

# **RANCANG BANGUN APLIKASI PENGINGAT TUGAS BERBASIS ANDORID**



## **TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Menyelesaikan Studi  
Pada Program Studi D IV Teknik Informatika

**Oleh :**

**Nama : Kukuh Yulian Santi**

**NIM : 15090106**

**POLITEKNIK HARAPAN BERSAMA  
KOTA TEGAL**

**2019**

## **HALAMAN PERNYATAAN**

Yang bertanda tangan dibawah ini :

Nama : Kukuh Yulian Santoso

NIM : 14090067

Adalah mahasiswa Program Studi D IV Teknik Informatika Politeknik Harapan Bersama, dengan ini saya menyatakan bahwa laporan Tugas Akhir yang berjudul : “APLIKASI PENGINGAT TUGAS BERBASIS ANDROID” merupakan hasil pemikiran sendiri secara orisinil dan saya susun secara mandiri dengan tidak melanggar kode etik hak karya cipta. Pada laporan Tugas Akhir ini juga bukan merupakan karya yang pernah diajukan untuk memperoleh gelar akademik tertentu di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila dikemudian hari ternyata Laporan Tugas Akhir ini terbukti melanggar kode etik karya cipta atau merupakan karya yang dikategorikan mengandung unsur plagiarisme, maka saya bersedia untuk melakukan penelitian baru dan menyusun laporannya sebagai laporan Tugas Akhir, sesuai dengan ketentuan yang berlaku.

Demikian pernyataan ini saya buat dengan sebenarnya dan sesungguhnya.

Tegal, September 2019

Yang membuat pernyataan,



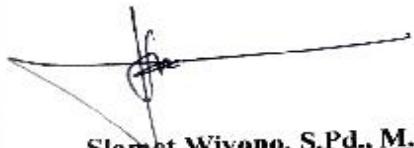
## **HALAMAN REKOMENDASI**

Nama : Kukuh Yulian Santoso  
NIM : 14090067  
Program Studi : D IV Teknik Informatika  
Judul Tugas Akhir : Aplikasi Pengingat Tugas Berbasis Android

Mahasiswa tersebut telah dinyatakan selesai melaksanakan bimbingan dan dapat mengikuti Ujian Tugas Akhir pada tahun akademik 2018/2019.

Tegal, Agustus 2019

Pembimbing I



Slamet Wiyono, S.Pd., M. Eng

NIPY. 08. 015. 222

Pembimbing II



Taufiq Abidin, S.Pd., M. Kom

NJPY. 06. 014. 184

## **HALAMAN PENGESAHAN**

Nama : Kukuh Yulian Santoso

NIM : 14090067

Program Studi : D IV Teknik Informatika

Judul Tugas Akhir : Aplikasi Pengingat Tugas Berbasis Android

**Dinyatakan LULUS / TIDAK-LULUS setelah dipertahankan di hadapan  
Dewan Pengaji Tugas Akhir Program Studi D IV Teknik Informatika  
Politeknik Harapan Bersama**

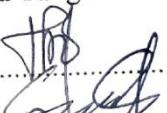
Tegal, September 2019

Dewan Pengaji :

Nama

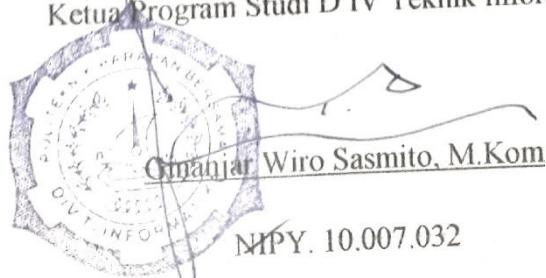
1. Ketua : Dairoh,M.Sc
2. Anggota I : SenaWijayanto,S.Pd.,M.T
3. Anggota II:Taufiq Abidin,S.Pd.,M.Kom

Tanda Tangan

1. 
2. 
3. 

Mengetahui,

Ketua Program Studi D IV Teknik Informatika



## **ABSTRAK**

Penelitian ini dilatar belakangi oleh banyaknya aktivitas mahasiswa yang terkadang membuat mahasiswa terkadang lupa atau terlewat mengenai kegiatan yang harus di kerjakannya pada tepat waktu salah satu kegiatan yang terkadang lupa bahkan terlewat yaitu tugas. Tugas adalah kegiatan yang di lakukan oleh sekumpulan orang dalam Melakukan kegiatan pembelajaran. Tujuan dari penelitian ini yaitu merancang aplikasi pengingat tugas berbasis android yang mempu mengingatkan mahasiswa tentang tugas perkuliahan perkuliahan, mampu mengingat mahasiswa tentang batas akhir pengumpulan tugas serta informasi-informasi lainnya yang mengenai kegiatan perkuliahan. Metode perancangan aplikasi yang di gunakan adalah metode *waterfall*. Penelitian Penelitian ini diuji dengan menggunakan metode whitebox dan blackbox. Hasil pengujian menunjukkan bahwa aplikasi telah benar, tidak memiliki kesalahan dari segi logika dan fungsi, dan secara fungsional dapat menghasilkan output yang diharapkan. Berdasarkan hal tersebut dapat disimpulkan tujuan penelitian ini sudah tercapai. Kata Kunci: Android, Reminder, Pengingat, Jadwal Kuliah, Java, MySQL.

**Kata Kunci:** **Android, Reminder, Pengingat, Tugas, Java, MySQL.**

## KATA PENGANTAR

Dengan memanjatkan puji syukur kehadirat Allah Subhanahu Wa Ta'ala Tuhan Yang Maha Pengasih dan Maha Penyayang yang telah melimpahkan segala rahmat, hidayah dan inayah-Nya hingga terselesaikannya laporan Tugas Akhir dengan judul "Aplikasi Pengingat Tugas Berbasis Android".

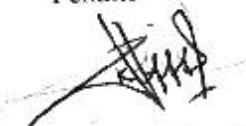
Tugas Akhir merupakan suatu kewajiban yang harus dilaksanakan untuk memenuhi salah satu syarat kelulusan dalam mencapai derajat Sarjana Sains Terapan pada Program Studi D IV Teknik Informatika Politeknik Harapan Bersama. Selama melaksanakan penelitian dan kemudian tersusun dalam laporan Tugas Akhir ini, banyak pihak yang telah memberikan bantuan, dukungan dan bimbingan.

Pada kesempatan ini, tak lupa penulis mengucapkan terima kasih yang sebesar-besarnya kepada :

1. Orang Tua yang telah mendukung, membantu serta mendoakan.
2. Moch. Chambali, B.Eng, M.Kom selaku Direktur Politeknik Harapan Bersama Tegal.
3. Ginanjar Wiro Sasmito, M.Kom selaku Ketua Program Studi D IV Teknik Informatika Politeknik Harapan Bersama Tegal.
4. Slamet Wiyono,S.Pd.,M.Eng selaku dosen pembimbing I.
5. Taufiq Abidin,S.Pd,M.Kom selaku dosen pembimbing II.

Tegal, Agustus 2019

Penulis



Kukuh Yulian Santoso

## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PERNYATAAN .....	ii
HALAMAN REKOMENDASI .....	iii
HALAMAN PENGESAHAN .....	iv
ABSTRAK .....	v
KATA PENGANTAR .....	vi
DAFTAR ISI .....	vii
DAFTAR TABEL .....	ix
DAFTAR GAMBAR .....	xi
DAFTAR LAMPIRAN .....	xv
BAB I. PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah .....	2
1.3. Pembatasan Masalah .....	3
1.4. Tujuan Penelitian .....	3
1.5. Manfaat Penelitian .....	3
1.6. Metodologi Penelitian .....	5
1.7. Sistematika Penulisan .....	11
BAB II. TINJAUAN PUSTAKA .....	13
BAB III. LANDASAN TEORI .....	16
3.1. Aplikasi .....	16
3.2. <i>Android Studio</i> .....	17
3.3. Klasifikasi Aplikasi .....	18
3.4. <i>Android</i> .....	18
3.5. Penjadwalan .....	19
3.6. <i>Kelas</i> .....	19
3.7. Tugas .....	20
3.8. <i>Metode Waterfall</i> .....	20
3.9. UML .....	22
BAB IV. PERANCANGAN DAN DESAIN .....	23
4.1. Perancangan Sistem .....	36
4.2. Perancangan <i>UML</i> .....	37
4.3. Desain <i>Interface</i> .....	60
BAB V. HASIL DAN PEMBAHASAN .....	66
5.1. Hasil Penelitian .....	76
5.1.1. Implementasi Antarmuka .....	74
5.2. Pengujian Black-box Testing .....	96
5.2.1. Pengujian <i>Black-box Testing</i> .....	96
5.3. Pengujian White-box Testing .....	119
BAB VI. PENUTUP .....	131
6.1. Kesimpulan .....	131
6.2. Saran .....	131
DAFTAR PUSTAKA .....	132
LAMPIRAN .....	133

## DAFTAR TABEL

Tabel 1.1	Alat Penelitian .....	6
Tabel 3.1	Komponen <i>Use Case Diagram</i> .....	23
Tabel 3.2	Komponen <i>Sequence Diagram</i> .....	28
Tabel 3.2	Komponen <i>Activity Diagram</i> .....	31
Tabel 3.4	Komponen <i>Class Diagram</i> .....	23
Tabel 4.1	Identifikasi Aktor .....	37
Tabel 4.2	Identifikasi Kebutuhan <i>Use Case</i> .....	40
Tabel 5.1	Pengujian login admin.....	97
Tabel 5.2	Pengujian Halaman Tambah Data Mahasiswa .....	99
Tabel 5.3	Pengujian Halaman Tambah Data Dosen .....	102
Tabel 5.4	Pengujian Halaman Tambah Ruangan .....	105
Tabel 5.5	Pengujian Halaman Login Mahasiswa .....	107
Tabel 5.6	Pengujian Halaman Self Reminder .....	110
Tabel 5.7	Pengujian Halaman Login Dosen .....	112
Tabel 5.8	Pengujian Halaman Tugas Dosen .....	115
Tabel 5.9	Pengujian Halaman Self Reminder Dosen.....	117

## DAFTAR GAMBAR

Gambar 4.1	Use case Diagram .....	38
Gambar 4.2	Activity Diagram Login admin .....	41
Gambar 4.3	Activity Diagram Input Data Mahasiswa .....	42
Gambar 4.4	Activity Diagram Input Data Dosen .....	43
Gambar 4.5	Activity Diagram Input Ruangan .....	44
Gambar 4.6	Activity Diagram Login Mahasiswa .....	45
Gambar 4.7	Activity Diagram Schedule Mahasiswa.....	46
Gambar 4.8	Activity Diagram menu tugas.....	47
Gambar 4.9	Activity Diagram Self Reminder .....	48
Gambar 4.10	Activity Diagram Login Dosen .....	49
Gambar 4.11	Activity Diagram my Schedule Dosen .....	50
Gambar 4.12	Activity Diagram Tugas .....	51
Gambar 4.13	Sequence Diagram login admin .....	53
Gambar 4.14	Sequence Diagram Data Mahasiswa.....	53
Gambar 4.15	Sequence Diagram Data Dosen .....	54
Gambar 4.16	Sequence Diagram Login Mahasiswa .....	55
Gambar 4.17	Sequence Diagram My schedule.....	55
Gambar 4.18	Sequence Diagram Tugas.....	56
Gambar 4.19	Sequence Diagram Self Reminder .....	57
Gambar 4.20	Sequence Data Login Dosen .....	57
Gambar 4.21	Sequence Diagram my Schedule .....	58
Gambar 4.22	Sequence Diagram Tugas.....	58
Gambar 4.23	Sequuence Diagram Jadwal Kuliah .....	59
Gambar 4.23	Desain Halaman Login .....	65
Gambar 4.24	Desain Halaman Beranda.....	66
Gambar 4.25	Desain Halaman Tambah Data Mahasiswa .....	67
Gambar 4.26	Desain Halaman Tambah Data Dosen .....	68
Gambar 4.27	Halaman Tambah Ruangan .....	68
Gambar 4.28	Desain Halaman Login .....	69
Gambar 4.29	Desain Halaman My Schedule .....	70
Gambar 4.30	Desain Halaman Tugas.....	71
Gambar 4.31	Desain Halaman Self Reminder .....	71
Gambar 4.32	Desain Halaman Login .....	72
Gambar 4.33	Desain Halaman My Schedule .....	73
Gambar 4.34	Desain Halaman Self Reminder .....	73
Gambar 4.35	Desain Halaman Order Login .....	74
Gambar 4.36	Desain Halaman My Schedule .....	74
Gambar 4.37	Desain Halaman Tugas.....	75
Gambar 5.1	Tampilan Halaman Login Admin.....	77
Gambar 5.2	Tampilan Halaman Beranda Admin .....	77
Gambar 5.3	Tampilan Halaman Tambah Data Mahasiswa .....	78
Gambar 5.4	Tampilan Halaman Tambah Data Dosen .....	79
Gambar 5.5	Tampilan Halaman Tambah Mata Kuliah .....	80
Gambar 5.6	Tampilan Halaman Tambah Ruangan .....	81
Gambar 5.7	Tampilan Halaman Login.....	82
Gambar 5.8	Tampilan Halaman My Schedule .....	83
Gambar 5.9	Tampilan Halaman Tugas .....	84
Gambar 5.10	Tampilan Halaman Self Reminder .....	85
Gambar 5.11	Tampilan Halaman Login .....	86
Gambar 5.12	Tampilan Halaman Self Reminder .....	87
Gambar 5.13	Tampilan Halaman Tugas .....	88

## **LAMPIRAN**

Lampiran I Surat Kesepakatan Bimbingan Tugas Akhir .....	A-1
Lampiran II Lembar Bimbingan Tugas Akhir .....	B-1
Lampiran III Surat Pengambilan Data .....	C-1

**“RANCANG BANGUN APLIKASI PENGINGAT TUGAS BERBASIS  
ANDROID”**



**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Untuk Menempuh Tugas Akhir Pada Program  
Studi D IV Teknik Informatika

**Oleh :**

**Nama : Kukuh Yulian Santoso**  
**NIM : 14090067**

**POLITEKNIK HARAPAN BERSAMA  
KOTA TEGAL**

**2019**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pendidikan adalah pembelajaran pengetahuan, keterampilan, dan kebiasaan sekelompok orang yang diturunkan dari satu generasi ke generasi berikutnya melalui pengajaran, pelatihan, atau penelitian. Pendidikan sering terjadi di bawah bimbingan orang lain. Di tingkat perguruan tinggi khususnya di Politeknik Harapan Bersama, juga turut menerapkan metode belajar dalam kelas. adapun hal yang berkaitan dengan pendidikan adalah tugas. Tugas berperan penting dalam metode belajar mengajar karna dari tugas sekelompok mahasiswa dapat mengasah ilmu pengetahuan yang telah di milikinya serta tugas berperan penting dalam penilaian kecerdasan mahasiswa itu sendiri.

Dalam kehidupan sehari-hari pada umumnya mahasiswa terkadang kesulitan bahkan ada beberapa orang yang lupa untuk mengatur dan menepati kegiatan hariannya khususnya sering lupa dalam mengingat tugas, untuk itu diperlukan sebuah aplikasi yang dapat membantu agenda mahasiswa sekaligus mengingatkan mahasiswa tersebut dimana pun dan kapan pun mereka berada bersama dengan smartphone mereka.

Salah satunya adalah dengan memanfaatkan kecanggihan smartphone android yang mereka punya karena tak jarang mahasiswa di era modern ini susah lepas dari gadget atau smartphone mereka. Untuk itu diperlukan sebuah aplikasi yang bisa menjadi partner yang dapat membantu memberikan informasi jadwal dan tugas kuliah mereka. Dengan memanfaatkan smartphone yang sudah banyak digunakan oleh mahasiswa, peneliti mencoba mengembangkan aplikasi mobile berbasis android dalam pembuatannya. Dari uraian di atas, penulis mencoba membangun sebuah Rancang Bangun Aplikasi Pengingat Tugas berbasis android yang mana aplikasi ini nantinya akan memberikan manfaat dalam mengingatkan jadwal atau agenda kegiatan mahasiswa. Penulis mengangkat masalah ini dalam penyusunan skripsi yang berjudul “Rancang Bangun Aplikasi Pengingat Tugas Berbasis Andorid”

## 1.2 Perumusan Masalah

Berdasarkan latar belakang masalah yang telah diuraikan diatas dapat diketahui bahwa pokok permasalahan yang dihadapi yaitu “Bagaimana cara membuat aplikasi pengingat tugas dengan menggunakan smartphone yang sering digunakan lebih bermanfaat bagi mahasiswa sekaligus menjadi partner yang efektif, efisien dan optimal.

### **1.3 Batasan Masalah**

Agar ruang lingkup yang dibahas tidak meluas ke hal lain maka perlu adanya batasan masalah, masalah yang akan dibahas dalam hal ini adalah :

1. Pengguna aplikasi ini adalah mahasiswa, admin dan dosen.
2. Aplikasi ini di buat menggunakan android studio versi 3.2
3. Aplikasi ini di gunakan pada *platform android*.

### **1.4 Tujuan Penelitian**

dari penelitian ini Rancang Bangun Aplikasi Pengingat Tugas berbasis Android yang mampu

1. Untuk menyelesaikan masalah yang terjadi tentang penyebaran informasi yang kurang efektif.
2. Untuk meminimalisir masalah dalam faktor daya ingat mahasiswa dengan tugas dalam jangka waktu panjang.
3. Untuk mempermudah dosen dalam menyebarkan tugas kepada mahasiswa.

### **1.5 Manfaat Penelitian**

Adapun manfaat dari pembuatan sistem informasi ini adalah sebagai berikut:

1. Bagi Pengguna
  - a. Untuk memudahkan mahasiswa dalam menerima informasi-informasi tentang kegiatan belajar mengajar.
  - b. Untuk memudahkan mahasiswa dalam menerima dan mengingat tugas.
  - c. Untuk memudahkan dosen dalam menyebarkan tugas

kepada mahasiswa

## 2. Bagi Peneliti

- a. Untuk menambah pengetahuan dan menerapkan ilmu pengetahuan yang telah diperoleh saat perkuliahan dalam membuat sebuah aplikasi.
- b. Untuk menyalurkan ilmu pengetahuan yang dimiliki dan diperoleh terutama dibidang Teknik Informatika

## 3. Bagi Politeknik Harapan Bersama Tegal

- a. Menambah informasi serta referensi bagi mahasiswa D IV Teknik Informatika yang akan membuat dan mengembangkan tema sejenis.
- b. Mengembangkan dan menambah kepustakaan di Politeknik Harapan Bersama

## **1.6 Metodelogi Penelitian**

### **1.6.1 Bahan Penelitian**

Bahan utama penelitian yang akan digunakan ini menggunakan data-data yang berkaitan dengan aplikasi pengingat tugas ini adalah seperti

data mahasiswa ,data dosen, data jadwal perkuliahan

### **1.6.1 Alat Penelitian**

Dalam perancangan sistem informasi pada pembuatan aplikasi pengingat tugas terdapat beberapa perangkat lunak dan perangkat keras yang digunakan untuk mendukung proses perancangan sistem.

## 1. Perangkat Keras

Perangkat keras yang digunakan untuk mengembangkan dan mengumpulkan data pada aplikasi ini adalah sebagai berikut:

- a. *Smartphone*
- b. Laptop

## 2. Perangkat Lunak

Berikut adalah perangkat lunak yang perlukan dalam pembuatan aplikasi penngingat tugas

Tabel 1.1 Alat Penelitian

No	Jenis Perangkat	Fungsi
1.	<i>Windows 7</i>	<i>Operating System</i>
2.	<i>Android Studio</i>	Pembuatan Aplikasi
3.	<i>Star UML</i>	Media dalam membuat perancangan sistem
4.	<i>Balsamiq</i>	<i>Design</i>
5.	<i>FireBase</i>	<i>Database</i>

### 1.6.2 Alur Penelitian

Dalam penelitian ini digunakan metode watetrfall, metode ini dipilih karena model ini melakukan pendekatan secara sistematis dan urut mulai dari level kebutuhan sistem lalu menuju ke tahap analisis, desain sistem, implementasi dan testing / verification. Berikut alur rencana penelitian yang akan dibuat.

## 1. Identifikasi Masalah

Berdasarkan dari analisis sistem yang sedang berjalan dapat ketahui bahwa ketika Informasi tentang tugas serta kegiatan belajar mengajar terkadang tidak diketahui oleh seluruh mahasiswa tersebut beberapa penyebabnya yaitu kurangnya efektifnya sarana informasi yang ada, keterlambatan penyebaran informasi, dan lain sebagainya. Rancang Bangun Aplikasi Pengingat Tugas merupakan aplikasi yang membantu dosen maupun mahasiswa untuk mengingat tugas yang diprogramkan dan juga memberikan informasi dari admin ke mahasiswa tentang informasi seputar kegiatan belajar mengajar . Rancang Bangun Aplikasi Pengingat Tugas menyediakan informasi mengenai data perkuliahan yang di dapatkan dari *server* kemudian disimpan dalam *database* aplikasi dan di tampilkan secara rapih sesuai kelas yang dapat di baca oleh mahasiswa dengan mudah. Data tersebut diperoleh dari *database pada prodi teknik informatika*.

## 2. Analisis Kebutuhan

Analisis kebutuhan ini diperlukan pada Rancang Bangun Aplikasi Pengingat Tugas yang akan dibuat, dengan tujuan agar sistem dapat mencapai tujuan yang diharapkan. Analisis sistem yang sudah diharapkan mampu berjalan dengan keinginan sebagai berikut

- a. Aplikasi dapat di gunakan untuk mahasiswa, admin dan dosen.
- b. Aplikasi dapat menampilkan informasi tentang jadwal kuliah, tugas dan notifikasi lainnya.

## 3. Metode Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan dengan metode wawancara.

- a. Wawancara

Wawancara dilakukan langsung kepada mahasiswa yang dijadikan *sample* penelitian untuk mendapatkan data secara *real* dilapangan tentang bagaimana mahasiswa menerima informasi tentang jadwal kuliah, tugas, dan notifikasi lainnya.

#### 4. Desain Sistem

Pada tahap ini dilakukan pemodelan sistem dengan UML. Data yang sudah terkumpul pada tahap level kebutuhan sistem dan analisis kebutuhan dikumpulkan dan direkap yang nantinya akan dijadikan satu dalam sistem yang akan dibuat. Pada tahap desain sistem ini dilakukan penggambaran kebutuhan - kebutuhan dan fungsional informasi pada Rancang Bangun Aplikasi Pengingat Tugas.

#### 5. Implementasi

Pada tahapan ini dilakukan pembuatan sistem per bagian yang telah ditentukan, seperti menu aplikasi dan profile pengguna. Tools yang digunakan pada tahapan ini adalah android studio.

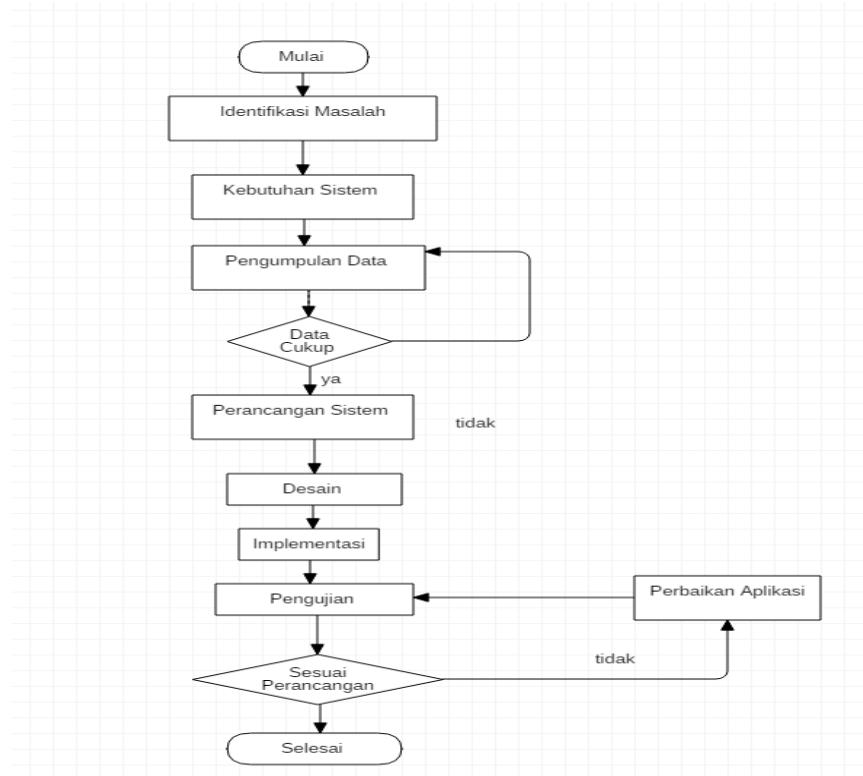
Pada tahapan ini juga dilakukan penyesuaian dengan modul yang sudah dibuat, apakah sudah memenuhi kebutuhan atau belum.

#### 6. Pengujian Sistem

Pada tahap ini dilakukan pengujian pada perangkat lunak yang telah dibuat, pengevaluasian hasil yang diperoleh serta perbaikan program (revisi), jika hasil belum sesuai dengan tujuan yang diharapkan. Apabila tahap ini selesai maka sistem sudah dapat diaplikasikan.

#### 1.6.4 Flowchart Alur Penelitian

Flowchart merupakan penggambaran urutan atau alur dari penelitian yang di lakukan, flowchart alur penelitian di gambarkan sebagai berikut



## **1.7 Sistematis Penulisan**

### **5.7.1.1. BAB 1 : PENDAHULUAN**

Dalam bab pendahuluan berisi tentang latar belakang diambil judul “Rancang Bangun Aplikasi Pengingat Tugas”. Perumusan masalah, batasan masalah, tujuan, manfaat, metodelogi dan sistematika penulisan yang menjelaskan secara garis besar substansi yang di berikan pada masing- masing bab.

### **5.7.1.2. BAB II : TINJAUAN PUSTAKA**

Bab ini menguraikan dari latar belakang yang berkaitan dengan penelitian-penelitian serupa yang pernah di lakukan yang di bahas secara kritis. berisi tentang penjelasan inti sari latar belakang, tujuan, metode, dan hasil penelitian

### **5.7.1.3. BAB III : LANDASAN TEORI**

Dalam bab ini tentang landasan teori yang akan digunakan untuk penyelesaian laporan penelitian yang berkaitan dengan Rancang Bangun Aplikasi Pengingat Tugas.

#### **5.7.1.4. BAB IV : PERANCANGAN DAN DESAIN**

Bab ini menguraikan secara detail rancangan dan desain terhadap penelitian yang di lakukan dari sistem yang dibangun mengenai data-data yang dibuat dalam suatu perancangan *UML*, rancangan ini terdiri dari : *Use Case Diagram, Activity Diagram, Sequance Diagram dan Class Diagram.*

#### **5.7.1.5 BAB V : HASIL DAN PEMBAHASAN PENELITIAN**

Dalam bab ini berisi tentang *Screnshoot* hasil penelitian dan hasil penelitian tersebut di bahas secara detail berdasarkan penelitian yang selesai di lakukan.

## **BAB II**

### **TINJAUAN PUSTAKA**

Findra. dkk. (2016), telah melakukan penelitian tentang *Rancang Bangun Aplikasi Pengingat Kegiatan Akademik Berbasis Android*. Latar belakang pada penelitian ini Kegiatan akademik seperti kegiatan belajar mengajar, pelaksanaan ujian semester, ujian kerja praktek dan ujian tugas akhir (pendadaran) merupakan kegiatan operasional sehari-hari yang terjadi di sebuah lembaga pendidikan, khususnya di perguruan tinggi. Kegiatan belajar mengajar merupakan hal yang sudah menjadi rutinitas, namun masih seringkali terjadi, mahasiswa maupun dosen lupa dengan jadwal kegiatan akademik yang menjadi kewajibannya, entah karena adanya perubahan jadwal, ataupun alasan lainnya. Perubahan jadwal reguler biasanya didistribusikan oleh pihak Tata Usaha ke mahasiswa atau dosen secara manual melalui papan pengumuman. Oleh karena itu, dibutuhkan suatu aplikasi untuk membantu pendistribusian perubahan informasi akademik secara realtime dan mengingatkan dosen maupun mahasiswa untuk melaksanakan kegiatan akademik sesuai jadwal. Target yang ingin dicapai adalah menyediakan sebuah aplikasi pengingat jadwal kegiatan akademik. Aplikasi ini terdiri atas dua bagian, yaitu aplikasi web untuk memasukkan perubahan jadwal, dan aplikasi mobile yang digunakan untuk menampilkan jadwal kegiatan akademik harian berserta notifikasi pengingat jadwal tersebut [1].

Alisa dkk. (2017), telah melakukan penelitian tentang *Rancang Bangun Aplikasi Penjadwalan Tugas Berbasis Mobile Device di dukung Google Task dan Google Drive*. Latar belakang pada penelitian ini, yakni dalam suatu pengerjaan tugas tentunya diperlukan wadah untuk menyimpan file-file penting di tempat yang aman. Beberapa tahun belakangan ini, *Cloud Computing* semakin marak penggunaannya karena data yang kita punya tersimpan dengan aman. Pada aplikasi-aplikasi tersebut diatas, masih belum ada wadah untuk menampung data-data pendukung dari suatu tugas. Sehingga dalam proses pengerjaan tugas membutuhkan suatu wadah penyimpanan yang terpisah dari aplikasi *reminder* seperti *flashdisk*, *hardisk*, maupun menggunakan *cloud computing*. Aplikasi Penjadwalan Tugas berbasis *Mobile Device* didukung *Google Tasks* dan *Google Drive* ini dibuat untuk menggabungkan fitur *reminder* dan fitur penyimpanan data-data penunjang suatu tugas. Program ini menggunakan *Google Tasks* sebagai wadah penjadwalan tugas dan juga menggunakan *Google Drive* sebagai wadah penyimpanan data-data penunjang. Sehingga cukup dengan satu aplikasi mencakup kebutuhan pengguna dalam pelaksanaan pembuatan tugas yang tidak didapati pada aplikasi lain [2].

Wahyu Joni Kurniawan , dkk. (2017), telah melakukan penelitian tentang *Pembangunan Aplikasi Manajemen Tugas Mahasiswa Berbasis Mobilr Android*. Latar belakang pada penellitian ini, yakni Tugas mahasiswa merupakan salah satu penilaian hasil belajar mahasiswa, adanya mahasiswa yang tidak mangumpulkan tugas dikarenakan lupa ataupun tidak mengetahui adanya tugas dari dosen sangat merugikan mahasiswa. Tugas yang dikumpul dengan jenis *softcopy* terkadang

menyusahkan mahasiswa dalam mengumpulkan dan juga menyulitkan dosen dalam mengupulkan semua tugas mahasiswa. Dengan memanfaatkan teknologi dalam penunjang perkuliahan, aplikasi mobile yang dibangun sangat membantu mahasiswa maupun dosen dalam perkuliahan. Aplikasi yang dibangun terdiri dari dua sisi, pertama aplikasi web yang nantinya akan digunakan dosen yang bersangkutan dalam memanajemen tugas mahasiswa. Kedua, aplikasi berbasis mobile yang akan dipasangkan ke perangkat mobile mahasiswa yang akan berinteraksi dengan dosen [3].

