

BAB IV UJI COBA DAN PEMBAHASAN

4.1 Analisis Kebutuhan

Mengidentifikasi kebutuhan fungsional dan non-fungsional dari sistem informasi surat berbasis *web* di Universitas Nasional Karangturi.

a. Analisis Kebutuhan Fungsional

Tabel 4. 1 Analisis kebutuhan fungsional

No	Deskripsi
1.	Sistem dapat menampilkan menu <i>login</i> dan <i>logout</i>
2.	Sistem dapat menampilkan halaman <i>dashboard</i>
3.	Sistem dapat menampilkan dan menyimpan data <i>master jabatan</i>
4.	Sistem dapat menampilkan dan menyimpan data <i>master users</i>
5.	Sistem dapat menampilkan dan menyimpan data kota
6.	Sistem dapat menampilkan dan menyimpan data surat masuk
7.	Sistem dapat menampilkan dan menyimpan data surat keluar
8.	Sistem dapat menampilkan dan cetak <i>file</i> laporan
9.	Sistem dapat mencari <i>file</i> data yang di cari

b. Analisis Kebutuhan Non Fungsional

Tabel 4. 2 Analisis kebutuhan non fungsional

No	Deskripsi
1.	Sistem dapat diakses dengan perangkat PC maupun laptop
2.	Sistem dapat diakses menggunakan banyak <i>browser (google chrome, mozilla firefox dll.</i>
3.	Sistem memiliki penyimpanan <i>database server mysql</i>
4.	Sistem diakses dengan melakukan <i>login</i>
5.	Sistem dapat digunakan di mana saja

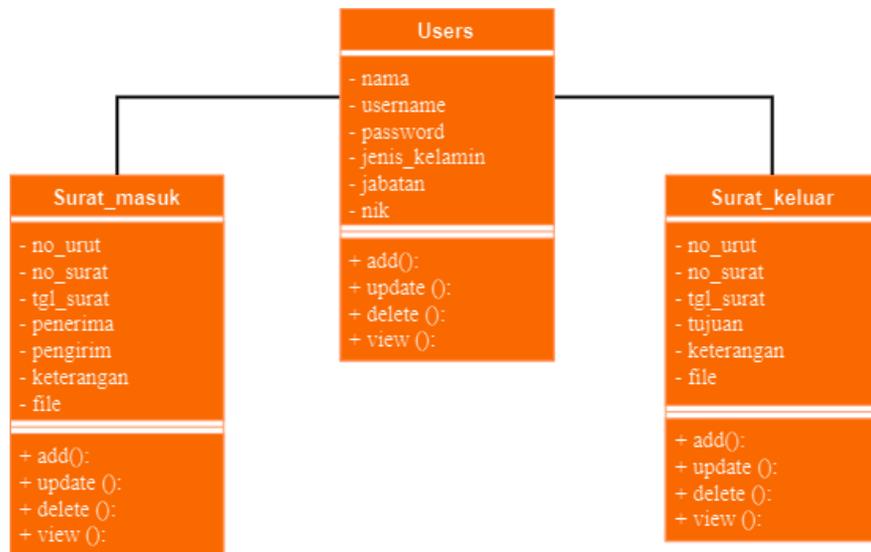
4.2 Perancangan Desain Sistem

Dalam perancangan sistem informasi surat, pengembang melaksanakan serangkaian langkah untuk merancang, membangun, dan mengimplementasikan

sistem informasi surat yang sesuai dengan kebutuhan Universitas Nasional Karangturi. Kebutuhan dari perancangan sistem informasi surat ini adalah untuk mengatasi permasalahan yang muncul akibat pengelolaan surat secara manual, seperti kesulitan dalam mencari kembali surat masuk dan keluar, kebutuhan banyak kertas yang harus diisi, serta risiko terhadap keamanan dan kerusakan berkas.

a. Class Diagram

Pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *Class diagram* digunakan untuk merepresentasikan *class* yang ada dalam sistem informasi surat. Setiap *class* dalam *diagram* ini mewakili objek dalam sistem seperti surat dan *users*. *Class diagram* memperlihatkan atribut-atribut dan metode-metode yang dimiliki oleh setiap kelas, serta hubungan dan asosiasi antar *class*. Dengan menggunakan *class diagram*, pengembang dapat memahami struktur dan hubungan antar komponen dalam sistem informasi surat, sehingga memudahkan dalam perancangan dan pengembangan sistem yang sesuai dengan kebutuhan dan harapan pengguna di lingkungan Universitas Nasional Karangturi.

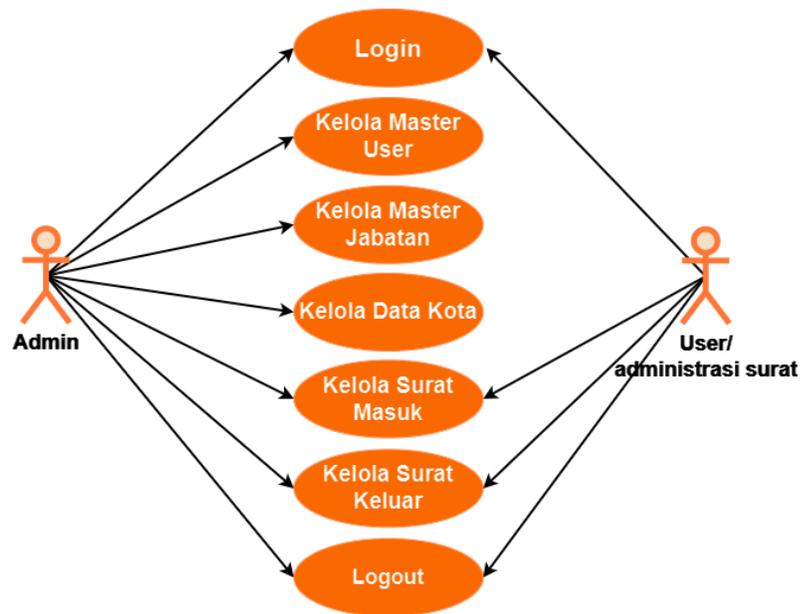


Gambar 4. 1 Class Diagram

Dalam analisis dan perancangan sistem surat, terdapat dua *class diagram* utama, yaitu *class surat_masuk* dan *surat_keluar*. *Class users* memiliki hubungan langsung dengan *class surat_masuk* dan *surat_keluar*.

b. Use Case Diagram

Pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *use case diagram* digunakan untuk mengidentifikasi, menggambarkan fungsi dan fitur yang dapat dilakukan oleh pengguna ketika berinteraksi dengan sistem informasi surat berbasis *web*. *Diagram* ini mencakup beberapa skenario penggunaan yang menggambarkan aktivitas saat pengguna menggunakan sistem untuk mengirim, menerima, mencari, dan mengelola surat secara *online*. *Use case diagram* membantu memvisualisasikan interaksi antara pengguna dan sistem informasi surat dengan lebih jelas, sehingga memudahkan perancangan dan pengembangan sistem yang sesuai dengan kebutuhan dan harapan pengguna di lingkungan Universitas Nasional Karangturi.

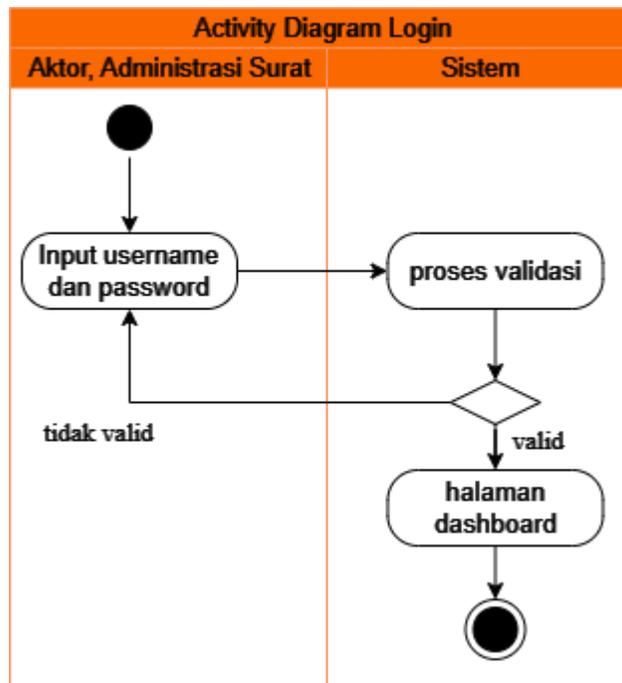


Gambar 4. 2 Use case diagram

Use case diagram di atas menampilkan dua aktor, yaitu admin dan administrasi surat. Admin memiliki akses ke fungsi *master user*, *master jabatan*, mengelola surat masuk, mengelola surat keluar, dan *logout*. Administrasi surat memiliki akses ke fungsi *login*, mengelola surat masuk, mengelola surat keluar, dan *logout*.

