

BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Aplikasi

4.1.1 Perancangan Kebutuhan Alat

Analisis kebutuhan alat diperlukan oleh peneliti dalam mengembangkan dan menerapkan Metode RAD Pada Perancangan Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika UNISNU Jepara, meliputi ;

a. *Hardware* (Perangkat Keras)

Adapun untuk perangkat keras yang digunakan untuk perancangan aplikasi kali ini adalah sebuah Laptop dan *Smartphone* dengan spesifikasi sebagai berikut ;

➤ *Smartphone* Android

- RAM 2 GB
- Storage 16 Gb
- Layar 5 Inci
- Quad-core 1.4 GHz Cortex-A53
- Android 9.0 (Android Pie)

➤ Laptop

- RAM 4 GB
- SSD 120 GB
- HDD 500 GB
- Layar 14 Inci.
- Intel Core i3-3217U
- AMD Radeon 8530M + Intel HD 4000
- Wireless 802.11 b/g/n, Gigabit LAN
- Sistem Operasi Windows 10

b. *Software* (Perangkat Lunak)

Untuk *software* atau perangkat lunak yang digunakan untuk perancangan sistem aplikasi kali ini adalah

- Microsoft Windows 10 sebagai OS atau sistem operasi
- StarUML dan Edraw Max untuk membuat beberapa diagram UML
- Xampp sebagai web server (Apache) dan database server (MySQL)
- Sublime Text Editor untuk menuliskan kode program yang digunakan untuk *Backend* pada aplikasi kali ini
- Google Chrome untuk menjalankan *Backend* aplikasi

c. *Brainware* (Pengguna atau User)

Penelitian kali ini membutuhkan pengguna untuk mengoperasikan aplikasi ini, adapun untuk pengguna atau user kali ini adalah

- Admin
- Mahasiswa
- Dosen
- Kaprodi

4.1.2 Analisis Kebutuhan Dan Informasi

Dalam proses pembuatan aplikasi ini, peneliti memerlukan beberapa data yang berhubungan dengan proses skripsi yang ada di Program Studi Teknik Informatika Universitas Nahdlatul Ulama Jepara. Data ini diperoleh dengan cara melakukan studi pustaka, yaitu dengan mencari dan memahami literatur tentang proses layanan tugas akhir mahasiswa dari berbagai perguruan tinggi. Peneliti juga memanfaatkan data jumlah mahasiswa dan dosen Teknik Informatika Universitas Islam Nahdlatul Ulama Jepara. Data penunjang selanjutnya adalah hasil kuesioner yang diberikan oleh peneliti kepada beberapa mahasiswa dan dosen Teknik Informatika Universitas Nahdlatul Ulama Jepara. Dan data penunjang yang

terakhir yaitu hasil wawancara yang dilakukan oleh peneliti dengan Ketua Program Studi Teknik Informatika Universitas Islam Nahdlatul Ulama Jepara. Adapun data – data yang diperoleh adalah sebagai berikut ;

- a. Data jumlah mahasiswa yang sudah lulus, belum lulus, dan drop out
- b. Data rekapitulasi hasil kuesioner mahasiswa dan dosen

Data di atas disajikan dalam bentuk lampiran pada bagian akhir skripsi.

4.1.3 Analisis Kebutuhan Fungsional

Kebutuhan yang disediakan oleh aplikasi yang sesuai dengan kebutuhan yaitu ;

- a. Sebagai Admin
 - Input data mahasiswa, dosen, dan kaprodi
 - Input berita seputar pelaksanaan skripsi
 - Eksport laporan hasil bimbingan yang dilakukan oleh mahasiswa
 - Eksport rekapitulasi judul ajuan mahasiswa
- b. Sebagai Mahasiswa
 - Input judul ajuan ke dosen pembimbing akademik
 - Melihat status judul ajuan
 - Mengajukan jadwal bimbingan kepada dosen pembimbing
 - Melihat riwayat bimbingan yang diinputkan oleh dosen pembimbing
- c. Sebagai Dosen
 - Memberikan informasi terkait waktu yang disediakan oleh dosen kepada mahasiswa untuk melakukan bimbingan skripsi

- Melihat ajuan judul mahasiswa
- Memberikan komentar atau catatan pada ajuan judul mahasiswa
- Menolak ajuan judul mahasiswa
- Menerima dan meneruskan ajuan judul mahasiswa ke kaprodi
- Input riwayat bimbingan mahasiswa
- Mencari riwayat mahasiswa dalam melakukan bimbingan

d. Sebagai Kaprodi

- Melihhan ajuan judul mahasiswa yang telah disetujui oleh dosen pembimbing akademik
- Memberikan komentar atau catatan pada ajuan judul mahasiswa
- Menerima ajuan judul mahasiswa
- Menolak ajuan judul mahasiswa

4.1.4 Analisis Kebutuhan Non Fungsional

Merupakan kebutuhan yang tidak memiliki hubungan langsung secara spesifik dengan aplikasi yang ada. Kebutuhan ini lebih mengarah pada aksesibilitas aplikasi, adapun untuk kebutuhan non fungsional aplikasi ini antara lain ;

a. Sebagai Admin

- Dalam proses input data akan lebih efektif dan maksimal jika dilakukan dengan sarana laptop atau komputer yang telah tersambung dengan internet
- Aplikasi ini menggunakan web browser sebagai halaman admin dalam melakukan tugasnya

b. Sebagai Mahasiswa

- Aplikasi ini hanya bisa digunakan dan dioperasikan melalui smartphone android. Apabila menginginkan aplikasi ini

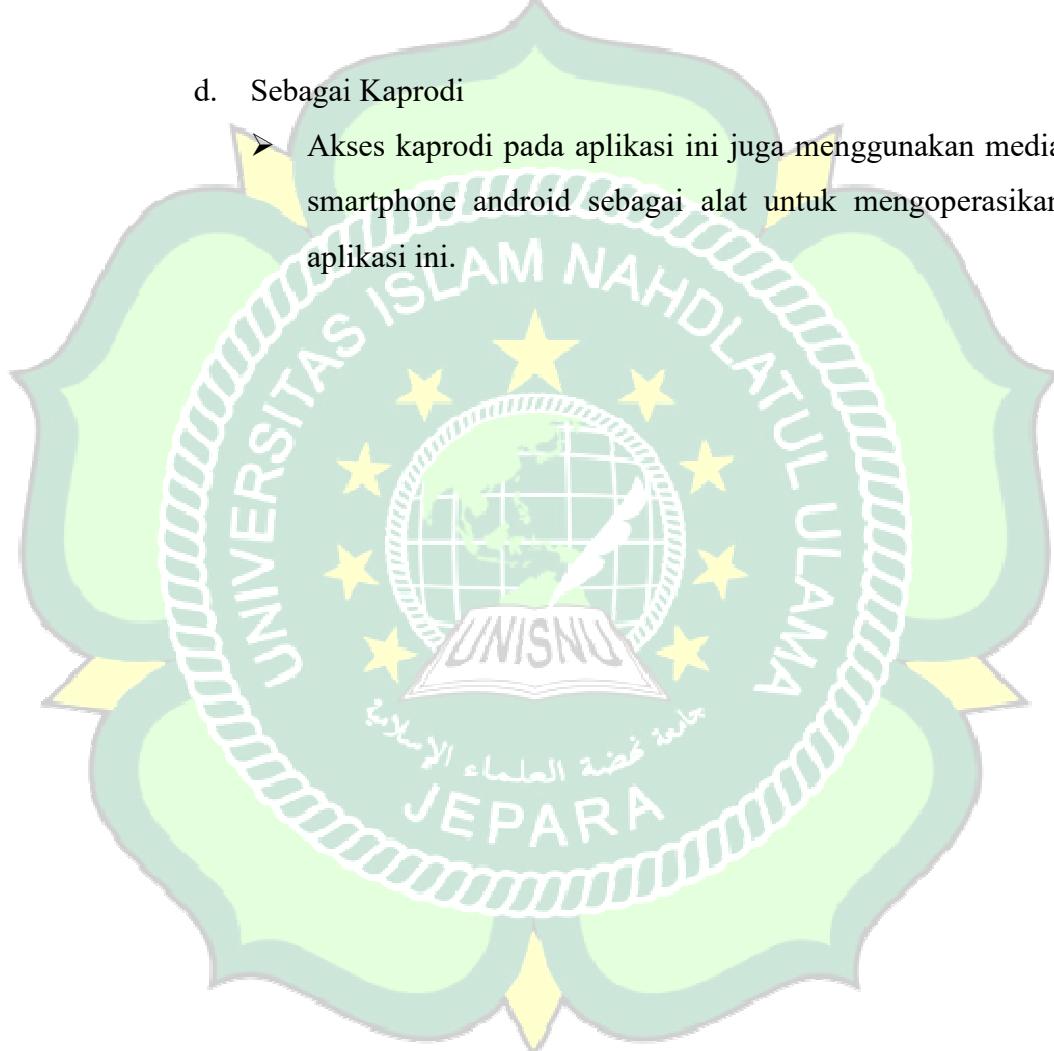
bisa berjalan dan dioperasikan di laptop atau komputer, maka bisa menggunakan aplikasi emulator android dari pihak penyedia layanan aplikasi

c. Sebagai Dosen

- Aplikasi layanan skripsi ini juga menggunakan smartphone android sebagai media pengoperasian untuk dosen

d. Sebagai Kaprodi

- Akses kaprodi pada aplikasi ini juga menggunakan media smartphone android sebagai alat untuk mengoperasikan aplikasi ini.

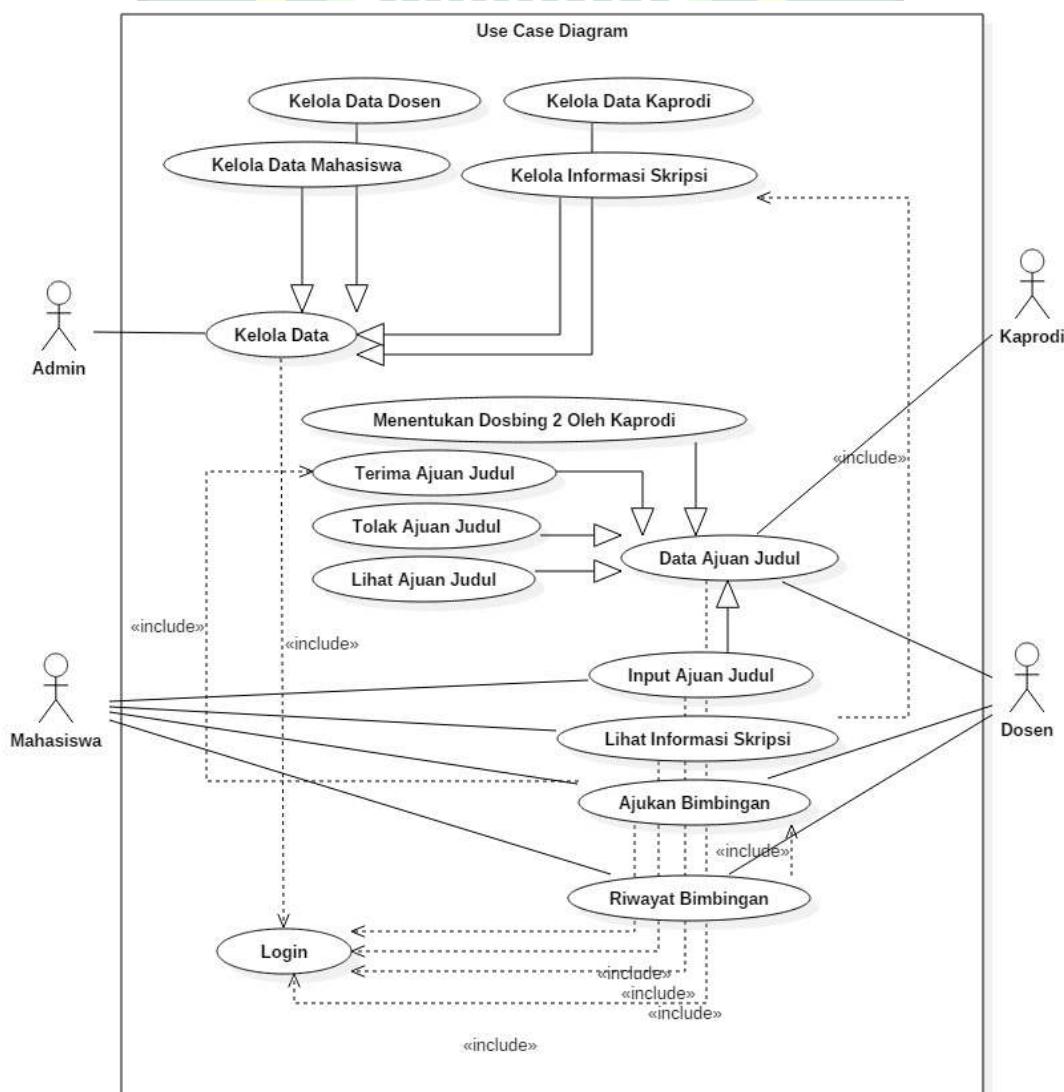


4.2 Desain Sistem

Peneliti merancang aplikasi menggunakan diagram UML (*Unified Modelling Language*) yang digambarkan melalui perancangan diagram seperti ;

4.2.1 Use Case Diagram

Merupakan suatu diagram yang digunakan untuk menunjukkan fitur yang ada di dalam aplikasi. *Use Case Diagram* mendeskripsikan sebuah interaksi antara satu atau lebih fungsi – fungsi di dalam Aplikasi layanan Skripsi Mahasiswa Teknik Informatika Unisnu Jepara.



Gambar 4. 1 Use Case Diagram Aplikasi Layanan Skripsi

Tabel 4. 1 Definisi *Use Case* Aktor

No	Aktor	Deskripsi
1	Admin	Orang yang memiliki hak akses untuk melakukan pengelolaan data mahasiswa, dosen, kaprodi, data ajuan judul, dan data bimbingan.
2	Mahasiswa	Orang yang memiliki akses untuk mengajukan judul, melihat informasi seputar pelaksanaan skripsi, mengajukan bimbingan, dan melihat riwayat bimbingan yang pernah dilakukan.
3	Dosen	Orang yang memiliki akses untuk melihat ajuan judul yang dikirimkan oleh mahasiswa, menolak judul ajuan, mengirimkan judul ajuan ke kaprodi, dan input riwayat bimbingan mahasiswa
4	Kaprodi	Orang yang memiliki akses untuk menerima atau menolak judul ajuan yang dikirimkan oleh mahasiswa melalui dosen pembimbing akademik dan menentukan dosen pembimbing 2 untuk mahasiswa judulnya diterima.

Tabel 4. 2 Definisi *Use Case Diagram* Admin

No	Use Case	Deskripsi
1	Login	Proses login ke aplikasi web admin
2	Kelola Data	Proses pengelolaan beberapa data yang meliputi ; data mahasiswa ; data dosen ; data kaprodi ; informasi skripsi
3	Kelola Data Dosen	Proses create, read, update, dan delete bioadata dosen
4	Kelola Data Mahasiswa	Proses create, read, update, dan delete bioadata mahasiswa

5	Kelola Data Kaprodi	Proses create, read, update, dan delete bioadata kaprodi
6	Kelola Informasi Skripsi	Proses create, read, update, dan delete informasi seputar pelaksanaan skripsi
7	Kelola Data Ajuan Judul Mahasiswa	Merupakan proses melihat data ajuan judul yang dilakukan mahasiswa, mengecek status ajuan judul mahasiswa
8	Kelola Riwayat Bimbingan Mahasiswa	Merupakan proses melihat dan mendownload hasil riwayat bimbingan mahasiswa

Tabel 4. 3 Definisi *Use Case* Mahasiswa

No	Use Case	Deskripsi
1	Login	Proses login ke aplikasi android
2	Input Ajuan Judul	Merupakan proses pengajuan judul skripsi mahasiswa yang akan muncul di halaman ajuan judul dosen
3	Lihat Status Ajuan Judul	Proses melihat status judul skripsi yang diajukan
4	Ajukan Bimbingan	Merupakan proses yang dilakukan ketika judul ajuan telah diterima dan dosen pembimbing 2 telah ditentukan oleh kaprodi
5	Lihat Riwayat Bimbingan	Merupakan proses melihat daftar riwayat bimbingan yang telah dilakukan oleh mahasiswa kepada dosen pembimbing
6	Lihat Informasi Skripsi	Merupakan proses untuk melihat informasi seputar pelaksanaan skripsi yang diinformasikan dari pihak program studi

Tabel 4. 4 Definisi *Use Case* Dosen

No	Use Case	Deskripsi
1	Login	Proses login ke aplikasi android
2	Kelola Ajuan Judul	Merupakan proses melihat ajuan judul yang dikirimkan oleh mahasiswa untuk diteruskan ke kaprodi atau ditolak
3	Melihat Ajuan Bimbingan	Merupakan proses melihat ajuan bimbingan yang diajukan oleh mahasiswa
4	Input Riwayat Bimbingan	Merupakan input riwayat bimbingan yang telah dilakukan mahasiswa

Tabel 4. 5 Definisi *Use Case* Kaprodi

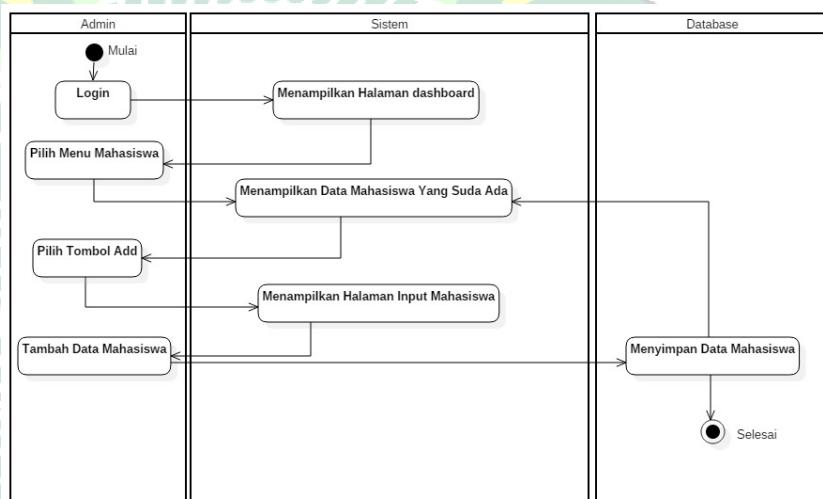
No	Use Case	Deskripsi
1	Login	Proses login ke aplikasi android
2	Kelola Ajuan Judul	Merupakan proses melihat ajuan judul mahasiswa yang telah dikirimkan oleh dosen pembimbing
3	Menentukan Dosen Pembimbing	Merupakan proses menentukan dosen pembimbing 2 untuk mahasiswa yang ajuan judul skripsinya diterima oleh kaprodi

4.2.2 Activity Diagram

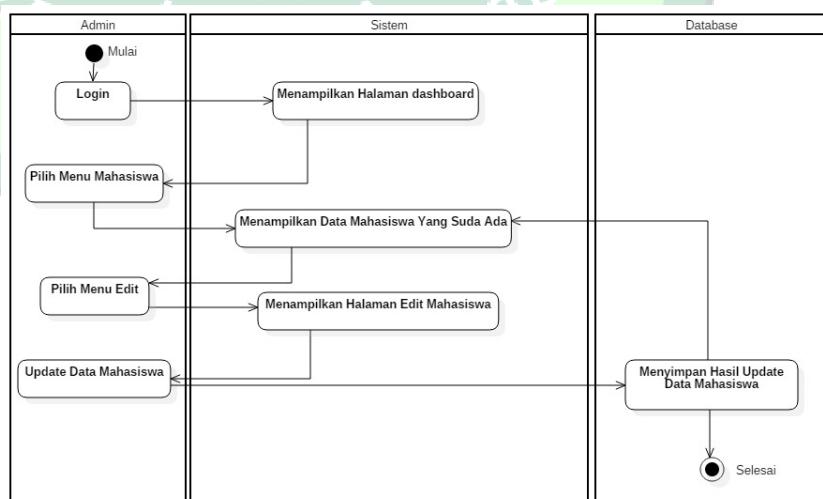
Merupakan diagram yang memperlihatkan aliran dari suatu aktifitas ke aktifitas yang lain dalam suatu sistem terhadap aktor. Berikut ini merupakan activity digram dari Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Universitas Nahdlatul Ulama Jepara;

a. Kelola Data Mahasiswa Oleh Admin

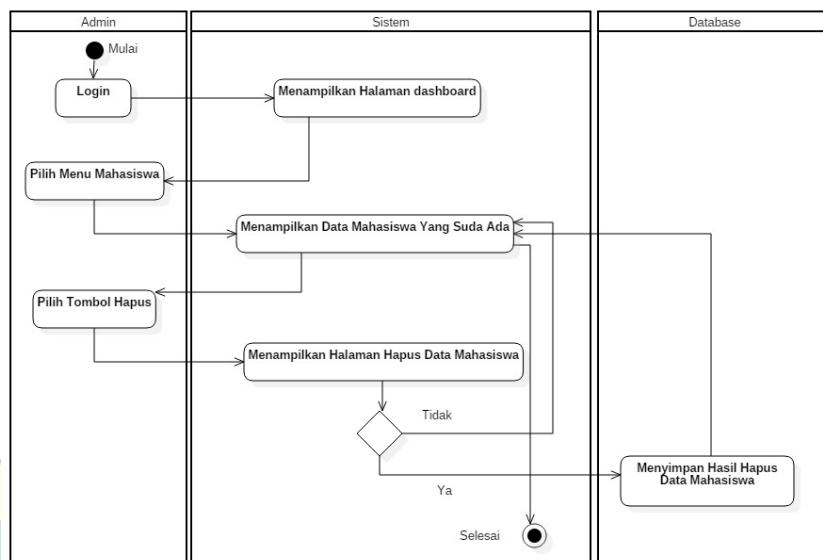
Admin dapat mengelola data mahasiswa yang sudah bisa melakukan proses pengajuan judul skripsi. Mulai dari melihat, menambah, mengedit, dan menghapus data mahasiswa.