Amuharnis (2019), telah melakukan penelitian tentang *Aplikasi Mobile Student Assistant Prodi Sistem Informasi Berbasis Andorid* Latar belakang pada penelitian ini yaitu Mobile learning (m-learning) merupakan model pembelajaran yang memanfaatkan teknologi informasi dan komunikasi, membawa manfaat ketersediaan materi ajar yang dapat diakses setiap saat berupa konten teks, gambar, maupun video. Hampir seluruh materi yang disajikan untuk perkuliahan di jurusan sistem informasi pada STMIK Indonesia berupa konten m-learning. Hal ini menyebabkan dibutuhkannya sebuah aplikasi yang dapat memudahkan untuk berbagi *m-learning* baik itu materi ajar yang diberikan dosen maupun tugas yang telah diselesaikan mahasiswa. *Aplikasi Mobile Student Assistant* ini dikembangkan berbasis Android dengan bahasa pemrograman *Java mobile* dan *Firebase* sebagai basis data. Metode pengembangan sistem untuk peneitian ini adalah *System Development Life Cycle* (SDLC) dengan pendekatan Waterfall. Hasil akhir dari penelitian ini adalah

terciptanya *Aplikasi Mobile Student Assistant* berbasis Android yang dapat digunakan untuk berbagi m-learning, mengupload, dan download tugas perkuliahan oleh mahasiswa program studi Sistem Informasi pada STMIK Indonesia Padang [4].

Muslihudin. (2016), telah melakukan peneltian tentang *Perancangan Sistem Pengingat Aktifitas Akademik*. Latar belakang pada penelitian ini yaitu Aktifitas akademik dosen antara lain mengajar sesuai jadwal sebanyak 14 pertemuan di kelas per semester, dan jadwal ujian baik tengah semester maupun akhir semester. Pada ujian yang harus dilakukan yaitu pengumpulan soal dua pekan sebelum ujian dilakukan dan pengumpulan nilai sepekan setelah ujian. Aktifitas yang lain bisa merupakan tugas sebagai pimpinan di kampus, pelaksanaan penelitian dan pengabdian, tugas di masyarakat. Kesibukan yang dialami dosen menyebabkan beberapa tugas yang sudah terjadwal dan harus dilaksanakan berdampak pada lupa atau mengalami keteledoran yang tidak disengaja. Kejadian yang sering menjadi penyebab kegagalan dalam melaksanakan tugas tepat waktu karena dosen sangat fokus pada suatu pekerjaan yang sedang dikerjakan. Fokus dalam satu aktifitas menyebabkan aktifitas yang lain terlupakan [5].

## **BAB III**

### **LANDASAN TEORI**

#### **3.1 Aplikasi**

Aplikasi adalah penggunaan dalam suatu komputer,instruksi (*instruction*) atau pernyataan (*statement*) yang disusun sedemikian rupa sehingga komputer dapat memproses input menjadi *output* . Menurut Kamus Kamus Besar Bahasa Indonesia, Aplikasi adalah penerapan dari rancang sistem untuk mengolah data yang menggunakan aturan atau ketentuan bahasa pemrograman tertentu. Aplikasi adalah suatu program komputer yang dibuat untuk mengerjakan dan melaksanakan tugas khusus dari pengguna. Aplikasi merupakan rangkaian kegiatan atau perintah untuk dieksekusi oleh komputer. Program merupakan kumpulan *instruction set* yang akan dijalankan oleh pemroses, yaitu berupa *software*. Bagaimana sebuah sistem komputer berpikir diatur oleh program ini.

Program inilah yang mengendalikan semua aktifitas yang ada pada pemroses. Program berisi konstruksi logika yang dibuat oleh manusia, dan sudah diterjemahkan ke dalam bahasa mesin sesuai dengan format yang ada pada *instructionset*. Program aplikasi merupakan program siap pakai yang dirancang untuk melaksanakan suatu fungsi bagi pengguna atau aplikasi yang lain. Contoh-contoh aplikasi ialah program pemroses kata dan *Web Browser*. Aplikasi akan menggunakan *system* operasi (OS) komputer dan aplikasi yang lainnya yang mendukung. Istilah ini mulai perlahan masuk ke

dalam istilah Teknologi Informasi semenjak tahun 1993, yang biasanya juga disingkat dengan app.

Secara historis, aplikasi adalah software yang dikembangkan oleh sebuah perusahaan. App adalah software yang dibeli perusahaan dari tempat pembuatnya. Industri PC tampaknya menciptakan istilah ini untuk merefleksikan medan pertempuran persaingan yang baru, yang parallel dengan yang terjadi antar sistem operasi yang dimunculkan[6].

### 3.2 Android Studio

(Jadibaru, Pengenalan Android Studio, 2015) Android Studio adalah sebuah IDE untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer, yaitu IntelliJ IDEA. Android Studio merupakan IDE resmi untuk pengembangan aplikasi Android. Sebagai pengembangan dari Eclipse, Android Studio mempunyai banyak fitur-fitur baru dibandingkan dengan Eclipse IDE. Berbeda dengan Eclipse yang menggunakan Ant, Android Studio menggunakan Gradle sebagai build environment. Fitur-fitur lainnya adalah sebagai berikut :

- Menggunakan Gradle-based build system yang fleksibel.
- Bisa mem-build multiple APK .

- Template support untuk Google Services dan berbagai macam tipe perangkat.
- Layout editor yang lebih bagus.
- Built-in support untuk Google Cloud Platform, sehingga mudah untuk integrasi dengan Google Cloud Messaging dan App Engine.
- Import library langsung dari Mavenrepository

### **3.3 Klasifikasi Aplikasi**

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain:

1. Perangkat lunak perusahaan (enterprise)
2. Perangkat lunak infrastruktur perusahaan
3. Perangkat lunak informasi kerja
4. Perangkat lunak media dan hiburan
5. Perangkat lunak pendidikan
6. Perangkat lunak pengembangan media
7. Perangkat lunak rekayasa produk

### **3.4 Android**

Android adalah sistem operasi berbasis Linux yang dirancang untuk perangkat bergerak layar sentuh seperti telepon pintar dan komputer tablet. Android awalnya dikembangkan oleh Android, Inc., dengan dukungan finansial dari Google, yang kemudian membelinya pada tahun 2005. Sistem operasi ini dirilis secara resmi pada tahun 2007, bersamaan dengan didirikannya Open Handset Alliance, konsorsium dari perusahaan-perusahaan perangkat keras, perangkat lunak, dan telekomunikasi yang bertujuan untuk memajukan standar terbuka perangkat seluler. Ponsel Android pertama mulai dijual pada bulan Oktober 2008 (“Android (Sistem Operasi)”, 2016). Antarmuka pengguna Android umumnya berupa manipulasi langsung, menggunakan gerakan sentuh yang serupa dengan tindakan nyata,

Android adalah sistem operasi dengan sumber terbuka, dan Google merilis kodennya di bawah Lisensi Apache. Kode dengan sumber terbuka dan

lisensi perizinan pada Android memungkinkan perangkat lunak untuk dimodifikasi secara bebas dan didistribusikan oleh para pembuat perangkat, operator nirkabel, dan pengembang aplikasi. Selain itu, Android memiliki sejumlah besar komunitas pengembang aplikasi (*apps*) yang memperluas fungsionalitas perangkat, umumnya ditulis dalam versi kustomisasi bahasa pemrograman Java. Pada bulan Oktober 2013, ada lebih dari satu juta aplikasi yang tersedia untuk Android, dan sekitar 50 miliar aplikasi telah diunduh dari Google Play, toko aplikasi utama Android. Sebuah survei pada bulan April-Mei 2013 menemukan bahwa Android adalah platform paling populer bagi para pengembang, digunakan oleh 71% pengembang aplikasi bergerak. Di Google I/O 2014, Google melaporkan terdapat lebih dari satu miliar pengguna aktif bulanan Android, meningkat dari 583 juta pada bulan Juni 2013 [7].

### **3.5 Penjadwalan**

Penjadwalan Pengertian jadwal menurut kamus besar bahasa Indonesia adalah pembagian waktu berdasarkan rencana pengaturan urutan kerja, daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Sedangkan pengertian penjadwalan adalah proses, cara, perbuatan menjadwalkan atau memasukkan ke dalam jadwal. menyatakan bahwa jadwal didefinisikan sebagai sesuatu yang menjelaskan di mana dan kapan orang-orang dan sumber daya berada pada suatu waktu. Berdasarkan Kamus Besar Bahasa Indonesia, jadwal merupakan pembagian waktu berdasarkan rencana pengaturan urutan kerja.

Jadwal juga didefinisikan sebagai daftar atau tabel kegiatan atau rencana kegiatan dengan pembagian waktu pelaksanaan yang terperinci. Kebanyakan orang terbiasa dengan jadwal pelajaran yang disajikan sebagai tabel hari dalam seminggu dan jangka waktu. Dapat dilihat bahwa setiap hari dibagi ke dalam jangka waktu. Setiap jangka waktu memiliki daftar mata kuliah yang sedang diajarkan, oleh siapa dan di mana. Jadwal dapat dinyatakan dalam sejumlah cara yang berbeda, masing-masing mahasiswa harus memiliki jadwal sendiri tergantung pada mata pelajaran, begitu juga masing-masing guru dan STIKOM ruang, semua ini adalah perspektif yang berbeda pada jadwal yang sama[8].

### 3.6 Kelas

Kelas dapat diartikan sebagai sekelompok murid yang menghadapi pelajaran ataupun kuliah tertentu di perguruan tinggi, sekolah, maupun lembaga pendidikan. Kelas juga juga dapat diartikan sebagai kegiatan belajar - mengajar itu sendiri. Kelas dapat pula berupa sekelompok murid di tingkatan yang sama dalam sebuah institusi. Kelas dapat pula merujuk ke ruangan, bangunan, atau wahana di mana pelajaran diajarkan bahwa kelas adalah sebagai suatu masyarakat kecil yang merupakan bagian dari masyarakat sekolah, yang sebagai satu kesatuan diorganisasi menjadi unit kerja yang secara dinamis menyelenggarakan berbagai kegiatan pembelajaran yang kreatif untuk mencapai suatu tujuan.

Selain itu Nawawi juga menegaskan bahwa definisi kelas dibagi dua yaitu:

Kelas dalam arti luas adalah suatu masyarakat kecil yang merupakan bagian dari masyarakat sekolah yang sebagai kesatuan diorganisir menjadi unit kerja secara dinamis menyelenggarakan berbagai kegiatan belajar-mengajar yang kreatif untuk mencapai suatu tujuan.

kelas adalah suatu kelompok orang yang melakukan kegiatan belajar bersama yang mendapatkan pengajaran dari guru. Pengertian ini jelas ditinjau dari segi anak didik karena dalam pengertian tersebut ada frase kelompok orang. Sedangkan menurut Suharsini Arikunto, kelas adalah sekelompok siswa yang pada waktu yang sama menerima pelajaran yang sama[9].

### 3.7 Metode *Waterfall*

model *waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*. Nama model ini sebenarnya adalah “*Linear Sequential Model*”. Model ini sering disebut juga dengan “*classic life cycle*” atau metode waterfall. Model ini termasuk ke dalam model *generic* pada rekayasa perangkat lunak dan pertama kali diperkenalkan oleh Winston Royce sekitar tahun 1970 sehingga sering dianggap kuno, tetapi merupakan model yang paling banyak dipakai dalam *Software Engineering* (SE). Model ini melakukan pendekatan secara sistematis dan berurutan. Disebut dengan *waterfall* karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

Fase-fase dalam *Waterfall Model* menurut referensi Pressman :

a. *Communication (Project Initiation & Requirements Gathering)*

Sebelum memulai pekerjaan yang bersifat teknis, sangat diperlukan adanya komunikasi dengan *customer* demi memahami dan mencapai tujuan yang ingin dicapai. Hasil dari komunikasi tersebut adalah inisialisasi proyek, seperti menganalisis permasalahan yang dihadapi dan mengumpulkan data-data yang diperlukan, serta membantu mendefinisikan fitur dan fungsi *software*. Pengumpulan data-data tambahan bisa juga diambil dari jurnal, artikel, dan internet.

b. *Planning (Estimating, Scheduling, Tracking)*

Tahap berikutnya adalah tahapan perencanaan yang menjelaskan tentang estimasi tugas-tugas teknis yang akan dilakukan, resiko-resiko yang dapat terjadi, sumber daya yang diperlukan dalam membuat sistem, produk kerja yang ingin dihasilkan, penjadwalan kerja yang akan dilaksanakan, dan *tracking* proses penggeraan sistem.

c. *Modeling (Analysis & Design)*

Tahapan ini adalah tahap perancangan dan permodelan arsitektur sistem yang berfokus pada perancangan struktur data, arsitektu *software*, tampilan *interface*, dan algoritma program. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.

d. *Construction (Code & Test)*

Tahapan *Construction* ini merupakan proses penerjemahan bentuk desain menjadi kode atau bentuk/bahasa yang dapat dibaca oleh mesin.

## **BAB IV**

### **PERANCANGAN DAN DESAIN**

#### **4.1. Perancangan Sistem**

“Rancang Bangun Aplikasi Pengingat Tugas ” ini dibuat menggunakan Android versi 3.0.5 yang dikombinasikan dengan *Firebase* sebagai database server.

Perancangan aplikasi ini sebagai unsur atau tahapan penting dari keseluruhan pengembangan aplikasi. Pada tahap ini, bagian bagian yang ada dalam aplikasi tersebut adalah sebagai berikut :

- a. Menu *Login*
- b. Menu *My Schedule*
- c. Menu Tugas
- d. Menu *Self Reminder*

Dengan menggunakan program tersebut maka diharapkan dapat mempermudah mahasiswa dalam mengingat kegiatan belajar mengajar.

Sistem ini dilakukan dengan cara membuat *usecase diagram*, *activity diagram*, *sequence diagram*.

## 4.2. Perancangan UML (*Unified Modeling Language*)

### 4.2.1. Usecase Diagram.

Mendeskripsikan interaksi antara aktor, dan hubungan fungsional dari perancangan aplikasi. Berikut merupakan *usecase diagram* menggambarkan sebuah interaksi yang terjadi antara aktor dengan aplikasi.

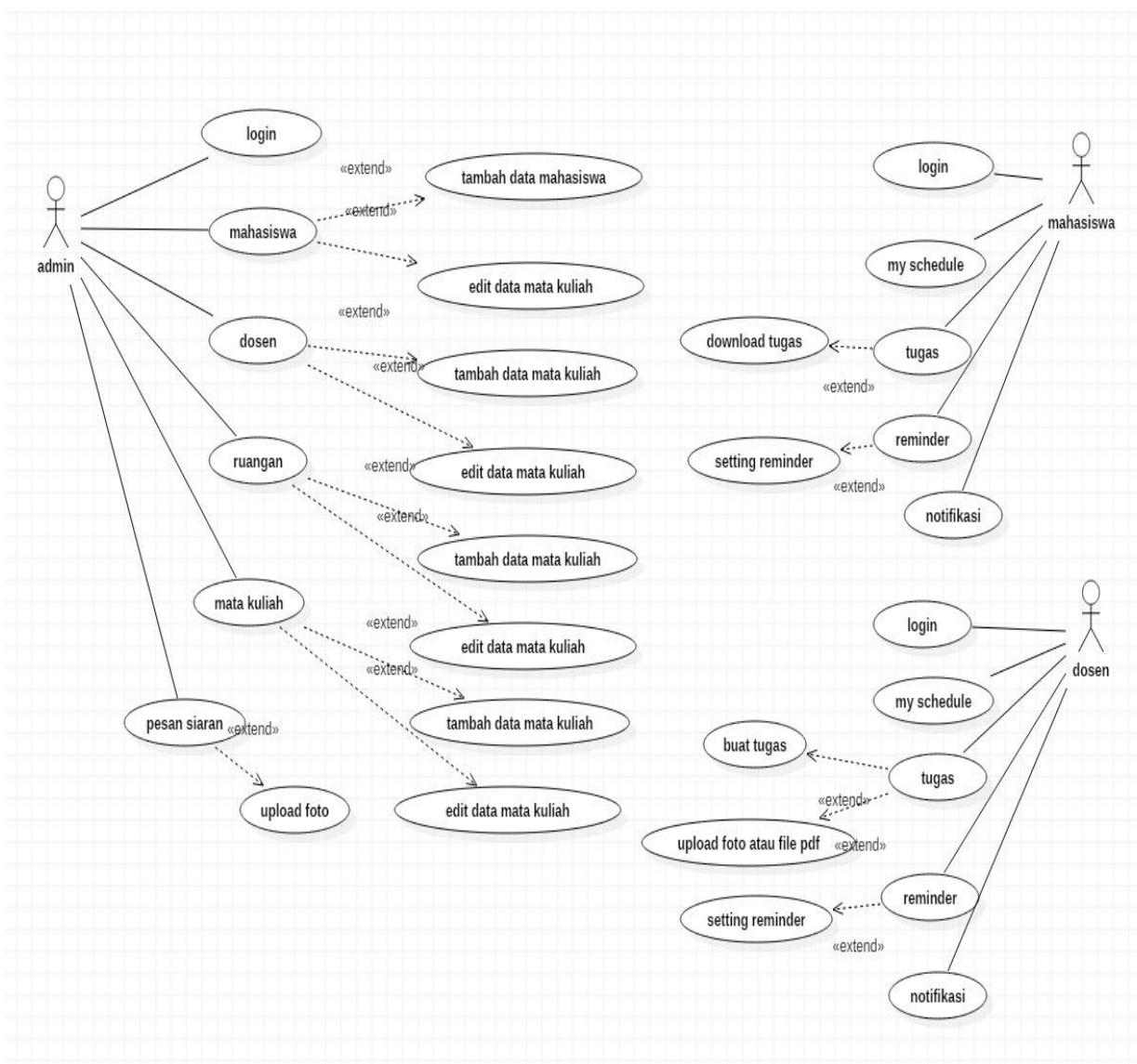
#### 1. Identifikasi Aktor.

Identifikasi aktor mendeskripsikan interaksi antar aktor dengan system yang ada pada rancang bangun aplikasi pengingat tugas. Tabel identifikasi aktor di tunjukan pada tabel 4.1 di bawah ini

Tabel 4.1 Identifikasi Diagram *Use Case*.

No.	Nama <i>Usecase</i>	Deskripsi	Aktor
1.	<i>Login</i>	Menggambarkan aktivitas memasukkan <i>Username</i> dan <i>password</i> untuk mengakses aplikasi sebagai <i>user</i> .	Mahasiswa, Dosen dan Admin
2.	My Schedule	Menggambarkan jadwal kuliah mahasiswa sesuai kelas	Mahasiswa dan Dosen

3.	Menu Tugas	Menggambarkan sebuah dimana mahasiswa dapat menerima informasi tugas yang di berikan dosen dalam waktu jangka panjang dan jangka pendek, serta mahasiswa dapat mengatur waktu pengingat tugas itu sendiri	Mahasiswa dan Dosen
4	Menu Self Reminder	Menggambarkan kegiatan dimana user dapat mengatur reminder	Mahasiswa dan Dosen



Gambar 4.1. *Use Case Diagram*.

### 1. Identifikasi Aktor

Identifikasi actor mendeskripsikan interaksi antar actor dengan sistem yang ada di rancang bangun aplikasi pengingat tugas. Tabel identifikasi actor di tunjukan pada table 4.1 di bawah ini :

Tabel 4.1 Identifikasi Aktor

No	Aktor	Deskripsi
1	Admin	Admin memiliki hak akses untuk melakukan tambah data mahasiswa, tambah data dosen, tambah data ruangan dan tambah data matakuliah serta mengatur jadwal kegiatan perkuliahan.
2	Dosen	Dosen memiliki hak akses untuk melihat jadwal kegiatan mengajar, memberikan tugas serta membuat remainder tugas.
3	Mahasiswa	Mahasiswa memiliki hak akses untuk melihat jadwal kegiatan perkuliahan, menerima tugas, menerima notifikasi serta mengaktifkan reminder tugas.

## 2. Identifikasi kebutuhan *Use Case*

Identifikasi kebutuhan *Use Case* menggambarkan interaksi antar sistem yang terdapat di rancang bangun aplikasi pengingat tugas. Penjelasan dari interaksi antar *Use Case* di jelaskan pada table 4.2 di bawah ini :

Tabel 4.2 Identifikasi Kebutuhan *Use Case*

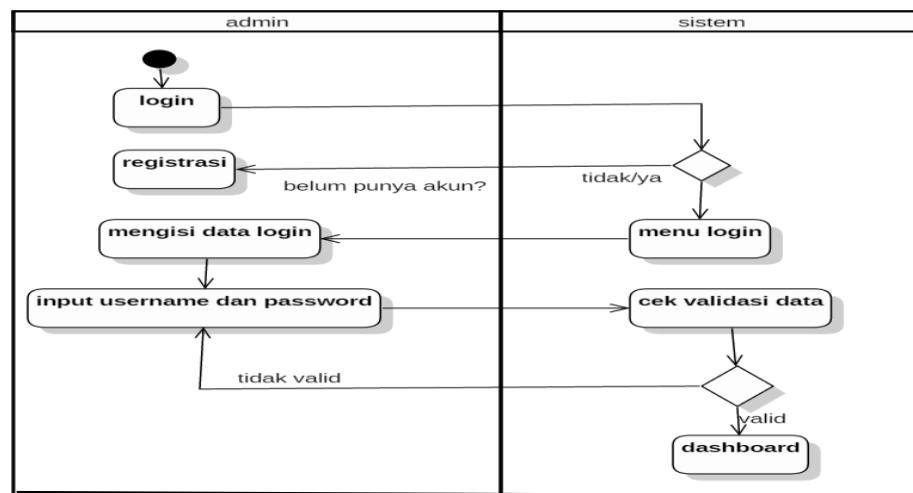
No	Nama <i>Use Case</i>	Deskripsi	Aktor
1	Login	<i>Use case login</i> menggambarkan kegiatan memasukan <i>username</i> dan <i>password</i> untuk mengakses sistem	Admin, dosen dan mahasiswa
2	<i>My Schedule</i>	Bertujuan untuk menampilkan kegiatan jadwal perkuliahan.	Dosen dan Mahasiswa
3	Tugas	Bertujuan untuk menampilkan tugas yang telah di buat oleh dosen dan di terima oleh mahasiswa.	Dosen dan Mahasiswa
4	Menu <i>Self Reminder</i>	Bertujuan untuk menampilkan reminder yang telah di atur sebelumnya untuk mengingat tugas yang telah di buat oleh dosen dan diterima oleh mahasiswa	Dosen dan Mahasiswa

#### 4.2.2. Activity Diagram.

*Activity diagram* adalah teknik untuk mendeskripsikan alur program dalam banyak kasus. *Activity diagram* mendukung aktivitas paralel. Berikut ini adalah *activity diagram* untuk masing-masing *use case*.

##### 4.2.2.1 Activity Diagram Login admin

*Activity diagram login* merupakan gambaran dari sebuah aktifitas login atau alur kerja login yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *login admin* dijelaskan pada Gambar 4.2 di bawah ini :

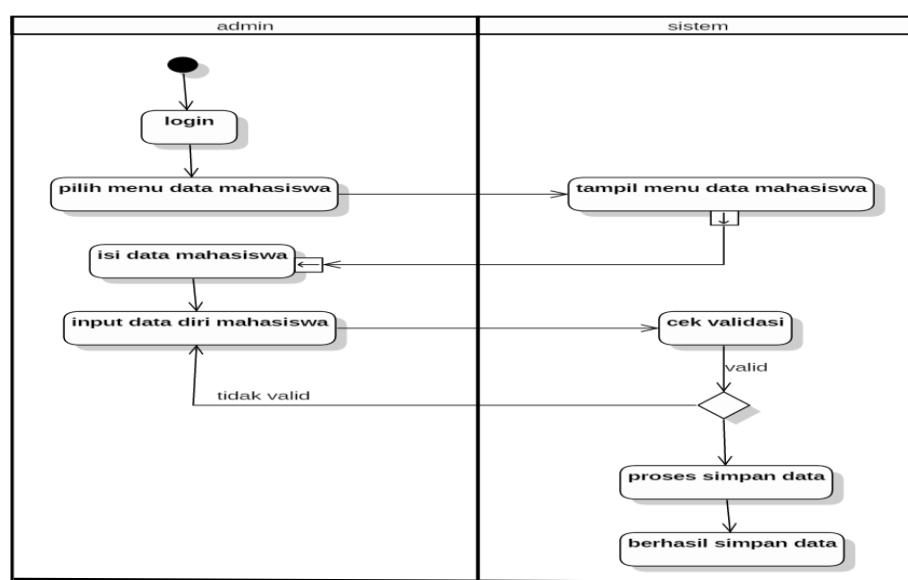


Gambar 4.2. Activity Diagram *Login admin*.

Pada gambar 4.2 menjelaskan *activity diagram login* admin dengan cara memasukan *e-mail* dan *password*.

#### 4.2.2.2 Activity Diagram input data mahasiswa.

*Activity diagram input mahasiswa* merupakan gambaran dari sebuah aktifitas input data mahasiswa yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. . Alur aktifitas *input data mahasiswa* di jelaskan pada Gambar 4.3 di bawah ini :

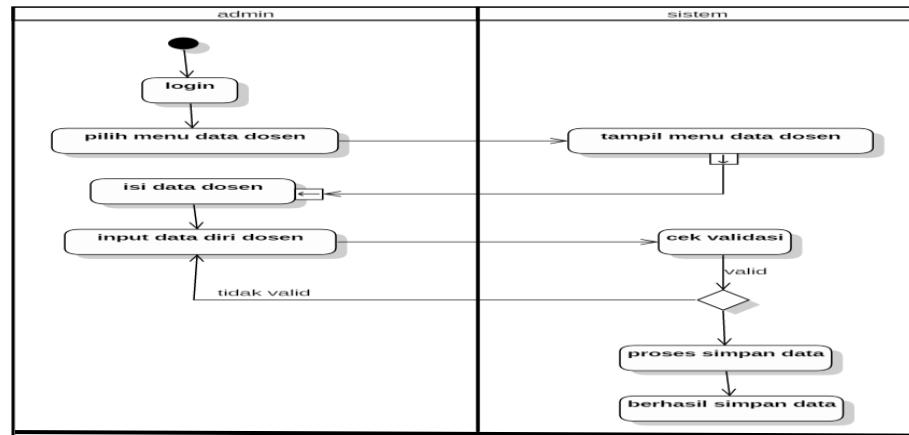


Gambar 4.3. Activity Diagram *input* mahasiswa.

Pada gambar 4.3 menjelaskan *activity input* mahasiswa yang dimana admin dapat melakukan penambahan data mahasiswa baru atau edit data mahasiswa yang sudah ada.

#### 4.2.2.3 Activity Diagram *input dosen*.

*Activity diagram input dosen* merupakan gambaran dari sebuah aktifitas input data dosen yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *input data dosen* di jelaskan pada Gambar 4.4 di bawah ini :

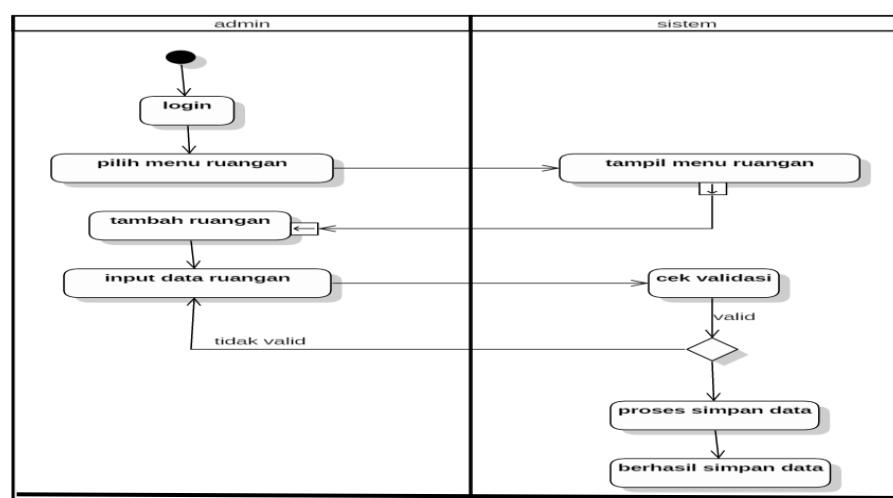


Gambar4.4. *Activity Diagram input dosen*.

Pada gambar 4.4 menjelaskan *activity diagram input dosen* dimana *admin* dapat menambahkan data dosen baru dan melakukan edit data dosen yang sudah ada.

#### 4.2.2.4 Activity Diagram input ruangan.

*Activity diagram input ruangan merupakan gambaran dari sebuah aktifitas input data ruangan yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *input ruangan* di jelaskan pada Gambar 4.5 di bawah ini :*

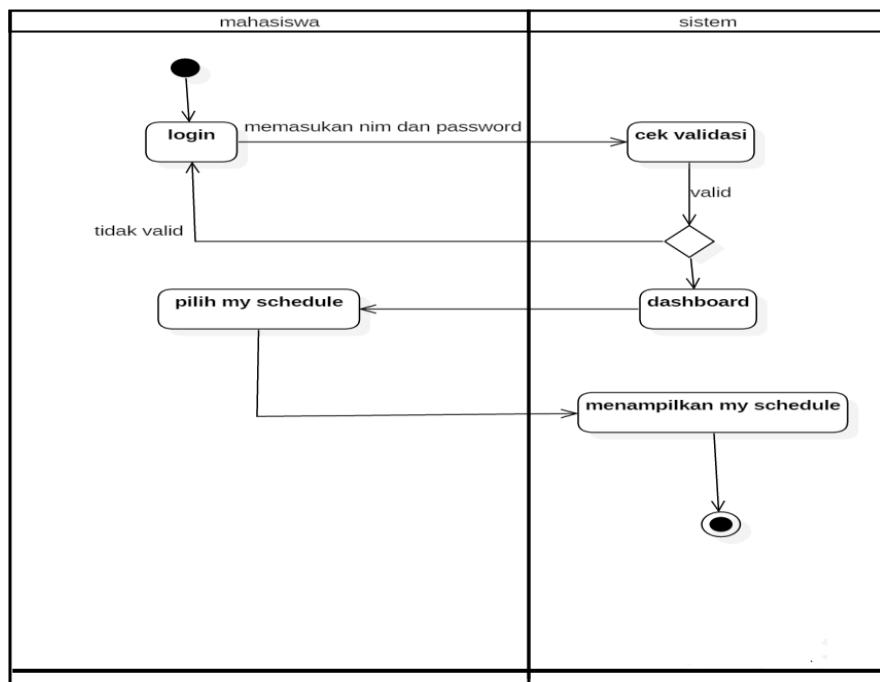


Gambar 4.5. *Activity Diagram input ruangan.*

Pada gambar 4.5 menjelaskan *activity diagram input ruangan* dimana Admin dapat melakukan tambah ruangan baru atau edit ruangan yang sudah ada.

