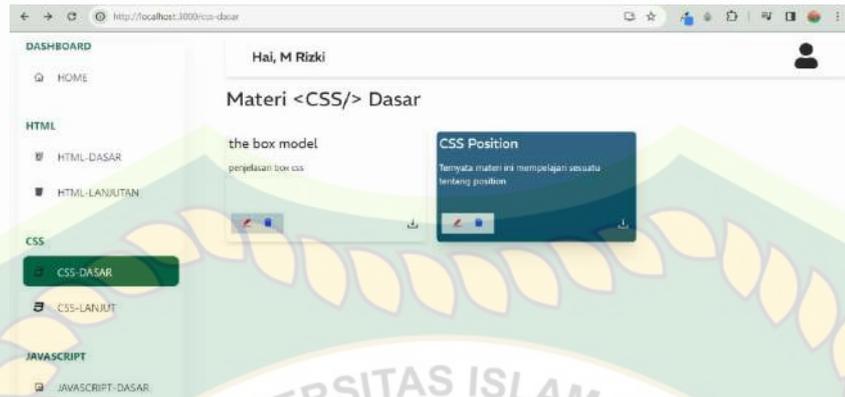


Gambar 4.34 *Field Edit* Materi



Gambar 4.35 *Ubah Deskripsi*





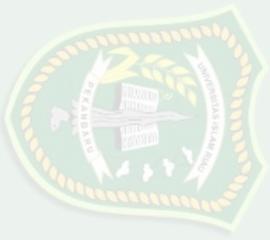
Gambar 4.36 Berhasil Ubah Dekripsi Materi

Pada gambar 4.35 diatas adalah pengujian untuk merubah deskripsi materi sebelumnya, lalu apabila semua sudah dirasa yakin *user* pengajar bisa langsung menekan tombol *enter* dan pada gambar 4.36 kita dapat melihat bahwa deskripsi materi pembelajaran berhasil diubah.

b. Menghapus materi

User pengajar selain dapat melakukan ubah *data* materi atau *edit* materi juga dapat melakukan penghapusan *data* pada materi tersebut, hal ini memungkinkan apabila *user* pengajar salah memasukkan *data* materi pembelajaran kemudian ingin menghapusnya, dalam pengujian hapus materi *user* pengajar bisa menekan tombol tempat sampah, untuk lebih jelasnya dapat dilihat

**UNIVERSITAS
ISLAM RIAU**

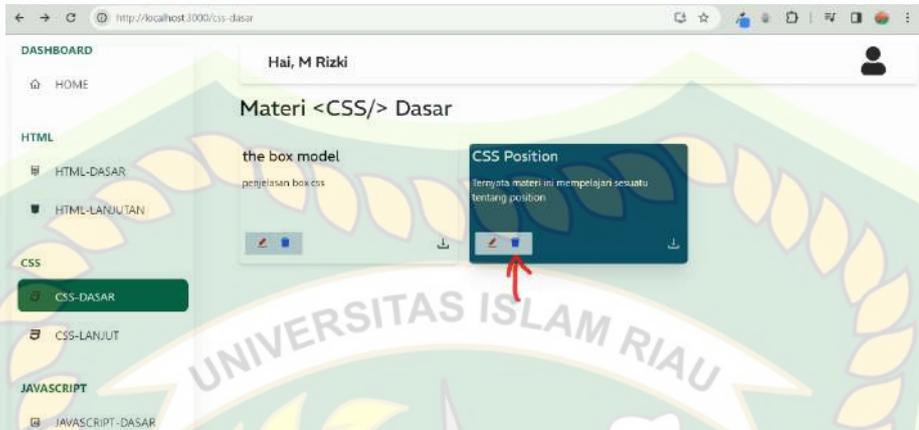


DOKUMEN INI ADALAH ARSIP MILIK :

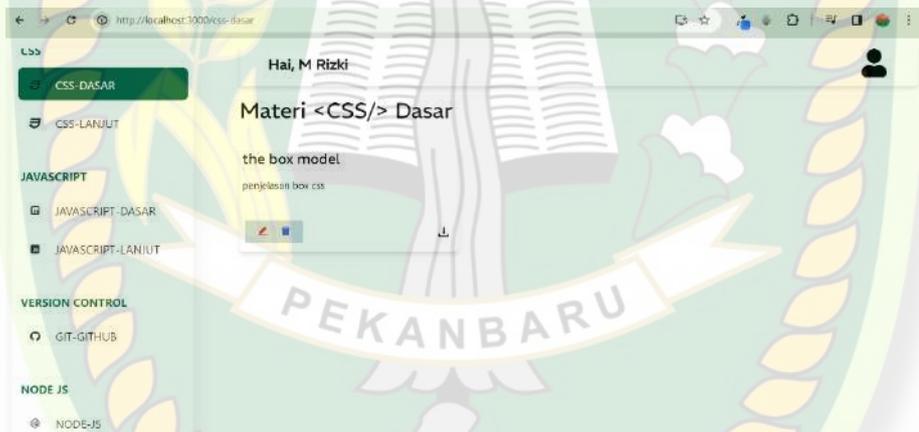
PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU

pada gambar 4.37 hingga 4.38 dibawah ini.



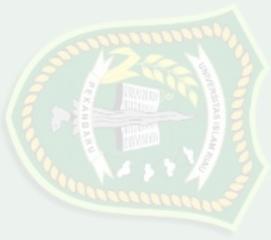
Gambar 4.37 Remove Materi



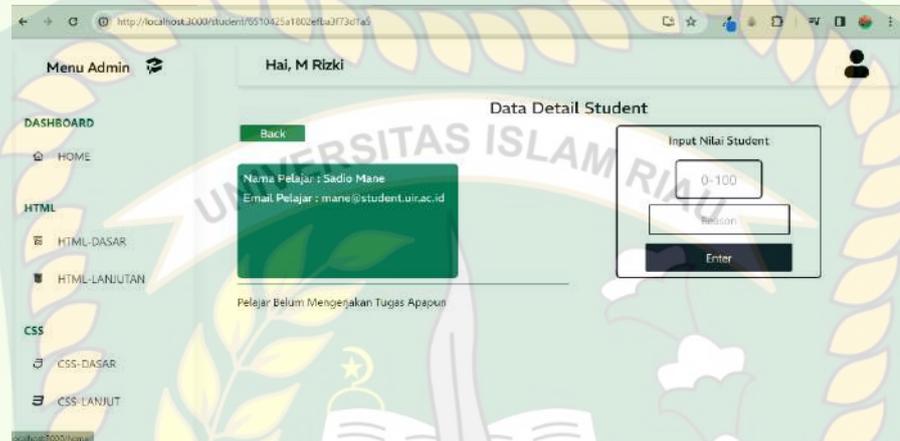
Gambar 4. 38 Success Remove Materi

Pada gambar 4.37 adalah tombol yang digunakan untuk menghapus materi, apabila *user* pengajar menekan tombol tersebut maka materi akan terhapus seperti pada gambar 4.38 materi CSS *Position* sudah hilang.

Setelah pengujian penginputan materi, *mengedit* materi dan menghapus materi berhasil dilakukan, maka pengujian selanjutnya adalah terkait melihat *data* siswa baik *data* pengguna maupun dapat



melihat apakah *user* pelajar tersebut sudah mengerjakan soal tugas akhirnya terkait pembelajaran atau belum, untuk lebih jelasnya dapat dilihat pada gambar 4.39 hingga 4.40



Gambar 4.39 Halaman *Detail* Pelajar

Terlihat pada gambar diatas adalah halaman detail *user* pelajar dimana terdapat beberapa informasi seperti nama pelajar dan *email* pelajar, di halaman ini juga *user* pengajar bisa melakukan penginputan nilai tugas akhir yang sudah menyelesaikan keseluruhan *course* dan tugas akhirnya, namun pada gambar diatas adalah *case* dimana pelajar belum mengerjakan tugas apapun, sehingga *user* pengajar mungkin tidak akan melakukan penginputan nilai terhadap pelajar tersebut. Lalu pada gambar 4.40 adalah contoh halaman detail *user* pelajar yang sudah mengerjakan *course*

