

BAB III

METODE PENELITIAN

3.1 Pengumpulan Data

3.1.1. Lokasi Penelitian

Pelaksanaan penelitian ini berlokasi di Badan Pengembangan Sumber Daya Manusia daerah Provinsi Lampung, Jalan Raya Hajimena, Kec. Natar, Kabupaten Lampung Selatan.

3.1.2. Waktu Penelitian

Kegiatan penelitian ini membutuhkan waktu 3 bulan pada bulan November, Desember tahun 2019 dan Januari tahun 2020.

3.1.3. Metode Penelitian

Metode yang digunakan dalam penelitian ini adalah metode deskriptif. Metode deskriptif adalah suatu metode dalam meneliti sekelompok manusia, suatu objek, suatu metode yang mengemukakan masalah dengan mengumpulkan data-data yang disajikan untuk menggambarkan karakteristik suatu keadaan atau objek penelitian dan mengambil kesimpulan yang akan dilaksanakan.

3.1.4. Metode Pengumpulan Data

1. Wawancara atau Interview

Pengumpulan data yang dilakukan penulis dalam menunjang kelengkapan data melalui metode wawancara atau *interview*. Penulis melakukan Tanya jawab dengan pihak yang bertanggung jawab dalam pengolahan data jadwal pembelajaran diklat di BPSDM Provinsi Lampung mengenai bagaimana langkah – langkah dalam pengelolaan jadwal pembelajaran.

2. Observasi

Metode yang digunakan dalam pengumpulan data tidak hanya dengan metode wawancara atau *interview* tetapi juga melalui metode observasi. Penulis mencari data – data yang dibutuhkan dalam pembuatan aplikasi jadwal mulai dari

pencatatan, pengolahan, penyimpanan, pemeliharaan hingga pembagian jadwal dengan melakukan survey di BPSDM Provinsi Lampung.

3. Studi Pustaka

Metode studi pustaka ialah salah satu pencarian dan pengumpulan data dengan cara membaca buku, laporan – laporan yang berkaitan dengan objek penelitian dan dapat dijadikan sebagai dasar teori serta dapat dijadikan bahan perbandingan.

3.2 Analisa Sistem yang berjalan

Tahap analisis system berjalan dalam pembuatan jadwal diklat yang ada pada BPSDM Provinsi Lampung ini masih menggunakan aplikasi perkantoran (*Microsoft Office*) kemudian hasilnya dibagikan dan disimpan di map – map berupa dokumen arsip.

3.3 Analisa Kebutuhan Sistem baru

3.3.1 Gambaran Umum Aplikasi

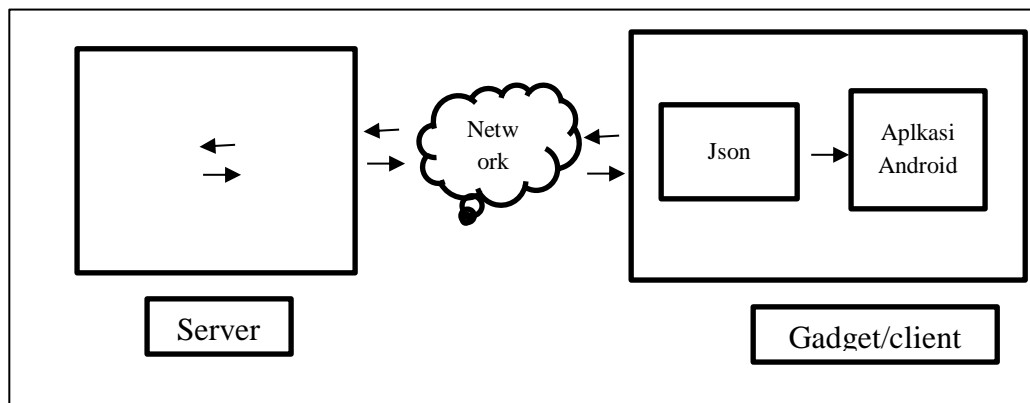
Secara umum aplikasi *mobile* yang diusulkan merupakan sebuah perangkat lunak yang berfungsi untuk melakukan pengambilan, memasang pengingat, dan menampilkan data jadwal dari Basis data Sistem berbasis web secara otomatis. Aplikasi mobile dihubungkan oleh sebuah *web service* untuk mengambil data dari basis data *system* berbasis web. Proses pengambilan data ini akan berlangsung secara berkala pada waktu-waktu tertentu untuk memastikan informasi jadwal yang tersimpan merupakan jadwal yang terbaru. Informasi jadwal yang diperoleh selanjutnya disimpan dalam basis data lokal aplikasi mobile, sehingga proses menampilkan dan proses pemasangan pengingat dapat dilakukan dengan lebih cepat.

Pengembangan aplikasi mobile diarahkan pada dua sisi sebagai berikut:

1. Mengembangkan aplikasi input jadwal berbasis *web* dan *web service* yang berfungsi untuk melakukan administrasi jadwal dan sebagai penghubung antara aplikasi mobile dengan server basis data.
2. Mengembangkan aplikasi mobile berbasis android yang merupakan aplikasi mobile di sisi client yang dapat mengakses layanan *web service*.

3.3.2 Arsitektur Aplikasi

Berdasarkan gambar 3.1 pada saat aplikasi mengirim data ke *server*, maka harus melalui jaringan internet, setelah itu memanggil file *php* dan selanjutnya *php* akan melakukan pengecekan ke database *MySQL* dan mengirimkannya kembali ke aplikasi android, lalu dari *MySQL* akan diterjemahkan ke dalam bentuk *JSON (Javascript Object Nation)* dan disimpan ke database *SQLite*. *JSON* berfungsi untuk melakukan *parsing* data dalam bentuk *string* dengan format tertentu agar bias dibaca oleh aplikasi android dari *MySQL*.



Gambar 3.1 Arsitektur Aplikasi

3.4 Rancangan Sistem

Pada tahapan ini, penulis memberikan informasi mengenai rancangan sistem yang penulis ajukan guna menanggulangi permasalahan – permasalahan sistem yang telah disebutkan sebelumnya.

3.4.1 Use Case Model Diagram

3.4.1.1 Identifikasi Aktor

Actor mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah *actor* mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. *Actor* hanya berinteraksi dengan *use case*, tetapi tidak memiliki kendali atas *use case*. Adapun *actor* yang terlibat dalam aplikasi ini dapat diklasifikasikan menjadi 3 *actor* yaitu Peserta, Pengajar dan Admin Web. Untuk lebih jelasnya mengenai masing-masing *actor* yang ada pada sistem dapat dilihat pada table 3.1. Identifikasi *actor*.

Tabel 3.1 Identifikasi Actor

Aktor	Deskripsi
Peserta	Menerima notifikasi dari Admin web
Pengajar	Menerima notifikasi dari Admin web
Web Admin	Mengirim notifikasi ke seluruh <i>actor</i>

3.4.1.2 Identifikasi Use Case

Setelah *actor* teridentifikasi, selanjutnya mengidentifikasi *use case*.

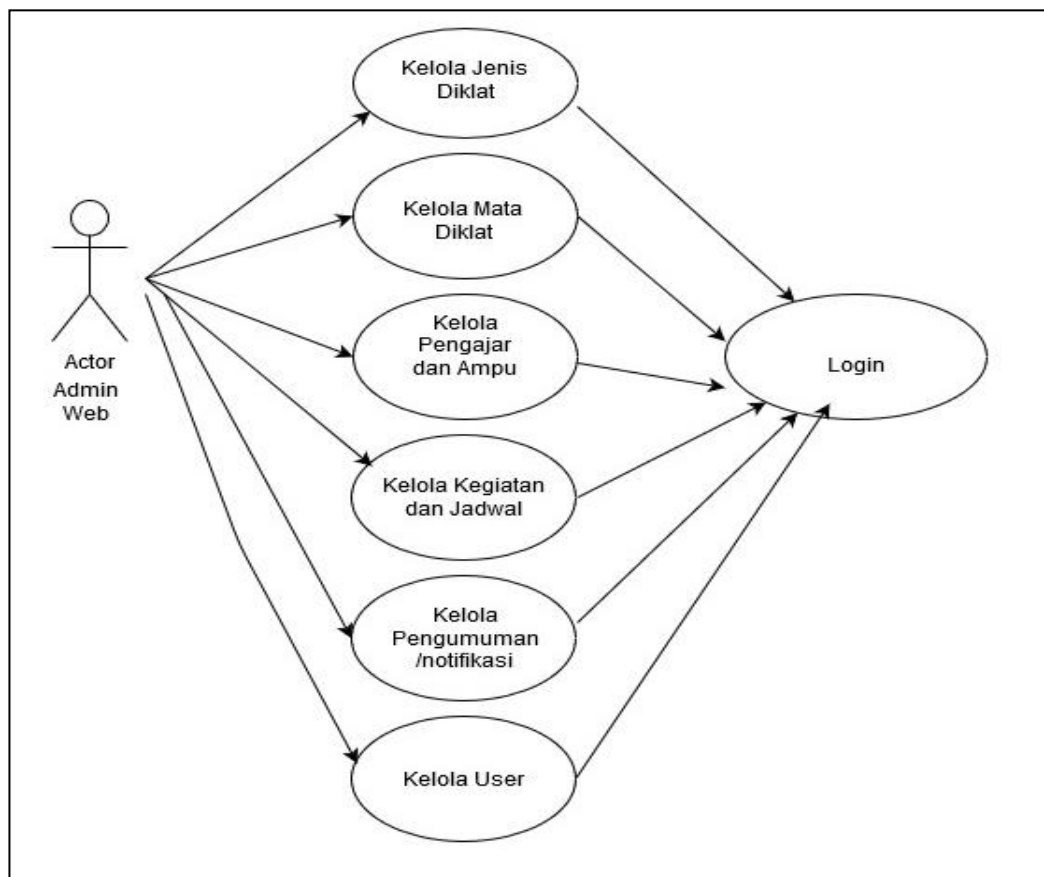
Tabel 3.2 Identifikasi Use case

No	Nama Use Case	Deskripsi	Aktor
1	Login	<i>Use case</i> menggambarkan kegiatan login ke dalam sistem dengan memasukkan <i>username</i> dan <i>password</i> untuk masuk ke halaman utama	Semua pengguna
2	Kirim Pesan Notifikasi (<i>Broadcast</i>)	<i>Use case</i> ini menggambarkan kegiatan mengirim pesan <i>broadcast</i> notifikasi	Web admin
3	Membuat Jadwal Diklat	<i>Use case</i> ini menggambarkan kegiatan membuat jadwal diklat	Web admin
3	Lihat Jadwal Diklat	<i>Use case</i> ini menggambarkan kegiatan melihat jadwal diklat	Semua pengguna
4	Lihat Pengumuman	<i>Use case</i> ini menggambarkan kegiatan melihat jadwal pengumuman	Semua pengguna

