# PERANCANGAN SISTEM INFORMASI LAYANAN LPPAIK (LEMBAGA PENGKAJIAN DAN PENERAPAN AL-ISLAM & KEMUHAMMADIYAHAN) UNIVERSITAS MUHAMMADIYAH KENDARI BERBASIS *WEB*



**MUHAMMAD ARSIL ALHABSY**

**NIM. 21916042**

**PROPOSAL PENELITIAN**

*Diajukan Sebagai Salah Satu Syarat Untuk Mengajukan Seminar Proposal Penelitian*

**PROGRAM STUDI PENDIDIKAN TEKNOLOGI INFORMASI**

**FAKULTAS KEGURUAN DAN ILMU PENDIDIKAN**

**UNIVERSITAS MUHAMMADIYAH KENDARI**

**KENDARI**

**2023**

# HALAMAN PERSETUJUAN PEMBIMBING

Telah diperikas dan disetujui oleh Pembimbing I dan Pembimbing II untuk dipertahankan dihadapan panitian Ujian Skripsi pada Program Studi Pendidikan Teknologi Informasi Fakultas Keguruan dan Ilmu Pendidikan Universitas Muhammadiyah Kendari.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nama | | : | Muhammad Arsil Alhabsy | | | | | | |
| Nim | | : | 21916042 | | | | | | |
| Judul | | : | Perancangan Sistem Informasi Layanan LPPAIK (Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan) Universitas Muhammadiyah Kendari Berbasis *Web* | | | | | | |
|  | |  |  | | | | | | |
| **Menyetujui** | | | | | | | | | |
| **Pembimbing I** | | | | |  | | **Pembimbing II** | | |
|  | | | | |  | |  | | |
| **Alfiah Fajriani, ST., M.Eng** | | | | | |  | **Ilcham, S.SI., M.Eng** | | |
| **NIDN. 0919029203** | | | | |  | | **NIDN.** **RD444514** | | |
| **Mengetahui** | | | | | | | | | |
|  | **Ketua Program Studi Pendidikan Teknologi Informasi**  **Fakultas Keguruan Dan Ilmu Pendidikan**  **Universitas Muhammadiyah Kendari** | | | | | | | |  |
|  | | | | |  | |  | | |
|  | | | | **Zila Razilu, S.Pd., M.Pd** | | | |  | |
|  | | | | | **NIDN. 0922059102** | |  | | |

# DAFTAR ISI

[Cover](#_Toc152524457)

[HALAMAN PERSETUJUAN PEMBIMBING i](#_Toc152524458)

[DAFTAR ISI ii](#_Toc152524459)

[DAFTAR GAMBAR iv](#_Toc152524460)

[DAFTAR TABEL v](#_Toc152524461)

[BAB I PENDAHULUAN 1](#_Toc152524462)

[A. Latar Belakang 1](#_Toc152524463)

[B. Rumusan Masalah 5](#_Toc152524464)

[C. Batasan Masalah 5](#_Toc152524465)

[D. Tujuan Penelitian 6](#_Toc152524466)

[E. Manfaat 6](#_Toc152524467)

[BAB II LANDASAN TEORI 8](#_Toc152524468)

[A. Perancangan 8](#_Toc152524469)

[B. Sistem Informasi 9](#_Toc152524470)

[1. Sistem 9](#_Toc152524471)

[2. Informasi 11](#_Toc152524472)

[3. Sistem Informasi 13](#_Toc152524473)

[C. LPPAIK 14](#_Toc152524474)

[D. Aplikasi Berbasis Web 14](#_Toc152524475)

[1. *Website* 14](#_Toc152524476)

[2. Aplikasi berbasis web 15](#_Toc152524477)

[E. *Web Service* 16](#_Toc152524478)

[F. RESTful API 17](#_Toc152524479)

[G. *Framework* 18](#_Toc152524480)

[1. *ReactJS* 18](#_Toc152524481)

[2. *Spring* dan *Spring Boot* 19](#_Toc152524482)

[H. Basis data 20](#_Toc152524483)

[1. *Database* 20](#_Toc152524484)

[2. *Database Management System* (DBMS) 22](#_Toc152524485)

[3. *MySQL* 22](#_Toc152524486)

[I. Bahasa Pemrogramman 23](#_Toc152524487)

[1. Java 23](#_Toc152524488)

[2. *JavaScript* 25](#_Toc152524489)

[3. *NodeJs* 26](#_Toc152524490)

[J. *Programming Tools* 27](#_Toc152524491)

[1. Intelij IDEA 27](#_Toc152524492)

[2. VSCode 27](#_Toc152524493)

[3. Postman 28](#_Toc152524494)

[K. *Software Development Life Cycle* (SDLC) 28](#_Toc152524495)

[L. *Unifide Model Language* (UML) 29](#_Toc152524496)

[M. Penelitian Relevan 30](#_Toc152524497)

[BAB III METODOLOGI PENELITIAN 33](#_Toc152524498)

[A. Jenis Penelitian 33](#_Toc152524499)

[B. Prosedur Penelitian 33](#_Toc152524500)

[1. Analisis (Analaysis) 34](#_Toc152524501)

[2. Desain (*Design*) 38](#_Toc152524502)

[3. Penulisan Kode (*Implementation*) 56](#_Toc152524503)

[4. Pengujian (*Testing*) 57](#_Toc152524504)

[5. Pemeliharan (*Maintenance*) 58](#_Toc152524505)

[C. Tempat dan Waktu Penelitian 58](#_Toc152524506)

[D. Alat dan Bahan Penelitian 59](#_Toc152524507)

[1. Alat penelitian 59](#_Toc152524508)

[2. Bahan Penelitian 60](#_Toc152524509)

[DAFTAR PUSTAKA 62](#_Toc152524510)

# DAFTAR GAMBAR

[Gambar 2. 1 Komunikasi Antara Web Browser dan Aplikasi Web 16](#_Toc152525042)

[Gambar 2. 2 Model Siklus Pengembangan Sistem 29](#_Toc152525043)

[Gambar 3. 1 Model Waterfall (Findawati, 2018) 33](#_Toc152525055)

[Gambar 3. 2 *Use Case* Diagram Mahasiswa 39](#_Toc152525056)

[Gambar 3. 3 *Use Case* Diagram Admin 39](#_Toc152525057)

[Gambar 3. 4 *Use Case* Tutor Sebaya 40](#_Toc152525058)

[Gambar 3. 5 *Use Case* Dosen 40](#_Toc152525059)

[Gambar 3. 6 *Activity* Diagram *Login Users* 42](#_Toc152525060)

[Gambar 3. 7 *Activity* Diagram Kegiatan Kajian 43](#_Toc152525061)

[Gambar 3. 8 *Activity* Diagram Module BTQ 44](#_Toc152525062)

[Gambar 3. 9 ERD Sistem Infromasi Layanan LPPAIK 47](#_Toc152525063)

[Gambar 3. 10 *User* *Interface* Tampilan *Login* 48](#_Toc152525064)

[Gambar 3. 11 *User Interface* Tamplian *Landing* 49](#_Toc152525065)

[Gambar 3. 12 User Interface Tampilan Activities 50](#_Toc152525066)

[Gambar 3. 13 *User Interface* Tampilan *Activity Detail* 51](#_Toc152525067)

[Gambar 3. 14 *User Interface* Tampilan *Certificate* 52](#_Toc152525068)

[Gambar 3. 15 *User Interface* Tampilan *Users* 53](#_Toc152525069)

[Gambar 3. 16 *User Interface* Tampilan *User Detail* 54](#_Toc152525070)

[Gambar 3. 17 *User Interface* Tampilan BTQ 54](#_Toc152525071)

[Gambar 3. 18 *User Interface* Tampilan *User Activities* 55](#_Toc152525072)

# DAFTAR TABEL

[Tabel 3. 1 Evaluasi Sistem 35](#_Toc152525788)

[Tabel 3. 2 Tabel users 45](#_Toc152525789)

[Tabel 3. 3 Tabel activities 45](#_Toc152525790)

[Tabel 3. 4 Tabel user\_activities 45](#_Toc152525791)

[Tabel 3. 5 Tabel majors 45](#_Toc152525792)

[Tabel 3. 6 Tabel certificates 46](#_Toc152525793)

[Tabel 3. 7 Tabel roles 46](#_Toc152525794)

[Tabel 3. 8 Tabel user\_roles 46](#_Toc152525795)

[Tabel 3. 9 Tabel images 46](#_Toc152525796)

[Tabel 3. 10 Tabel activity\_images 46](#_Toc152525797)

[Tabel 3. 11 Tabel activity\_registers 46](#_Toc152525798)

[Tabel 3. 12 Tabel control\_book\_details 47](#_Toc152525799)

[Tabel 3. 13 Tabel Kisi Kisi Uji Black Box Testing 56](#_Toc152525800)

[Tabel 3. 14 Tabel Rencana Kerja Sistem 57](#_Toc152525801)

# PENDAHULUAN

## Latar Belakang

Teknologi informasi akhir-akhir ini sangat pesat sekali perkembangannya jika dibandingkan dengan teknologi lainnya. Manusia telah banyak merasakan segala macam kemudahan dalam menjalani hidup dengan bantuan dari teknologi dan sistem informasi, khususnya dalam menyelenggarakan kegiatan bisnis dan pendidikan. Teknologi informasi merupakan penggabungan antara teknologi komputerisasi serta interaksi yang akan membentuk sebuah sistem perangkat lunak (*software*) serta perangkat keras (*hardware*)(Putri Primawanti & Ali, 2022). Sedangkan Sistem Informasi merupakan suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolah sebuah transaksi dengan mendukung operasi yang bersifat manajerial dan kegiatan strategi dari suatu organisasi untuk menyediakan sebuah *output* kepada pihak luar tertentu dengan berbentuk laporan-laporan yang dibutuhkan(Hutahaean, 2015).

Penggunaan sistem informasi berbasis *website* telah banyak dikembangkan di zaman sekarang, dengan menyediakan kemudahan dan kelebihan-kelebihan yang dimiliki-nya, seperti yang dikemukakan oleh(Sujiwa & Rochman, 2019) cukup dengan *HTML* versi 5 *web browser* sudah bisa diakses di seluruh perangkat diantaranya *desktop*, laptop maupun *smart phone.* Internet sangat dibutuhkan dalam kemajuan teknologi, melalui internet kita dapat melakukan interaksi, bertukar ide, mengetahui analisa produk dan pasar, dan memperoleh informasi dari sumber yang berbeda yang tersebar di seluruh dunia. Penggabungan internet dan *website* dapat membantu meningkatkan efisiensi dengan meningkatkan kredibilitas dan memberikan efek positif dari segi keuangan untuk menghemat pengeluaran dari sisi finansial(Maharani dkk., 2021).

Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan (LPPAIK) merupakan salah satu lembaga di Universitas Muhammadiyah Kendari yang belum mempunyai platform atau wadah khusus untuk menyampaikan informasi kegiatan - kegiatan atau program yang telah berjalan. Sebagai mana hasil wawancara penulis lakukan pada tanggal 03 April 2023 dengan ketua LPPAIK bahwahsanya lembaga ini belum mempunyai platform atau wadah khusus untuk menyampaikan informasi terhadap kegiatan maupun program yang sedang berlangsung. LPPAIK memiliki beberapa modul kegiatan diantaranya kajian untuk Dosen dan Tenaga Kependidikan (Tendik), kajian untuk Mahasiswa, Sertifikasi Baca Tulis Al - Qur’an (BTQ) dan lain sebagainya. Infromasi dari kegiatan – kegiatan dan program dari LPPAIK menjadi sangat penting untuk diketahui dikarenakan ada beberapa modul dari kegiatan yang diselengarakan menjadi syarat untuk menawar matakuliah lanjutan maupun menjadi syarat wajib dari kelulusan di UMK yaitu modul kajian untuk mahasiswa dan sertifikasi BTQ.

Berdasarkan hasil observasi penulis untuk mengikuti kegiatan kajian untuk mahasiswa, diperlukan absen untuk mencatat kehadiran dari mahasiswa, yang nantinya data dari absennya akan menjadi syarat untuk menawar matakuliah selanjutnya. Begitu pula dengan mengikuti kegiatan Sertifikasi BTQ, diperlukan satu buku kontrol untuk mengetahui sejauh mana capaian dari mahasiswa dalam mengikuti kegiatan BTQ. Melakukan penyimpanan data dalam bentuk catatan kertas menjadi tidak dapat terjamin kualitasya seperti yang dijelaskan (Rochman dkk., 2018) bahwa penyimpanan laporan dalam bentuk kertas tidak dapat terjamin keamannya baik ancaman fisik seperti kerusakan dalam penyimpanan hasil laporan maupun informasi penting bisa jadi tidak dapat diketahui oleh pihak yang berkepentingan untuk meperoleh informasi tersebut.

Berdasarkan permasalahan diatas penulis menawarkan suatu sistem layanan informasi untuk Lembaga Pengkajian dan Penerapan Al–Islam & Kemuhammadiyahan Universitas Muhammadiyah Kendari berbasis *web* dengan menggunakan *framework ReactJS* sebagai *Library* untuk membuat tampilan antarmuka berbasis *components* atau *JavaScript function* yang memungkinkan pengembang mengabungkan *components* secara *independent* dan *Spring Boot* sebagai *web service*.

Perancangan Sistem Informasi LPPAIK Universitas Muhammadiyah Berbasis *Web* ini diharapkan mampu mempermudah penyebaran infromasi dan penyimpanan data antar *stackholder* LPPAIK dan mahasiswa, mempermudah *stackholder* untuk mengontrol dan memantau perkembangan pemahaman AIK melalui kegiatan kajian untuk mahasiswa. Membantu mahasiswa dalam menyelesaikan adiminstrasi perkuliahan dengan LPPAIK dan program studi masing-masing. Dan juga menjamin data-data dari infromasi transaksi tetap berkulitas, akurat dan terjamin keabsahanya.

Berdasarkan uraian latar belakang diatas, maka penulis tertarik untuk melakukan penelitian dengan judul: “Perancangan Sistem Infromasi Layanan LPPAIK (Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan) Universitas Muhammadiyah Kendari Berbasis *Web*”.

## Rumusan Masalah

Berdasarkan penjelasan yang telah diuraikan dari latar belakang diatas, maka masalah yang akan diangkat dalam skripsi ini adalah “Bagaimana merancang Sistem Informasi Layanan LPPAIK Universitas Muhammadiyah Berbasis *Web*?”.

## Batasan Masalah

Adapun batasan masalah yang terdapat dalam penelitian ini adalah sebagai berikut:

1. *Website* yang akan dirancang adalah Sistem Informasi Layanan LPPAIK (Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan) Universitas Muhammadiyah Kendari Berbasis *Web*.
2. Menerapkan 4 diagram dari *Unified Modeling Language* (UML), yaitu *use case* diagram, *activity* diagram, *Entity Relational Diagram* (ERD) dan *mockup*, untuk memastikan kejelasan dan pemahaman yang baik terkait dengan perancangan *website*.
3. Perancangan *website* menggunakan *framework ReactJS* yang berbasis *JavaScript* sebagai tampilan antarmuka dan *framework Spring Boot* yang berbasis *Java* sebagai layanan *rest* *web service*.
4. Pengembangan *website* ini menggunakan *analysis*, *design,* *implementation,* *testing* dan *maintenance* melalui siklus *Software Development Life Cycle* (SDLC) dengan menggunakan model *waterfall*.
5. Cakupan modul yang akan dikembangkan adalah modul BTQ dan Kajian untuk mahasiswa.
6. Tidak secara khusus membahas aspek keamanan *website*.

## Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah merancang Sistem Informasi Layanan LPPAIK (Lembaga Pengkajian dan Penerapan Al-islam & Kemuhammadiyahan) Universitas Muhammadiyah Kendari Berbasis *Web* dengan memastikan fitur yang dibangun pada modul BTQ dan Kajian dapat berjalan dengan baik.

## Manfaat

Sesuai permasalahan dan tujuan penelitian yang telah disebutkan pada latar belakang beserta tujuan dia atas, maka manfaat yang diharapkan dapat diperoleh dari penelitian sistem ini adalah sebagai berikut:

1. Manfaat teoritis

Sebagai sumbangan ide bagi kepentingan ilmu pengetahuan, dan juga menambah pengetahuan khususnya dalam perancangan Sistem Informasi Layanan LPPAIK (Lembaga Pengkajian dan Penerapan Al-islam & Kemuhammadiyahan) Universitas Muhammadiyah Kendari Berbasis *Web*, dan juga dapat menjadi rujukan untuk penelitian berikutnya yang relevan dengan penelitian ini.

1. Manfaat praktisi
2. Bagi lembaga

Dapat mempermudah dalam melakukan pengontrolan kegiatan-kegiatan yang akan dilakukan dan mempermudah rekapitulasi informasi mengenai pencapaian-pencapaian yang diperoleh oleh mahasiswa.

1. Bagi mahasiswa
2. Dapat mengetahui informasi kegiatan-kegiatan yang diadakan oleh LPPAIK.
3. Mengetahui sejauh mana pencapaian mahasiswa itu sendiri dalam mengikuti kegiatan-kegiatan yang telah mereka ikuti.
4. Mempermudah mahasiswa dalam urusan memperoleh sertifikat kelulusan BTQ dan keterangan telah mengikuti kegiatan kajian.
5. Manfaat bagi peneliti

Untuk memeroleh gelar sarjana, menambah pengetahuan dan wawasan dalam perancangan suatu sistem informasi berbasis *web*.

# LANDASAN TEORI

## Perancangan

Perancangan merupakan tahapan awal untuk melakukan rancang bangun pengaplikasian suatu sistem yang menggambarkan bagaimana cara sebuah sistem itu dibentuk yang dapat berupa gambaran, rancangan dan pembuatan sektsa atau pengaturan dari beberapa elemen terpisah ke dalam suatu kesatuan yang lengkap(Rianto dkk., 2017).

Didalam (Fariyanto dkk., 2021) menjelaskan tentang perancangan bahwa perancangan merupakan proses merencanakan, mendeskripsikan, mensketsa atau menyusun beberapa komponen-komponen elemen yang independen menjadi satu kepaduan fungsional yang utuh. Menurut Soetam Rizky dikutip dari (Rianto dkk., 2017) “Perancangan merupakan sebuah proses untuk mendefinsisikan sesuatu yang akan dikerjakan dengan menggunkaan teknik yang bervariasi serta didalamnya melibatkan deskripsi mengenai arsitektur serta detail komponen dan juga keterbatasan yang akan dialami dalam proses pengerjaanya”. Menurut Jogiyanto dikutipan dari (Fariyanto dkk., 2021) “Perancangan mempunyai dua tujuan, diantaranya yaitu untuk memenuhi kebutuhan pengguna sistem, dan untuk memberikan gambaran yang jelas bagi pengembang komputer dan pakar teknis terkait lainnya”. Perancangan sistem dapat dirancang dalam format diagram alir sistem *(system flowchart)*, yaitu merupakan suatu alat grafis yang dapat digunakan untuk menampilkan proses urutan perancangan sebuah sistem.

## Sistem Informasi

### Sistem

Dalam (Anggraeni, 2017) menjelaskan tentang pengertian sistem, Sistem merupakan sekelompok unsur yang erat hubungannya dengan satu sama lain yang saling bekerja sama dengan ketentuan-ketentuan aturan yang sistematis dan terstruktur untuk membentuk suatu kesatuan yang melaksanakan suatu fungsi untuk mencapai tujuan. Beberapa pengertian sistem menurut para ahli dalam (Hutahaean, 2015) diantaranya sebagai berikut:

1. Menurut Fat “Sistem adalah suatu himpunan ‘benda’ nyata atau abstrak *(a set of things)* yang terdiri dari bagian-bagian atau komponen-komponen yang saling berkaitan, berhubungan, berketergantungan, saling mendukung, yang secara keseluruhan bersatu dalam suatu kesatuan *(Unity)* untuk mencapai tujuan tertentu secara efisien dan efektif”.
2. Menurut Jogianto “Sistem adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem ini menggambarkan suatu kejadian-kejadian dan kesatuan yang nyata adalah suatu objek nyata, seperti tempat, benda, orang-orang yang betul-betul ada dan terjadi”.
3. Menurut Jerry FutzGerald “Sistem adalah jaringan kerja dari prosedur – prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu”.

Secara keseluruhan sistem merupakan suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu aktifitas atau kegiatan untuk mencapai sasaran yang ingin dicapai secara bersama-sama dengan efektif dan efisien. Pendekatan sistem yang merupakan jaringan kerja dari prosedur yang lebih menekankan langkah-langkah urutan operasi di dalam suatu sistem(Hutahaean, 2015).

Terdapat beberapa karakteristik atau sifat dari suatu sistem yaitu, sistem memiliki komponen sistem, batasan sistem, lingkungan luar sistem, penghubung sistem, masukan sistem, keluaran sistem, pengelolahan sistem dan sasaran sistem (Anggraeni, 2017). Di dalam (Hutahaean, 2015) menjelaskan lebih rinci dari karakteristik dari sebuah sistem. Sistem yang dapat dikatakan sistem yang baik yaitu sistem yang memiliki karakter sebagai berikut:

1. Komponen merupakan sebuah jaringan sistem yang terdiri dari sejumlah komponen-komponen yang saling berinteraksi dengan memiliki subsistem atau bagian-bagian dari sebuah sistem dengan memiliki makna saling bekerja sama membentuk suatu integritas jaringan sistem.
2. Batasan Sistem *(boundry)* adalah ruang lingkup *(scope)* atau wilayah dari suatu sistem yang membatasi antara suatu sistem dengan sistem lainnya atau dengan wilayah luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu keutuhan sistem.
3. Lingkungan luar sistem *(environment)* merupakan luaran batasan dari sistem yang mempengaruhi sistem operasi dari suatu sistem. Lingkungan dari suatu sistem dapat bersifat menguntungkan dan merugikan perilaku yang menguntungkan harus dijaga dan perilaku yang merugikan harus dikendalikan untuk menjaga keberlangsungan hidup dari suatu sistem.
4. Penghubung sistem *(interface)* adalah media penghubung antara suatu subsistem dengan subsistem yang lainnya. Dengan adanya penghubung ini memungkinkan sumber-sumber daya dapat mengalir dari satu subsistem ke subsistem lainya.
5. Masukkan sistem *(input)* merupakan energi yang diberikan kedalam suatu sistem, yang dapat berupa perawatan *(maintenance input)*, dan masukkan sinyal *(signal input)*. *Maintenance* input merupakan energi yang dimasukan dengan tujuan sistem dapat beropresi dengan semestinya sedangkan *signal input* merupakan energi yang diproses untuk mendapatkan suatu keluaran *(output)*.
6. Keluaran sistem *(output)* merupakan hasil dari energi yang telah diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan atau tidak berguna. Sebagai contoh komputer melakukan pembuangan udara panas dari suatu proses yang dilakukan, disisi lain hasil dari suatu proses yang dilakukan komputer dapat berupa informasi sebagai keluaran yang butuhkan.

### Informasi

Informasi merupakan sekumpulan data atau fakta yang terorganisasi dan diolah dengan cara tertentu sehingga mempunyai makna bagi penerimanya. Data yang telah diolah menjadi sesuatu yang berguna bagi penerima dan dapat memberikan keterangan atau pengetahuan. Informasi dapat juga dikatakan sebagai sebuah pengetahuan yang diperoleh dari pengalaman, pembelajaran, atau instruksi yang menjadi sumber data dari suatu informasi (Anggraeni, 2017).