#### 4.2.2.5 Activity Diagram login mahasiswa.

*Activity diagram login mahasiswa merupakan gambaran dari sebuah aktifitas login yang dilakukan oleh mahasiswa yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas input login mahasiswa di jelaskan pada Gambar 4.6 dibawah ini :*

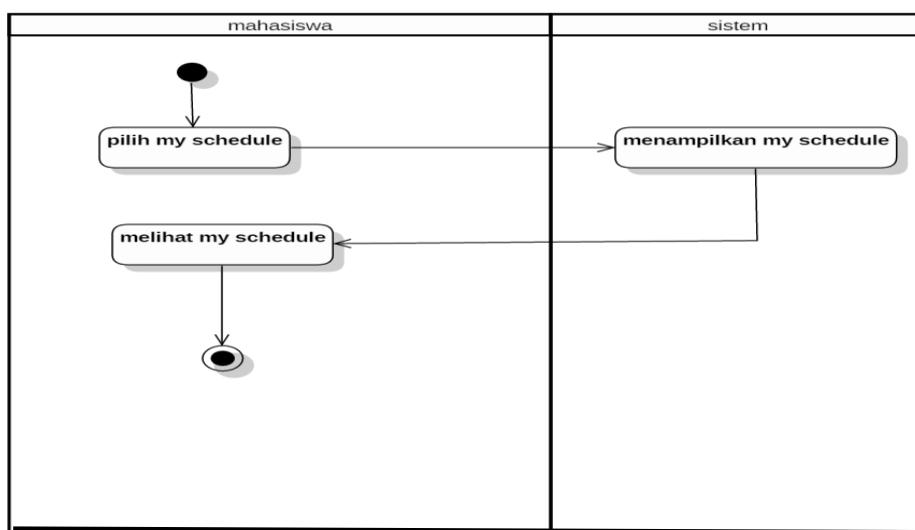


Gambar 4.6. *Activity Diagram login mahasiswa.*

Pada gambar 4.6 menjelaskan *activity diagram login* mahasiswa dimana mahasiswa dapat mempunyai hak akses untuk menggunakan *aplikasi* dengan memasukan *username* dan *password*.

#### 4.2.2.6 Activity Diagram *schedule mahasiswa.*

*Activity diagram schedule mahasiswa* merupakan gambaran dari sebuah aktifitas yang dilakukan oleh mahasiswa yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *schedule mahasiswa* dijelaskan pada Gambar 4.7 di bawah ini :

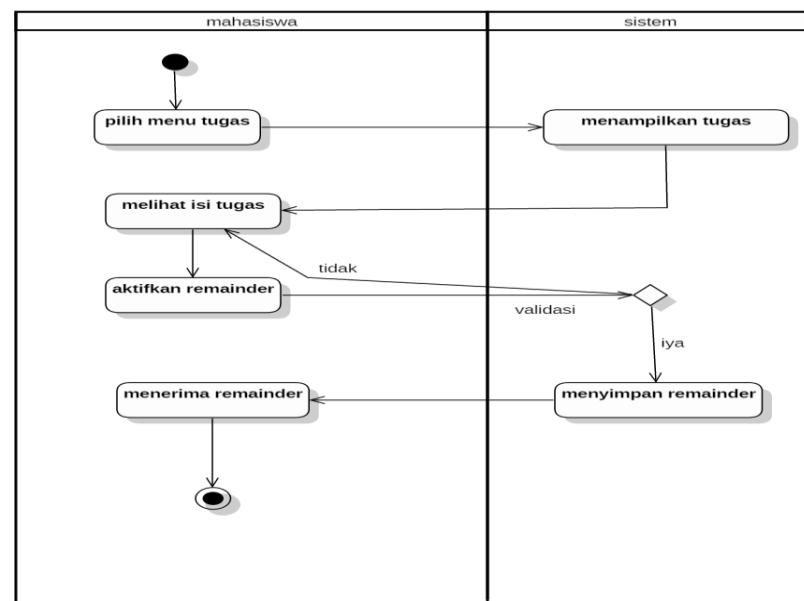


Gambar 4.7. Activity Diagram *schedule*.

Pada gambar 4.7 menjelaskan *activity diagram schedule* dimana mahasiswa dapat melihat kegiatan perkuliahan dari jadwal kuliah serta tugas

#### 4.2.2.7 Activity Diargam Menu Tugas.

*Activity diagram tugas* mahasiswa merupakan gambaran dari sebuah aktifitas yang di lakukan oleh mahasiswa yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *tugas* di jelaskan pada Gambar 4.8 di bawah ini :

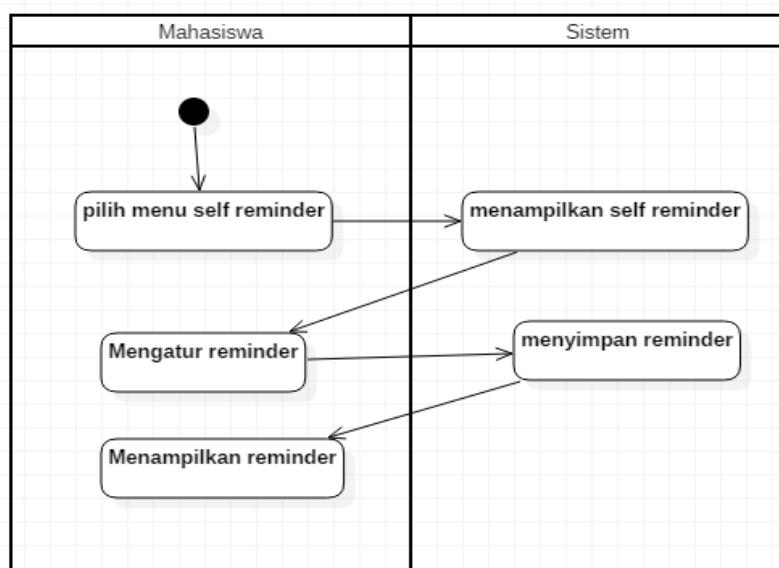


Gambar 4.8. Activity Diagram tugas.

Pada gambar 4.8 menjelaskan *activity diagram tugas* dimana mahasiswa dapat melihat kegiatan perkuliahan dari jadwal kuliah sertatugas.

#### 4.2.2.8 Activity Diagram Self Reminder.

*Activity diagram self reminder* merupakan gambaran dari sebuah aktifitas yang dilakukan oleh mahasiswa dan dosen yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas self reminder dijelaskan pada Gambar 4.9 di bawah ini.

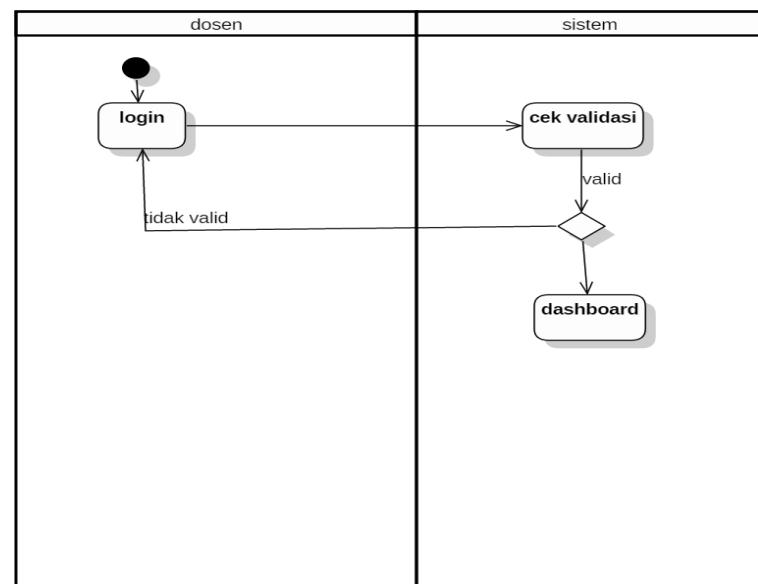


Gambar 4.9. *Activity Self Reminder*.

Pada gambar 4.9 menjelaskan *activity self reminder* dimana mahasiswa dapat mengatur reminder kegiatan perkuliahan tugas.

#### 4.2.2.9 Activity Diagram Login Dosen.

*Activity diagram login dosen* merupakan gambaran dari sebuah aktifitas yang di lakukan oleh dosen yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *login dosen* di jelaskan pada Gambar 4.10 di bawah ini :

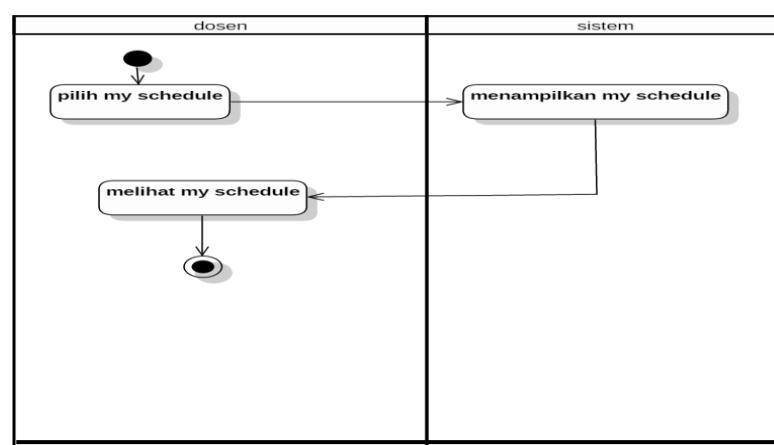


Gambar 4.10 *Activity Diagram Login Dosen*.

Pada gambar 4.10 menjelaskan *activity login dosen* dimana dosen dapat mememiliki hak akses ketika memasukan *username* dan *password*.

#### 4.2.2.10 Activity Diagram My Schedule Dosen.

Activity diagram login dosen merupakan gambaran dari sebuah aktifitas yang di lakukan oleh dosen yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *schedule dosen* di jelaskan pada Gambar 4.11 di bawah ini :

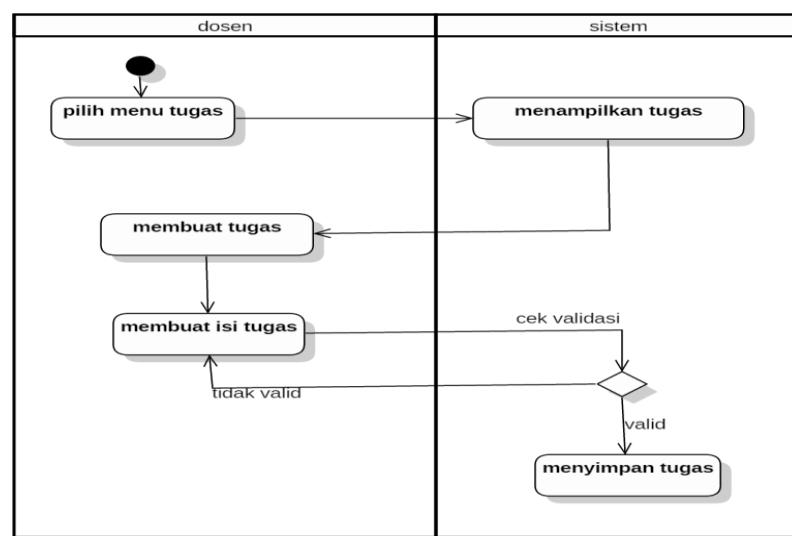


Gambar 4.11 Activity Diagram My Schedule.

Pada gambar 4.11 menjelaskan *activity my schedule* dimana dosen dapat melihat jadwal mengajar.

#### 4.2.2.11 Activity Diagram Tugas.

*Activity diagram tugas* merupakan gambaran dari sebuah aktifitas yang di lakukan oleh dosen yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *tugas* di jelaskan pada Gambar 4.12 di bawah ini :

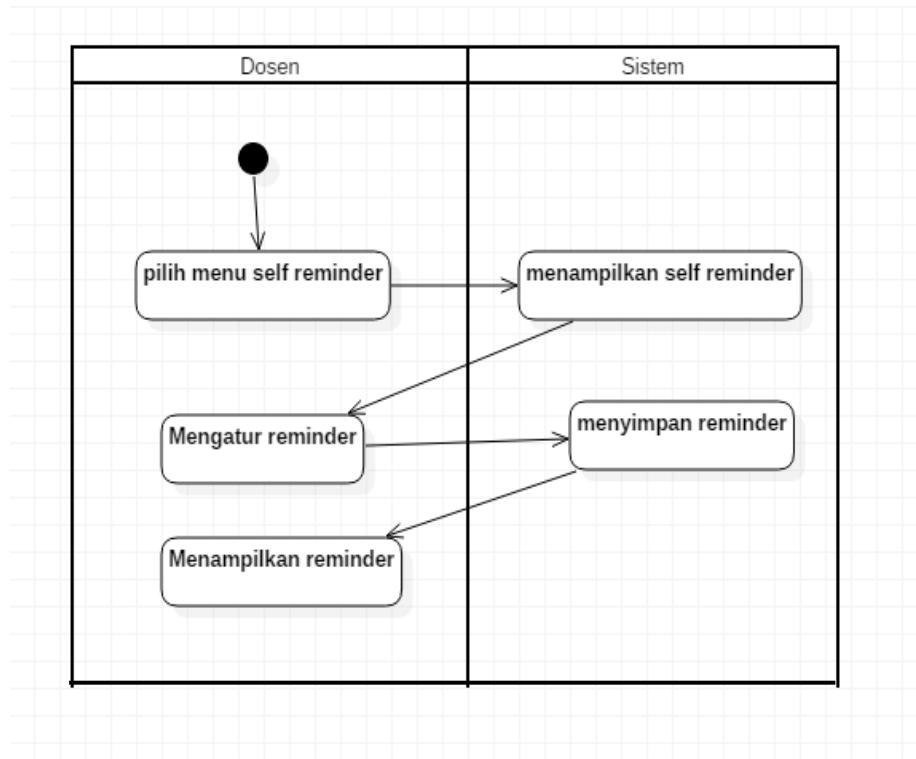


Gambar 4.12 *Activity Diagram Tugas*.

Pada gambar 4.12 menjelaskan *activity tugas* dimana dosen dapat membuat tugas dan mengirimkan tugas.

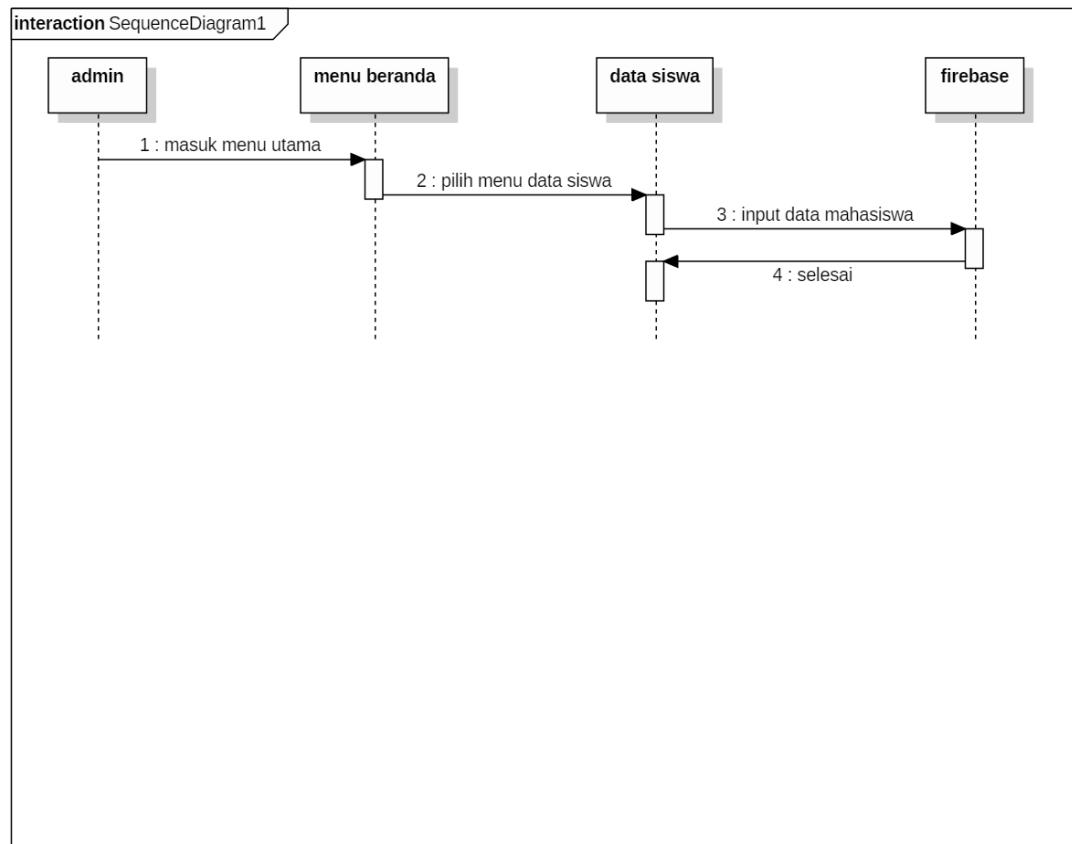
#### 4.2.2.12 Activity Diagram Self Reminder.

*Activity diagram tugas* merupakan gambaran dari sebuah aktifitas yang di lakukan oleh dosen yang terdapat pada sistem rancang bangun aplikasi pengingat tugas. Alur aktifitas *input data mahasiswa* di jelaskan pada Gambar 4.3 di bawah ini :



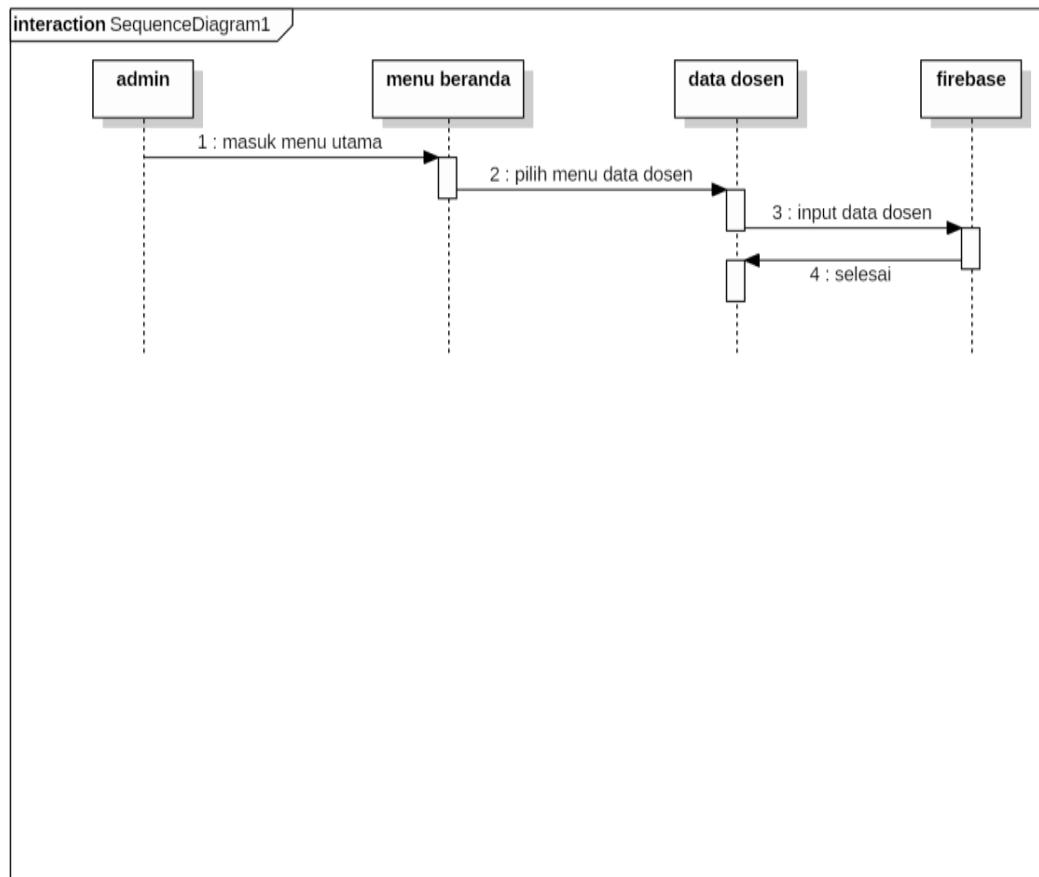
Gambar 4.13 *self reminder*

Pada gambar 4.13 menjelaskan *activity selfreminder* dimana dapat membuat reminder untuk mengingatkan tugas



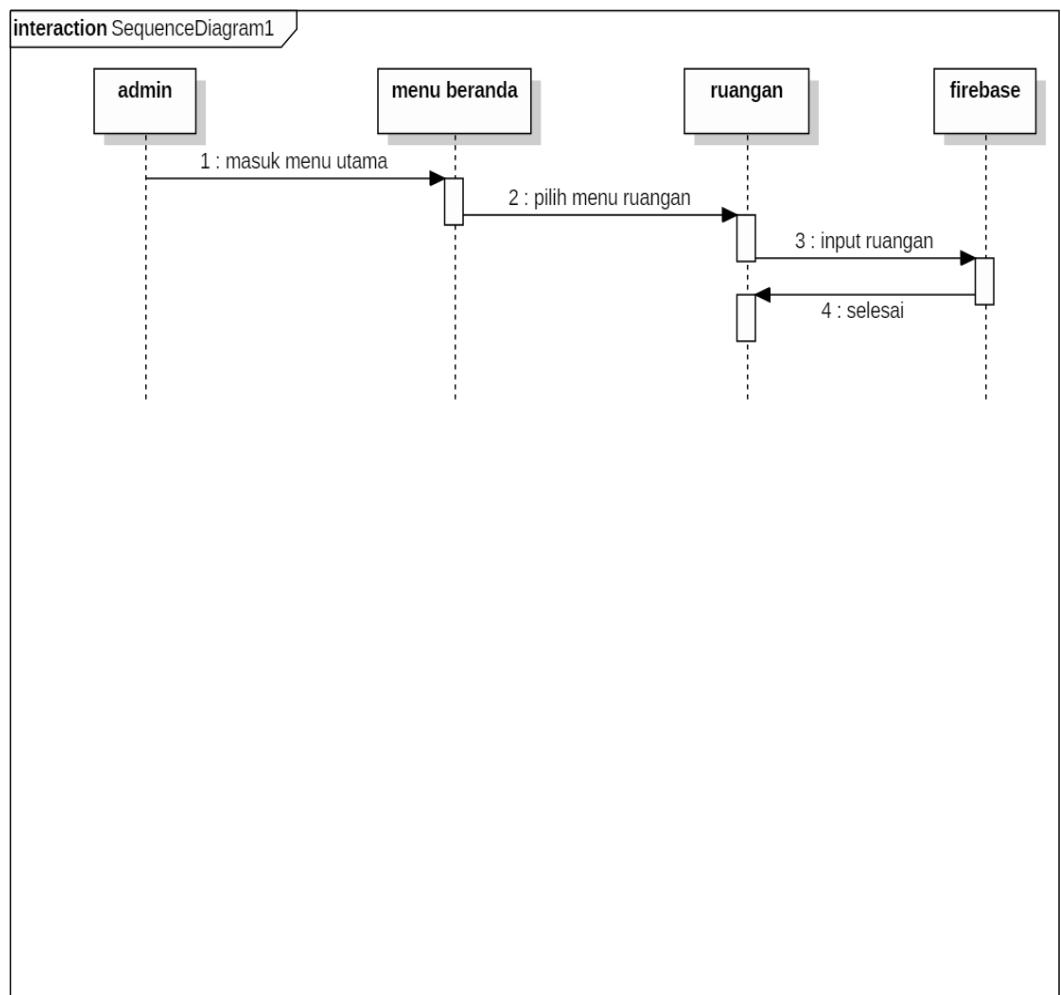
Gambar 4.15 *Sequence Diagram Data Mahasiswa.*

#### 4.2.3.1 Sequence Diagram Data Dosen.



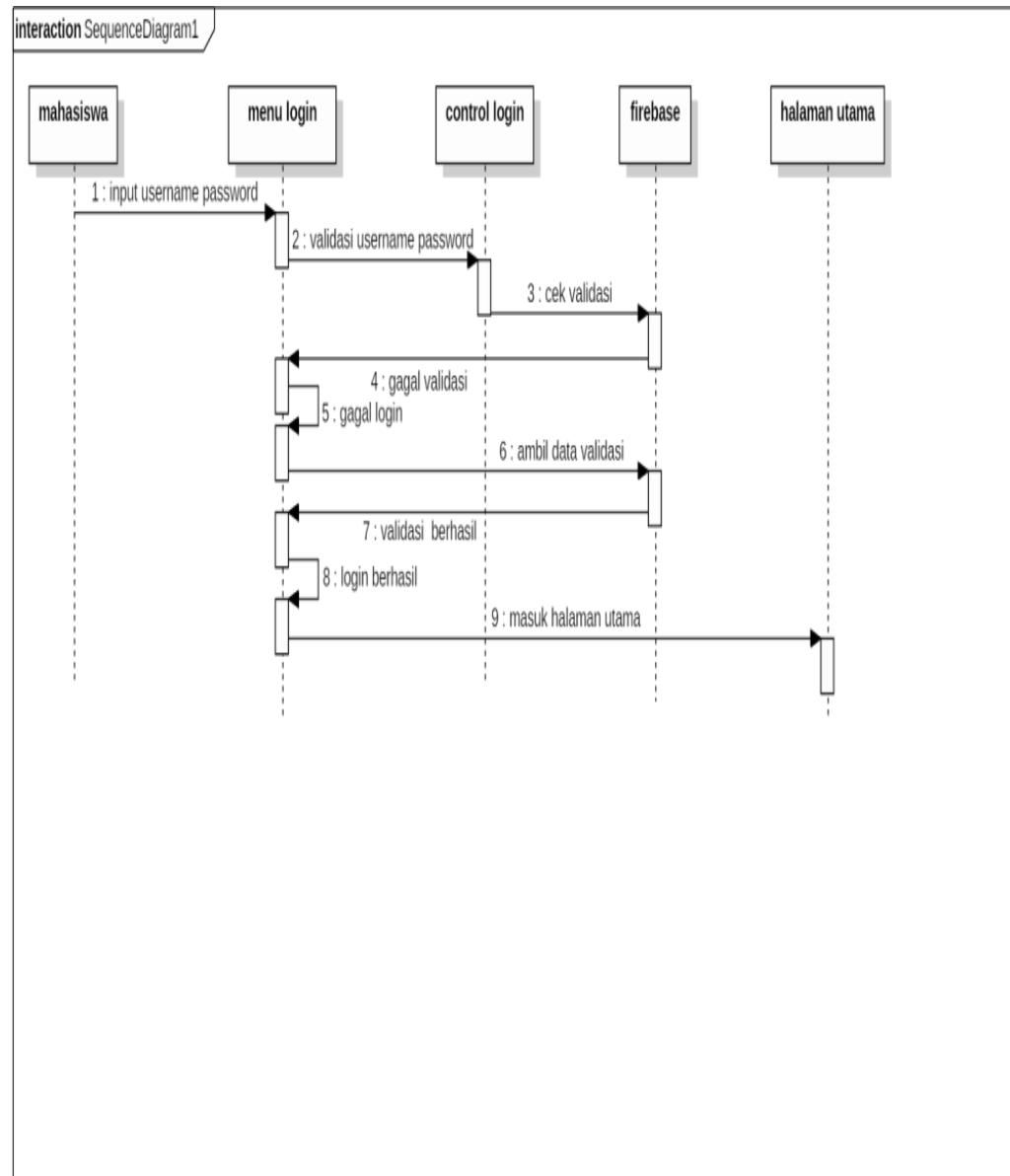
Gambar 4.16. *Sequence Diagram Data Dosen.*

#### 4.2.3.2 Sequence Diagram Ruangan.



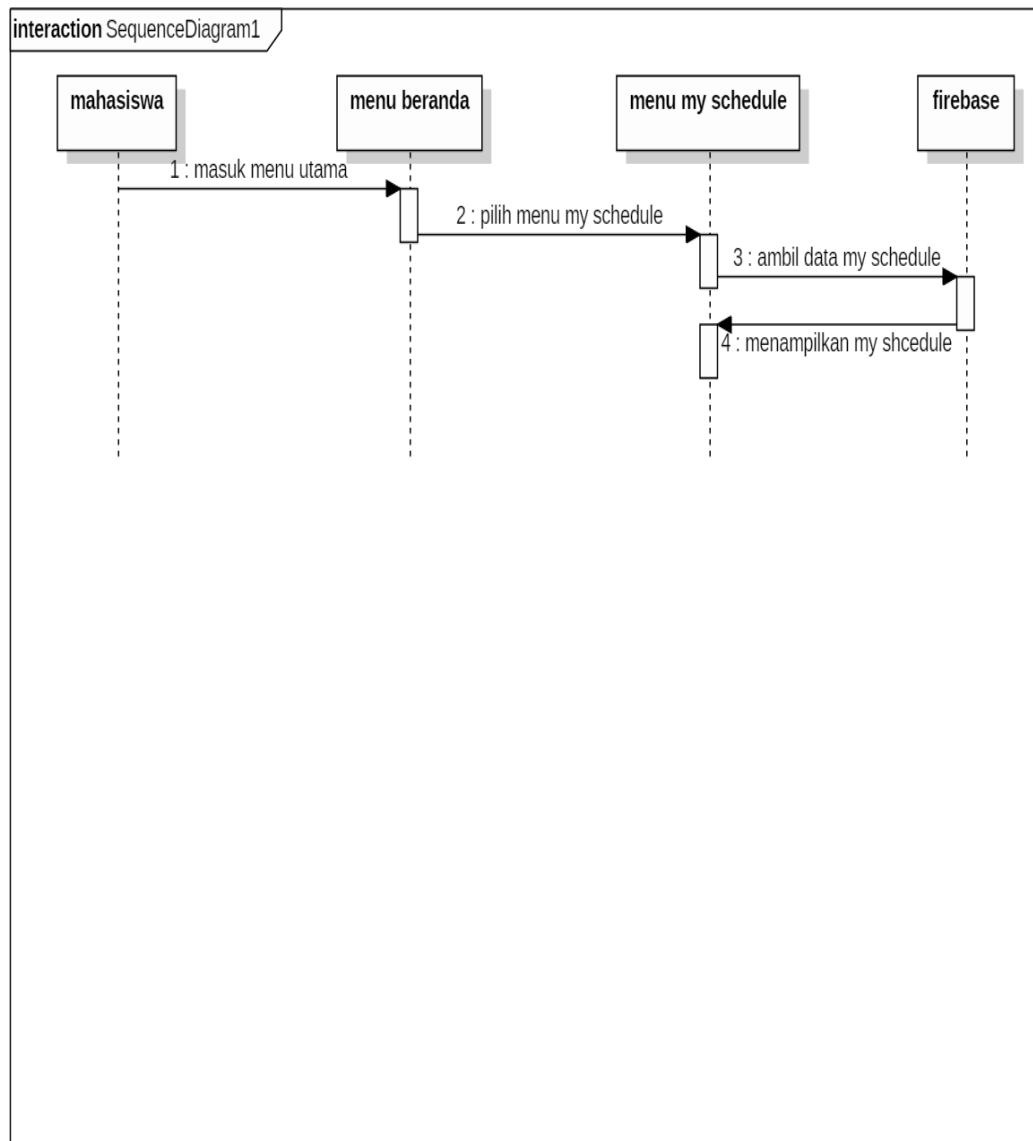
Gambar 4.17. *Sequence Diagram Ruangan.*

#### 4.2.3.3 Sequence Diagram Log in mahasiswa.



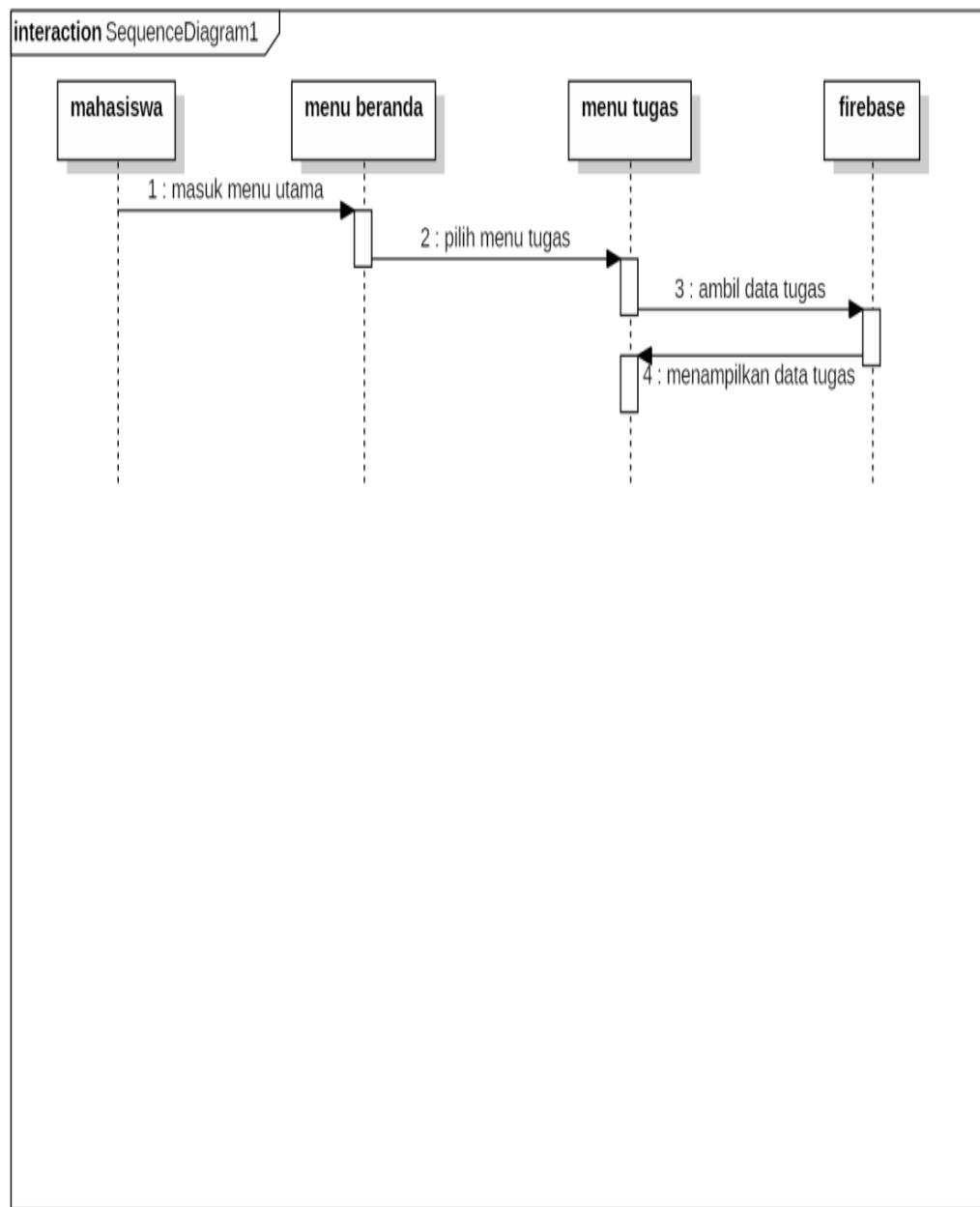
Gambar 4.18. Sequence Diagram Login Mahasiswa.