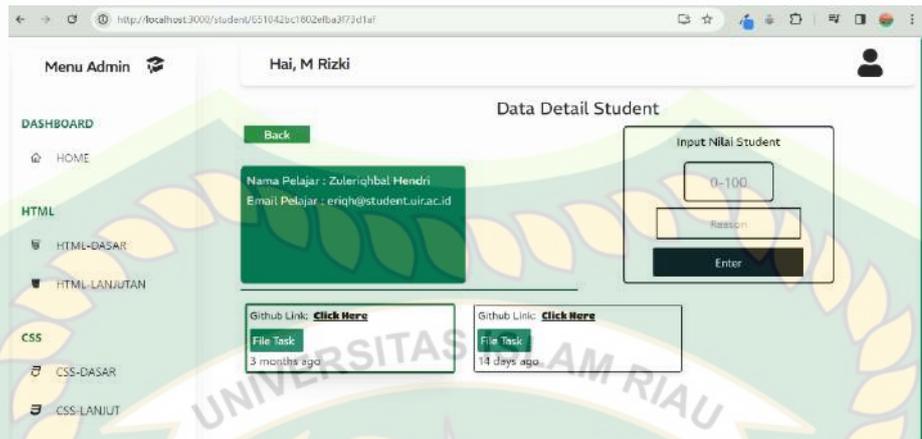
**UNIVERSITAS
ISLAM RIAU**



DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU



Gambar 4.40 Halaman Detail Pelajar Mengerjakan *Task*

pada gambar diatas adalah contoh halaman detail pelajar yang menunjukkan bahwasannya pelajar tersebut sudah mengerjakan *task* tugas akhir yang ada di sistem *E-Learning* pembelajaran *frontend development*, sehingga *user* pengajar nantinya akan melakukan penginputan nilai terhadap pelajar tersebut, untuk melihat lebih jelasnya dapat dilihat pada gambar 4.41 hingga 4.42.



Gambar 4.41 Pesan Peringatan *Form*

Gambar 4.41 diatas adalah contoh apabila *form* penilaian ada yang kosong atau *form* jika kosong semua, pesan warning ini bertujuan supaya *user* pengajar jika menginputkan nilai terhadap

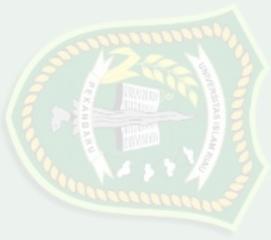


user pelajar harus lengkap dan tidak boleh ada yang kosong di *data* terkait penilaian pelajar. Lalu pada gambar 4.42 adalah contoh pengerjaan yang benar soal penginputan nilai dari pelajar, untuk lebih jelasnya dapat dilihat pada gambar dibawah.



Gambar 4.42 Form Berhasil Input Nilai

Pada gambar 4.42 diatas adalah contoh halaman detail pelajar yang berhasil dalam menginputkan nilai untuk pelajar. lalu aktivitas terakhir yang dapat dilakukan di halaman *dashboard user* pengajar adalah penginputan soal tugas akhir, yang mana soal yang diinputkan adalah soal yang menentukan apakah pelajar benar-benar memahami pembelajaran yang diberikan atau tidak, selain itu dengan adanya soal tugas akhir juga dapat melatih logical dan pemahaman terkait pengodingan yang ada di dunia industri, berikut adalah gambar terkait *form* yang ada di halaman *dashboard* pengajar dapat dilihat pada gambar 4.43

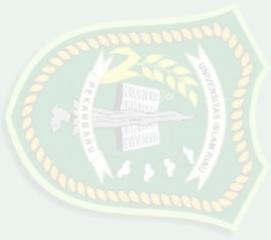


Gambar 4.43 Form Soal Tugas Akhir

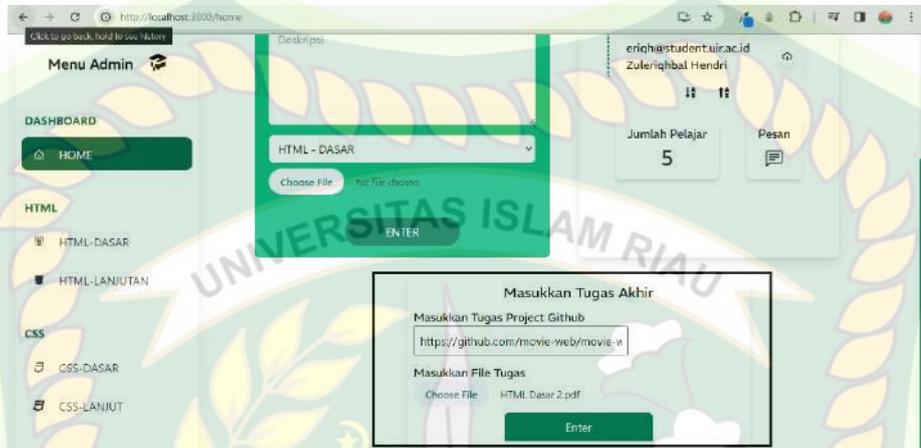
Pada gambar diatas yang didalam kotak garis merah adalah *form* untuk melakukan penginputan soal tugas akhir apabila *form* tersebut tidak terisi atau salah satu nya kosong maka *form* tersebut akan merespon dengan memberikan pesan warning, berikut lebih jelasnya dapat dilihat pada gambar 4.44.

Gambar 4.44 Field Tugas Akhir Tidak Terisi

Pada gambar diatas adalah contoh pesan warning apabila *field* tugas akhir tidak terisi semua, lalu jika semua *field* yang ada di *form* tugas akhir terisi maka seluruh isi yang ada di *form* akan hilang yang mana artinya inputan tugas akhir telah berhasil dilakukan, berikut gambar

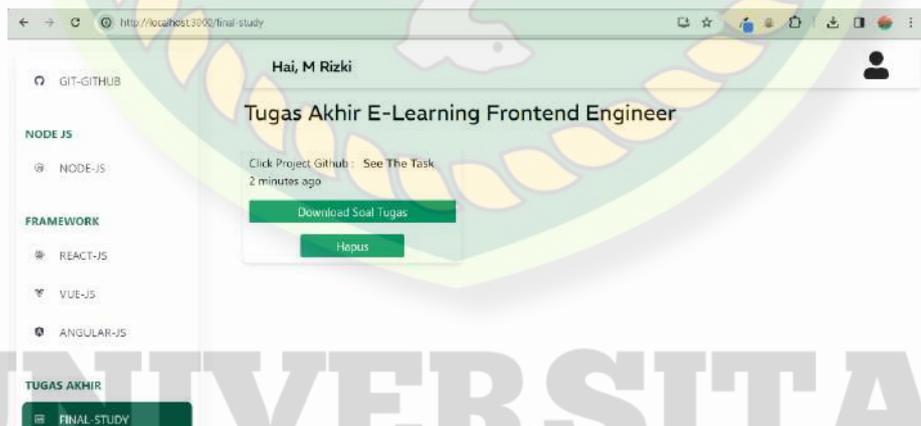


4.45 dibawah adalah contoh dari *form* yang telah berhasil dilakukan penginputan.



Gambar 4.45 *Field* Tugas Akhir Terisi

Pada gambar 4.45 diatas adalah contoh *form* masukkan tugas akhir terisi semua dan jika *user* pengajar menekan tombol *enter* maka *data* soal tugas akhir akan tersimpan di *database* sistem, lalu untuk membuktikan suksesnya penginputan bisa kita lihat pada gambar 4.46.



Gambar 4.46 *Menu* Tugas Akhir

UNIVERSITAS
ISLAM RIAU



Pada gambar 4.46 diatas adalah contoh bahwa penginputan soal tugas akhir pembelajaran *frontend development* telah berhasil dilakukan dan menunjukkan kapan soal tersebut dibuat.

Berikut adalah rincian dari seluruh aktivitas yang bisa dilakukan di halaman *dashboard* pengajar dapat dilihat pada tabel 4.5.