Gambar 4. 2 Activity Diagram Manambah Data Mahasiswa



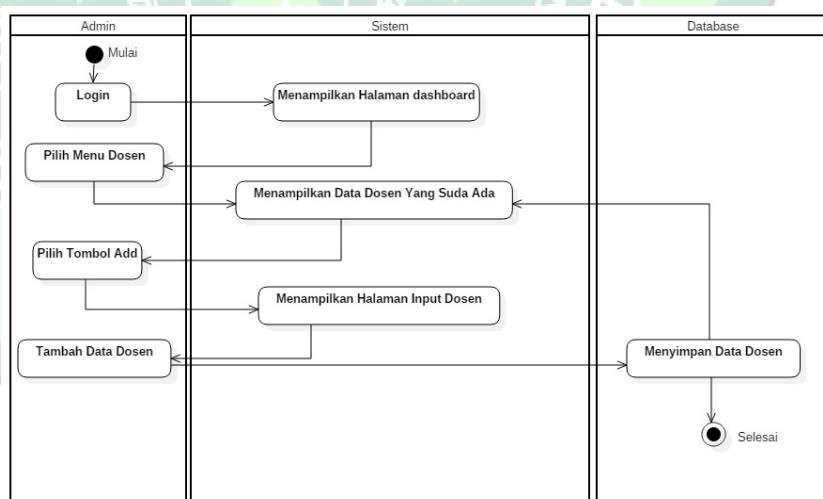
Gambar 4. 3 Activity Diagram Edit Data Mahasiswa



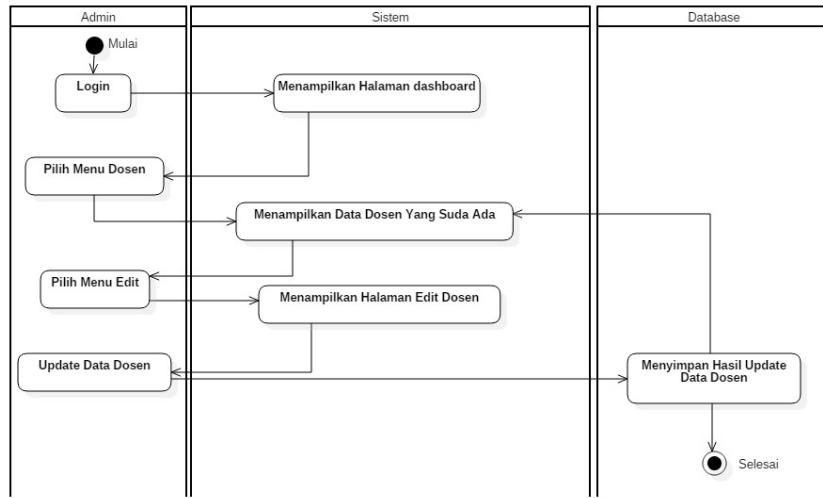
Gambar 4. 4 Activity Diagram Hapus Data Mahasiswa

b. Kelola Data Dosen Oleh Admin

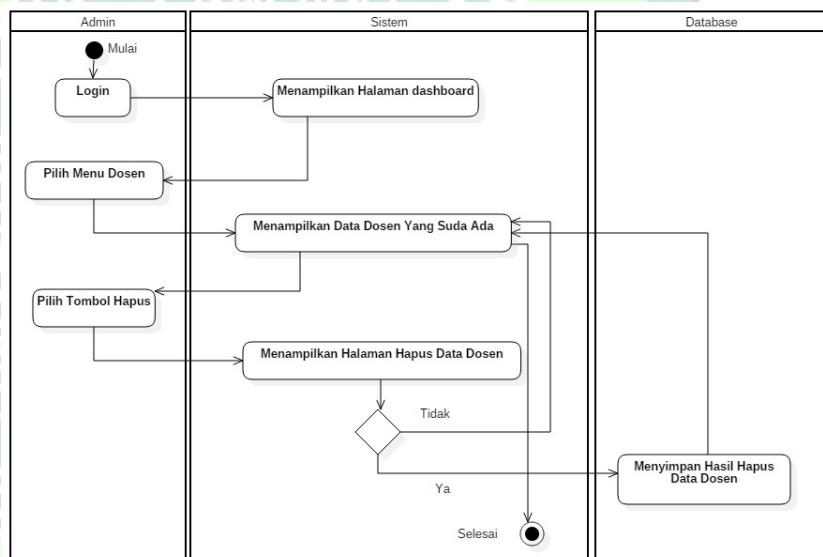
Admin dapat mengelola data dosen yang sudah menjadi dosen pembimbing. Mulai dari melihat, menambah, mengedit, dan menghapus data dosen



Gambar 4. 5 Activity Diagram Tambah Data Dosen



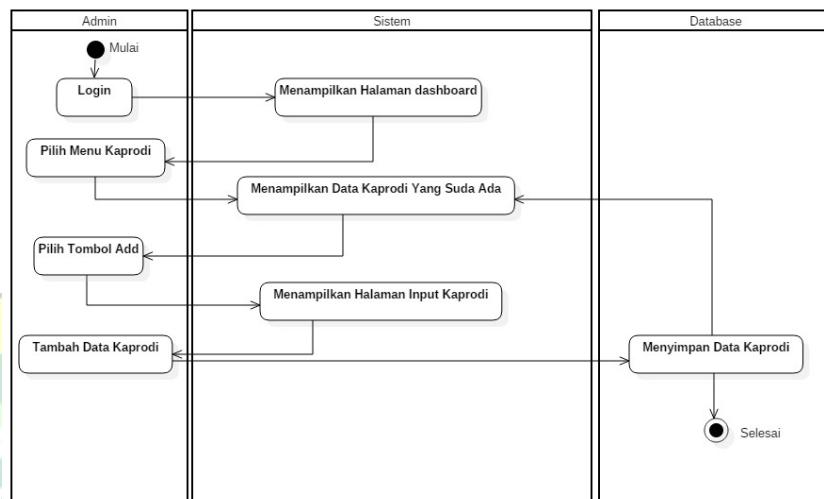
Gambar 4. 6 *Activity Diagram Edit Data Dosen*



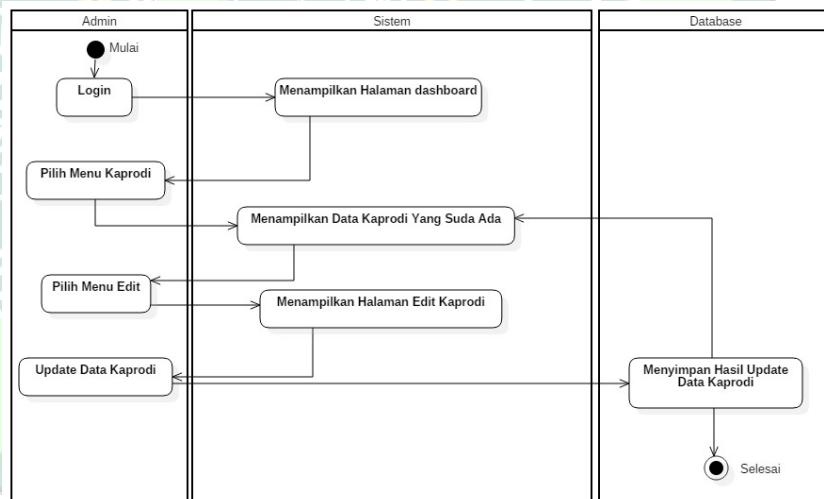
Gambar 4. 7 *Activity Diagram Hapus Data Dosen*

c. Kelola Data Kaprodi Oleh Admin

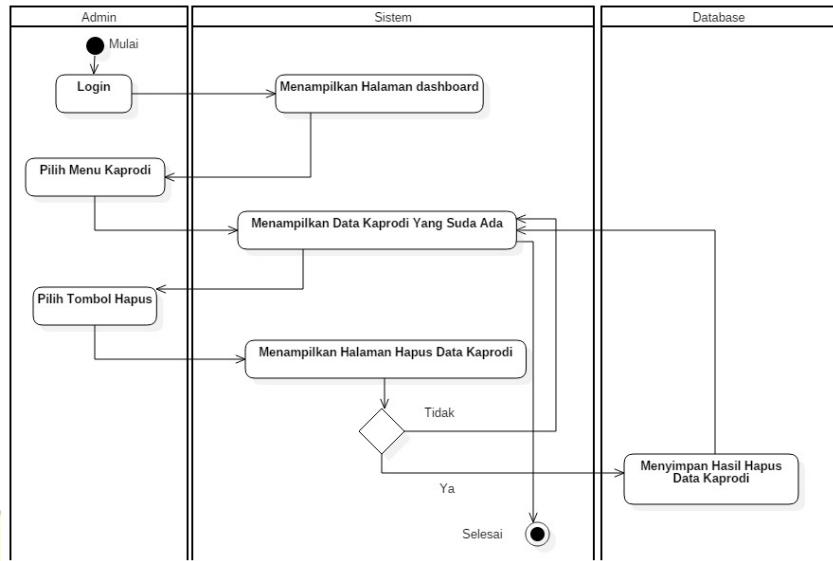
Admin dapat mengelola data kaprodi. Mulai dari melihat, menambah, mengedit, dan menghapus data kaprodi



Gambar 4. 8 *Activity Diagram* Tambah Data Kaprodi

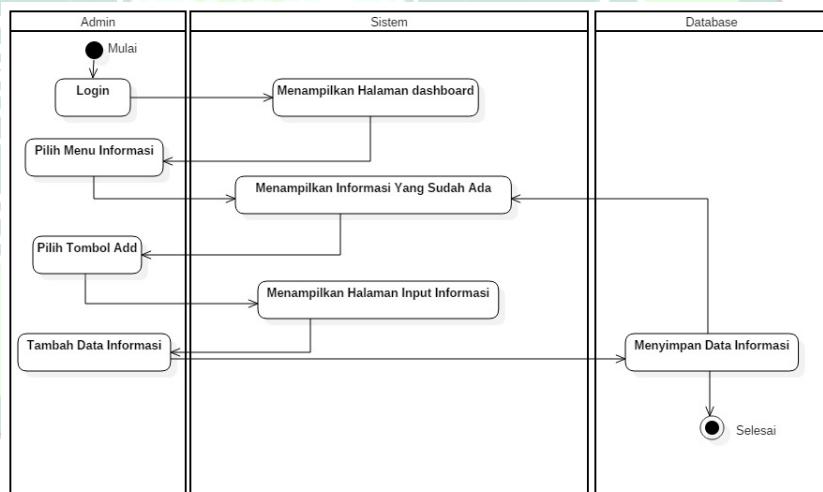


Gambar 4. 9 *Activity Diagram* Edit Data Kaprodi

Gambar 4. 10 *Activity Diagram* Hapus Data Kaprodi

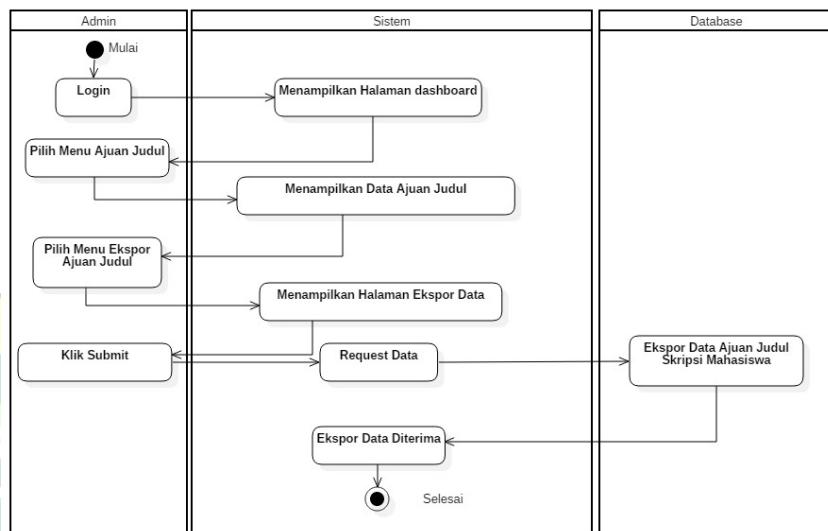
d. Kelola Informasi Skripsi Oleh Admin

Admin dapat mengelola informasi terkait pelaksanaan skripsi. Mulai dari melihat, menambah, mengedit, dan menghapus informasi

Gambar 4. 11 *Activity Diagram* Input Informasi Sepertu Skripsi

e. Kelola Ajuan Judul Mahasiswa Oleh Admin

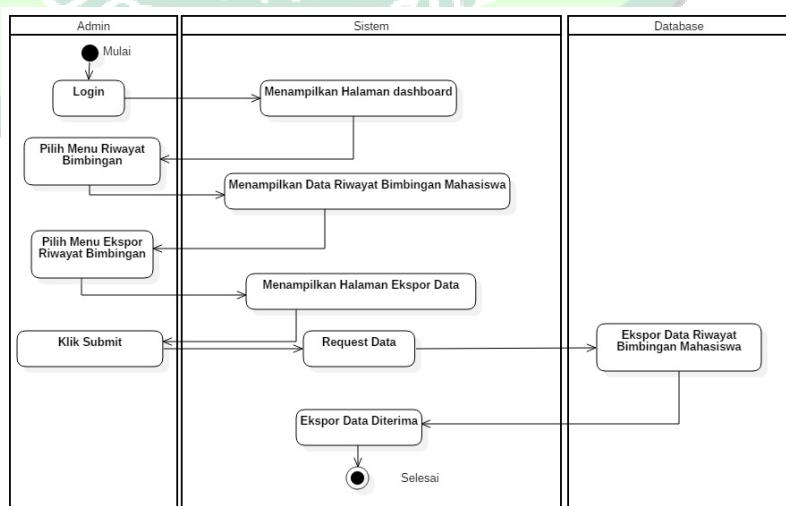
Admin dapat mengelola ajuan judul yang telah dikirimkan oleh mahasiswa. Mulai dari melihat status ajuan judul, mengedit dan menghapus judul mahasiswa.



Gambar 4. 12 *Activity Diagram* Ekspor Data Ajuan Judul Skripsi Mahasiswa

f. Kelola Riwayat Bimbingan Mahasiswa Oleh Admin

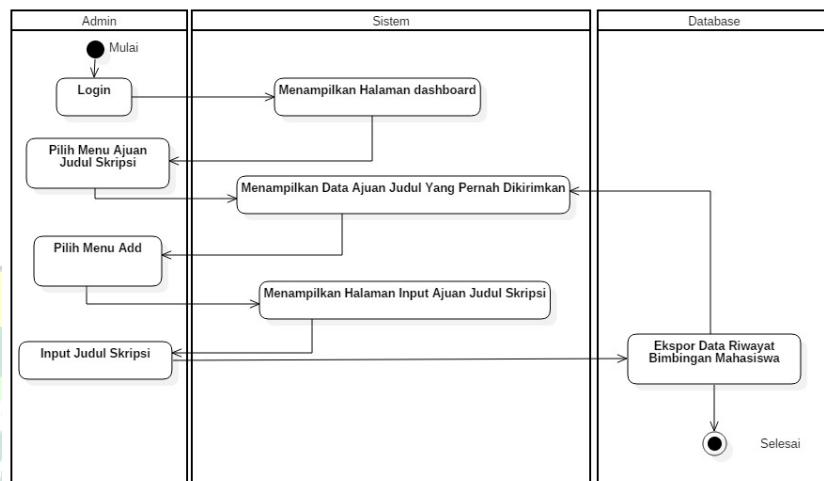
Admin dapat mengelola riwayat bimbingan mahasiswa yang telah diinputkan dosen pembimbing sebagai tanda bahwa mahasiswa telah melakukan proses bimbingan kepada dosen.



Gambar 4. 13 *Activity Diagram* Ekspor Riwayat Bimbingan

g. Input Ajuan Judul Oleh Mahasiswa

Proses ini merupakan proses pengajuan judul skripsi yang dilakukan oleh mahasiswa dan akan diteruskan ke aplikasi dosen.

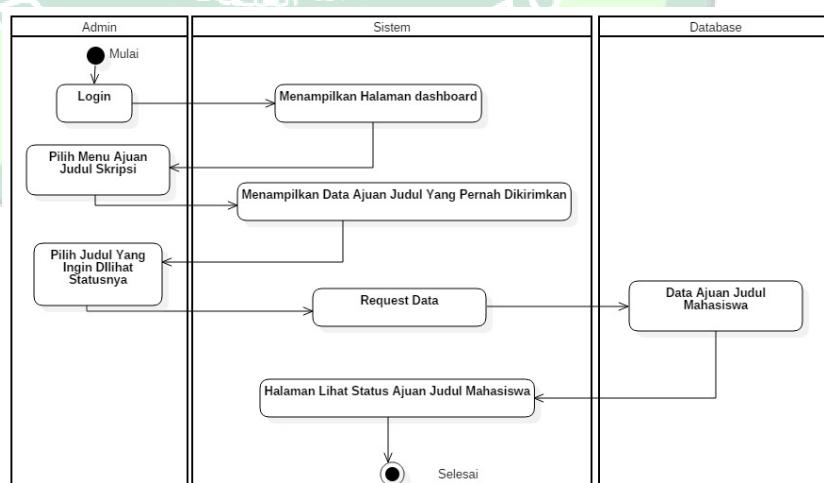


Gambar 4. 14 *Activity Diagram* Input Ajuan Judul

Mahasiswa

h. Lihat Status Ajuan Judul Oleh Mahasiswa

Mahasiswa bisa melihat status ajuan judul skripsi yang telah diajukannya, akan ada status apakah judul diterima atau ditolak.

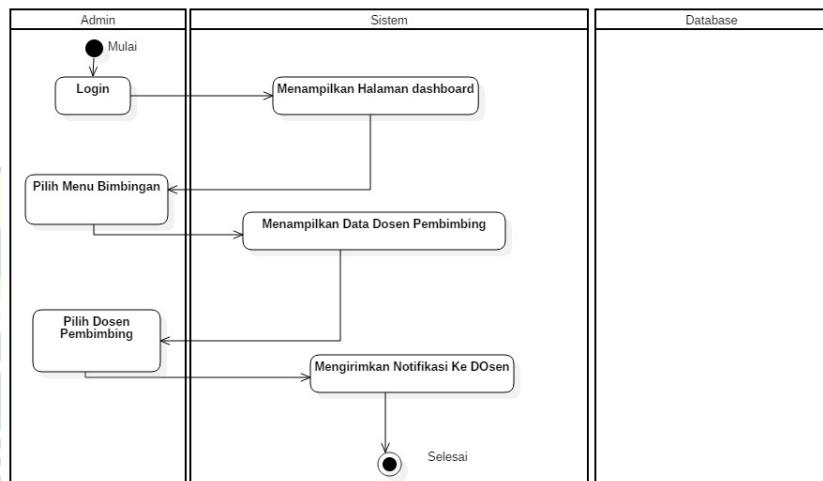


Gambar 4. 15 *Activity Diagram* Lihat Status Ajuan Judul

Mahasiswa

i. Mengajukan Bimbingan Skripsi Kepada Dosen Oleh Mahasiswa

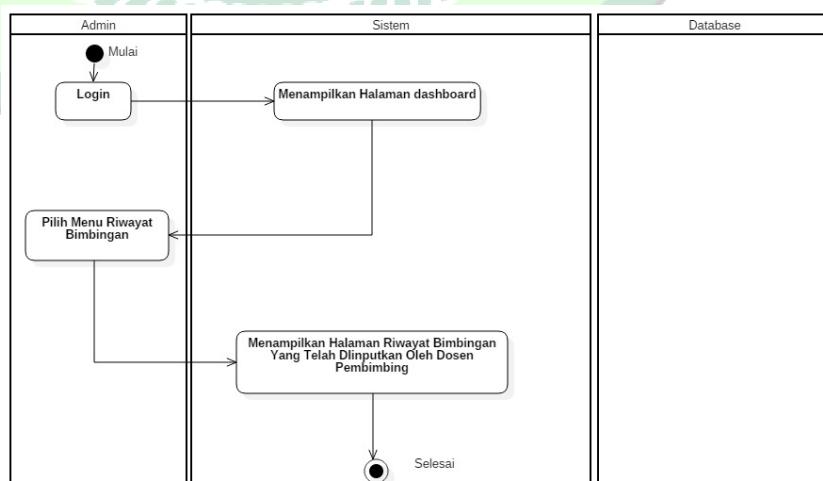
Mahasiswa dapat mengajukan bimbingan kepada dosen pembimbing dengan melihat jadwal ketersediaan dosen dalam memberikan bimbingan kepada mahasiswa.