3.4.1.3 Use Case Diagram

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. Berikut ini adalah *use case* diagram dari sistem yang akan dibangun :

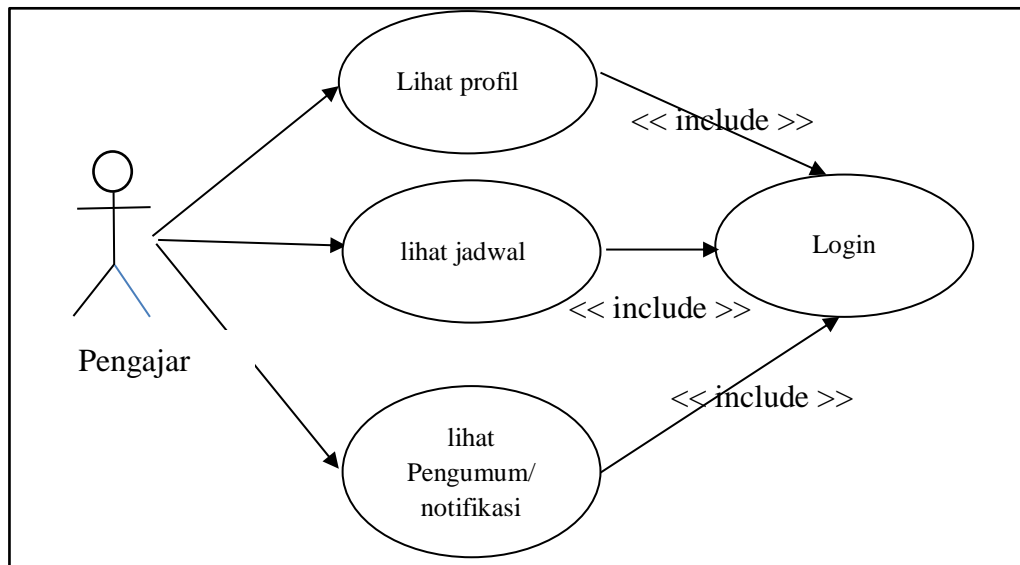
1. Use case admin web



Gambar 3.2 Use Case Web admin

Admin web mempunyai hak akses penuh untuk mengelola data jenis diklat, mata diklat, pengajar dan ampu, kegiatan dan jadwal, dan mengelola notifikasi/pengumuman serta mengelola data *user*. Dalam mengelola semua data tersebut Admin web harus terlebih dahulu login menggunakan *username* dan *password*, setelah sukses admin web bisa dapat mengelola data yang ada disistem.

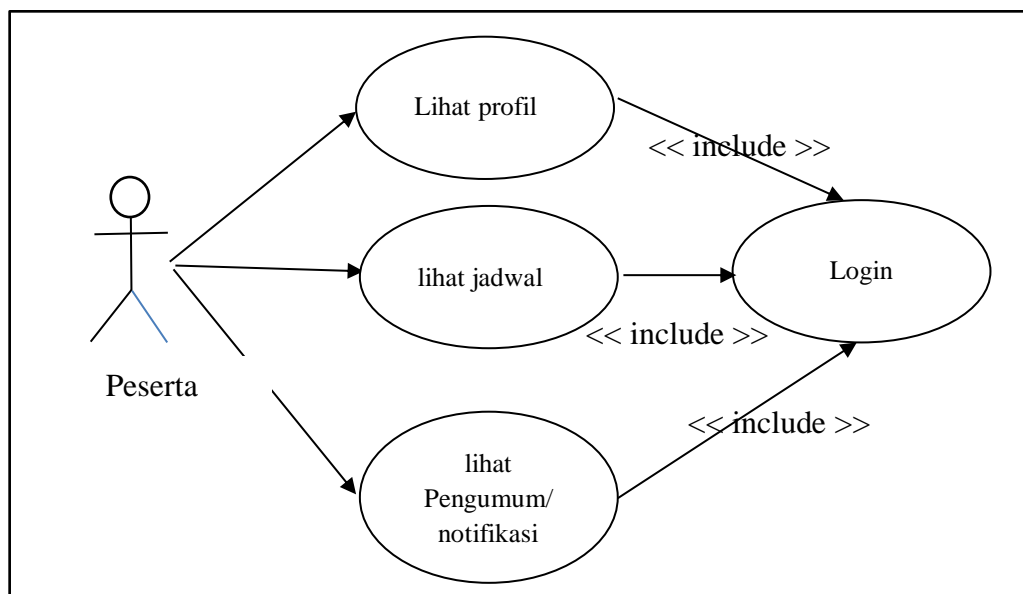
2. Use case pengajar



Gambar 3.3 Use case pengajar pada aplikasi android.

Pengajar merupakan pengguna yang akan mengakses informasi jadwal, Mata diklat dan pengumuman dari aplikasi mobile, pengguna diharuskan login ke dalam sistem untuk dapat mengakses konten yang disajikan dalam aplikasi mobile.

3. Use case peserta



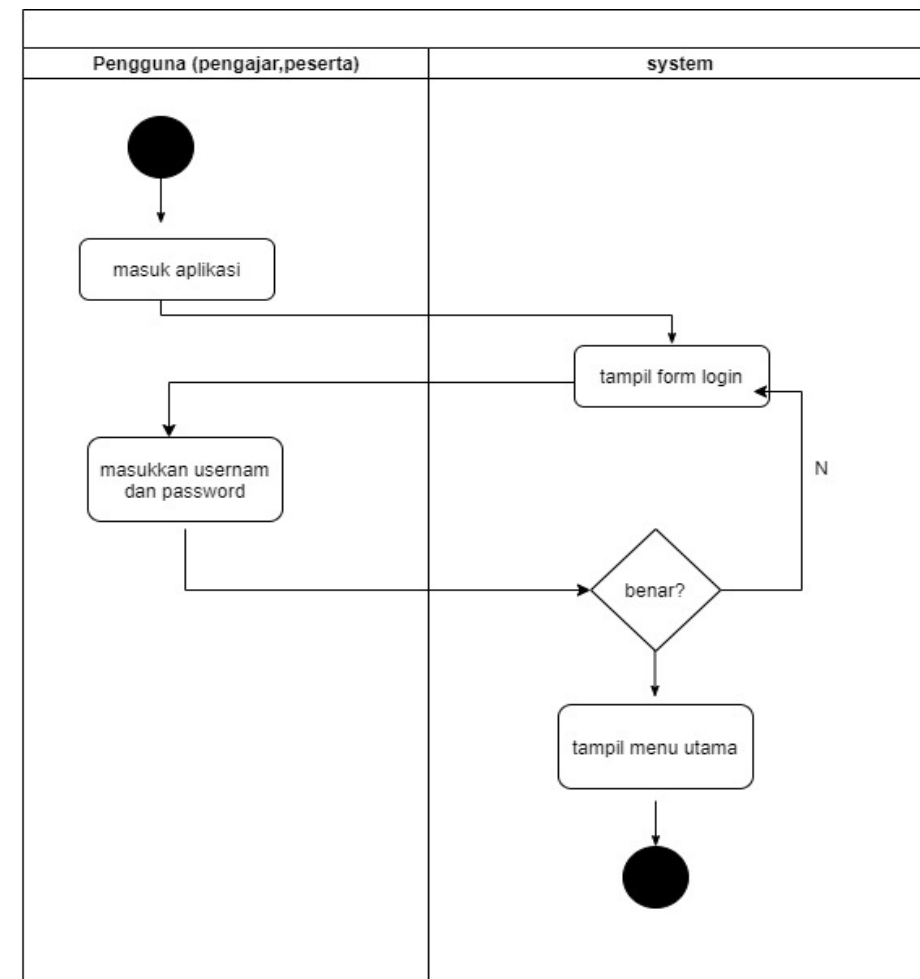
Gambar 3.4 Use Case Peserta pada Aplikasi android

Peserta merupakan pengguna yang akan mengakses informasi jadwal Mata diklat dan pengumuman dari aplikasi *mobile*, Pengguna diharuskan login ke dalam sistem untuk dapat mengakses konten yang disajikan dalam aplikasi mobile.