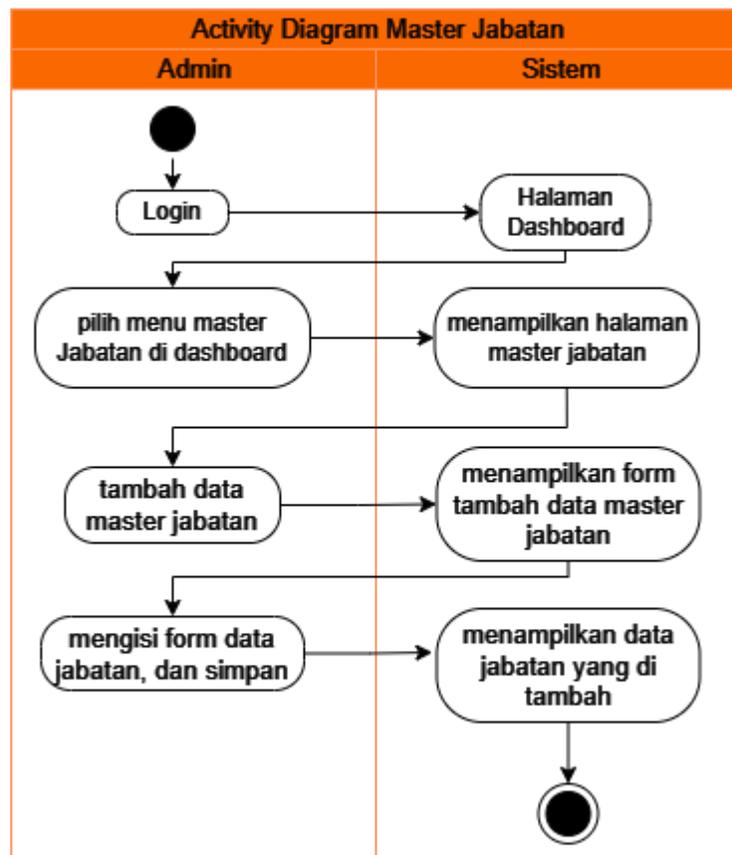
c. Activity Diagram

Pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *activity diagram* dapat digunakan untuk menggambarkan langkah-langkah dalam proses pengelolaan surat secara *visual*, termasuk aktivitas seperti pengiriman, penerimaan, dan pengelolaan surat secara *online*. *Diagram* ini membantu memvisualisasikan interaksi antara pengguna dan sistem dalam mengelola surat dengan efektif dan efisien.



Gambar 4. 3 Activity diagram login

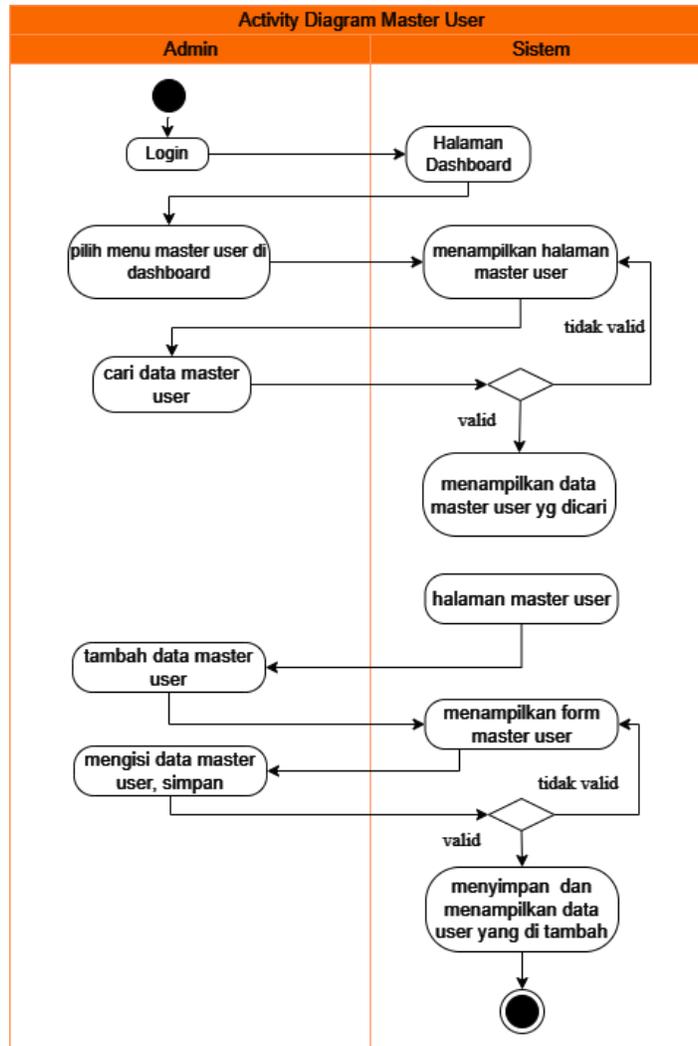
Activity diagram login menggambarkan aktivitas *login* ke *web* sistem informasi surat dengan cara memasukkan *username* dan *password* kemudian sistem memproses dan memvalidasi data, apabila data yang dimasukkan *valid* maka sistem akan menampilkan halaman *dashboard*, dan jika data yang dimasukkan salah maka tetap berada di halaman *login* dan ada notifikasi kesalahan dalam memasukkan data.



Gambar 4. 4 Activity diagram master jabatan

Activity diagram master jabatan menggambarkan langkah-langkah dalam sistem informasi untuk melakukan tambah data *master* jabatan. Ketika pengguna klik tombol tambah data jabatan, sistem akan menampilkan *form* untuk memasukkan data *master* jabatan. Kemudian *admin* mengisi *form* data jabatan lalu simpan, maka sistem akan

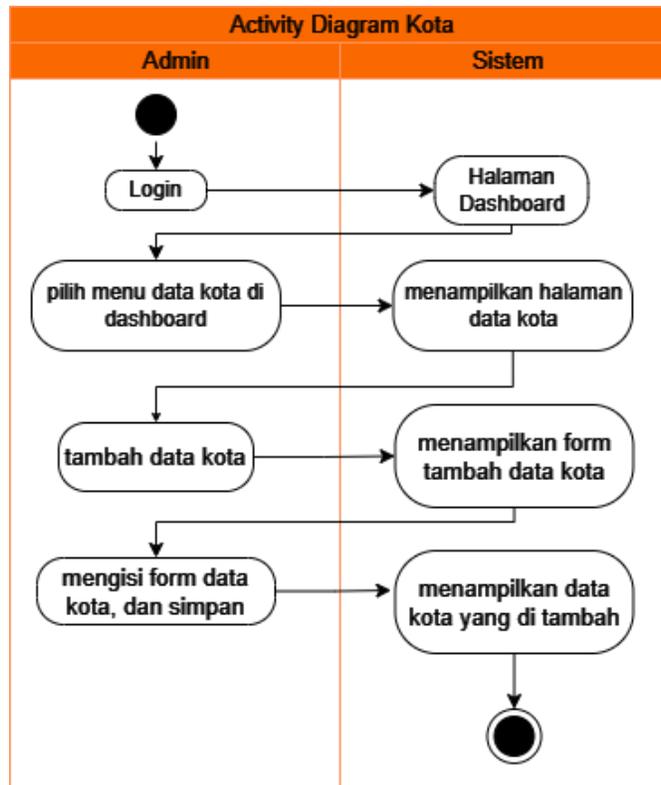
menampilkan data jabatan yang di tambah dan akan menampilkan notifikasi data jabatan berhasil ditambahkan.