Tabel 4.5 Pengujian Halaman *Dashboard* Pengajar

No	Komponen Yang diuji	Skenario Pengujian	Hasil yang diharapkan	Hasil
1	Menambah Materi Pembelajaran	Tidak mengisi semua <i>field</i> yang ada di <i>form</i> .	<i>Form</i> akan merespon dengan memberi pesan peringatan “ <i>Please fill out this field</i> ”.	Berhasil
		Tidak mengisi salah satu <i>field</i> yang ada di <i>form</i> .	<i>Form</i> akan merespon dengan memberi pesan peringatan “ <i>Please fill out this field</i> ”.	Berhasil
		Mengisi semua <i>field</i> yang ada di <i>form</i> pengisian materi terbaru.	<i>Form</i> akan mereset semua <i>field</i> yang ada di <i>form</i> masukkan materi terbaru.	Berhasil
2	Mengedit Materi Pembelajaran	Tidak mengisi semua <i>field</i> yang ada di <i>form</i> mengedit materi pembelajaran.	Sistem akan merespon dengan memberi pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil
		Tidak mengisi salah satu <i>field</i> yang ada di	Sistem akan merespon dengan memberi pesan	Berhasil

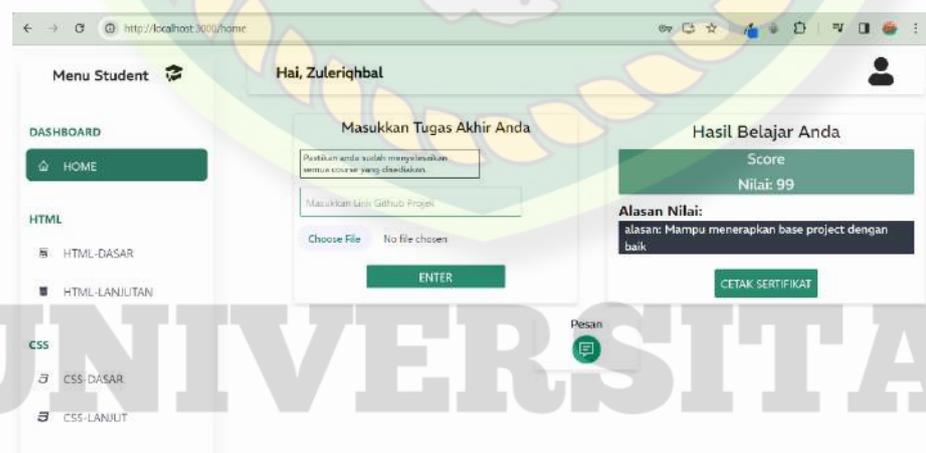


		<i>form</i> mengedit materi pembelajaran.	peringatan “ <i>please fill out this field</i> ”.	
		Mengisi semua <i>field</i> yang ada di <i>form</i> mengedit materi pembelajaran.	Sistem akan kembali ke menu pembelajaran dan melihat <i>data</i> materi berubah.	Berhasil
3	Menghapus Materi Pembelajaran	Menekan tombol <i>icon</i> trash pada setiap card pembelajaran.	Sistem akan menghapus materi pembelajaran.	Berhasil
4	Menginputkan Nilai para pelajar	Tidak mengisi semua <i>field</i> yang ada <i>form</i> penginputan nilai.	Sistem akan menolak dan memberi pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil
		Tidak mengisi salah satu <i>field</i> yang ada di <i>form</i> penginputan nilai.	Sistem akan menolak dan memberi pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil
		Mengisi semua <i>field</i> yang ada di <i>form</i> penginputan nilai pelajar.	Sistem akan memberikan pop up yang bertuliskan “nilai berhasil diinputkan”.	Berhasil
5	Menambahkan soal tugas akhir pembelajaran <i>frontend development</i>	Tidak mengisi semua <i>field</i> yang ada di <i>form</i> soal tugas akhir.	Sistem akan menolak dan memberikan pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil
		Tidak mengisi salah satu <i>field</i> yang ada di <i>form</i> soal tugas akhir	Sistem akan menolak dan memberikan pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil

		Mengisi semua <i>field</i> yang ada di <i>form</i> soal tugas akhir	Sistem akan menghilangkan isi yang ada di <i>field</i> soal tugas akhir	Berhasil
6	Menghapus soal tugas akhir pembelajaran <i>frontend development</i>	Menekan tombol hapus yang ada di card soal tugas akhir.	Sistem akan menghapus card atau soal tugas akhir pembelajaran <i>frontend development</i> .	Berhasil

4.2.4.3 Pengujian Halaman *Dashboard* Pelajar

Pengujian halaman *dashboard* pelajar adalah pengujian yang bertujuan untuk memastikan *user* pelajar bisa menggunakan sistem dengan nyaman, beberapa pengujian yang dilakukan meliputi penginputan jawaban soal tugas akhir, dan *download* sertifikat serta *download* materi pembelajaran dan *download* soal tugas akhir, berikut untuk lebih jelasnya dapat dilihat pada gambar 4.47.



Gambar 4.47 *Dashboard* Pelajar

Pada gambar diatas terdapat beberapa komponen seperti *form* masukkan tugas akhir dan hasil belajar anda, dimana pada halaman ini *user* pelajar bisa melakukan inputan terkait jawaban yang pelajar kerjakan, pada *form* masukkan tugas akhir pelajar diharuskan untuk mengisi seluruh *field* yang ada apabila terdapat *form* yang kosong, maka akan terdapat pesan warning yang menandakan pelajar belum melengkapi *form*-nya, berikut untuk lebih jelasnya dapat dilihat pada gambar 4.48.

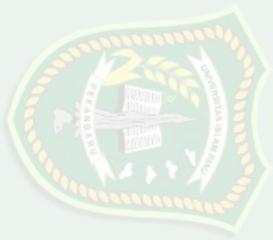


Gambar 4.48 Pesan Warning Field Link Github



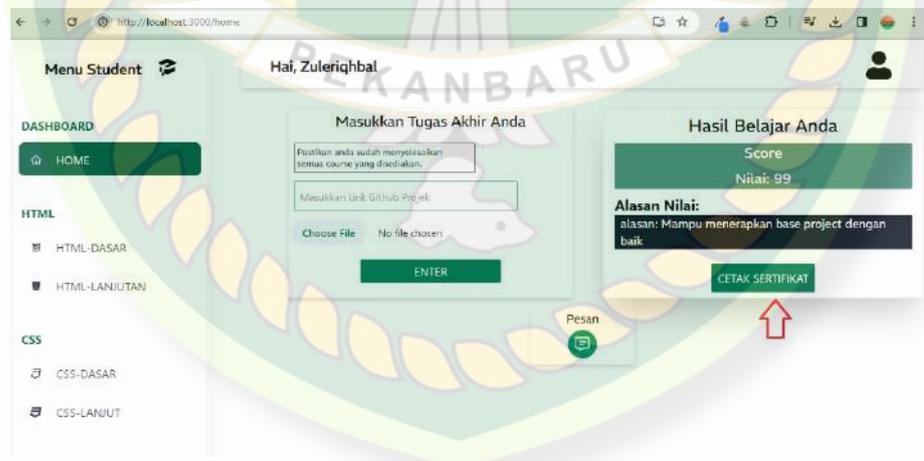
Gambar 4.49 Pesan Warning Field File

kedua gambar diatas adalah contoh apabila *user* pelajar mengosongkan *field* yang ada di *form* "masukkan tugas akhir anda",



pada gambar 4.48 adalah pesan error yang menunjukkan bahwa error tersebut disebabkan *user* pelajar belum mengisi semua *field*nya, namun jika *user* pelajar sudah mengisi link base projectnya tetapi tidak mengisi file nya maka akan terdapat pesan warning yang menandakan *user* pelajar tersebut belum mengisi file dari jawaban tugas akhirnya.

Pada halaman *dashboard* pelajar tidak hanya menguji terkait *form* pengisian jawaban tugas akhir tetapi apakah *user* pelajar bisa melakukan *download* sertifikat atau tidak, apabila *user* pelajar memiliki nilai maka bisa melakukan *download*-an, jika *user* pelajar belum memiliki nilai maka tidak dapat melakukan *download* untuk sertifikat, berikut pada gambar 4.50 hingga 4.51.

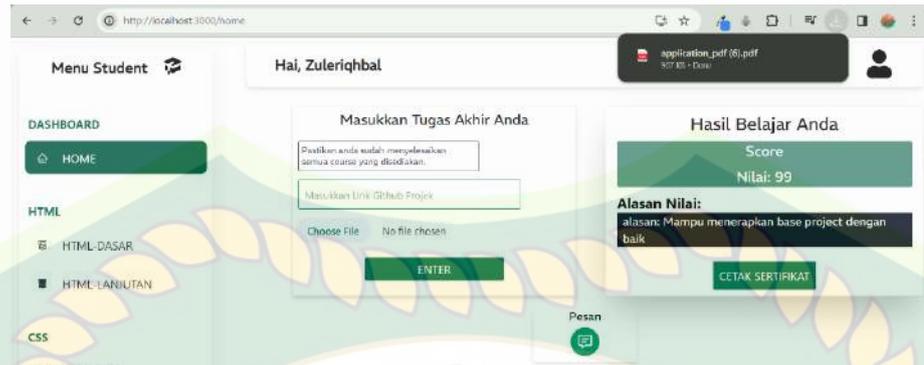


Gambar 4.50 Tombol Cetak Sertifikat

UNIVERSITAS
ISLAM RIAU



DOKUMEN INI ADALAH ARSIP MILIK :
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU

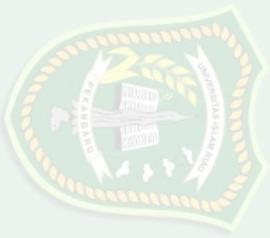


Gambar 4.51 Berhasil Download Sertifikat

Pada gambar 4.50 diatas yang ditandakan dengan panah merah adalah tombol untuk mendownload sertifikat, apabila berhasil maka tampilannya ada di gambar 4.51, namun jika gagal maka tampilannya tidak akan merespon apapun.