#### 4.2.3.4 Sequance Diagram *My Schedule*.



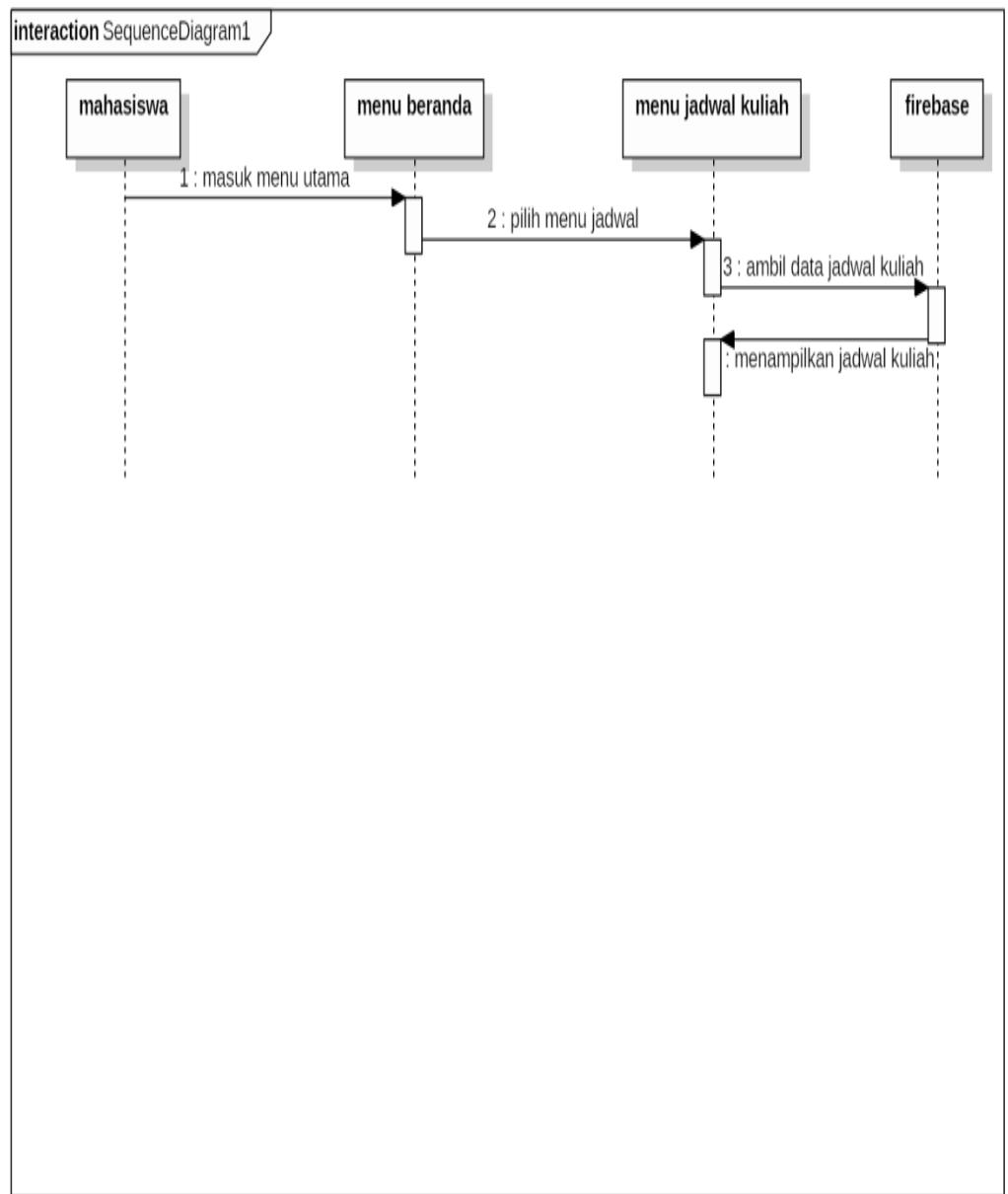
Gambar 4.14. *Sequance Diagram My Schedule*

#### 4.2.3.5 Sequance Diagram Tugas



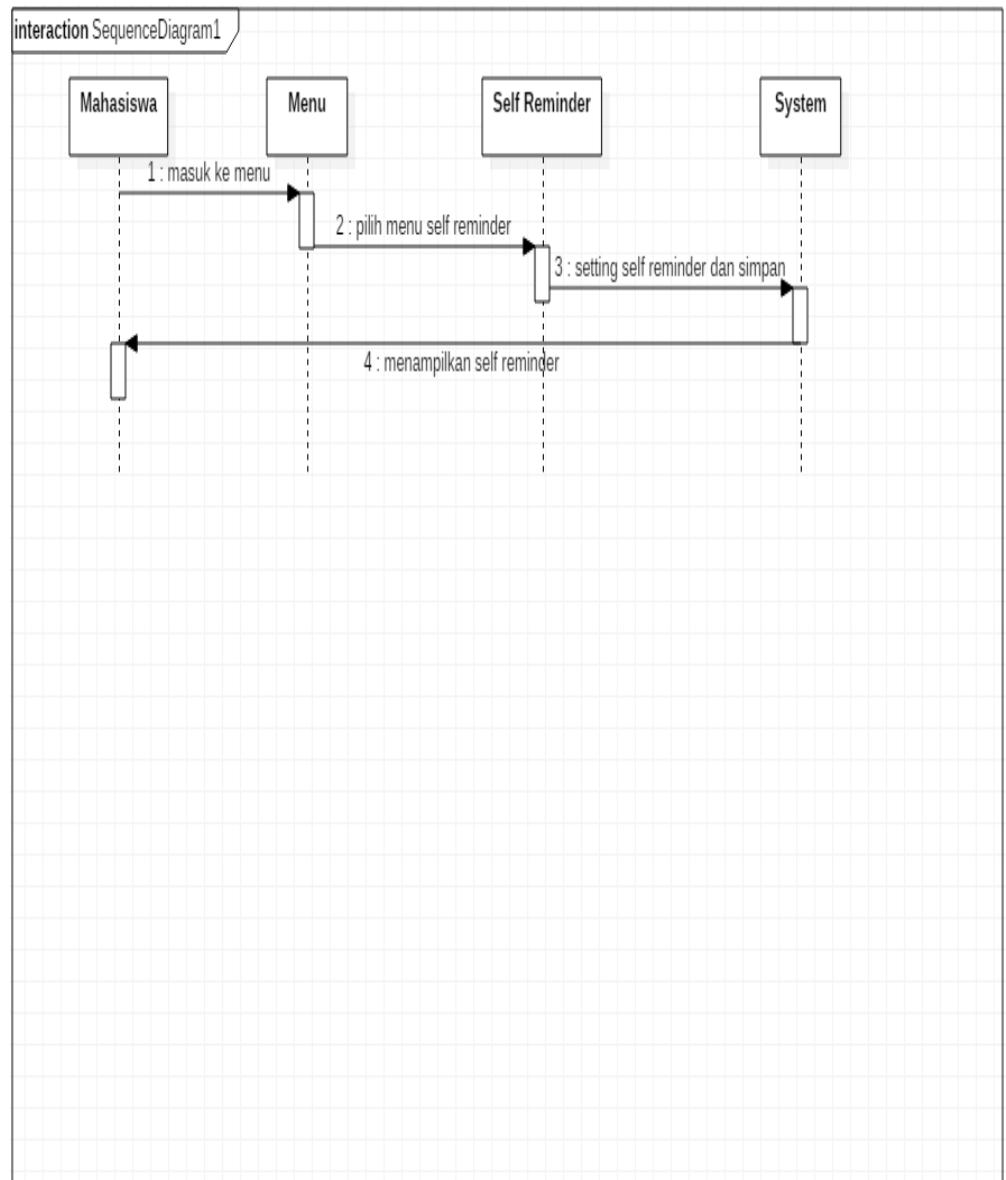
Gambar 4.15. Sequence Diagram Tugas

#### 4.2.3.6 Sequence Diagram *Self Reminder*



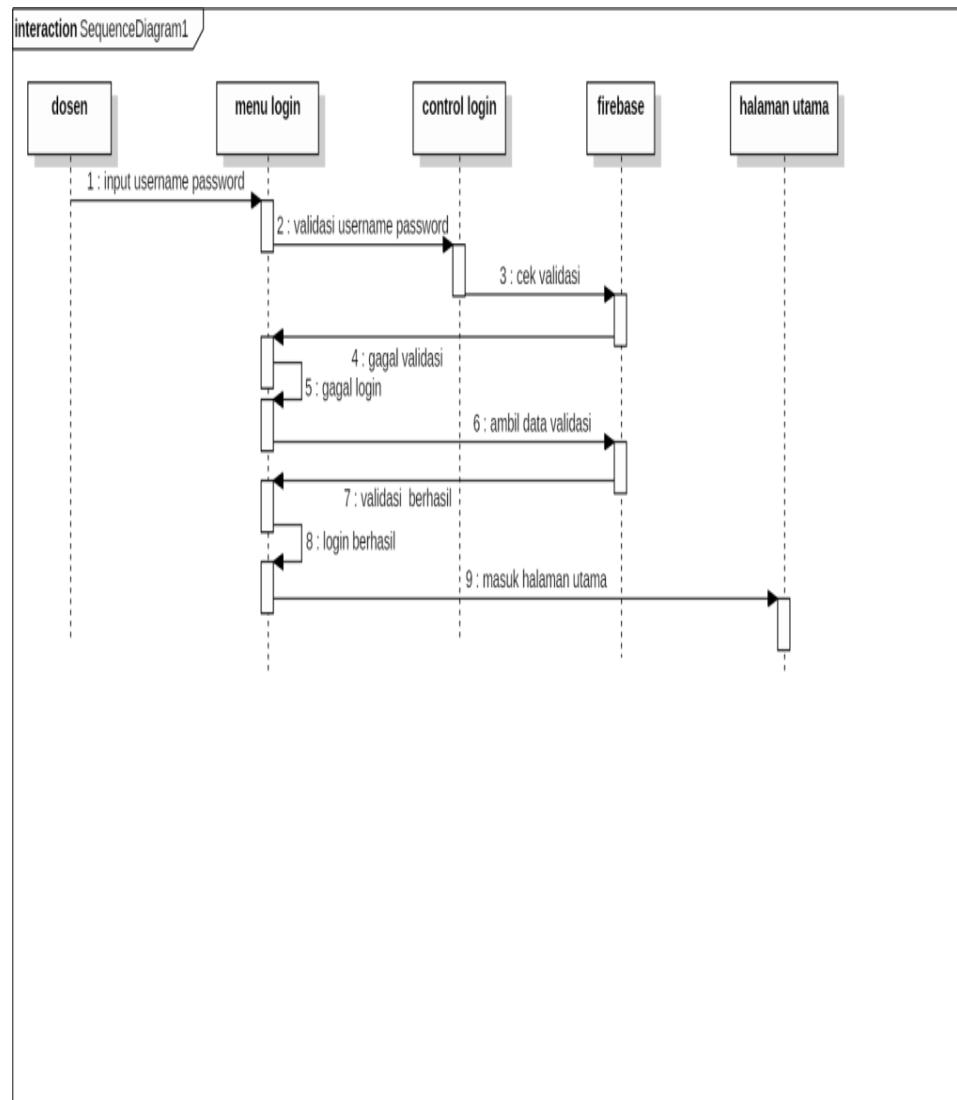
Gambar 4.16. *Sequence Diagram Jadwal Kuliah*

#### 4.2.3.7 Sequence Diagram *SelfReminder*



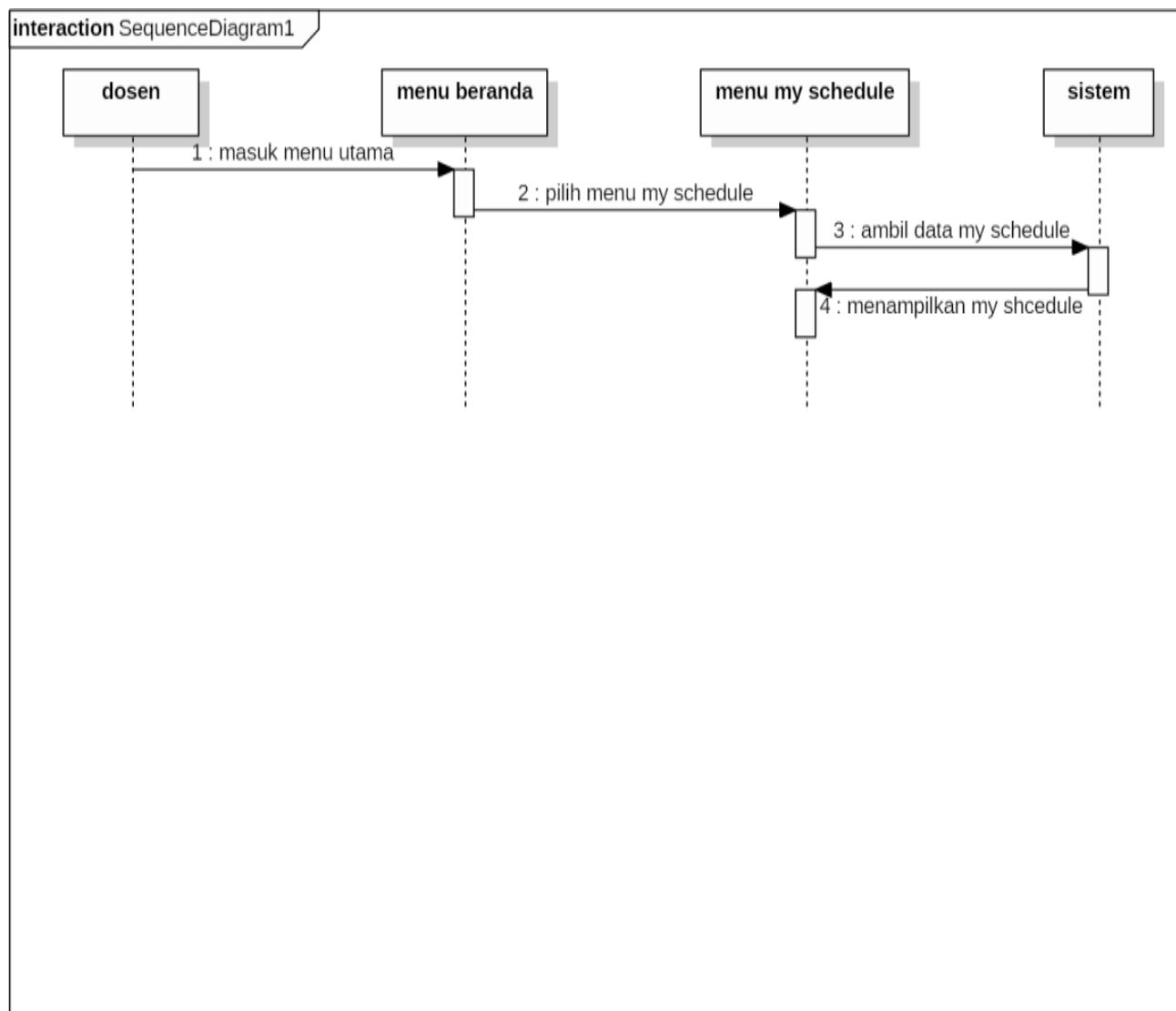
Gambar 4.17. *Sequence Diagram SelfReminder*

#### 4.2.3.8 Sequence Diagram Login Dosen



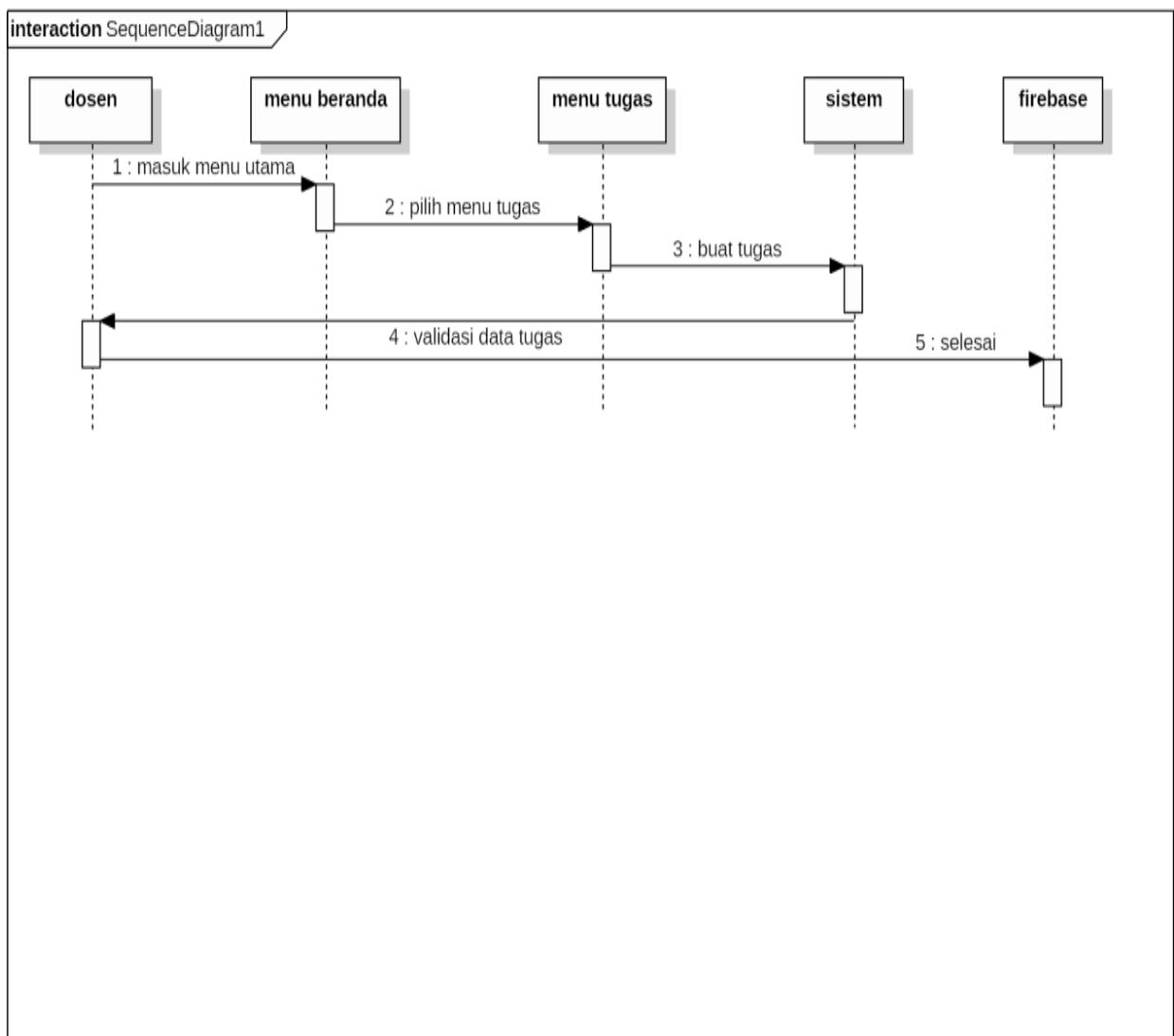
Gambar 4.18. Sequence Diagram Login Dosen

#### 4.2.3.9 Sequence Diagram My Schedule



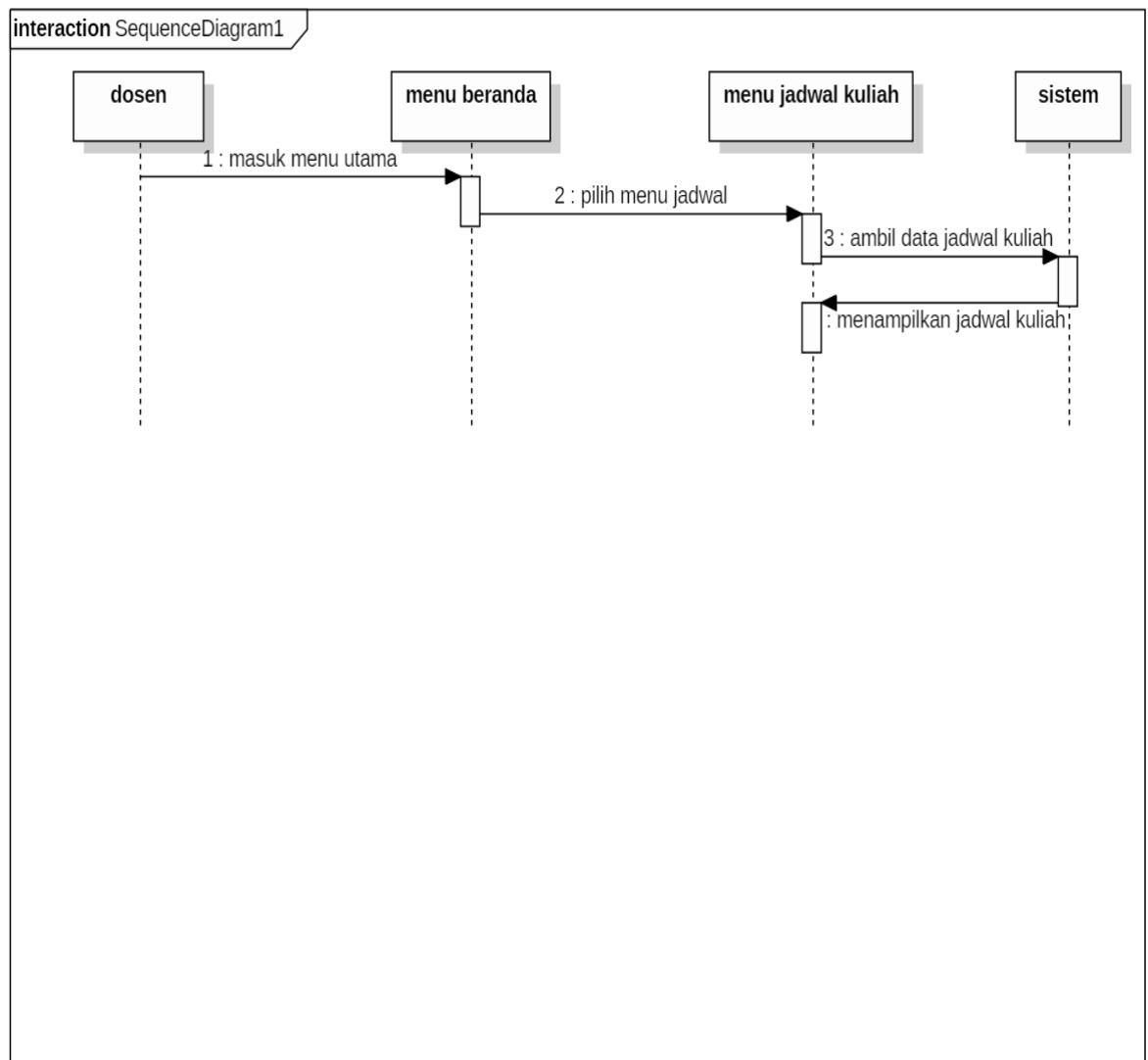
Gambar 4.18. Sequence Diagram My Schedule

#### 4.2.3.10 Sequence Diagram Tugas



Gambar 4.19 Sequence Diagram Tugas

#### 4.2.3.11 Sequance Diagram Jadwal Kuliah



Gambar 4.20 Sequence Diagram Jadwal Kuliah

### 4.3 Desain *Input* dan *Output*

Berikut merupakan rancangan desain *interface* desain Aplikasi *Class Remainder* studi kasus D.4 Teknik Informatika

#### 1. Desain Halaman Login Mahasiswa

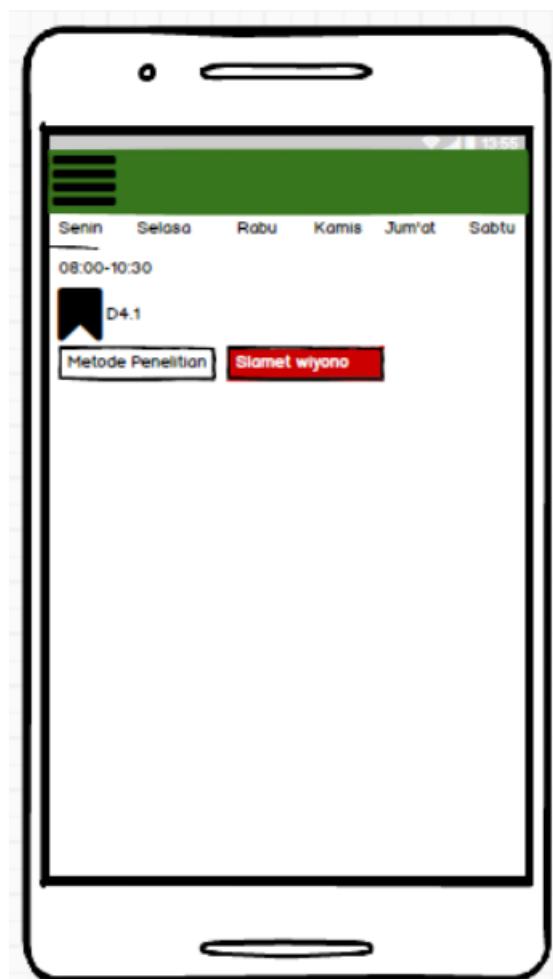
Pada gambar 4.18 ini mahasiswa harus melakukan login untuk masuk ke system dengan menginputkan username dan password



Gambar 4.18 Halaman Login Mahasiswa

## 2. Desain Halaman *My Schedule* Mahasiswa

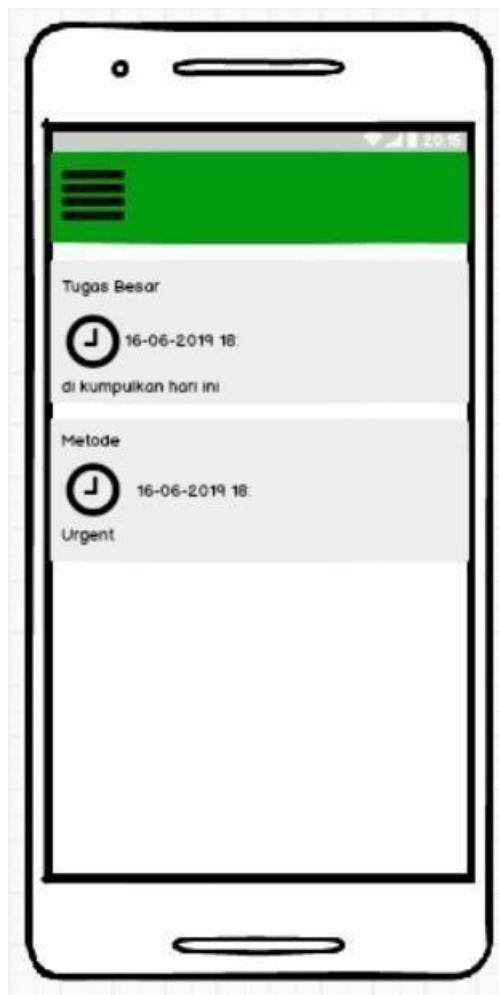
Pada gambar 4.19 dibawah ini mahasiswa dapat melihat my shcedule sesuai kelas



Gambar 4.19 Halaman My Schedule

### 3. Desain Halaman Menu Tugas Mahasiswa

Pada gambar 4.20 dibawah ini mahasiswa dapat menerima informasi seputar tugas yang telah diberikan oleh dosen



Gambar 4.20 Halaman Menu Tugas

#### 4. Desain Menu Login Admin

Pada gambar 4.21 dibawah ini admin harus melakukan login dengan memasukan email dan passworduntuk masuk ke system



Gambar 4.21 Desain Login admin

## 5. Desain dashboard admin

Pada gambar 4.22 dibawah ini system akan menampilkan menu dashboard yang dapat di akses oleh admin