Gambar 4. 5 Activity diagram master user

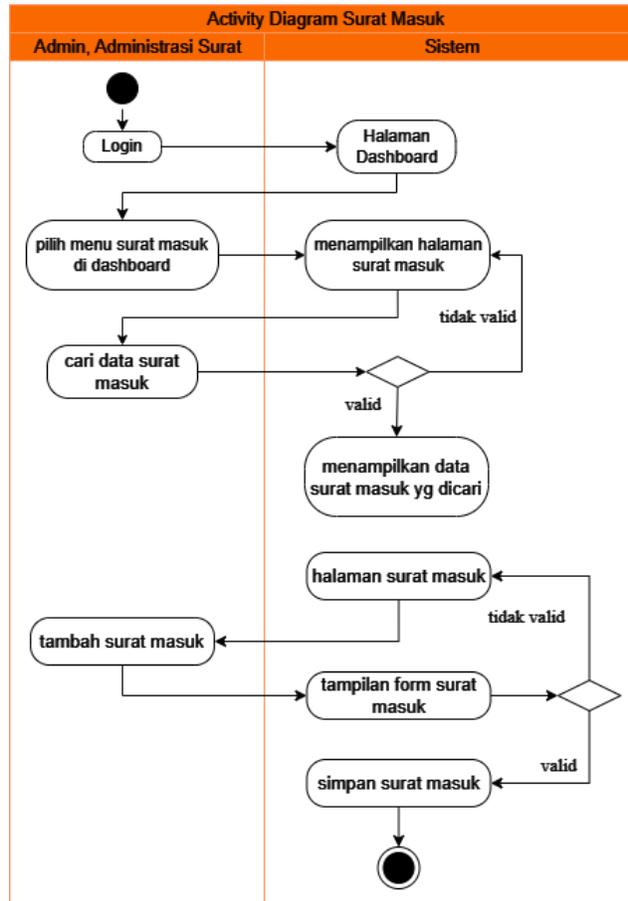
Activity diagram master user menggambarkan langkah-langkah dalam sistem informasi untuk melakukan pencarian data *master user*. Ketika pengguna mencari data *master user*, sistem akan memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, sistem akan menampilkan halaman dengan data yang dicari. Namun, jika data tidak *valid*, pengguna akan kembali ke halaman *master user*. Selanjutnya, untuk

menambahkan data *user*, sistem akan menampilkan *form* untuk memasukkan data *master user*. Data yang dimasukkan akan divalidasi, jika data lengkap, informasi surat masuk dapat disimpan, jika data yang dimasukkan tidak lengkap notifikasi kesalahan muncul di halaman *form* tambah data *user*.



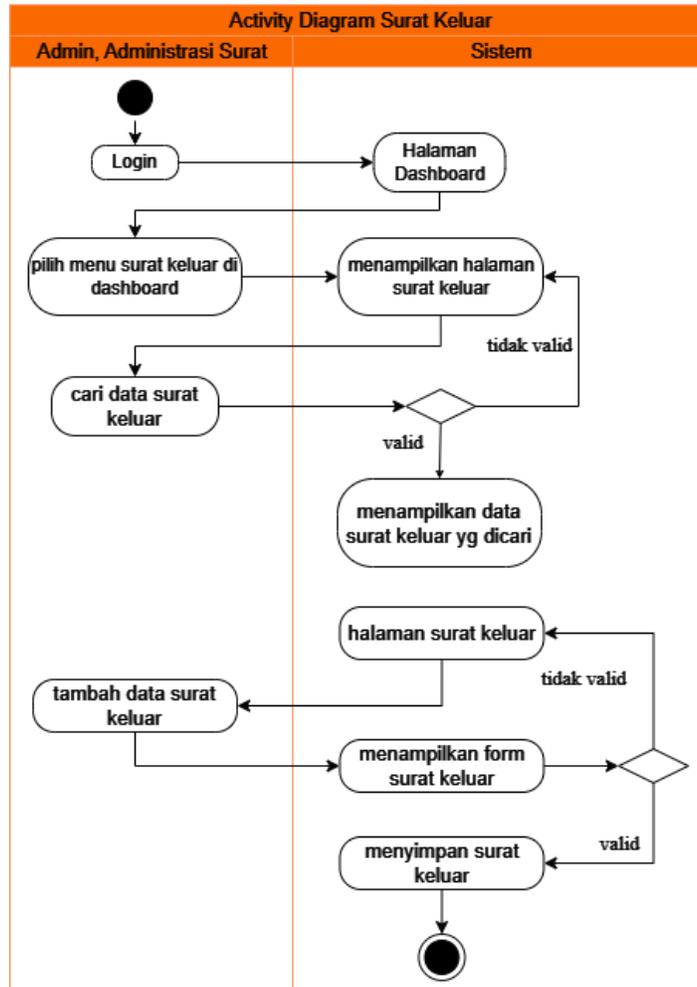
Gambar 4. 6 Activity diagram data kota

Activity diagram data kota menggambarkan langkah-langkah dalam sistem informasi untuk melakukan tambah data kota. Ketika pengguna klik tombol tambah data kota, sistem akan menampilkan *form* untuk memasukkan data *master kota*. Kemudian *admin* mengisi *form* data kota lalu simpan, maka sistem akan menampilkan data kota yang ditambah dan akan menampilkan notifikasi data kota berhasil ditambahkan.



Gambar 4. 7 Activity diagram surat masuk

Activity diagram surat masuk menggambarkan langkah-langkah dalam sistem informasi untuk melakukan pencarian data surat masuk. Ketika pengguna mencari data surat masuk, sistem akan memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, sistem akan menampilkan halaman dengan data yang dicari. Namun, jika data tidak *valid*, pengguna akan kembali ke halaman surat masuk. Selanjutnya, untuk menambahkan data surat masuk, sistem akan menampilkan *form* untuk memasukkan data surat masuk. Data yang dimasukkan akan divalidasi, jika data lengkap, informasi surat masuk dapat disimpan.



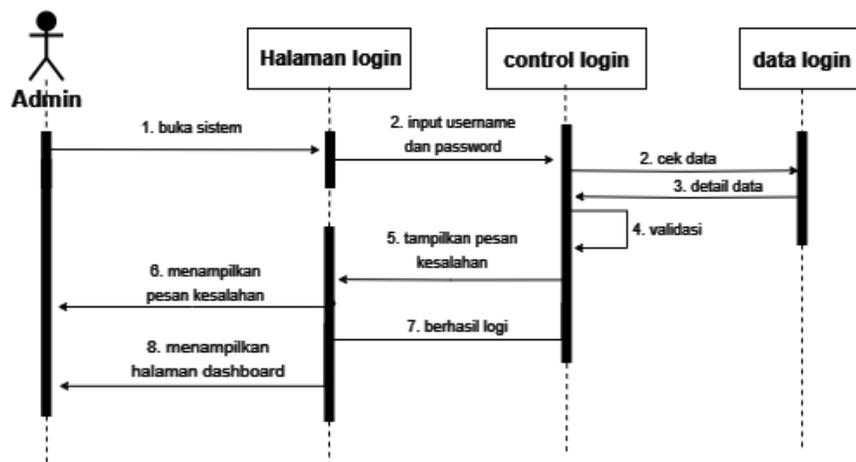
Gambar 4. 8 Activity diagram surat keluar

Activity diagram surat keluar menggambarkan langkah-langkah dalam sistem informasi untuk melakukan pencarian data surat keluar. Ketika pengguna mencari data surat keluar, sistem akan memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, sistem akan menampilkan halaman dengan data yang dicari. Namun, jika data tidak *valid*, pengguna akan kembali ke halaman surat keluar. Selanjutnya, untuk menambahkan data surat keluar, sistem akan menampilkan *form* untuk memasukkan data surat keluar. Data yang dimasukkan akan divalidasi, jika data lengkap, informasi surat keluar dapat disimpan.

d. *Sequence Diagram*

Dalam pengembangan sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, *Sequence diagram* digunakan untuk menggambarkan interaksi antara berbagai komponen dalam sistem informasi surat. *Diagram* ini membantu memahami alur logika program serta visualisasi pesan-pesan antar objek selama pengelolaan surat. Tim pengembang menggunakan *diagram* ini untuk memeriksa dan memvalidasi perilaku sistem dalam situasi tertentu, sehingga perbaikan dapat dilakukan jika ada kekurangan ataupun masalah dalam interaksi komponen. *Sequence diagram* menjadi alat efektif untuk merancang, memodelkan, dan memahami interaksi dalam sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, serta meningkatkan efisiensi dan efektivitas pengelolaan surat. Berikut beberapa gambar *sequence diagram* sistem informasi surat berbasis *web* di Universitas Nasional Karangturi :

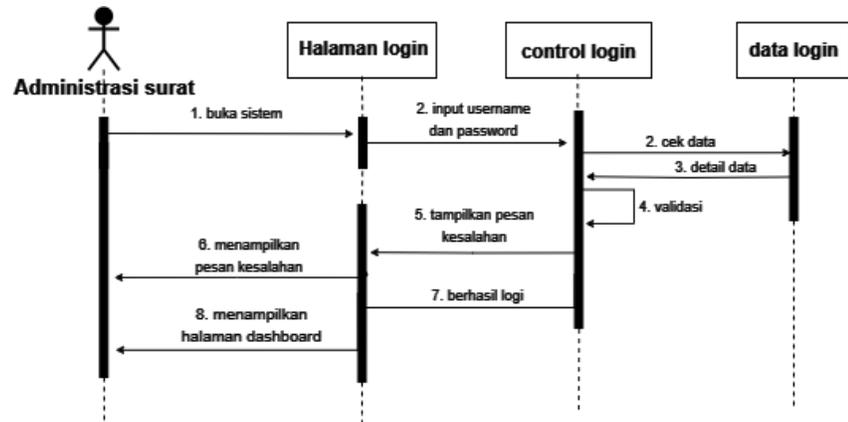
1) *Sequence diagram login*



Gambar 4. 9 *Sequence diagram login admin*

Sequence diagram diatas menampilkan admin dalam melakukan *login*. Sistem membutuhkan *username* dan *password*