Tabel 4.6 Pengujian Form Tugas Akhir

No	Komponen Yang diuji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
1	Form Mengisi Jawaban Tugas Akhir	User pelajar tidak mengisi seluruh <i>field</i> yang ada <i>form</i> “masukkan tugas akhir anda”.	Sistem akan menolak dan memberi pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil
		User pelajar tidak mengisi salah satu <i>field</i> yang ada di <i>form</i> “masukkan tugas akhir anda”.	Sistem akan menolak dan memberi pesan peringatan “ <i>please fill out this field</i> ”.	Berhasil
		User pelajar mengisi semua	Sistem akan merespon dan akan	Berhasil



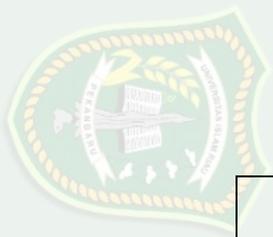
		<i>field</i> yang ada di <i>form</i> “masukkan tugas akhir anda”.	menghilangkan seluruh isi <i>field</i> yang ada di <i>form</i> “masukkan tugas akhir anda”.	
2	<i>Download</i> sertifikat hasil mengerjakan <i>task</i> tugas akhir.	<i>User</i> pelajar yang memiliki nilai dibawah atau sama dengan 75.	Tidak dapat melakukan <i>download</i> sertifikat.	Berhasil
		<i>User</i> pelajar yang memiliki nilai diatas 75	Dapat melakukan <i>download</i> sertifikat.	Berhasil

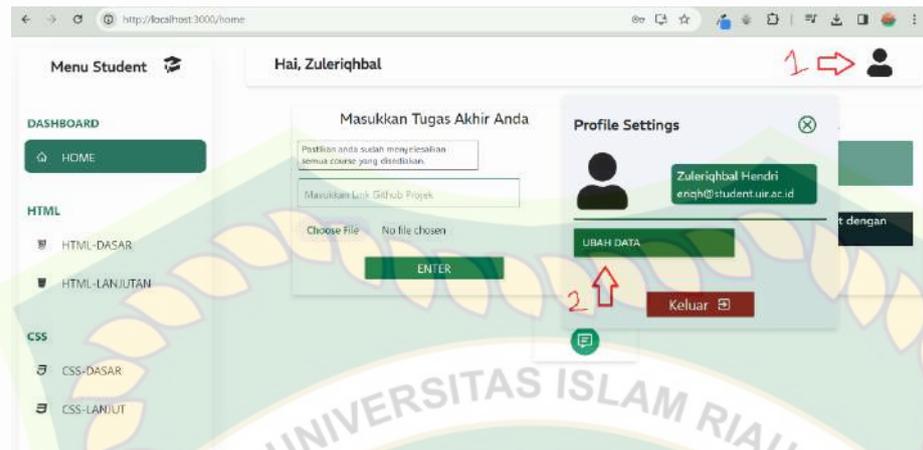
Dari hasil pengujian yang sudah dilakukan pada halaman *dashboard* maka dapat disimpulkan setiap aspek dan fungsi yang diharapkan telah berhasil.

4.2.5 Pengujian *Edit Data User*

Pengujian kali ini adalah melakukan sebuah *edit data user* dimana setiap *user* baik pelajar dan pengajar bisa melakukan perubahan *data* yang mereka miliki. Hal ini bertujuan agar sewaktu-waktu apabila *user* mendapati kesalahan dalam melakukan registrasi atau membuat akun ada *data* yang salah. Pengujian kali ini akan dilakukan oleh setiap *user*, berikut gambar 4.52 adalah detail terkait tombol ubah *data*.

**UNIVERSITAS
ISLAM RIAU**





Gambar 4.52 Tombol *Edit Data User* Pelajar

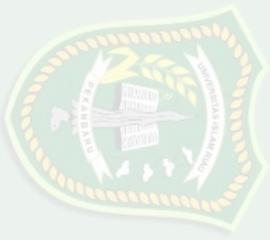
Pada gambar diatas adalah langkah-langkah untuk mengarah ke halaman ubah *data*, jika *user* menekan tombol *icon* profile pada nomor 1 maka akan muncul sebuah pop up *Profile settings* kemudian apabila menemukan tombol ubah *data*, maka tampilannya akan berganti seperti pada gambar 4.53.

4.2.5.1 *Edit Data User* Pelajar



Gambar 4.53 Tampilan *Ubah Data User*

Terlihat pada gambar 4.53 diatas adalah halaman untuk mengubah *data user* pelajar, hal ini memungkinkan setiap pelajar bisa merubah *datanya* apabila mendapati kesalahan, pada gambar diatas juga terdapat

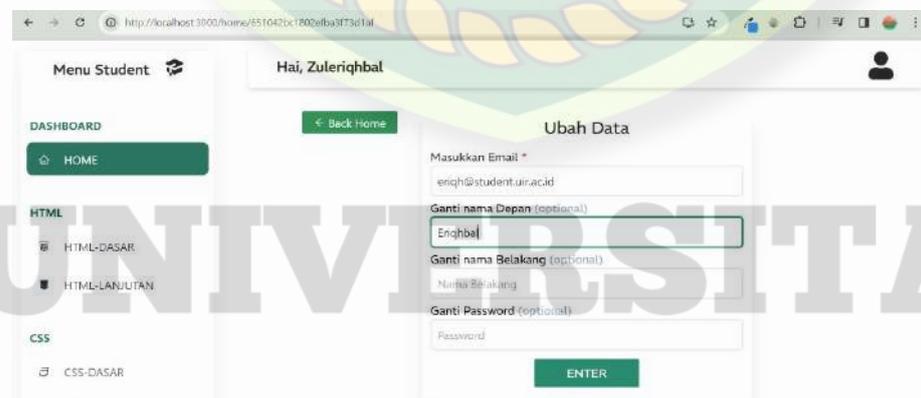


beberapa *field* yang harus diisi, terutama pada *field email* karena sifatnya yang required karena apabila *user* tidak mengisi *field email* maka proses perubahan *data* tidak dapat ditindaklanjuti oleh sistem, seperti pada gambar 4.54 adalah contoh tampilan apabila mendapati pesan warning.



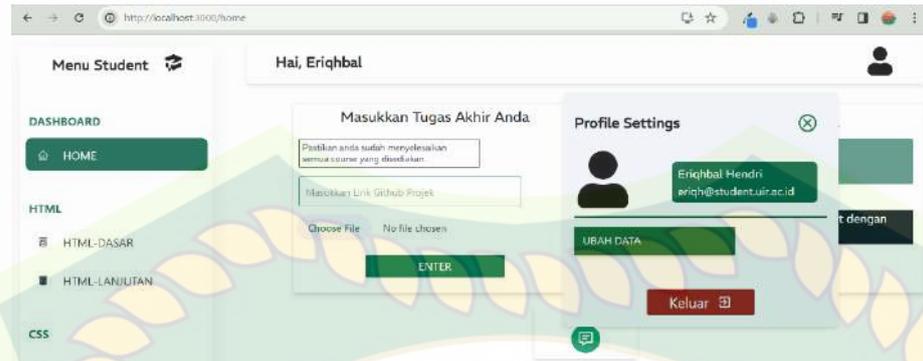
Gambar 4.54 Pesan Warning Edit Data User

Pada gambar 4.54 diatas adalah contoh tampilan warning message yang disebabkan *user* pelajar tidak mengisi *field email*, namun apabila *user* mengisi *field email* maka perubahan *ubah data* dapat dilakukan, untuk pengisian *field* selain *email* boleh kosong karena sifatnya yang optional maka tidak menjadi masalah apabila *field* tersebut kosong.