Gambar 4.22 Halaman Dashboard admin

## 6. Desain Halaman classes admin

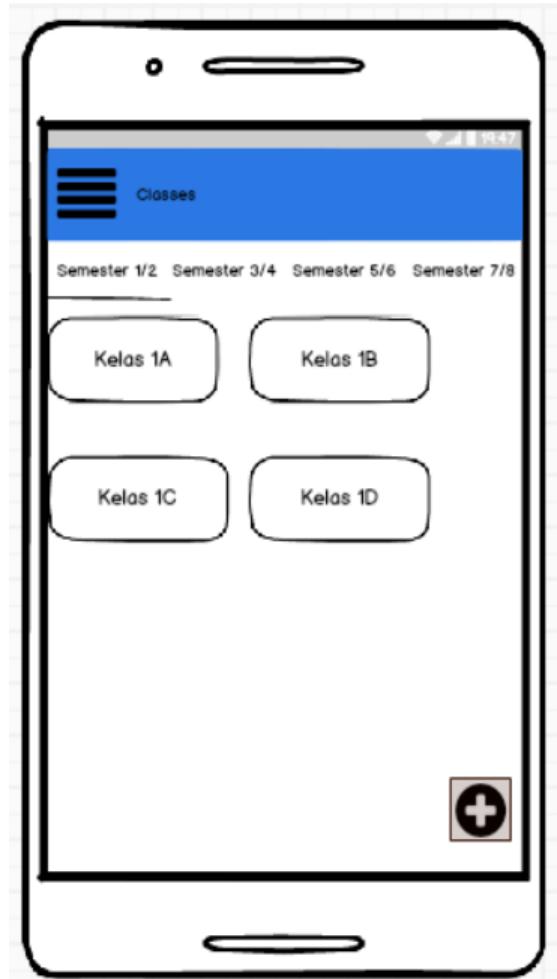
Pada gambar 4.23 dibawah ini admin dapat memilih menu sesuai yang di inginkan



Gambar 4.23 Halaman Classes admin

## 7. Desain Halaman Kelas

Berikut merupakan rancangan desain tampilan kelas Pada gambar 4.24 dibawah ini system akan menampilkan menu kelas yang dapat diatur oleh admin



Gambar 4.24 Halaman data kelas

## 8. Desain Halaman Tambah dan Edit

Pada gambar 4.25 dibawah ini admin dapat melakukan tambah data dan hapus data



Gambar 4.25 Halaman tambah dan editambah dan edi

## 9. Desain Halaman Login Dosen

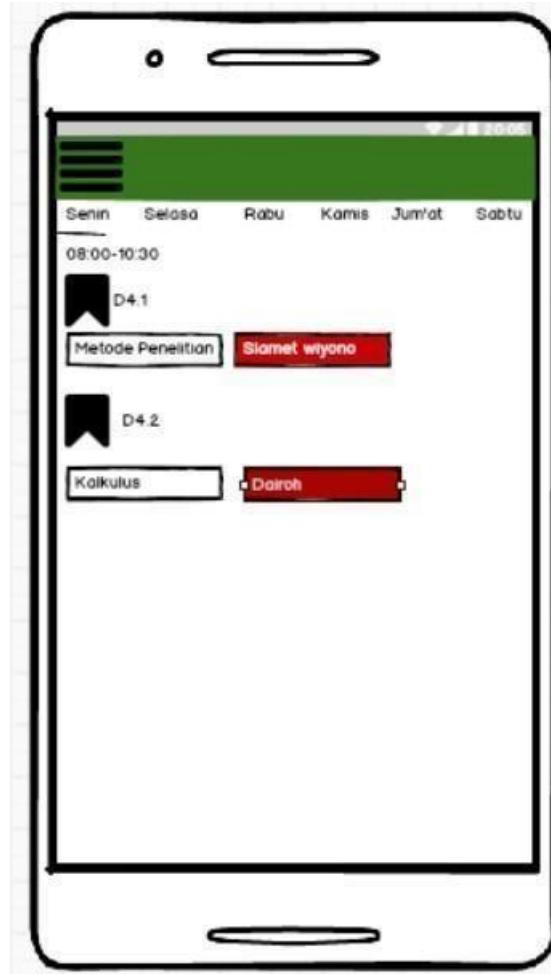
Pada gambar 4.26 dibawah ini dosen harus login dengan memasukan username dan password untuk bisa mengakses system



Gambar 4.26 Halaman login dosen

## 10. Desain Halaman My Schedule Dosen

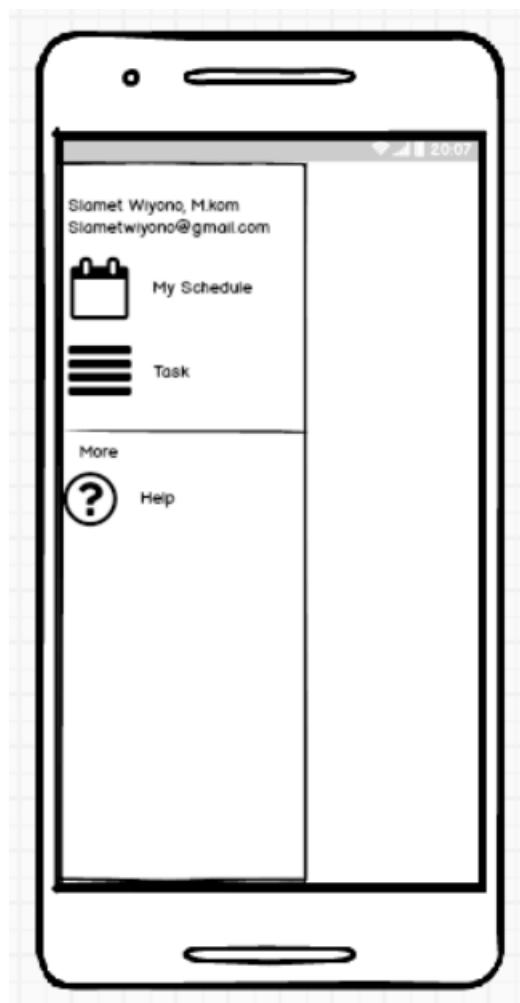
Pada gambar 4.27 dibawah ini system akan menampilkan my shcedule dosen



Gambar 4.27 Halaman My schedule dosen

## 11. Desain halaman Task Dosen

Pada gambar 4.28 dibawah ini system akan menampilkan task dosen yang dimana dosen dapat membuat tugas untuk mahasiswa



Gambar 4.28 Halaman Task Dosen

## 12. Desain Halaman Buat Tugas

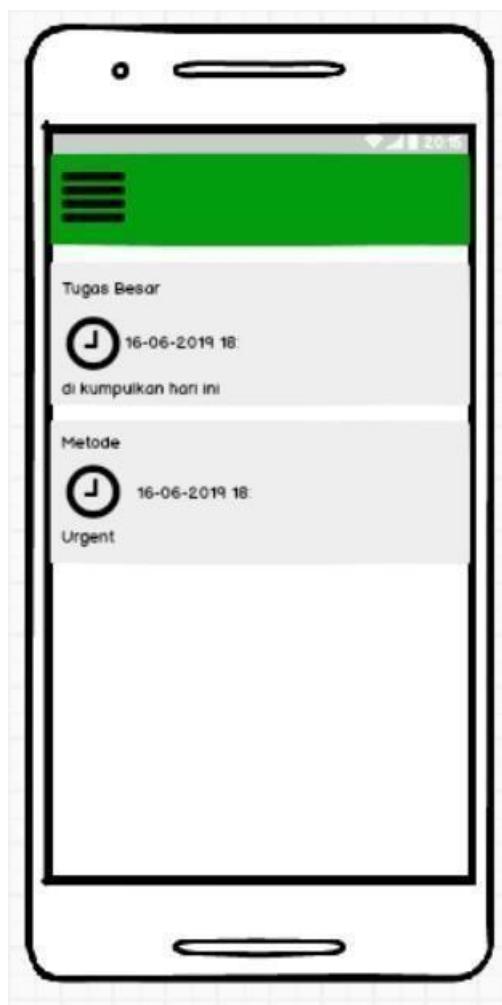
Pada gambar 4.29 dibawah ini dosen dapat membuat tugas untuk mahasiswa



Gambar 4.29 Halaman buat tugas

### **13. Desain Halaman detail tugas**

Pada gambar 4.30 di bawah ini dosen dapat melihat detail tugas yang telah di buat



Gambar 4.30 Halaman Detail Tugas

## **BAB V**

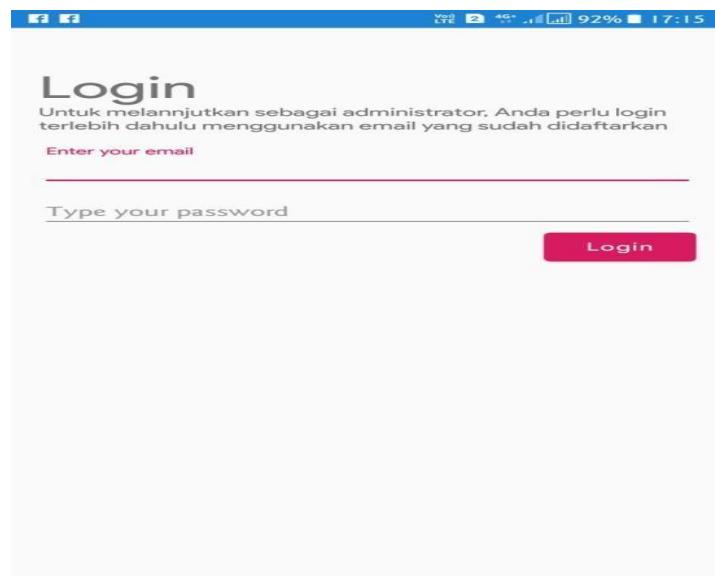
### **HASIL DAN PEMBAHASAN PENELITIAN**

#### **5.1. Hasil Penelitian**

Hasil penelitian berupa rancang bangun aplikasi pengingat tugas yang dilakukan pengujian dengan menggunakan white box dan black box.

##### **5.1.1 Tampilan aplikasi login admin**

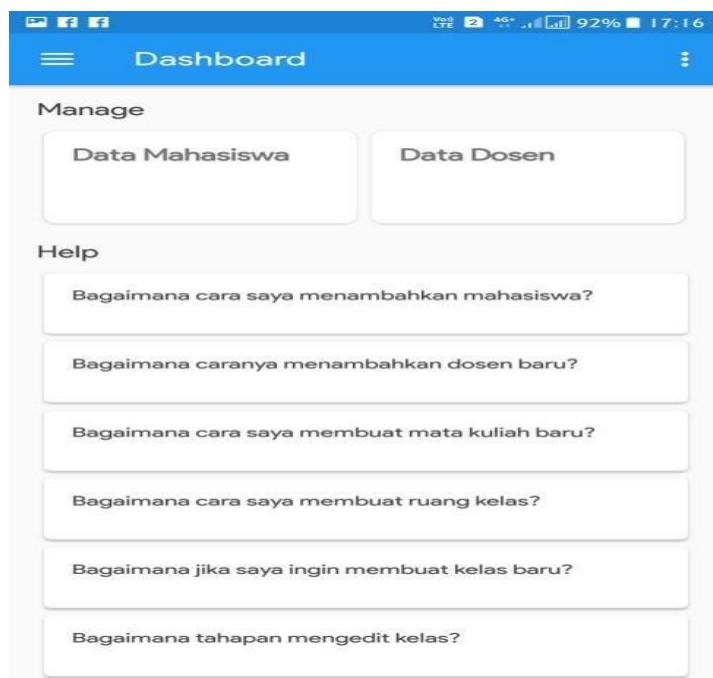
Halaman utama merupakan halaman *login* yang dapat digunakan untuk *login admin*. Tampilan halaman *login admin* dapat dilihat dari gambar 5.1.



Gambar 5.1 Tampilan Login admin.

### 5.1.2 Tampilan Halaman Beranda *admin*

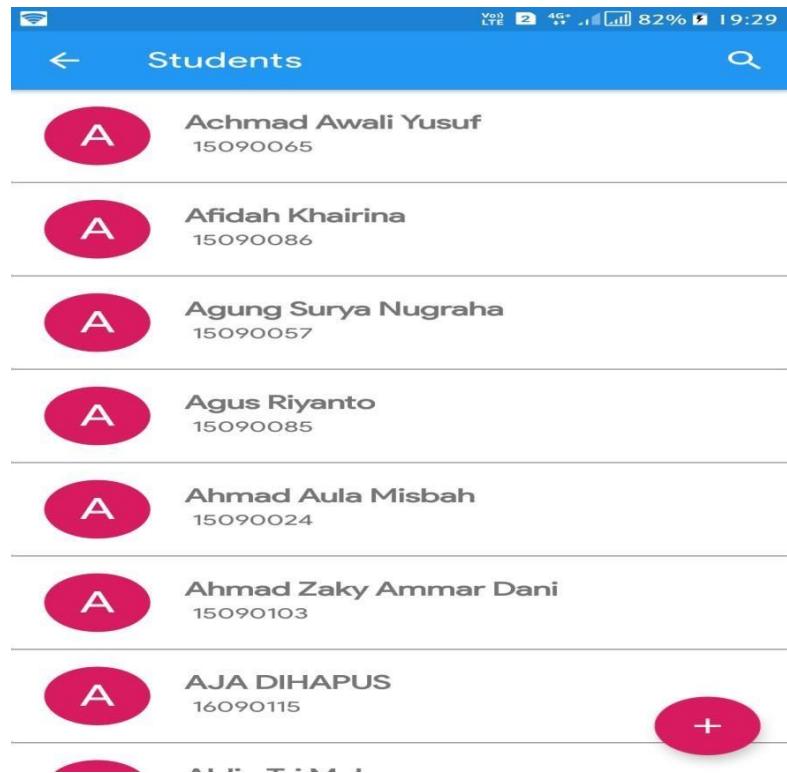
Halaman beranda *admin* merupakan halaman yang berisi daftar Data Mahasiswa, Data Dosen, Ruangan, Mata Kuliah. Halaman *dashboard* merupakan halaman awal yang tampil setelah *admin* melakukan *login* diawal aplikasi. Tampilan halaman *dashboard admin* dapat dilihat pada gambar 5.2.



Gambar 5.2 Tampilan Halaman Beranda *admin*.

### 5.1.3 Tampilan Halaman Data Mahasiswa

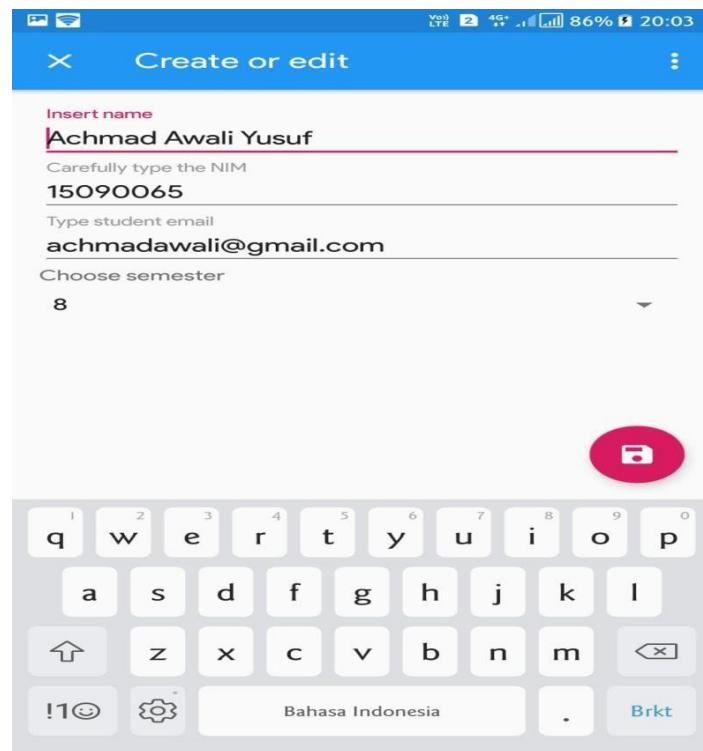
Halaman Data Mahasiswa merupakan halaman yang berisi Data Mahasiswa yang bisa di tambahkan oleh *admin* setelah *admin* melakukan login. Data Mahasiswa ini terdiri dari Nama, NIM, dan E-Mail. Tampilan halaman Data Mahasiswa dapat dilihat pada gambar 5.3.



Gambar 5.3. Tampilan Halaman Data Mahasiswa

#### 5.1.4 Tampilan Halaman Detail Data Mahasiswa

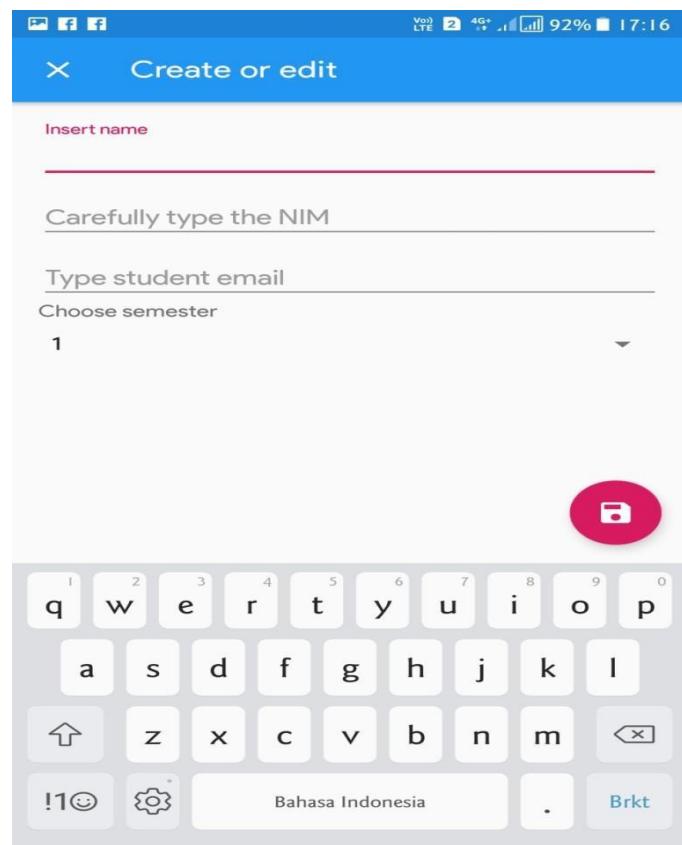
Halaman Detail Data Mahasiswa adalah halaman yang dimana *admin* dapat melihat Data Mahasiswa dan dapat melakukan *edit*, *delete*, dan *update* Data Mahasiswa. Tampilan Halaman Detail Data Mahasiswa dapat dilihat pada gambar 5.4.



Gambar 5.4. Tampilan Halaman Detail Data Mahasiswa

### 5.1.5 Tampilan Halaman Tambah Mahasiswa

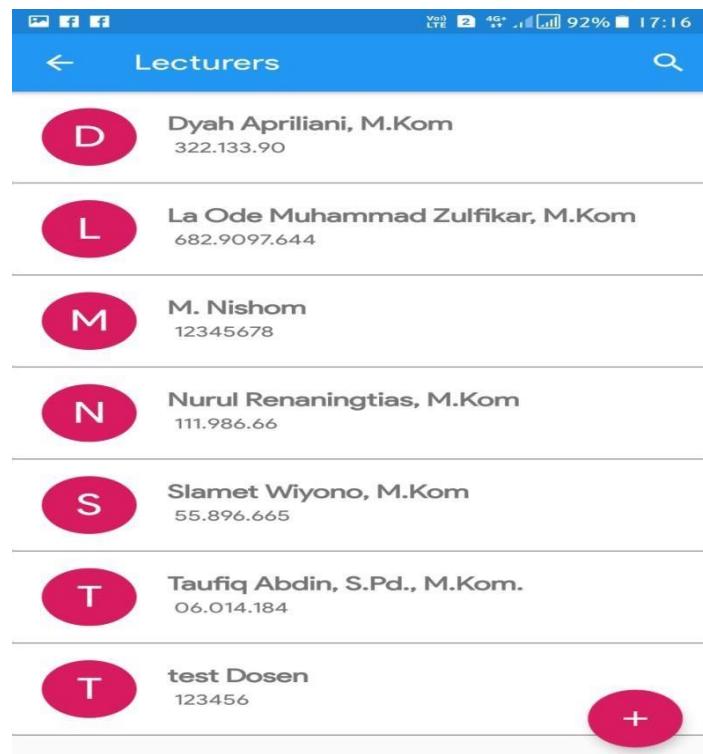
Halaman Tambah Mahasiswa merupakan halaman dimana *admin* dapat menambahkan Data Mahasiswa Baru yang dimana halaman ini dapat diakses *admin* ketika *admin* sudah melakukan *login* ke *system*. Tampilan halaman tambah mahasiswa dapat dilihat pada gambar 5.5.



Gambar 5.5 Tampilan Halaman Tambah Data Mahasiswa

### 5.1.6 Tampilan Halaman Data Dosen

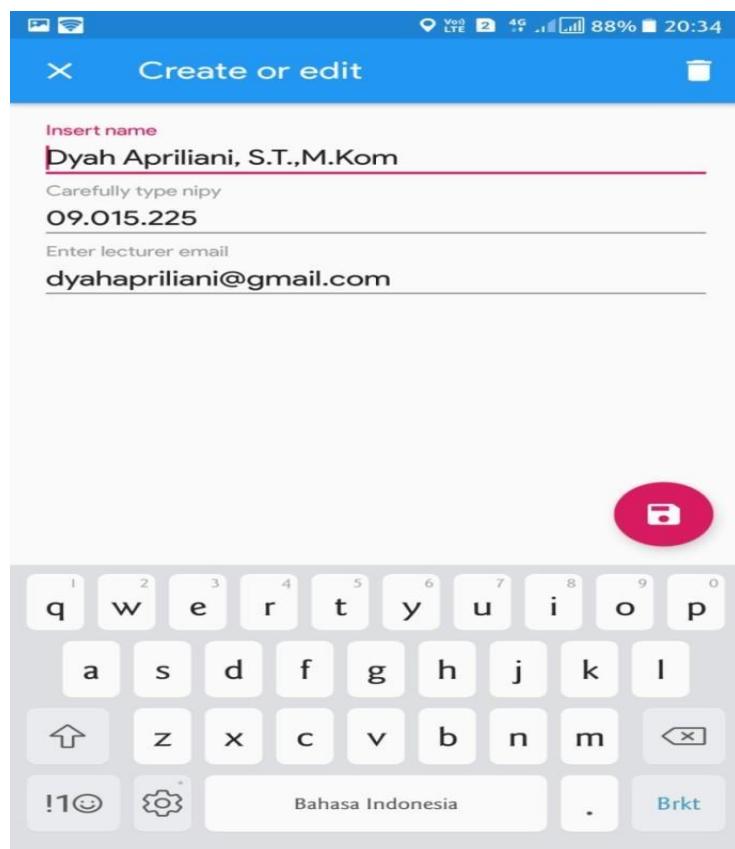
Halaman Data Dosen merupakan halaman yang berisi Data Dosen yang bisa ditambahkan oleh *admin* setelah *admin* melakukan login. Data Dosen ini terdiri dari Nama, NIPY, dan E- Mail. Tampilan halaman Data Dosen dapat dilihat pada gambar 5.6.



Gambar 5.6 Tampilan Halaman Tambah Data Dosen

### 5.1.7 Tampilan Halaman Detail Data Dosen

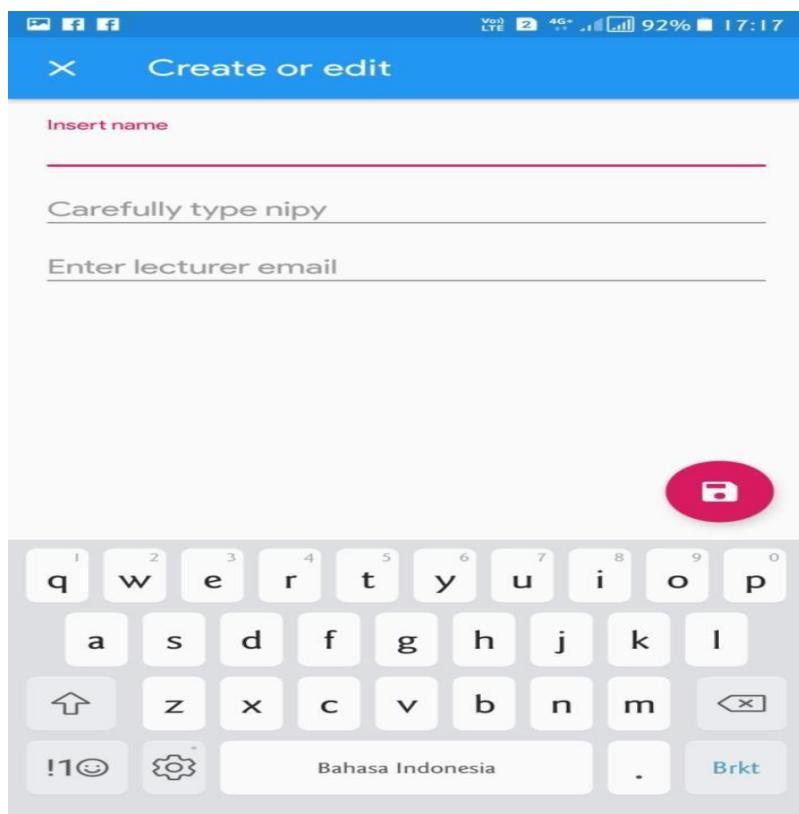
Halaman Detail Data Dosen adalah halaman yang dimana *admin* dapat melihat Data Dosen dan dapat melakukan *edit*, *delete*, dan *update* Data Dosen. Tampilan Halaman Detail Data Dosen dapat dilihat pada gambar 5.7.



Gambar 5.7 Tampilan Halaman Detail Data Dosen

### 5.1.8 Tampilan Halaman Tambah Dosen

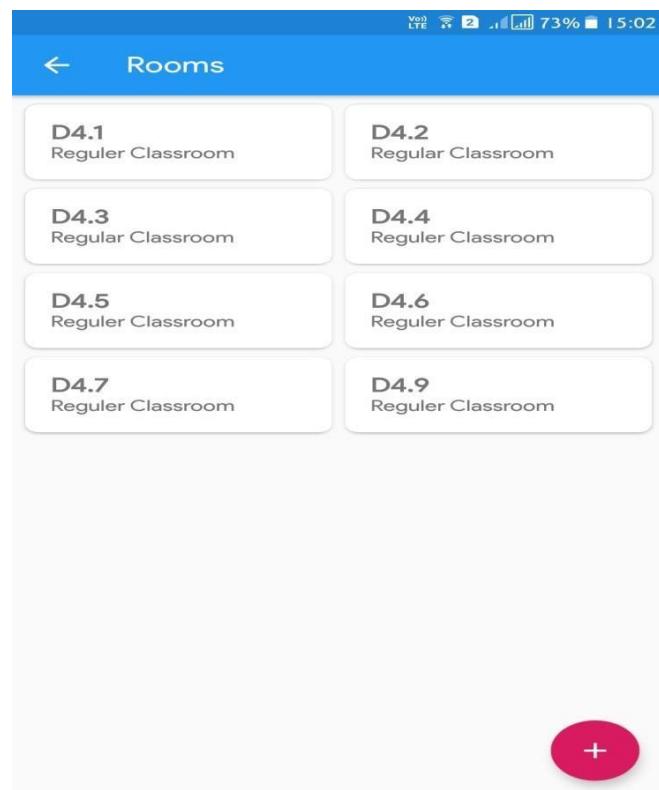
Halaman Tambah Dosen merupakan halaman dimana *admin* dapat menambahkan Data Dosen Baru, yang dimana halaman ini dapat diakses *admin* ketika *admin* sudah melakukan *login* ke *system*. Tampilan halaman tambah mahasiswa dapat dilihat pada gambar 5.8.



Gambar 5.8 Tampilan Halaman Tambah Data Dosen.

### 5.1.9 Tampilan Halaman Menu Ruangan

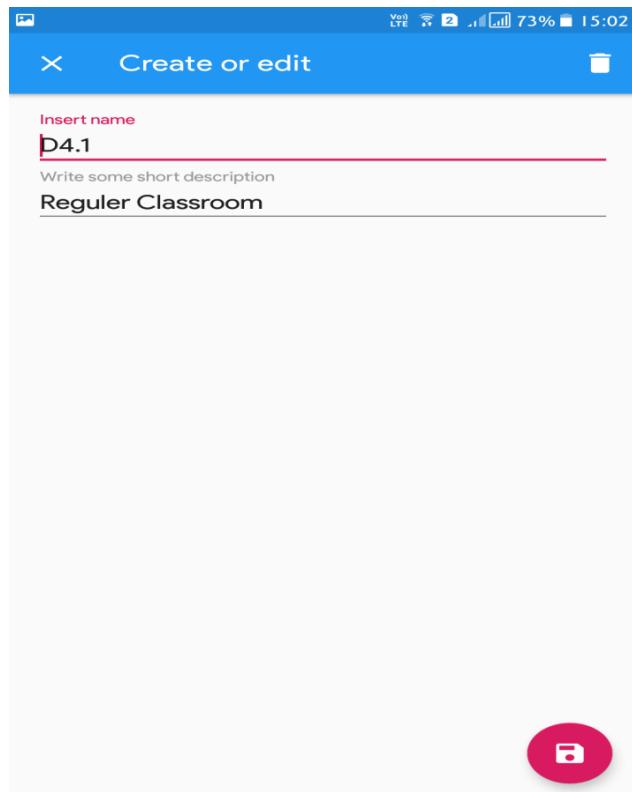
Halaman menu ruangan merupakan halaman dimana admin dapat menambahkan ruangan yang tersedia. Halaman ini dapat diakses ketika *admin* sudah melakukan *login* ke *system*. Tampilan halaman menu ruangan dapat di lihat pada gambar 5.9.



Gambar 5.9 Tampilan Halaman Menu Ruangan

### 5.1.10 Tampilan Halaman Detail Ruangan

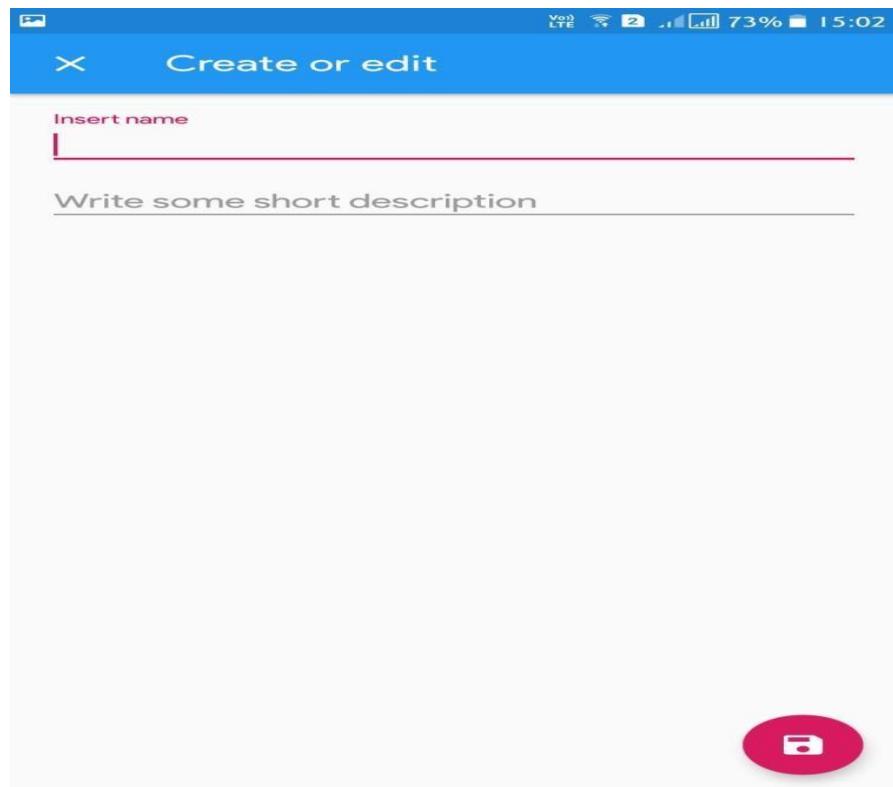
Halaman ini merupakan halaman dimana *admin* dapat melihat *Detail* ruangan yang telah ditambahkan *admin*. Halaman ini dapat diakses oleh *admin* ketika sudah melakukan *login* ke *system*. *Admin* dapat melakukan edit, hapus dan update data ruangan. Tampilan halaman detail ruangan dapat dilihat pada gambar 5.10.