Sedangkan menurut Sutabri kutipan (Prehanto, 2020) Informasi merupakan pengolahan data yang dapat ditafsirkan maupun diklasifikasi yang dapat dipakai dalam proses untuk mengambil sebuah keputusan. Sumber dari suatu informasi berupa data yang menggambarkan sebuah kejadian secara nyata yang telah terjadi pada waktu tertentu. Sumber ini perlu dilakukan pengelolahan memalui siklus yang dinamakan siklus pengelolahan data *(data processing life cycle)* yang didalamnya terdapat *input* (data), model (pengolahan) dan *output* (informasi). Hasil penggolahan dari suatu sumber informasi dapat dikatakan berharga jika informasi tersebut berguna dalam proses pengambilan keputusan. Berikut adalah ciri-ciri dari sebuah informasi yang berkualitas menurut Raymod Mc. Leod kutipan (Anggraeni, 2017) diantaranya adalah:

1. Informasi harus mengambarkan keadaan yang sebenarnya dan juga informasi harus bebas dari kesalahan-kesalahan, dengan kata lain sebuah informasi haruslah akurat.
2. Informasi itu harus tersedia atau dapat diakses pada saat informasi tersebut diperlukan dan tidak terhambat, dengan kata lain sebuah informasi haruslah tepat waktu.
3. Informasi yang disajikan harus sesuai dengan yang dibutuhkan, dengan kata lain sebuah informasi haruslah tepat dan relevan.
4. Informasi harus disajikan secara lengkap karena bila informasi yang dihasilkan setengah-setengah akan mepengaruhi dalam pengambilan keputusan.

### Sistem Informasi

Didalam (Hutahaean, 2015) menjelaskan bahwa, Sistem Informasi merupakan suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengelolah sebuah transaksi dengan mendukung operasi yang bersifat manajerial dan kegiatan strategi dari suatu organisasi untuk menyediakan sebuah *output* kepada pihak luar tertentu dengan berbentuk laporan-laporan yang dibutuhkan.

Sistem informasi didalam suatu organisasi juga dapat dikatakan sebagai suatu sistem yang mampu menyediakan informasi bagi semua hirarki atau tingkatan didalam suatu organisasi tersebut kapan saja diperlukan. Sistem ini mampu menyimpan informasi, mengambil informasi, mengubah informasi, mengolah dan mengkomunikasikan informasi yang diterima dengan menggunakan sistem informasi atau peralatan sistem lainnya(Anggraeni, 2017). Berikut merupakan ciri-ciri dari sistem informasi diantaranya yaitu:

1. Informasi yang disajikan sama sekali adalah informasi yang baru dan segar bagi penerimanya.
2. Dapat diperbaruhi atau memberikan modifikasi terhadap informasi yang sebelumnya telah ada.
3. Dapat menjadi suatu koreksi dari infomasi yang terjadi kesalah sebelumnya.
4. Dapat mempertegas informasi yang telah ada sebelumnya.

## LPPAIK

Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan (LPPAIK) merupakan salah satu lembaga yang beraada di Universitas Muhammadiyah Kendari dengan memiliki visi menjadikan civitas akademik Universitas Muhammadiyah Kendari bebas buta Baca Tulis Al-Qur’an (BTQ), bertakwa dan berakhlakul karima. Untuk mewujudkan visi itu LPPAIK memiliki misi untuk melaksanakan pemetaan kemampuan BTQ dan pemahaman keislaman, melaksanakan pembinaan BTQ secara intensif dengan mengembangkan metode pembinaan BTQ yang efisien dan efektif, dan juga menanamkan nilai-nilai keislaman kepada civitas akademik Universitas Muhammadiyah Kendari.

## Aplikasi Berbasis Web

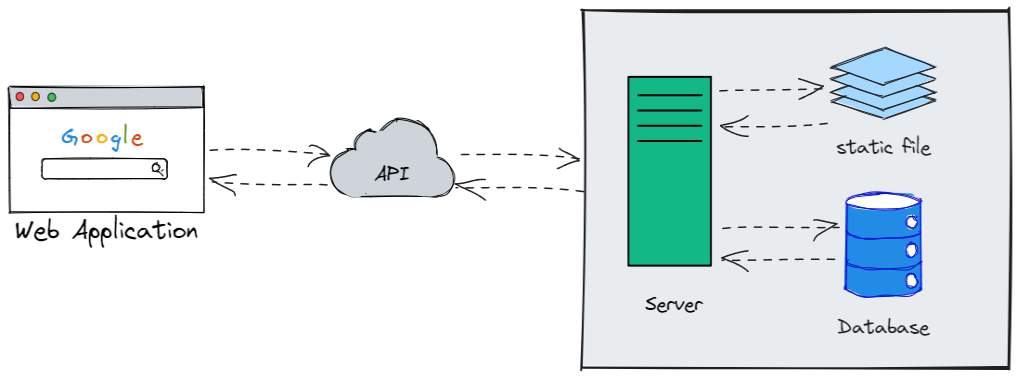
### *Website*

*Web* adalah salah satu layanan yang dapat digunakan oleh seluruh pengguna internet yang terhubung secara *online*. Pada mulanya *web* merupakan ruangan informasi dalam internet dengan menggunakan teknologi *hypertext*, yang dimana pengguna dituntun untuk menemukan informasi dengan mengikuti tautan dalam dokumen *web* yang ditampilkan dalam sebuah *web browser* (Gunawan, 2019). Sedangkan *website* merupakan kumpulan-kumpulan dari halaman *web* yang didalamnya terdapat sebuah domain yang mengandung informasi yang dibutuhkan. *Website* pada umumnya dibangun dari halaman-halaman *web* yang saling berhubungan, dengan artian bahwa *website* merupakan halaman yang berisikan infromasi teks, gambar, dengan memiliki animasi, suara atau gabungan dari semuanya, baik itu yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan sederhana maupun yang kompleks dengan saling memiliki keterkaitan dengan menggunakan *hyperlink*, sedangkan teks yang menjadi media penghubung disebut *hypertext*(Isa & Hartawan, 2017).

### Aplikasi berbasis *web*

Aplikasi berbasis *web* merupakan sebuah program komputer yang memanfaatkan *web browser* serta tekologi dari *web*. Aplikasi ini menggunakan *sciprt server-side* seperti *PHP* dan *ASP* *(Active Server Pages)* untuk memproses dan mengelolah data yang disimpan pada *server*. Selain itu, *script client-side* seperti *HTML* dan *JavaScript* juga digunakan untuk menampilkan informasi dan memungkinkan pengguna berinteraksi dengan aplikasi melalui *form online*, sistem manajemen konten, kolom komentar, dan fitur lainnya. Dengan aplikasi *web, user* dapat membuat dan berbagi dokumen, bekerja sama dalam suatu proyek, serta mengedit dokumen yang sama dengan menggunakan perangkat yang berbeda dari lokasi yang berbeda pula(Suryawinata, 2019).

Setiap permintaan yang dilakukan oleh *user* pada *website* melalui *web browser* akan diproses lagi oleh aplikasi *web* dan hasilnya akan dikembalikan lagi kepada *user* yang berupa halaman-halaman *web* yang akan tampil dilayar *web browser* mereka. *Website* yang bersifat dinamis tergantung dari nilai data atau parameter yang dimasukkan oleh *user* (Gunawan, 2019). Berikut adalah gambaran komunikasi antara *client* atau pengguna *(Web browser)* dan aplikasi *web.*



Gambar 2. Komunikasi Antara *Web* Browser dan Aplikasi *Web*

## *Web Service*

*Web service* adalah kumpulan layanan dari *web* dengan menggunakan jaringan protokol *HTTP* (*Hypertext Transfer Protocol*) untuk berkomunikasi yang dapat diakses dan digunakan oleh pengguna dengan berbagai macam bahasa pemrograman, arsitektur, dan sistem operasi yang berbeda, layanan *web service* ini dapat dibuat dalam bentuk format *JSON, XML* maupun teks. *Web service* merupakan teknologi yang dapat digunakan untuk melakukan integrasi sistem dan pengembangan aplikasi dengan berbasis *multiplatform*. *Web service* dikembangkan sebagai jembatan komunikasi untuk sistem-sistem yang berbeda satu dengan yang lainnya, sehingga aplikasi memiliki jaringan, standar protokol dengan cara berkomunikasi yang sama, yang ditetapkan oleh pengembang dari *web service*. Oleh karena itu dengan menggunakan *web service*, ketergantungan pada *website* konvensional akan dapat teratasi. Pada umumnya ada 2 jenis protokol komunikasi *web* standar yaitu *SOAP* dan *REST*, Namun saat ini protokol yang paling banyak digunakan dengan memiliki kinerja yang sangat baik dan optimal adalah *Representational State Transfer* (*REST*). *REST* yang dapat diterapkan pada *web service* yang saat ini dikenal dengan nama *RESTful* *web servcie*(Achsan & Susetyo, 2022).

## *RESTful API*

*Representational State Transfer* (*REST*) merupakan arsitektur perangkat lunak yang memberlakukan syarat mengenai cara *API* bekerja. *REST* pada awalnya dibuat sebagai panduan untuk mengelolah komunikasi pada jaringan kompleks seperti internet([Aws.amazon.com](https://aws.amazon.com/id/what-is/restful-api/)). Roy Fielding dari *University of California* merupakan pencipta dari *REST*. *REST* merupakan layanan *web* yang lebih sederhanan dibandignkan arsitektur jenis *SOAP, REST* yang menjadi pemisah antara klien dan *server* memungkinkan pengembang aplikasi yang independen baik dari sisi *platform*, bahasa maupun dari sisi skalabilitas(Prilsafira dkk., 2023).

*Application Programming Interface* (API) merupakan antarmuka yang memungkinkan aplikasi *backend* dapat berinteraksi dan membagikan data yang diambil melalui *database*. *API* sendiri terdiri dari elemen *function*, *protocols*, dan *tools* lainnya yang digunakan pengembang untuk membuat sebuah aplikasi. *Web* *API* berjalan pada semua jenis *server* baik itu *Apache* pada *PHP*, maupun *Tomcat* pada *Spring boot* dan lain sebagainya(Prilsafira dkk., 2023). *RESTful API* adalah antarmuka yang digunakan oleh dua sistem komputer untuk bertukar informasi secara aman melalui internet, dengan memiliki 6 prinsip yang harus diimplementasikan diantaranya (Restfulapi.net):

1. *Uniform Interface*: Menyederhanakan prinsip untuk meningkatkan visibilitas interaksi.
2. *Client Server:*Membantu komponen klient dan *server* berkembang secara mandiri untuk meningkatkan portabilitas antarmuka pengguna di beragai platform dan meningkatkan skalabilitas dengan menyederhanakan komponen *server****.***
3. *Stateless*: Status sesi dari aplikasi sepenuhnya disimpan pada klien dan *server* tidak pernah bergantung pada informasi dari permintaan sebelumnya dari kelien dalam artian setiap sesinya adalah independent.
4. *Cacheable*: Respon yang disyaratan harus secara implisit atau explisit melebeli dirinya sendiri sebagai dapat di-*cache* atau tidak dapat di-*cache*. *Caching* sendiri memliki kemampuan untuk menyimpan salinan data yang biasa diakses di beberapa tempat misal *email*, *username* dan *password*.
5. *Layered System*: Sistem berlapis *(Layered System)* memungkinkan mendesain arsitektur terdiri dari lapisan hierarki dengan batasan interaksi diantara mereka.

## *Framework*

*Framework* merupakan seperangkat *library* yang teroganisasi dalam bentuk arsitektur untuk memberikan kecepatan, ketepatan akurasi, kanyamanan dan juga konsistensi dalam pengembangan aplikasi(Sandhika Jaya dkk., 2017). Dalam (Sallaby & Kanedi, 2020) menjelaskan tentang *framework, Framework* merupakan kumpulan perintah atau instruksi-instruksi yang dipadukan dalam bentuk *class* dan *function-function* dengan memiliki fungsi masing-masing yang bertujuan untuk mempermudah pengembang dalam memanggil perintah tanpa harus menuliskan *code* program yang sama berulang-ulang serta dengan tujuan dapat menghamat waktu pengembangan.

### *ReactJS*

*ReactJs* adalah *opensource front-end library* yang dikembangkan oleh Facebook untuk memfasilitasi pengembang dalam membuat komponen UI yang lebih interaktif, *stateful* & *reusable*. *ReactJs* memiliki beberapa keunggulan dimana kerangka kerja ini memberikan kecepatan, *simplicity*, dan *scalability*. (Nursaid dkk., 2020). *ReactJs* merupakan salah satu *web framework* paling popular di dunia *Nodejs*, dengan memiliki dokumentasi yang baik membuat para pengembang *web* mudah untuk mengembangkan produk *web* aplikasi ataupun *RESTful API* yang akan dikembangka. *ReactJs* juga dapat diggunakan untuk menangani pengembangan pada aplikasi *single page* & aplikasi *mobile* dan dapat digunakan untuk menjadi dasar untuk mengembangkan *web framwork* yang lebih kompleks. *ReactJs* merupakan *JavaScript framework* yang paling populer pada saat ini dengan memiliki kemanan yang kuat dan juga memiliki komunitas yang besar. *ReactJs* memungkinkan pengembang untuk membuat komponen *user interface* (UI) yang dapat digunakan kembali(Wali & Ahmad, 2018)­. Dalam perumpamaan MVC *(Model, View, Controller)* *ReactJs* dapat merepresentasikan pada bagian *View* saja dan ini merupakan bagian terbaik dalam penyederhanaan(Nursaid dkk., 2020).

### *Spring* dan *Spring Boot*

*Spring* merupakan *framework* yang paling populer untuk bahasa pemrogramman *Java*. *Spring* juga membuat bahasa *Java* lebih cepat, lebih mudah, dan lebih aman untuk semua orang. Dengan berfokus pada kecepatan, kesederhanaan dan produktifitas (Spring.io). *Spring* adalah *framework* *opensource* yang dikembangkan oleh Rod Johnshon, pada tahun 1996 akhir, *Sun* *Microsystem* menerbitkan spesifikasi *Java beans*. Spesifikasi ini menjelaskan aturan pengkodean *Java* yang memungkinkan objek menjadi komponen yang dapat digunakan kembali dalam aplikasi *Java* yang lebih kompleks(Pratama & Robbani, 2023). *Spring* memiliki banyak *sub framework* salah satunya adalah *Spring Boot*. *Sprinng Boot* menyediakan module-module siap pakai, yang dapat memudahkan untuk membangun sebuah aplikasi berbasis *RESTful* *web service* (Umbu Dagha, 2021).

## Basis data

### *Database*

*Database* atau basis data terbagi menjadi dua kata yakni basis dan data. Basis merupakan gudang atau tempat berkumpul dan data merupakan representasi fakta mengenai dunia atau sesuatu kejadian. Pengetahuan tentang fakta yang direkam dan mempunyai unsur makna yang implisit. Dengan kata lain basis data merupakan kumpulan kelompok data yang berketerkaitan dan terorganisasi dengan sedemikian rupa agar dapat digunakan secara cepat dan mudah, dengan tujuan agar sekumpulan data yang saling berhubungan akan tersimpan secara tersendiri tanpa adanya perulangan data yang tidak perlu(Dhika dkk., 2019). Berikut adalah beberapa definisi database atau basis data menurut para ahli yang dikutip oleh (Jayanti & Sumiari, 2018) diantaranya sebagai berikut:

1. Menurut McFadden, Hoffer, dan Presscot: “*An organized Collection of logicall related data*”.
2. Menurut Date, basis data dapat dianggap sebagai “Tempat untuk sekumpulan berkas data terkomputerisasi, dengan tujuan utama mememlihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan”.
3. Menurut Chou, “Basis data sebagai sekumpulan informasi bermanfaat yang dioragnisasikan ke dalam tatacara yang khusus”.

Dalam (Yanto, 2016) menyebutkan beberapa komponen dalam sebuah sistem basis data diantaranya sebagai berikut:

1. Data: merupakan informasi yang terintegrasi dan disimpan dalam suatu struktur tertentu.
2. *Hardware*: merupakan perangkat keras komputer sebagai media penyimpanan data, dikarenakan pada umumnya basis data memerlukan suatu wadah untuk melakukan penyimpanan data.
3. Sistem Operasi: merupakan program yang mampu mengaktifkan dan mengorganisasikan sistem komputer, mengendalikan seluruh sumber daya didalam komputer, dan melakukan operasi dasar didalam komputer berupa *input*, proses dan *output*.
4. Basis Data: sebagai inti dari sistem basis data. Basis data mampu menyimpan data serta struktur sistem basis data baik untuk entitas maupun objek-objek secara terperinci.
5. *Database Management System*: merupakan perangkat lunak yang mampu melakukan pengelolaan basis data seperti *Mysql, Sql Server, Oracel* dan masih banyak lagi.
6. *User*: merupakan pengguna yang secara langsung dapat berinteraksi dengan data yang terlah tersimpan dan terkelola.
7. Aplikasi lainya: perogram yang dirancang untuk memberikan *interface* kepada pengguna atau *user* sehingga lebih mudah mengakses, melakukan operasi dan mengoragisasikan data di basis data.

### *Database Management System* (DBMS)

*Database Management System* (DBMS) pada awalnya didesain oleh Carles Bachman di perusahaan *Genaral Electric* pada awal tahun 1960, yang disebut sebagai Peyimpanan data Terintegrasi *(Integrated Data Store)*. DBMS merupakan perangkat lunak yang didesain untuk menangani pemeliharaan dan utilitas kumpulan data dalam jumlah yang besar. DBMS dapat menjadi alternatif penggunaan secara khusus untuk apliaksi. Sebagai contoh penyimpanan data dalam *field* dan menulis kode aplikasi spesifik untuk pengaturannya (Jayanti & Sumiari, 2018). Dalam (Yanto, 2016) Menjelaskan bahwa DBMS merupakan paket program *(software)* yang dibuat supaya memudahkan dan mengefisienkan operasi pemasukan, pengediatan, penghapusan dan pengambilan informasi terhadap *database*. *Software* yang tergolong kedalam DMBS antara lain, *Mysql, Sql Server, Oracel* dan masih banyak lagi.

### *MySQL*

*MySQL* merupakan *Relational Database Management System* (RDBMS) yang telah didistribusikan dengan gratis dibawah lisensi GPL *(General Public Licenses).* *MySQL* mulai dikembangkan pada tahun 1979 dengan *tool database* UNIREG yang dibuat Micheal Monty Widenius untuk perusahan TcX di swedia(Yanto, 2016). *MySQL* merupakan turunan salah satu konsep utama yang terdapat dalam database sejak dulu yaitu *Structured Query Language* (Dhika dkk., 2019). *Structured Query Language* atau *SQL* merupakan bahasa yang terstruktur untuk mengakses atau mengolah data pada *database* dan entitas -entitas yang ada pada database tersebut. *SQL* juga merupakan bahasa yang sudah standar digunakan dalam berbagai *database* sehingga pengembang mudah untuk menggunakanya walaupun berpindah dari satu *database* ke *database* lainnya(Pamungkas, 2018). MySQL merupakan *System management database* yang mendukung bahasa *query* yang bersifat *open source* atau gratis yang awalnya didistribusikan untuk *Linux* namun sekarang *MySQL* sudah bisa berjalan dibanyak *platform* dengan dilengkapi *soruce code*. Informasi yang tersimpan dalam bahasa *quey* terbagi atas dua bagian yaitu baris yang sering disebut *record* dan kolom yeng lebih dikenal dengan *field*(Yanto, 2016). Berikut keunggulan dari *System management database* MySQL daintaranya:

1. Kecepatan.
2. Kemudahan bagi pengguna dalam melakukan pengoprasiannya.
3. Bersifat *open source* atau gratis.
4. Mendukung bahasa *query*.
5. Pengguna mampu mengakses lebih dari satu dalam waktu yang bersamaan.
6. Pengguna dimungkinkan untuk mengakses data dari mana saja dengan memiliki fasilitas internet.
7. Mudah dipahami dikarenakan berisifat *open source* dengan dukungan komunitas yang banyak.

## Bahasa Pemrogramman

### *Java*

Saat ini *Java* merupakan salah satu bahasa pemrogramman yang sangat populer disebabkan dengan menggunakan bahasa ini kita bisa membuat aplikasi yang sangatlah luas cakupanya, mulai dari komputer hingga mobile phone. *Java* sendiri memiliki *tagline* “*Write Once, Run Anywhere”* yang bermakna program yang ditulis satu kali dan dapat berjalan pada banyak *platform*. Dengan demikian tidak mengherankan apabila aplikasi yang dibuat menggunakan *Java* bisa ditemukan di lingkungan komputer dan *mobile phone* tanpa ada perbedaan yang berarti (Jubilee, 2015). *Java* adalah bahasa pemrogramman yang *powerful* dan serbaguna untuk pengembagan perangkat lunak yang berjalan diperangkat seluler, kompuer *desktop* dan *server*. Bahasa pemrograman *Java* pertamakali dikembangkan oleh *Sun Microsystem* yang dimulai oleh James Gosling dan di *launching* pada tahun 1995. Saat ini *Sun Microsystem* telah diakuisisi oleh *Oracel Corporation* pada tahun 2010. Awalnya *Java* dipanggil dengan sebutan *Oak*, *Java* dirancang pada tahun 1991 untuk diggunakan dalam *chip* ternama pada peralatan elektronik yang ada di pasaran (Jubilee, 2017). Berikut adalah fitur-fitur yang disediakan pada bahasa pemrogramman *Java*:

1. Berorientasi *Object*: dalam *Java* semua direpresentasikan dalam sebuah *object*.
2. Platfrom Independen: *Java* di-*compile* ke dalam *bit code platform independen*.
3. Aman: Kita bisa membuat sistem yang bebas dari virus dengan fitur keamanan yang telah disediakan *Java*.
4. Sederhana: *Java* sendiri didesain oleh pengembangnya untuk mudah dipahami dan dipelajari.
5. Bersifat *Architectural-neutral*: *Java Compiler* membuat *format file* objek yang *architectrual-netural*, yang membuat kode yang *decompile* dapat dieksekusi pada berbagai prosesor yang memiliki sistem *runtime* dari *Java*.
6. *Robust* dan *powerful*: *Java* akan melakukan pengecekan awal saat waktu *compile* dan *runtime*.
7. *Portable*: Karena adanya fitur platform independen dan *architectural-neutral* yang membuat *Java* menjadi *portable*.
8. *Multithreaded*: Pengguna bisa membuat program yang dapat mengerjakan banyak tugas di waktu yang bersamaan secara bersamaan.
9. Terintepretasi: Kode dari *Java bite code* ditranslasi secara langsung pada instruksi mesin dan tidak disimpan.
10. Performa: *Java* memiliki performa yang tinggatanya tinggi dikarenakan menggunakan *compiler* secara langsung.
11. Terdistribusi: Desain dari java untuk lingkungan distribusi internet.
12. Dinamis: *Java* didesain untuk dapat beradaptasi dengan lingkungan pengembangan (Jubilee, 2015).

### *JavaScript*

Harus dipahami bahwa *JavaScript* bukalah *Java*. Keduanya merupakan bahasa yang sangat jauh berbeda. *Java* dikembangkan oleh *Sun Microsystems* sedangkan *JavaScript* sendiri dikembangkan oleh NetScape. Aplikasi yang dibangun pada bahasa pemrogramman *Java* bersifat paltform independen, sedangkan program *JavaScript* ditanam ke dalam halaman *web* dan harus dijalankan pada jendela *browser* pengguna. *Java* memiliki bahasa dengan aturan yang super ketat, sedangkan *JavaScipt* memiliki bahasa dengan aturan yang fleksible dan tidak terlalu ketat. *JavaScript* adalah bahasa skript populer yang dipakai untuk menciptakan halaman *web* yang dapat melakukan interaksi dengan penggunanya dan dapat merespon *event* yang terjadi pada sebuah halaman. *Javascript* merupakan penghubung yang akan menjahit dan menyatukan *page-page* halaman *web*.