Gambar 4. 16 *Activity Diagram Ajukan Bimbingan*

j. Melihat Riwayat Bimbingan Yang Telah Diisi Dosen Oleh Mahasiswa

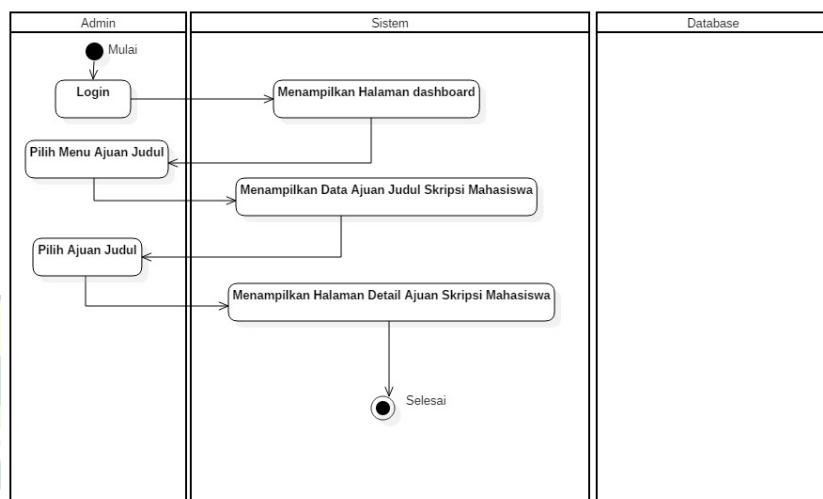
Mahasiswa dapat melihat riwayat bimbingan yang telah diinputkan oleh dosen pembimbing setelah melakukan bimbingan skripsi



Gambar 4. 17 *Activity Diagram Lihat Riwayat Bimbingan*

k. Melihat Ajuan Judul Skripsi Mahasiswa Oleh Dosen

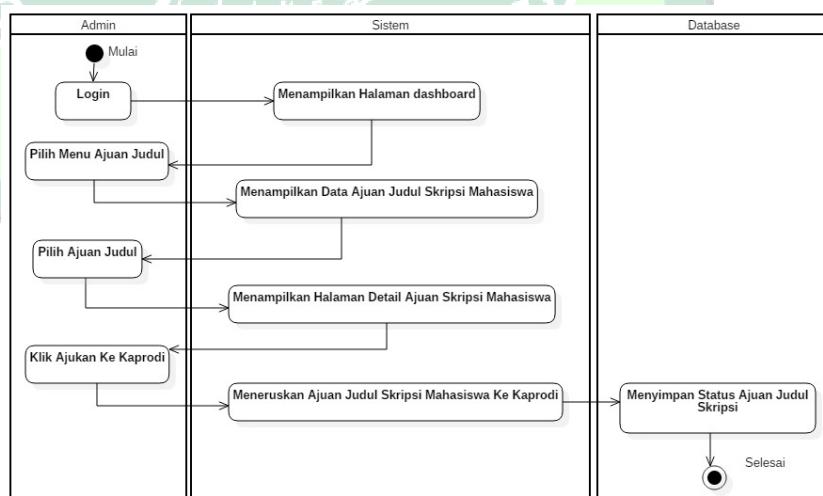
Dosen dapat melihat ajuan judul skripsi yang telah diinputkan oleh mahasiswa.



Gambar 4. 18 *Activity Diagram* Lihat Ajuan Judul Skripsi Oleh Dosen

1. Menerima Dan Meneruskan Ajuan Judul Skripsi Mahasiswa Ke Kaprodi Oleh Dosen

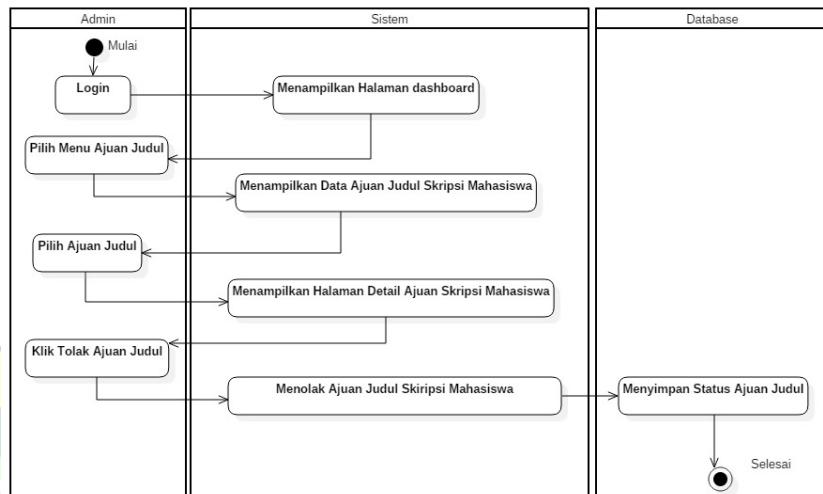
Diagram aktifitas dosen dalam menerima dan meneruskan ajuan judul skripsi mahasiswa ke kaprodi.



Gambar 4. 19 *Activity Diagram* Menerima Dan Meneruskan Judul Ke Kaprodi Oleh Dosen

m. Menolak Ajuan Judul Skripsi Mahasiswa Oleh Dosen

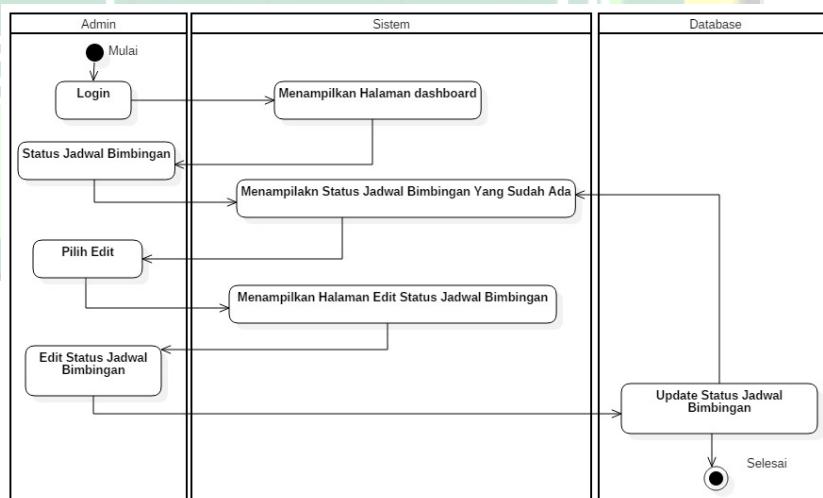
Diagram aktifitas dosen dalam menolak ajuan judul skripsi mahasiswa.



Gambar 4. 20 *Activity Diagram Tolak Ajuan Judul Skripsi Mahasiswa Oleh Dosen*

n. Mengedit Status Atau Jadwal Ketersediaan Dosen Dalam Melakukan Bimbingan Skripsi

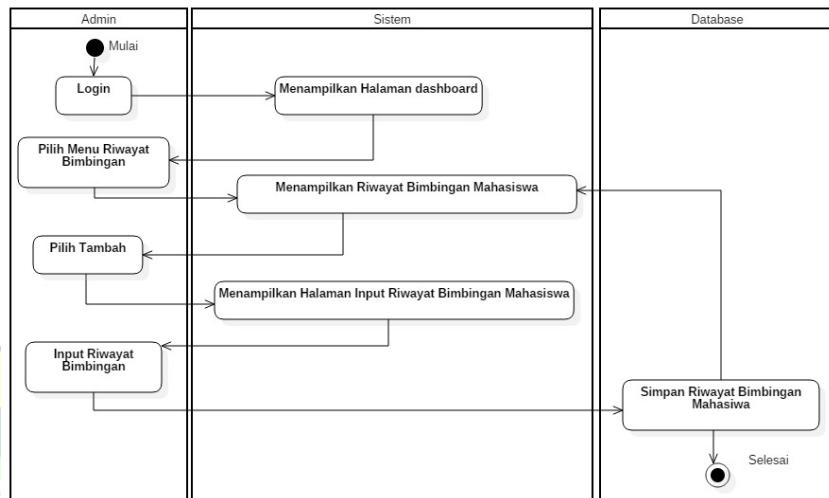
Dosen dapat mengedit status jadwal ketersediaan dosen dalam memberikan waktu bimbingan bagi mahasiswa.



Gambar 4. 21 *Activity Diagram Edit Status Jadwal Bimbingan Oleh Dosen*

o. Input Riwayat Bimbingan Mahasiswa Oleh Dosen

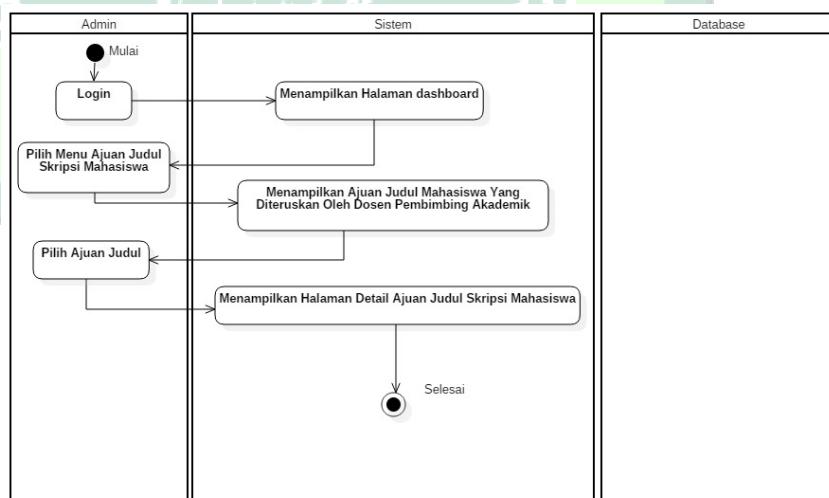
Dosen dapat melakukan proses input riwayat bimbingan yang telah dilakukan oleh mahasiswa.



Gambar 4. 22 *Activity Diagram Input Riwayat Bimbingan Mahasiswa Oleh Dosen*

p. Melihat Ajuan Judul Skripsi Mahasiswa Yang Telah Disetujui Dosen Oleh Kaprodi

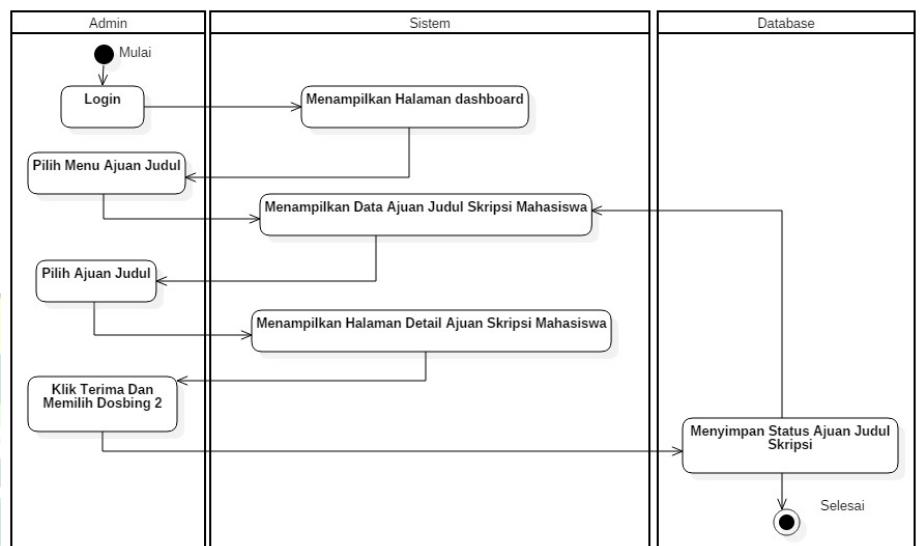
Kaprodi melihat judul ajuan skripsi mahasiswa yang sebelumnya telah disetujui dan diteruskan oleh dosen pembimbing akademik ke kaprodi.



Gambar 4. 23 *Activity Diagram Lihat Ajuan Judul Skripsi Mahasiswa Oleh Kaprodi*

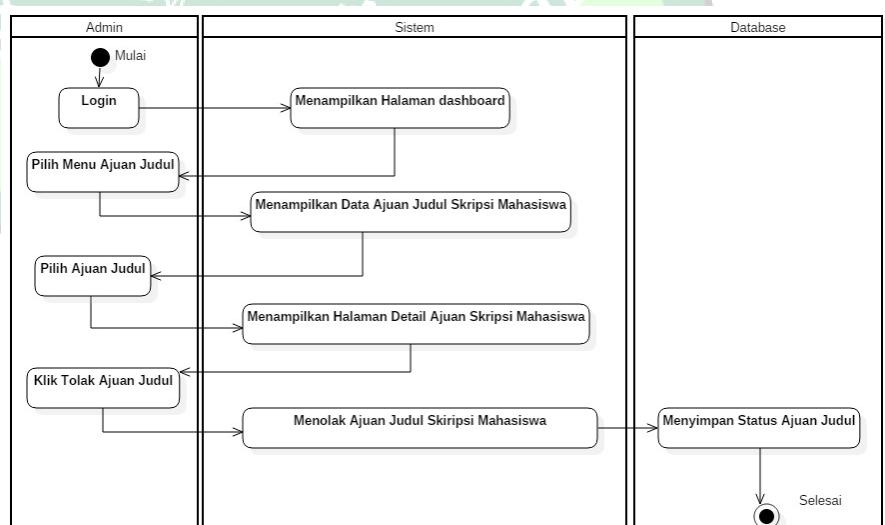
- q. Menerima Dan Menentukan Dosen Pembimbing 2 Pada Mahasiswa

Diagram aktifitas kaprodi dalam menerima ajuan judul skripsi mahasiswa dan menentukan dosen pembimbing 2.



Gambar 4. 24 *Activity Diagram* Menerima Dan Menentukan Dosen Pembimbing 2 Oleh Kaprodi

- r. Menolak Ajuan Judul Skripsi Mahasiswa Oleh Kaprodi
Kaprodi dapat menolak ajuan judul skripsi mahasiswa yang telah disetujui dan diteruskan dosen pembimbing.



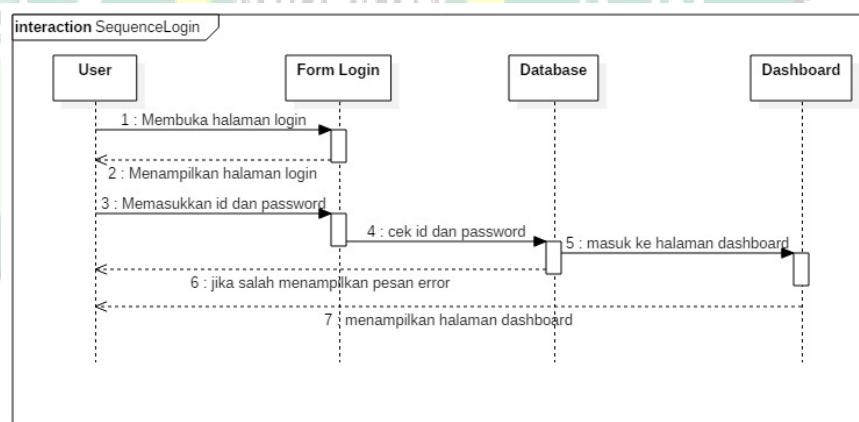
Gambar 4. 25 *Activity Diagram* Menolak Ajuan Judul Skripsi Oleh Kaprodi

4.2.3 Sequence Diagram

Diagram yang menggambarkan perilaku objek pada usecase dengan mendeskripsikan objek dan *message* yang dikirim serta diterima antar objek dalam Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Universitas Islam Nahdlatul Ulama Jepara yang disusun berdasarkan kondisi waktu.

a. Sequence Diagram Login

Menjelaskan proses dimana pengguna yang dalam hal ini adalah mahasiswa, dosen, dan kaprodi mengakses halaman *login* melalui validasi nomor induk mahasiswa/dosen dan *password*. Jika nomor induk mahasiswa/dosen dan *password* oleh pengguna ditemukan atau diterima maka pengguna dapat berhasil masuk halaman dashboard aplikasi sesuai validasi data akun yang dimasukkan, sedangkan jika nomor induk mahasiswa/dosen dan *password* yang dimasukkan oleh pengguna tidak berhasil atau gagal maka akan menampilkan pemberitahuan pada halaman *login* aplikasi.



Gambar 4. 26 Sequence Diagram Login

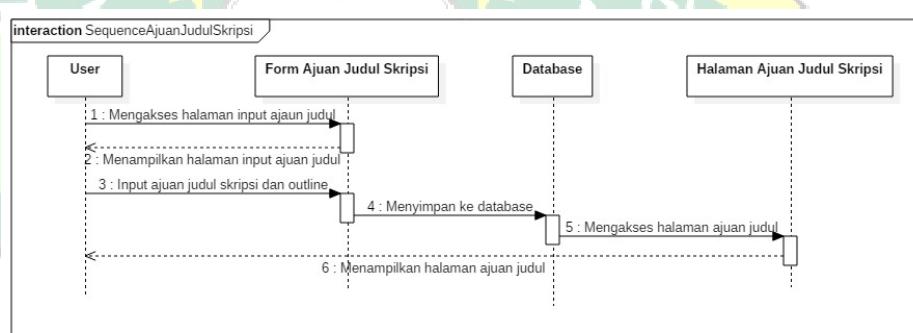
Deskripsi dari gambar

- User membuka halaman login aplikasi
- Sistem menampilkan halaman login

- User memasukkan id dan password untuk masuk ke dalam aplikasi
- Sistem akan mengecek id dan password yang dimasukkan oleh user dengan id dan password yang ada didatabase
- Sistem akan menampilkan halaman awal atau dashboard aplikasi ketika id dan password yang dimasukkan oleh user sesuai. Jika id dan password yang dimasukkan tidak sesuai, maka sistem akan memberikan pesan error kepada user.

b. *Sequence Diagram Pengajuan Judul Skripsi Oleh Mahasiswa*

Menjelaskan proses dimana mahasiswa yang dalam hal ini adalah sebagai pengguna mengakses halaman input ajuan judul skripsi dan menginputkan judul skripsi dan outline pada halaman input ajuan judul skripsi. Kemudian setelah ajuan judul itu dikirimkan, maka ajuan judul skripsi mahasiswa akan masuk ke dalam database dan akan muncul halaman status ajuan judul yang telah diinputkan tadi.



Gambar 4. 27 *Sequence Diagram Pengajuan Judul Skripsi*

Oleh Mahasiswa

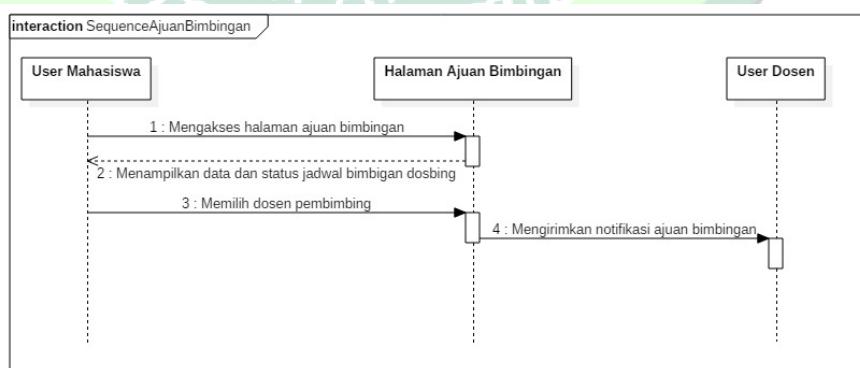
Deskripsi dari gambar

- Mahasiswa masuk ke dalam aplikasi dan mengakses halaman input ajuan judul skripsi

- Sistem akan menampilkan halaman input ajuan judul skripsi
- Mahasiswa melakukan input data berupa judul dan outline skripsi
- Sistem akan menyimpan data judul dan outline yang diinputkan oleh mahasiswa ke database
- Sistem menampilkan halaman ajuan judul skripsi, di halaman ini mahasiswa bisa melihat judul yang sebelumnya pernah diajukan namun ditolak oleh dosen atau kaprodi.

c. Sequence Diagram Pengajuan Bimbingan Skripsi Oleh Mahasiswa

Menjelaskan proses dimana mahasiswa yang dalam hal ini adalah pengguna mengakses halaman ajuan bimbingan kepada dosen pembimbing 1 atau dosen pembimbing 2, hal ini bisa dilakukan ketika ajuan judul skripsi yang telah diajukan telah disetujui dan dosen pembimbing 2 telah ditentukan oleh kaprodi. Ajuan bimbingan kali ini hanya bersifat pemberitahuan kepada akun dosen bahwa akan ada mahasiswa yang mengajukan bimbingan sesuai dengan status jadwal bimbingan dosen.