Gambar 5.10 Tampilan Halaman Menu Ruangan

### 5.1.11 Tampilan Halaman Tambah Ruangan

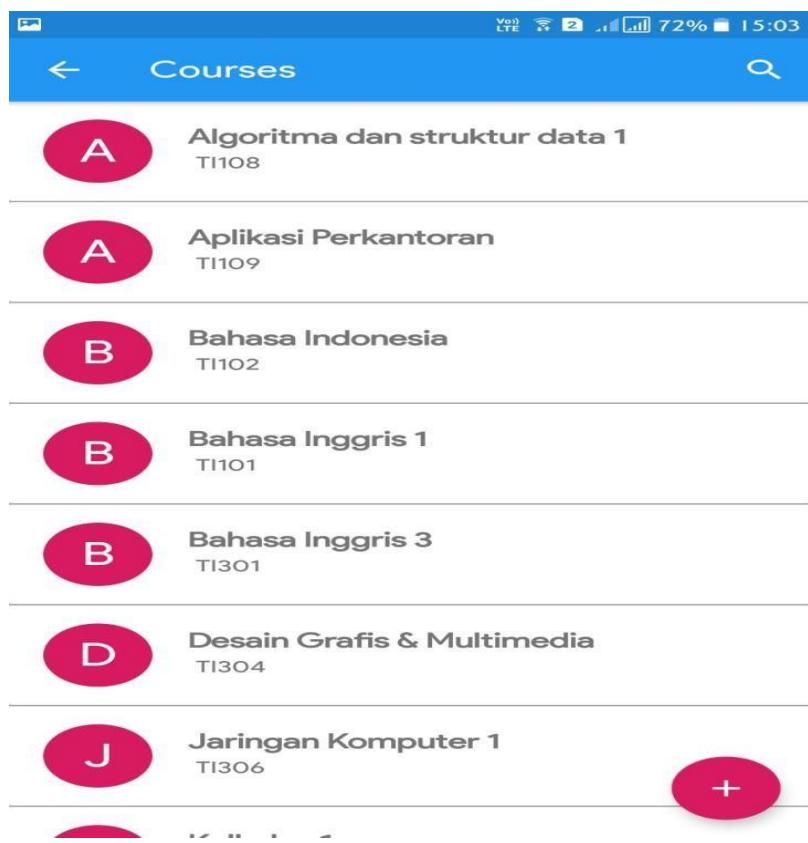
Halaman ini merupakan halaman dimana *admin* dapat menambahkan ruangan baru. Halaman ini dapat diakses ketika *admin* sudah melakukan *login* ke *system*. Tampilan halaman tambah ruangan dapat dilihat pada gambar 5.11.



Gambar 5.11 Tampilan Halaman Tambah Ruangan

### 5.1.12 Tampilan Halaman Mata Kuliah

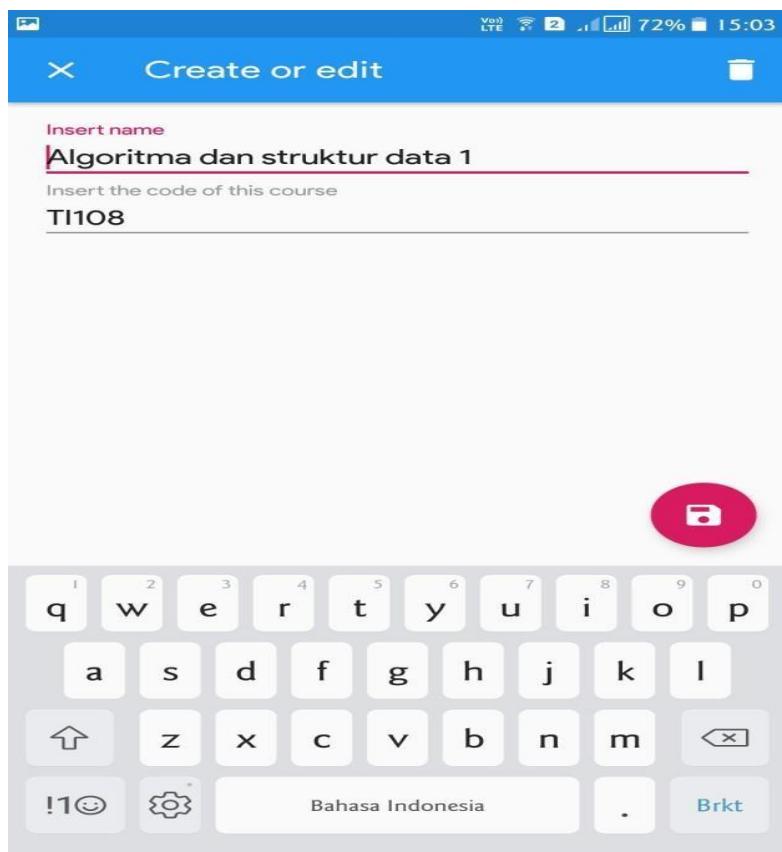
Halaman ini merupakan halaman dimana *admin* dapat melihat data mata kuliah, halaman ini dapat diakses *admin* ketika *admin* sudah melakukan *login* ke *system*. halaman mata kuliah dapat dilihat pada gambar 5.12.



Gambar 5.12 Tampilan Halaman Mata Kuliah

### 5.1.13 Tampilan Halaman Detail Mata Kuliah

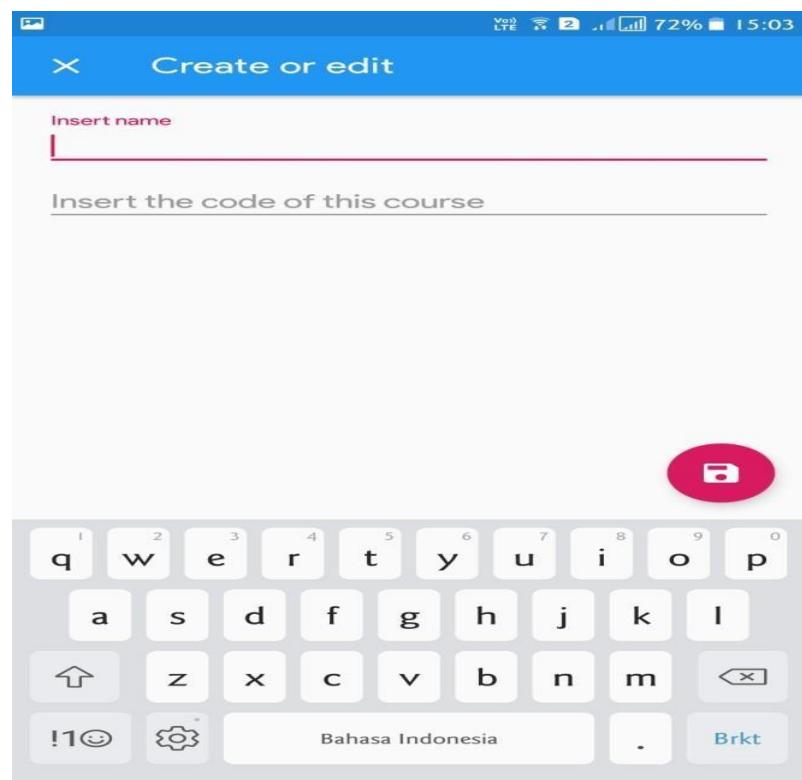
Halaman ini merupakan halaman dimana *admin* dapat melihat, edit, hapus dan update data mata kuliah yang sudah ditambahkan. Halaman ini dapat di akses setelah *admin* melakukan *login* ke *system*. halaman detail mata kuliah dapat dilihat pada gambar 5.13.



Gambar 5.13 Tampilan Halaman Detail Mata Kuliah

#### 5.1.14 Tampilan Halaman Tambah Mata Kuliah

Halaman tambah ruangan merupakan halaman dimana *admin* dapat menambahkan ruangan baru, halaman ini dapat diakses *admin* ketika *admin* sudah melakukan *login* ke *system*. Tampilan Halaman Tambah Mata Kuliah dapat dilihat pada gambar 5.14.

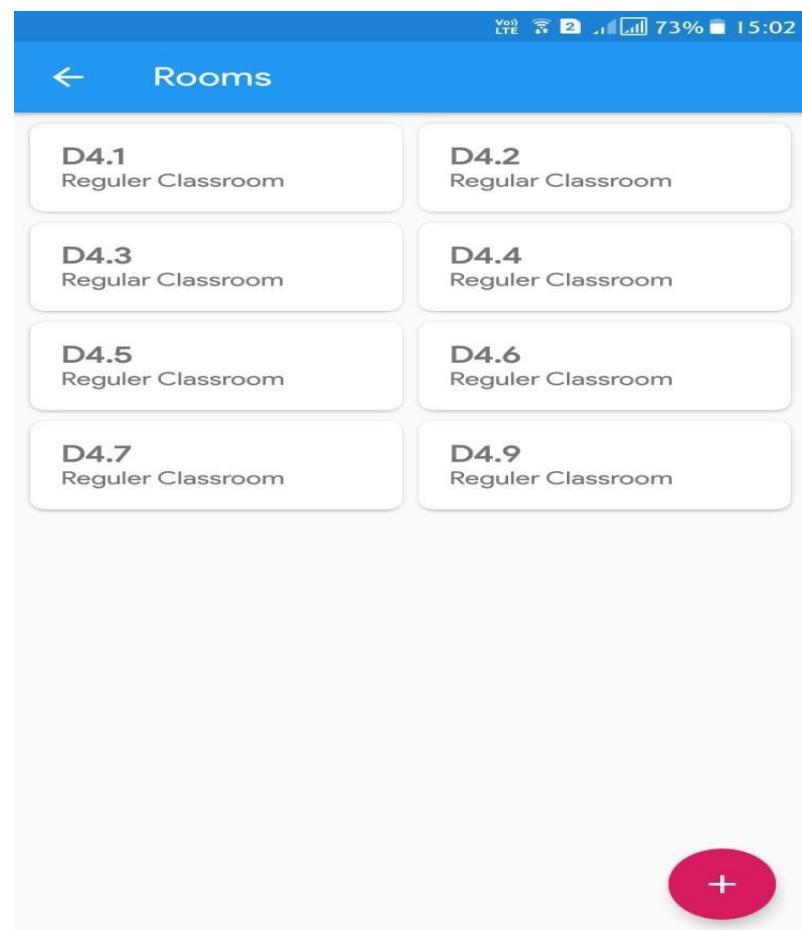


Gambar 5.14 Tampilan Halaman Tambah Mata Kuliah

### 5.1.15 Tampilan Halaman Kelas

Halaman kelas merupakan halaman dimana *admin* dapat melihat kelas yang telah ditambahkan, halaman ini dapat diakses *admin* ketika *admin* sudah melakukan *login* ke *system*.

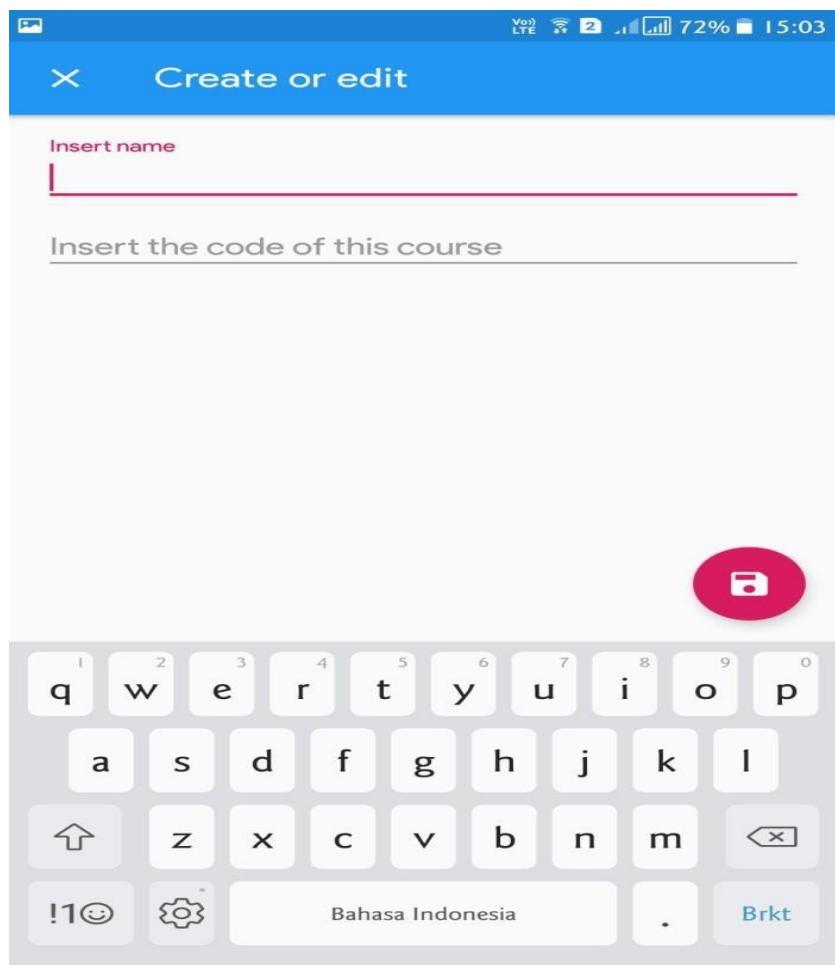
Tampilan Halaman Kelas dapat dilihat pada gambar 5.15.



Gambar 5.15 Tampilan Halaman Kelas

### 5.1.16 Tampilan Halaman Detail Kelas

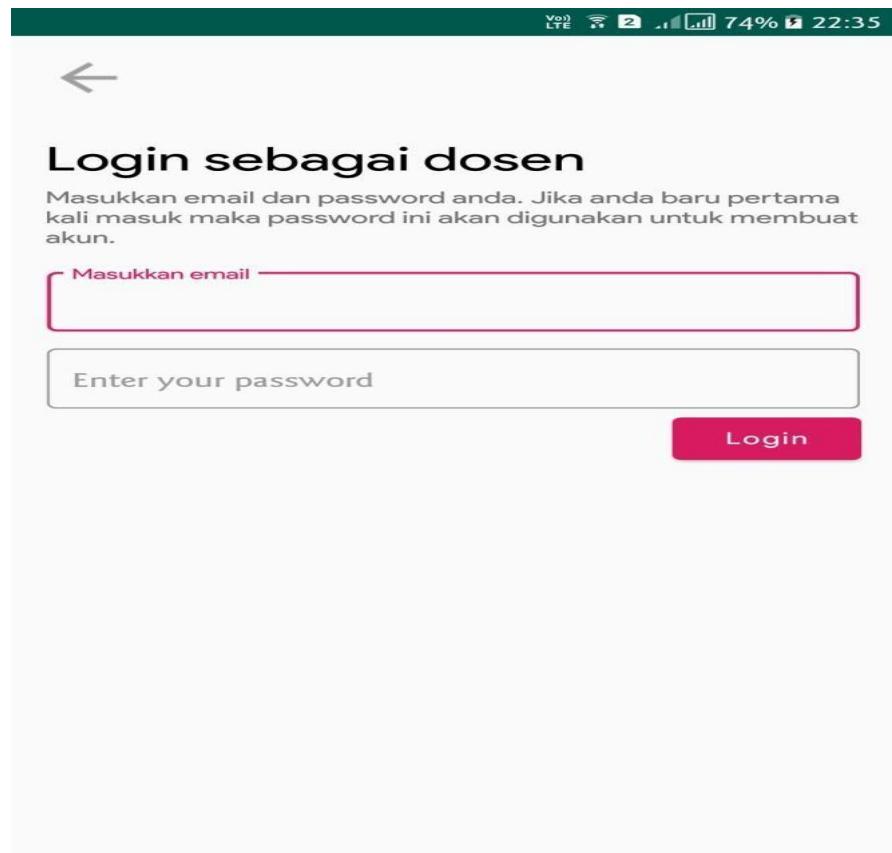
Halaman Detail Kelas merupakan halaman dimana *admin* dapat melihat detail keseluruhan kelas. Tampilan halaman *detail* kelas dapat dilihat pada gambar 5.16 di bawah ini.



Gambar 5.16 Tampilan Halaman detail kelas

### 5.1.17 Tampilan Halaman Login Dosen

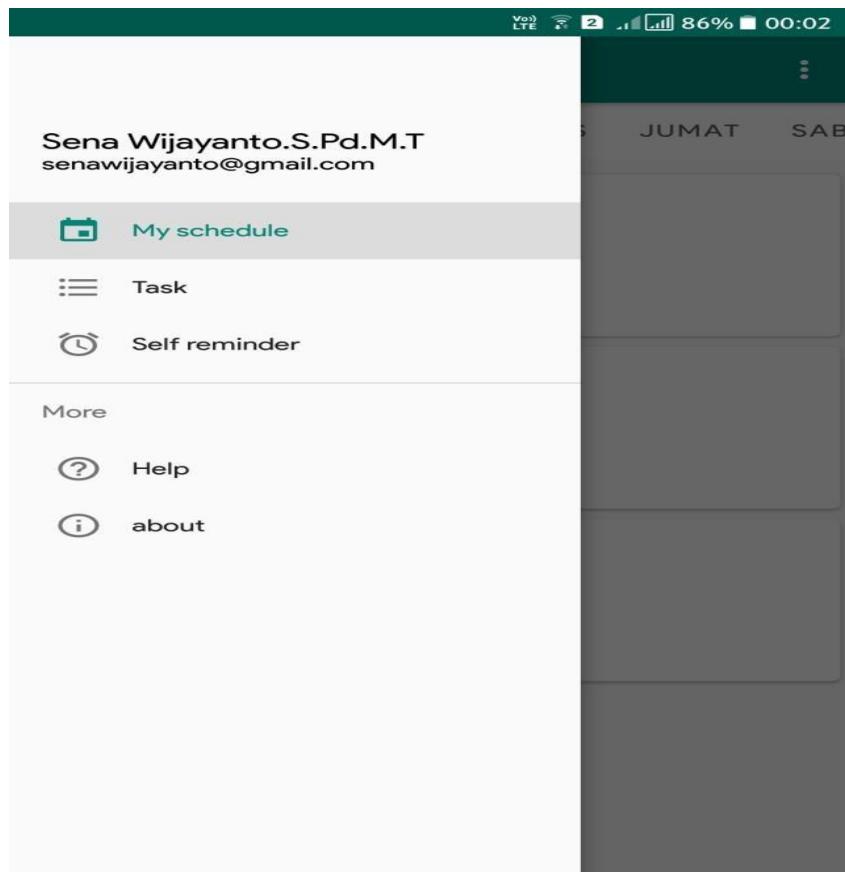
Halaman *login* dosen merupakan halaman yang dapat digunakan untuk *login* dosen untuk masuk ke *system*. Tampilan halaman *login* dosen dapat dilihat dari gambar 5.17.



Gambar 5.17 Tampilan Halaman login dosen

### 5.1.18 Tampilan Halaman Dashboard Dosen

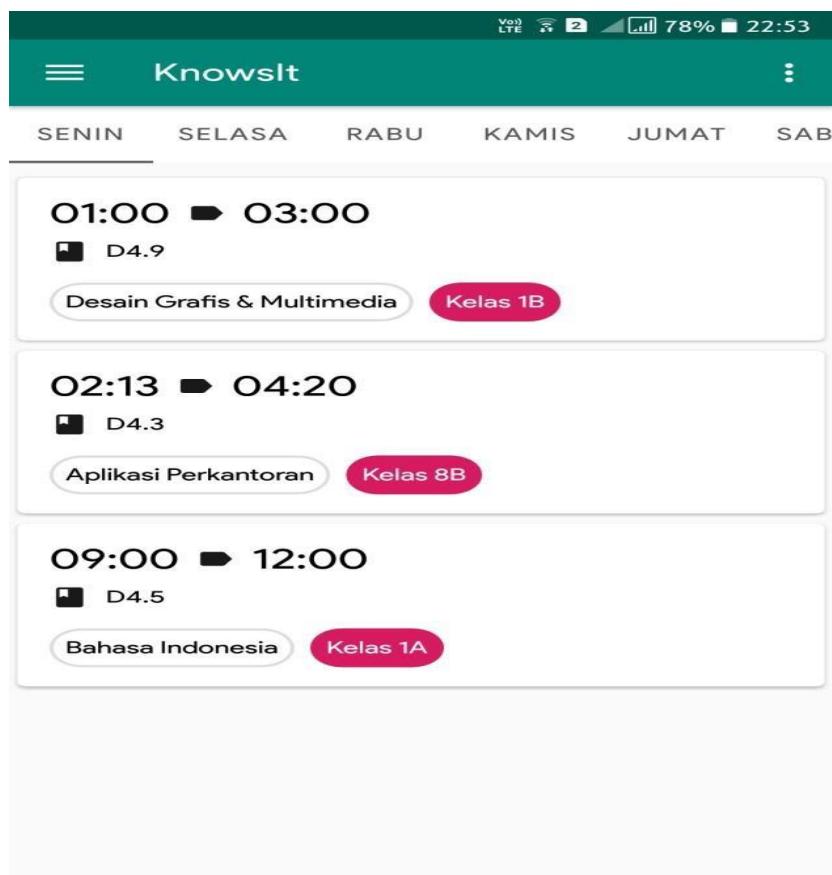
Halaman dashboard Dosen merupakan halaman yang berisi daftar Data My Schedule, Tugas, *Set Reminder*. Halaman *dashboard* merupakan halaman awal yang tampil setelah *admin* melakukan *login* diawal aplikasi. Tampilan halaman *dashboard dosen* dapat dilihat pada gambar 5.18.



Gambar 5.18 Tampilan Halaman Dashboard Dosen

### 5.1.19 Tampilan Halaman My Schedule Dosen

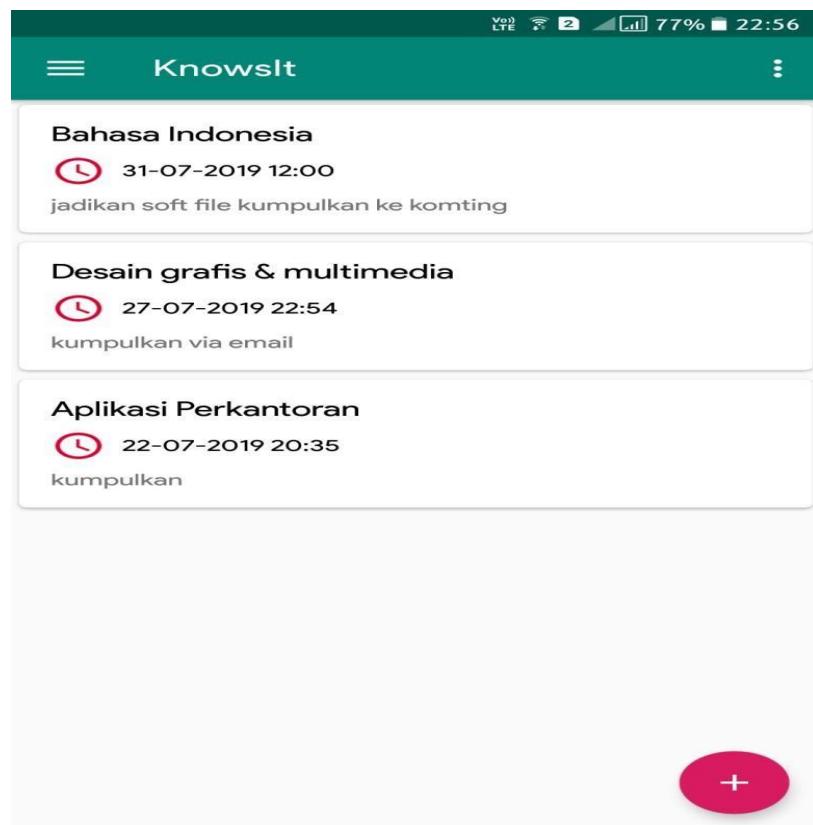
Halaman My Schedule merupakan halaman yang berisi *schedule* atau jadwal kegiatan dosen. Halaman *My Schedule* dapat dilihat setelah dosen melakukan *login* ke *system*. tampilan halaman *My Schedule* dosen dapat dilihat pada gambar 5.19.



Gambar 5.19 Tampilan Halaman *My Schedule* dosen.

### 5.1.20 Tampilan Halaman Task

Halaman *task* merupakan halaman dimana dosen dapat membuat tugas untuk mahasiswa. Halaman task ini dapat diakses ketika dosen sudah melakukan *login* untuk masuk ke *system*. Tampilan halaman task ini dapat dilihat pada gambar 5.20.

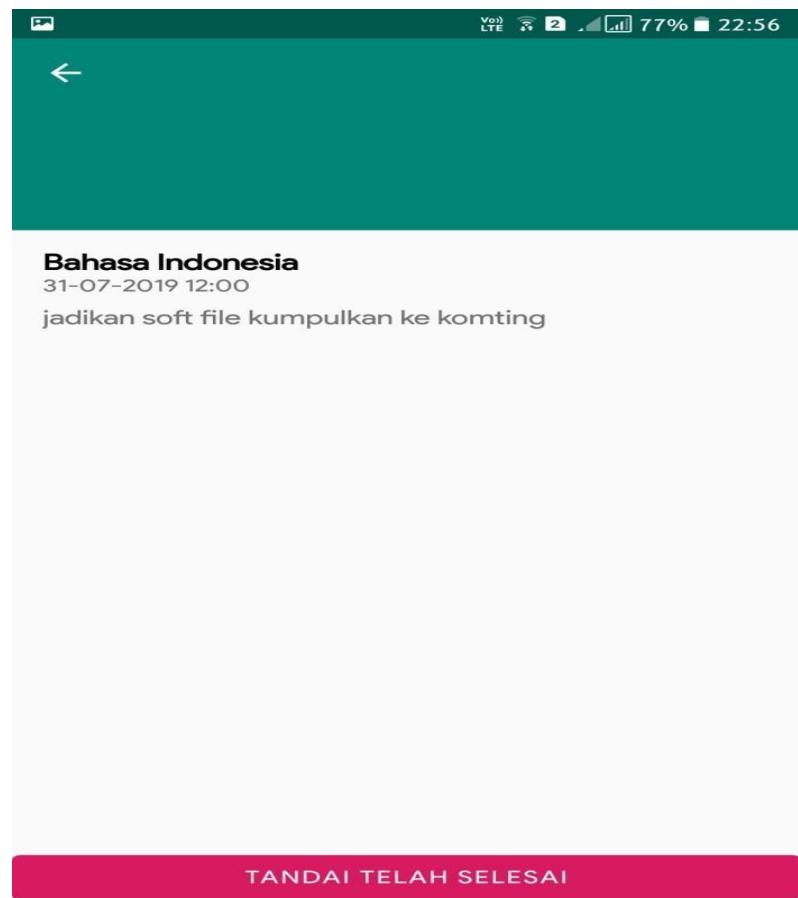


Gambar 5.20 Tampilan Halaman *Task* dosen

### 5.1.21 Tampilan Halaman *Detail Task* dosen

Halaman *detail task* dosen merupakan halaman dimana dosen dapat melihat dan menghapus tugas yang telah dibuat.

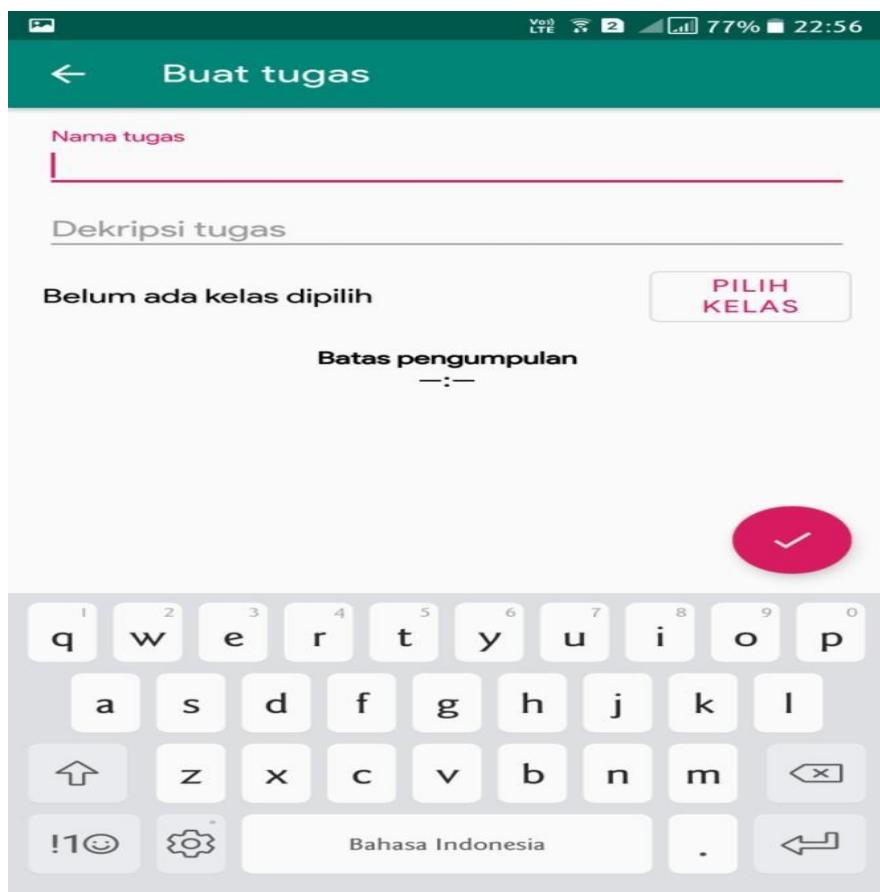
Halaman ini dapat diakses ketika dosen sudah melakukan *login* ke *system*. Tampilan halaman detail task dosen dapat dilihat pada gambar 5.21.