*JavaScript*, pada mulanya dikenal sebagai *LiveScript*, yang dikembangkan oleh Brendan Eich di *Netscape* pada tahun 1995 yang menjadi bagian yang terhubung di dalam *Netscape Navigator* 2.0. *Javascript* merupakan bahasa skript yang membuat halama-halaman *HTML* menjadi dinamis. *Javascript* dapat dijalankan pada hampir semua platform. *Javascript* merupakan bahasa sisi-klien yang didesain untuk *browser* komputer pengguna, bukan pada *server*. Ia dibangun secara langsung ke dalam *browser*, *Mozila Firefox, Google Chrome, Opera* dan *browser* lain sebagainya (Sianipar, 2015).

Pada era saat ini akan sangat susah menjumpai halaman *web* komersial yang tidak membangun *website* mereka tanpa membuat kode *JavaScript*. Dikarenakan *JavaScript* terkait dengan *browser*, *JavaScript* juga sangat terintegrasi dengan *HTML*. Ketika *browser* memuat sebuah halaman, *server* akan mengirim konten utuh dengan bentuk dokumen, termasuk *HTML* dan perintah-perintah dari *JavaScript*. Konten *HTML* kemudian dibaca dan diinterpretasi baris-demi-barisnya sampai *tag* pembuka *JavaScript* dibaca, pada saat itu interpreter *JavaScript* mengambil alih. Ketika *tag* penutup *JavaScript* diraih, pemrossesan *HTML* akan berlanjut (Sianipar, 2015).

### *NodeJs*

*NodeJs* merupakan *runtime environment* yang digunakan untuk pengembangan aplikasi berbasis *web*. *JavaScript* hanya bisa dijalankan pada *web browser* saja sedangkan *NodeJs* dapat dieksekusi sebagai aplikasi *server*. *NodeJs* dapat berjalan dari sisi *server* dikarenakan dukungan dari *V8 Engine* buatan *Google* dengan disediakan beberapa modul bawaan yang terintegrasi seperti module *filesystem, http,* dan yang lainnnya(Fajrin, 2017).

*Node Package Manager* (NPM) merupakan sebuah penyedia *package repository* *open source*, NPM memungkinkan pengguna *NodeJs* mempublikasikan hasil karyanya agar dapat digunakan oleh orang lain yang membutuhkan tanpa adanya *proses screening*(Haryana, 2019). NPM akan dibutuhkan untuk menginstal *dependency* pendukung dalam pengembangan aplikasi *NodeJs* berbasis *web*.

## *Programming Tools*

### Intelij IDEA

Intelij IDEA merupakan *Integrated Developent Environment* yang banyak digunakan para pengembang program *Java*. InteliJ IDEA merupakan hasil karya dari JetBrains yang pertama kali diluncurkan pada januari 2001 dengan diusung sebagai aplikasi pengembang program *Java* pertama dengan penavigasian dan prefektur kode program dengan tingkat lanjut. Dengan tujuan pembuatannya untuk dapat digunakan pengembang program atau aplikasi. InteliJ IDEA juga dapat diintegrasikan keberbagai platform diantaranya VCS, GIT, SVN dan lain sebagainya(Saputra & Stefani, 2023). Pada penelitian ini Intelij IDEA akan digunakan sebagai *teks editor* atau IDEA untuk pengembangan *REST* *Web Service* menggunakan *Spring Boot* dan menggunakan bahasa pemrogramman *Java*.

### VSCode

Salah satu *teks editor* yang ringan dan memiliki kemampuan yang handal dibuat oleh Microsoft yang dapat berjalan di berbagai sistem operasi yaitu *Visual Studio Code* atau yang lebih dikenal dengan VSCode. Teks editor yang di buat oleh Microsoft ini mendukung berbagai macam bahasa pemrogramman untuk dapat dijalankan diatas *teks editor* yang satu ini diantaranya *JavaScript*, *Typescript, NodeJs* serta banyak lagi bahasa pemrogramman lainnya.

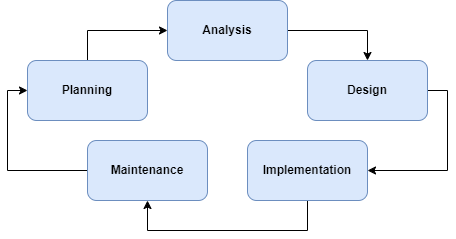
VSCode menjadi sangat poluler dikarekan VSCode merupakan *teks editor* yang *open source* dengan memiliki banyak fitur yang bisa digunakan para pengembang atau *developer* diantaranya *Intellisense, Git Integration, Debuging* dan fitur extensi yang membuat kemampuan teks editor ini manjadi sangat *powerful* dan mempermudah dalam proses *development*. Fitur yang disebutkan diatas terus mengalami perbaikan dan penambahan fitur. VSCode sendri selalu melakukan pembaruan setiap bulan ini lah yang membuat VSCode berbeda dengan *teks editor* yang lainnya(Salama, 2021). Dalam penelititan ini VSCode akan diggunakan untuk pengembangan *front-end developement* dengan menggunakan *JavaScript*, *HTML,* *Tailwind* dan lain sebagainya.

### Postman

*Postman* adalah sebuah *software* *Testing REST Client* dengan berbasis web yang tersedia dalam bentuk ekstensi pada Google Chrome. Postman mempunyai tampilan *user interface* yang memudahkan dalam meggunakannya. Postman juga mempunyai fitur-fitur pendukung diantaranya *design, API test, build,* dan documentasi API(Fajrin, 2017).

## *Software Development Life Cycle* (SDLC)

*Software Development Life Cycle* (SDLC) adalah proses pengembangan suatu sistem perangkat lunak dengan menggunakan model-model dan metodologi yang digunakan untuk mengembangkan sistem perangkat lunak. SDLC juga bisa berupa pola yang diggunakan untuk mengembangkan sistem perangkat lunak, yang terdiri dari beberapa tahap diantaranya: *planning* (perencannan), *analysis* (penganalisisan), *design* (pendesainan), *implementation* (pengimplementasian), *testing* (pengujicobaan) dan yang terakhir adalah *maintenance* (pengelolaan/perawatan). Didalam proses rekayasa perangkat lunak, konsep SDLC dapat menjadi dasar dari berbagai jenis pengembangan metodologi perangkat lunak. Metodologi ini dapat membentuk sebuah kerangka kerja untuk proses perancanaan dan pengendalian pembuat sebuah sistem informasi. SDLC sendiri memiliki beberapa model yang dapat digunakan diantaranya model *Waterfall*, model *Prototype*, model *Rapid Application Development* (RAD) dan lain sebagainya(Findawati, 2018).

****

Gambar 2. Model Siklus Pengembangan Sistem

## *Unifide Model Language* (UML)

*Unifide Model Language* (UML) adalah bentuk pemodelan sistem perangkat lunak standar berbais komponen dan berorientasi objek. Penggunaan UML sendiri diharapkan mampu memberikan penjelasan secara visual pada model sistem perangkat lunak. Tidak hanya untuk memberikan sebuah bantuk gambar visual, UML juga mampu membantu menyelesaikan permasalahan dalam pengembangan perangkat lunak yang berkualitas tinggi dalam jangka waktu yang masuk akal. UML sangatlah berperan penting pada saat dimana klien memiliki beberapa persyaratan tinggi yang dapat berupa desain, pemeliharaan, pengelolaan dan pengembangan terhadap perangkat lunak dengan mencakup pemangkasan siklus pengembangan perangkat lunak, peningkatan kualitas perangkat, dan perluindungan pengembangan perangkat(Nursaid dkk., 2020).

## Penelitian Relevan

Pada proses melakukan penelitian ilmiah dibutuhkan adanya satu kajian pustaka. Proses menkaji kajian pustaka dianggap memiliki peran yang penting karena diangap sebagai landasan dalam penyusunana laporan penelitian yang berfungsi sebagai pencegahan teradap adaanya duplikasi atau kesamaan dari sebuah penelitian(Ridwan dkk., 2021). Berikut ini merupakan beberapa penelitian yang dilakukan oleh peneliti terdahulu yang dapat dijadikan sebagai landasan atau gambaran untuk penelitian yang sekarang sedang dilakukan.

Tabel 2. Penelitian Relevan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | Peneliti/ Tahun | Judul | Metode Yang Digunakan | Hasil Penelitian |
| 1. | Hernata & Samsudin / 2017 | Sistem Informasi Lembaga Pengkajian Studi Islam (LPSI) Universitas Islam Indragiri Tembilahan Berbasis Web. | Siklus pengembangan system menggunakan SDLC. | Sistem rancangan yang di buat mampu memberi kemudahan dalam mengelolah data LPSI dan layak di implementasikan. |
| 2. | Karman & Rahmanto / 2020 | Sistem Informasi Jadwal Kegiatan Majelis Ilmu pada Kajian Linggau Mengaji Berbasis Web Mobile. | Metode pengembangan system menggunakan *Waterfall* | Sistem ini dapat membantu dalam penyebaran infrormasi jadwal kajian linggau mengaji khususnya dikawasan kota lubuklinggau dan sekitarnya. |
| 3. | Firdhayanti dkk / 2023 | Perancangan Sistem Informasi Daftar Kajian Islam di Bandar Lampung Berbasis Web. | Metode pengembangan system menggunakan *Rapid Application Development (RAD).* | Sistem mampu mepermudah akses dan pengelolaan informasi mengenai kajian-kajian Islam di Bandar lampung dan juga dapat mempermuda mencari dan memperoleh informasi yang bermanfaat bagi pengembangan diri dan pemahaman agama. |

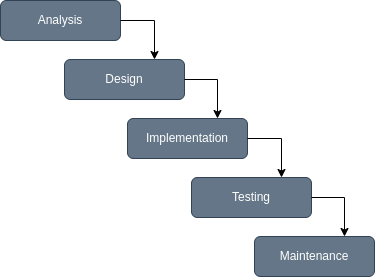
Berdasarkan dari ketiga penelitian relevan diatas, perbedaan yang akan dilakukan oleh penulis dengan judul “Perancangan Sistem Informasi Layanan Lembaga Pengkajian dan Penerapan Al-Islam Kemuhammadiyahan Universitas Muhammadiyah Kendari Berbasis Web” ialah sebagai berikut.

1. Penelitian pertama oleh (Hernata & Samsudin, 2017) terdapat perbedaan yaitu pada fitur atau halaman transaksi, laporan pembayaran dan galeri, sistem yang akan penulis buat tidak mengandung fitur atau halaman transaksi, laporan pembayaran dan galeri.
2. Penelitian kedua yang dilakukan oleh (Karman & Rahmanto, 2020) juga memiliki halaman galeri sedangkan pada sistem yang akan dibuat penulis tidak memiliki halaman galeri.
3. Penelitian ketiga yang dilakukan oleh (Firdhayanti dkk., 2023) dibuat menggunakan *PHP* dan *MySQL* dengan hanya berfokus pada fitur atau halaman informasi daftar kegiatan kajian saja, sedangkan pada sistem yang akan dibuat penulis menggunakan *ReactJs* dan *Spring Boot* dengan memiliki fitur lebih banyak dibandingan penelitian sebelumnya.

# METODOLOGI PENELITIAN

## Jenis Penelitian

Penelitian ini akan menggunakan metode *waterfall* untuk pengembangan sistem. Metode *waterfall* sendiri merupakan salah satu model pengembangan sistem dari *Software* *Development Life Cycle* (SDLC) dengan melakukan pendekatan klasik dalam pengembangan perangkat lunak yang menggambarkan metode pegembangan linier dan berurutan(Findawati, 2018). Adapun tahap-tahap dari model *waterfall* yang digunakan dalam perancangan sistem infromasi ini dapat dilihat pada gambar dibawah ini.



Gambar 3. Model Waterfall (Findawati, 2018)

## Prosedur Penelitian

Untuk memudahkan peneliti dan meningkatkan pemahaman tentang kebutuhan dalam melakukan penelitan sesuai dengan Gambar 3.1 model *waterfall*, maka prosedur dari perancangan sistem ini adalah sebagai berikut:

### Analisis (Analaysis)

*Analiysis* merupakan tahapan atau proses pengumpulan kebutuhan sistem secara menyeluruh yang selanjutnya akan didefinisikan kebutuhan yang harus dipenuhi oleh sistem yang di rancang(Findawati, 2018). Pada tahap ini penulis melakukan proses pengumpulan data yang dibutuhkan dengan cara melakukan observasi, wawancara, dokumentasi dan studi pustaka terkait dengan sistem yang akan dirancang.

1. Analisis sistem berjalan

Sistem yang berjalan pada lembaga LPPAIK saat ini masih bersifat manual baik itu kegiatan BTQ maupun kajian untuk mahasiswa, sehingga proses dari pelaksanaan kegiatan mulai dari penyimpanan informasi sampai dengan capaian pembelajaran maupun pengabsean kehadiran yang berbentuk laporan masih disimpan dalam bentuk kertas.

Tahapan analisis terhadap sistem yang sedang berjalan dilakukan agar penulis mengetahui alur kerja sistem yang sedang berjalan dengan tujuan melakukan penilaian serta memberikan gambaran rencana pemecahan masalah yang akan dihadapi. Berikut adalah prosedur yang sedang berjalan pada LPPAIK adalah sebagai berikut:

1. Modul BTQ
2. Mahasiswa datang langsung ke ruangan BTQ.
3. Mahasiswa melakukan kegiatan belajar Al-Qur’an dengan tutor sebaya.
4. Tutor sebaya memberikan penilain pada buku kontrol BTQ.
5. Proses pembelajaran dilakukan secara terus menerus sampai dengan mahasiswa paham bagaimana cara membaca Al-Qur’an dengan baik dan benar.
6. Jika bacaan mahasiswa sudah dirasa mampu atau cukup, Sertifikat kelulusan BTQ akan diberikan setelahnya (pada saat kelulusan).
7. Kajian Mahasiswa
8. Mahasiswa datang langsung ke tempat kajian.
9. Mahasiswa mengikuti kajian.
10. Mahasiswa melakukan absen.
11. Program studi mendapatkan absen setiap kelas.

Evaluasi sistem ditujukan untuk mencari kelemahan dari sistem yang akan dirancang. Berdasarkan uraian mengenai analisis sistem berjalan pada LPPAIK, maka penulis menemukan beberapa kekurangan atau kelemahan yang diantaranya:

Tabel 3. Evaluasi Sistem

|  |  |  |
| --- | --- | --- |
| No | Kelemahan | Solusi |
| 1. | Disebabkan kegitan belajar Al-Qur’an yang berulang dilakukan ini, sehingga mahasiswa diwajibkan membawa buku kontrol setiap melakukan kegiatan belajar Al-Qur’an. | Menggunakan sistem informasi yang didalamnya mampu menyimpan dan menambahkan infromasi data/*history* pembelajaran BTQ. |
| 2. | Sertifikat kelulusan yang diberikan hanya sekali dalam bentuk kertas, jika terjadi hal hal yang tidak diinginkan seperti kehujanan, hilang dan lain sebagainya maka hal ini bisa menjadi masalah. | Menggunakan sistem informasi yang dapat memberikan sertifikat digital yang bisa diprint kapan saja dan dimana saja, ketika dibutuhkan. |
| 3. | Terkadang informasi pembelajaran ini tidak hanya dibutuhkan oleh mahasiswa saja melainkan pihak lain yang berkepentingan mengetahuinya. | Menggunkan sistem informasi yang dapat diakses oleh orang yang berkepentingan dan bukan hanya mahasiswa yang bersangkutan saja. |
| 4. | Terkadang sulit dan membutuhkan waktu untuk mencari nama mahasiswa yang bersangkutan jika nama tersebut tersimpan didalam absen | Menggunakan sistem informasi yang mampu melakukan pencarian dengan cepat. |

1. Analisis sistem ajuan

Berdasarkan alalisa sistem yang diusulkan maka diketahui sistem yang lama masih bersifat manual dan memiliki beberpa tahap yang bisa dimaksimalkan lagi dengan menggunakan sistem informasi, sehingga diangap kurang mampu untuk memenuhi kebutuhan dalam pengelolahan data kegiatan maupun hasil dari laporan kegiatan yang diselengarakan. Setelah seluruh kebutuhan dari sistem diketahui maka langkah selanjutnya adalah merancang sistem informasi layanan LPPAIK yang bertujuan untuk memenuhi kebutuhan dari lembaga dalam melakukan pengelolaan kegiatan secara efektif dan efisien.

Secara keseluruhan prosedur yang dibuat tidak terlalu banyak mengalami perubahan dari sistem yang sedang berjalan, hanya saja berbeda dalam hal penyimpanan informasi yang berupa laporan hasil kegiatan dengan memaksimalkan sistem informasi komputerisasi dan teknologi internet yang diharapkan dapat mengefektifkan pengelolaan kegiatan yang diselengarakan lembaga.

Berikut adalah prosedur sistem informasi layanan pada LPPAIK yang diusulkan diantarnaya:

1. Module BTQ
2. Mahasiswa login ke sistem.
3. Mengikuti pembelajaran seperti bagaimana mestinya.
4. Tutor sebaya memberi penilaian terhadap capaian bacaan Al-Qur’an mahasiswa.
5. Mahasiswa mendapatkan laporan hasil pembelajaran yang dilakukan.
6. Mahasiswa mendapatkan sertifikat digital (.pdf) file dari LPPAIK (jika dinyatakan sudah cukup mampu membaca Al-Qur’an pada saat kelulusan/ sarjana).
7. Kajian Mahasiswa
8. Mahasiswa login ke sistem.
9. Mahasiswa membaca kegiatan yang diselengarakan lembaga LPPAIK.
10. Mahasiswa mengikuti kegiatan (mandatory atau opsional) tergantung spesifikasi kegiatan yang telah dibuat oleh admin.
11. Mahasiswa mendapatkan laporan hasil kegiatan yang telah ia ikuti.
12. Mahasiswa mendapatkan laporan digital (.pdf) dari lembaga yang nantinya diggunakan untuk menawar matakuliah selanjutnya.
13. Kebutuhan data

Data yang dibutuhkan dalam merancang sistem informasi layanan LPPAIK berbasis *web* diantaranya ialah sebagai berikut:

1. Data seluruh tutor sebaya.
2. Data kegiatan pada LPPAIK.
3. Data jurusan yang berada di UMK.

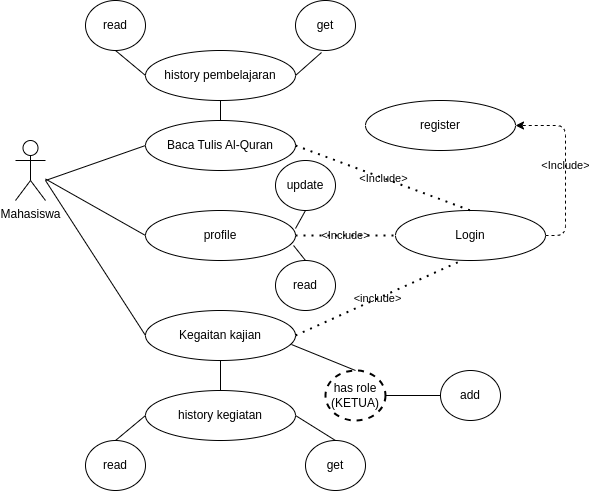
### Desain (*Design*)

Pada tahap *design*/desain ini penulis akan menghasilkan rancangan awal suatu sistem secara keseluruhan dan menentukan alur dari perangkat lunak(Findawati, 2018). Tahap desain sistem infromasi LPPAIK ini menggunakan bahasa pemodelan *Unified Modeling Language* (UML) diantranya *use case diagram*, *activity diagram, ERD* dan *mockup*.

1. *Use case* diagram

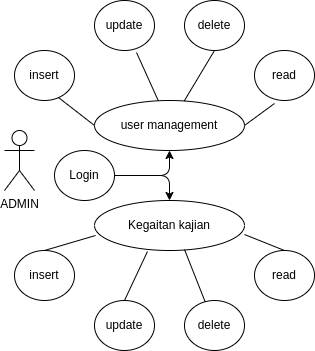
*Use case* diagram merupakan teknik untuk membuat persyaratan fungsional pada sebuah sistem dengan mereperesentasikan sebuah interaksi antara aktor pada sistem, yang dapat mengambarkan fungsional yang diharapkan dari sebuah sistem. *Use case* diagram lebih terfokus kepada “apa” yang dilakukan oleh sistem, dan bukan “bagaimana”(Findawati, 2018). Pada diagram *use case* ini akan menampilkan fungsional para aktor pada sistem infromasi LPPAIK diantaranya sebagai berikut:

1. *Use case* diagram mahasiswa

****

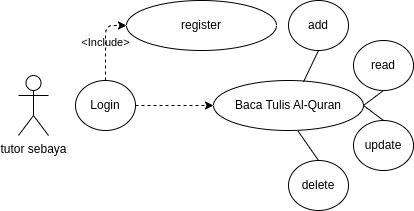
Gambar 3. *Use Case* Diagram Mahasiswa

1. *Use case* diagram admin

****

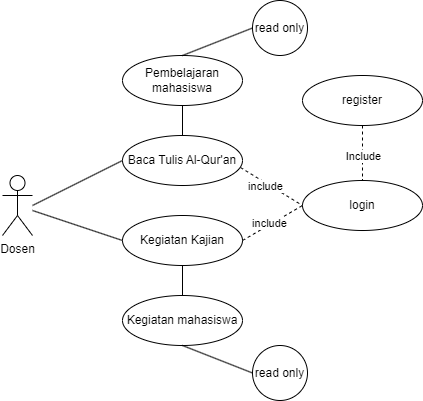
Gambar 3. *Use Case* Diagram Admin

1. *Use case* tutor sebaya

****

Gambar 3. *Use Case* Tutor Sebaya

1. *Use case* dosen

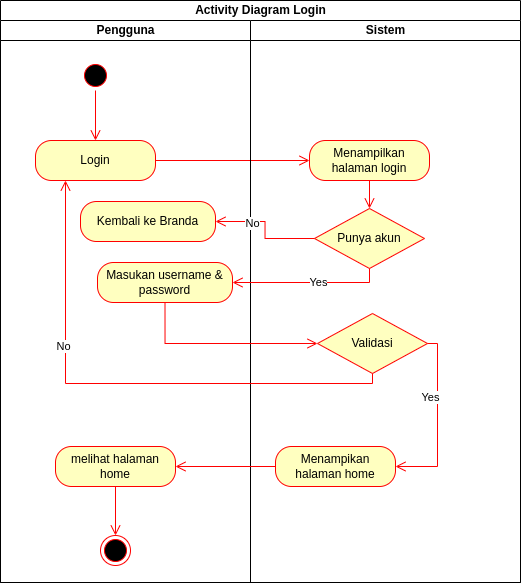


Gambar 3. *Use Case* Dosen

1. *Activity* diagram

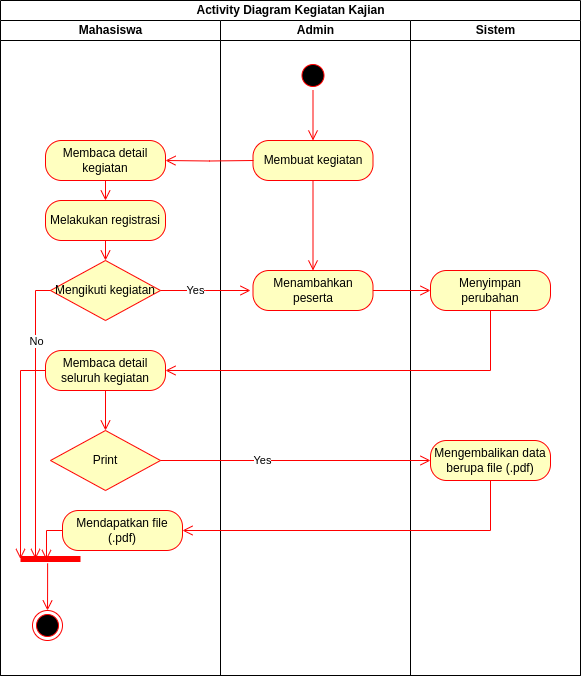
*Activity* diagram atau diagram aktifitas adalah tempat diagram khusus, yang dimana sebahagian besar tempat atau *state* memiliki *action* perilaku dan sebagai merupakan besar transisi yang di-*trigger* oleh selesainya *state* sebelumnya. *Activity* diagram juga digunakan untuk manggambarkan langkah-langkah atau aktivitas pada suatu sistem dengan memiliki kemampuan mengambarkan alir *activity* dalam sistem yang sedang dirancang, bagaimana tiap-tiap alir berawal, kondisi yang mungkin terjadi, dan juga bagaimana mereka mengakhiri aktivitas(Findawati, 2018). Pada *activity* diagram ini akan mengambarkan setiap *action* pada *state* para aktor di sistem informasi LPPAIK diantaranya sebagai berikut:

1. *Activity* diagram login user



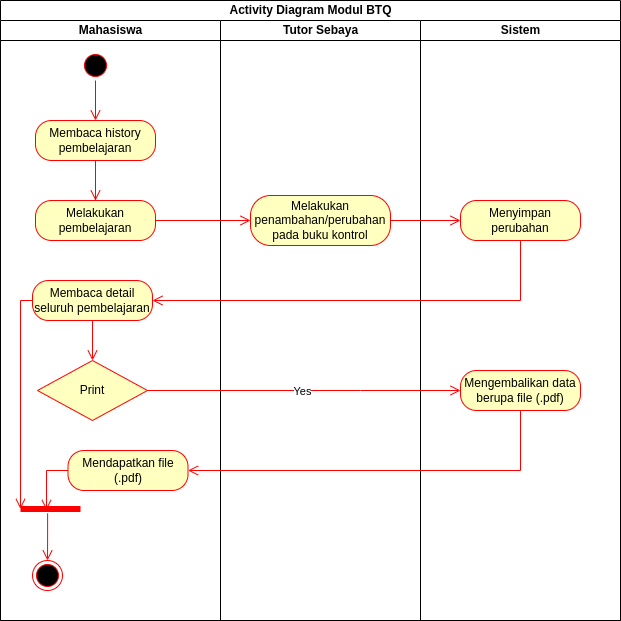
Gambar 3. *Activity* Diagram *Login Users*

1. *Activity* diagram kegiatan kajian



Gambar 3. *Activity* Diagram Kegiatan Kajian

1. *Activity* diagram module BTQ



Gambar 3. *Activity* Diagram Module BTQ

1. Desain *database*

Perancangan *database* adalah kegiatan menentukan isi data-data yang dibutuhkan untuk mendukung perancangan *website*. Model *relationship* adalah model yang dipakai untuk perancangan *website* sistem infromasi ini dimana seluruh tabel data yang digunakan diantaranya sebagai berikut:

Tabel 3. Tabel users

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| username | VARCHAR(10) | PRIMARY KEY |
| name | VARCHAR(255) |  |
| password | VARCHAR(255) |  |
| email | VARCHAR(255) | UNIQUE |
| avatar | VARCHAR(255) |  |
| completed | BIT(1) |  |
| gender | VARCHAR(20) |  |
| major | VARCHAR(255) | FOREIGN KEY |
| motto | MEDIUMTEXT |  |
| token | VARCHAR(255) |  |
| token\_expired\_at | BIGINT |  |

Tabel 3. Tabel activities

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| title | VARCHAR(255) |  |
| mandatory | BIT(1) |  |
| date | DATE |  |
| start\_time | TIME |  |
| end\_time | TIME |  |
| description | MEDIUMTEXT |  |
| location | VARCHAR(255) |  |
| link | TEXT |  |

Tabel 3. Tabel user\_activities

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| user\_username | VARCHAR(10) | PRIMARY KEY |
| activity\_id | VARCHAR(255) | PRIMARY KEY |

Tabel 3. Tabel majors

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| name | VARHCAR(255) |  |

Tabel 3. Tabel certificates

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| user\_username | VARCHAR(10) | PRIMARY KEY |

Tabel 3. Tabel roles

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| role\_id | VARCHAR(255) | PRIMARY KEY |
| user\_username | VARCHAR(10) | PRIMARY KEY |

Tabel 3. Tabel user\_roles

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| name | ENUM | PRIMARY KEY |

Tabel 3. Tabel images

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | INT | PRIMARY KEY |
| path | TEXT |  |
| data | LONGBLOB |  |
| type | VARCHAR(100) |  |

Tabel 3. Tabel activity\_images

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| activity\_id | VARCHAR(255) | FOREIGN KEY |
| image | MEDIUMTEXFT |  |

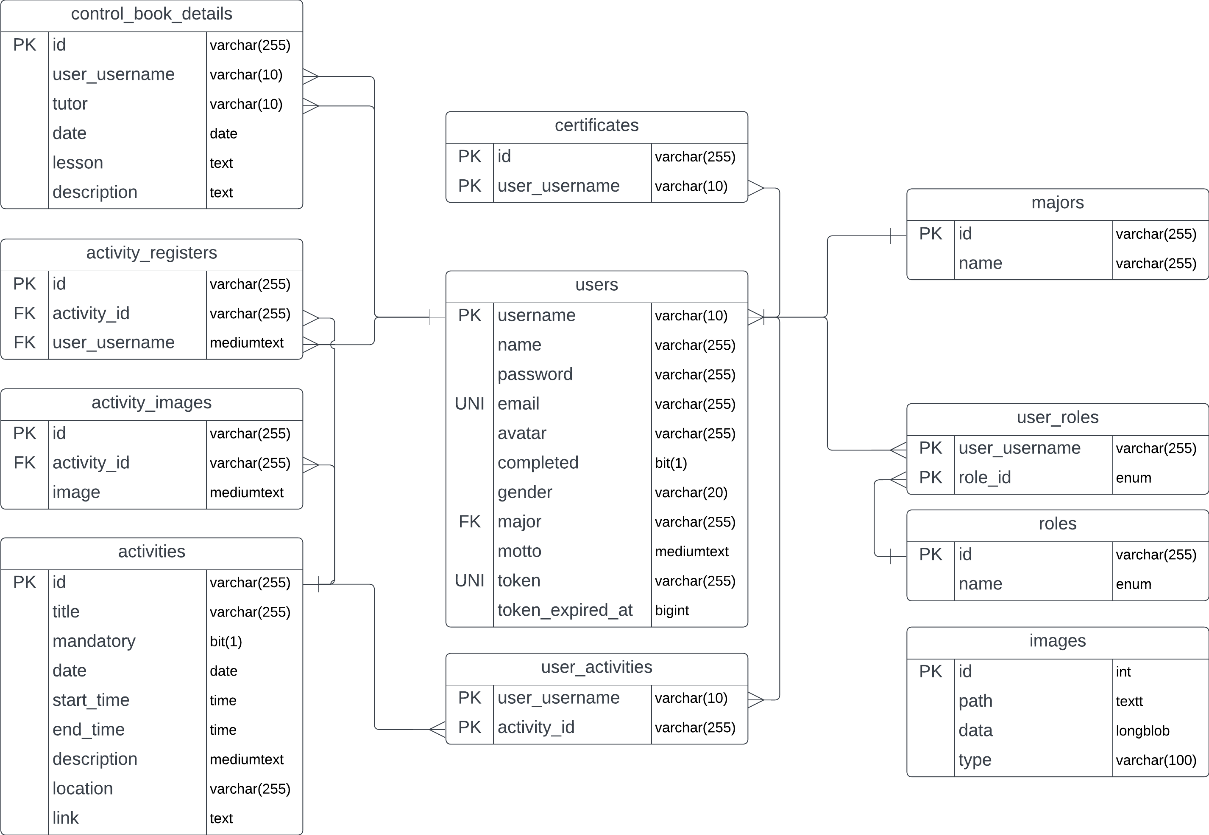
Tabel 3. Tabel activity\_registers

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| user\_username | VARCHAR(10) | PRIMARY KEY |

Tabel 3. Tabel control\_book\_details

|  |  |  |
| --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Keterangan** |
| id | VARCHAR(255) | PRIMARY KEY |
| user\_username | VARCHAR(10) | FOREIGN KEY |
| lesson | TEXT |  |
| tutor | VARCHAR(10) | FOREIGN KEY |
| description | TEXT |  |
| date | DATE |  |

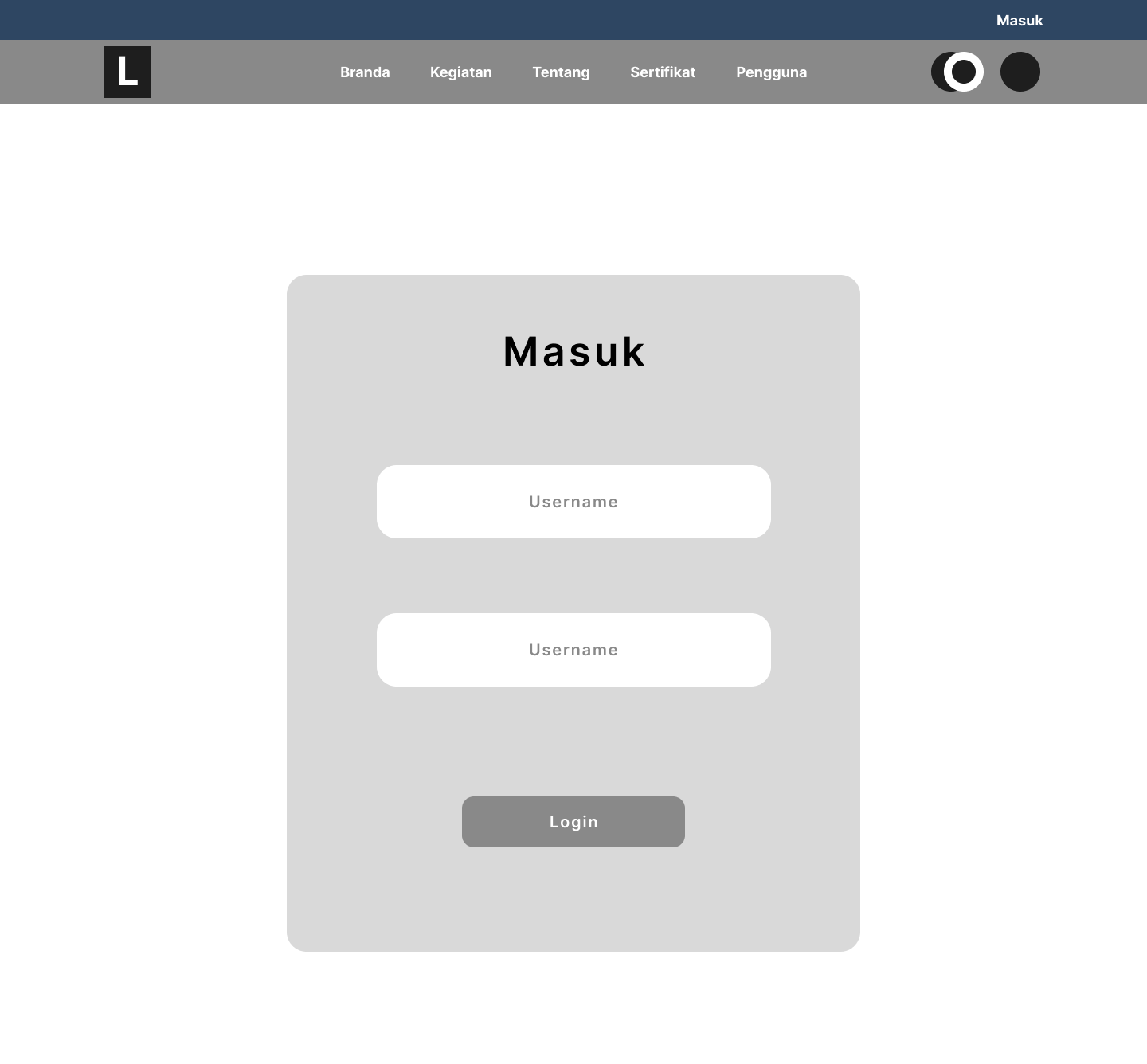
Entity Relational Diagram (ERD) merupakan diagram dalam bentuk gambar-gambar atau simbol-simbol yang mengindentifikasi tipe maupun jenis dari entitas pada suatu sistem yang disampaikan dalam bentuk data dengan atributnya, dengan menjelaskan hubungan atau relasi diantara entitas(Findawati, 2018). Perancangan database ini menggunakan ERD, dalam perancangannya hal ini melihat dari table database yang dibuat diatas mengenai data tabel berdasarkan fungsi yang diberikan. Berikut adalah gambaran ERD dari database sistem informasi layanan LPPAIK.



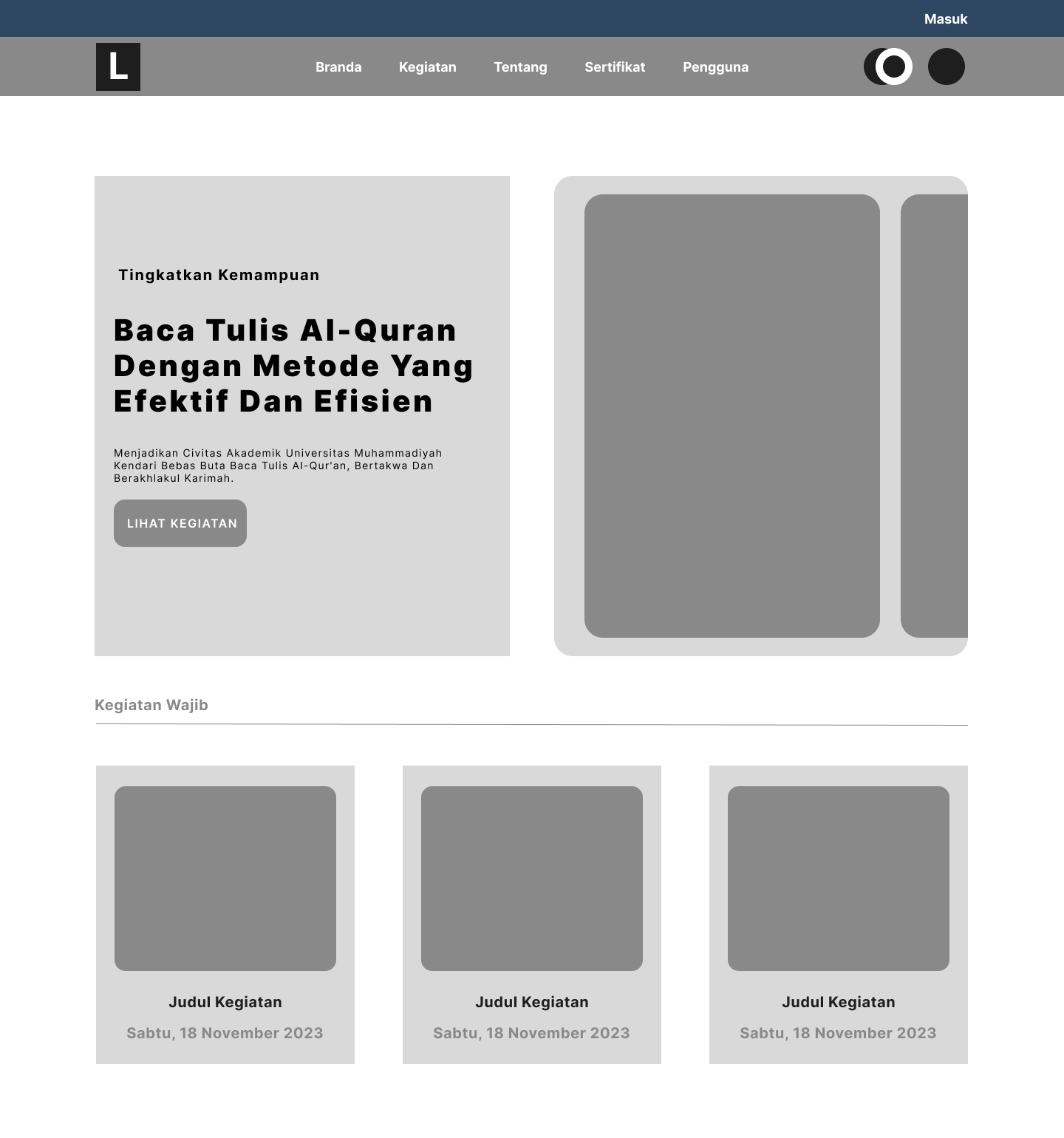
Gambar 3. ERD Sistem Infromasi Layanan LPPAIK

1. *Mockup*

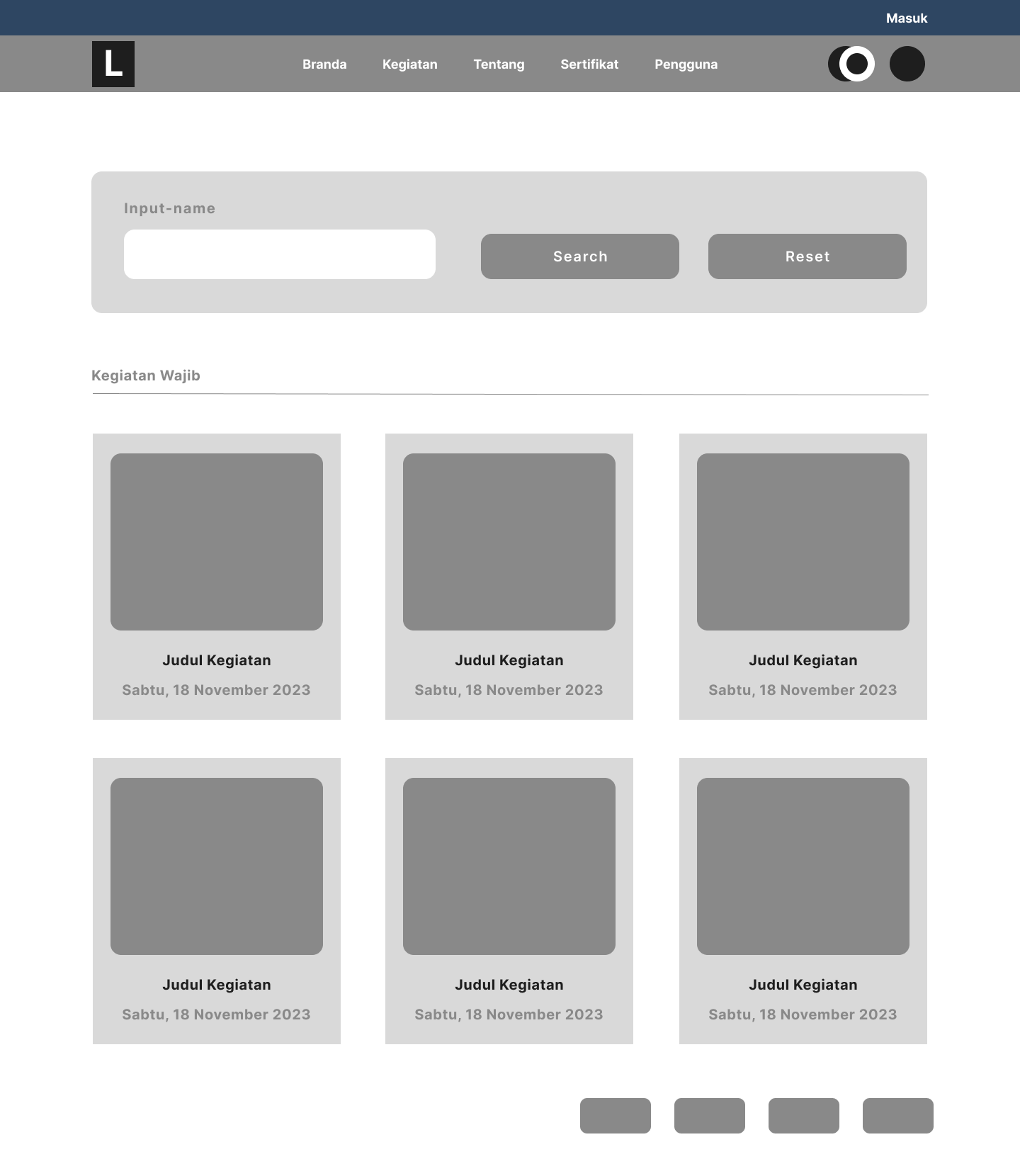
*Mokup* merupakan tampilan sementara dari *User interface* atau antarmuka pengguna yang dibuat untuk memudahkan proses desain antarmuka sistem yang akan dikembangkan. Sebuah *interface* dibuat dengan tujuan untuk mengkomunikasikan fitur-fitur *system* yang tersedia agar user mengerti dan dapat menggunakan *website* tersebut. Penyusunan desain tampilan antarmuka perlu diperhatikan untuk menghasilkan tampilan antar muka yang bagus dan menarik. Berikut ini adalah tampilan penjelasan sistem informasi layanan LPPAIK.