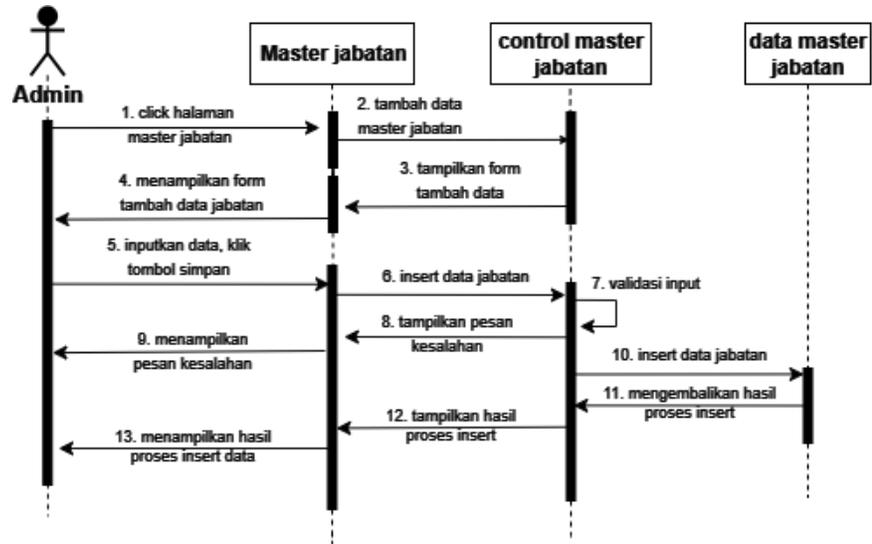
untuk *login* ke sistem informasi surat, apabila data *username* dan *password* yang di validasi benar maka pengguna akan dibawa ke halaman *dashboard*, dan apabila data yang divalidasi salah maka sistem akan memberikan notifikasi kesalahan di halaman *login*.



Gambar 4. 10 Sequence diagram login administrasi surat

Sequence diagram di atas menampilkan administrasi surat (*user*) dalam melakukan *login*. Sistem membutuhkan *username* dan *password* untuk *login* ke sistem informasi surat, apabila data *username* dan *password* yang di validasi benar maka pengguna akan dibawa ke halaman *dashboard*, dan apabila data yang divalidasi salah maka sistem akan memberikan notifikasi kesalahan di halaman *login*.

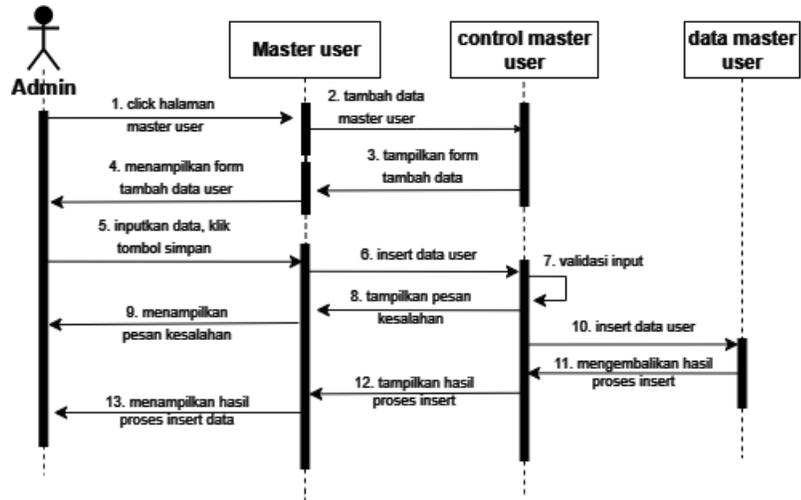
2) Sequence diagram master jabatan



Gambar 4. 11 Sequence diagram master jabatan

Sequence diagram di atas menampilkan satu aktor, yaitu *admin*, karena hanya *admin* yang memiliki akses ke halaman *master jabatan*. Aktivitas *admin* dalam *diagram* tersebut menggambarkan proses penambahan data jabatan ke *master jabatan*. Pertama, *admin* mengirim permintaan untuk menambahkan data ke *control master jabatan*. Kemudian, *control master jabatan request* tampilkan *form* tambah data ke *view* surat masuk untuk menampilkan *form* penambahan data, dan halaman *form* penambahan data ditampilkan. Setelah itu, *admin* memasukkan data dan mengklik tombol Simpan. Setelah itu, *master jabatan* melakukan penambahan data jabatan ke *control master jabatan* dan juga memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data jabatan akan di *insert* ke *data master jabatan (database)*. Selanjutnya, *database* akan mengembalikan hasil data yang dimasukkan dan *view master jabatan* akan menampilkan data yang di *input*.

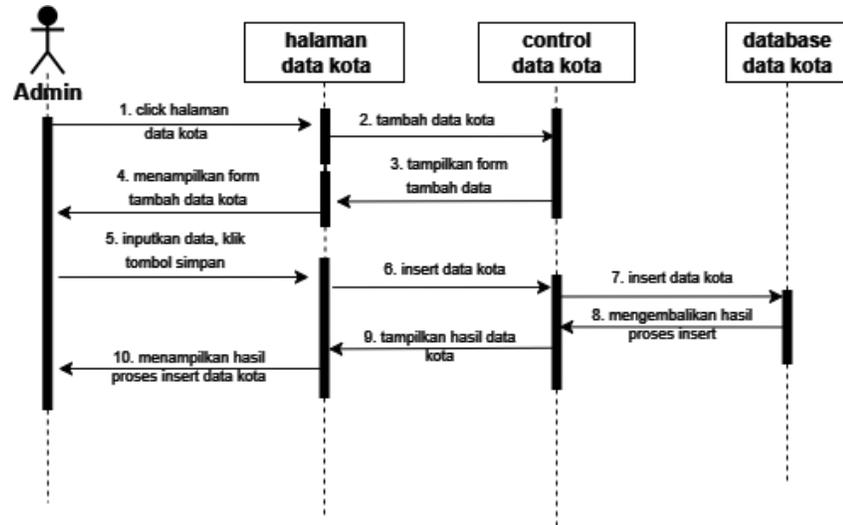
3) Sequence diagram master users



Gambar 4. 12 Sequence diagram master users

Sequence diagram di atas menampilkan satu aktor, yaitu *admin*, karena hanya *admin* yang memiliki akses ke halaman *master users*. Aktivitas *admin* dalam diagram tersebut menggambarkan proses penambahan data *user* ke *master users*. Pertama, *admin* mengirim permintaan untuk menambahkan data ke kontrol *master users*. Kemudian, kontrol *master users* melakukan eksekusi ke view *master users* untuk menampilkan *form* penambahan data, dan halaman *form* penambahan data ditampilkan. Setelah itu, *admin* memasukkan data dan mengklik tombol Simpan. Setelah itu, *master users* melakukan penambahan data jabatan ke kontrol *master users* dan juga memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data *users* akan di *insert* ke data *master users* (*database*). Selanjutnya, *database* akan menyimpan data dan menampilkannya ke view *master users*, apabila data yang di masukan salah maka di halaman *master users* akan menampilkan informasi kesalahan dalam memasukkan data.