Gambar 5.21 Tampilan Halaman *Detail Task* dosen

### 5.1.23 Tampilan Halaman buat Task dosen

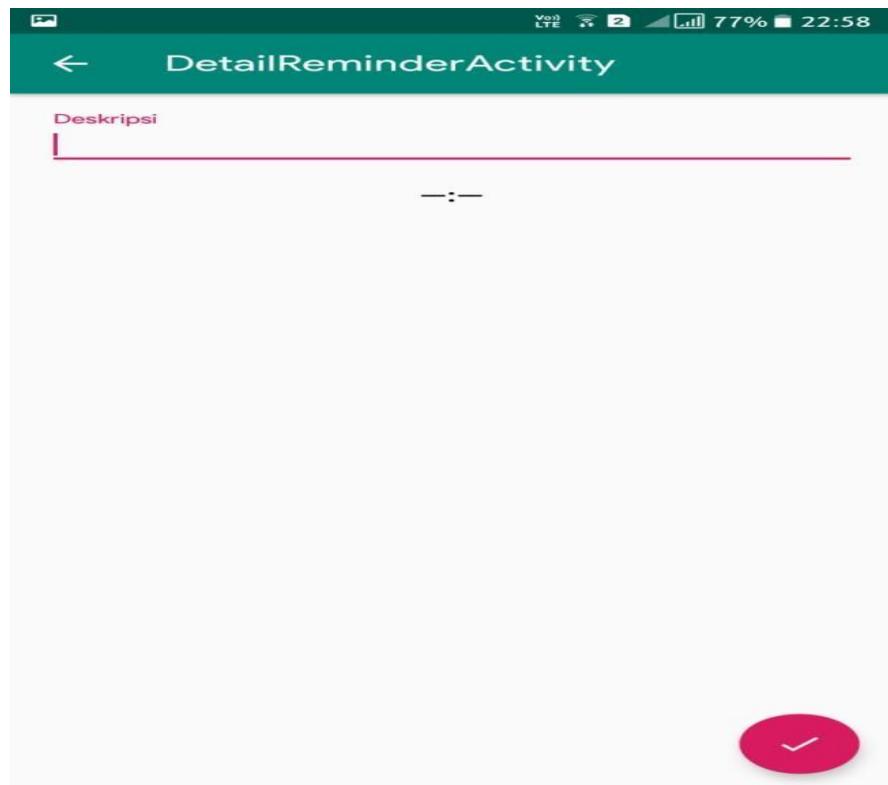
Halaman buat *task* dosen merupakan halaman dimana dosen dapat membuat tugas baru untuk mahasiswa. Halaman ini dapat diakses dosen ketika dosen sudah melakukan *login* ke *system*. Tampilan Halaman Buat *Task* Dosen dapat dilihat pada gambar 5.22.



Gambar 5.22 Tampilan Halaman buat *Task* dosen

### 5.1.24 Tampilan Halaman *self Reminder* Dosen

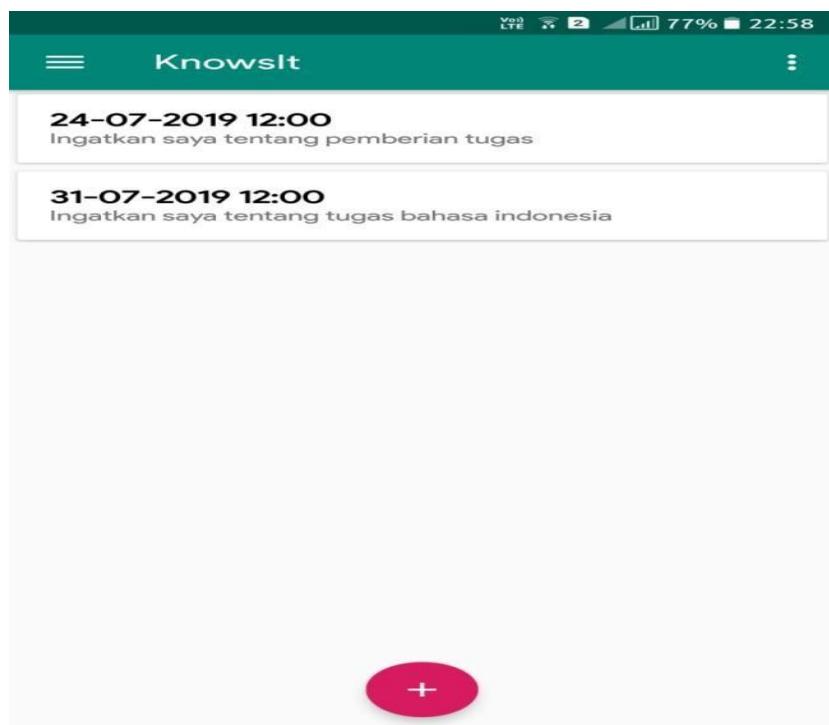
Halaman *self reminder* merupakan halaman yang dapat diakses admin untuk mengatur *reminder* sebagai pengingat. Tampilan halaman *self reminder* ini dapat diakses ketika dosen sudah melakukan *login* ke *system*. Tampilan Halaman *self reminder* dapat dilihat pada gambar 5.23.



Gambar 5.23 Tampilan Halaman *self reminder*

### 5.1.25 Tampilan Halaman *detail self reminder*

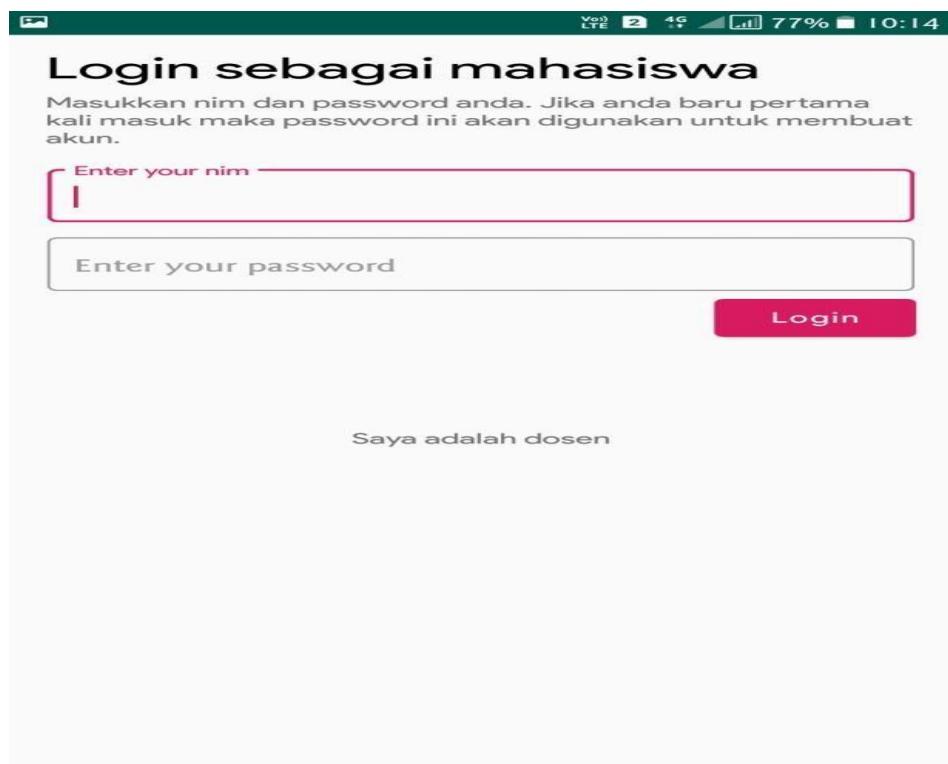
Halaman *detail self reminder* merupakan halaman yang dapat diakses oleh dosen setelah mengaktifkan *reminder*. Tampilan halaman *detail self reminder* dapat diakses dosen ketika dosen sudah melakukan *login* ke *system*. Tampilan halaman detail self reminder dapat dilihat pada gambar 5.24.



Gambar 5.24 Tampilan Halaman *detail selfreminde*

### 5.1.26 Tampilan Halaman *login* mahasiswa

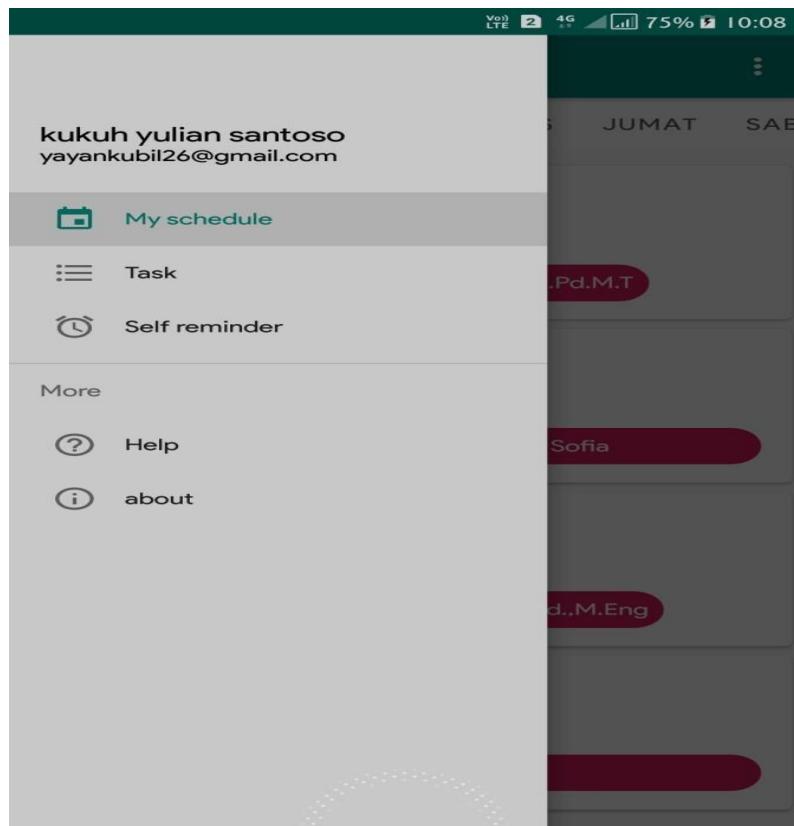
Halaman *login* mahasiswa merupakan halaman yang dapat digunakan untuk *login* mahasiswa untuk masuk ke *system*. Tampilan halaman *login* mahasiswa dapat dilihat dari gambar 5.25.



Gambar 5.25 Tampilan Halaman *login* mahasiswa

### 5.1.27 Tampilan Halaman Dashboard Mahasiswa

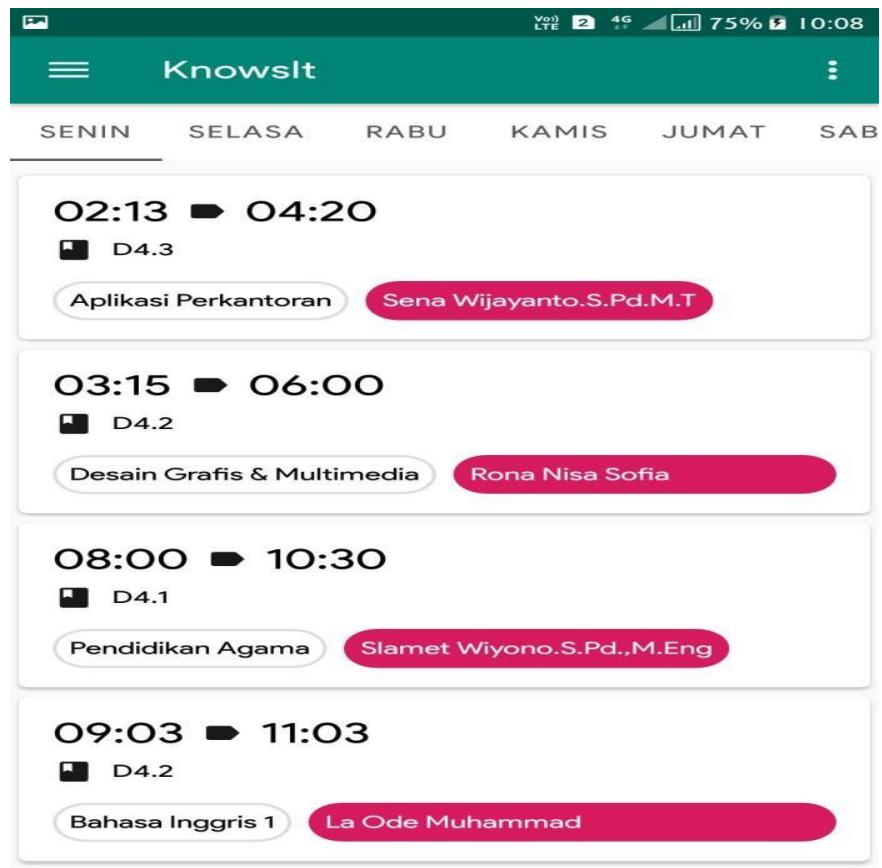
Halaman dashboard Mahasiswa merupakan halaman yang berisi daftar Data My Schedule, Tugas, *Set Reminder*. Halaman *dashboard* merupakan halaman awal yang tampil setelah mahasiswa melakukan *login* diawal aplikasi. Tampilan halaman *dashboard* mahasiswa dapat dilihat pada gambar 5.26.



Gambar 5.26 Tampilan Halaman *dashboard* mahasiswa

### 5.1.28 Tampilan Halaman My Schedule Mahasiswa

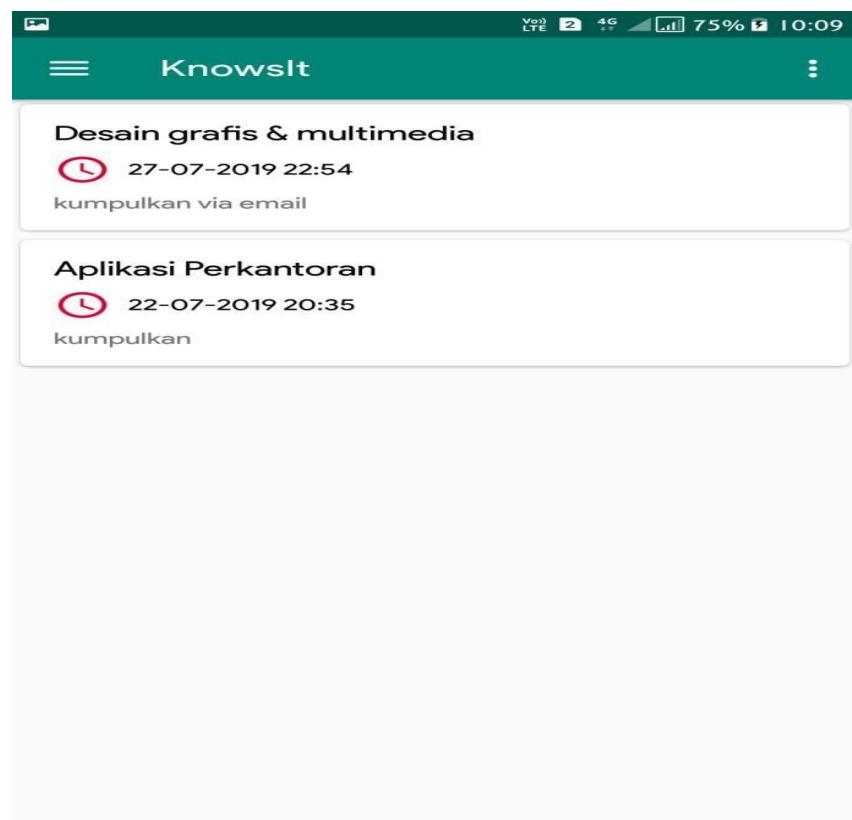
halaman My Schedule merupakan halaman yang berisi *schedule* atau jadwal kegiatan mahasiswa sesuai kelas. Halaman *My Schedule* dapat dilihat setelah mahasiswa melakukan *login* ke *system*. Tampilan halaman *My Schedule* dosen dapat dilihat pada gambar 5.27.



Gambar 5.27 Tampilan Halaman *My Schedule* Mahasiswa

### 5.1.29 Tampilan Halaman Task Mahasiswa

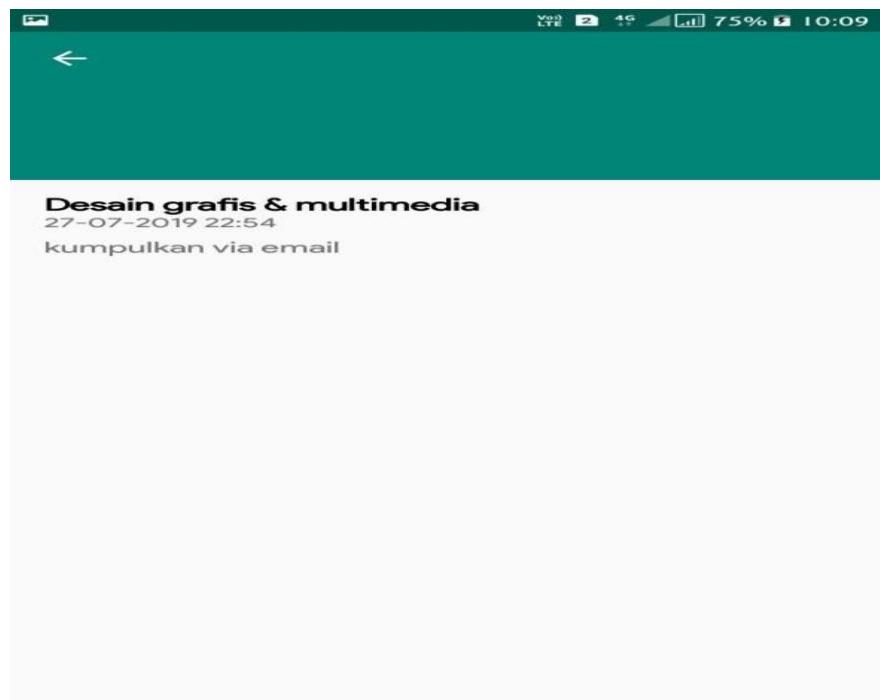
Halaman *task* merupakan halaman dimana mahasiswa dapat menerima informasi tentang tugas. Halaman task ini dapat diakses ketika mahasiswa sudah melakukan *login* untuk masuk ke *system* dan dosen sudah membuat tugas untuk mahasiswa . Tampilan halaman task ini dapat dilihat pada gambar 5.28.



Gambar 5.28 Tampilan Halaman *Task Mahasiswa*

### 5.1.30 Tampilan Halaman *Detail Task Mahasiswa*

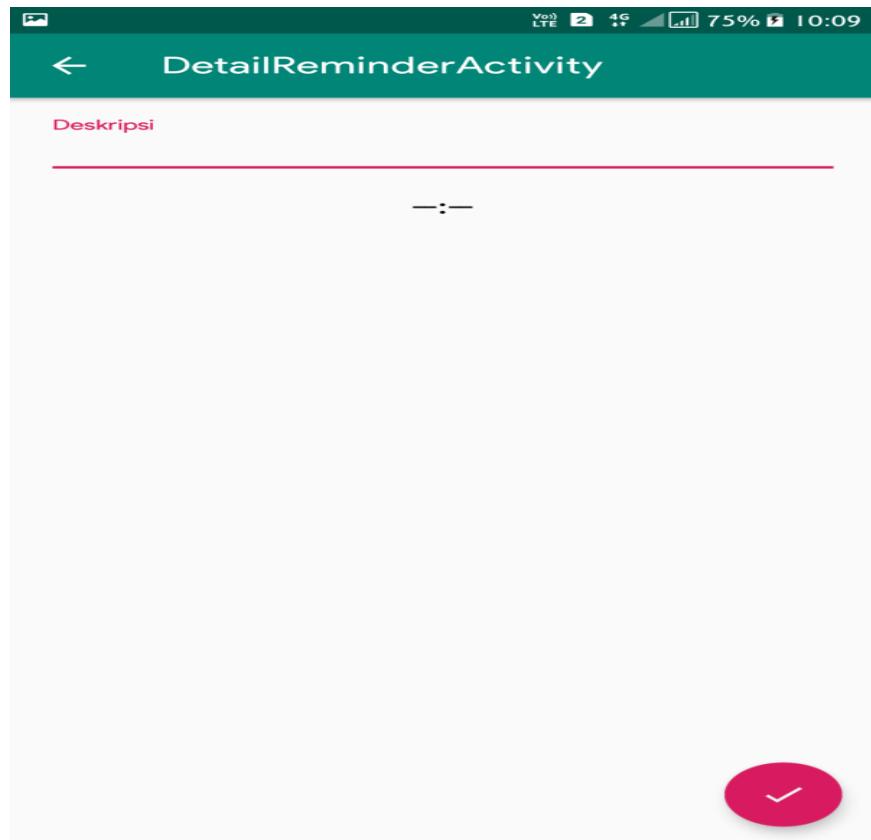
Halaman *detail task* mahasiswa merupakan halaman dimana mahasiswa dapat melihat detail tugas yang telah diberikan oleh dosen. Halaman ini dapat diakses ketika mahasiswa sudah melakukan *login* ke *system*. Tampilan halaman detail task mahasiswa dapat dilihat pada gambar 5.29.



Gambar 5.29 Tampilan Halaman *Detail Task Mahasiswa*

### 5.1.31 Tampilan Halaman *Self Reminder* Mahasiswa

Halaman *self reminder* merupakan halaman yang dapat diakses mahasiswa untuk mengatur *reminder* sebagai pengingat. Tampilan halaman *self reminder* ini dapat diakses ketika mahasiswa sudah melakukan *login* ke *system*. Tampilan Halaman *self reminder* dapat dilihat pada gambar 5.30.



Gambar 5.30 Tampilan Halaman *Self Reminder*

## 5.2 Pengujian *Black-Box Testing*

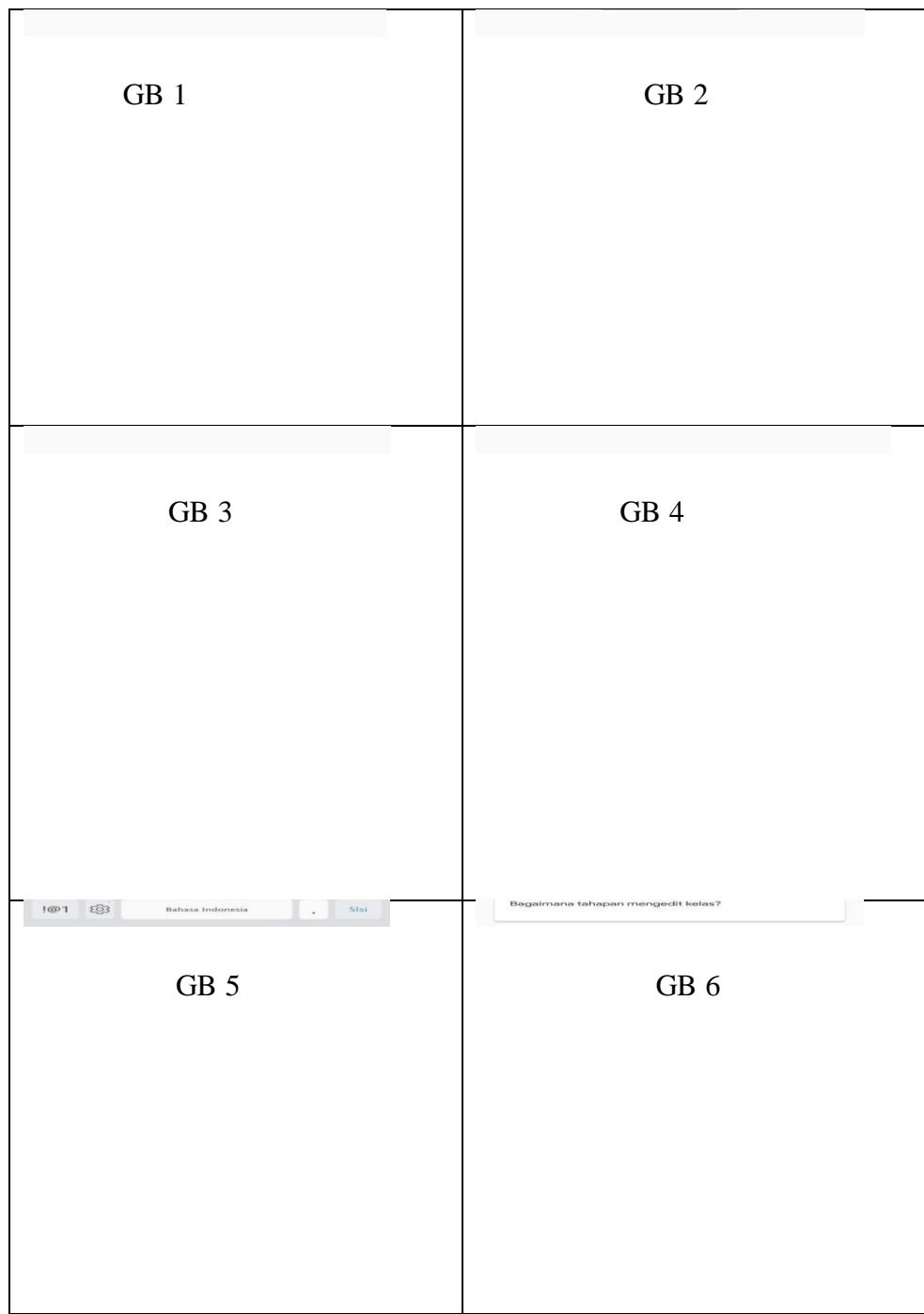
Pengujian *Black-Box Testing* dilakukan dengan mengecek fungsionalitas dan *output* dari aplikasi yang dibuat. Yaitu pengujian terhadap halaman yang mempunyai akun (*input*) dan keluaran (*output*) untuk mengetahui apakah suatu fungsi ketika terjadi proses lalu lintas data sudah sesuai dengan yang diaharapkan atau tidak.

### 5.2.1 Pengujian Halaman *Login Admin*

Pengujian halaman *login admin* adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan *login*. Pengujian halaman *login* dapat dilihat pada tabel 5.1 di bawah ini.

Tabel 5.1 Pengujian Halaman *Login admin*

No	Skenario	Di harapkan	Kesimpulan
1	Mengosongkan email dan password, lalu menekan tombol login (GB1)	System akan menolak akses login dan menampilkan pesan eror (GB2)	Valid
2	Memasukan email dan password dengan data tidak ada pada database (GB3)	System akan menolak akses login dan menampilkan pesan eror (GB4)	Valid
3	Mengisi email dan password dengan data yang sudah ada pada data base (5)	System akan menerima akses login dan menampilkan dashboard (6)	Valid

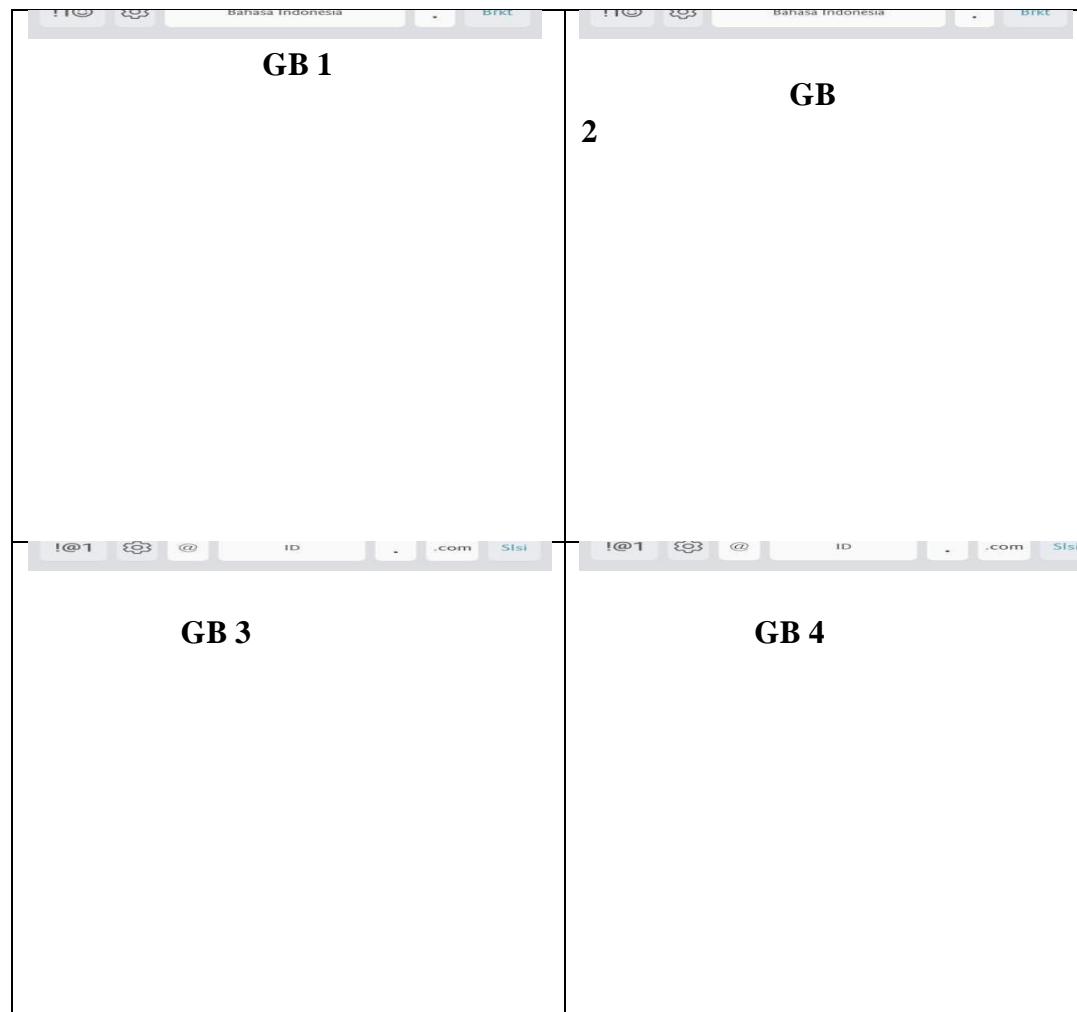


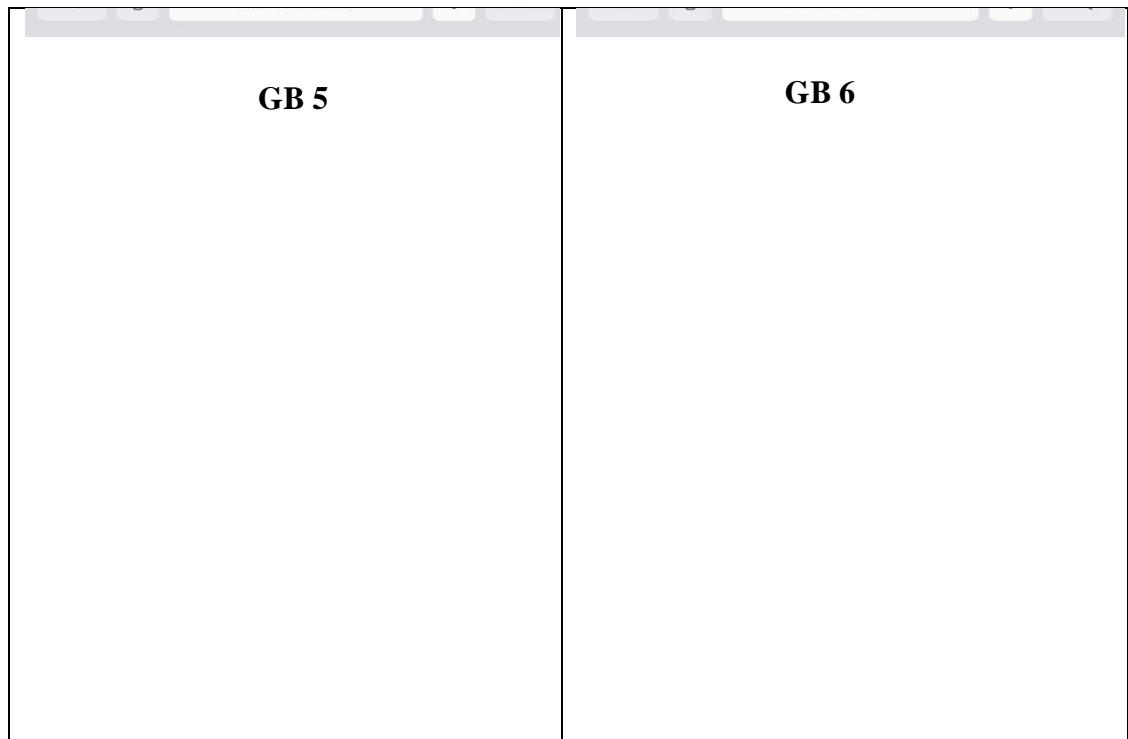
Gambar 5.31 Hasil pengujian halaman *login admin*