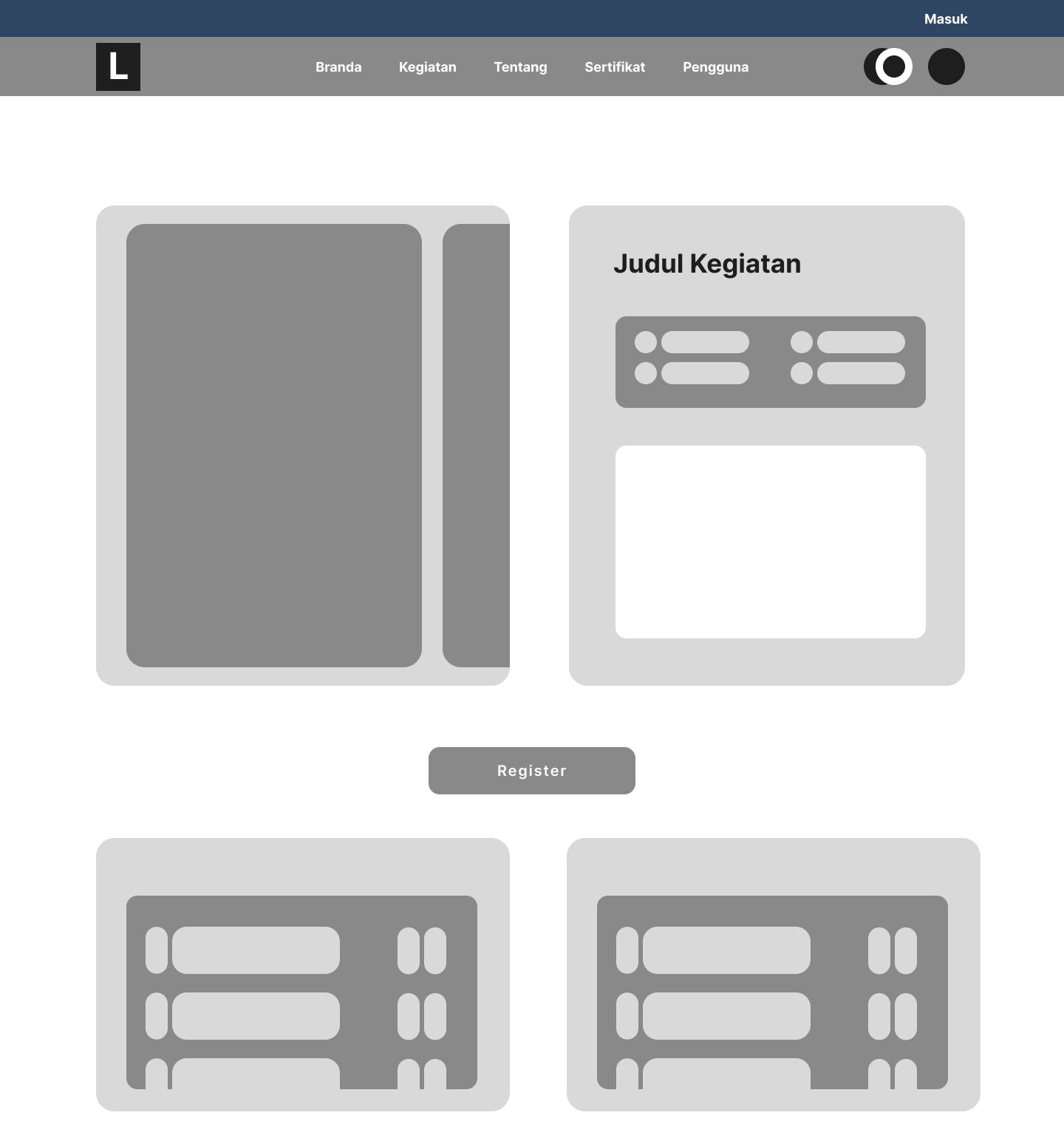
Gambar 3. *User* *Interface* Tampilan *Login*



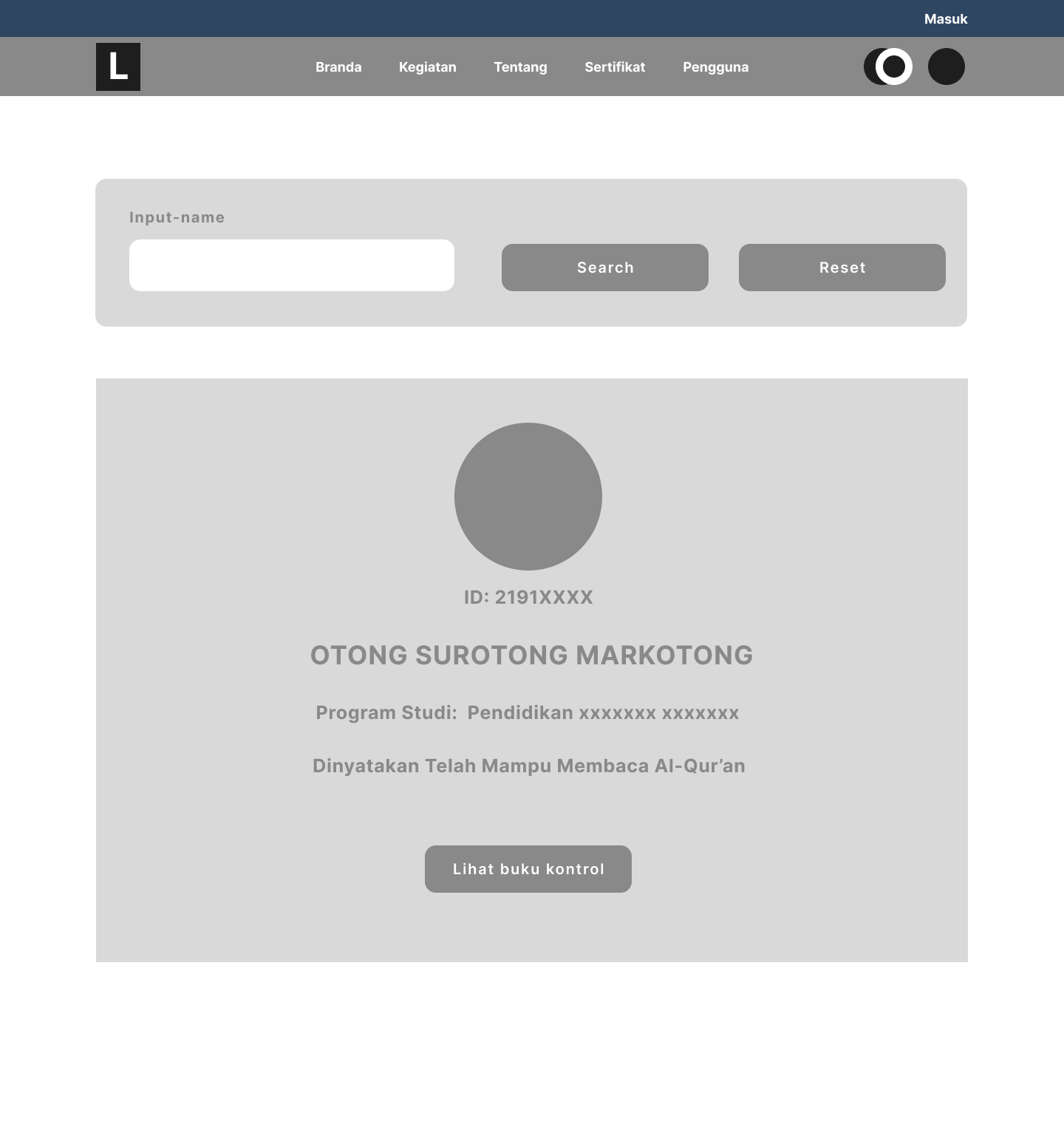
Gambar 3. *User Interface* Tamplian *Landing*



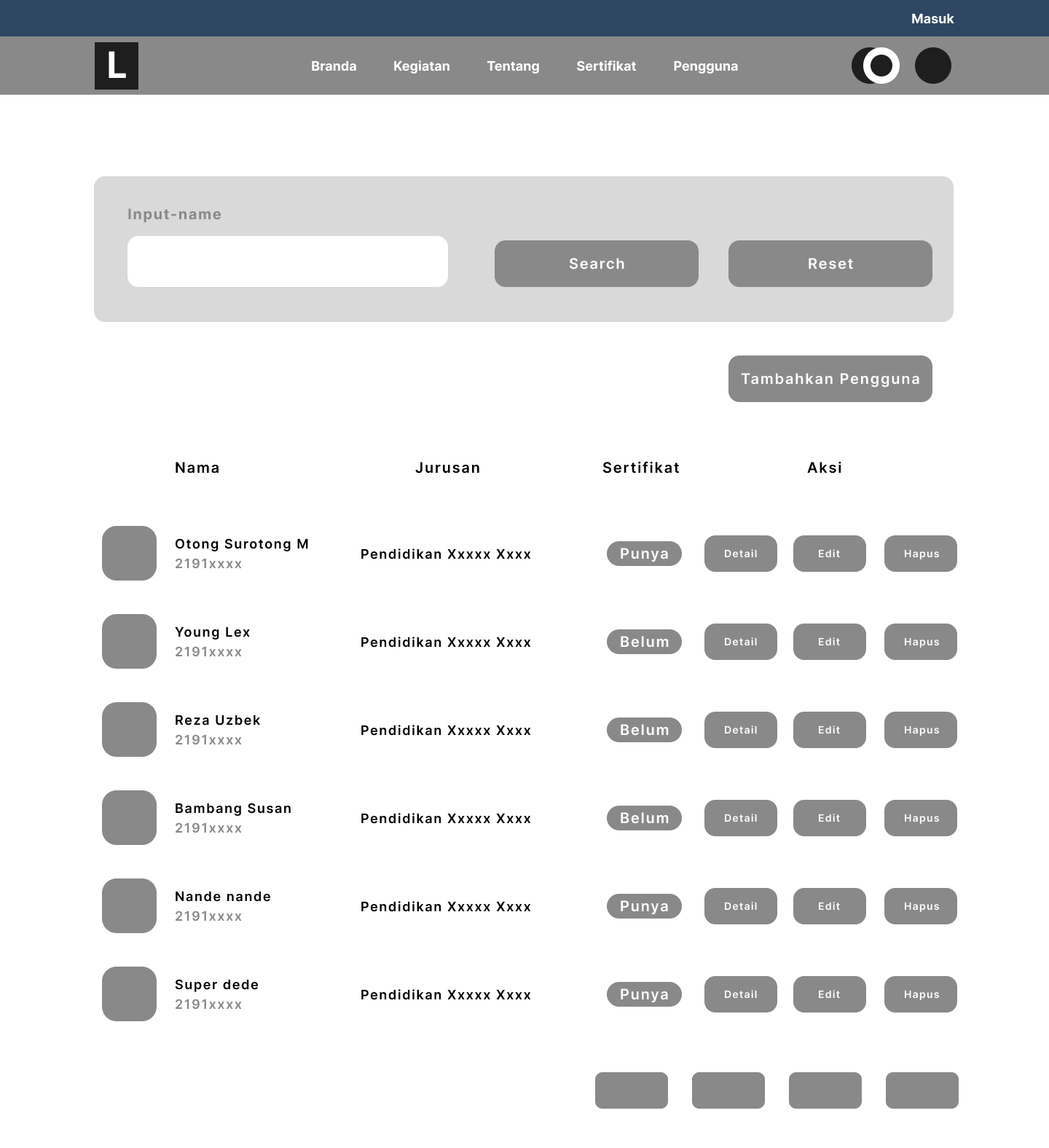
Gambar 3. User Interface Tampilan Activities



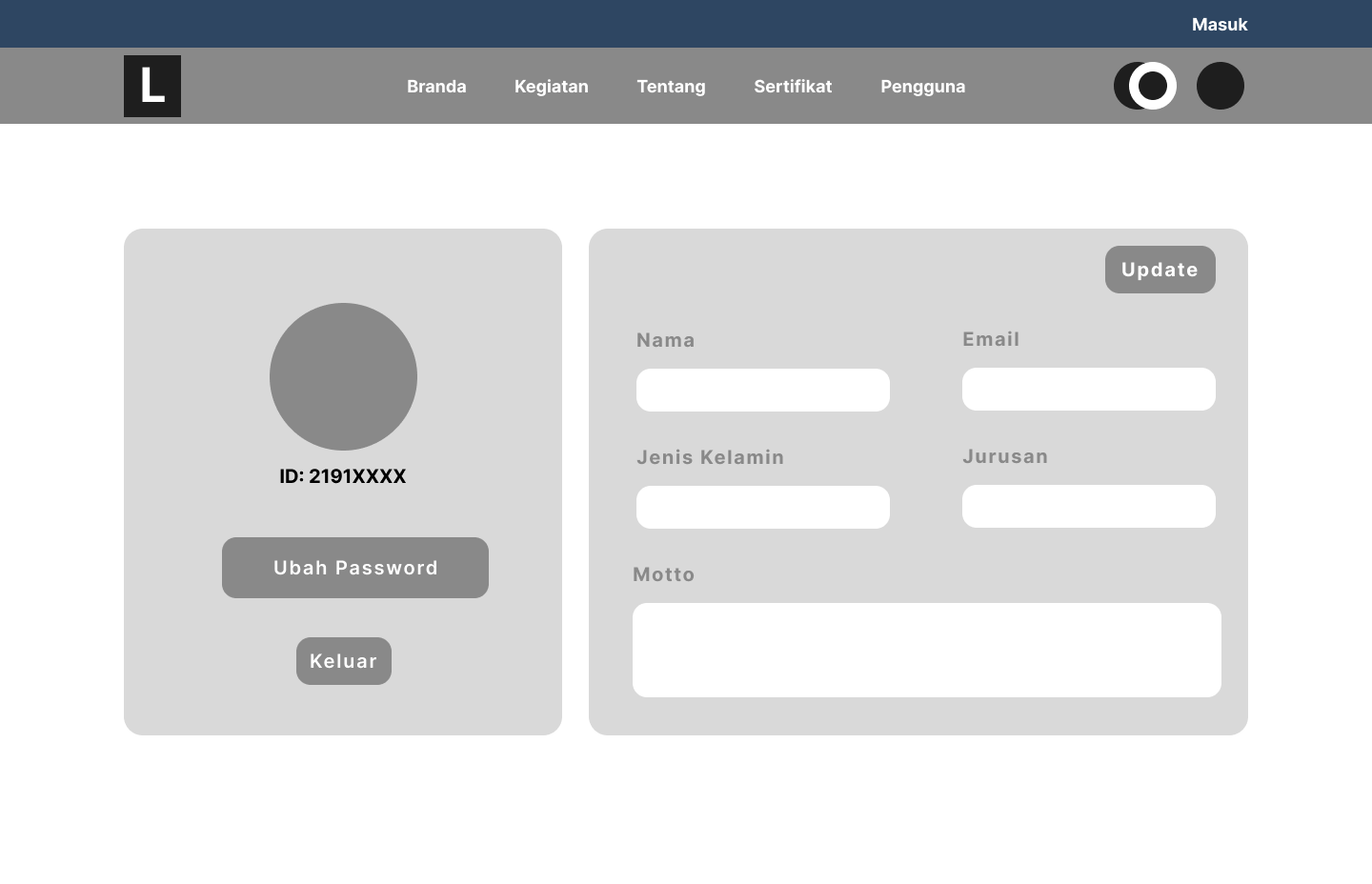
Gambar 3. *User Interface* Tampilan *Activity Detail*



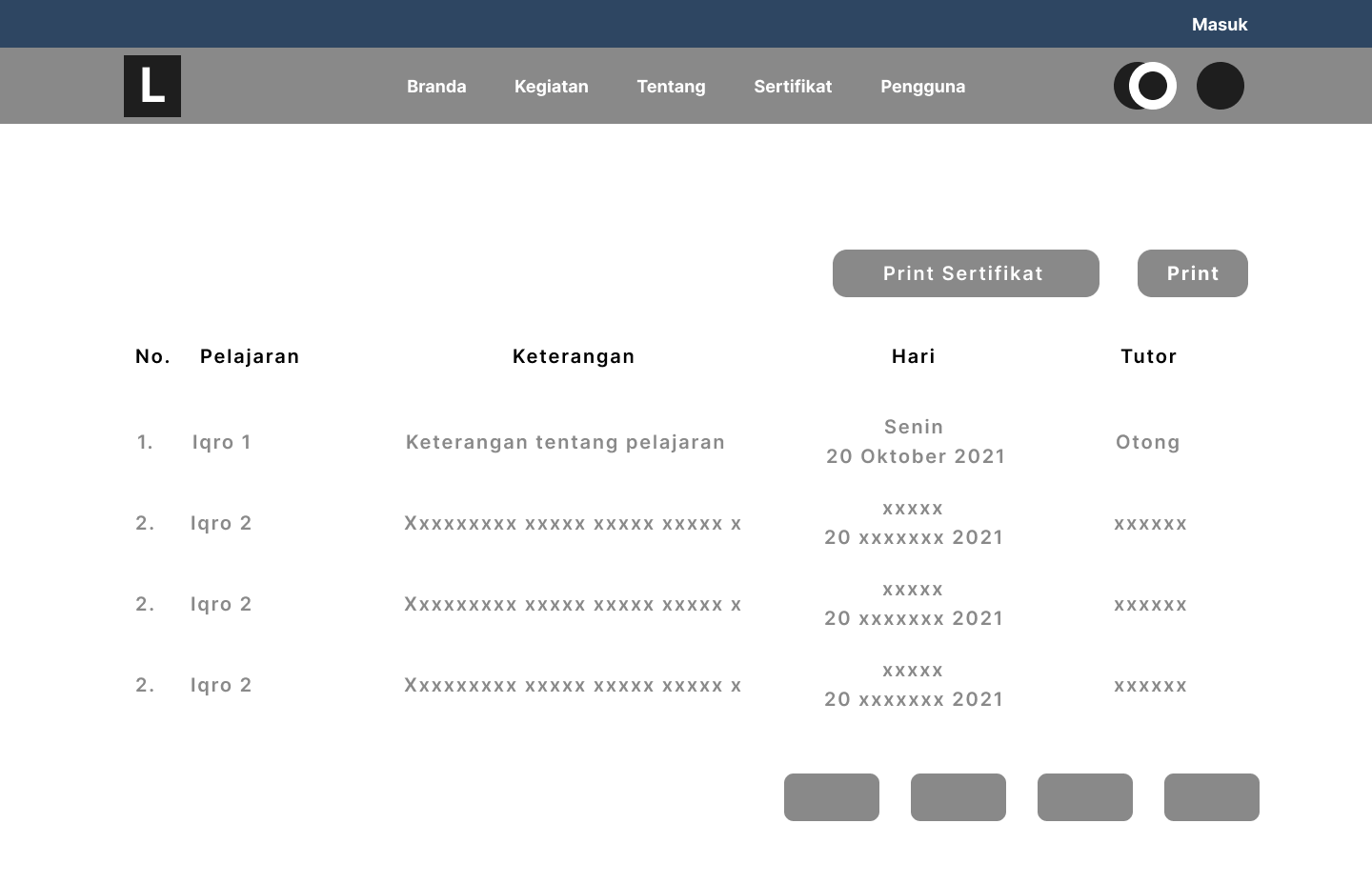
Gambar 3. *User Interface* Tampilan *Certificate*



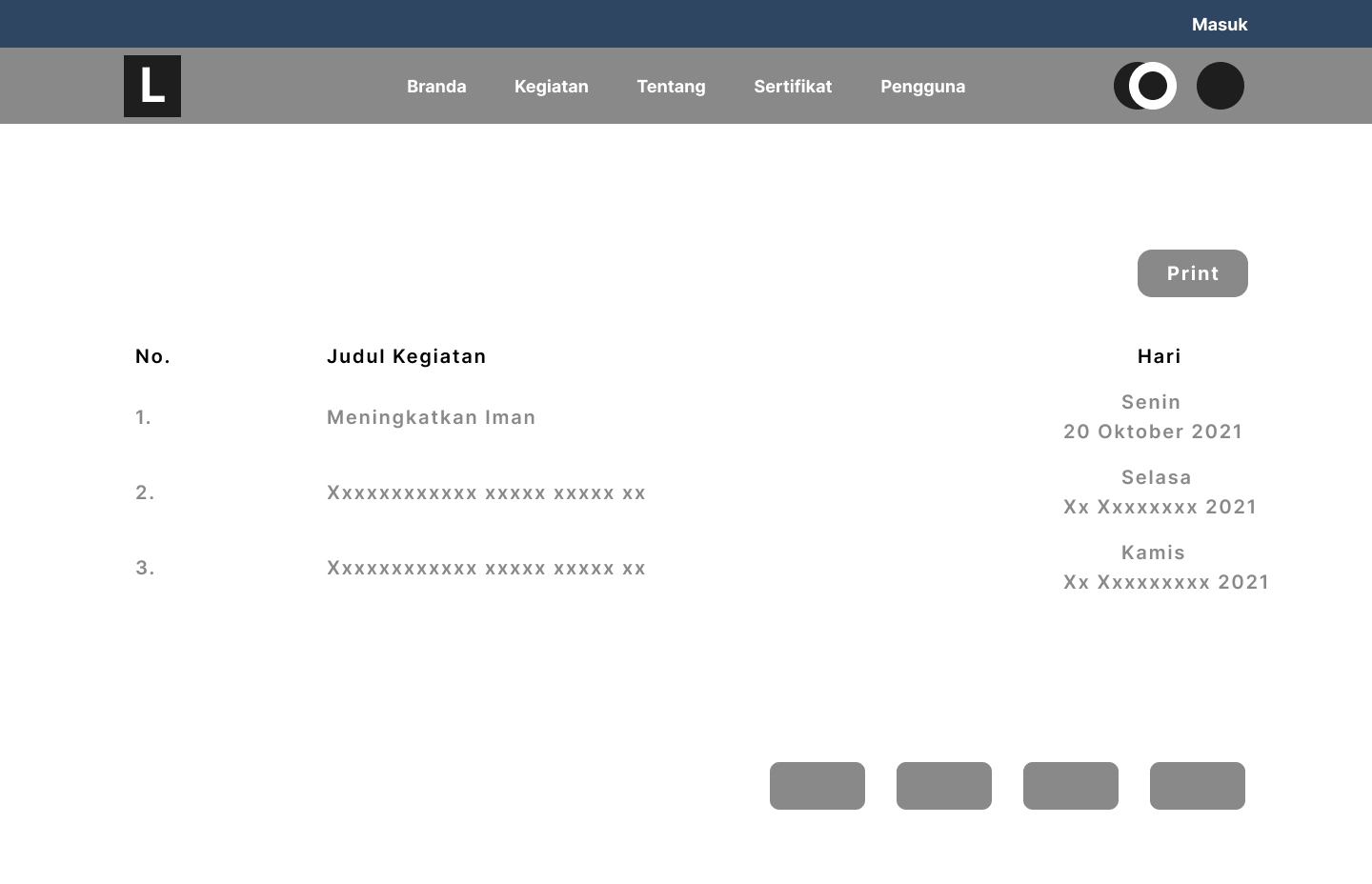
Gambar 3. *User Interface* Tampilan *Users*



Gambar 3. *User Interface* Tampilan *User Detail*



Gambar 3. *User Interface* Tampilan BTQ



Gambar 3. *User Interface* Tampilan *User Activities*

### Penulisan Kode (*Implementation*)

Proses implementasi merupakan tahap dimana seluruh desain ditransformasikan dalam bentuk kode-kode program. Kode program yang dihasilkan berupa bagian-bagian terpisah yang kemudian akan digabungkan menjadi satu kesatuan sistem yang lengkap(Findawati, 2018). Peneliti melakukan penyesuaian dengan desain dari produk yang akan dirancangkan, selanjutnya dilakukan ke tahap *development*. *Framework* yang diggunakan untuk membangun sistem informasi layanan LPPAIK ini menggunakan *Spring Boot* dan *ReactJs* untuk memudahkan dalam melakukan proses pemrograman dan implementasi desain produk. Berikut kode sederhana dari *Java* untuk menampilkan *text* pada *console*:

public class SpringApplication{

public static void main (String[] args ){

System.out.println(“Sistem Informasi Layanan LPPAIK”);}}

### Pengujian (*Testing*)

Proses melakukan pengujian pada penelitian ini adalah menggunakan pengujian *Black Box*. *Black Box* dipakai untuk menguji fungsionalitas spesifik dari *website* yang dibuat. Kevalidad *website* yang diujikan dapat ditinjau berdasarkan *outuput* yang diperoleh dari data-data atau kondisi masukan yang di inputkan terhadap fungsi-fungsi yang ada tanpa memperdulikan bagaimana proses untuk mendapatkan *output* tersebut. Berdasarkan *output* yang dihasilkan, kemampuan sistem dalam memenuhi kebutuhan penguna yang diukur sekaligus dapat diketahui permasalahanya.

Berikut merupakan kisi-kisi ujian *black box* dalam pengujian sistem berdasarkan perananaanya yang ditampilkan pada table 3.13 Kisi-kisi uji black box testing dibawah ini:

Tabel 3. Tabel Kisi Kisi Uji Black Box Testing

|  |  |  |  |
| --- | --- | --- | --- |
| **Aktivitas Pengujian** | **Relalisi yang diharapkan** | **Hasil Pengujian** | **Kesimpulan** |
| Masuk ke halaman login | Masukan *username* dan password |  |  |
| Masuk ke halaman register | Masukan data diri |  |  |
| Admin *klik* pada navigasi bar kajian | Menampilkan halaman kegiatan kajian/BTQ |  |  |
| Mahasiswa *klik* pada navigasi bar kegiatan | Menampilkan halaman kegiatan kajian/BTQ |  |  |
| Tutor sebaya *klik* pada navigasi bar kegiatan | Menampilkan halaman kegiatan kajian/BTQ |  |  |
| Mahasiswa *klik* pada avatar untuk melihat detail data diri | Menampilkan halaman detail mahasiswa/user |  |  |

### Pemeliharan (*Maintenance*)

Pemeliharaan termasuk dalam tahap melakukan perbaikan kesalahan yang tidak terdeteksi pada tahapan sebelumnya. Pemeliharaan juga adalah bagian evaluasi apakah sistem sudah siap dan sesuai dengan apa yang diharapkan pengembang. Apabila sudah memenui haparan, maka diteruskan ke tahap akhir yaitu pemakaian sistem.

## Tempat dan Waktu Penelitian

Penelitian ini dilaksanakan di Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan Universitas Muhammadiyah Kendari pada bulan April 2023 sampai dengan Juli 2023.