3.4.2 Activity Diagram

Berdasarkan dari *use case* model diagram pada gambar 3.2, gambar 3.3 dan gambar 3.4, maka dapat digambarkan aktivitas-aktivitas yang terjadi atau alur kerja dalam *use case*. Aliran kerja tersebut digambarkan secara grafis dengan *Activity diagram*. Berikut ini *Activity diagram* dari masing-masing *use case*:

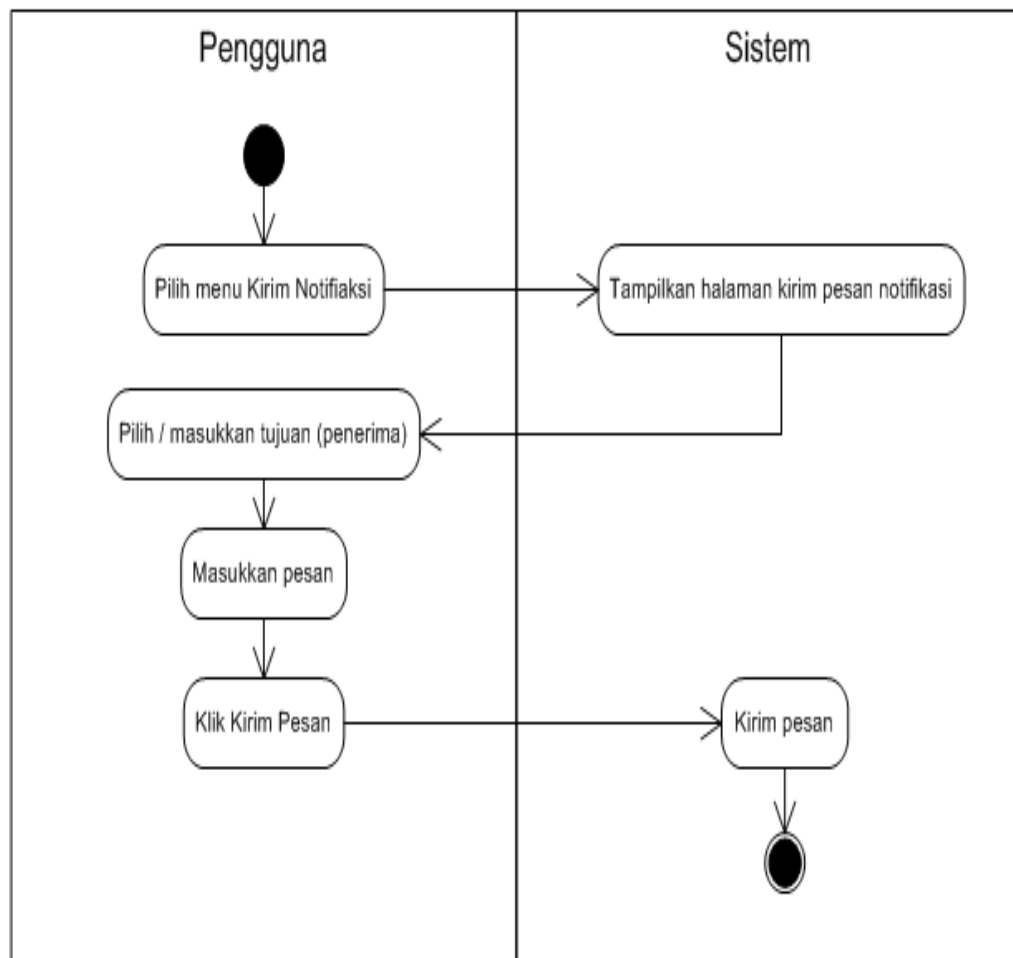
3.4.2.1 Activity Diagram Login



Gambar 3.5 Activity Diagram Login

Aktivitas pada gambar 3.4 dilakukan oleh pengguna pengajar dan peserta dengan melakukan pengisian *username* dan *password* pada halaman login, jika *username* dan *password* yang dimasukkan salah, maka sistem akan menampilkan peringatan *username* dan *password* yang dimasukkan salah. Jika benar, maka *user* akan dapat mengakses menu utama aplikasi tersebut.

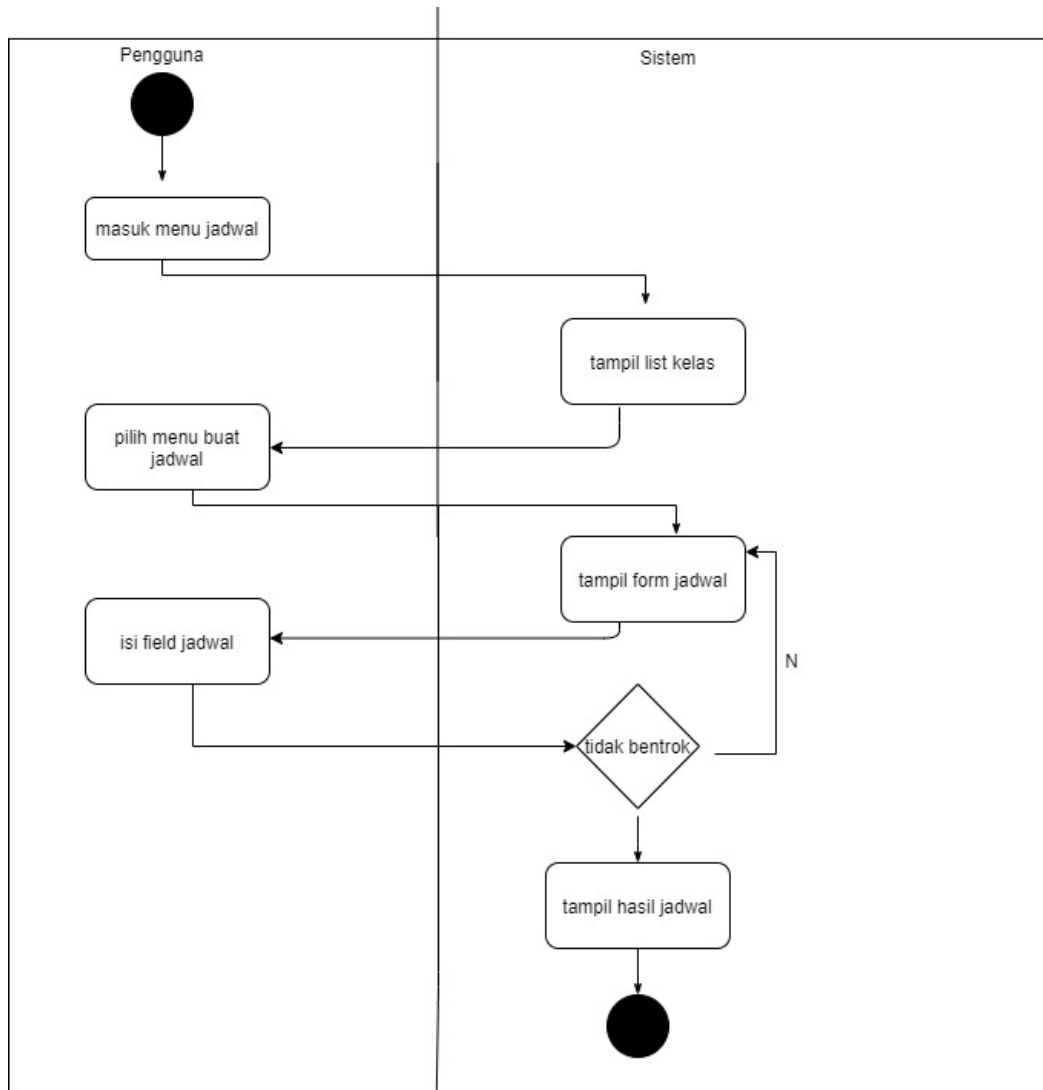
3.4.2.2 Activity Diagram Pesan Notifikasi



Gambar 3.6 Activity Pesan Notifikasi

Pesan notifikasi dikirim oleh admin web, agar bisa mengirim pesan notifikasi, maka pengguna admin web tersebut harus *login* terlebih dahulu. Setelah berhasil *login* admin web memilih menu pesan notifikasi, kemudian memasukkan isi teks yang akan dikirim, lalu setelah itu memilih tujuan penerima.

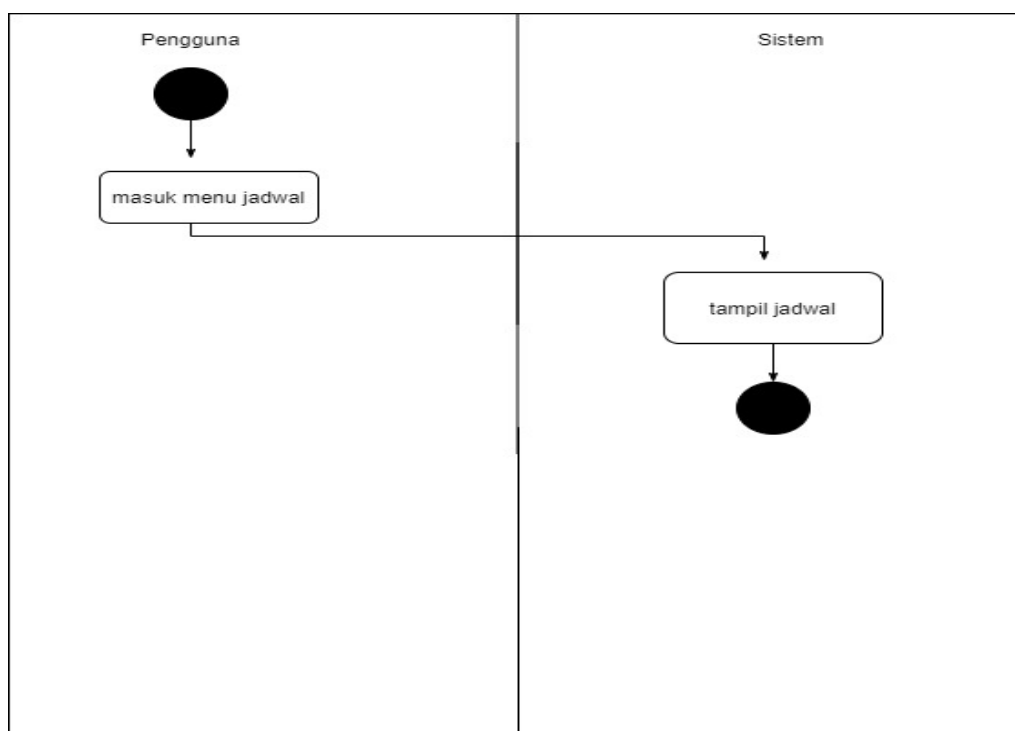
3.4.2.3 Activity Diagram membuat Jadwal.