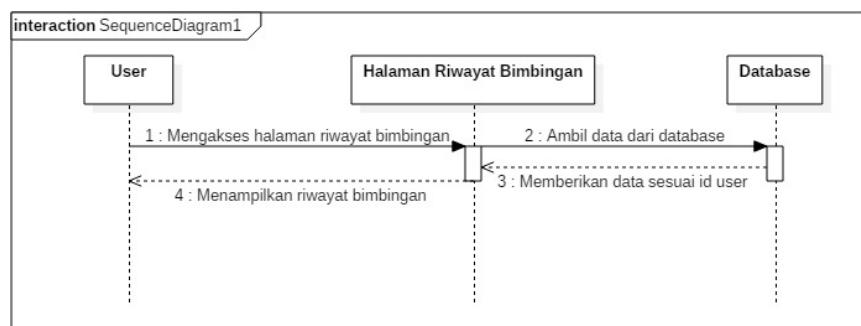
Gambar 4. 28 *Sequence Diagram* Pengajuan Bimbingan Skripsi Oleh Mahasiswa

Deskripsi dari gambar

- Mahasiswa masuk ke dalam aplikasi dan mengakses halaman ajuan bimbingan
- Sistem akan menampilkan halaman ajuan bimbingan, halaman ini memampulkan data dosen pembimbing 1 dan data dosen pembimbing 2 berupa status jadwal ketersediaan dosen dalam memberikan bimbingan
- Mahasiswa mengklik salah satu dosen yang akan dimintai bimbingan
- Sistem akan mengirimkan notifikasi kepada akun dosen pembimbing.

d. Sequence Diagram Melihat Riwayat Bimbingan Mahasiswa

Merupakan proses dimana mahasiswa yang dalam hal ini adalah pengguna mengakses halaman riwayat bimbingan yang telah diinputkan oleh dosen pembimbing 1 atau dosen pembimbing 2.



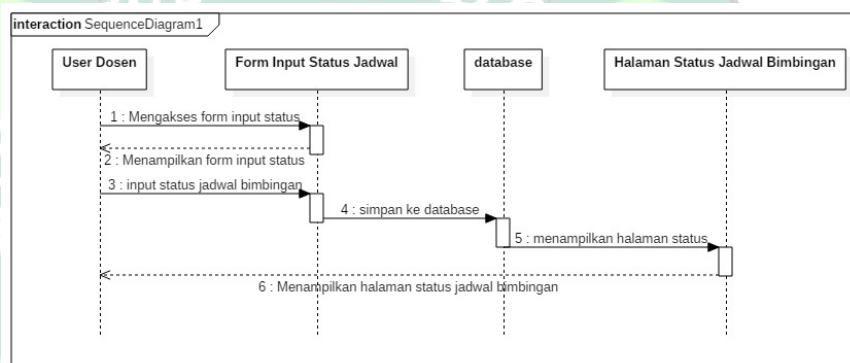
Gambar 4. 29 Sequence Diagram Melihat Riwayat Bimbingan

Deskripsi dari gambar

- Mahasiswa masuk aplikasi dan mengakses halaman riwayat bimbingan
- Sistem akan mengambil data riwayat bimbingan yang sebelumnya sudah pernah diinputkan oleh dosen dan tersimpan di database

- Sistem akan menerima data sesuai id mahasiswa dari database
 - Sistem akan menampilkan data riwayat bimbingan
- e. Sequence Diagram Input Status Jadwal Bimbingan Oleh Dosen

Merupakan proses dimana dosen yang dalam hal ini adalah pengguna mengakses halaman input status jadwal bimbingan dan menginputkan status jadwal bimbingan sesuai dengan kehendak dosen dalam memberikan ketersediaan menerima ajuan bimbingan dari mahasiswa.



Gambar 4. 30 Sequence Diagram Input Status Jadwal

Bimbingan Dosen

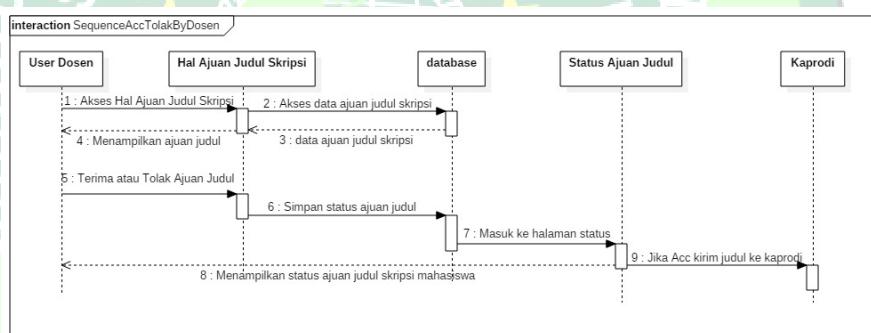
Deskripsi dari gambar

- Dosen masuk aplikasi dan mengakses halaman form input status jadwal ketersediaan dosen dalam memberikan bimbingan kepada mahasiswa
- Sistem akan menampilkan halaman form input status jadwal ketersediaan dosen dalam memberikan bimbingan kepada mahasiswa
- Dosen input atau update status jadwal ketersediaan dosen dalam memberikan bimbingan kepada mahasiswa
- Sistem akan menyimpan status jadwal ketersediaan dosen dalam memberikan bimbingan kepada mahasiswa

- Sistem akan menampilkan halaman status jadwal bimbingan ketika input status jadwal telah disimpan di database.

f. Sequence Diagram Terima/Tolak Ajuan Judul Skripsi Mahasiswa Oleh Dosen

Merupakan proses dimana dosen yang dalam hal ini adalah pengguna mengakses halaman ajuan judul skripsi yang telah diinputkan oleh mahasiswa. Dosen akan melihat detail judul ajuan mahasiswa dan akan memberikan respon berupa Acc/Tolak ajuan skripsi mahasiswa yang kemudian akan disimpan oleh database apakah ajuan tersebut diterima atau ditolak oleh dosen. Jika ajuan diterima, maka ajuan judul akan diteruskan ke akun kaprodi dengan status telah diterima oleh dosen pembimbing.



Gambar 4. 31 *Sequence Diagram* Terima/Tolak Ajuan Judul Skripsi Mahasiswa Oleh Dosen

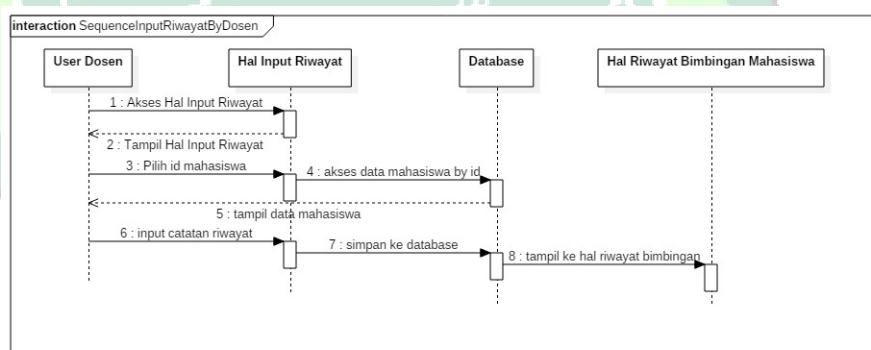
Deskripsi dari gambar

- Dosen masuk aplikasi dan mengakses halaman detail ajuan judul skripsi.
- Sistem akan mengambil detail data ajuan judul skripsi yang telah diinputkan oleh mahasiswa sebelumnya dari database
- Sistem akan menampilkan detail data ajuan judul skripsi pada halaman ajuan judul skripsi

- Dosen akan menerima atau menolak ajuan judul skripsi mahasiswa
- Sistem akan menyimpan status yang telah diberikan oleh dosen terhadap ajuan judul skripsi yang telah diajukan oleh mahasiswa
- Sistem akan menampilkan halaman dari hasil simpan status ajuan judul skripsi
- Jika ajuan judul skripsi mahasiswa diterima, maka ajuan judul akan secara otomatis terkirim atau masuk ke halaman ajuan judul skripsi mahasiswa di akun kaprodi.

g. Sequence Diagram Input Riwayat Bimbingan Mahasiswa Oleh Dosen

Merupakan proses dimana dosen yang dalam hal ini adalah pengguna mengakses halaman input catatan riwayat bimbingan mahasiswa. Dosen akan memilih data mahasiswa berdasarkan NIM yang menjadi id dari mahasiswa, kemudian memberi catatan tentang proses bimbingan yang telah dilaksanakan, dan akan disimpan di database untuk kemudian ditampilkan pada halaman riwayat bimbingan mahasiswa.



Gambar 4. 32 Sequence Diagram Input Riwayat Bimbingan Mahasiswa Oleh Dosen

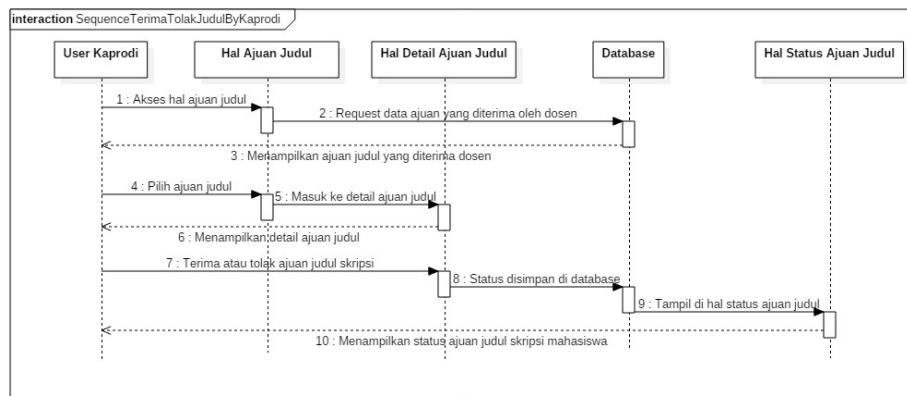
Deskripsi dari gambar

- Dosen masuk aplikasi dan mengakses halaman input riwayat bimbingan

- Sistem akan menampilkan halaman input riwayat bimbingan
- Dosen memilih id mahasiswa yang akan diberikan catatan terkait proses bimbingan yang telah dilaksanakan
- Sistem mengambil data mahasiswa berdasarkan id yang telah diinputkan oleh admin
- Sistem akan menampilkan data mahasiswa yang akan diberikan catatan bimbingan
- Dosen menginputkan catatan terkait proses bimbingan yang telah dilaksanakan.
- Sistem akan menyimpan input catatan riwayat ke database
- Setelah berhasil menyimpan, sistem akan menampilkan halaman riwayat bimbingan mahasiswa.

h. Sequence Diagram Terima/Tolak Ajuan Judul Skripsi Mahasiswa Oleh Kaprodi

Merupakan proses dimana kaprodi yang dalam hal ini adalah pengguna mengakses ajuan judul skripsi mahasiswa yang statusnya telah diterima oleh dosen pembimbing akademik. Kaprodi akan memilih judul ajuan skripsi mahasiswa, kemudian kaprodi akan menerima ajuan judul skripsi dan menentukan dosen pembimbing 2 untuk mahasiswa yang ajukan judul skripsinya diterima, serta akan tersimpan di database bahwa ajuan judul tersebut dengan status diterima oleh kaprodi. Jika ditolak, maka ajuan judul tersebut akan tersimpan ke database dengan status ditolak oleh kaprodi. Status ini nantinya akan ditampilkan pada aplikasi mahasiswa untuk memberikan informasi terkait status ajuan judul skripsi yang mereka ajukan.



Gambar 4. 33 *Sequence Diagram* Terima/Tolak Ajuan Judul

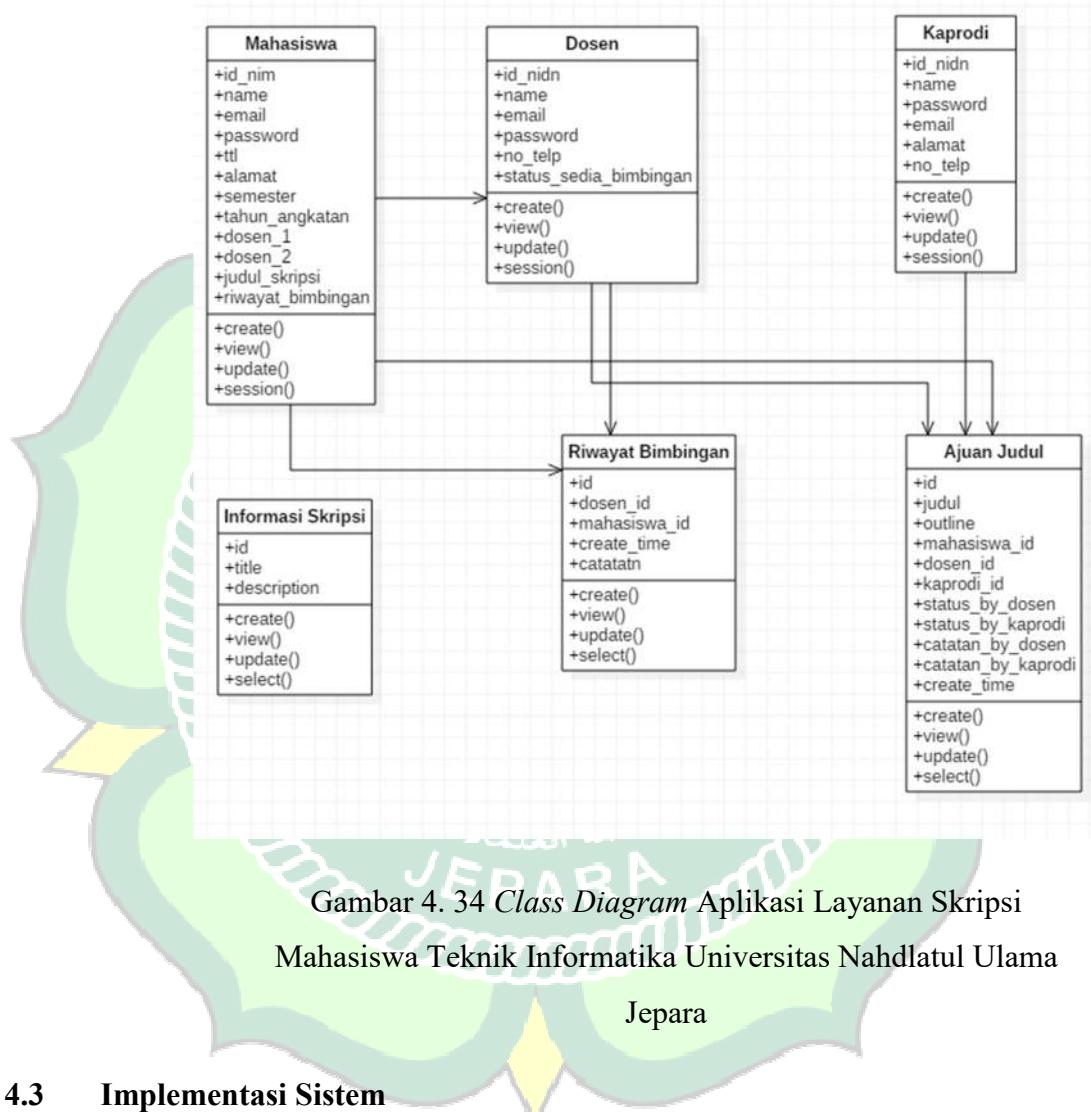
Skripsi Mahasiswa Oleh Kaprodi

Deskripsi dari gambar

- Kaprodi masuk aplikasi dan mengakses halaman ajuan judul skripsi mahasiswa
- Sistem akan meminta data ajuan judul skripsi mahasiswa yang sebelumnya sudah diterima oleh dosen pembimbing
- Sistem akan menampilkan data ajuan judul skripsi mahasiswa dari database
- Kaprodi memilih ajuan judul skripsi mahasiswa
- Sistem akan menampilkan halaman detail ajuan judul skripsi mahasiswa
- Dosen akan menerima atau menolak ajuan skripsi mahasiswa
- Sistem akan menyimpan status judul ajuan skripsi mahasiswa ke database
- Sistem akan menampilkan halaman result status ajuan judul mahasiswa.

4.2.4 Class Diagram

Penggambaran struktur sistem yang didefinisikan melalui kelas per kelas. Berikut ini adalah gambaran class diagram pada Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Universitas Islam Nahdlatul Ulama Jepara.

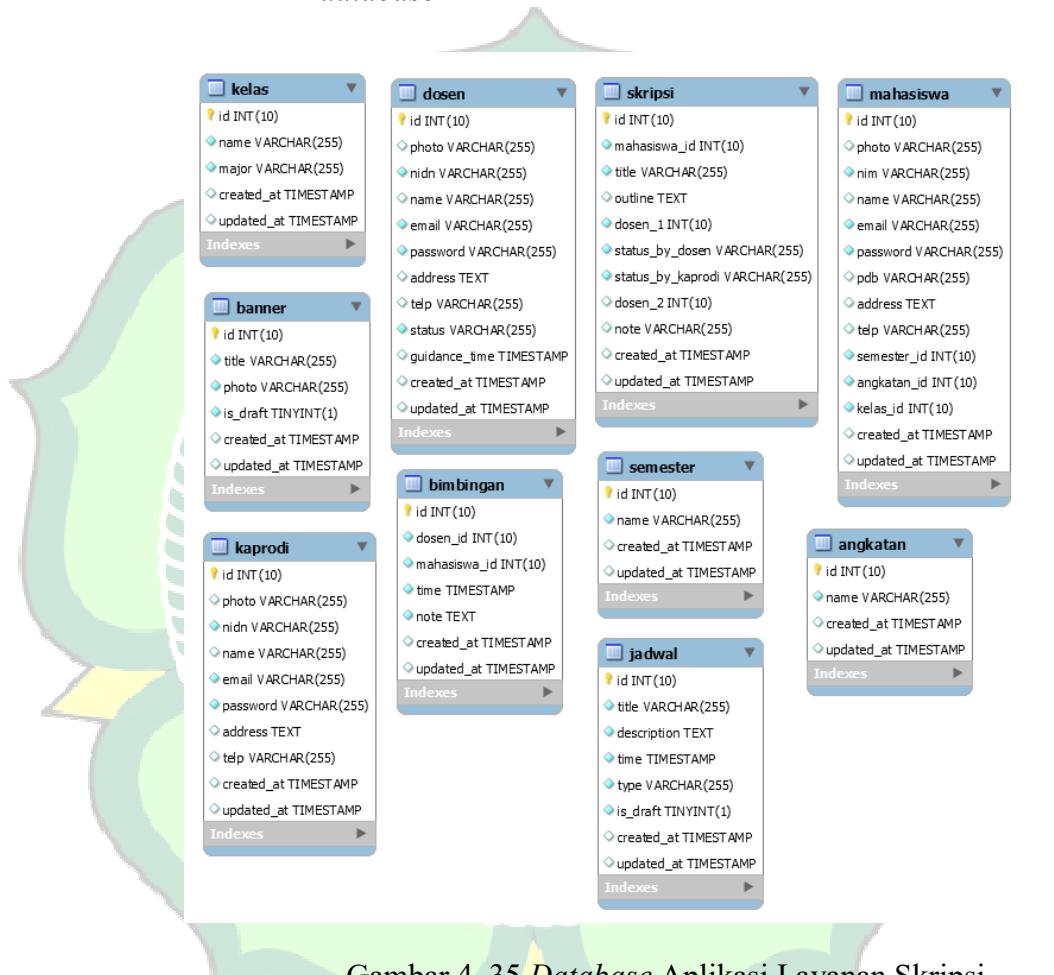


4.3 Implementasi Sistem

Dalam implementasi sistem Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Universitas Islam Nahdlatul Ulama menggunakan *Visual Studio Code* sebagai text editor untuk aplikasi *backend* dan *frontend*, serta MySQL untuk proses pengolahan database.

4.3.1 Basis Data

Berikut ini adalah basis data yang digunakan untuk mengolah data dan informasi pada sistem Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Universitas Islam Nahdlatul Ulama Jepara dengan mengakses Xampp dan menghidupkan Apache dan MySql. Selanjutnya membuka web browser dan masuk ke localhost untuk membuat *database*.