Adapun rencana kerja yang penulis buat untuk pembuatan sistem infromasi layanan LPPAIK ini dapat dilihat pada tabel berikut:

Tabel 3. Tabel Rencana Kerja Sistem­­­

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Fase** | **Uraian kegiatan** | **2023** | | | |
| **Agu** | **Spe** | **Okt** | **Nov** |
| Analysis | Wawancara sistem berjalan |  |  |  |  |
| Perancangan sistem ajuan |  |  |  |  |
| Design | Perancangan UML |  | ­­ |  |  |
| Perancangan ERD |  |  |  |  |
| Perancangan Mockup |  |  |  |  |
| Implementation | Coding |  |  |  |  |
| Testing | Proses Testing |  |  |  |  |
| Maintenance | Perawatan |  |  |  |  |

## Alat dan Bahan Penelitian

Penelitian ini membutuhkan beberapa *hardware* dan *software* dalam pengembangannya. Peralatan *hardware* maupun *software* dipilih dan disesuaikan dengan kebutuhan pengembangan *website* sebagai berikut:

### Alat penelitian

1. Hardware (Perangkat Keras):

|  |  |  |  |
| --- | --- | --- | --- |
| Laptop HP | Operating System | : | Linux Mint 21.1 Cinnamon |
| Cinnamon Version | : | 5.6.8 |
| Linux Karnel | : | 5.15.0-73-generic |
| Processor | : | Intel © Celeron © CPU N3060 @ 1.60GHz x 2 |
| Memory | : | 3.7 GiB |

1. Software (Perangkat Lunak)

|  |  |  |
| --- | --- | --- |
| InteliJ IDEA | : | Text editor untuk menulis *code* *Spring Boot* |
| VSCode | : | Text editor untuk menulis *code* *ReactJs* |
| DBaver | : | *Software* pengelolah *database* |
| Postman | : | *Software* API *testing* |
| Git | : | VCS (*Version Control System*) |
| Google Chrome | : | *Web browser* untuk menampilkan hasil UI |

### Bahan Penelitian

Bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. Observasi

Penulis melakukan pengamatan secara langsung ke lapangan dengan mengamati proses pelaksanaan kegiatan yang dilakukan Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan Universitas Muhammadiyah Kendari.

1. Dokumentasi

Dokumentasi merupakan tahapan untuk mengumpulkan data dengan menggunakan sumber-sumber tertulis seperti mengenai data dari tutor sebaya dan stackholder yang berkepentingan di Lembaga Pengkajian dan Penerapan Al-Islam & Kemuhammadiyahan Universitas Muhammadiyah Kendari.

1. Studi Literatur

Studi literatur dalam penelitian ini yaitu mengumpulkan data dari jurnal buku dan skripsi yang relevan guna menunjang peneliti membangun sebuah sistem infromasi lebih baik kedepannya.

# HASIL PENELITIAN DAN PEMBAHASAN

## Hasil Penelitian

### Implementasi Sistem Infromasi Layanan LPPAIK Universitas Muhammadiyah Kendari

Hasil dari penelitian ini adalah terciptanya Sistem Informasi Layanan LPPAIK berbasis *web* yang dibangun dengan menggunakan bahasa pemrogramman *Java* dan *JavaScript* sebagai bahasa utama. Dengan menggunakan *framework* *Spring Boot* dan *ReactJs* guna mempermudah pengambangan sistem informasi LPPAIK berbasis *web*. Sistem informasi yang dibangun ini, dikerjakan melalui *Visual Studio Code* dan *Intelij IDEA*. Data-data dari sistem informasi di simpan pada *database* *MySQL.* Seluruh *code* telah di *upload* pada *cloud provider* sehingga sistem informasi dapat diakses secara *online* melalui *web browser* pengguna.

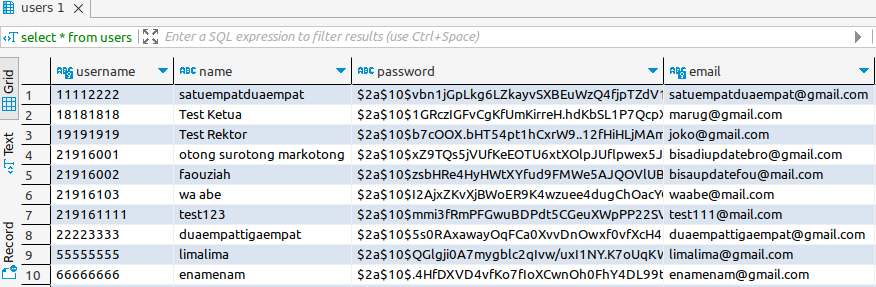
### Database

*Database* merupakan wadah atau tempat penyimpanan data yang terkomputerisasi yang tujuan utamanya untuk mememlihara informasi dan membuat informasi tersebut tersedia saat dibutuhkan dengan cepat. Pada pengembangan sistem ini menggunakan *MySQL* database yang dimana data-data tersimpan dalam bentuk *column and row.* Adapuan table yang digunakan adalah berjumlah 11 tabel. Untuk menjalankan database dari *cloud provider* di permudah dengan menggunakan *software* *DBaver* agar dapat diakses melalui komputer secara lokal.

Berikut tampilan *DBaver* beserta tabel yang digunakan:

1. Tampilan tabel *users*

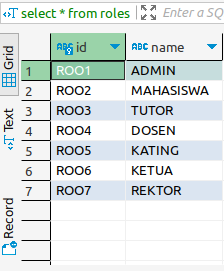
Tabel *users* ini digunakan untuk menampung data dari pengguna sistem baik itu admin, tutor, mahasiswa dan lain sebagainya.



Gambar 4. Tampilan Tabel *users*

1. Tampilan tabel *roles*

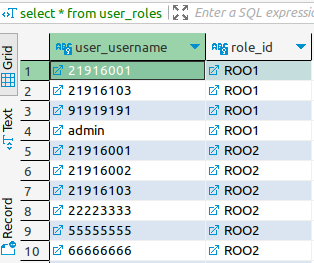
Tabel *roles* digunakan untuk menentukan level pengguna baik level admin, tutor, mahasiswa dan lain sebagainya.



Gambar 4. Tampilan Tabel *roles*

1. Tampilan tabel *user\_roles*

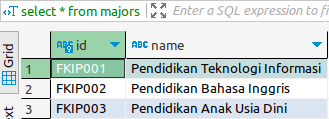
Tabel *user\_roles* ini digunakan untuk menampung relasi *many to many* antara *users* dan *roles*, yang dimana nantinya pengguna diperbolehkan memiliki banyak *roles*.



Gambar 4. Tampilan Tabel *user\_roles*

1. Tampilan tabel *majors*

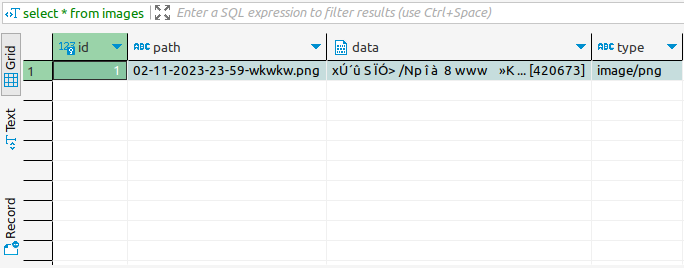
Tabel *majors*/jurusan ini digunakan untuk menampung seluruh jurusan yang ada di Universitas Muhammadiyah Kendari, yang berfungsi mengidentifikasi jurusan pengguna, dengan memiliki relasi *one to many* yang berarti satu *majors*/jurusan bisa dimiliki oleh banyak pengguna.



Gambar 4. Tampilan Tabel *majors*

1. Tampilan tabel *images*

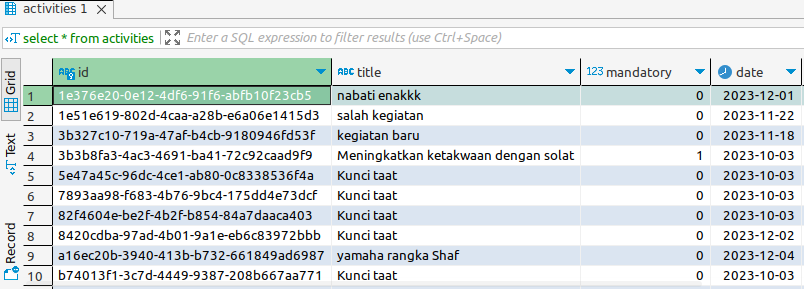
Tabel *images* berfungsi sebagai tabel yang menampung data berupa *file* gambar *avatar* pengguna.



Gambar 4. Tampilan Tabel *images*

1. Tampilan tabel *activities*

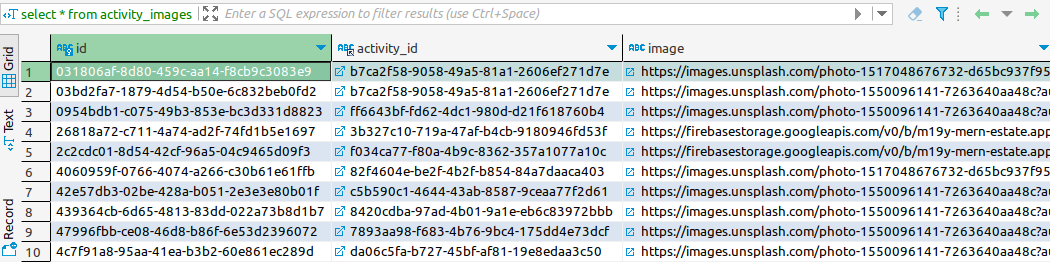
Tabel *activities* digunakan untuk menampung data-data yang berisikan informasi dari kegiatan yang dibuat oleh pengguna yang memiliki *role* level admin.



Gambar 4. Tampilan Tabel *activities*

1. Tampilan tabel *activity\_images*

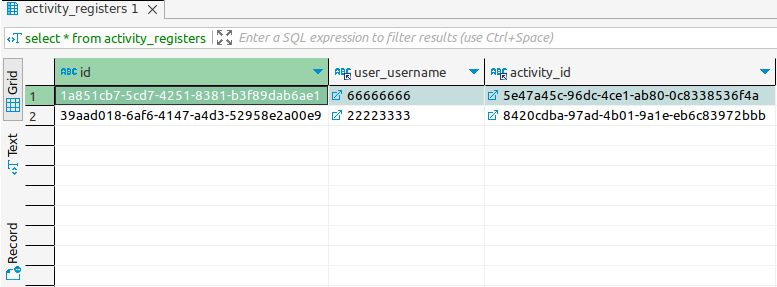
Tabel *activity\_images* berbeda dengan tabel *images*. Tabel ini di peruntukan untuk menampung data berupa *mediumtext* yang berisikan *URL* gambar yang telah di *upload* melalui *cloud provider*. Dengan memiliki relasi *many to one*, yang berarti banyak gambar bisa diperoleh oleh satu *activity*.



Gambar 4. Tampilan Tabel *activity\_images*

1. Tampilan tabel *activity\_registers*

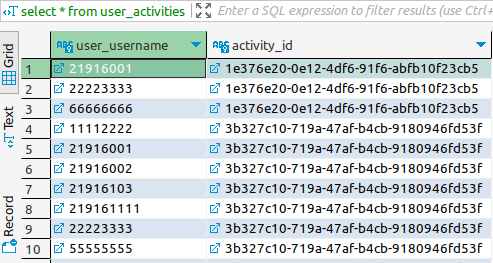
Tabel *activity\_registers* ini digunakan untuk menampung data pengguna yang ingin mengikuti kegiatan. Sebelum pengguna dipastikan hadir pada kegiatan yang telah dibuat, pengguna harus melakukan registrasi terlebih dahulu. Yang selanjutnya akan diteruskan ke tabel *user\_activities*.



Gambar 4. Tampilan Tabel *activity\_registers*

1. Tampilan tabel *user\_activities*

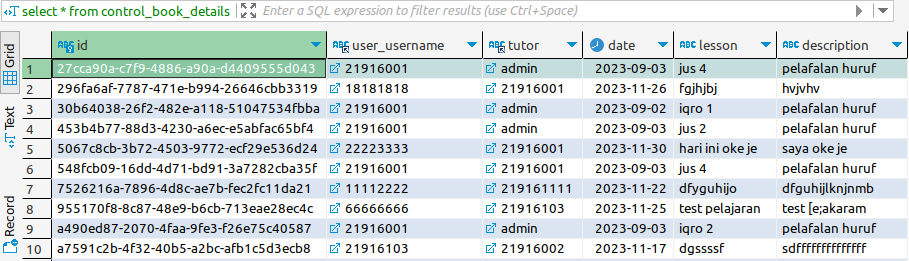
Tabel *users\_activities* ini digunakan untuk menampung data pengguna yang telah melakukan *registers* dan dinyatakan mengikuti kegiatan secara penuh. Dengan memiliki relasi *many to many* dengan *primary key* yang *unique*, yang berarti *users* diperbolehkan mengikuti kegiatan sebanyak mungkin namun tidak boleh mengikuti kegiatan dengan *id* yang sama dua kali. Selain itu juga tabel ini bisa dijadikan referensi pengguna sebagai informasi seberapa banyak kegiatan yang telah mereka ikuti.



Gambar 4. Tampilan Tabel *user\_activities*

1. Tampilan tabel *control\_book\_details*

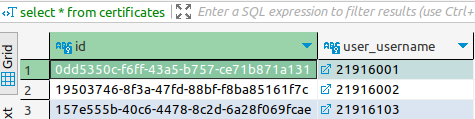
Tabel *control\_book\_details* ini digunakan untuk menampung data pelajaran BTQ (Baca Tulis Al-Qur’an) yang telah dilakukan pengguna dengan memiliki relasi *many to one relational*, yang berarti satu pengguna bisa memiliki banyak *control\_book\_details*. Selain itu juga tabel ini bisa dijadikan referensi pengguna sebagai informasi seberapa banyak pelajaran yang telah mereka lakukan.



Gambar 4. Tampilan Tabel *control\_book\_details*

1. Tampilan tabel *certificates*

Tabel *certificates* ini digunakan untuk menampung data pengguna yang telah memiliki sertifikat dengan memiliki relasi *one to one relational*, yang berarti satu pengguna hanya boleh memiliki satu *certificates*.



Gambar 4. Tampilan Tabel *certificates*

### Tampilan Sistem Informasi Layanan LPPAIK Universitas Muhammadiyah Kendari

1. Halaman Tanpa Login

Berikut adalah halaman – halaman yang terdapat pada sistem informasi LPPAIK yang dapat diakses oleh pengguna tanpa harus melakukan *login* terlebih dahulu.

1. MenuBeranda

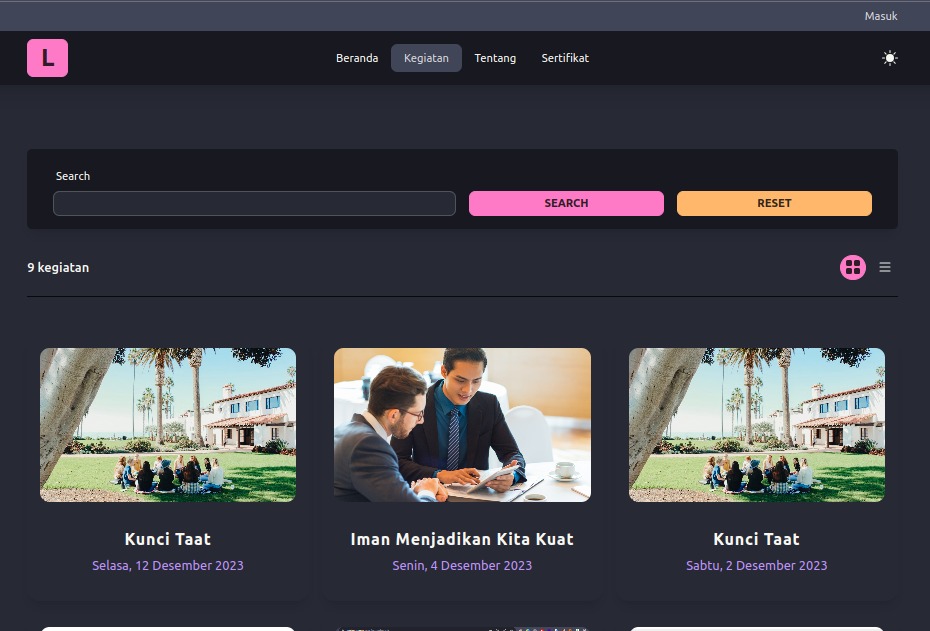
Menuberanda merupakan tampilan awal yang akan *users* lihat pertama kali ketika mengakses sistem informasi. Pada menu ini juga terdapat kegiatan-kegiatan wajib yang dilakukan lembaga.



Gambar 4. Tampilan Menu Beranda

1. MenuKegiatan

Menukegiatan ini akan menampilkan semua kegiatan-kegiatan yang ada di LPPAIK, sehingga pengguna yang belum ataupun sudah teregistrasi bisa mengetahui lebih lagi tentang kegiatan-kegiatan apa saja yang ada di LPPAIK dengan memiliki kemampuan pencarian guna mempermudah pengguna menemukan kegiatan.



Gambar 4. Tampilan Menu Kegiatan

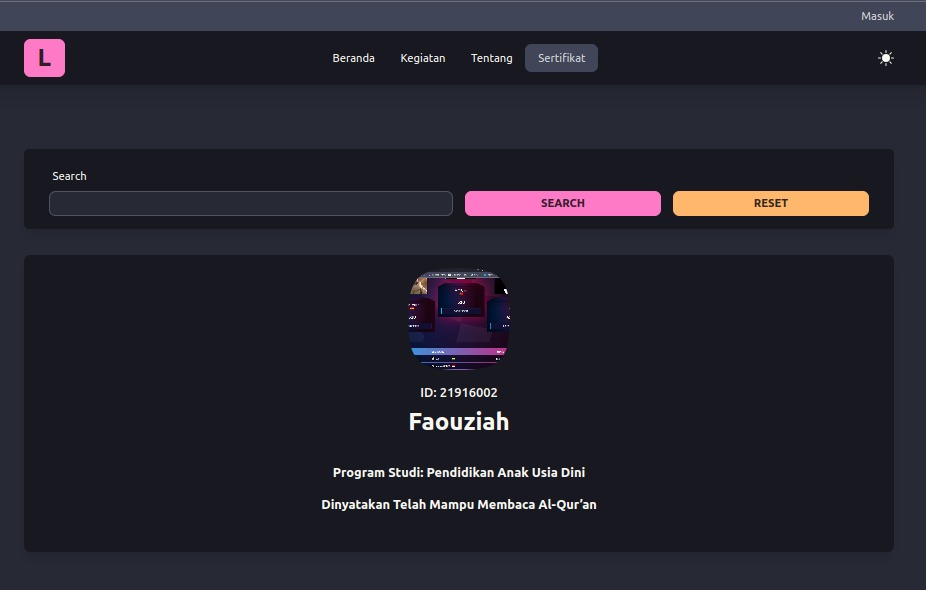
1. MenuTentang

Menutentang ini akan menampilkan hal-hal yang mendetail tentang LPPAIK Universitas Muhammadiyah Kendari sehingga pengguna yang belum ataupun sudah teregistrasi bisa mengetahui lebih lagi tentang LPPAIK.

Gambar 4. Tampilan Menu Tentang

1. Menu Sertifikat

Menusertifikat adalah tempat pengguna bisa melakukan validasi sertifikatnya ataupun sertifikat pengguna lainnya dengan cara memasukan nomor sertifikat pada *search* bar. Selain itu apa bila sertifikat sudah berbentuk kertas ataupun file dengan format pdf, pengguna bisa melakukan *scan QRcode* yang terdapat pada file yang nantinya akan diarahkan ke menu halaman sertifikat ini, sehingga pengguna dapat mengetahui keaslian dari sertifikat tersebut*.*

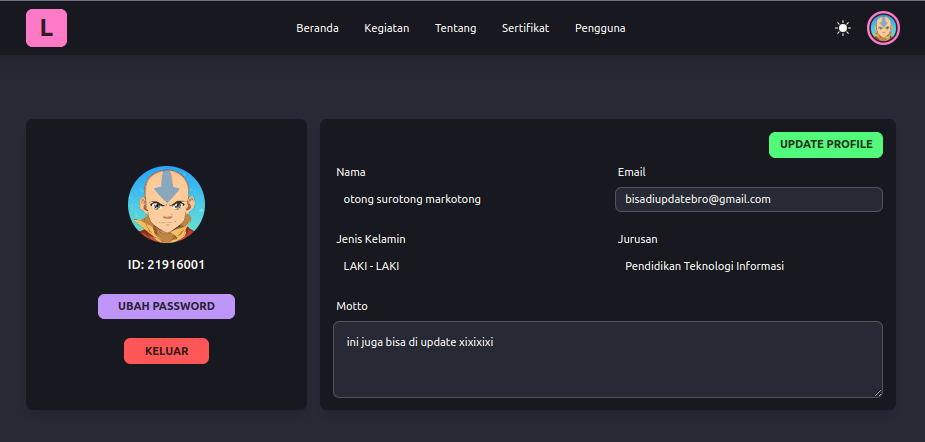


1. Halaman Users

Halaman *users* adalah halaman yang akan ditampilkan secara *general* pada setipa pengguna yang telah teregistrasi oleh sistem yang dapat diakses dengan mengklik avatar yang terdapat pada kanan atas menu bar. Berikut adalah sub menu yang akan ditampilkan setelah pengguna melakukan *login*.

1. *Profile*

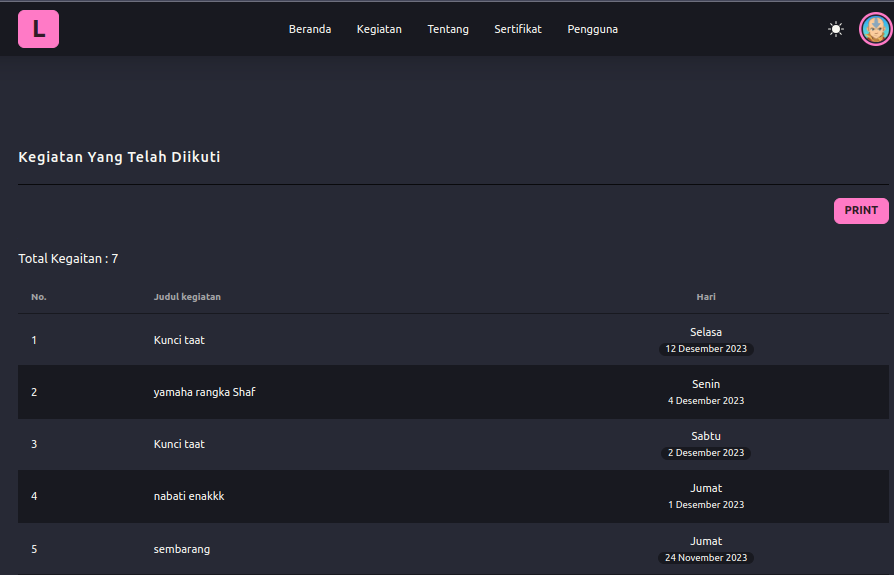
Pada menu halaman *profile* ini pengguna bisa melihat dan juga mengedit data diri mereka. Selain itu juga pengguna bisa melakukan perubahan pada *password,* dikarenakan pada saat pengguna diregistrasikan oleh admin secara otomatis *password* dari pengguna adalah sama dengan *username*-nya. Untuk mengubah *password,* pengguna dapat mengklik tombol ubah *password*.