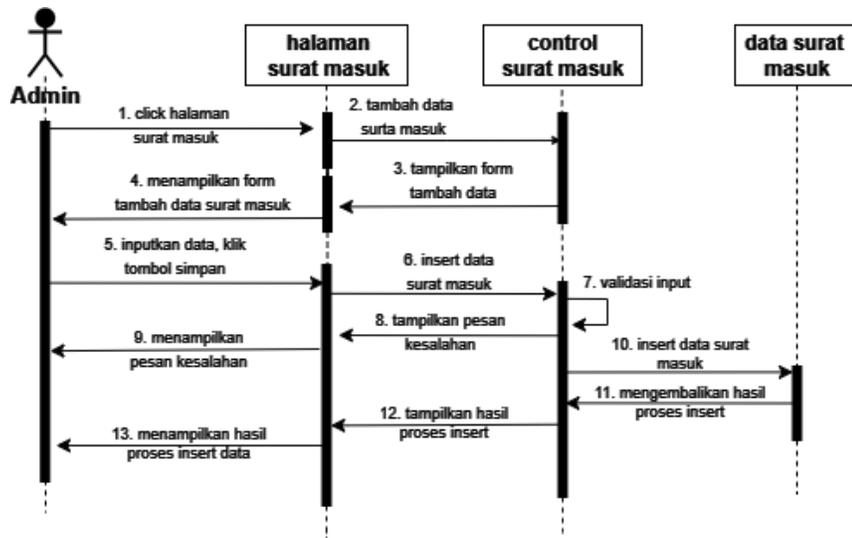
4) *Sequence diagram* data kota



Gambar 4. 13 *Sequence diagram* data kota

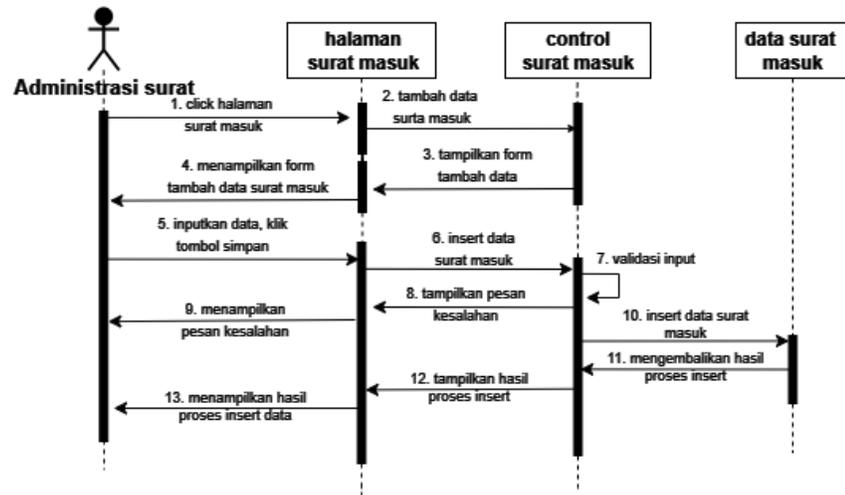
Sequence diagram di atas menampilkan satu aktor, yaitu *admin*, karena hanya *admin* yang memiliki akses ke halaman data kota. Aktivitas *admin* dalam *diagram* tersebut menggambarkan proses penambahan data kota ke sistem. Pertama, *admin* mengirim permintaan untuk menambahkan data ke kontrol data kota. Kemudian, kontrol data kota melakukan eksekusi ke *view* data kota untuk menampilkan *form* penambahan data, dan halaman *form* penambahan data ditampilkan. Setelah itu, *admin* memasukkan data dan mengklik tombol simpan. Data kota melakukan penambahan data kota ke kontrol data kota dan juga memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data kota akan di insert ke *database* data kota. Selanjutnya, *database* akan mengembalikan hasil data yang di *input*, dan menampilkannya ke *view* data kota.

5) *Sequence diagram* surat masuk



Gambar 4. 14 *Sequence diagram* surat masuk admin

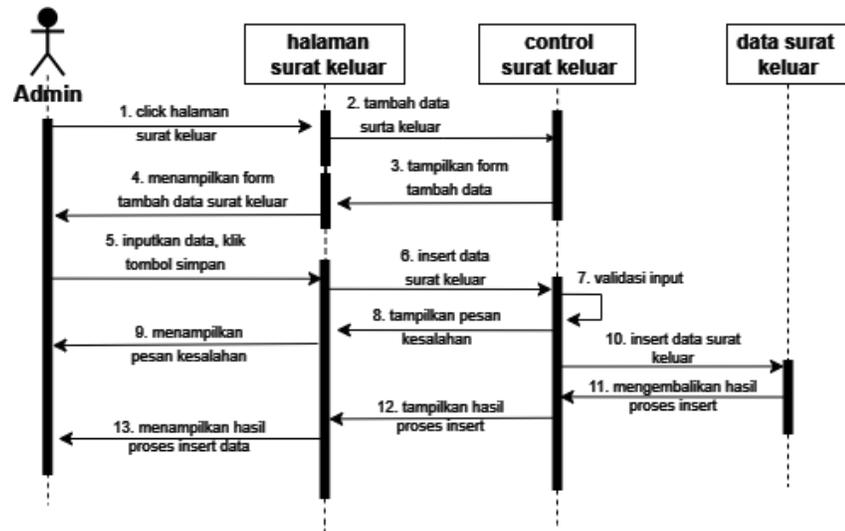
Sequence diagram di atas menampilkan aktor *admin* menggambarkan proses penambahan data surat masuk. Pertama, pengguna mengirim permintaan untuk menambahkan data surat masuk ke kontrol surat masuk. Kemudian, kontrol surat masuk *request* tampilkan *form* tambah data ke *view* surat masuk untuk menampilkan *form* penambahan data, dan halaman *form* penambahan data ditampilkan. Setelah itu, pengguna memasukkan data dan mengklik tombol simpan. Setelah itu, kontrol surat masuk memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data yang dimasukkan akan di *insert* ke data surat masuk (*database*). Selanjutnya, *database* akan menyimpan data dan menampilkannya ke *view* surat masuk. Jika terdapat kesalahan dalam memasukkan data, sistem akan menampilkan informasi kesalahan kepada pengguna.



Gambar 4. 15 Sequence diagram surat masuk administrasi surat

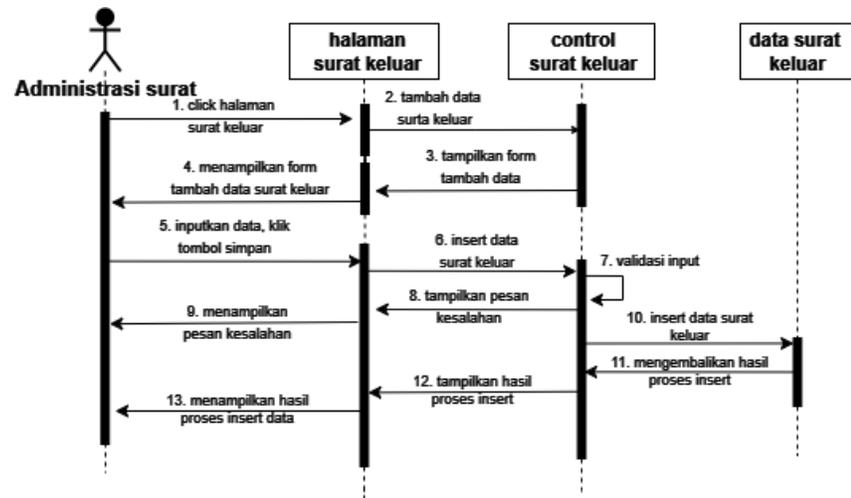
Sequence diagram di atas menampilkan aktor *administrasi surat (user)* menggambarkan proses penambahan data surat masuk. Pertama, pengguna mengirim permintaan untuk menambahkan data surat masuk ke kontrol surat masuk. Kemudian, kontrol surat masuk *request* tampilkan *form* tambah data ke *view* surat masuk untuk menampilkan *form* penambahan data, dan halaman *form* penambahan data ditampilkan. Setelah itu, pengguna memasukkan data dan mengklik tombol simpan. Setelah itu, kontrol surat masuk memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data yang dimasukkan akan di *insert* ke data surat masuk (*database*). Selanjutnya, *database* akan menyimpan data dan menampilkannya ke *view* surat masuk. Jika terdapat kesalahan dalam memasukkan data, sistem akan menampilkan informasi kesalahan kepada pengguna.

6) *Sequence diagram* surat keluar



Gambar 4. 16 *Sequence diagram* surat keluar admin

Sequence diagram di atas menampilkan *admin* menggambarkan proses penambahan data surat keluar. Pertama, pengguna mengirim permintaan untuk menambahkan data surat keluar ke kontrol surat keluar. Kemudian, kontrol surat keluar mengirim pesan untuk menampilkan *form* tambah data ke surat keluar dan halaman *form* penambahan data ditampilkan. Setelah itu, pengguna memasukkan data dan mengklik tombol simpan, kemudian kontrol surat keluar memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data yang dimasukkan akan di *insert* ke *database* surat keluar. Selanjutnya, sistem akan mengembalikan hasil data yang dimasukkan, dan surat keluar akan menampilkan data yang dimasukkan. Jika terdapat kesalahan dalam memasukkan data, sistem akan menampilkan informasi kesalahan kepada pengguna.