Gambar 4. 35 Database Aplikasi Layanan Skripsi

Berikut adalah nama – nama tabel dan field field yang terdapat dalam database Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Universitas Islam Nahdlatul Ulama Jepara.

a. Tabel Mahasiswa

Digunakan untuk menyimpan data mahasiswa

Tabel 4. 6 *Database* Mahasiswa

No	Nama Field	Tipe Data	Keterangan
1	id	int(10)	primary key
2	photo	varchar(255)	
3	nim	varchar(255)	
4	name	varchar(255)	
5	email	varchar(255)	
6	password	varchar(255)	
7	pdb	varchar(255)	
8	address	text	
9	telp	varchar(255)	
10	semester_id	int(10)	
11	angkatan_id	int(10)	
12	kelas_id	int(10)	
13	created_at	timestamp	
14	updated_at	timestamp	

b. Tabel Dosen

Digunakan untuk menyimpan data dosen

Tabel 4. 7 *Database* Dosen

No	Nama Field	Tipe Data	Keterangan
1	id	int(10)	primary key
2	photo	varchar(255)	
3	nidn	varchar(255)	
4	name	varchar(255)	
5	email	varchar(255)	
6	password	varchar(255)	

7	address	text	
8	telp	varchar(255)	
9	status	varchar(255)	
10	created_at	timestamp	
11	updated_at	timestamp	

c. Tabel Kaprodi

Digunakan untuk menyimpan data kaprodi

Tabel 4. 8 *Database Kaprodi*

No	Nama Field	Tipe Data	Keterangan
1	id	int(10)	primary key
2	photo	varchar(255)	
3	nidn	varchar(255)	
4	name	varchar(255)	
5	email	varchar(255)	
6	password	varchar(255)	
7	address	text	
8	telp	varchar(255)	
9	created_at	timestamp	
10	updated_at	timestamp	

d. Tabel Skripsi

Digunakan untuk menyimpan ajuan judul skripsi mahasiswa

Tabel 4. 9 Database Ajuan Judul Skripsi

No	Nama Field	Tipe Data	Keterangan
1	id	int(10)	primary key
2	mahasiswa_id	int(10)	
3	title	varchar(255)	
4	outline	text	
5	dosen_1	int(10)	
6	status_by_dosen	varchar(255)	
7	status_by_kaprodi	varchar(255)	
8	dosen_2	int(10)	
9	note	varchar(255)	
10	created_at	timestamp	
11	updated_at	timestamp	

e. Tabel Bimbingan

Digunakan untuk menyimpan riwayat bimbingan mahasiswa

Tabel 4. 10 Database Riwayat Bimbingan

No	Nama Field	Tipe Data	Keterangan
1	id	int(10)	primary key
2	dosen_id	int(10)	
3	mahasiswa_id	int(10)	
4	time	timestamp	
5	note	text	
6	created_at	timestamp	
7	updated_at	timestamp	

f. Tabel Informasi

Digunakan untuk menyimpan informasi terkait pelaksanaan skripsi

Tabel 4. 11 *Database* Informasi Skripsi

No	Nama Field	Tipe Data	Keterangan
1	id	int(10)	primary key
2	title	varchar(255)	
3	description	varchar(255)	
4	time	timestamp	
5	type	varchar(255)	
6	is_draft	TINYINT(1)	
7	created_at	timestamp	
8	updated_at	timestamp	

4.3.2 Pengkodean

Berikut ini merupakan pengkodean sistem yang digunakan untuk membuat Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Unisnu Jepara dengan mengkases aplikasi *Xampp* untuk menghidupkan *MySql*. Aplikasi ini menggunakan *Framework Laravel* sebagai admin dan *Android Kotlin* untuk membangun aplikasi android. Dalam pengkodean aplikasi ini menggunakan 2 aplikasi IDE, yaitu *Visual Studio Code* untuk pengkodean admin dan *Android Studio* untuk pengkodean android. Adapun isi kode dapat dilihat sebagai berikut :

1. Kode Menambahkan Mahasiswa

Berikut ini merupakan sebagian kode program yang digunakan untuk menambahkan data mahasiswa pada sistem.



```

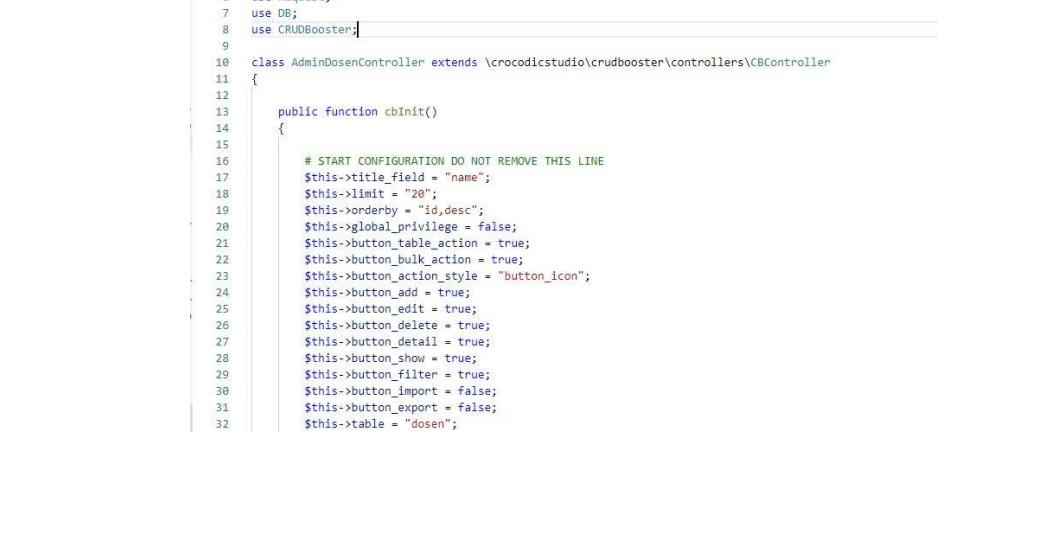
AdminMahasiswaController.php > AdminMahasiswaController.php
app > Http > Controllers > AdminMahasiswaController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Session;
6  use Request;
7  use DB;
8  use CRUDBooster;
9
10 class AdminMahasiswaController extends \crocodicstudio\crudbooster\controllers\CBController
11 {
12
13     public function cbInit()
14     {
15
16         # START CONFIGURATION DO NOT REMOVE THIS LINE
17         $this->title_field = "name";
18         $this->limit = "20";
19         $this->orderby = "id,desc";
20         $this->global_privilege = false;
21         $this->button_table_action = true;
22         $this->button_bulk_action = true;
23         $this->button_action_style = "button_icon";
24         $this->button_add = true;
25         $this->button_edit = true;
26         $this->button_delete = true;
27         $this->button_detail = true;
28         $this->button_show = true;
29         $this->button_filter = true;
30         $this->button_import = true;
31         $this->button_export = true;
32         $this->table = "mahasiswa";

```

Gambar 4. 36 Kode Menambahkan Mahasiswa

2. Kode Menambahkan Dosen

Berikut ini merupakan sebagian kode program yang digunakan untuk menambahkan data dosen.



```

AdminMahasiswaController.php > AdminDosenController.php
app > Http > Controllers > AdminDosenController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Session;
6  use Request;
7  use DB;
8  use CRUDBooster;
9
10 class AdminDosenController extends \crocodicstudio\crudbooster\controllers\CBController
11 {
12
13     public function cbInit()
14     {
15
16         # START CONFIGURATION DO NOT REMOVE THIS LINE
17         $this->title_field = "name";
18         $this->limit = "20";
19         $this->orderby = "id,desc";
20         $this->global_privilege = false;
21         $this->button_table_action = true;
22         $this->button_bulk_action = true;
23         $this->button_action_style = "button_icon";
24         $this->button_add = true;
25         $this->button_edit = true;
26         $this->button_delete = true;
27         $this->button_detail = true;
28         $this->button_show = true;
29         $this->button_filter = true;
30         $this->button_import = false;
31         $this->button_export = false;
32         $this->table = "dosen";

```

Gambar 4. 37 Kode Menambahkan Dosen

3. Kode Menambahkan Kaprodi

Berikut ini merupakan sebagian kode program yang digunakan untuk menambahkan data kaprodi.



```

AdminMahasiswaController.php AdminKaprodiController.php ×
app > Http > Controllers > AdminKaprodiController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Session;
6  use Request;
7  use DB;
8  use CRUDBooster;
9
10 class AdminKaprodiController extends \crocodicstudio\crudbooster\controllers\CBController
11 {
12
13     public function cbInit()
14     {
15
16         # START CONFIGURATION DO NOT REMOVE THIS LINE
17         $this->title_field = "name";
18         $this->limit = "20";
19         $this->orderby = "id,desc";
20         $this->global_privilege = false;
21         $this->button_table_action = true;
22         $this->button_bulk_action = true;
23         $this->button_action_style = "button_icon";
24         $this->button_add = true;
25         $this->button_edit = true;
26         $this->button_delete = true;
27         $this->button_detail = true;
28         $this->button_show = true;
29         $this->button_filter = true;
30         $this->button_import = false;
31         $this->button_export = false;
32         $this->table = "kaprodi";

```

Gambar 4. 38 Kode Menambahkan Kaprodi

4. Kode Menambahkan Informasi Seputar Skripsi

Berikut ini merupakan sebagian kode program yang digunakan untuk menambahkan informasi terkait proses pelaksanaan skripsi



```

app > Http > Controllers > AdminInformasiController.php X
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Session;
6  use Request;
7  use DB;
8  use CRUDBooster;
9
10 class AdminInformasiController extends \crocodicstudio\crudbooster\controllers\CBController
11 {
12
13     public function cbInit()
14     {
15
16         # START CONFIGURATION DO NOT REMOVE THIS LINE
17         $this->title_field = "title";
18         $this->limit = "20";
19         $this->orderby = "id,desc";
20         $this->global_privilege = false;
21         $this->button_table_action = true;
22         $this->button_bulk_action = true;
23         $this->button_action_style = "button_icon";
24         $this->button_add = true;
25         $this->button_edit = true;
26         $this->button_delete = true;
27         $this->button_detail = true;
28         $this->button_show = true;
29         $this->button_filter = true;
30         $this->button_import = false;
31         $this->button_export = false;
32         $this->table = "informasi";

```

Gambar 4. 39 Kode Menambahkan Informasi Seputar Skripsi

5. Kode Login

Berikut ini merupakan sebagian kode program yang digunakan untuk *login*



```

1  class LoginFragment : Fragment(), View.OnClickListener, OnBackPressedListener {
2
3      private val vm: LoginVM by inject()
4
5      override fun onCreateView(
6          inflater: LayoutInflater, container: ViewGroup?,
7          savedInstanceState: Bundle?
8      ): View? {
9          // Inflate the layout for this fragment
10         return inflater.inflate(R.layout.fragment_login, container, attachToRoot: false)
11     }
12
13     override fun onActivityCreated(savedInstanceState: Bundle?) {
14         super.onActivityCreated(savedInstanceState)
15
16         vm.loginState.observe(viewLifecycleOwner, startObservable)
17
18         LoginFragmentBTN.setOnClickListener(this)
19     }
20
21 }

```

Gambar 4. 40 Kode *Login*

```

44     override fun onClick(v: View?) {
45         when (v?.id) {
46             R.id.LoginFragmentBTN -> {
47                 progressBarHolderLoginCL.visibility = View.VISIBLE
48                 val username = usernameLoginFragmentEDT.text.toString()
49                 val password = passwordLoginFragmentEDT.text.toString()
50                 if (username.isNullOrEmpty() || password.isNullOrEmpty()) {
51                     Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
52                 } else if (username.isNullOrEmpty()) {
53                     Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
54                 } else if (password.isNullOrEmpty()) {
55                     Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
56                 } else {
57                     vm.login(username, password)
58                 }
59             }
60         }
61     }
62 }
63 }
64 }
65 }
66 }
67 }
68 }
69 }
70 }
71 }
72 }
73 }
74 }
75 }
76 }
77 }
78 }
79 }
80 }
81 }
82 }
83 }
84 }
85 }
86 }
87 }

```

Gambar 4. 41 Kode Login 2

```

67     private val startObservable = Observable<LoginState> { login ->
68         when (login) {
69             is LoginSuccessMahasiswa -> {
70                 progressBarHolderLoginCL.visibility = View.GONE
71             }
72             is LoginSuccessDosen -> {
73                 progressBarHolderLoginCL.visibility = View.GONE
74             }
75             FirebaseInstanceId.getInstance().instanceId.addOnSuccessListener { instanceIdResult ->
76                 val deviceToken = instanceIdResult.token
77                 val msg = "InstanceID Token: ${deviceToken}"
78                 Utils.INSTANCE.snackBarMessage(this.requireView(), msg)
79                 vm.updateToken(
80                     login.topLoginResponse.data.id,
81                     deviceToken,
82                     login.topLoginResponse.data.type.toString()
83                 )
84             }
85         }
86         findNavController().navigate(R.id.action_LoginFragment_to_dosenActivity2)
87     }

```

Gambar 4. 42 Kode Login 3

```

89     }
90     is TokenUpdatedSuccess -> {
91         progressBarHolderLoginCL.visibility = View.GONE
92         Toast.makeText(
93             this.requireContext(),
94             login.topUpdateTokenResponse.api_message,
95             Toast.LENGTH_LONG
96         ).show()
97     }
98     is LoginSuccessKaprodi -> {
99         progressBarHolderLoginCL.visibility = View.GONE
100        findNavController().navigate(R.id.action_LoginFragment_to_kaprodiActivity)
101    }
102    is LoginInNotSuccess -> {
103        progressBarHolderLoginCL.visibility = View.GONE
104        Utils.INSTANCE.snackBarMessage(
105            this.requireView(),
106            login.topLoginResponse.api_message
107        )
108    }
109    is ErrorState -> {
110        progressBarHolderLoginCL.visibility = View.GONE
111    }
112 }
113 }
114 }
115 }

```

Gambar 4. 43 Kode Login 4

6. Kode Ajuan Judul Skripsi Mahasiswa

Berikut ini merupakan sebagian kode program yang digunakan untuk proses mengajukan judul skripsi mahasiswa

```

51     private val startObservable = Observer<AjuanState> { ajuan ->
52         when(ajuan){
53             is AddSkripsiSuccess ->{
54                 progressBarHolderLoginCL.visibility = View.GONE
55                 Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Ajukan Sukses...")
56                 findNavController().navigate(R.id.action_addAjuanFragment_to_homeActivity2)
57
58                 judulAddAjuanFragmentEDT.setText("")
59                 outlineAddAjuanFragmentEDT.setText("")
60             }
61             is DetailFormDataLoaded ->{
62                 progressBarHolderLoginCL.visibility = View.GONE
63                 val data = ajuan.detailFormDomain
64                 nameAddAjuanFragmentTV.text = data.name
65                 nisAddAjuanFragmentTV.text = data.nim
66                 drAddAjuanFragmentTV.text = data.dosen
67             }
68             is ErrorState ->{
69                 progressBarHolderLoginCL.visibility = View.GONE
70             }
71         }
72     }
73 }
74
75 }
76

```

Gambar 4. 44 Kode Ajuan Judul Skripsi Mahasiswa

```

22 class AddAjuanFragment : Fragment(), View.OnClickListener {
23
24     val vm:AjuanViewModel by inject()
25
26     override fun onCreateView(
27         inflater: LayoutInflater, container: ViewGroup?,
28         savedInstanceState: Bundle?
29     ): View? {
30         // Inflate the layout for this fragment
31         return inflater.inflate(R.layout.fragment_add_ajuan, container, attachToRoot: false)
32     }
33
34     override fun onActivityCreated(savedInstanceState: Bundle?) {
35         super.onActivityCreated(savedInstanceState)
36
37         progressBarHolderLoginCL.visibility = View.VISIBLE
38
39         vm.ajuanState.observe(viewLifecycleOwner, startObservable)
40
41         (activity as AppCompatActivity).supportActionBar?.title = "Form ajuan judul dan outline"
42
43         ajuanAddAjuanFragmentBTN.setOnClickListener(this)
44         backAddAjuanFragmentIV.setOnClickListener(this)
45
46         val id = Hawk.get( key: "mahasiswa_id", defaultValue: "")
47         vm.getDetailForm(id)
48     }
49
50

```

Gambar 4. 45 Kode Ajuan Judul Skripsi Mahasiswa 2

```

77    override fun onClick(v: View?) {
78        when(v?.id){
79            R.id.ajukanAddAjuanFragmentBTN ->{
80                val judul = judulAddAjuanFragmentEDT.text.toString()
81                val outline = outlineAddAjuanFragmentEDT.text.toString()
82                val id = Hawk.get( key: "mahasiswa_id", defaultValue: "")
83                if (judul.isNullOrEmpty() || outline.isNullOrEmpty()){
84                    Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
85                }else if (judul.isNullOrEmpty()){
86                    Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
87                }else if (outline.isNullOrEmpty()){
88                    Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
89                }else{
90                    progressBarHolderLoginCL.visibility = View.VISIBLE
91                    vm.addSkripsi(id.toInt(), judul, outline)
92                }
93            }
94        }
95    }
96    R.id.backAddAjuanFragmentIV ->{
97        this.activity?.onBackPressed()
98    }
99}
100}
101}
102}

```

Gambar 4. 46 Kode Ajuan Judul Skripsi Mahasiswa 3

7. Kode Ajukan Bimbingan

Berikut ini merupakan sebagian kode program yang digunakan untuk memberikan notifikasi kepada dosen pembimbing bahwa ada mahasiswa yang mengajukan bimbingan

```

76    private val starObservable = Observer<BimbinganState> { bimbingan ->
77        when(bimbingan){
78            is BimbinganDataLoaded ->{
79                progressBarHolderLoginCL.visibility = View.GONE
80                bimbinganFragmentRV.visibility = View.VISIBLE
81                dataBelumAdaFragmentTV.visibility = View.GONE
82                adapterBimbingan.clear()
83                bimbingan.bimbinganDomain.map { it: BimbinganDomain
84                    adapterBimbingan.add(BimbinganItemView(it))
85                }
86            }
87        }
88        is BimbinganDosenDataLoaded ->{
89            progressBarHolderLoginCL.visibility = View.GONE
90            dosenBimbinganFragmentRV.visibility = View.VISIBLE
91            dataBelumAdaFragmentTV.visibility = View.GONE
92            adapterDosenBimbingan.clear()
93            bimbingan.bimbinganDosenListDomain.map { it: BimbinganDosenListDomain
94                adapterDosenBimbingan.add(BimbinganDosenListItemView(it, mahasiswa, onClickDosenListener: this))
95            }
96        }
97    }

```

Gambar 4. 47 Kode Ajukan Bimbingan

```

98     }
99
100    }
101   }
102   }
103   }
104   }
105   }
106   }
107   }
108   }
109   }
110   }
111   }
112   }
113   }
114   }
115   }
116   @Override
117   public void onClickSendNotif(int dosen_id, int mahasiswa_id) {
118       Utils.INSTANCE.snackBarMessage(this.requireView(), message: "sending notification to dosen")
119   }
120 }
121

```

Gambar 4. 48 Kode Ajukan Bimbingan 2

8. Kode Edit Status Jadwal Bimbingan Dosen

Berikut ini merupakan sebagian kode program yang digunakan untuk mengedit staus jadwal ketersediaan dosen pembimbing dalam memberikan bimbingan

```

76     private val starObservable = Observer<BimbinganState> { bimbingan ->
77         when(bimbingan){
78             is BimbinganDataLoaded ->{
79                 progressBarHolderLoginCL.visibility = View.GONE
80                 bimbinganFragmentRV.visibility = View.VISIBLE
81                 dataBelumAdaFragmentTV.visibility = View.GONE
82                 adapterBimbingan.clear()
83                 bimbingan.bimbinganDomain.map { it: BimbinganDomain
84                     adapterBimbingan.add(BimbinganItemView(it))
85                 }
86             }
87             is BimbinganDosenDataLoaded ->{
88                 progressBarHolderLoginCL.visibility = View.GONE
89                 dosenBimbinganFragmentRV.visibility = View.VISIBLE
90                 dataBelumAdaFragmentTV.visibility = View.GONE
91                 adapterDosenBimbingan.clear()
92                 bimbingan.bimbinganDosenListDomain.map { it: BimbinganDosenListDomain
93                     adapterDosenBimbingan.add(BimbinganDosenListItemView(it, mahasiswa, onClickDosenListener: this))
94                 }
95             }
96         }
97     }

```

Gambar 4. 49 Kode Edit Status Jadwal Bimbingan Dosen

```

98     }
99
100    }
101   }
102   }
103   }
104   }
105   }
106   }
107   }
108   }
109   }
110   }
111   }
112   }
113   }
114   }
115
116   *| override fun onClickSendNotif(dosen_id: Int, mahasiswa_id: Int) {
117       Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Send Success")
118       vm.sendNotif(dosen_id, mahasiswa_id)
119   }
120
121

```

Gambar 4. 50 Kode Edit Status Jadwal Bimbingan Dosen 2

9. Kode Acc/Tolak Ajuan Judul Skripsi

Berikut ini merupakan sebagian kode program yang digunakan untuk memberikan status diterima atau ditolak pada ajuan judul skripsi mahasiswa.

```

78     private val starObservable = Observable<DosenAjuanState>{ dosenUpdate ->
79         when(dosenUpdate){
80             is UpdateSkripsiByDosenSuccess ->{
81                 progressBarHolderLoginCL.visibility = View.GONE
82                 this.activity?.onBackPressed()
83             }
84             is DosenDetailSkripsiDataLoaded ->{
85                 progressBarHolderLoginCL.visibility = View.GONE
86                 val data = dosenUpdate.dosenDetailSkripsiDomain[0]
87                 judulDosenDetailSkripsiTV.text = data.title
88                 namaMahasiswaDetailSkripsiFragmentTV.text = data.mahasiswa
89                 tanggalDosenDetailSkripsiFragmentTV.text = data.created_at
90                 statusAjuanDosenDetailFragmentTV.text = data.status_by_dosen
91                 outlineDosenDetailSkripsiFragmentTV.text = data.outline
92                 catatanDosenDetailSkripsiEDT.text = data
93             }
94             when(data.status_by_dosen){
95                 "Approved" ->{
96                     statusAjuanDosenDetailFragmentTV.setTextColor(ContextCompat.getColor(this.requireContext(), R.color.green))
97                 }
98                 "Rejected" ->{
99                     statusAjuanDosenDetailFragmentTV.setTextColor(ContextCompat.getColor(this.requireContext(), R.color.red))
100                }
101                "Waiting" ->{
102                    statusAjuanDosenDetailFragmentTV.setTextColor(ContextCompat.getColor(this.requireContext(), R.color.blue))
103                }
104            }
105        }