Gambar 3.7 Activity Diagram membuat jadwal

Pada aktivitas ini dibuat oleh pengguna web admin, dimana setelah login pengguna memilih menu jadwal, kemudian tampil list kelas dan pilih menu buat jadwal. Jika kondisi pengajar tidak bentrok maka jadwal akan tersimpan, jika tidak maka akan kembali ke form jadwal.

3.4.2.4 Activity Diagram Lihat Jadwal



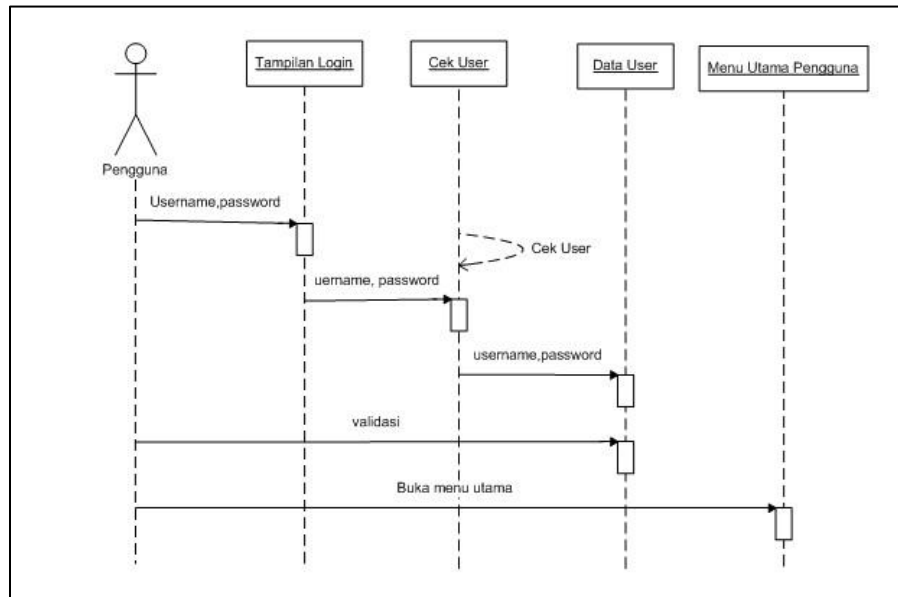
Gambar 3.8 Activity Diagram Lihat jadwal

Pada aktivitas ini pengguna baik pengajar maupun memilih menu jadwal, yang sebelumnya sudah login terlebih dahulu, maka akan tampil jadwal sesuai dengan kelas yang diikutinya atau kelas yang diampu oleh pengajar.

3.4.3 Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek – obyek yang melakukan suatu tugas atau aksi tertentu. Obyek – obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram. Pada diagram ini, dimensi vertikal merepresentasikan waktu. Bagian paling atas dari diagram menjadi titik awal dan waktu berjalan ke bawah sampai dengan bagian dasar dari diagram.

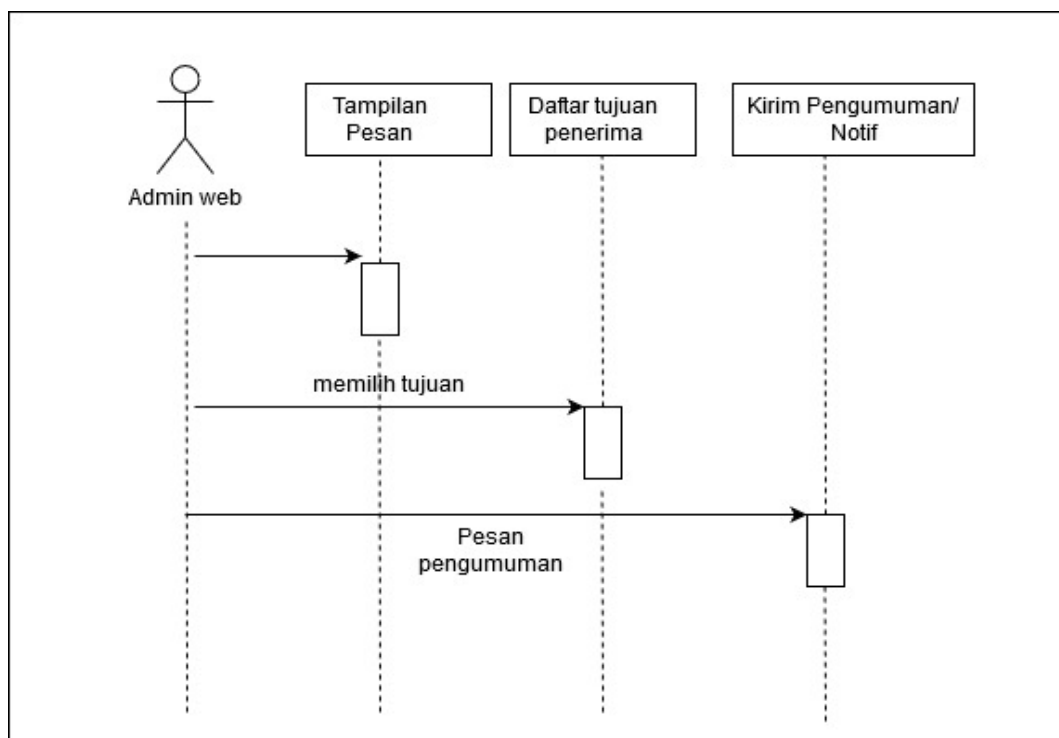
3.4.3.1. Sequence Diagram Login



Gambar 3.9 Sequence Diagram Login

Diagram ini menggambarkan proses login, mulai dari user memasukkan username dan password, lalu sistem mengeceknya ke database, dan menampilkan halaman utama.

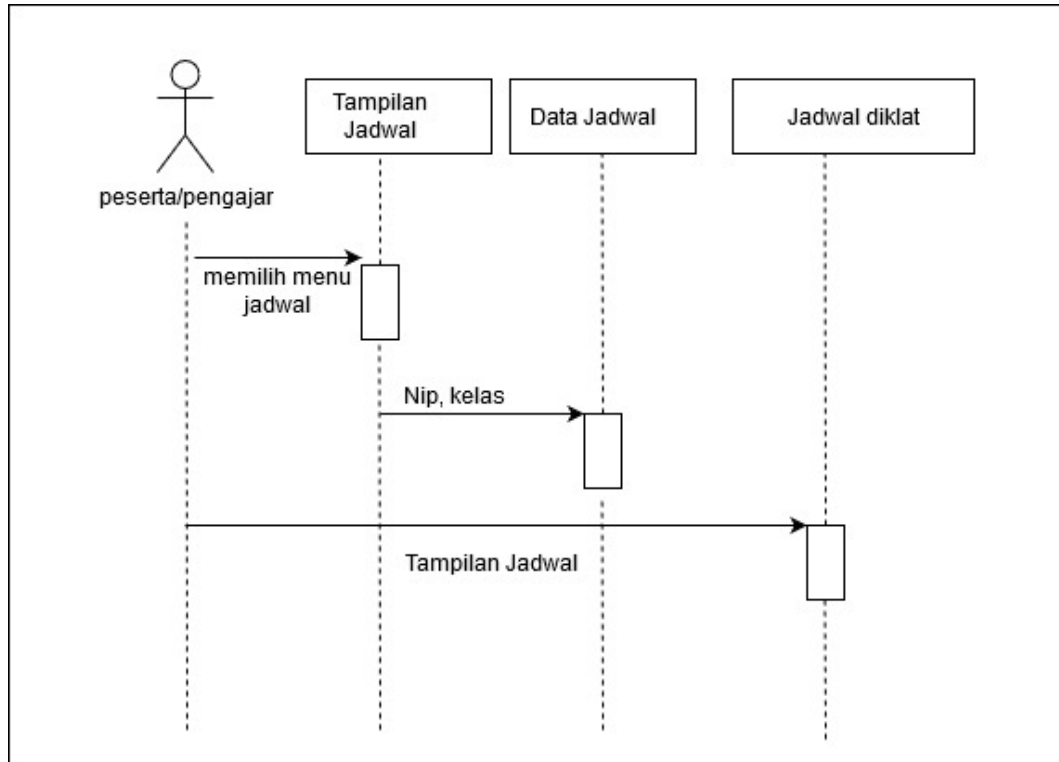
3.4.3.2. Sequence Diagram Pengumuman/Notifikasi



Gambar 3.10 Sequence Diagram Pengumuman/notifikasi

Pengguna admin web menulis pesan pengumuman, dan setelah menulis pesan pengguna diarahkan untuk memilih tujuan penerima, lalu setelah itu pesan akan dikirim sesuai dengan tujuan penerima yang dipilih.