Gambar 4. Tampilan *Profile User*

1. Kegiatan Ku

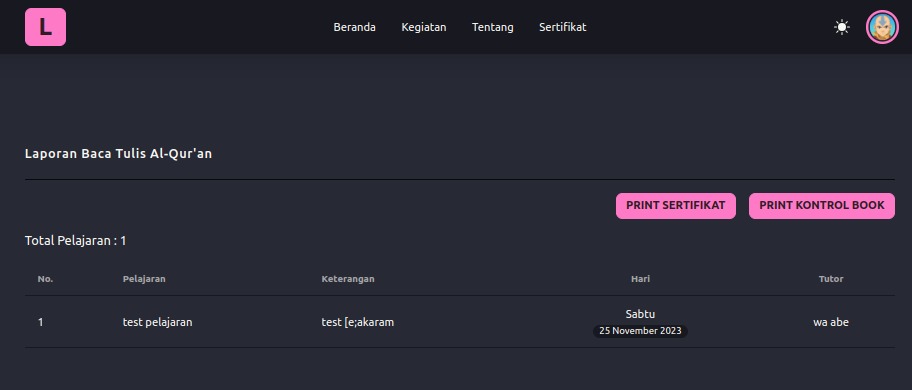
Setiap penggunadapat melihat kegiatan yang telah mereka ikuti. Pada menu kegiatan ku ini akan menampilkan list dari kegiatan-kegiatan yang pernah pengguna ikuti. Selain itu juga pengguna dapat melakukan *print* kegiatan dalam bentuk format *file pdf* dengan cara mengklik tombol *print* pada kanan atas list kegiatan, *file* hasil *download* akan menjadi bukti bahwa pengguna pernah mengikuti kegiatan yang telah dilaksanakan.



Gambar 4. Tampilan Menu Kegiatan Ku

1. Buku Kontrol

Sama halnya dengan menu kegiatan ku, menu buku kontrol ini akan menampilkan list dari pelajaran Baca Tulis Al-Qur’an (BTQ) yang telah *user* lakukan. Selain itu juga pengguna dapat melakukan *print* buku kontrol dengan format *file pdf*, *file* hasil *download* akan menjadi bukti bahwa pengguna (mahasiswa) pernah melakukan pembelajaran BTQ. Selain itu apabila pengguna (mahasiswa) telah dinyatakan lulus oleh pengguna (tutor) maka tombol *print* sertifikat akan ditampilkan, dan pengguna (mahasiswa) dapat melakukan *print* sertifikat dengan cara mengklik tombol *print* sertifikat.

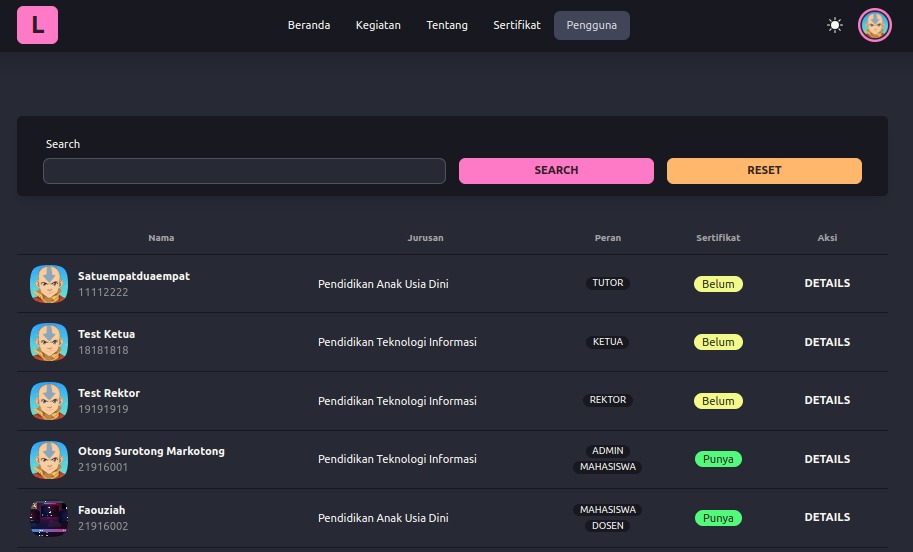


Gambar 4. Tampilan Menu Buku Kontrol

1. Halaman Khusus

Halaman khusus merupakan halaman-halaman dengan menu istimewa yang akan ditampilkan kepada pengguna yang memiliki role tertentu. Pengguna yang memiliki keistimewaan diantaranya pengguna yang memiliki *role admin*, *tutor*, kating dan dosen. Perbedaan di antara keempat *role* tersebut, yaitu ada pada tombol-tombol yang dapat mengoperasikan sistem informasi ini diantaranya sebagai berikut.

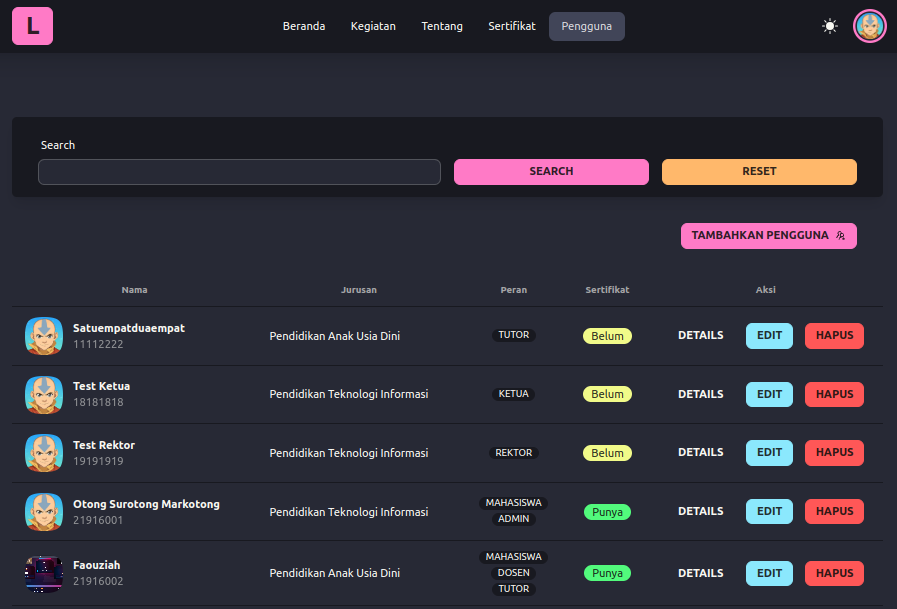
1. Menu Pengguna (dosen dan tutor)

Pada menu pengguna ini, pengguna dengan *role* dosen dan tutor dapat melakukan pencarian dan melihat detail dari pengguna lainya. Pencarian digunakan guna mempermudah dosen dan tutor mencari pengguna.

Gambar 4. Tampilan Menu Pengguna (dosen dan tutor)

1. Menu Pengguna (admin)

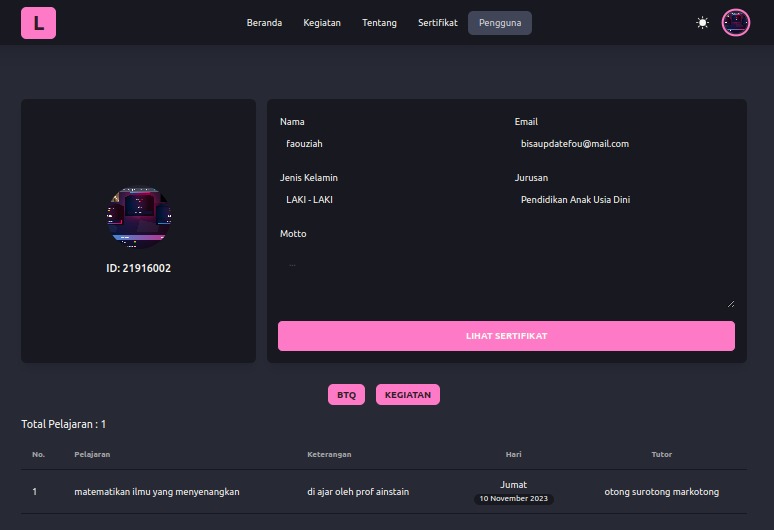
Pada menu pengguna ini, pengguna dengan *role* admin selain memiliki kemampuan melakukan pencarian dan melihat detail dari pengguna,pengguna dengan *role* admin juga dapat melakukan manipulasi terhadap pengguna lainya baik itu menambahkan, mengedit dan menghapus pengguna.

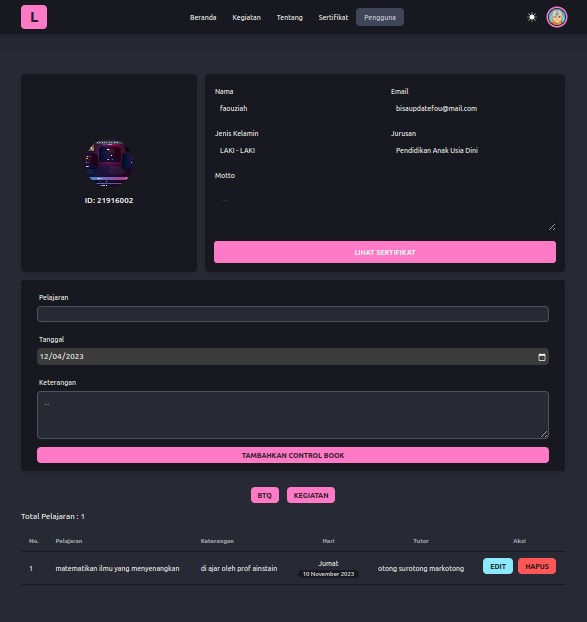


Gambar 4. Tampilan Menu Pengguna (admin)

1. Menu Detail Pengguna

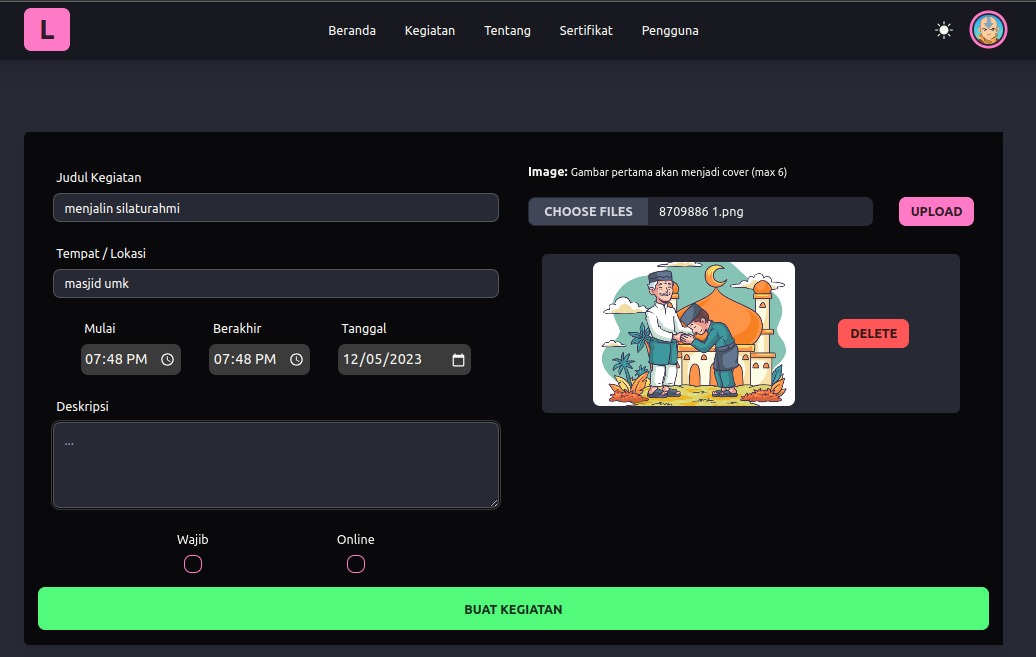
Menu detail pengguna akan menampilkan detail dari pengguna- pengguna yang ingin di lihat. Pengguna yang memiliki *role* dosen hanya bisa melihat data-data buku kontrol dan kegiatan yang telah dilakukan oleh pengguna tersebut. Sedangkan pengguna yang memiliki *role* admin atau tutor, selain dapat melihat detail pengguna-pengguna, pengguna dengan *role* ini juga dapat melakukan operasi menambahkan, mengedit dan menghapus buku kontrol pada pengguna tersebut, guna mempermudah tutor atau admin melakukan operasi manipulasi keterangan pembelajaran BTQ yang dilakukan pengguna.





1. Menu Kegiatan Setelah *Login*

Pada saat pengguna- pengguna *login* dengan *role* admin, menu kegiatan akan menampilkan tombol pada kanan bawah layar, sehingga admin bisa menambahkan kegiatan baru. Setelah mengklik tombol tersebut admin akan diminta untuk mengisi *form* yang tersedia.

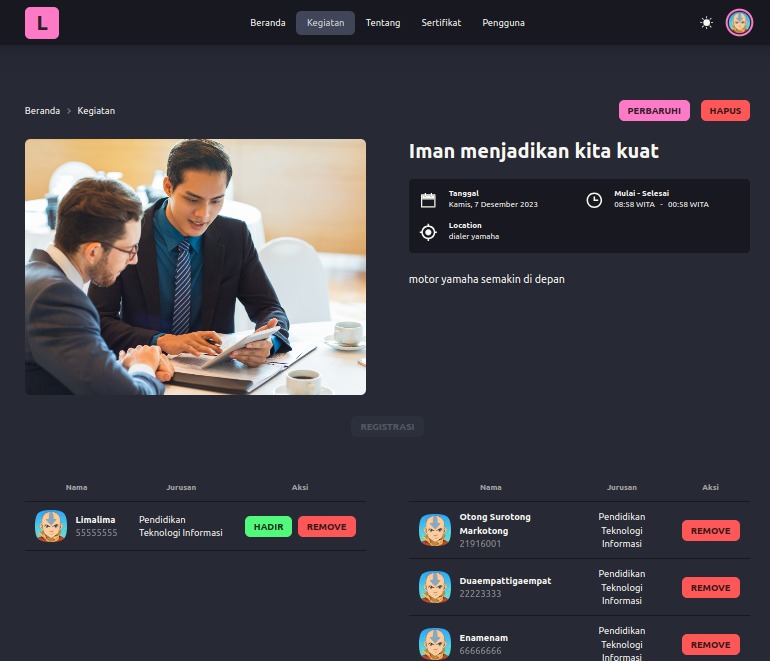


1. Menu Detail Kegiatan Setelah *Login*

Pada sub menu kegiatan ini seluruh pengguna dapat meregistrasikan diri, namun terdapat perbedaan pada tampilan-tampilan pada pengguna-pengguna dengan *role* tertentu. Pengguna dengan role dosen hanya memiliki kemampuan melihat pengguna-pengguna yang hadir dalam kegiatan ini. Sedangkan yang memiliki *role* admin dan kating dapat melihat dan melakukan absen kepada seluruh pengguna-pengguna yang telah melakukan registrasi. Selain itu juga admin memiliki kemampuan memanipulasi kegiatan, seperti mengabsen pengguna secara langsung tanpa pengguna tersebut melakukan registrasi, admin juga memiliki kemampuan untuk mengedit dan menghapus kegiatan.







## Pengujian Sistem

Pengujian sistem informasi layanan LPPAIK pada Universitas Muhammadiyah Kendari ini menggunakan pengujian *black box*. Pengujian sistem dilakukan pada setiap fitur yang di bangun, terdapat \_\_ fitur yang tersedia di dalam sistem informasi seperti fitur \_\_\_.

Berikut adalah tampilan table testing yang akan diggunakan:

1. Fitur *Login*

Tabel 4. Uji *Black Box Testing* Fitur *Login*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| *Login* | Menguji *Form Login* | Tidak mengisi *Username* dan *Password* | *Klik* Tombol *Login* | Sistem akan menolak dan menampilkan banyak pesan error | sesuai |  |
| Mengisikan *username*, dan tidak mengisi *password* | *Klik* Tombol *Login* | Sistem akan menolak dan menampilkan pesan "P*assword must not be blank*" | sesuai |  |
| Mengisikan *password*, dan tidak mengisi *username* | *Klik* Tombol *Login* | Sistem akan menolak dan menampilkan pesan "*Username must be not blank*" | sesuai |  |
| Mengisikan *username* yang tidak sesuai (salah), dan mengisi *password* | *Klik* Tombol *Login* | Sistem akan menolak dan menampilkan pesan "*This username maybe not exist*" | sesuai |  |
| Mengisikan password yang tidak sesuai (salah), dan mengisi username | *Klik* Tombol *Login* | Sistem akan menolak dan menampilkan pesan "*You send wrong password*" | sesuai |  |
| Mengisi username, pasword dengan benar | *Klik* Tombol *Login* | Sistem akan menerima dan kembali menampilkan halaman beranda | sesuai |  |

1. Fitur Beranda

Tabel 4. Uji *Black Box Testing Fitur* Beranda

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Beranda | Menguji menu Beranda | Melihat isi tampilan beranda | Klik menu Beranda | Sistem menampilkan halaman utama dan menampilkan halaman kegiatan wajib | sesuai |  |
| Menguji tombol Lihat Kegiatan | Menuju halaman kegiatan | Klik Tombol Lihat Kegiatan | Sistem menampilkan halaman berisikan kegiatan-kegiatan yang ada | sesuai |  |
| Menguji tombol masuk | Menuju halaman login | Klik Tombol Masuk | Sistem menampilkan halaman *login* | sesuai |  |
| Menguji *card* kegiatan wajib | Menuju halaman detail kegiatan wajib | Klik Card Kegiatan | Sistem menampilkan halaman detail kegiatan sesuai dengan card kegiatan yang dipilih | sesuai |  |

1. Fitur Kegiatan

Tabel 4. Uji *Black Box Testing* Fitur Kegiatan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Kegiatan | Menguji menu Kegiatan | Melihat isi tampilan kegiatan | Klik menu Kegiatan | Sistem menampilkan halaman kegiatan berserta menu *search* | sesuai |  |
| Menguji menu *Search* | Mengisi *form search* dengan nama kegiatan(ada) yang ingin di cari | Klik tombol *Search* | Sistem akan menampilkan kegiatan yang dicari | sesuai |  |
| Mengisi *form search* dengan nama kegiatan (tidak ada) yang ingin di cari | Klik tombol *Search* | Sistem akan menampilkan pesan “*Sorry, no activity matched your search*” | sesuai |  |
| Melakukan reset *search* | Klik tombol Reset | Sistem akan menampilkan seluruh kegiatan seperti semula | sesuai |  |
| Menguji tampilan kegiatan berbentuk *list* | Melihat tampilan kegiatan berbentuk *list* | Klik tombol dengan ikon *list* | Sistem akan menampilkan seluruh kegiatan berbentuk *list* | sesuai |  |
| Menguji tampilan kegiatan berbentuk *card* | Melihat tampilan kegiatan berbentuk *list* | Klik tombol dengan ikon *card* | Sistem akan menampilkan seluruh kegiatan berbentuk *card* | sesuai |  |
| Menguji *list* kegiatan | Menuju halaman detail kegiatan | Klik list Kegiatan | Sistem menampilkan halaman detail kegiatan sesuai dengan list kegiatan yang dipilih | sesuai |  |
| Menguji *card* kegiatan wajib | Menuju halaman detail kegiatan wajib | Klik card Kegiatan | Sistem menampilkan halaman detail kegiatan sesuai dengan card kegiatan yang dipilih | sesuai |  |
| Menguji menu tambahkan kegiatan | Menuju halaman tambahkan kegiatan | Klik tombol dengan ikon plus | Sistem menampilkan halaman tambah kegiatan | sesuai |  |
| Menguji menu *paging* | Menuju halaman kegiatan sebelumnya | Klik tombol *Prev* | Sistem menampilkan halaman kegiatan pada *page* sebelumnya | sesuai |  |
| Menuju halaman kegiatan selanjutnya | Klik tombol *Next* | Sistem menampilkan halaman kegiatan pada *page* selanjutnya | sesuai |  |
| Menuju halaman kegaitan sesuai dengan nomor *page* | Klik tombol nomor (page) | Sistem menampilkan halaman kegiatan dengan *page* nomor pilihan | sesuai |  |

1. Fitur Tambah Kegiatan

Tabel 4. Uji Black Box Testing Fitur Tambah Kegiatan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Tambah kegiatan | Menguji *form* | Tidak menambahkan gambar | Klik tombol Buat Kegiatan | Sistem akan mendisable tombol Buat Kegiatan |  |  |
| Menambahkan gambar | Klik tombol *Upload* | Sistem menampilkan gambar pada *list* gambar | sesuai |  |
| Menambahkan gambar tidak mengisi form | Klik tombol Buat Kegiatan | Sistem menampilkan 3 pesan *error* | sesuai |  |
| Menambahkan gambar, mengisi judul kegiatan, tidak mengisi tempat, dan tidak mengisi deskripsi | Klik tombol Buat Kegiatan | Sistem menampilkan 2 pesan *error* | sesuai |  |
| Menambahkan gambar, mengisi judul kegiatan, mengisi tempat, dan tidak mengisi deskripsi | Klik tombol Buat Kegiatan | Sistem menampilkan 1 pesan *error* | sesuai |  |
| Menambahkan gambar dan mengisi form | Klik tombol Buat Kegiatan | Sistem menampilkan pesan “*Success create new activity*” dan melakukan *redirect* ke halaman kegiatan | sesuai |  |
| Menguji hapus gambar | Menghapus gambar | Klik tombol  *Delet*e | Sistem akan menghapus gambar dari list gambar kegiatan | sesuai |  |

1. Fitur detail kegiatan

Tabel 4. Uji *Black Box Testing* Fitur Detail Kegiatan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Detail kegiatan | Menguji tombol perbarui | Menuju halaman perbarui kegiatan | Klik tombol Perbarui | Sistem menampilkan halaman perbarui kegiatan | sesuai |  |
| Menguji tombol hapus kegiatan | Menghapus kegiatan | Klik tombol  Hapus | Sistem akan menghapus kegiatan dengan pesan “*Activity has been deleted*” | sesuai |  |
| Menguji tombol registrasi | Waktu kegiatan telah selesai | Klik tombol Registrasi | Sistem akan mendisable tombol Register | sesuai |  |
| *User* melakukan registrasi | Klik tombol Registrasi | Sistem akan menampilkan pesan “*Registerd has been successfully*” | sesuai |  |
| User sudah melakukan registrasi | Klik tombol Registrasi | Sistem akan mendisable tombol Register | sesuai |  |
| Menguji tombol hadir | Admin atau kating mengabsen user | Klik tombol Hadir | Sistem akan menampilkan pesan “Success add activity for user with id ...” | sesuai |  |
| Menguji tombol *remove* pada daftar list *user* registrasi kegiatan | Admin atau kating mentidak hadirkan user | Klik tombol Remove | Sistem akan menampilkan pesan “*User has been successfully removed*” dan user di hilangkan dari list | sesuai |  |
| Menguji tombol *remove* pada daftar list hadirkegiatan | Admin atau kating mentidak hadirkan user | Klik tombol Remove | Sistem akan menampilkan pesan “*User ... has been removed*” dan user di hilangkan dari list | sesuai |  |

1. Fitur perbarui kegiatan

Tabel 4. Uji *Black Box Testing* Fitur Perbarui Kegiatan

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Perbarui kegiatan | Menguji *form* | Tidak menambahkan gambar | Klik tombol *Update* Kegiatan | Sistem akan mendisable tombol *Update* Kegiatan |  |  |
| Menambahkan gambar | Klik tombol *Upload* | Sistem menampilkan gambar pada *list* gambar | sesuai |  |
| Menambahkan gambar tidak mengisi form | Klik tombol *Update* Kegiatan | Sistem menampilkan 3 pesan *error* | sesuai |  |
| Menambahkan gambar, mengisi judul kegiatan, tidak mengisi tempat, dan tidak mengisi deskripsi | Klik tombol *Update* Kegiatan | Sistem menampilkan 2 pesan *error* | sesuai |  |
| Menambahkan gambar, mengisi judul kegiatan, mengisi tempat, dan tidak mengisi deskripsi | Klik tombol U*pdate* Kegiatan | Sistem menampilkan 1 pesan *error* | sesuai |  |
| Menambahkan gambar dan mengisi *form* | Klik tombol *Update* Kegiatan | Sistem menampilkan pesan “*Activity has been updated*” dan melakukan *redirect* ke halaman kegiatan | sesuai |  |
| Menguji hapus gambar | Menghapus gambar | Klik tombol  *Delet*e | Sistem akan menghapus gambar dari list gambar kegiatan | sesuai |  |