```

Gambar 4. 51 Kode Acc/Tolak Ajuan Judul Skripsi

```

52     override fun onClick(v: View?) {
53         when(v?.id){
54             R.id.tolaDosenDetailSkripsiBTN ->{
55                 val note = catatanDosenDetailSkripsiEDT.text.toString()
56                 if (note.isNotEmpty()){
57                     progressBarHolderLoginCL.visibility = View.VISIBLE
58                     vm.updateSkripsiByDosen(id, status_by_dosen: "Rejected", note)
59                 }else{
60                     Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
61                 }
62             }
63             R.id.accDosenDetailSkripsiBTN ->{
64                 val note = catatanDosenDetailSkripsiEDT.text.toString()
65                 if (note.isNotEmpty()){
66                     progressBarHolderLoginCL.visibility = View.VISIBLE
67                     vm.updateSkripsiByDosen(id, status_by_dosen: "Approved", note)
68                 }else{
69                     Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
70                 }
71             }
72         }
73     }
74 }
75
76 }
```

Gambar 4. 52 Kode Acc/Tolak Ajuan Judul Skripsi 2

```

107     is ErrorState ->{
108         progressBarHolderLoginCL.visibility = View.GONE
109         Utils.INSTANCE.snackBarMessage(this.requireView(), dosenUpdate.message.toString())
110     }
111 }
112 }
113 }
114 }
115 }
116 }
```

Gambar 4. 53 Kode Acc/Tolak Ajuan Judul Skripsi 3

10. Kode Input Riwayat Bimbingan

Berikut ini merupakan sebagian kode program yang digunakan untuk input riwayat bimbingan mahasiswa

```

45  override fun onClick(v: View?) {
46      when(v?.id){
47          R.id.submitInputRiwayatFragmentBTN ->{
48              val nim = nimMahaInputRiwayatEDT.text.toString()
49              val time = timeInputRiwayatFragmentEDT.text.toString()
50              val catatan = catatanDosenDetailSkripsiEDT.text.toString()
51
52              if (nim.isNullOrEmpty() && time.isNullOrEmpty() && catatan.isNullOrEmpty()){
53                  Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
54              }else if (nim.isNullOrEmpty()){
55                  Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
56              }else if (time.isNullOrEmpty()){
57                  Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
58              }else if (catatan.isNullOrEmpty()){
59                  Utils.INSTANCE.snackBarMessage(this.requireView(), message: "Silahkan isi dahulu....")
60              }else{
61                  progressBarHolderLoginCL.visibility = View.VISIBLE
62                  vm.dosenInputRiwayat(dosen_id,nim,time,catatan)
63              }
64          }
65          R.id.timeInputRiwayatFragmentEDT ->
66              Utils.INSTANCE.chooseDate(requireContext(), timeInputRiwayatFragmentEDT)
67      }
68  }
69 }
70 }
71 }
72 }

```

Gambar 4. 54 Kode Input Riwayat Bimbingan

```

73  private val starObservable = Observer<InputRiwayatState> { it: InputRiwayatState!
74      when(it){
75          is DosenInputRiwayatSuccess ->{
76              progressBarHolderLoginCL.visibility = View.GONE
77              Utils.INSTANCE.snackBarMessage(this.requireView(), it.topDosenAddBimbinganResponse.api_message)
78          }
79          is UserDataLoaded ->{
80              dosen_id = it.userEntity[0].id
81          }
82          is InputNotSuccess ->{
83              progressBarHolderLoginCL.visibility = View.GONE
84              Utils.INSTANCE.snackBarMessage(this.requireView(), it.topDosenAddBimbinganResponse.api_message)
85          }
86          is ErrorState ->{
87              progressBarHolderLoginCL.visibility = View.GONE
88          }
89      }
90  }
91 }
92 }
93 }
94 }

```

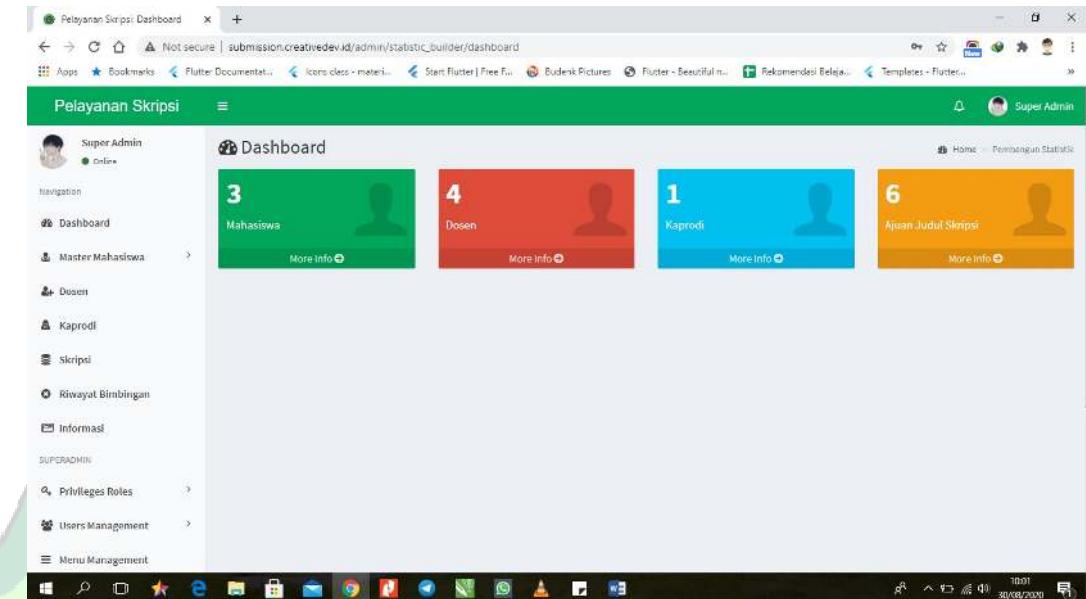
Gambar 4. 55 Kode Input Riwayat Bimbingan 2

4.3.3 Hasil Implementasi

Berikut ini merupakan hasil implementasi dari perancangan antarmuka dan kode program yang telah dirancang sebelumnya. Hasil implementasi dari perancangan tersebut dapat dilihat dengan membuka *web browser* dan membuka aplikasi android. Berikut adalah hasilnya :

1. Halaman Admin

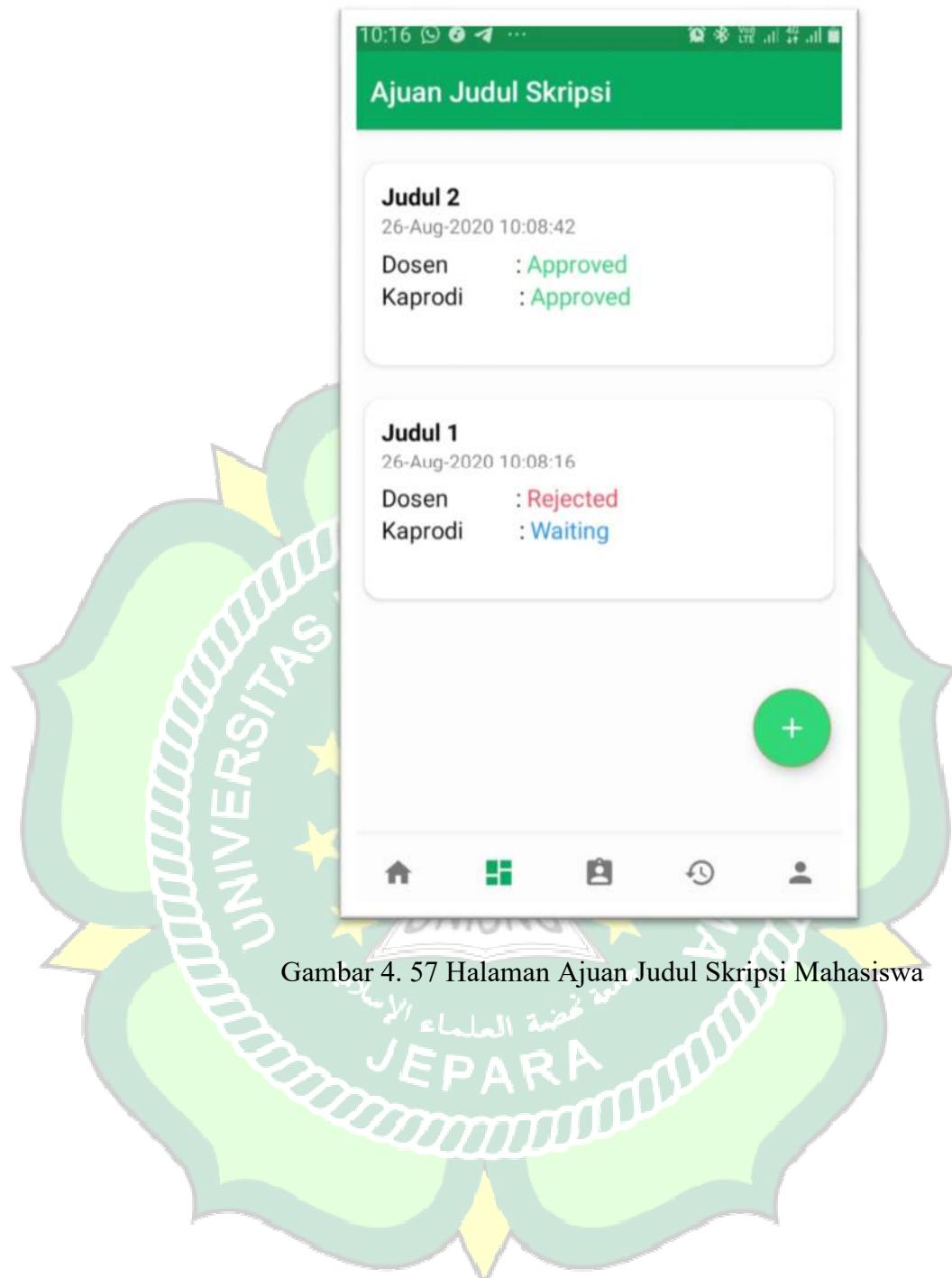
Halaman ini merupakan halaman admin dari aplikasi android yang diakses melalui *web browser*



Gambar 4. 56 Halaman Admin

2. Halaman Ajuan Judul Skripsi Mahasiswa

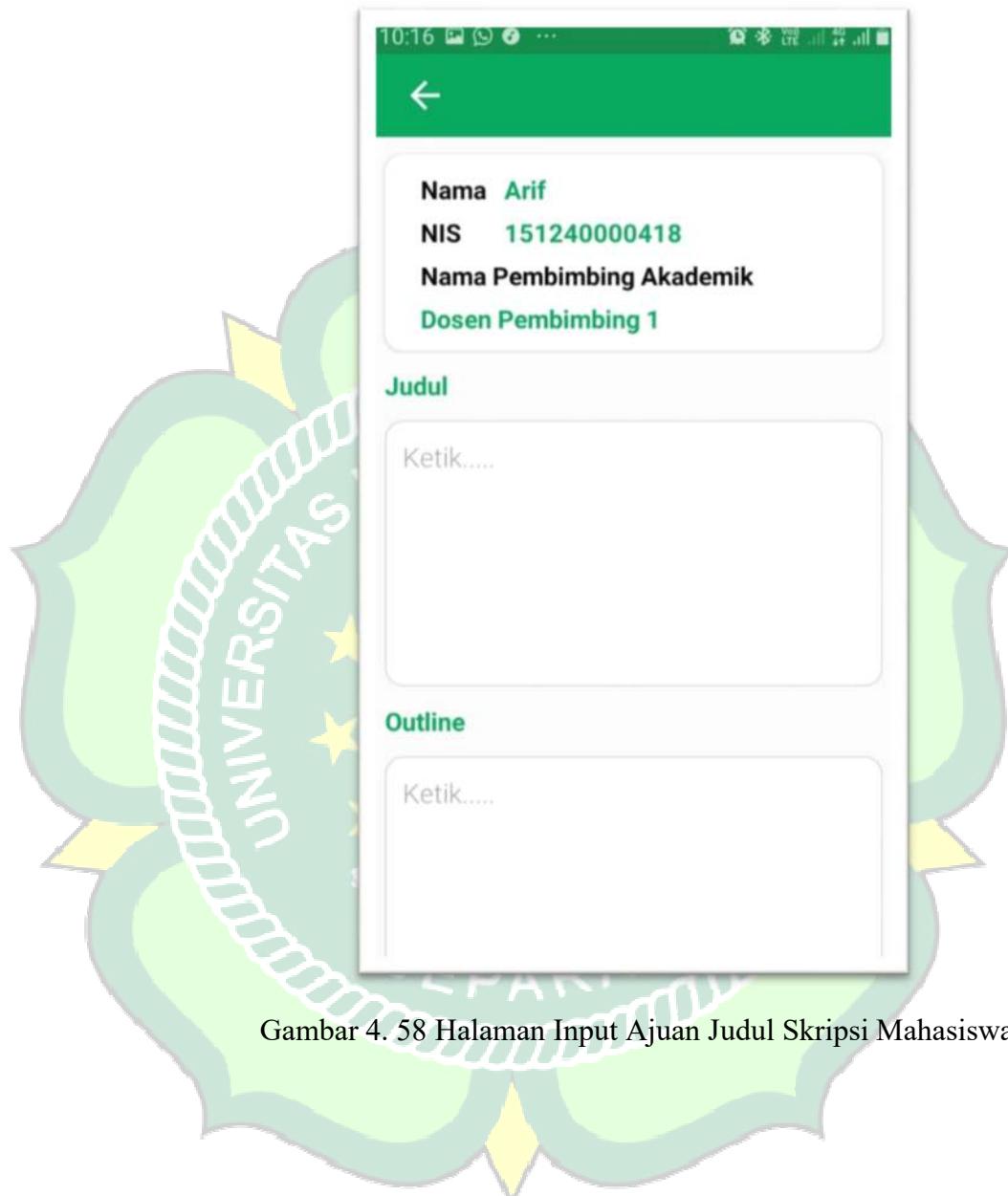
Halaman yang menampilkan seluruh ajuan judul skripsi yang pernah diajukan oleh mahasiswa



Gambar 4. 57 Halaman Ajuan Judul Skripsi Mahasiswa

3. Halaman Input Ajuan Judul Skripsi

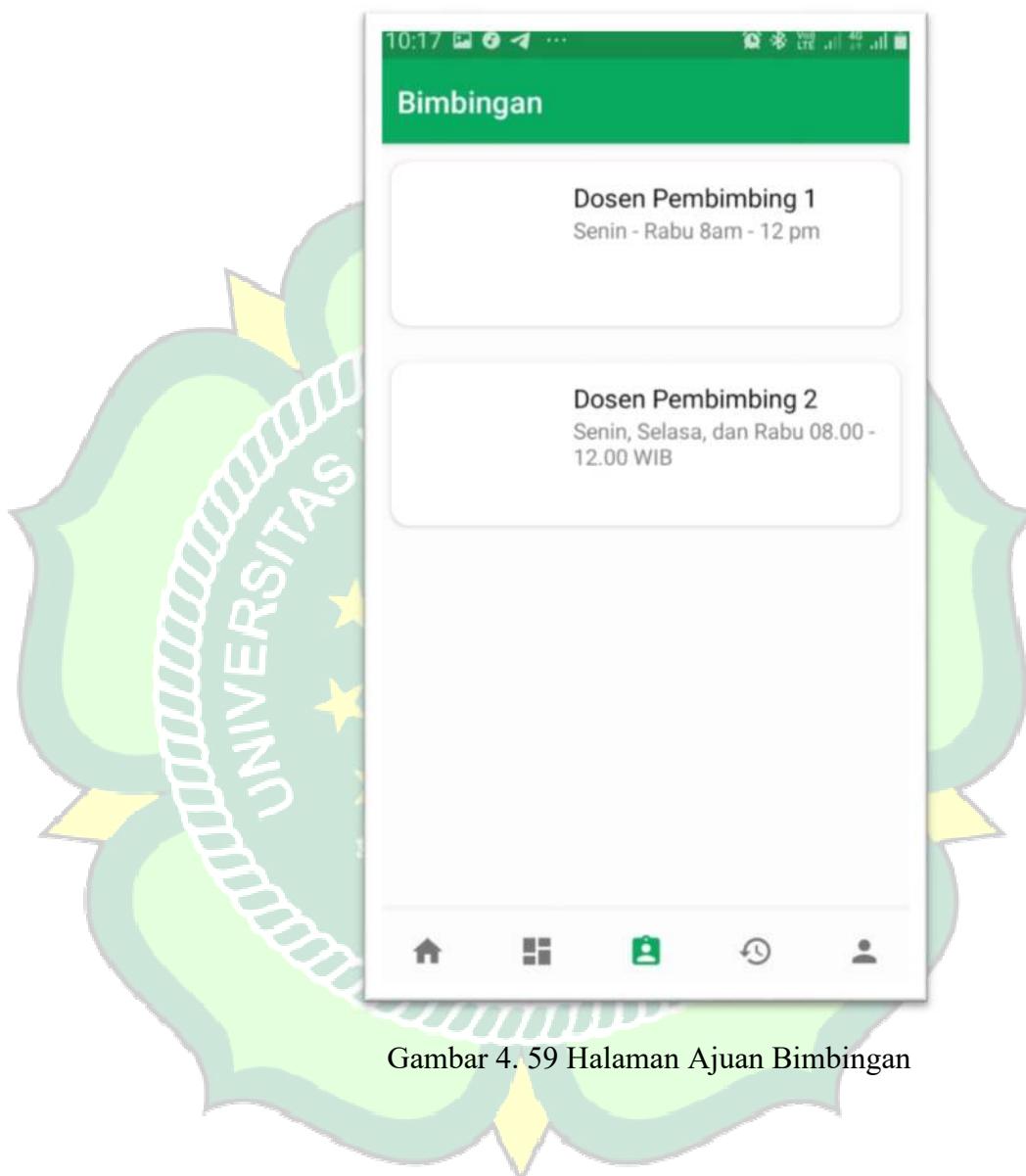
Halaman yang meampulkan form input ajuan judul skripsi mahasiswa



Gambar 4. 58 Halaman Input Ajuan Judul Skripsi Mahasiswa

4. Halaman Ajuan Bimbingan

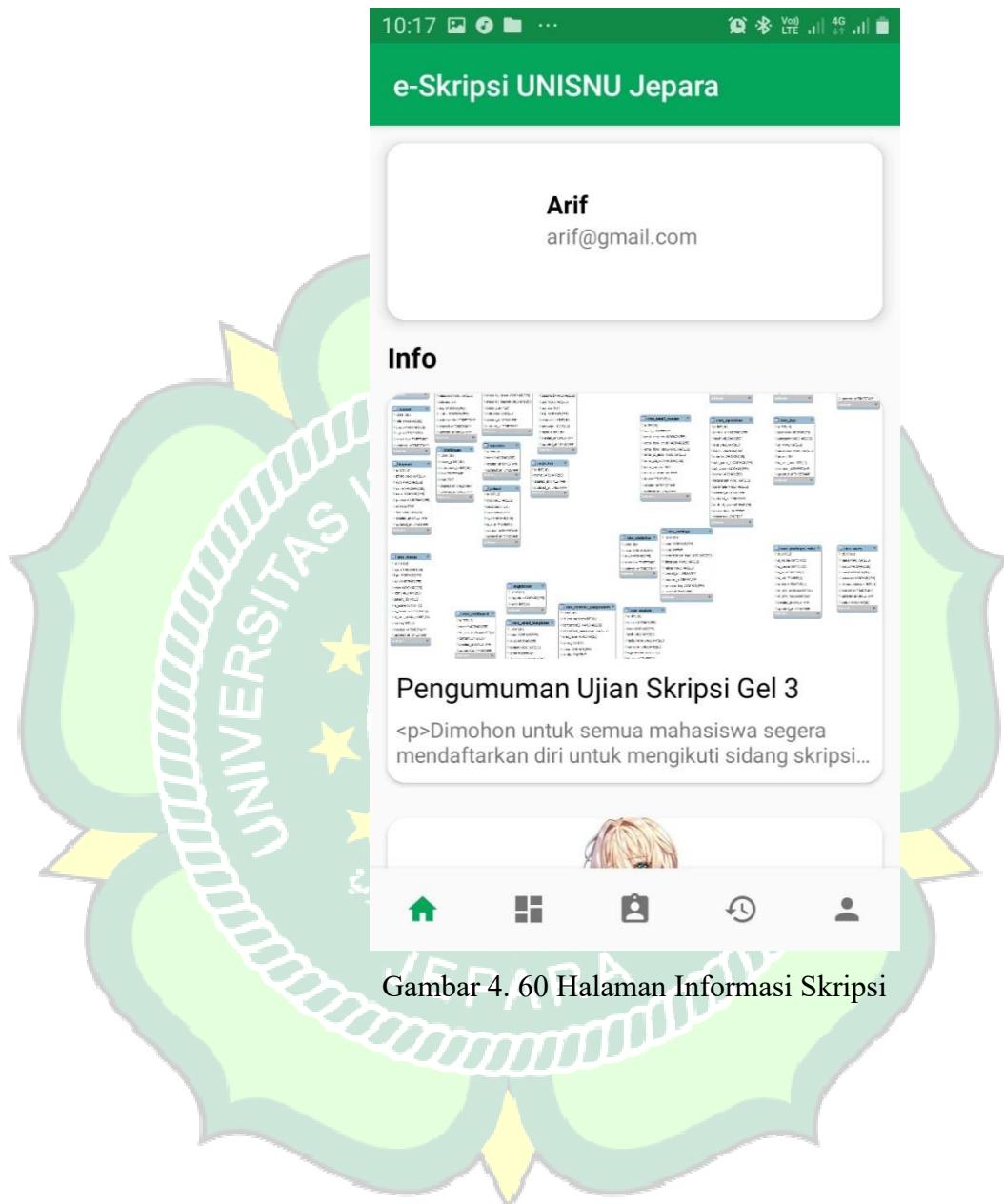
Halaman yang menampilkan jadwal ketersediaan dosen pembimbing 1 dan dosen pembimbing 2 dalam memberikan bimbingan