3.4.3.3. *Sequence Diagram* Jadwal diklat



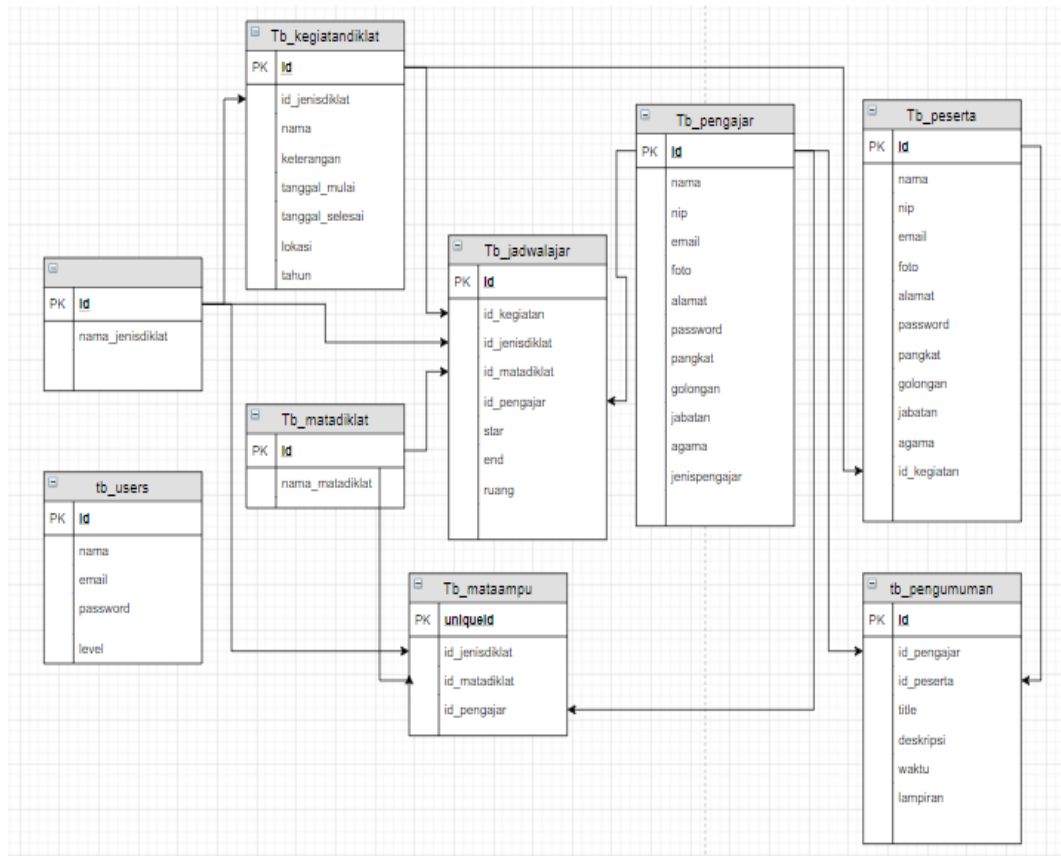
Gambar 3.11 Sequence Diagram jadwal diklat

Jadwal bisa di akses oleh pengguna jika sudah melakukan *login*, setelah pengguna memilih jadwal , sistem akan mengecek data jadwal pada *database* dan jadwal akan ditampilkan sesuai dengan kelas yang sedang diikuti.

3.5. **Pemodelan Data**

Pemodelan data pada perancangan aplikasi mobile meliputi penggambaran *entity relationship diagram*, merancang tabel-tabel yang dibutuhkan pada basis data dan membuat relasi antar *table Entity relationship* yang dibuat memiliki keterkaitan antara data yang satu dengan data yang lainnya. Tabel-tabel yang telah dirancang kemudian direalisasikan antara tabel satu dengan yang lain untuk mendukung kelancaran pengolahan data.

Tabel ini dibuat di database MySQL. MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang *multithread*, dan *multi-user*. Pada gambar 3.12 menggambarkan Tabel *Realitionship* dari sistem yang akan diajukan yaitu tabel “tb_jenisdiklat”, “tb_matadiklat”, “tb_kegiatandiklat”, “tb_jadwalajar”, “tb_mataampu”, “tb_pengajar”, “tb_peserta”, “tb_pengumuman” dan “tb_user” adalah tabel yang tidak berelasi.



Gambar 3.12 Relasi Tabel

Adapun rincian tabel-tabel yang akan dibuat adalah sebagai berikut:

1. Tabel Jenis diklat
 Nam *Database* : db_simpelasn
 Nama Tabel : tb_jenisdiklat

Tabel 3.3. Tabel Jenis Diklat

Nama Field	Type Data	Size	Keterangan
id	Int	10	Primary Key
nama_jenisdiklat	Varchar	100	

2. Tabel Mata Diklat

Nam *Database* : db_simpelasn

Nama Tabel : tb_matadiklat

Tabel 3.4. Tabel Mata Diklat

Nama Field	Tipe Data	Size	Keterangan
id	Int	10	<i>Primary Key</i>
Id_jenisdiklat	Int	10	<i>Foreign key</i>
nama_matadiklat	varchar	100	

3. Tabel Kegiatan Diklat

Nam *Database* : db_simpelasn

Nama Tabel : tb_kegiatandiklat

Tabel 3.5. Tabel Kegiatan Diklat

Nama Field	Tipe Data	Size	Keterangan
id	Int	10	<i>Primary Key</i>
Id_jenisdiklat	Int	10	<i>Foreign key</i>
nama	Varchar	255	
tahun	Varchar	10	
tanggal_mulai	Date		
tanggal_selesai	Date		
lokasi	Varchar	255	
keterangan	Varchar	255	

4. Tabel Jadwal Ajar

Nam *Database* : db_simpelasn

Nama Tabel : tb_jadwalajar

Tabel 3.6. Tabel Jadwal ajar

Nama Field	Tipe Data	Size	Keterangan
id	Int	10	<i>Primary Key</i>
id_kegiatandiklat	Int	10	<i>Foreign key</i>
id_jenisdiklat	Int	10	<i>Foreign key</i>
id_matadiklat	Int	10	<i>Foreign key</i>
star	datetime		
end	datetime		
ruang	varchar	255	

5. Tabel Mata Ampu

Nam *Database* : db_simpelasn

Nama Tabel : tb_mataampu

Tabel 3.7. Tabel Mata ampu

Nama Field	Tipe Data	Size	Keterangan
id	Int	10	<i>Primary Key</i>
id_jenisdiklat	Int	10	<i>Foreign key</i>
id_matadiklat	Int	10	<i>Foreign key</i>
id_pengajar	Int	10	<i>Foreign key</i>

6. Tabel Pengajar

Nam *Database* : db_simpelasn

Nama Tabel : tb_pengajar

Tabel 3.8. Pengajar

Nama Field	Tipe Data	Size	Keterangan
id	Int	10	<i>Primary Key</i>
nama	varchar	255	
nip	varchar	15	
email	varchar	255	
foto	varchar	255	
pangkat	varchar	255	
golongan	varchar	255	
jabatan	varchar	255	
agama	varchar	255	
jenispendidikan	varchar	255	
alamat	Text		
password	varchar	255	

7. Tabel User

Nam *Database* : db_simpelasn

Nama Tabel : tb_user

Tabel 3.9. Tabel User

Nama Field	Tipe Data	Size	Keterangan
id	Int	10	<i>Primary Key</i>
nama	Varchar	255	
email	Varchar	255	
password	Varchar	255	
level	Enum		

8. Tabel Peserta

Nam *Database* : db_simpelasn

Nama Tabel : tb_peserta

Tabel 3.10. Peserta

Nama Field	Tipe Data	Size	Keterangan
Id	Int	10	<i>Primary Key</i>
nama	Varchar	255	
Nip	Varchar	15	
email	Varchar	255	
Foto	Varchar	255	
pangkat	Varchar	255	
golongan	Varchar	255	
jabatan	Varchar	255	
agama	Varchar	255	
Id_kegiatan	Int	10	<i>Foreign key</i>
alamat	Varchar	255	
password	Varchar	255	

9. Tabel Pengumuman

Nam *Database* : db_simpelasn

Nama Tabel : tb_pengumuman

Tabel 3.11. Tabel Pengumuman

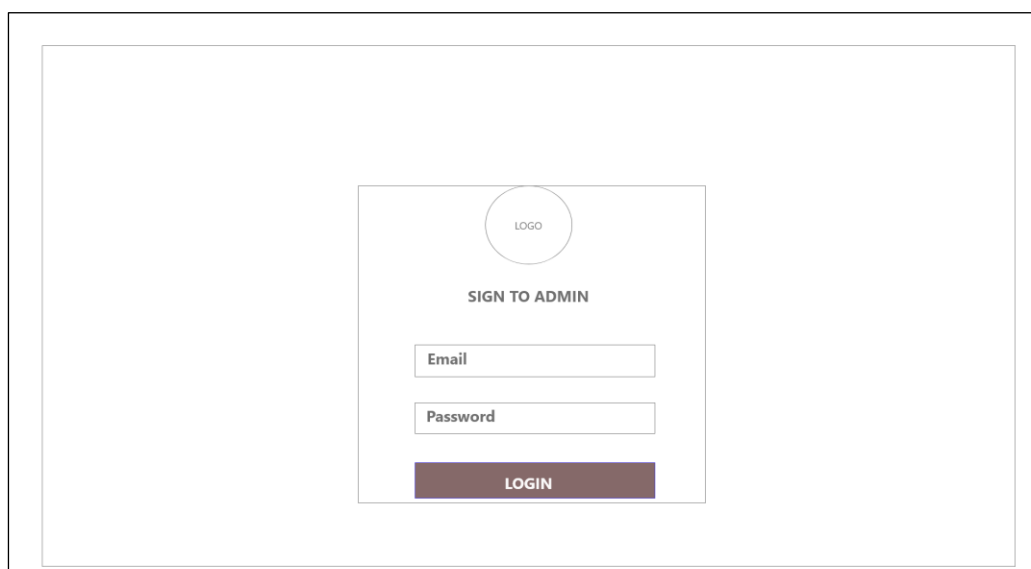
Nama Field	Tipe Data	Size	Keterangan
Id	Int	10	<i>Primary Key</i>
Id_pengajar	Int	10	<i>Foreign key</i>
Id_pesert	Int	10	<i>Foreign key</i>
Title	varchar	255	
deskripsi	varchar	255	
waktu	datetime		
Lampiran	varchar	255	

3.6. Desain Antar Muka

Desain antar muka aplikasi digunakan untuk memberikan gambaran antar muka aplikasi yang sedang dirancang. Ada beberapa rancangan antar muka yang terbagi kedalam tiga rancangan sesuai dengan hak akses aplikasi yaitu admin web, pengajar dan peserta, dimana setiap rancangan dari tiap hak akses tersebut akan dijabarkan sebagai berikut :