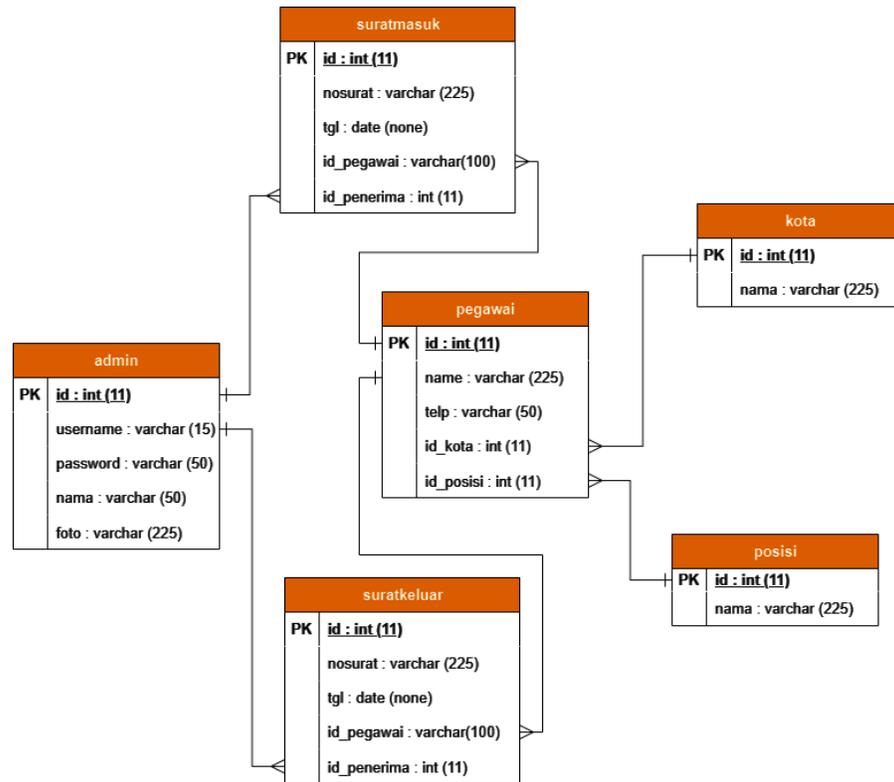
Gambar 4. 17 Sequence diagram surat keluar administrasi surat

Sequence diagram di atas menampilkan *administrasi surat (user)* menggambarkan proses penambahan data surat keluar. Pertama, pengguna mengirim permintaan untuk menambahkan data surat keluar ke kontrol surat keluar. Kemudian, kontrol surat keluar mengirim pesan untuk menampilkan *form* tambah data ke surat keluar dan halaman *form* penambahan data ditampilkan. Setelah itu, pengguna memasukkan data dan mengklik tombol simpan, kemudian kontrol surat keluar memvalidasi data yang dimasukkan. Jika data yang dimasukkan *valid*, data yang dimasukkan akan di *insert* ke *database* surat keluar. Selanjutnya, sistem akan mengembalikan hasil data yang dimasukkan, dan surat keluar akan menampilkan data yang dimasukkan. Jika terdapat kesalahan dalam memasukkan data, sistem akan menampilkan informasi kesalahan kepada pengguna.

4.3 Perancangan Database

Dalam penelitian ini, bahasa SQL akan digunakan untuk membuat *database*. Selain itu, *database* yang terkait akan dihubungkan agar tabel-tabel di

dalamnya dapat disinkronkan. Berikut merupakan *diagram* ERD yang ada dalam sistem informasi surat berbasis *web* di Universitas Nasional Karangturi.



Gambar 4. 18 Relasi antar tabel dalam database (ERD)

Diagram Entity Relationship (ERD) yang disajikan memiliki peran *visual* yang efisien dalam menggambarkan bagaimana entitas-entitas saling terkait dan berinteraksi dalam kerangka sistem informasi surat berbasis *web*. ERD ini menjadi alat yang berperan penting dalam proses perancangan dan pemahaman tentang struktur dasar data serta relasi yang diperlukan antar tabel, yang sangat esensial dalam pembuatan sistem informasi sistem informasi surat berbasis *web* di Universitas Nasional Karangturi. Berikut merupakan daftar tabel yang ada dalam sistem informasi surat berbasis *web* di Universitas Nasional Karangturi.

a. Tabel *Admin*

Tabel *admin* memiliki 5 field yaitu *id*, *username*, *password*, *nama* dan *foto*. Tabel ini digunakan untuk menyimpan data *users* untuk *login*. Berikut desain dari tabel kelamin.

Tabel 4. 3 Admin

<i>Field Name</i>	<i>Type Data</i>	<i>Size</i>
<i>id</i>	Int	11
<i>username</i>	<i>varchar</i>	15
<i>password</i>	<i>varchar</i>	50
<i>nama</i>	<i>varchar</i>	50
<i>foto</i>	<i>varchar</i>	225

b. Tabel Kota

Tabel kota memiliki 2 *field* yaitu *id*, dan *nama*. Tabel ini digunakan untuk menyimpan data kota. Berikut desain dari tabel kota.

Tabel 4. 4 Kota

<i>Field Name</i>	<i>Type Data</i>	<i>Size</i>
<i>id</i>	Int	11
<i>nama</i>	<i>varchar</i>	255

Tabel pegawai memiliki 7 *field* yaitu *id*, *nama*, *tlp*, *id_kota*, *id_kelamin*, *id_posisi* dan *status*. Tabel ini digunakan untuk menyimpan data-data dari pegawai (*users*). Berikut desain dari tabel pegawai.

Tabel 4. 5 Pegawai

<i>Field Name</i>	<i>Type Data</i>	<i>Size</i>
<i>id</i>	Int	11
<i>nama</i>	<i>varchar</i>	255
<i>telp</i>	<i>varchar</i>	225
<i>id_kota</i>	Int	11
<i>id_kelamin</i>	Int	11
<i>id_posisi</i>	Int	11

c. Tabel Posisi

Tabel posisi memiliki 2 *field* yaitu *id*, dan nama. Tabel ini digunakan untuk menyimpan data posisi *users*. Berikut desain dari tabel posisi.

Tabel 4. 6 Posisi

<i>Field Name</i>	<i>Type Data</i>	<i>Size</i>
<i>id</i>	Int	11
nama	<i>varchar</i>	255

d. Tabel Surat Masuk

Tabel surat masuk memiliki 6 *field* yaitu *id*, nosurat, tgl, *id_pegawai*, *id_penerima* dan status. Tabel ini digunakan untuk menyimpan data-data dari surat masuk. Berikut desain dari tabel surat masuk.

Tabel 4. 7 Surat masuk

<i>Field Name</i>	<i>Type Data</i>	<i>Size</i>
<i>id</i>	Int	11
nosurat	<i>varchar</i>	255
tgl	<i>date</i>	none
<i>id_pegawai</i>	<i>varchar</i>	100
<i>id_penerima</i>	Int	11

e. Tabel Surat Keluar

Tabel surat keluar memiliki 6 *field* yaitu *id*, nosurat, tgl, *id_pegawai*, *id_penerima* dan status. Tabel ini digunakan untuk menyimpan data-data dari surat. Berikut desain dari tabel surat keluar.

Tabel 4. 8 Surat keluar

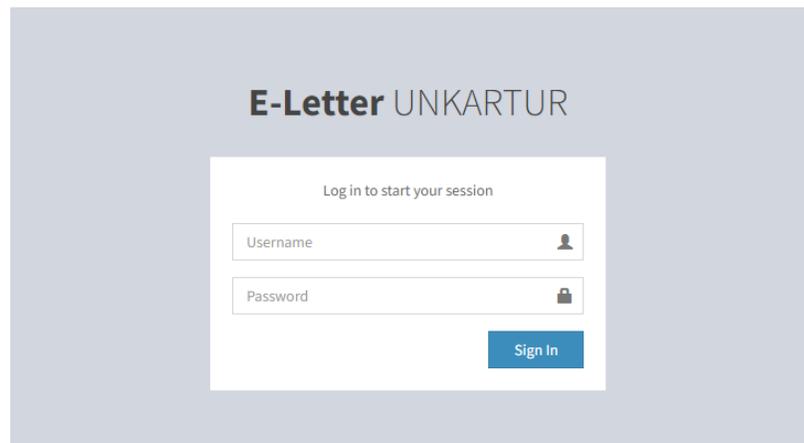
<i>Field Name</i>	<i>Type Data</i>	<i>Size</i>
<i>id</i>	Int	11
nosurat	<i>varchar</i>	255
tgl	<i>date</i>	none

<i>id_pegawai</i>	<i>varchar</i>	100
<i>id_penerima</i>	Int	11

4.4 Implementasi Antarmuka (*Interface*)

Implementasi desain antarmuka adalah langkah untuk mengimplementasikan desain antarmuka yang telah dirancang sebelumnya ke dalam sistem. Dalam penelitian terkait sistem informasi surat berbasis *web* di Universitas Nasional Karangturi, implementasi desain antarmuka berperan penting dalam menciptakan tampilan yang menarik dan mudah digunakan oleh pengguna. Dengan menerapkan desain antarmuka yang telah diuji dan dioptimalkan, sistem informasi surat dapat memberikan pengalaman pengguna yang lebih baik dan meningkatkan efisiensi dalam pengelolaan surat.