Gambar 4. 59 Halaman Ajuan Bimbingan

5. Halaman Informasi Skripsi

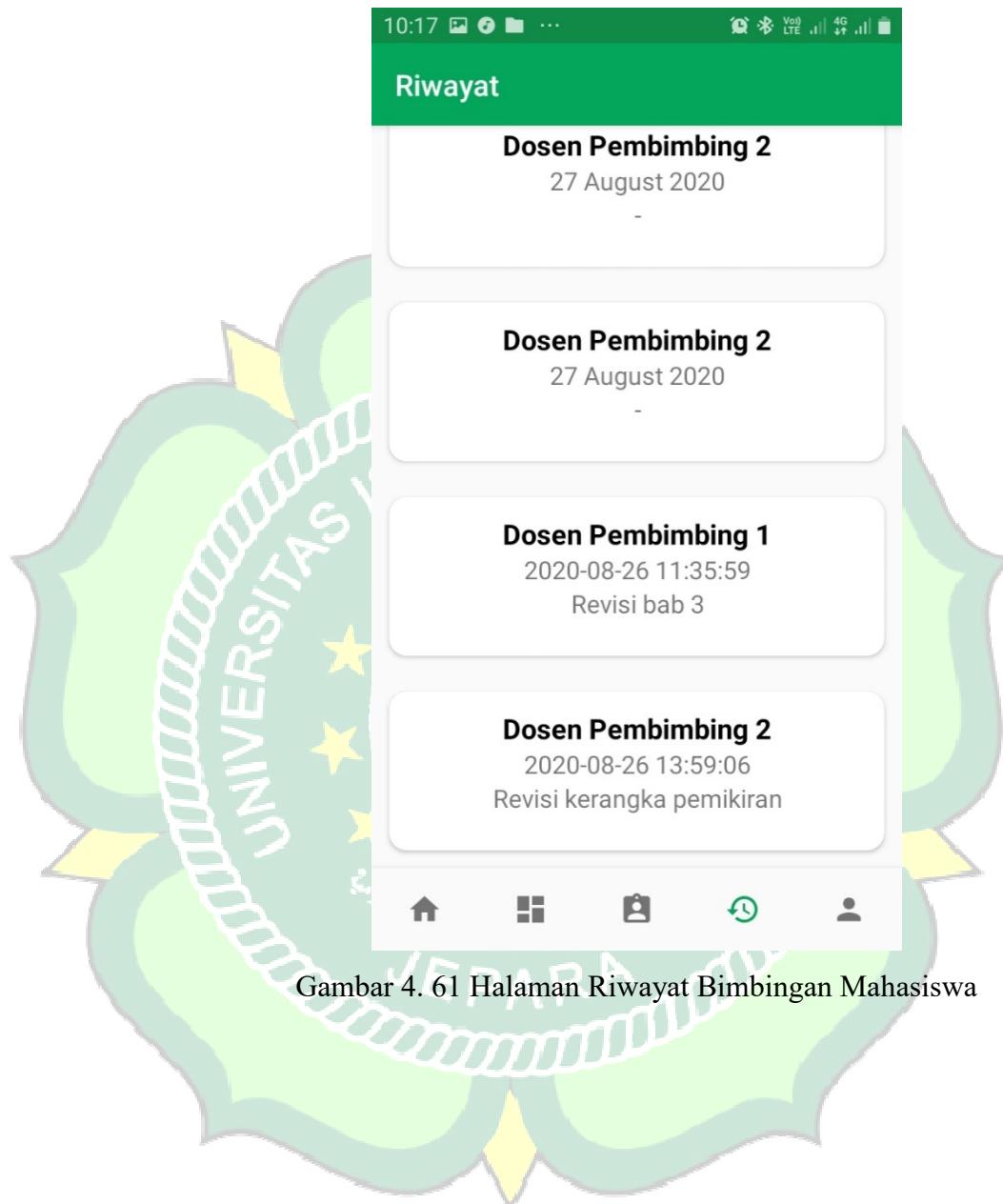
Halaman yang menampilkan informasi terkait proses pelaksanaan skripsi



Gambar 4. 60 Halaman Informasi Skripsi

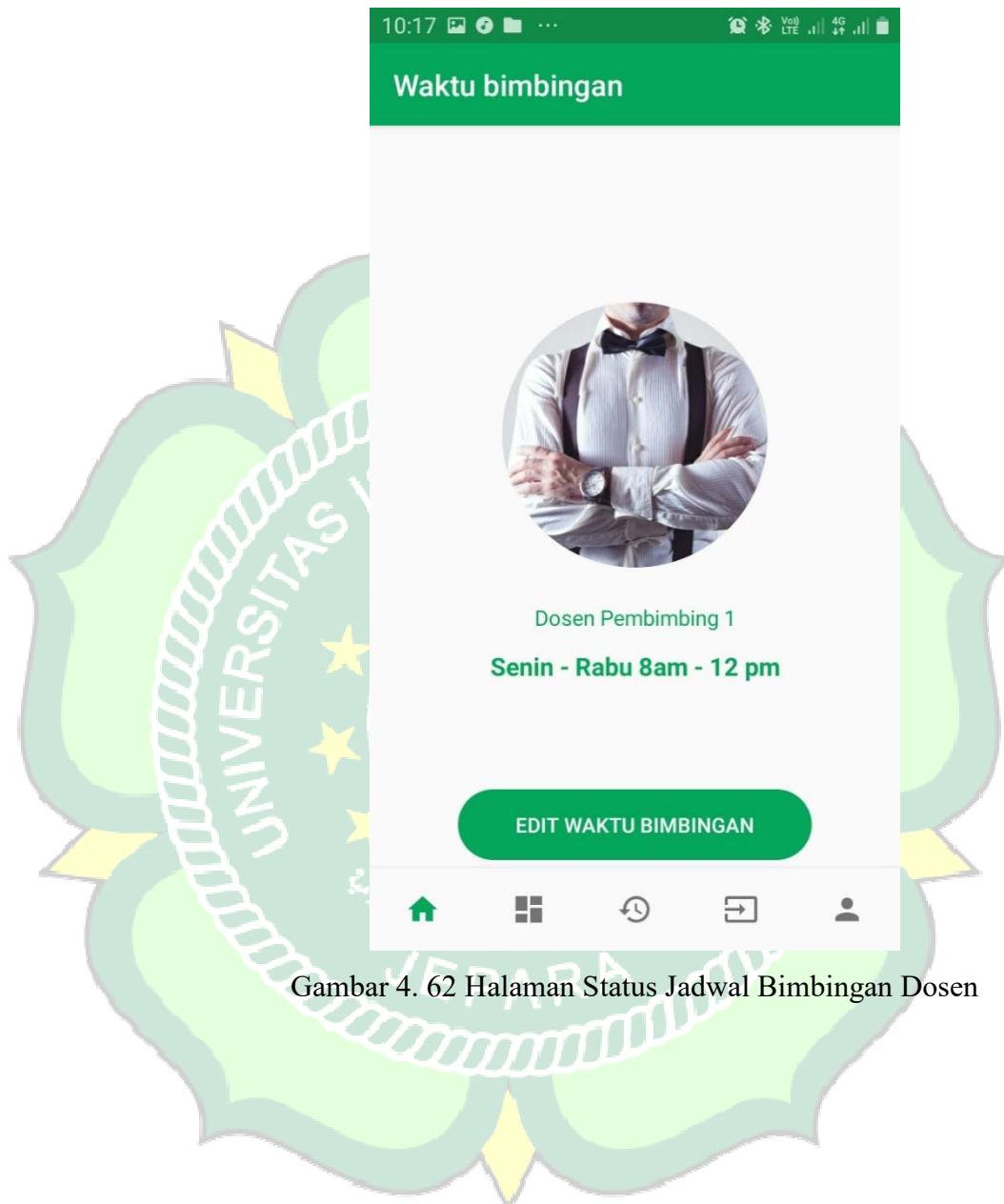
6. Halaman Riwayat Bimbingan Mahasiswa

Halaman yang menampilkan riwayat bimbingan yang telah dilakukan oleh mahasiswa



7. Halaman Status Jadwal Bimbingan Dosen

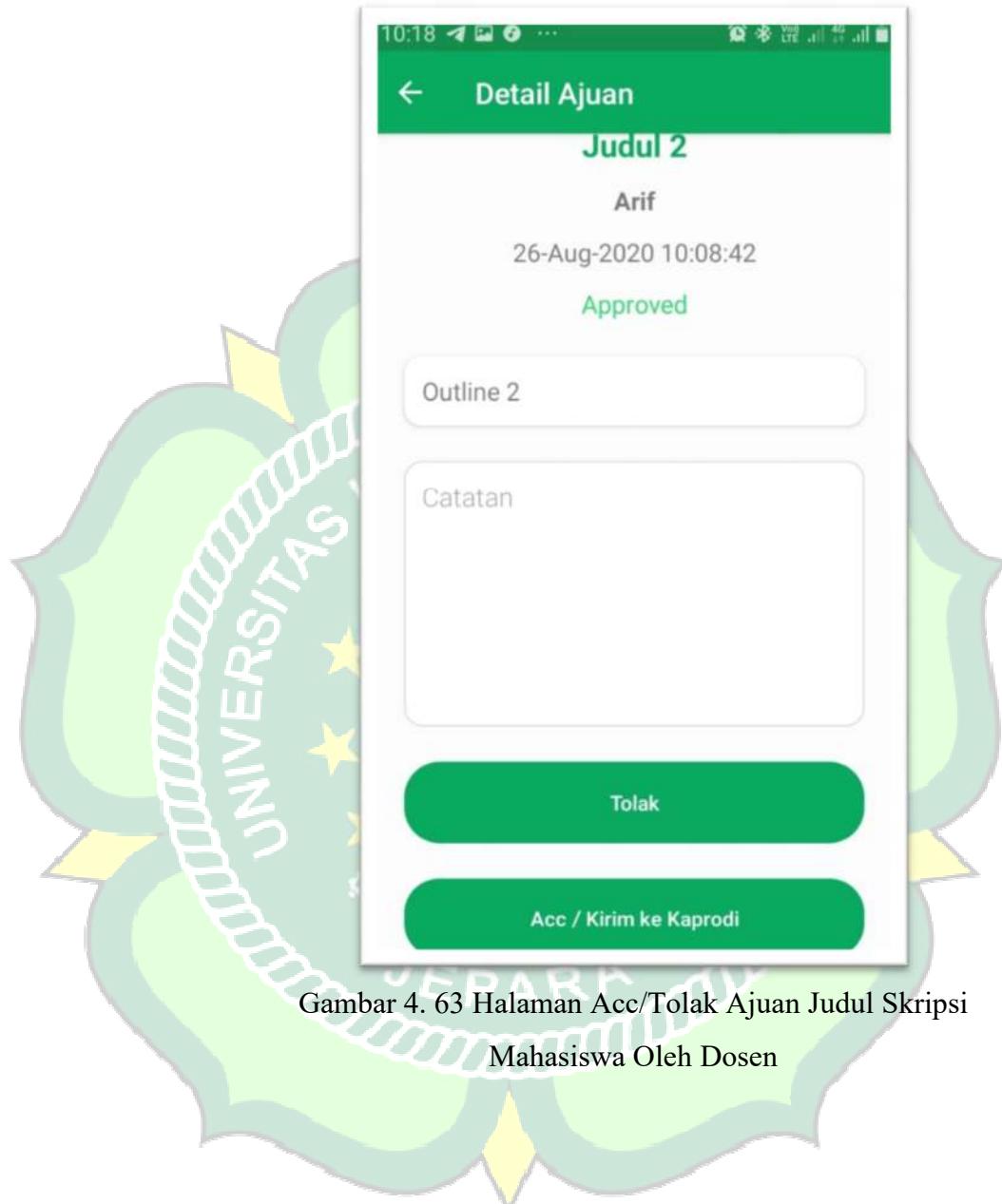
Halaman yang menampilkan status jadwal ketersediaan dosen dalam memberikan bimbingan



Gambar 4. 62 Halaman Status Jadwal Bimbingan Dosen

8. Halaman Acc/Tolak Ajuan Judul Skripsi Mahasiswa

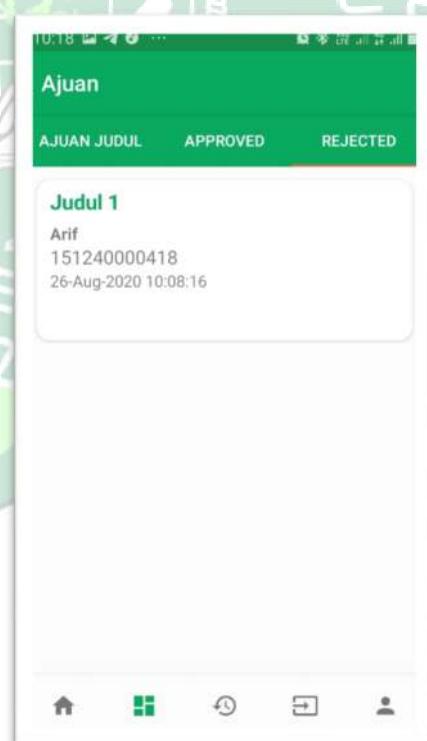
Halaman untuk memberikan status diterima atau ditolak pada ajuan judul skripsi mahasiswa



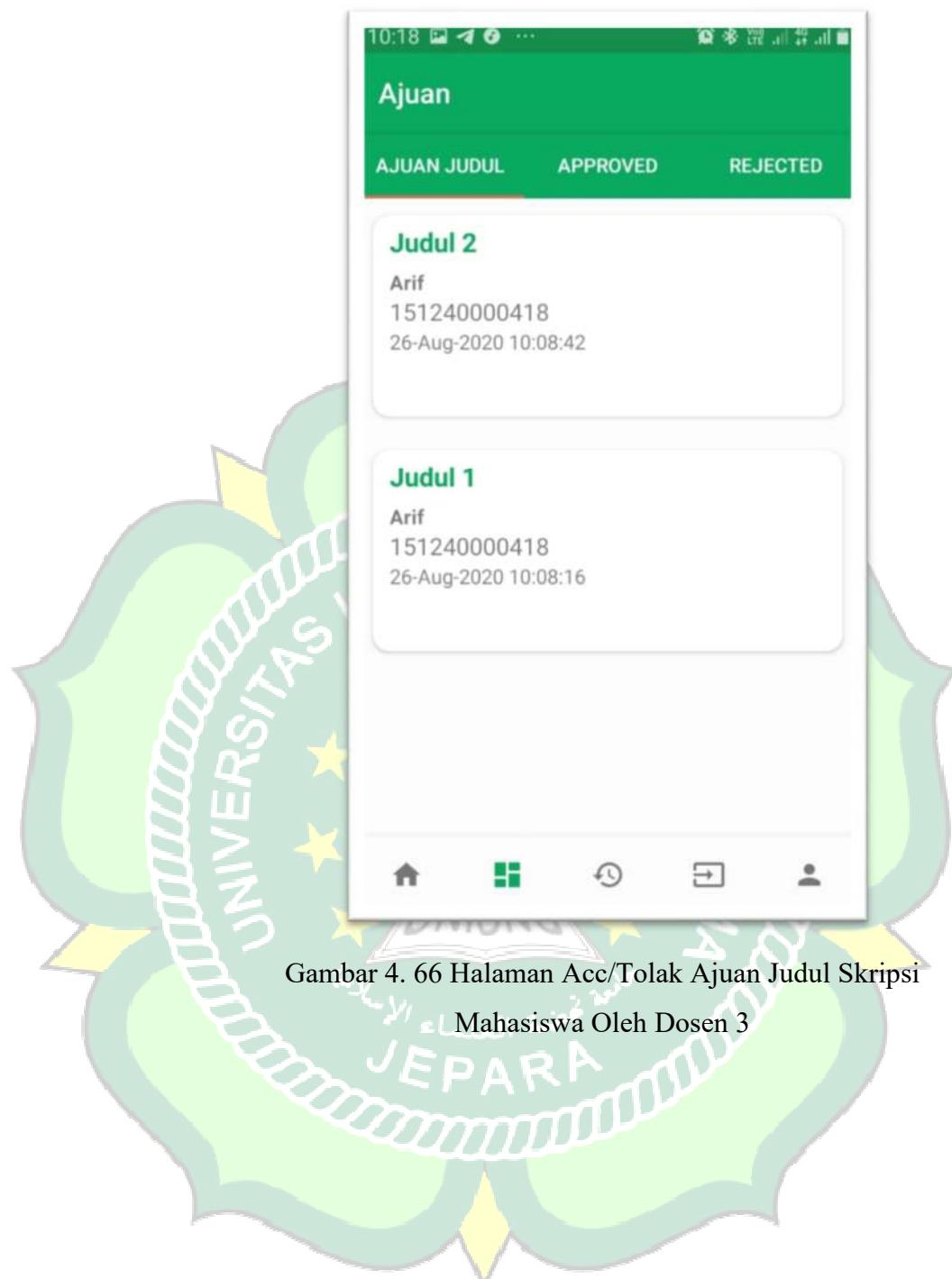
Gambar 4. 63 Halaman Acc/Tolak Ajuan Judul Skripsi
Mahasiswa Oleh Dosen