3.6.1. Desain Antar Muka Admin Web

1. Halaman Web Login

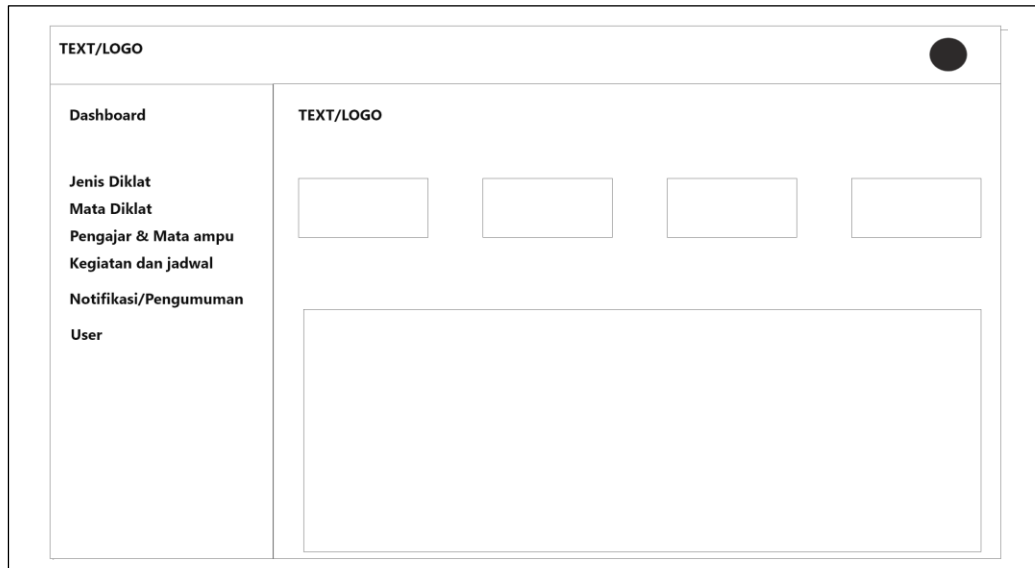


Gambar 3.13 Rancangan Halaman Login

Gambar 3.13 menggambarkan halaman utama yang pertama kali muncul ketika membuka aplikasi admin web, yaitu halaman login, pengguna harus memasukkan username berupa *email* dan *password* untuk dapat melanjutkan ke menu berikutnya.

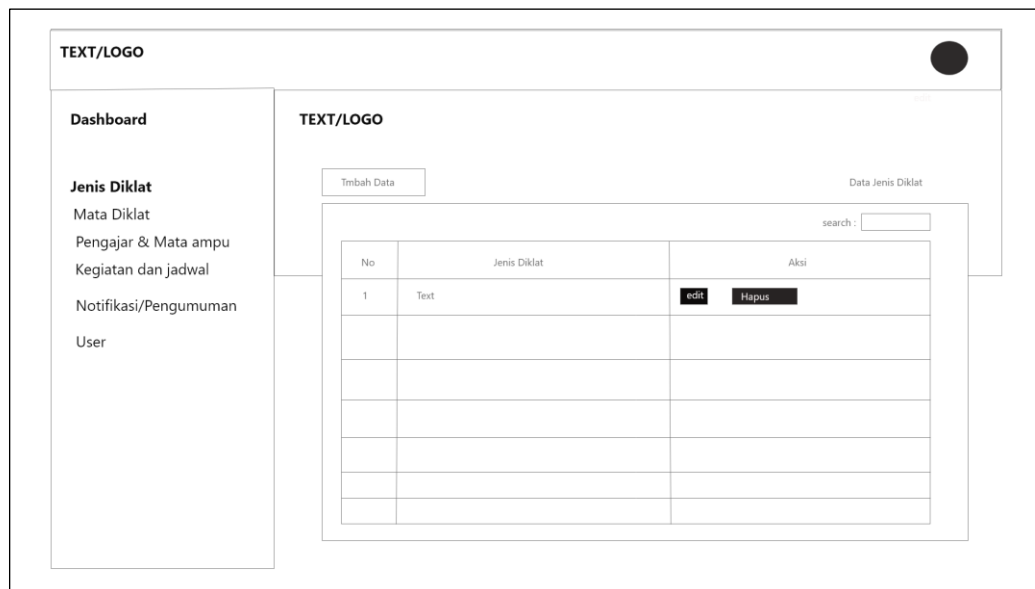
2. Halaman Web *Dashboard*

Gambar 3.14 Halaman *dashboard* Merupakan halaman menu pada aplikasi admin web setelah pengguna melakukan login dengan berhasil. Halama *dashboard* memiliki beberapa menu utama yang dapat diakses, diantaranya adalah menu jenis diklat, mata diklat, pengajar dan mata ampu. Kegiatan dan jadwal serta menu user.

Gambar 3.14 Halaman web *Dashboard*

3. Halaman Web Jenis Diklat

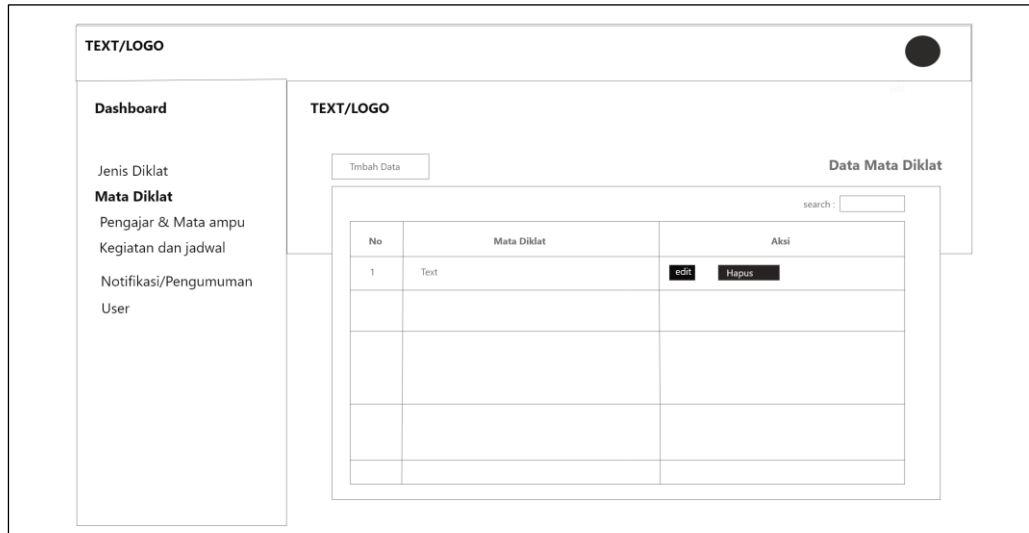
Halaman jenis diklat merupakan salah satu master data untuk mengelompokkan diklat kedalam jenis-jenis. Halaman web Jenis diklat mempunyai 2 (dua) halaman utama yaitu daftar jenis diklat dan *form input* yang dapat dilihat ketika pengguna memilih tombol tambah data, pengguna harus login terlebih dahulu untuk dapat mengakses halaman ini, rancangan tampilan web jenis diklat dapat dilihat gambar 3.15 berikut :



Gambar 3.15 Halaman Daftar jenis diklat

4. Halaman Web Mata Diklat

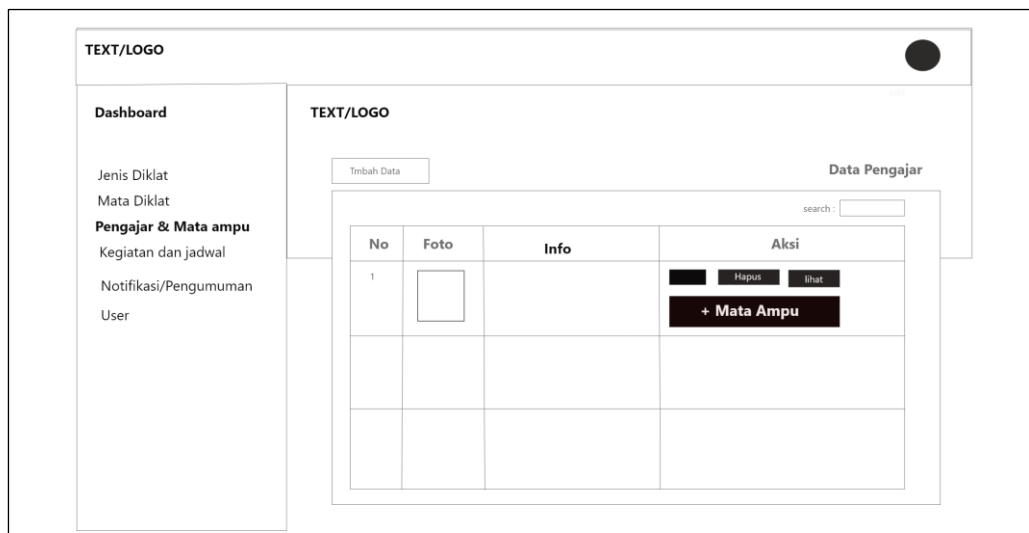
Halaman mata diklat dapat diakses melalui menu mata diklat ketika pengguna sudah login, di dalam menu mata diklat sesuai gambar 3.16 terdapat list mata diklat dan untuk menambah datanya dengan memilih tombol tambah data.



Gambar 3.16 Halaman Mata Diklat

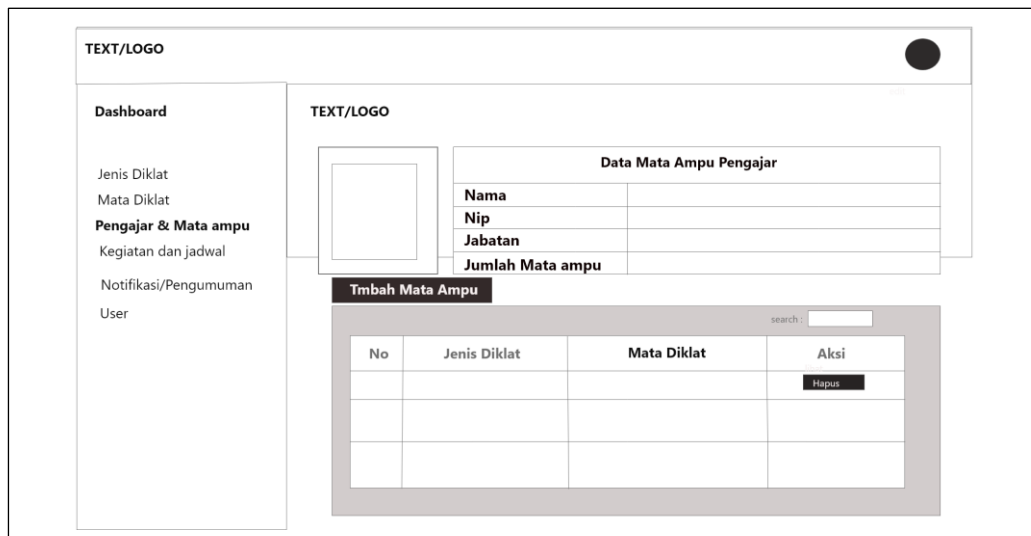
5. Halaman Web Data Pengajar

Halaman data pengajar dapat diakses melalui menu pengajar dan mata ampu ketika pengguna sudah login, di dalam menu sesuai gambar 3.17 terdapat list pengajar dan untuk menambah datanya dengan memilih tombol tambah data.