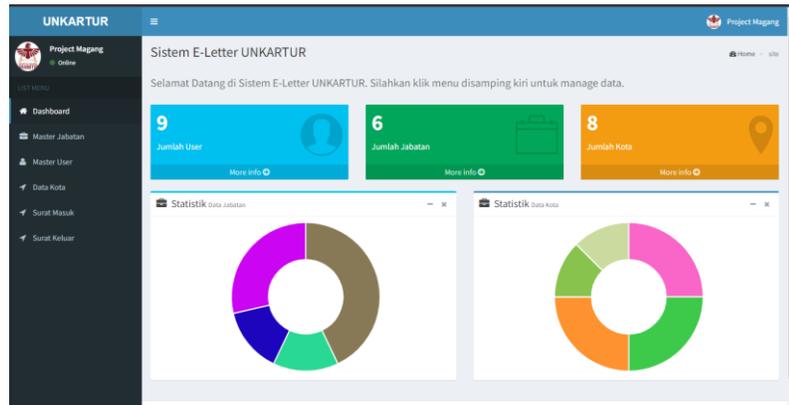
a. Implementasi Halaman *Login*



Gambar 4. 19 Implementasi halaman *login*

Tampilan untuk menu *login* yang perlu masukkan dari pengguna berupa *username* dan *password*. Data yang dimasukkan pengguna akan dicocokkan dengan data yang tersimpan di basis data. Jika data cocok proses akan dilanjutkan pada tampilan selanjutnya, jika tidak tampilan akan menampilkan pesan kesalahan.

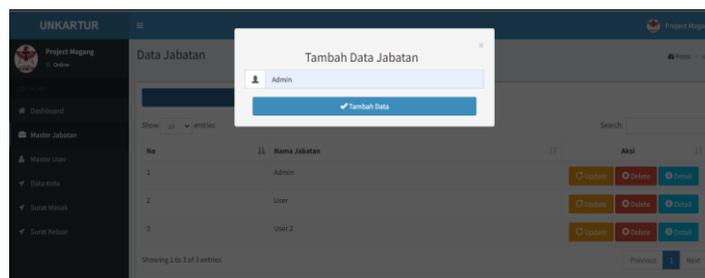
b. Implementasi Halaman *Dashboard*



Gambar 4. 20 Implementasi halaman *dashboard*

Tampilan halaman *dashboard* merupakan menu utama yang memuat menu seperti menu *master jabatan*, *master user*, data kota, surat masuk, dan surat keluar. Menu dapat dipilih oleh *user* melalui perangkat mouse.

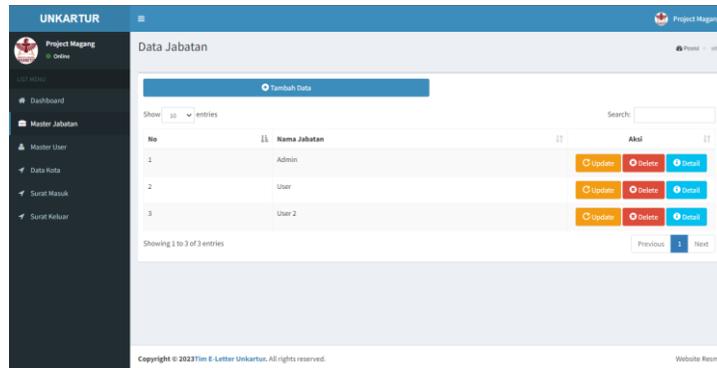
c. Implementasi *form* tambah data master jabatan



Gambar 4. 21 Halaman tambah data master jabatan

Tampilan halaman *form* tambah data master jabatan yang berisikan satu tabel level jabatan.

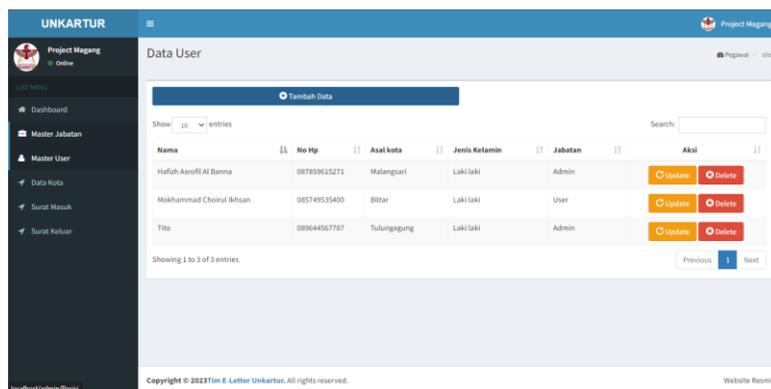
d. Implementasi Halaman Master Jabatan



Gambar 4. 22 Implementasi halaman master jabatan

Tampilan halaman menu jabatan merupakan menu yang digunakan untuk menghasilkan data jabatan yang terlibat dalam proses surat menyurat. Pada halaman master jabatan terdapat fitur yang menampilkan data seluruh pihak dengan detail jabatan dan aksi yang bisa dilakukan seperti edit dan hapus data, serta tambah data baru.

e. Implementasi Halaman *Master User*

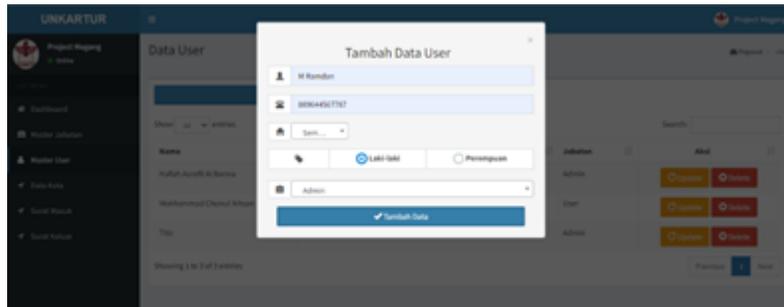


Gambar 4. 23 Implementasi halaman master users

Tampilan halaman menu master *users* merupakan menu yang digunakan untuk menghasilkan data user yang terlibat dalam proses surat menyurat. Pada halaman master *user* terdapat fitur yang menampilkan data

seluruh pihak dengan detail nama, jabatan, hp, kota, jabatan, dan aksi yang bisa dilakukan seperti edit dan hapus data, serta tambah data baru.

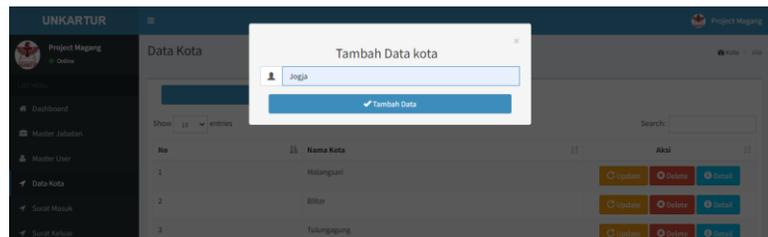
f. Implementasi Halaman *form* tambah data master *users*



Gambar 4. 24 Implementasi halaman *form* tambah data master *users*

Tampilan halaman *form* tambah data master *users* yang dapat memasukkan data master *users* berupa : nama, jabatan, no hp, kota, dan jabatan.

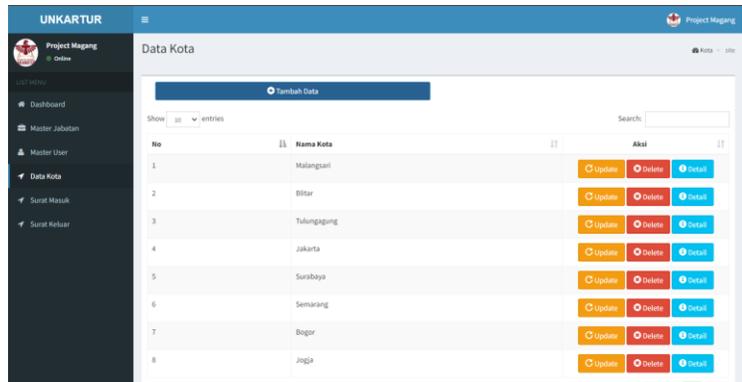
g. Implementasi Halaman *Form* Tambah Data Kota



Gambar 4. 25 Implementasi halaman *form* tambah data kota

Tampilan halaman *form* tambah data kota di atas dapat memasukkan data berupa nama kota.