Gambar 4.55 Mengisi Semua Field





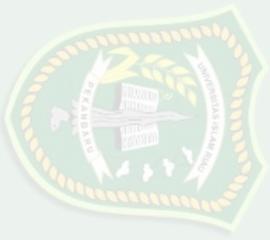
Gambar 4.56 Sukses Mengubah Data

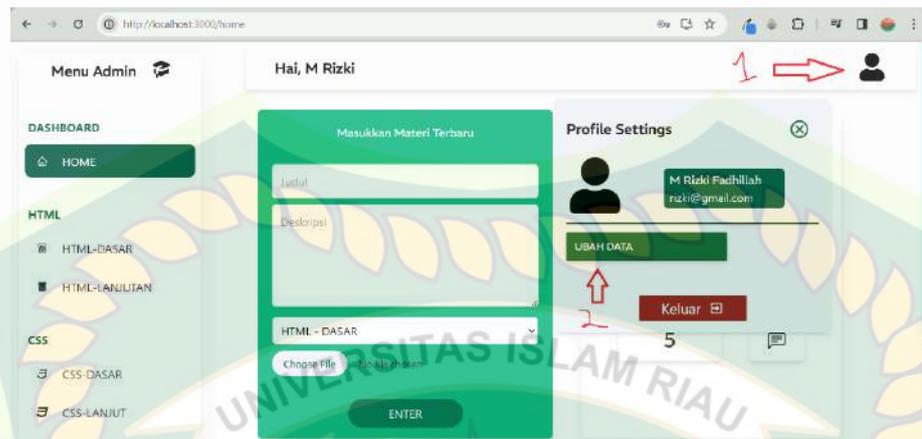
Pada gambar 4.55 diatas *user* pelajar telah mengisi *field email* dan ingin merubah *data* nama depan, setelah itu *user* dapat menekan tombol *enter*. *User* pelajar bisa memastikan apabila berhasil mengubah *data* maka sistem akan kembali ke menu *dashboard* dan dapat dilihat pada gambar 4.56 terdapat nama yang sudah berbeda di “Hai, Eriqbal” dan di *profile settings*.

4.2.5.2 Edit Data User Pengajar

Pengujian *edit data user* pengajar bertujuan untuk memastikan fungsi ubah *data* di *user* ini bisa berjalan dengan semestinya, untuk dapat melihat bagaimana tampilan dan contohnya dapat dilihat pada gambar 4.57.

**UNIVERSITAS
ISLAM RIAU**





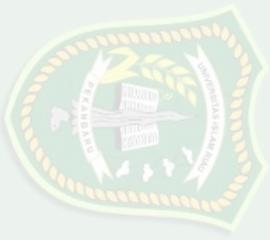
Gambar 4.57 Tampilan Ubah Data Pengajar

Pada gambar 4.57 diatas adalah tampilan untuk mengarah ke sebuah halaman ubah *data* pengajar. Hal pertama yang harus dilakukan adalah *user* pengajar menekan tombol nomor 1 yang ada *icon* profile, kemudian langkah berikutnya apabila sudah tampil sebuah pop up profile settings diatas maka tekan tombol ubah *data* nantinya sistem akan beralih halaman ke ubah *data* seperti pada gambar 4.58.



Gambar 4.58 Tampilan Ubah Data Pengajar

Gambar diatas adalah sebuah halaman ubah *data* untuk pengajar, untuk tampilan kurang lebih hampir serupa dengan

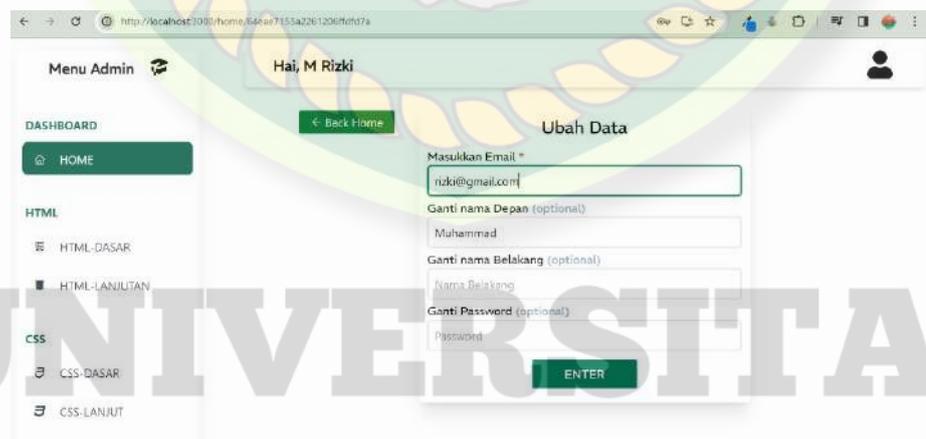


tampilan ubah *data* pelajar, dimana terdapat *field* email yang sifatnya required sedangkan lainnya optional yang artinya *field* tersebut boleh diisi atau tidak. Apabila *user* pengajar tidak mengisi *field* email maka tampilannya akan seperti gambar 4.59.



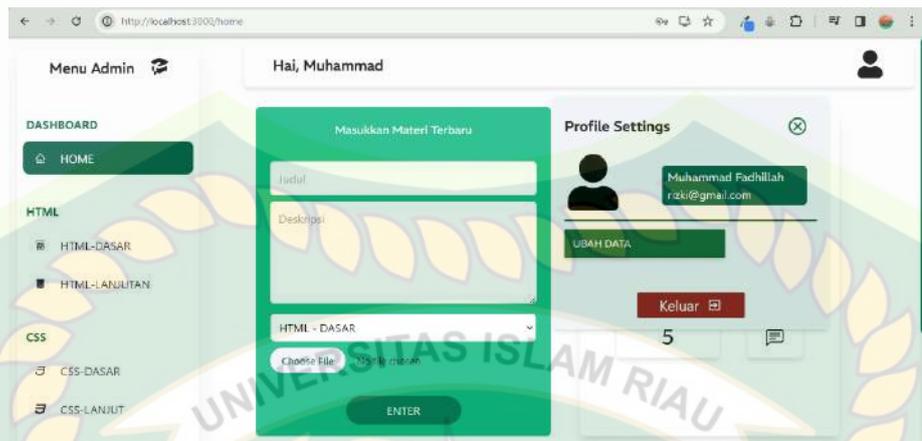
Gambar 4.59 Pesan Warning Field

Gambar diatas adalah contoh apabila *user* pengajar tidak mengisi *field* email untuk merubah *data*-nya, namun berbeda tampilan apabila *user* pengajar mengisi *field* email yang ada di form ubah *data* seperti pada gambar 4.60 hingga 4.61.



Gambar 4.60 Field Terisi Ubah Data





Gambar 4.61 Tampilan Berhasil Ubah Data

Pada gambar 4.60 *user* pengajar mengisi *field* email dan ingin merubah nama depan menjadi muhammad yang sebelumnya hanya m, lalu jika *user* pengajar menekan tombol enter maka sistem akan kembali ke *dashboard*, seperti pada gambar 4.61 dimana pada *dashboard* sistem sudah berhasil mengubah nama pengajar menjadi muhammad fadhillah. Untuk memahami keseluruhan dari fungsi *edit data user*, kita dapat melihat secara rinci melalui tabel dibawah.

Tabel 4.7 Rincian Pengujian *Edit Data User*

No	Komponen Yang Diuji	Skenario Pengujian	Hasil Yang Diharapkan	Hasil
1	<i>Edit data User</i> Pelajar	<i>Field</i> email tidak terisi saat ingin melakukan perubahan <i>data</i> .	Sistem akan menolak perubahan <i>data</i> dan akan merespon dengan memberikan pesan warning “ <i>please fill out this field</i> ”.	Berhasil
		<i>Field</i> email terisi saat ingin	Sistem akan memproses <i>data user</i>	Berhasil



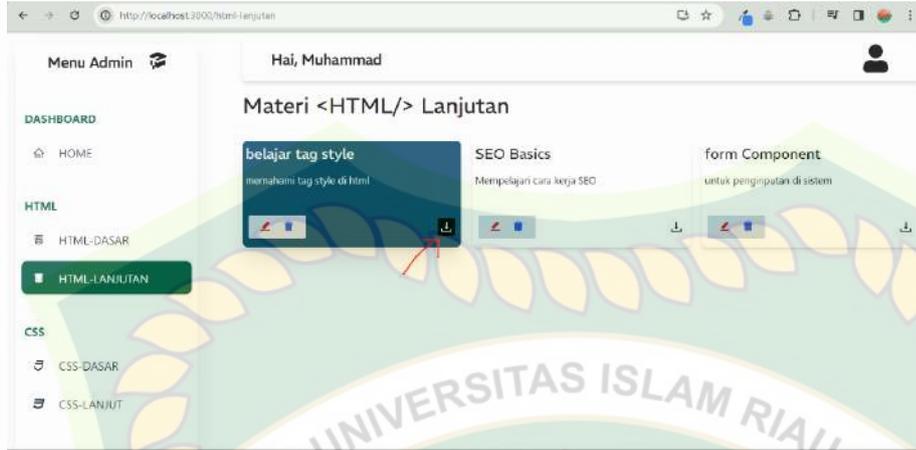
		melakukan perubahan <i>data</i> .	dan langsung memperbaiki <i>data user</i> pelajar tersebut.	
2	Edit data user Pengajar	<i>Field</i> email pengajar tidak terisi saat ingin melakukan perubahan <i>data</i> .	Sistem akan menolak perubahan <i>data</i> dan akan merespon dengan memberikan pesan warning “ <i>please fill out this field</i> ”.	Berhasil
		<i>Field</i> email terisi saat ingin melakukan perubahan <i>data</i> .	Sistem akan memproses <i>data user</i> dan langsung memperbaiki <i>data user</i> pengajar tersebut	Berhasil

Dari hasil pengujian yang sudah dilakukan pada halaman *edit data user* maka dapat disimpulkan setiap aspek dan fungsi yang diharapkan telah berhasil.