Gambar 3.17 Halaman web data pengajar

Selain menambah data pengajar, didalam menu pengajar terdapat menu mata ampu seperti gambar 3.18 berikut :

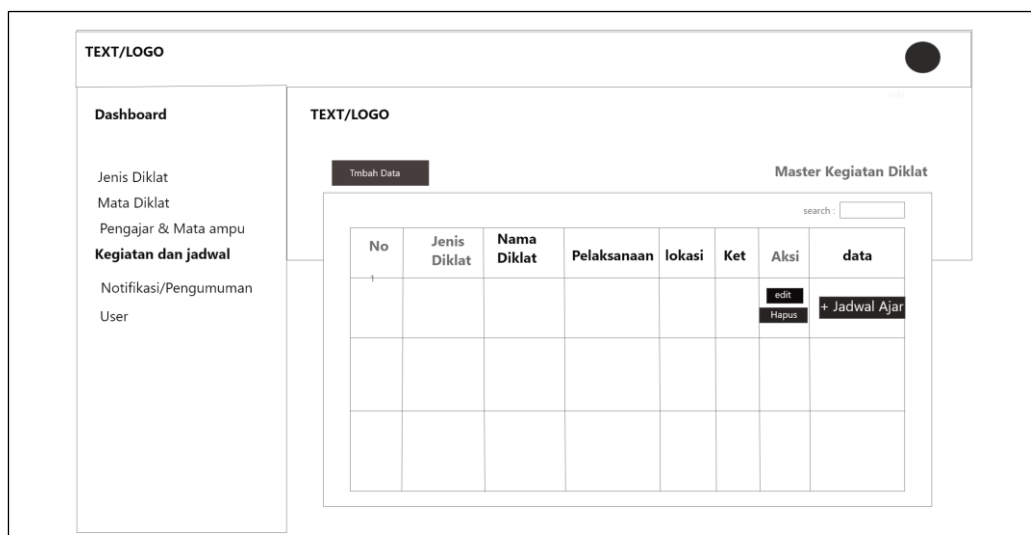


Gambar 3.18 Halaman web pengajar dan mata ampu

Di dalam menu list mata ampu terdapat menu tambah mata ampu untuk menambah data mata ampu, kemudian menu *search* untuk pencarian.

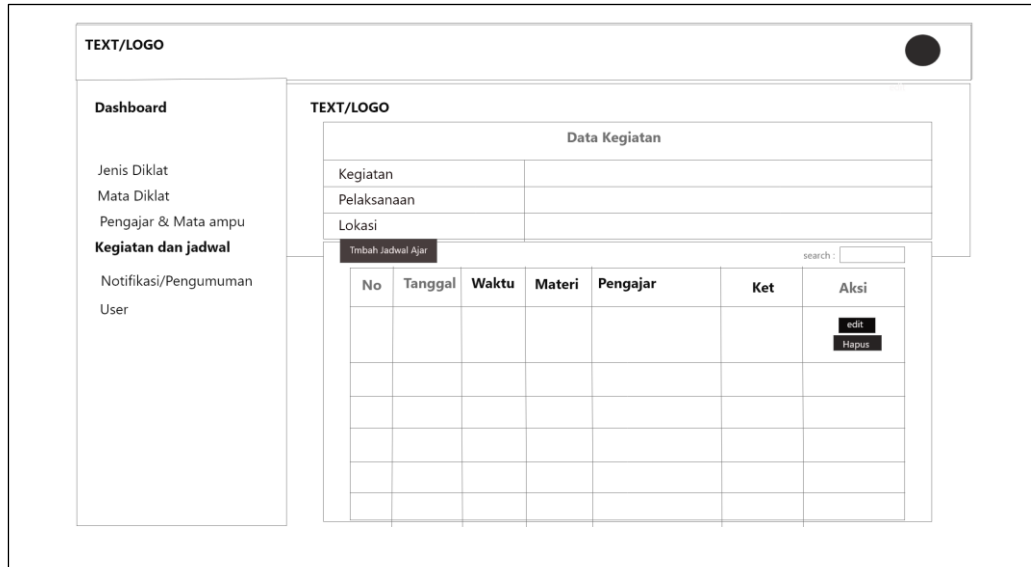
6. Halaman Web Data Kegiatan dan Jadwal

Halaman data kegiatan dapat diakses melalui menu kegiatan dan jadwal ketika pengguna sudah login, di dalam menu sesuai gambar 3.19 terdapat menu tambah data menambah datanya.



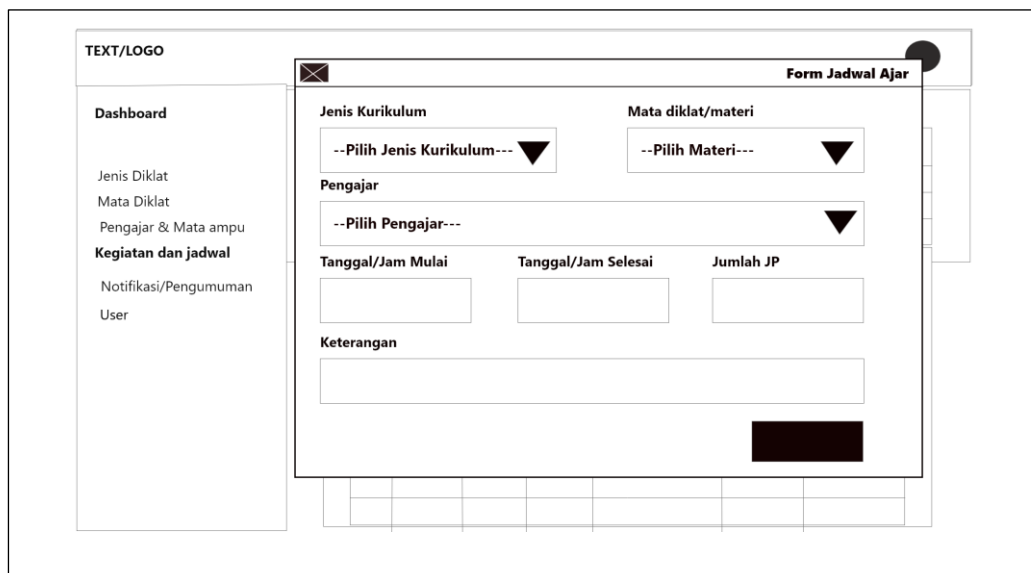
Gambar 3.19 Halaman kegiatan dan jadwal

Di dalam menu kegiatan, pengguna yang sudah login dapat melihat jadwal ajar melalui menu jadwal ajar seperti gambar 3.20, didalam halaman jadwal ajar terdapat menu tambah data dan *search* untuk pencarian data.



Gambar 3.20 Halaman Jadwal Ajar

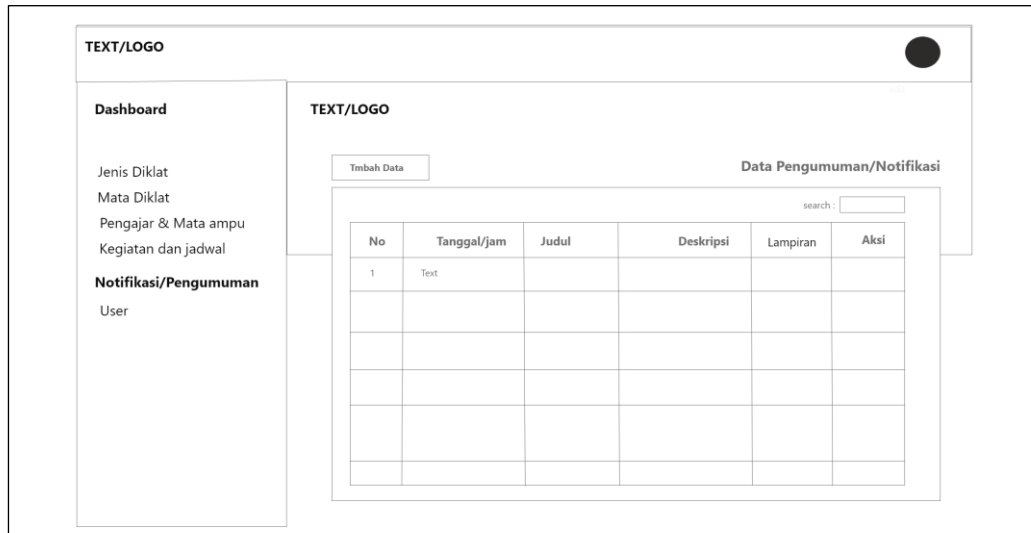
Di dalam daftar jadwal terdapat menu tambah jadwal ajar untuk menambah data jadwal ajar, seperti gambar 3.21 berikut :