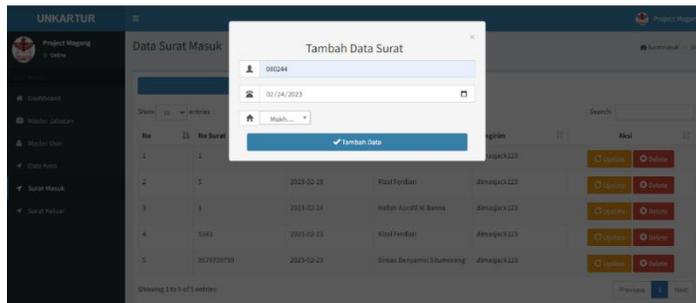
h. Implementasi Halaman Data Kota



Gambar 4. 26 Implementasi halaman data kota

Tampilan halaman menu data kota merupakan menu yang digunakan untuk menghasilkan kota yang terlibat dalam proses surat menyurat. Pada halaman data kota terdapat fitur yang menampilkan data seluruh kota dan aksi yang bisa dilakukan seperti edit dan hapus data, serta tambah data baru

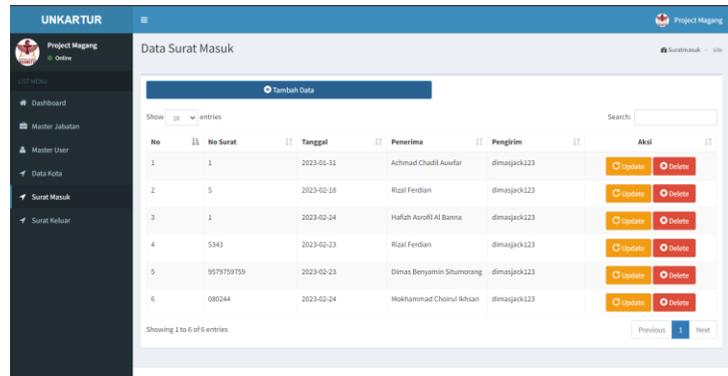
i. Implementasi Halaman *Form* Tambah Data Surat Masuk



Gambar 4. 27 Implementasi halaman *form* tambah data surat masuk

Tampilan halaman *form* tambah data surat masuk diatas dapat memasukkan data berupa nomor surat, tanggal dan penerima surat.

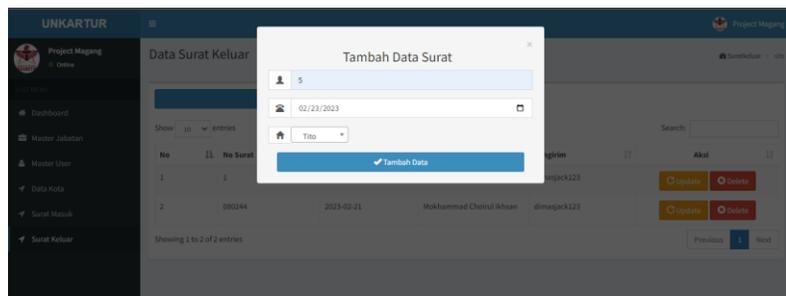
j. Implementasi Halaman Surat Masuk



Gambar 4. 28 Implementasi halaman surat masuk

Tampilan halaman menu surat masuk merupakan menu yang digunakan untuk menghasilkan data surat masuk. Pada halaman menu surat masuk terdapat fitur yang menampilkan data seluruh surat masuk, aksi yang bisa dilakukan seperti edit dan hapus data, serta tambah data baru.

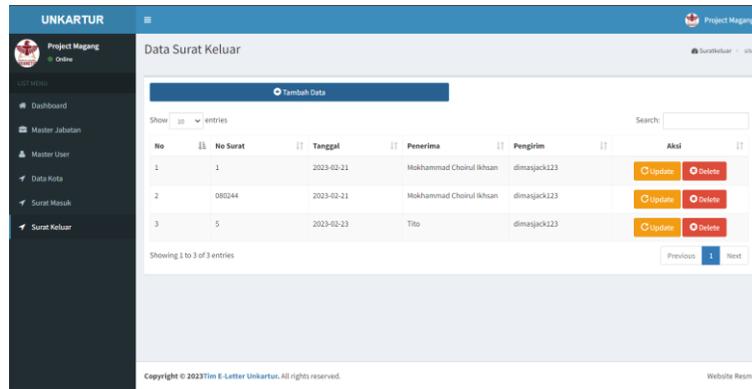
k. Implementasi Halaman *Form* Tambah Surat Keluar



Gambar 4. 29 Implementasi halaman *form* tambah data surat keluar

Tampilan halaman *form* tambah data surat keluar di atas dapat memasukkan data berupa nomor surat, tanggal dan pengirim surat.

1. Implementasi Halaman Surat Keluar



Gambar 4. 30 Implementasi halaman surat keluar

Tampilan halaman menu surat keluar merupakan menu yang digunakan untuk menghasilkan data surat keluar baik *internal* maupun *eksternal*. Pada halaman menu surat keluar terdapat fitur yang menampilkan data seluruh surat keluar, aksi yang bisa dilakukan seperti edit dan hapus data, serta tambah data.

4.5 Pengujian

Dalam pengujian sistem yang telah dibuat, metode yang digunakan adalah *blackbox testing*. Tujuan dari pengujian *blackbox* adalah untuk menguji hubungan antara *test case* yang diberikan dengan *output* yang dihasilkan oleh sistem, apakah sesuai dengan yang diharapkan atau tidak (Finanda & Fatmawati, 2019). Metode pengujian ini diterapkan pada sistem informasi surat berbasis *web* di Universitas Nasional Karangturi. Berikut hasil pengujian pada tabel 4.10.

Tabel 4. 9 Pengujian

No	Pengujian	Test Case	Hasil yang didapat	Hasil
1.	Login	Username dan password benar	Berhasil login ke dashboard	Valid
		Username dan password benar	Tetap di halaman login	Valid

No	Pengujian	Test Case	Hasil yang didapat	Hasil
2.	<i>Logout</i>	Menekan tombol <i>logout</i> di pojok kanan atas	Beralih ke halaman awal (<i>login</i>)	<i>Valid</i>
3.	Master Jabatan	<ul style="list-style-type: none"> a. Menekan tombol sub menu master jabatan. b. Menekan tombol tambah data. c. Menekan tombol update. d. Menekan tombol <i>delete</i>. e. Menekan tombol detail. 	<ul style="list-style-type: none"> a. Sistem menampilkan tabel data master jabatan. b. Menampilkan <i>form</i> tambah data master jabatan. c. Menampilkan <i>form</i> edit data master jabatan. d. Data master jabatan berhasil dihapus. e. Menampilkan detail data master jabatan. 	<i>Valid</i>
4.	Master <i>users</i>	<ul style="list-style-type: none"> a. Menekan tombol sub menu master <i>users</i>. b. Menekan tombol tambah data <i>user</i>. c. Menekan tombol <i>update</i>. d. Menekan tombol <i>delete</i>. 	<ul style="list-style-type: none"> a. Sistem menampilkan tabel data master <i>users</i>. b. Menampilkan <i>form</i> tambah data master <i>users</i>. c. Menampilkan <i>form update</i> data master <i>users</i>. d. Data master <i>users</i> berhasil dihapus. 	<i>Valid</i>
5.	Data Kota	<ul style="list-style-type: none"> a. Menekan tombol sub menu data kota. b. Menekan tombol tambah data kota. c. Menekan tombol <i>update</i>. d. Menekan tombol <i>delete</i>. e. Menekan tombol detail. 	<ul style="list-style-type: none"> a. Sistem menampilkan tabel data kota. b. Menampilkan <i>form</i> tambah data kota. c. Menampilkan <i>form update</i> data kota. d. Data kota berhasil dihapus. e. Menampilkan detail data kota. 	<i>Valid</i>
6.	Surat Masuk	<ul style="list-style-type: none"> a. Menekan tombol sub menu surat masuk. b. Menekan tombol tambah data surat masuk. 	<ul style="list-style-type: none"> a. Sistem menampilkan tabel surat masuk. b. Menampilkan <i>form</i> tambah data surat masuk. 	<i>Valid</i>

No	Pengujian	Test Case	Hasil yang didapat	Hasil
		c. Menekan tombol <i>update</i> . d. Menekan tombol <i>delete</i> .	c. Menampilkan <i>form update</i> data surat masuk. d. Data surat masuk berhasil dihapus.	
7.	Surat Keluar	a. Menekan tombol sub menu surat keluar. b. Menekan tombol tambah data surat keluar. c. Menekan tombol <i>update</i> . d. Menekan tombol <i>delete</i> .	a. Sistem menampilkan tabel surat keluar. b. Menampilkan <i>form</i> tambah data surat keluar. c. Menampilkan <i>form update</i> data surat keluar. d. Data surat keluar berhasil dihapus.	<i>Valid</i>
8.	<i>Search</i>	Menuliskan kata kunci yang ingin dicari di kolom <i>search</i>	Sistem menampilkan data yang dituliskan berdasarkan kata kunci	<i>Valid</i>
9.	Ubah Profil	Menekan foto profil yang ada di pojok kanan atas. Pilih menu <i>setting</i> . Pilih menu ubah <i>password</i>	Sistem menampilkan profil <i>user</i> yang sedang <i>login</i> . Menampilkan <i>form</i> edit profil. Menampilkan <i>form</i> ubah <i>password</i>	<i>Valid</i>
10.	Laporan surat	Cetak laporan surat	Laporan surat berhasil dicetak.	<i>Valid</i>

Berdasarkan hasil pengujian di atas, dapat disimpulkan bahwa sistem informasi surat berbasis *web* di Universitas Nasional Karangturi telah berhasil diuji dan memberikan hasil yang *valid* sesuai dengan fungsionalitas yang diharapkan. Sistem informasi surat siap untuk diimplementasikan dan digunakan oleh pengguna di Universitas Nasional Karangturi.