4.2.6 Pengujian Download Materi

Pengujian *download* materi adalah pengujian yang bertujuan untuk memastikan fungsi *download* sebuah file yang telah diinputkan oleh pengajar berhasil atau tidak, untuk itu dalam melakukan *download* sangat diutamakan *user* pelajar berhasil melakukannya. Untuk lebih jelasnya terkait pengujian *download* materi bisa dilihat pada gambar 4.62 hingga 4.63.

UNIVERSITAS
ISLAM RIAU



Gambar 4.62 Tombol *Download* Materi



Gambar 4.63 Hasil Sukses *Download*

Terlihat pada gambar 4.63 diatas dimana panah merah menunjukkan sebuah tombol untuk mendownload materi pembelajara, apabila *user* pelajar menekan tombol tersebut maka secara otomatis nantinya materi akan tersimpan di *local computer* masing-masing *user*, nantinya hasil setelah menekan tombol tersebut seperti pada gambar 4.63 yang artinya *download* materi pembelajaran telah berhasil dilakukan.

Tabel 4.8 Pengujian *Download* Materi

No	Komponen Yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil

1	<i>Download Materi</i>	<i>User</i> menekan tombol yang ada <i>icon</i> panah bawah.	Sistem secara otomatis akan men- <i>download</i> materi dan tersimpan di <i>local computer</i> masing-masing <i>user</i> .	Berhasil
---	------------------------	--------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------	----------

Dari hasil pengujian yang sudah dilakukan pada halaman *Download materi* maka dapat disimpulkan setiap aspek dan fungsi yang diharapkan telah berhasil.

4.3 Hasil Pengujian *Blackbox*

Berdasarkan dari hasil uji sistem *web* menggunakan metode *blacbox* bisa disimpulkan bahwa data yang dimasukkan harus sesuai dengan *tipe data* yang diminta oleh sistem, apabila data yang masuk tidak sesuai dengan *tipe data* yang diinginkan oleh sistem maka secara otomatis sistem akan memberikan pesan warning yang menandakan sistem menolak melanjutkan pemrosesan suatu action yang dilakukan *user*, tapi jika data yang masuk sesuai dengan *tipe data* yang ada di sistem maka sistem akan memproses dan berjalan dengan baik sesuai perencanaan diawal.

4.4 Pengujian Sistem Terhadap Pengguna

Pengujian sistem telah dilakukan dengan beberapa orang mahasiswa sebagai bentuk keyakinan apakah sistem yang dibuat sudah UAT (*User Acceptance Testing*) atau belum. Salah satu untuk mendapatkan data apakah

user menyukai sistem yang sudah dikembangkan adalah dengan cara memberikan beberapa pertanyaan yang bersifat kuesioner, jumlah dari pelajar yang diberikan kuesioner ada 10 orang dimana 10 orang ini akan mengisi jawaban masing-masing dari kuesionernya.

Untuk pertanyaan kuesioner ada berjumlah 7 dan masing-masing pertanyaan bisa dilihat dibawah ini:

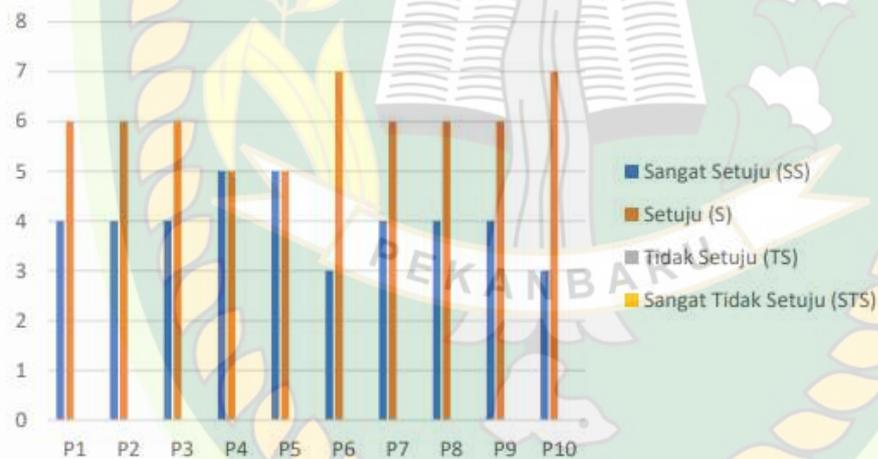
1. Sistem web *E-Learning* pembelajaran *frontend development* ini apakah sangat mudah digunakan?
2. Apakah sistem web-*E-Learning* ini memudahkan mahasiswa dalam mencapai tujuannya menjadi seorang *frontend development*?
3. Apakah sistem *E-Learning* ini sangat sesuai dengan tujuan utamanya sebagai sistem yang mendukung pelajar dalam mempelajari ilmu dibidang *frontend development*?
4. Apakah informasi pembelajaran yang ada di sistem ini sangat mudah dipahami?
5. Apakah menu pembelajaran *frontend development* yang ada di sistem sudah lengkap?
6. Apakah user interface dan *user experience* yang ada di sistem ini sudah baik?
7. Apakah sistem ini layak dipublikasikan?
8. Apakah sistem *E-Learning* yang dibangun sangat efektif dalam mendukung pembelajaran *frontend development*?
9. Apakah fitur *chat* di sistem ini sangat berguna?
10. Apakah fitur *chat* di sistem ini berfungsi dengan baik?



Responden dari pertanyaan diatas ada 4 jawaban yakni:

1. Sangat Setuju (SS)
2. Setuju (S)
3. Tidak Setuju (TS)
4. Sangat Tidak Setuju (STS)

Pada Pengujian sistem *E-Learning* ini, didapatkan responden yang berjumlah 10 orang yang sudah ditentukan diawal, maka hasil dari jawaban koresponden terhadap kinerja pada sistem berdasarkan pernyataan-pernyataan yang diajukan adalah sebagai berikut:

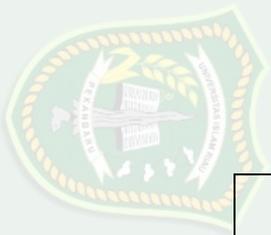


Gambar 4.64 Grafik Kuesioner Koresponden

Gambar 4.64 menunjukkan bahwa grafik diatas adalah hasil dari responden yang diambil melalui kuesioner dengan 10 orang pelajar dengan persentase rincian sebagai berikut:

No	Pernyataan	Jumlah Responden	Total

			Sangat Setuju	Setuju	Tidak Setuju	Sangat Tidak Setuju	
1	P1	Sistem <i>Web E-Learning</i> pembelajaran <i>frontend development</i> ini apakah sangat mudah digunakan	4 (40%)	6 (60%)	0 (0%)	0 (0%)	10 (100%)
2	P2	Apakah sistem <i>web E-Learning</i> ini memudahkan mahasiswa dalam mencapai tujuannya menjadi seorang <i>frontend development</i>	4 (40%)	6 (60%)	0 (0%)	0 (0%)	10 (100%)
3	P3	Apakah sistem <i>E-Learning</i> ini sangat sesuai dengan tujuan utamanya sebagai sistem yang mendukung pelajar dalam mempelajari ilmu	4 (40%)	6 (60%)	0 (0%)	0 (0%)	10 (100%)



		dibidang <i>frontend development</i>					
4	P4	Apakah informasi pembelajaran yang ada di sistem ini sangat mudah dipahami	5 (50%)	5 (50%)	0 (0%)	0 (0%)	10 (100%)
5	P5	Apakah menu pembelajaran <i>frontend development</i> yang ada di sistem sudah lengkap	5 (50%)	5 (50%)	0 (0%)	0 (0%)	10 (100%)
6	P6	Apakah user interface dan <i>user experience</i> yang ada di sistem ini sudah baik	3 (30%)	7 (70%)	0 (0%)	0 (0%)	10 (100%)
7	P7	Apakah sistem ini layak dipublikasikan	4 (40%)	6 (60%)	0 (0%)	0 (0%)	10 (100%)
8	P8	Apakah sistem <i>E-Learning</i> yang dibangun sangat efektif dalam mendukung pembelajaran	4 (40%)	6 (60%)	0 (0%)	0 (0%)	10 (100%)