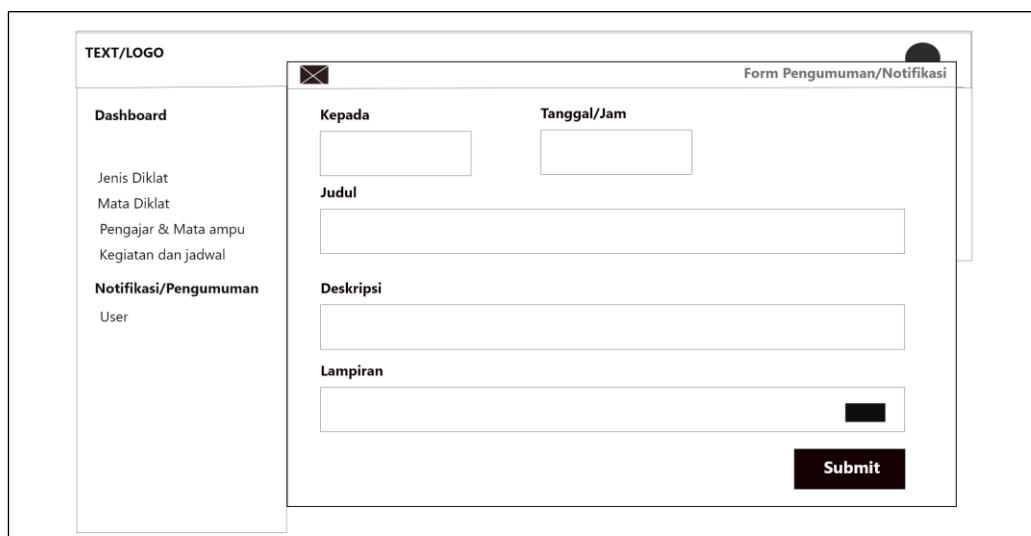
Gambar 3.21 Halaman Tambah Jadwal Ajar

7. Halaman Web Notifikasi/Pengumuman

Halaman data Notifikasi/Pengumuman dapat diakses melalui menu Notifikasi/Pengumuman ketika pengguna sudah login, di dalam menu sesuai gambar 3.22 terdapat menu tambah data menambahkan datanya.



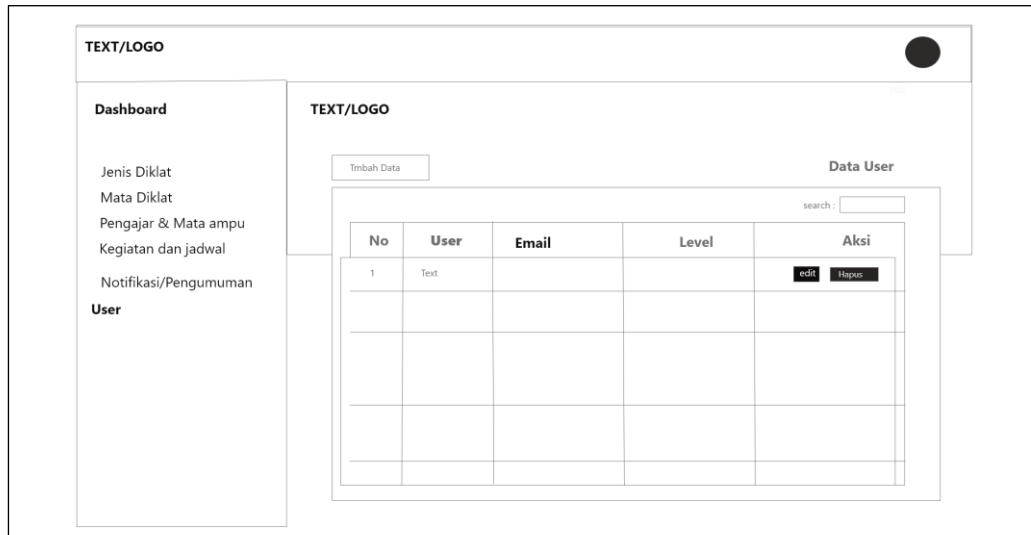
Gambar 3.22 Halaman Notifikasi/Pengumuman



Gambar 3.23 Halaman tambah Notifikasi/Pengumuman

8. Halaman Web Data User

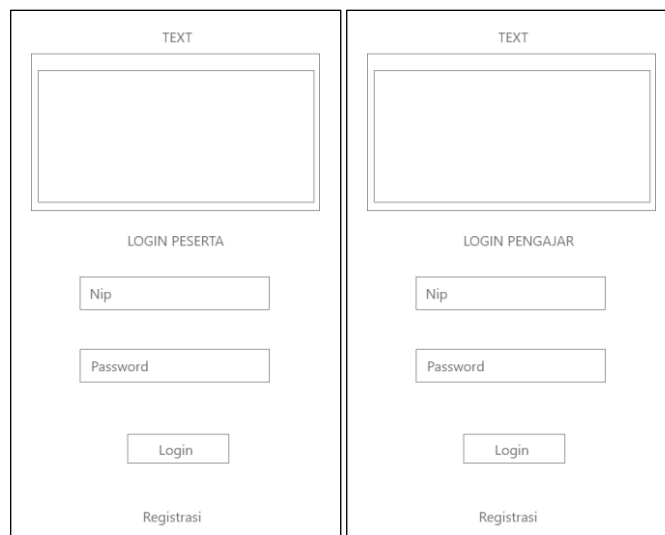
Halaman data user dapat diakses melalui menu User ketika pengguna sudah login, di dalam menu sesuai gambar 3.24 terdapat menu tambah data menambah data user.



Gambar 3.24 Data user

3.6.2. Desain Antar Muka Aplikasi Mobile

1. Halaman Login aplikasi mobile



Gambar 3.25 Halaman Login Aplikasi *Mobile*

Gambar 3.25 menggambarkan halaman login peserta dan pengajar, untuk dapat masuk ke menu utama dengan menginputkan nip dan password yang telah didaftarkan.

2. Halaman Registrasi Peserta

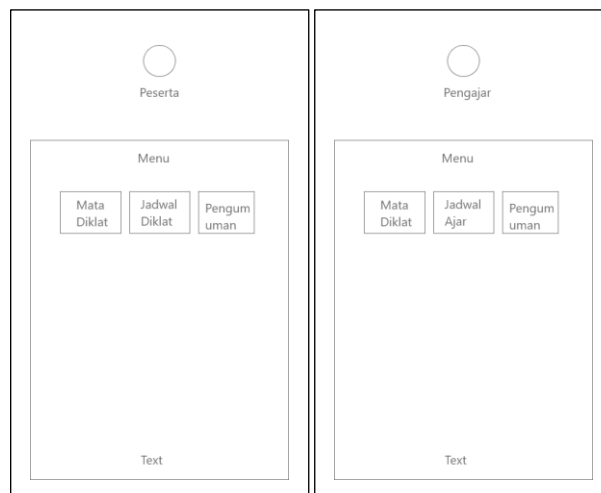


The image shows a mobile application interface for participant registration. At the top, there is a header labeled 'Peserta'. Below it, a 'Register' section is enclosed in a box. This section contains five input fields: 'Nip', 'Nama', 'Pilih tahun', 'Diklat yang diikuti', and 'Password'. At the bottom of the 'Register' section is a 'Login' button.

Gambar 3.26 Halaman Registrasi Peserta

Gambar 3.26 menggambarkan halaman registrasi peserta, terdapat *field* yang harus di isi diantaranya nip, nama, tahun, diklat yang diikuti serta password.

3. Halaman Dashboard Aplikasi mobile



Gambar 3.27 Halaman Dashboard Aplikasi mobile

Gambar 3.27 menggambarkan halaman dashboard peserta, didalam dashboard peserta ini terdapat beberapa menu yang didapat ketika pengguna sudah login yaitu menu mata diklat, menu jadwal diklat dan menu pengumuman.

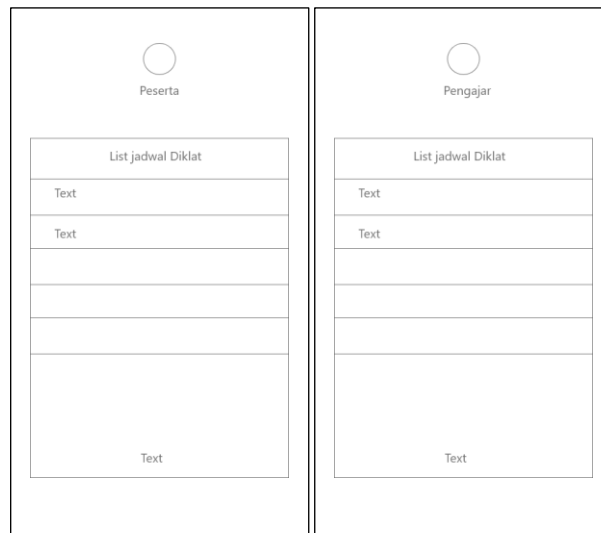
4. Halaman list mata diklat aplikasi mobile



Gambar 3.28 Halaman List Mata diklat

Gambar 3.28 menggambarkan halaman *list* mata diklat ketika pengguna yang sudah login memilih menu mata diklat, mata diklat yang tampil adalah mata diklat sesuai dengan kelas yang diikutinya.

5. Halaman Jadwal Diklat Aplikasi Mobile



Gambar 3.29 Halaman jadwal diklat

Gambar 3.29 menggambarkan halaman *list* jadwal diklat ketika pengguna yang sudah login memilih menu jadwal diklat, jadwal diklat yang tampil adalah jadwal diklat sesuai dengan kelas yang diikutinya.

6. Halaman Pengumuman dan notifikasi



Gambar 3.30 Halaman pengumuman

Gambar 3.30 menggambarkan halaman *list* pengumuman ketika pengguna yang sudah login memilih menu pengumuman, pengumuman yang dikirim pertama kali akan tampil diatas layar berupa notifikasi pada *handphone* pengguna.



Gambar 3.31 Halaman notifikasi

3.7. Uji Coba

Metode uji coba pada aplikasi ini menggunakan *black box testing* untuk menemukan kesalahan dalam beberapa kategori antara lain :

1. Fungsi-fungsi dari inputan data jadwal.
2. Kesalahan interface.
3. Kesalahan dalam akses database.