1. Fitur Tentang

Tabel 4. Uji *Black Box Testing* Fitur Tentang

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Tantang | Menguji menu kegiatan | Melihat tampilan tentang lembaga | Klik menu Tentang | Sistem menampilkan halaman tentang | sesuai |  |

1. Fitur Sertifikat

Tabel 4. *Uji Black Box Testing* Fitur Sertifikat

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Sertifikat | Menguji menu sertifikat | Melihat tampilan sertifikat | Klik menu Sertifikat | Sistem menampilkan halaman sertifikat beserta menu *search* | sesuai |  |
| Menguji menu *Search* | Mengisi *form search* dengan nomor sertifikat (ada) yang ingin di cari | Klik tombol *Search* | Sistem akan menampilkan sertifikat *user* yang dicari | sesuai |  |
| Mengisi *form search* dengan nomor sertifkat (tidak ada) yang ingin di cari | Klik tombol *Search* | Sistem akan menampilkan pesan “*Certificate is not found*” | sesuai |  |
| Melakukan reset *search* | Klik tombol Reset | Sistem menampilkan halaman sertifikat beserta menu *search* tanpa menampilkan sertifikat *user* | sesuai |  |
| Menguji tombol lihat buku kontrol | Melihat tampilan buku kontrol | Klik tombol Lihat Buku Kontrol BTQ | Sistem menampilkan halaman detail *user* beserta buku kontrolnya | sesuai |  |

1. Fitur Pengguna

Tabel 4. Uji *Black Box Testing* Fitur Pengguna

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| pengguna | Menguji menu pengguna | Melihat tampilan pengguna | Klik menu Pengguna | Sistem menampilkan halaman pengguna beserta menu *search* | sesuai |  |
| Menguji menu *Search* | Mengisi *form search* dengan nama atau id (ada) yang ingin di cari | Klik tombol *Search* | Sistem akan menampilkan *user* yang dicari | sesuai |  |
| Mengisi *form search* dengan nama atau id (tidak ada) yang ingin di cari | Klik tombol *Search* | Sistem akan menampilkan pesan “Kami tidak menemukan hasil pencarian anda” | sesuai |  |
| Melakukan reset *search* | Klik tombol Reset | Sistem menampilkan halaman pengguna beserta menu *search* seperti semula | sesuai |  |
| Menguji tombol tambah pengguna | Melihat tampilan tambah pengguna | Klik tombol Tambah Pengguna | Sistem menampilkan halaman tambah pengguna | sesuai |  |
| Menguji tombol detail | Melihat tampilan detail pengguna | Klik tombol Detail | Sistem menampilkan halaman detail pengguna | sesuai |  |
| Menguji tombol edit | Melihat tampilan edit pengguna | Klik tombol Edit | Sistem menampilkan halaman edit pengguna | sesuai |  |
| Menguji tombol hapus | Menghapus pengguna | Klik tombol Hapus | Sistem menampilkan pesan “*User deleted successfuly*” | sesuai |  |
| Menguji menu *paging* | Menuju halaman pengguna sebelumnya | Klik tombol *Prev* | Sistem menampilkan halaman pengguna pada *page* sebelumnya | sesuai |  |
| Menuju halaman pengguna selanjutnya | Klik tombol *Next* | Sistem menampilkan halaman pengguna pada *page* selanjutnya | sesuai |  |
| Menuju halaman pengguna sesuai dengan nomor *page* | Klik tombol nomor (page) | Sistem menampilkan halaman pengguna dengan *page* nomor pilihan | sesuai |  |

1. Fitur Tambah Pengguna

Tabel 4. Uji *Black Box Testing* Fitur Tambah Pengguna

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Tambah pengguna | Menguji form tambah pengguna | Tidak mengisi form satupun | Klik tombol Tambahkan | Sistem menampilkan pesan *error* pada username, name, email | sesuai |  |
| Mengisi semua *form* kecuali *username* | Klik tombol Tambahkan | Sistem menampilkan pesan *username error* | sesuai |  |
| Mengisi semua *form* kecuali *name* | Klik tombol Tambahkan | Sistem menampilkan pesan name error | sesuai |  |
| Mengisi semua *form* kecuali email | Klik tombol Tambahkan | Sistem menampilkan pesan email error | sesuai |  |
| Mengisi semua *form* *tanpa memilih satu role pun* | Klik tombol Tambahkan | Sistem menampilkan pesan role error | sesuai |  |
| Mengisi semua from | Klik tombol Tambahkan | Sistem menampilkan pesan “sucess create new user” dan melakukan *redirect* ke halaman pengguna | sesuai |  |

1. Fitur Edit Pengguna

Tabel 4. Uji *Black Box Testing* Fitur Edit Pengguna

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Tambah pengguna | Menguji form tambah pengguna | Tidak mengisi form satupun | Klik tombol *Update* | Sistem menampilkan pesan *error* pada username, name, email | sesuai |  |
| Mengisi semua *form* kecuali *name* | Klik tombol *Update* | Sistem menampilkan pesan name error | sesuai |  |
| Mengisi semua *form* kecuali email | Klik tombol *Update* | Sistem menampilkan pesan email error | sesuai |  |
| Mengisi semua *form* tanpa memilih satu *role* pun | Klik tombol *Update* | Sistem menampilkan pesan role error | sesuai |  |
| Mengisi semua from | Klik tombol *Update* | Sistem menampilkan pesan “*user has been updated*” dan melakukan *redirect* ke halaman pengguna | sesuai |  |

1. Fitur Detail Pengguna

Tabel 4. Uji *Black Box Testing* Fitur Detail Pengguna

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Detail Pengguna | Menguji tombol sertifkat | Melihat sertifikat (ada) | Klik tombol Sertifikat | Sistem mengarahkan ke halaman sertifikat | sesuai |  |
| Melihat sertifkat (tidak ada) | Klik tombol Sertifikat | Sistem mendisabel tombol Sertifikat | sesuai |  |
| Menguji menu tambahkan kontrol *book* | Tidak mengisi form kontrol *book* | Klik tombol Tambahkan *Control Book* | Sistem menampilkan pesan *error* untuk mengisi form | sesuai |  |
| Menguji menu tambahkan kontrol *book* | Mengisi form pelajaran dan tidak mengisi keterangan | Klik tombol Tambahkan *Control Book* | Sistem menampilkan pesan *error* “description: length must at least 5 characters” | sesuai |  |
| Menguji menu tambahkan kontrol *book* | Mengisi form pelajaran dan tidak mengisi keterangan | Klik tombol Tambahkan *Control Book* | Sistem menampilkan pesan *error* “description: length must between 5 – 255 characters” | sesuai |  |
| Menguji menu tambahkan kontrol *book* | Mengisi semua form pelajaran BTQ | Klik tombol Tambahkan *Control Book* | Sistem menampilkan pesan *error* “*Success add new lesson for user*” | sesuai |  |
| Menguji tombol BTQ | Melihat list pelajaran BTQ | Klik tombol BTQ | Sistem menampilkan list pelajaran BTQ | sesuai |  |
| Menguji tombol kegaitan | Melihat list kegiatan | Klik tombol Kegiatan | Sistem menampilkan list kegiatan yang diikuti | sesuai |  |
| Menguji tombol edit (BTQ) | Melihat menu edit pelajaran | Klik tombol Edit | Sistem menampilkan halaman edit pelajaran | sesuai |  |
| Menguji tombol hapus (BTQ) | Menghapus pelajaran | Klik tombol Hapus | Sistem menampilkan pesan “*lesson has been deleted*” | sesuai |  |
| Menguji menu *paging* (BTQ) | Menuju halaman pelajaran sebelumnya | Klik tombol *Prev* | Sistem menampilkan halaman kegiatanpada *page* sebelumnya | sesuai |  |
| Menuju halaman pelajaran selanjutnya | Klik tombol *Next* | Sistem menampilkan halaman pelajaran pada *page* selanjutnya | sesuai |  |
| Menuju halaman pelajaran sesuai dengan nomor *page* | Klik tombol nomor (page) | Sistem menampilkan halaman pelajaran dengan *page* nomor pilihan | sesuai |  |
| Menguji menu *paging* (Kegiatan) | Menuju halaman kegiatan sebelumnya | Klik tombol *Prev* | Sistem menampilkan halaman kegiatanpada *page* sebelumnya | sesuai |  |
| Menuju halaman kegiatan selanjutnya | Klik tombol *Next* | Sistem menampilkan halaman kegiatan pada *page* selanjutnya | sesuai |  |
| Menuju halaman kegiatan sesuai dengan nomor *page* | Klik tombol nomor (page) | Sistem menampilkan halaman kegiatan dengan *page* nomor pilihan | sesuai |  |

1. Fitur Akun

Tabel 4. Uji *Black Box Testing* Fitur Akun

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Akun | Menguji menu akun | Melihat sub menu akun | Klik avatar | Sistem menampilkan list sub menu (lihat profile, kegaitan ku, buku kontrol dan keluar) | sesuai |  |
| Menutup sub menu akun | Klik dimanapun (kecuali avatar) | Sistem menyembunyikan sub menu akun | sesuai |  |
| Menguji sub menu lihat *profile* | Melihat tampilan detail *user* | Klik menu Lihat Profile | Sistem menampilkan halaman detail *user* | sesuai |  |
| Menguji sub menu kegiatan ku | Melihat tampilan detail kegiatan *user* | Klik menu Kegiatan Ku | Sistem menampilkan halaman detail kegiatan *user* | sesuai |  |
| Menguji sub menu buku kontrol | Melihat tampilan detail pelajaran BTQ *user* | Klik menu Buku Kontrol | Sistem menampilkan halaman detail pelajaran BTQ *user* | sesuai |  |
| Menguji sub menu keluar | Keluar dari akun | Klik menu Lihat Keluar | Sistem menampilkan pesan “*logged out successfully*” | sesuai |  |

1. Fitur Lihat Profile

Tabel 4. Uji *Black Box Testing* Fitur Lihat *Profile*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Lihat profile | Menguji tombol ubah *password* | Melihat tampilan ubah *password* | Klik tombol *Ubah Password* | Sistem menampilkan halaman ubah password | sesuai |  |
| Menguji tombol keluar | Keluar dari akun | Klik tombol Keluar | Sistem menampilkan pesan “*logged out successfully*” | sesuai |  |
| Menguji tombol *update* | Mengisi *form* email dengan email yang telah digunakan user lain | Klik menu U*pdate Profile* | Sistem menampilkan pesan ‘Email *already taken*” | Tidak sesuai |  |
| Mengisi *form* email dengan email baru | Klik menu U*pdate Profile* | Sistem menampilkan pesan ‘*Detail has been updated*” | sesuai |  |
| Mengisi *form* motto | Klik menu *Update Profile* | Sistem menampilkan pesan “*Detail has been updated*” | sesuai |  |

1. Fitur Update Password

Tabel 4. Uji *Black Box Testing* Fitur *Update Password*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| *Update password* | Menguji menu *update password* | Tidak mengisi salah satu form atau dan *new password* tidak sama dengan *confirm password* | Klik tombol U*pdate* | Sistem menampilkan pesan ‘password and confirm password not match” | sesuai |  |
| Mengisi semua *form* dan *new password* sama dengan *confirm password* | Klik tombol Update | Sistem menampilkan pesan “password has been updated” | sesuai |  |

1. Fitur Kegiatan Ku

Tabel 4. Uji *Black Box Testing* Fitur Kegiatan Ku

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Kegiatan ku | Menguji tombol print kegiatan | Melihat laporan kegiatan | Klik tombol *Print* | Borwser mendownload file berupa pdf laporan kegiatan | sesuai |  |
| Menguji menu *paging* | Menuju halaman kegiatan sebelumnya | Klik tombol *Prev* | Sistem menampilkan halaman kegiatanpada *page* sebelumnya | sesuai |  |
| Menuju halaman kegiatan selanjutnya | Klik tombol *Next* | Sistem menampilkan halaman kegiatan pada *page* selanjutnya | sesuai |  |
| Menuju halaman kegiatan sesuai dengan nomor *page* | Klik tombol nomor (page) | Sistem menampilkan halaman kegiatan dengan *page* nomor pilihan | sesuai |  |

1. Fitur Buku Kontrol

Tabel 4. Uji *Black Box Testing* Fitur Buku Kontrol

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **FITUR** | **STEP DESCRIPTION** | **TEST STEP** | **ACTON** | **EXPECTED RESULT** | **STATUS** | **NOTE** |
| Buku kontrol | Menguji tombol print sertifikat | Melihat sertifkat | Klik tombol *Print* Sertifikat | *Browser* mendownload file sertifikat dengan format file pdf | sesuai |  |
| Menguji tombol print kontrol book | Melihat buku kontrol | Klik tombol *Print* Kontrol *Book* | Borwser mendownload file berupa pdf laporan kontrol *book* | sesuai |  |
| Menguji menu *paging* | Menuju halaman pelajaran sebelumnya | Klik tombol *Prev* | Sistem menampilkan halaman kegiatanpada *page* sebelumnya | sesuai |  |
| Menuju halaman pelajaran selanjutnya | Klik tombol *Next* | Sistem menampilkan halaman pelajaran pada *page* selanjutnya | sesuai |  |
| Menuju halaman pelajaran sesuai dengan nomor *page* | Klik tombol nomor (page) | Sistem menampilkan halaman pelajaran dengan *page* nomor pilihan | sesuai |  |

Dari hasil pengujian *black box* yang lampirkan di atas, hasil pengujian *black box* menunjukkan kesimpulan bahwa Sistem Informasi LPPAIK yang telah dirancang dan dibangun, secara fungsional fitur fitur yang dikembangkan sudah sesuai dengan kebutuhan dan yang diharapkan dengan melihat *expected result* dengan jumlah keseluruhan .... **nomor** .. skenario yang telah dilakukan pengujian.

# PENUTUP

## Kesimpulan

Asdfsadfaasadf sadf sadf as df asd fs adf sad f as dfsdfas df asd fasd f as f asdf asd f asdf asd f asdf as dfsdfas

Fasdfasdfas

Dfsdfas

dfasdfasdfasdfasdfasd

## Saran

asdfasdfa

# DAFTAR PUSTAKA

Achsan, S. A., & Susetyo, Y. A. (2022). Restful Web Service Implementation Using Spring Framework in Room Assets Management System. *Jurnal Teknik Informatika (JUTIF)*, *3*(2), 395–303. https://doi.org/10.20884/1.jutif.2022.3.2.213

Darwis, A., & Mahmud, H. (2017). Sistem Informasi Manajemen Pada Lembaga Pendidikan Islam. *Kelola: Journal of Islamic Education Management*, *2*(1), 64–77. https://doi.org/10.24256/kelola.v2i1.444

Dhika, H., Isnain, N., & Tofan, M. (2019). Manajemen Villa Menggunakan Java Netbeans Dan Mysql. *IKRA-ITH INFORMATIKA : Jurnal Komputer dan Informatika*, *3*(2), 104–110. https://journals.upi-yai.ac.id/index.php/ikraith-informatika/article/view/324

Fajrin, R. (2017). Rachmat Fajrin. *Jurnal Komputer Terapan*, *3*(1), 33–40. http://jurnal.pcr.ac.id

Fariyanto, F., Suaidah, S., & Ulum, F. (2021). Perancangan Aplikasi Pemilihan Kepala Desa Dengan Metode UX Design Thinking. *Jurnal Teknologi dan Sistem Informasi*, *2*(2), 52–60. http://jim.teknokrat.ac.id/index.php/sisteminformasi/article/view/853/351

Findawati, Y. (2018). Buku Ajar Rekayasa Perangkat Lunak. In *Buku Ajar Rekayasa Perangkat Lunak*. https://doi.org/10.21070/2018/978-602-5914-09-6

Firdhayanti, A., Saleh, S., Nurlistiani, R., Zainal, J., Pagar, A., Bandar, A., & Indonesia, L. (2023). *Perancangan Sistem Informasi Daftar Kajian Islam Di Bandarlampung Berbasis Web*. *17*(x), 89–100.

Isa, I. G. T., & Hartawan, G. P. (2017). Perancangan Aplikasi Koperasi Simpan Pinjam Berbasis Web (Studi. *Jurnal Ilmiah Ilmu Ekonomi*, *5*(10), 139–151.

Karman, J., & Rahmanto, D. (2020). Sistem Informasi Jadwal Kegiatan Majelis Ilmu Pada Kajian Linggau Mengaji Berbasis Web Mobile. *JUSIM (Jurnal Sistem Informasi Musirawas)*, *5*(01), 55–68. https://doi.org/10.32767/jusim.v5i01.837

Komalasari, R. (2020). Manfaat Teknologi Informasi Dan Komunikasi Di Masa Pandemi Covid 19. *Tematik*, *7*(1), 38–50. https://doi.org/10.38204/tematik.v7i1.369

Maharani, D., Helmiah, F., & Rahmadani, N. (2021). Penyuluhan Manfaat Menggunakan Internet dan Website Pada Masa Pandemi Covid-19. *Abdiformatika: Jurnal Pengabdian Masyarakat Informatika*, *1*(1), 1–7. https://doi.org/10.25008/abdiformatika.v1i1.130

Nursaid, F. F., Hendra Brata, A., & Kharisma, A. P. (2020). Pengembangan Sistem Informasi Pengelolaan Persediaan Barang Dengan ReactJS Dan React Native Menggunakan Prototype (Studi Kasus : Toko Uda Fajri). *J-Ptiik.Ub.Ac.Id*, *4*(1), 46–55. http://j-ptiik.ub.ac.id

Pamungkas, R. (2018). Optimalisasi Query Dalam Basis Data My Sql Menggunakan Index. *RESEARCH : Computer, Information System & Technology Management*, *1*(1), 27. https://doi.org/10.25273/research.v1i1.2453

Pratama, D. R., & Robbani, R. (2023). *Pengembagan REST API SIABANG (Sistem Administrasi Pembangunan) Menggunakan Java*. *4*(1), 133–142.

Prilsafira, T., Kunang, Y. N., Putra, M. H., Informasi, S., Darma, U. B., Komputer, I., Darma, U. B., & Academy, B. (2023). *REST API Backend Aplikasi E-Commerce Secondhand Menggunakan Framework Spring Boot*.

Putri Primawanti, E., & Ali, H. (2022). Pengaruh Teknologi Informasi, Sistem Informasi Berbasis Web Dan Knowledge Management Terhadap Kinerja Karyawan (Literature Review Executive Support Sistem (Ess) for Business). *Jurnal Ekonomi Manajemen Sistem Informasi*, *3*(3), 267–285. https://doi.org/10.31933/jemsi.v3i3.818

Rianto, D. A., Assegaf, S., & Fernando, E. (2017). Perancangan Aplikasi Sistem Informasi Geografis (Sig) Lokasi Minimarket Di Kota Jambi Berbasis Android. *Jurnal Ilmiah Media SISFO*, *9*(2), 295–304.

Ridwan, M., Ulum, B., Muhammad, F., Indragiri, I., & Sulthan Thaha Saifuddin Jambi, U. (2021). Pentingnya Penerapan Literature Review pada Penelitian Ilmiah (The Importance Of Application Of Literature Review In Scientific Research). *Jurnal Masohi*, *2*(1), 42–51. http://journal.fdi.or.id/index.php/jmas/article/view/356

Rochman, A., Sidik, A., & Nazahah, N. (2018). Perancangan Sistem Informasi Administrasi Pembayaran SPP Siswa Berbasis Web di SMK Al-Amanah. *Jurnal Sisfotek Global*, *8*(1). https://doi.org/10.38101/sisfotek.v8i1.170

Sallaby, A. F., & Kanedi, I. (2020). Perancangan Sistem Informasi Jadwal Dokter Menggunakan Framework Codeigniter. *Jurnal Media Infotama*, *16*(1), 48–53. https://doi.org/10.37676/jmi.v16i1.1121

Sandhika Jaya, T., Sahlinal, D., Manajemen Informatika, J., & Negeri Lampung, P. (2017). Perancangan Kantor Digital Berbasis Framework dengan Metode Waterfall pada Politeknik Negeri Lampung. *Jurnal Informatika: Jurnal Pengembangan IT*, *2*(2), 14–17. http://ejournal.poltektegal.ac.id/index.php/informatika/article/view/518

Sujiwa, A., & Rochman, S. (2019). Pengembangan Sistem Kontrol Serta Monitoring Suhu dan Volume Air Berbasis Web Pada Perangkat Desalinasi Air Laut. *Seminar Nasional Hasil Riset dan Pengabdian*, *II*, 1–9.

Suryawinata, M. (2019). Buku Ajar Mata Kuliah Pengembangan Aplikasi Berbasis Web Diterbitkan oleh UMSIDA PRESS. In *Buku Ajar Mata Kuliah Pengembangan Aplikasi Berbasis Web*. https://press.umsida.ac.id/index.php/umsidapress/article/view/978-602-5914-81-2/849; http://dx.doi.org/10.21070/2019/978-602-5914-81-2

Umbu Dagha, W. C. (2021). Web Event, Spring Boot, Java Pembangunan Aplikasi Web Event menggunakan Framework Spring Boot di PT XYZ. *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, *8*(3), 1457–1469. https://doi.org/10.35957/jatisi.v8i3.1052

---------- custom---------

Hernata, Y., & Samsudin, S. (2017). Sistem Informasi Lembaga Pengkajian Studi Islam (LPSI) Universitas Islam Indragiri Tembilahan Berbasis Web. Sistemasi: Jurnal Sistem Informasi, 6(2), 1-8.

Salamah, U. G., & ST, S. (2021). Tutorial Visual Studio Code. Media Sains Indonesia.

Saputra, B. D., & Stefanie, A. (2023). Automation Testing Api, Android, dan Website Menggunakan Serenity Bdd Pada Software Sistem Manajemen Rumah Sakit. Jurnal Ilmiah Wahana Pendidikan, 9(10), 114-126.

Haryana, P. P. (2019). Pengembangan Framework Asesmen Dampak Perambatan Vulnerability Pada Node Package Manager (NPM) (Doctoral dissertation, Universitas Gadjah Mada).

Wali, M., & Ahmad, L. (2018). Perancangan Access Open Journal System (AOJS) dengan menggunakan Framework Codeigniter dan ReactJs. Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi), 2(1), 48-56.

Gunawan, I., Akbar, T., & Anwar, K. (2019). Prototipe Sistem Monitoring Tegangan Panel Surya (Solar Cell) Pada Lampu Penerang Jalan Berbasis Web Aplikasi. Infotek J Inform dan Teknol, 2(2), 70-8.

Enterprise, J. (2015). Mengenal java dan database dengan netbeans. Elex Media Komputindo.

Enterprise, J. (2017). Java Komplet. Elex Media Komputindo.

Hutahaean, J. (2015). Konsep sistem informasi. Deepublish.

Anggraeni, E. Y. (2017). Pengantar sistem informasi. Penerbit Andi.

Prehanto, D. R., Kom, S., & Kom, M. (2020). Buku Ajar Konsep Sistem Informasi. Scopindo Media Pustaka.

Aws.amazon.com. (2023, 25 Mei) What is API RESTful Diakses pada 1 Juni 2023, dari https://aws.amazon.com/id/what-is/restful-api/

Restfulapi.net. (2022, 7 April) What is REST. Diakses pada 1 Juni 2023, dari https://restfulapi.net

Jayanti, N. K. D. A., & Sumiari, N. K. (2018). Teori basis data. Penerbit Andi.

Yanto, R. (2016). Manajemen Basis Data Menggunakan MySQL. Deepublish.

Lubis, A. (2016). Basis data dasar. Deepublish.