		<i>frontend development</i>					
9	P9	Apakah fitur <i>chat</i> di sistem ini sangat berguna	4 (40%)	6 (60%)	0 (0%)	0 (0%)	10 (100%)
10	P10	Apakah fitur <i>chat</i> di sistem ini berfungsi dengan baik	3 (30%)	7 (70%)	0 (0%)	0 (0%)	10 (100%)

Berdasarkan dari hasil jawaban kuesioner yang didapat rata-rata jawaban yang diperoleh sangat positif, sehingga dapat ditarik kesimpulan bahwa score akan diolah dengan memberikan label penilaian:

1. Sangat Setuju dengan nilai 4;
2. Setuju dengan nilai 3;
3. Tidak Setuju dengan nilai 2;
4. Sangat Tidak Setuju dengan nilai 0;

Berikut proses pengolahan data kuesioner menggunakan metode skala *likert*:

Rumus:

$$T \times P_n$$

Keterangan:

T = Total jumlah responden yang memilih

P_n = Angka *score* pilihan *Likert*

1. Sistem web *E-Learning* pembelajaran *frontend development* ini apakah sangat mudah digunakan.

Jawaban SS : (4 Responden) x 4 = 16.

Jawaban S : (6 Responden) x 3 = 18.

Jawaban TS : (0 Responden) x 2 = 0.

Jawabang STS : (0 Responden) x 1 = 0.

2. Apakah sistem web-*E-Learning* ini memudahkan mahasiswa dalam mencapai tujuannya menjadi seorang *frontend development*?

Jawaban SS : (4 Responden) x 4 = 16.

Jawaban S : (6 Responden) x 3 = 18.

Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

3. Apakah sistem *E-Learning* ini sangat sesuai dengan tujuan utamanya sebagai sistem yang mendukung pelajar dalam mempelajari ilmu dibidang *frontend development*?

Jawaban SS : (4 Responden) x 4 = 16.

Jawaban S : (6 Responden) x 3 = 18.

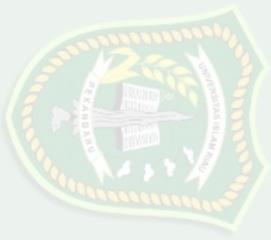
Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

4. Apakah informasi pembelajaran yang ada di sistem ini sangat mudah dipahami?

Jawaban SS : (5 Responden) x 4 = 20.

Jawaban S : (5 Responden) x 3 = 15.



Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

5. Apakah menu pembelajaran *frontend development* yang ada di sistem sudah lengkap?

Jawaban SS : (5 Responden) x 4 = 20.

Jawaban S : (5 Responden) x 3 = 15.

Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

6. Apakah user interface dan *user experience* yang ada di sistem ini sudah baik?

Jawaban SS : (3 Responden) x 4 = 12.

Jawaban S : (7 Responden) x 3 = 21.

Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

7. Apakah sistem ini layak dipublikasikan?

Jawaban SS : (4 Responden) x 4 = 16.

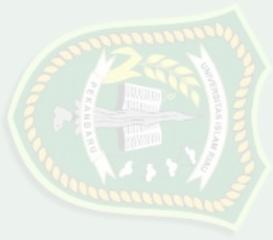
Jawaban S : (6 Responden) x 3 = 18.

Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

8. Apakah sistem *E-Learning* yang dibangun sangat efektif dalam mendukung pembelajaran *frontend development*?

Jawaban SS : (4 Responden) x 4 = 16.



Jawaban S : (6 Responden) x 3 = 18.

Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

9. Apakah fitur *chat* di sistem ini sangat berguna?

Jawaban SS : (4 Responden) x 4 = 16.

Jawaban S : (6 Responden) x 3 = 18.

Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

10. Apakah fitur *chat* di sistem ini berfungsi dengan baik?

Jawaban SS : (3 Responden) x 4 = 12.

Jawaban S : (7 Responden) x 3 = 21.

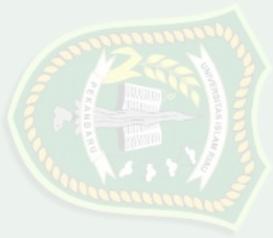
Jawaban TS : (0 Responden) x 2 = 0.

Jawaban STS : (0 Responden) x 1 = 0.

Dari perhitungan diatas, maka hasil yang didapatkan adalah:

- a. Total skor = **340**.
- b. Kemungkinan nilai tertinggi yang diraih adalah 4 (nilai jawaban tertinggi) x 10 (jumlah responden) x 10 (jumlah pertanyaan) = **400**.
- c. Kemungkinan nilai terendah = 1 (nilai jawaban tertinggi) x 10 (jumlah responden) x 10 (jumlah pertanyaan) = **100**.

Rumus Interval:



$$\begin{aligned}
 I. &= 100 / \text{jumlah score (Likert)} \\
 &= 100 / 4 \\
 &= 25
 \end{aligned}$$

Maka berikut interpretasi skornya berdasarkan *Interval* yang diperoleh:

Angka 0% - 24,99% = Sangat Tidak Setuju (STS)

Angka 25% - 49.55% = Tidak Setuju (TS)

Angka 50% - 74.55% = Setuju (S)

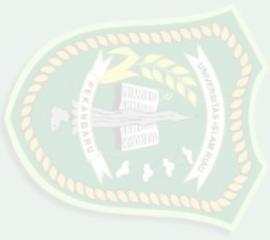
Angka 75% - 100% = Sangat Setuju (SS)

Rumus *index* (%):

$$\begin{aligned}
 \text{Skor Akhir} &= \text{Total skor} / \text{total skor tertinggi} \times 100 \\
 &= 340/400 \times 100 \\
 &= 85\% \text{ (Sangat Setuju)}
 \end{aligned}$$

Dari hasil perhitungan kuesioner diatas berada di angka **index 85%** yang artinya sistem *E-Learning* pembelajaran *frontend development* sangat baik dan sangat dibutuhkan dalam dunia pembelajaran khususnya dibidang *frontend development*.

**UNIVERSITAS
ISLAM RIAU**



DOKUMEN INI ADALAH ARSIP MILIK:
PERPUSTAKAAN SOEMAN HS
UNIVERSITAS ISLAM RIAU

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Setelah melakukan penelitian serta sudah menyelesaikan pengembangan dan pengujian pada sistem *E-Learning* guna mendukung proses pembelajaran *frontend development*, maka dapat ditarik kesimpulan sebagai berikut:

1. Telah berhasil mengembangkan sistem *E-Learning* guna mendukung pembelajaran sebagai seorang *frontend development* dengan mengujinya di beberapa mahasiswa universitas islam riau.
2. Dengan adanya sistem E-Learning yang dibuat khusus untuk meningkatkan skill khususnya dibidang *frontend development* para pelajar dapat dengan mudah memahami pembelajaran dan meniti karir sebagai seorang *frontend development*.
3. Berdasarkan jawaban dari para koresponden, rata-rata data yang didapat memiliki jawaban yang baik dan dengan nilai persentase untuk kategori baik sebesar 80% sehingga aplikasi ini bisa diterapkan secara publik.

5.2 Saran

Penelitian yang dilakukan ini bisa menjadi pedoman bagi para pembaca yang mengalami permasalahan hampir serupa. Meskipun begitu penulis akan menyarankan kepada para penelitian-penelitian selanjutnya untuk bisa mengembangkan sistem ini di berbagai platform baik itu mobile ataupun desktop agar bisa digunakan tanpa batasan akses platform.



DAFTAR PUSTAKA

- Agustini, & Kurniawan, W. J. (2019). Sistem *E-Learning* Do'a dan iqro' dalam peningkatan proses pembelajaran pada TK Amal Ikhlas. *Jurnal Mahasiswa Aplikasi Teknologi Komputer dan Informasi*, *I*, 154-159.
- Arfian, I. T., & Kurniawan, T. A. (2021). Pengembangan Sistem B2B E-Commerce Berbasis Layanan dengan Menggunakan Teknologi MERN STACK (Studi Kasus: Recaka Parfum Surabaya). *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, *V*, 3764-3772.
- Arora, R., & Aggarwal, R. R. (2013). Modelling and Querying *Data* in MongoDB. *International Journal of Scientific & Engineering Research*, *IV*, 141-144.
- Darnis, R., & Handayani, Y. (2022). Perancangan *E-Learning* Berbasis Responsive Web Studi Kasus Di Prodi Teknik Informatika Universitas Selamat Sri Kendal. *Jurnal Teknik Informatika dan Desain Komunikasi Visual*, *I*, 94-100.
- Efendi, A. (2021). Pengembangan Media Pembelajaran Berbasis Web Pada Materi Pemrograman Web di Kota Batam. 13-19.
- Haryadi, R., & Kansaa, H. N. (2021). Pengaruh Media Pembelajaran *E-Learning* Terhadap Hasil Belajar Siswa. *At-Ta'lim: Jurnal Pendidikan*, *VII*, 68-73.
- Hidayat, H., Hartono, & Sukiman. (2017). Pengembangan Learning Management System (LMS) Untuk Bahasa Pemrograman PHP. *JURNAL ILMIAH CORE IT*, *V*, 21-29.