Gambar 4. 64 Halaman Acc/Tolak Ajuan Judul Skripsi
Mahasiswa Oleh Dosen 2

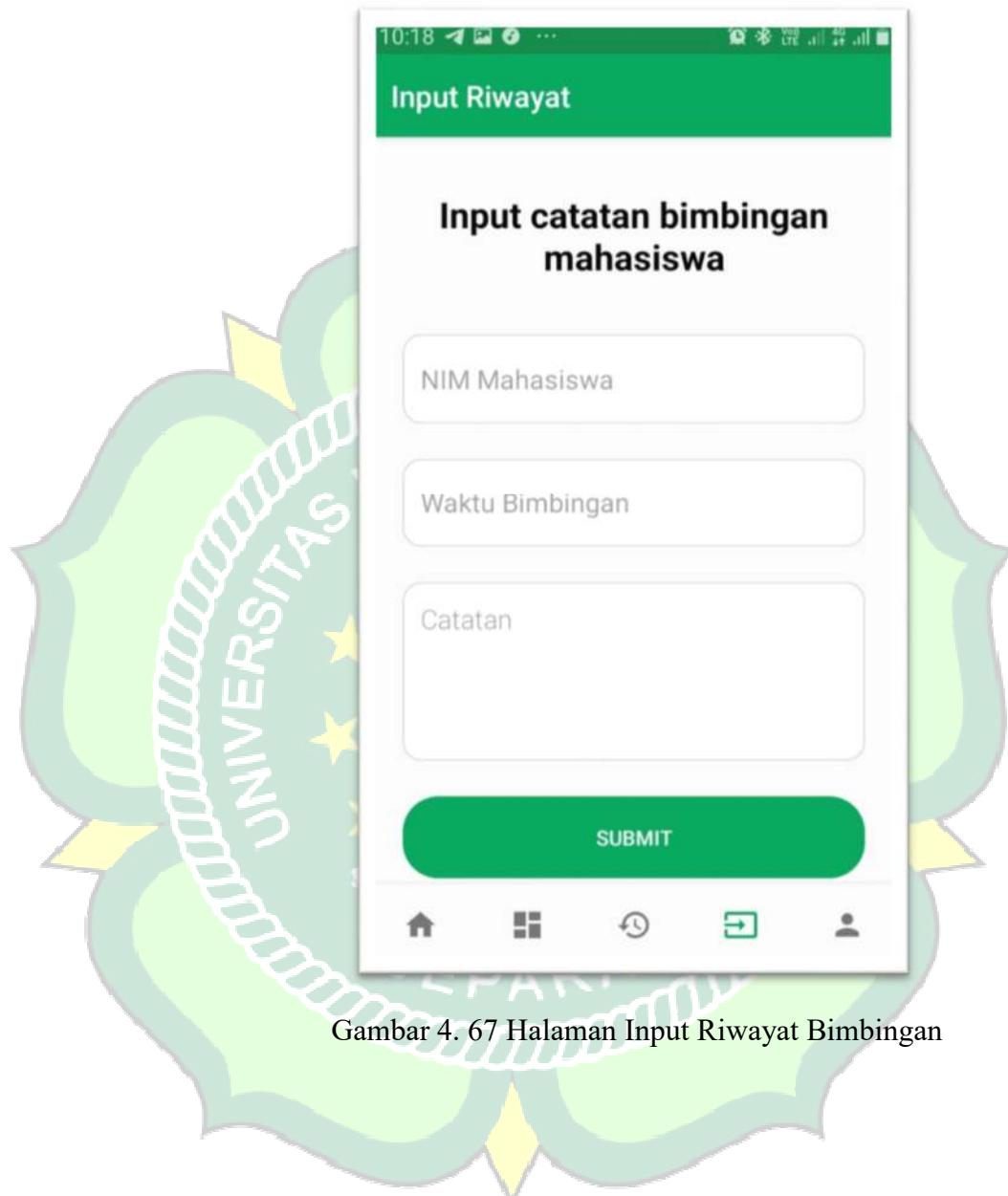


Gambar 4. 65 Halaman Acc/Tolak Ajuan Judul Skripsi
Mahasiswa Oleh Dosen 3



9. Halaman Input Riwayat Bimbingan

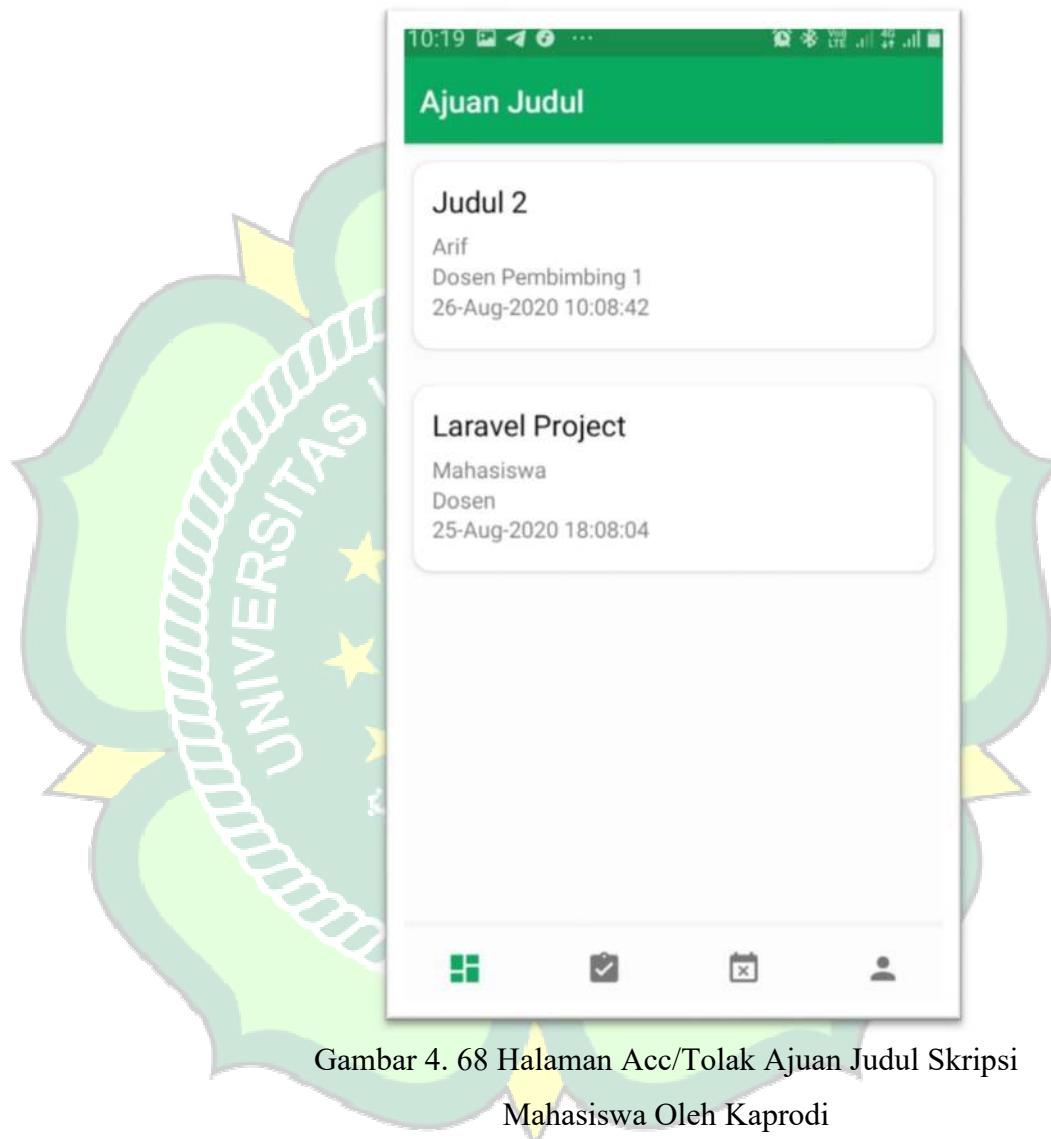
Halaman untuk menginputkan catatan pada saat mahasiswa melakukan bimbingan

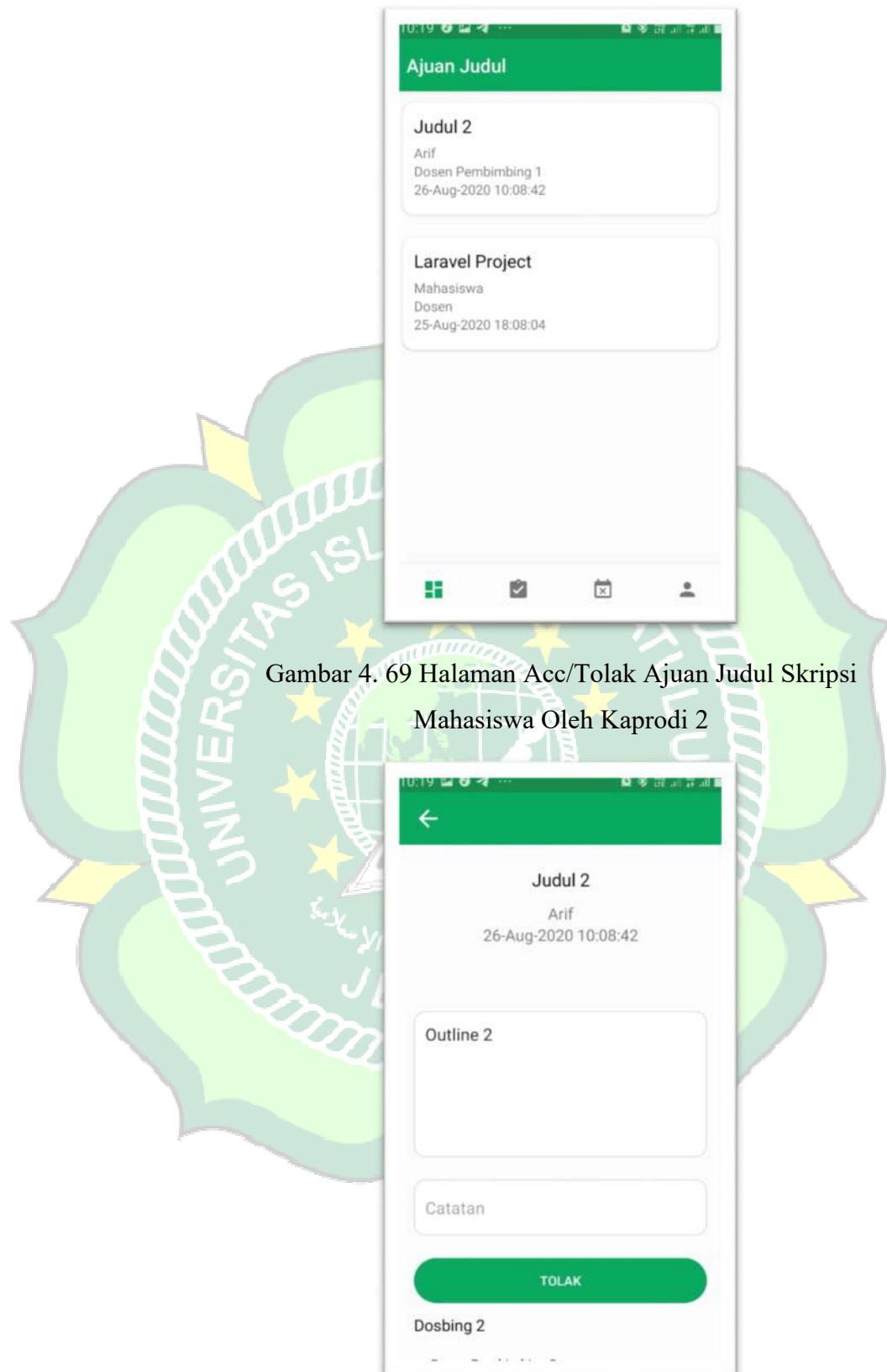


Gambar 4. 67 Halaman Input Riwayat Bimbingan

10. Halaman Acc/Tolak Ajuan Judul Skripsi Mahasiswa Oleh Kaprodi

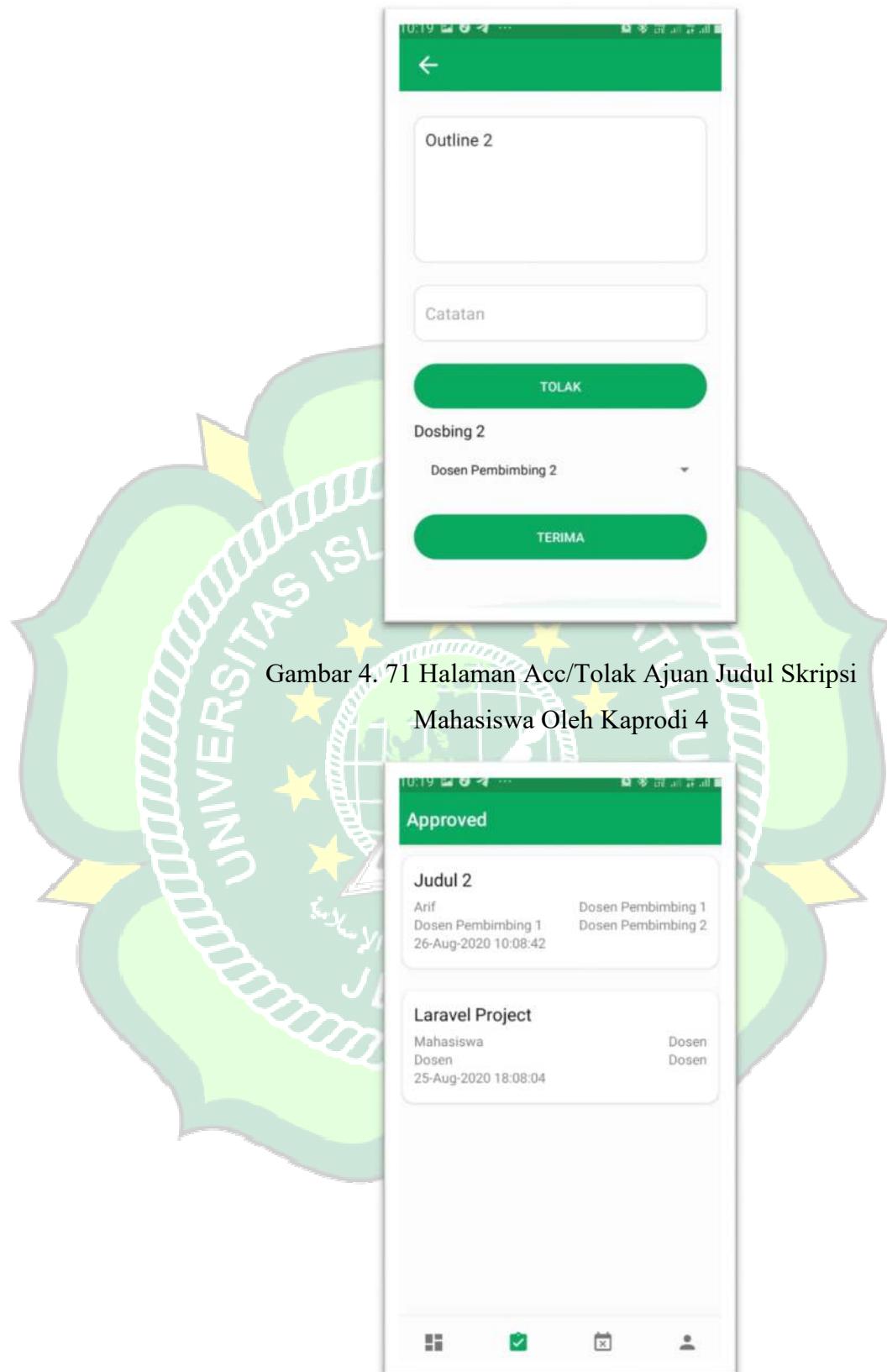
Halaman untuk memberikan status diterima atau ditolak dan menentukan dosen pembimbing 2 jika ajuan judul skripsi mahasiswa diterima oleh kaprodi

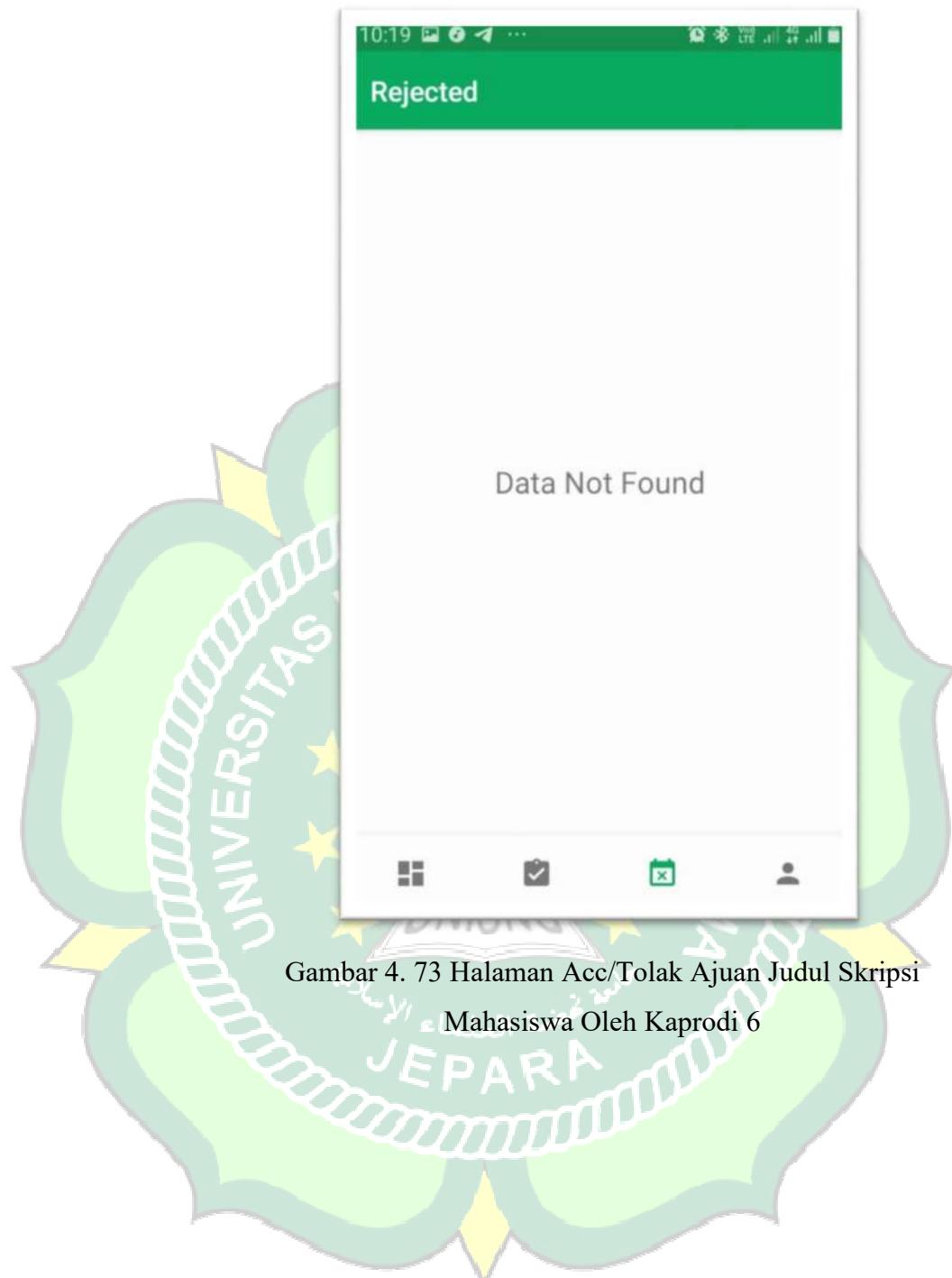




Gambar 4. 69 Halaman Acc/Tolak Ajuan Judul Skripsi
Mahasiswa Oleh Kaprodi 2

Gambar 4. 70 Halaman Acc/Tolak Ajuan Judul Skripsi
Mahasiswa Oleh Kaprodi 3





Gambar 4. 73 Halaman Acc/Tolak Ajuan Judul Skripsi
Mahasiswa Oleh Kaprodi 6

4.4 Pengujian Aplikasi

Sebelum Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Unisnu Jepara ini digunakan, maka perlu untuk dilakukan rangkaian pengujian agar terhindar dari *error* dan kesalahan yang dapat mengganggu admin ataupun pengguna aplikasi. Aplikasi ini diuji untuk mengetahui masukan dan keluaran program aplikasi sudah sesuai dengan apa yang diharapkan atau belum. Aplikasi ini diujikan kepada ahli media dan ahli materi, serta metode kuesioner.

4.4.1 Pengujian Ahli Media

Pengujian kali ini dilakukan dengan menguji fitur – fitur yang ada di dalam Aplikasi Layanan Skripsi apakah sesuai dengan fungsi tombol atau belum. Instrumen pengujian ada pada lampiran 1 yang menunjukkan pernyataan serta hasil dari pengujian telah diuji oleh Validator ahli media, dengan penilaian sebagai berikut :

VTR (Valid Tanpa Revisi)	= 3
VR (Valid Revisi)	= 2
TV (Tidak Valid)	= 1

Bapak Heru Saputro, M.Kom selaku dosen program studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Nahdlatul Ulama' Jepara Yang dipilih peneliti sebagai Validator ahli media untuk melakukan pengujian ahli media pada penelitian kali ini.

Tabel 4. 12 Pengujian Ahli Media

Jumlah Soal Pengujian	8
Jumlah Nilai Maksimal	24
Jumlah Nilai Pengujian (f)	24
Presentase Kelayakan	$(22/24) \times 100 = 91,66\%$
Kriteria	Sangat Layak

4.4.2 Pengujian Ahli Materi

Validasi ini dilakukan oleh ahli materi yaitu Bapak R.H Kusumodestoni, S.Kom., M.Kom sebagai dosen program studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Nahdlatul Ulama Jepara. Validasi kali ini dilakukan untuk mendapatkan saran dan pendapat dari ahli materi agar kebutuhan aplikasi ini bisa sesuai dengan apa yang dibutuhkan. Instrumen penilaian terlampirkan di lampiran 2, berikut adalah hasil dari pengujian yang dilakukan oleh Bapak R.H Kusumodestoni, S.Kom., M.Kom selaku validator ahli materi.

Tabel 4. 13 Pengujian Ahli Materi

Jumlah Soal Pengujian	7
Jumlah Nilai Maksimal	21
Jumlah Nilai Pengujian (f)	20
Presentase Kelayakan	$(20/21) \times 100 = 95,2\%$
Kriteria	Sangat Layak

4.4.3 Pengujian Kuesioner Responden

Di bawah ini merupakan hasil penilaian pengujian yang dilakukan oleh responden dengan memberikan form instrumen penilaian kepada dosen dan mahasiswa.

Tabel 4. 14 Responden Mahasiswa

Jumlah Soal Pengujian	9
Jumlah Nilai Maksimal	675
Jumlah Nilai Pengujian (f)	630
Presentase Kelayakan	$(630/675) \times 100 = 93,33\%$
Kriteria	Sangat Layak

Tabel 4. 15 Responden Dosen

Jumlah Soal Pengujian	8
Jumlah Nilai Maksimal	40
Jumlah Nilai Pengujian (f)	33
Presentase Kelayakan	$(33/40) \times 100 = 82,50\%$
Kriteria	Sangat Layak

Tabel 4. 16 Responden Kaprodi

Jumlah Soal Pengujian	7
Jumlah Nilai Maksimal	35
Jumlah Nilai Pengujian (f)	39
Presentase Kelayakan	$(29/35) \times 100 = 82,86\%$
Kriteria	Sangat Layak

4.5 Evaluasi dan Hasil

Berdasarkan hasil pengujian yang dilakukan oleh Bapak Heru Saputro, M.Kom sebagai validator ahli media, maka Aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Unisnu Jepara ini mendapatkan presentase kelayakan sebesar 91,66% dan termasuk dalam kriteria “Sangat Layak” untuk digunakan. Sedangkan pengujian yang dilakukan oleh Bapak R.H Kusumodestoni, S.Kom., M.Kom sebagai validator ahli materi mendapatkan kelayakan 95,2% yang termasuk dalam kategori “Sangat Layak” untuk digunakan oleh pengguna dalam hal ini adalah mahasiswa, dosen, dan kaprodi.

Berdasarkan penilaian yang dilakukan oleh responden mahasiswa dengan total nilai pengujian 675 dari 9 pertanyaan, mendapatkan nilai 630 dan mendapat presentase 93,33%. Penilaian selanjutnya dilakukan oleh dosen Teknik Infomatika Unisnu Jepara dengan total nilai pengujian 40 dari 8 pertanyaan, mendapatkan nilai 33 dan mendapat presentase 82,50% yang termasuk dalam kategori “Sangat Layak”. Sedangkan penilaian dari responden kaprodi yang juga merupakan salah satu pengguna aplikasi Layanan Skripsi dengan total nilai pengujian 35 dari 7 pertanyaan, mendapatkan nilai 29 dan mendapat presentase 82,86% yang termasuk dalam kategori “Sangat Layak”. Maka, aplikasi Layanan Skripsi Mahasiswa Teknik Informatika Unisnu Jepara masuk dalam kategori “Sangat Layak” untuk digunakan.