Hutagalung, J., Winata, H., & Jaya, H. (2019). Perancangan dan Implementasi *E-Learning* Berbasis Web Pada SMA Negeri 1 Siantar. *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD, II*, 62-68.

Ilyas, A. H. (2018). Penggunaan Model Pembelajaran *E-Learning* Dalam Meningkatkan Kualitas Pembelajaran. *Jurnal Warta Edisi : 56*.

Indrawan, I. P., & Nugraha, P. G. (2020). Rancangan dan Implementasi Sistem *E-Learning* Berbasis Web. *Jurnal Pedagogi dan Pembelajaran, III*, 367-374.

Latukolan, M. L., Arwan, A., & Ananta, M. T. (2019). Pengembangan Sistem Pemetaan Otomatis Entity Relationship Diagram ke Dalam *Database*. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, III*, 4058-4065.

Mallisza, D., Hadi, H. S., & Aulia, A. T. (2022). Implementasi Model Waterfall Dalam Perancangan Sistem Surat Perintah Perjalanan Dinas Berbasis Website dengan Metode SDLC. *Jurnal Teknik, Komputer, Agroteknologi dan Sains, I*, 24-35.

Panjaitan, J., & Pakpahan, A. F. (2021). Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase. *Jurnal Teknik Informatika dan Sistem Informasi, VII*, 20-34.

Sasmito, G. W. (2017). Penerapan Metode Waterfall Pada Desain Sistem Informasi Geografis Industri Kabupaten Tegal. *Jurnal Pengembangan IT (JPIT), II*, 6-12.

UNIVERSITAS
ISLAM RIAU



Sucipto, G. F., Purwantoro, & Soeharso, A. (2023). Pengembangan Aplikasi *E-Learning* Sukabaca Menggunakan Framework Express.js dan MongoDB. *Jurnal Pendidikan Tambusai*, VII, 18757-18766.

Supria, Enda, D., Yulia, I., Permata, A. A., Amirul, M. F., & Asyrofi, M. F. (2022). Perancangan Learning Management System (LMS) Untuk Pengembangan Karir Siswa Tingkat Menengah Atas. *Proceeding Applied Business and Engineering Conference*, 17-19.

Syuhada, F. A., & Handrianto, Y. (2023). Perancangan Aplikasi Learning Management System Berbasis Web Pada Trustco Cipta Madani. *Jurnal Komputer Antartika*, I, 158-166.

Triandy, R., & Santoso, N. (2020). Pengembangan Aplikasi Web Reservasi Paket Wisata Menggunakan MERN Stack. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, IV, 1616-1624.

Wassalam, O. J., Umar, R., & Yudhana, A. (2017). Implementasi Dan Pengembangan Sistem *E-Learning* Berbasis Web Pada STIMIK Muhammadiyah Paguyangan. *Prosiding Seminar Nasional Multi Disiplin Ilmu & Call For Papers Unisbank Ke-3*, 104-107.

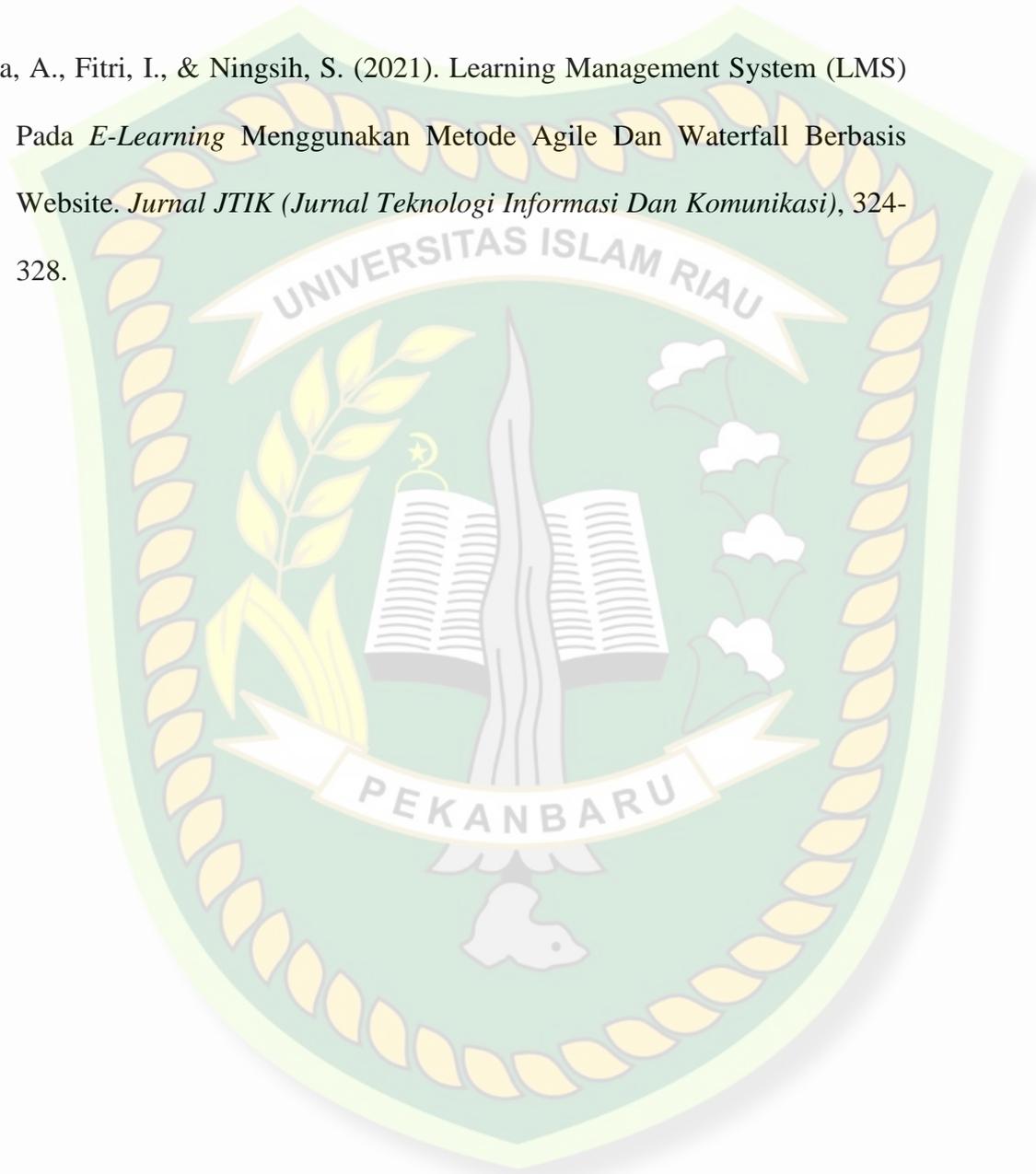
Wibowo, A. T., Akhlis, I., & Nugroho, S. E. (2014). Pengembangan LMS (Learning Management System) Berbasis WEB untuk Mengukur Pemahaman Konsep dan Karakter Siswa. *Scientific Journal of Informatics*, I, 127-137.

Wijaya, Y. D., & Astuti, M. W. (2021). Pengujian *Blacbox* Sistem Informasi Penilaian Kinerja Karyawan PT INKA (PERSERO) Berbasis



EQUIVALENCE PARTITIONS. *Jurnal Digital Teknologi Informasi*, IV, 22-26.

Yauma, A., Fitri, I., & Ningsih, S. (2021). Learning Management System (LMS) Pada *E-Learning* Menggunakan Metode Agile Dan Waterfall Berbasis Website. *Jurnal JTIC (Jurnal Teknologi Informasi Dan Komunikasi)*, 324-328.



**UNIVERSITAS
ISLAM RIAU**

DOKUMEN INI ADALAH ARSIP MILIK :

PERPUSTAKAAN SOEMAN HS

UNIVERSITAS ISLAM RIAU