### 5.2.2 Pengujian Tambah Data Mahasiswa

Pengujian halaman tambah data mahasiswa adalah pengujian yang dilakukan terhadap tiga kolom masukan yang dilakukan untuk melakukan tambah data mahasiswa. Pengujian halaman *login* dapat dilihat pada tabel 5.2 di bawah ini. Tabel 5.2 Pengujian Halaman Data Mahasiswa

No	Skenario	Di harapkan	Kesimpulan
1	Mengosongkan nama, nim, dan email lalu langsung klik tombol login (GB1)	System akan menolak akses simpan untuk simpan data dan akan menampilkan pesan eror (GB2)	Valid
2	Mengisi nama, email tetapi mengosongkan nim dan menekan tombol login (GB3)	System akan menolak akses untuk simpan data dan akan menampilkan pesan eror (GB4)	Valid
3	Mengisi nama, nim dan email dengan sesuai (GB5)	System akan menerima hak akses simpan data dan akan menampilkan menu (GB)	Valid





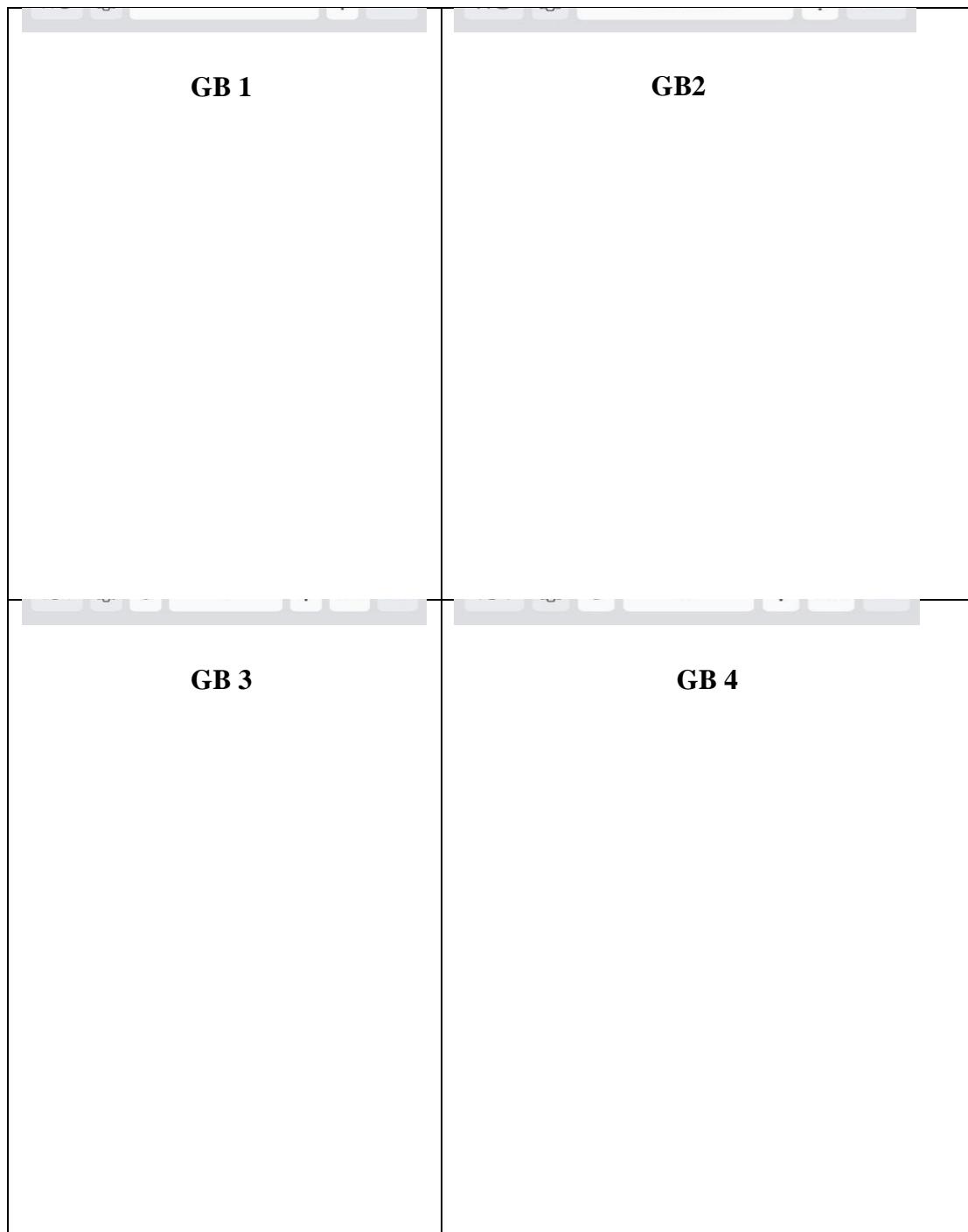
Gambar 5.32 Hasil pengujian halaman tambah data mahasiswa

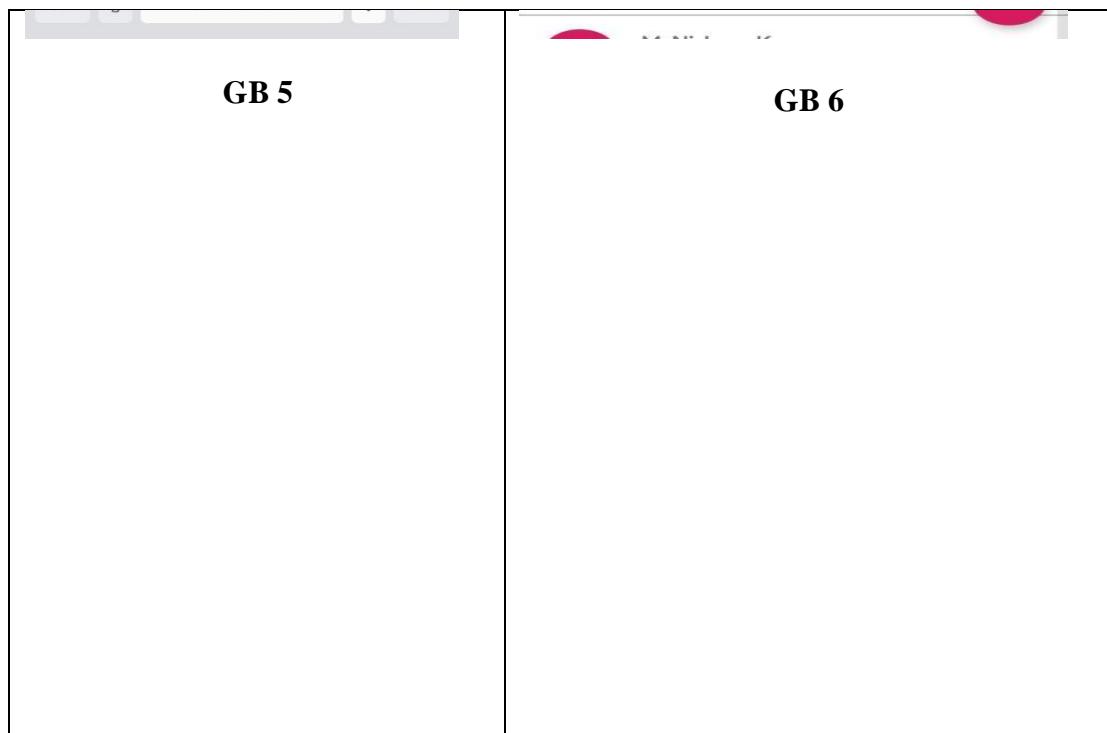
### 5.2.3 Pengujian Tambah Data Dosen

Pengujian halaman tambah data dosen adalah pengujian yang dilakukan terhadap tiga kolom masukan yang dilakukan untuk melakukan tambah data dosen. Pengujian halaman *login* dapat dilihat pada tabel 5.3 di bawah ini.

Tabel 5.3 Pengujian Halaman Tambah Data Dosen

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Mengosongkan nama, nipy, dan email lalu langsung klik tombol simpan (GB1)	System akan menolak akses simpan untuk simpan data dan akan menampilkan pesan eror (GB2)	Valid
2	Mengisi nipy, email dan mengosongkan nama dan menekan tombol simpan (GB3)	System akan menolak akses untuk simpan data ketika tidak mengisi seluruh data yang dibutuhkan dan akan menampilkan pesan eror Valid (GB4)	Valid
3	Mengisi nama, nipy dan email dengan sesuai (GB5)	System akan menerima hak akses dan menyimpan data.	Valid





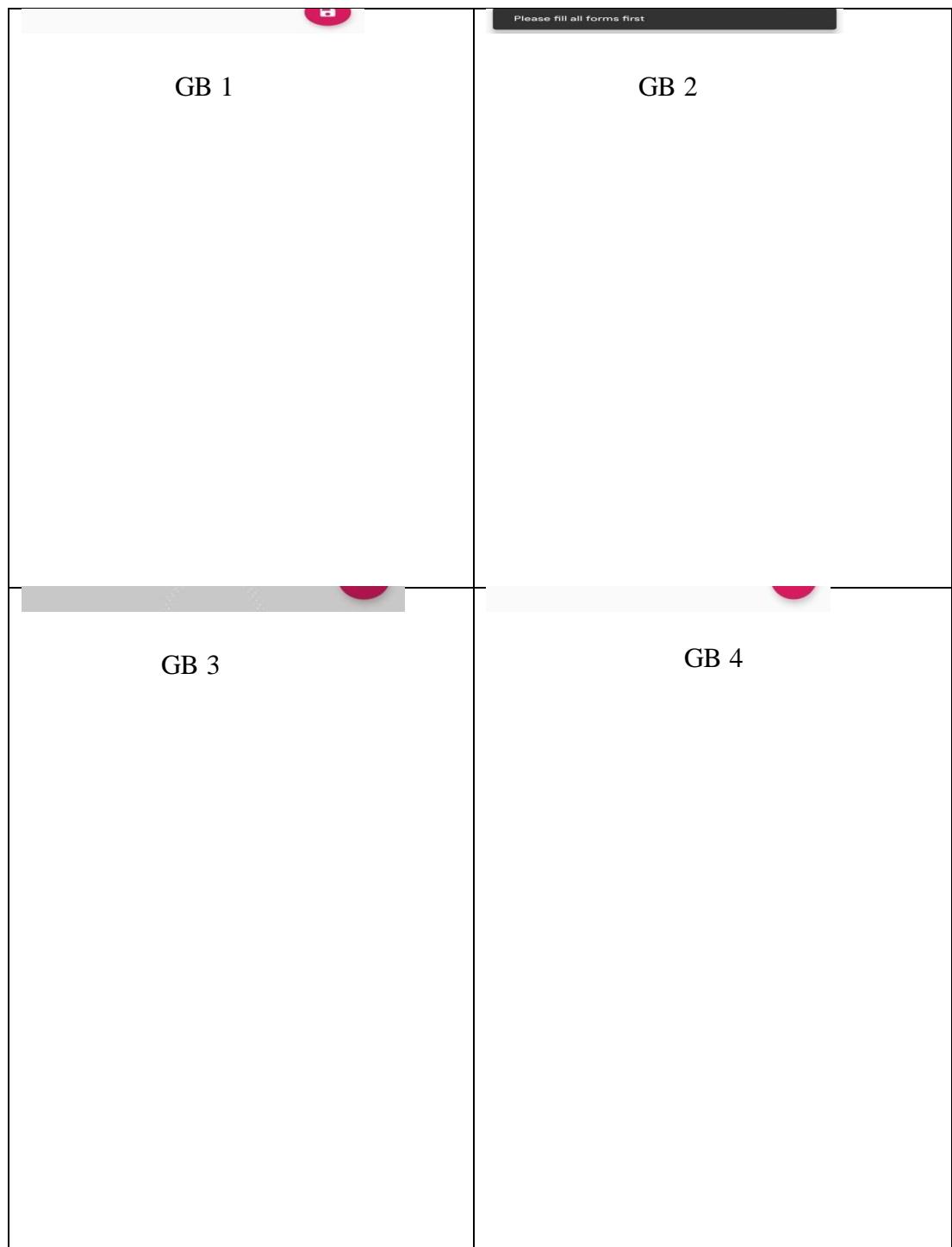
Gambar 5.33 Hasil pengujian halaman tambah data dosen

#### 5.2.4 Pengujian Tambah Ruangan

Pengujian halaman tambah ruangan adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan tambah ruangan. Pengujian halaman *login* dapat dilihat pada tabel 5.4 di bawah ini.

Tabel 5.4 Pengujian Halaman Tambah Ruangan

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Mengosongkan nama dan mengisi deskripsi lalu langsung klik tombol simpan (GB1)	System akan menolak akses simpan untuk simpan data dan akan menampilkan pesan eror (GB2)	Valid
2	Mengisi nama, deskripsi dan menekan tombol simpan (GB3)	System akan menerima akses untuk simpan (GB4)	Valid



Gambar 5.34 hasil pengujian aplikasi tambah ruangan

### 5.2.5 Pengujian Halaman *Login* Mahasiswa

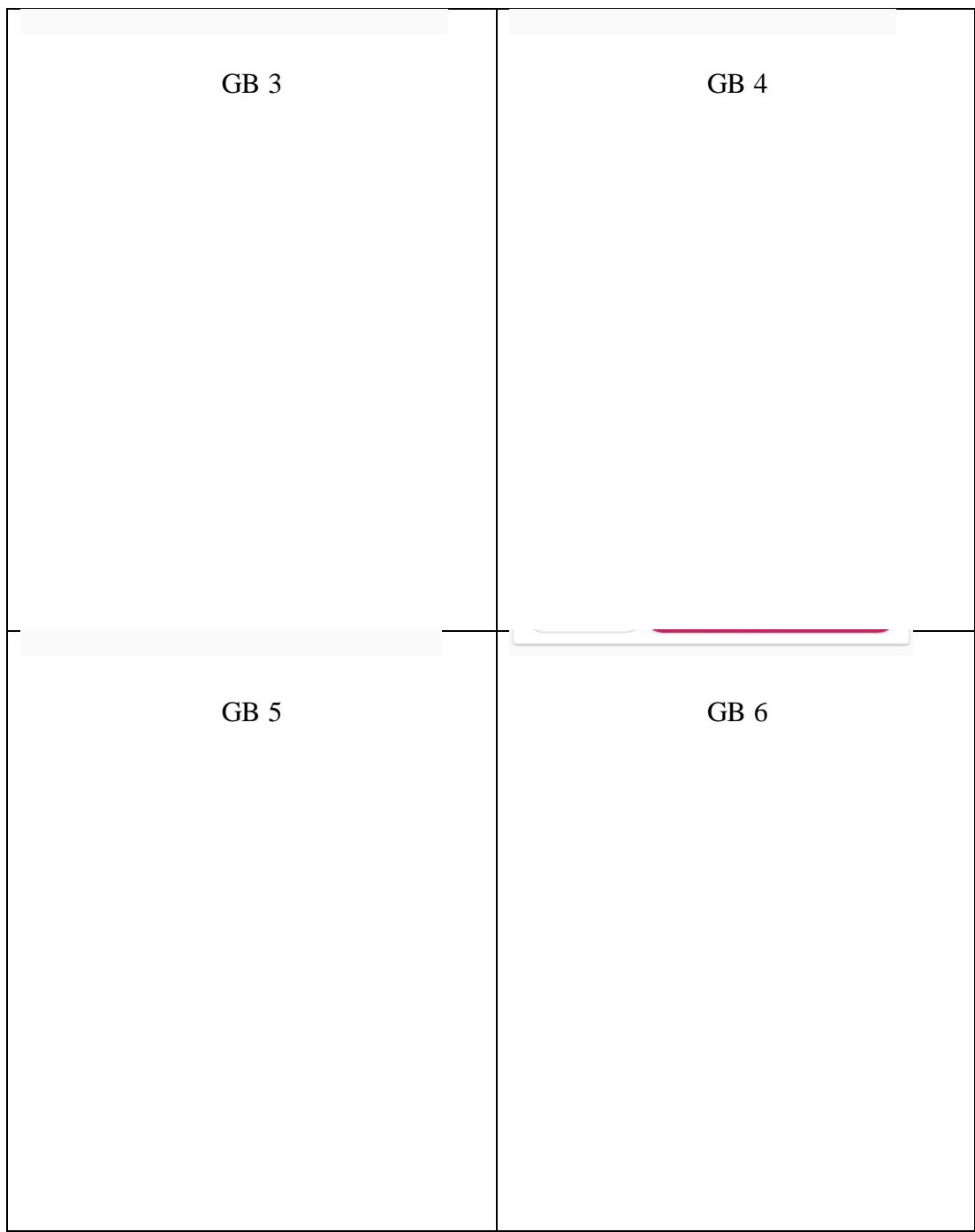
Pengujian halaman *login* Mahasiswa adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan *login*. Pengujian halaman *login* dapat dilihat pada tabel 5.4 di bawah ini.

Tabel 5.4 Pengujian Halaman *Login* Mahasiswa

No	Skenario	Hasiln yang di harapkan	Kesimpulan
1	Mengosongkan nim dan password lalu langsung klik tombol login (GB1)	System akan menolak akses login,menampilkan pesan eror (GB2)	Valid
2	Mengisi nim dan password dengan data tidak ada pada database. (GB 3)	System akan menolak akses login dan menampilkan pesan eror (GB4)	Valid

3	Mengisi nim dan password dengan data sudah ada pada databas (GB 5)	System akan menerima akses login dan menampilkan tampilan dashboard (GB 6)	Valid
---	--	--	-------

GB 1	GB 2
------	------



Gambar 5.35 Hasil pengujian halaman *login* Mahasiswa

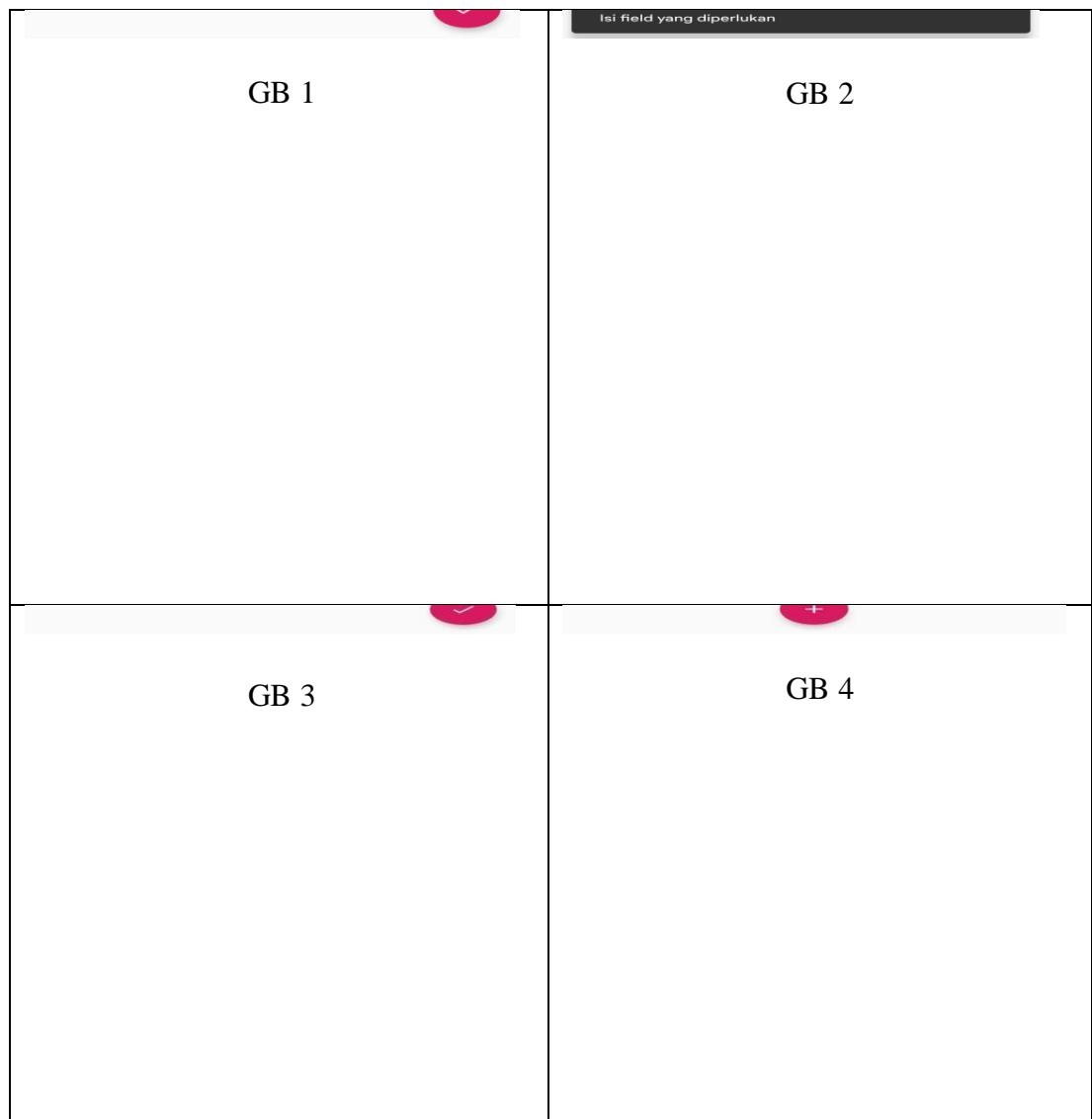
### 5.2.6 Pengujian Halaman *Self Reminder*

Pengujian halaman *Self Reminder* Mahasiswa adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan *Self Reminder*. Pengujian halaman *Self Reminder* dapat dilihat pada tabel 5.5 di bawah ini

Tabel 5.5 Pengujian Halaman *Self Reminder*

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Mengosongkan deskripsi dan <i>setting reminder</i> (GB1)	System akan menolak hak akses simpan data dan menampilkan <i>error Hasil pengujian</i> (GB2)	Valid
2	Mengisi deskripsi dan <i>setting reminder</i> dengan sesuai (GB3)	System akan menerima hak akses untuk simpan data dan menampilkan	Valid

		eror (GB4)	
--	--	------------	--



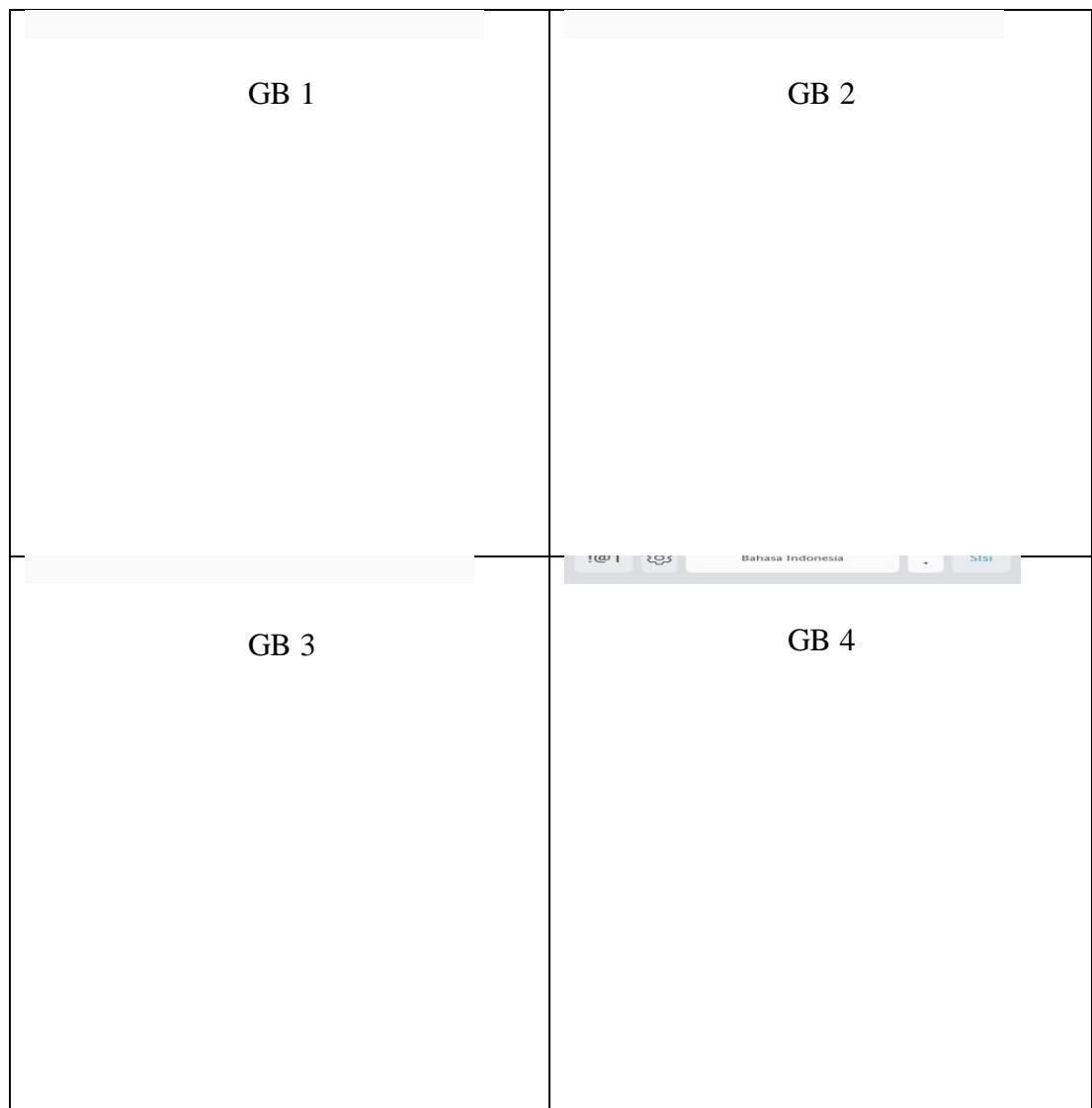
Gambar 5.36 Hasil pengujian Self Reminder

### 5.2.7 Pengujian Halaman *Login* Dosen

Pengujian halaman *login* Dosen adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan *login*. Pengujian halaman *login* dapat dilihat pada tabel 5.4 di bawah ini.

Tabel 5.4 Pengujian Halaman *Login* Dosen

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Mengosongkan email dan password lalu langsung klik tombol login (GB1)	System akan menolak akses login dan menampilkan pesan eror (GB2)	Valid
2	Mengisi nim dan password dengan data tidak ada pada database. (GB 3)	System akan menolak akses login dan menampilkan pesan eror (GB4)	Valid



GB 5	GB 6
------	------

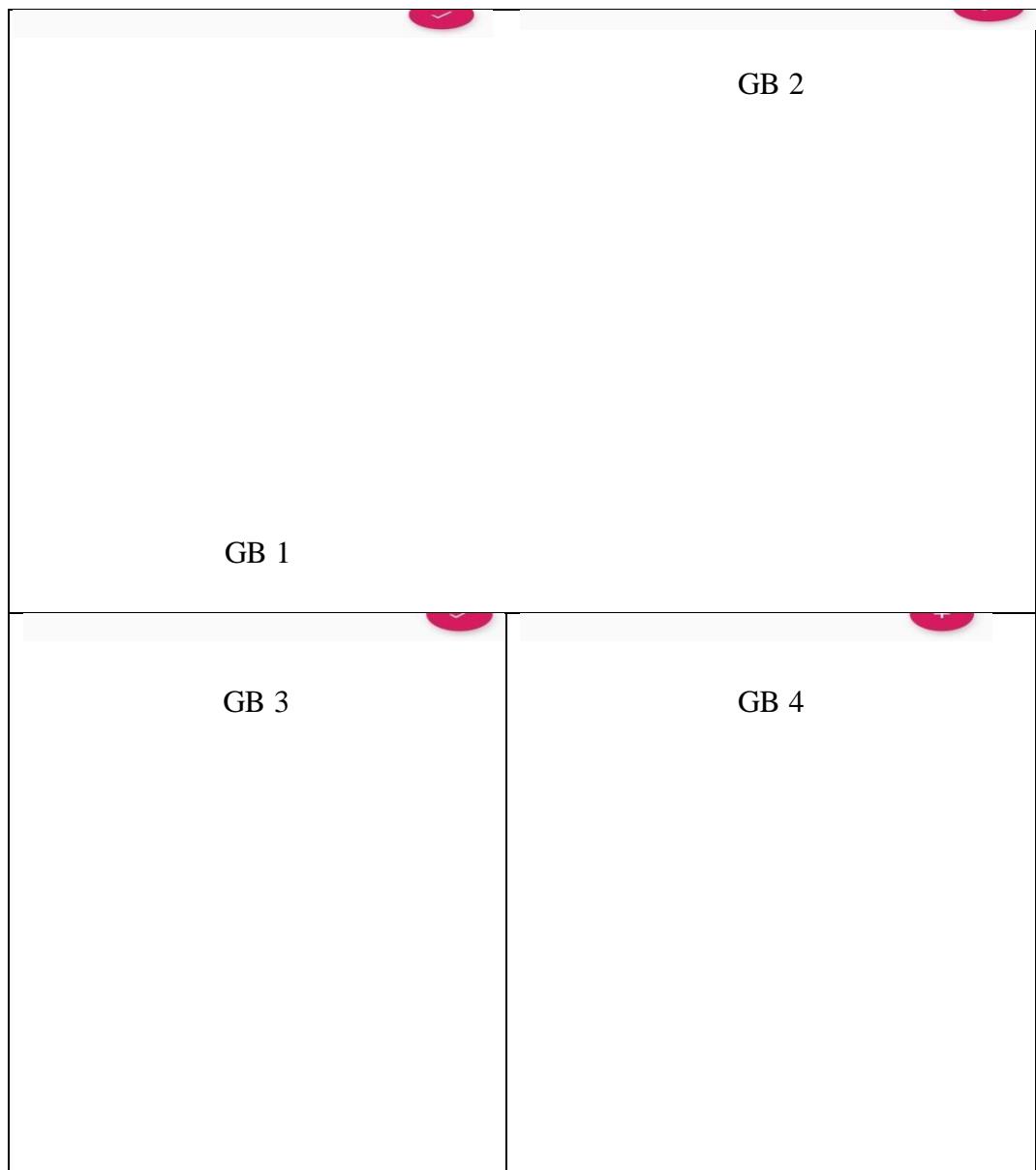
Gambar 5.37 Hasil pengujian halaman *login* Dosen

#### 5.2.8 Pengujian Halaman Tugas Dosen

Pengujian halaman Tugas Dosen adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan Pembuatan Tugas. Pengujian halaman Tugas dapat dilihat pada tabel 5.5.

Tabel 5.5 Pengujian Halaman Tugas Dosen

<b>No</b>	<b>Skenario</b>	<b>Hasil yang di harapkan</b>	<b>Kesimpulan</b>
1	Mengosongkan nama tugas, deskripsi dan tidak pilih kelas, langsung pilih simpan (GB1)	System akan menolak akses simpan data dan menampilkan pesan eror (GB2)	Valid
2	Mengisi nama tugas, deskripsi dan pilih kelas, kemudian pilih simpan (GB3)	System akan menerima akses untuk simpan data dan menampilkan hasil(GB4)	Valid



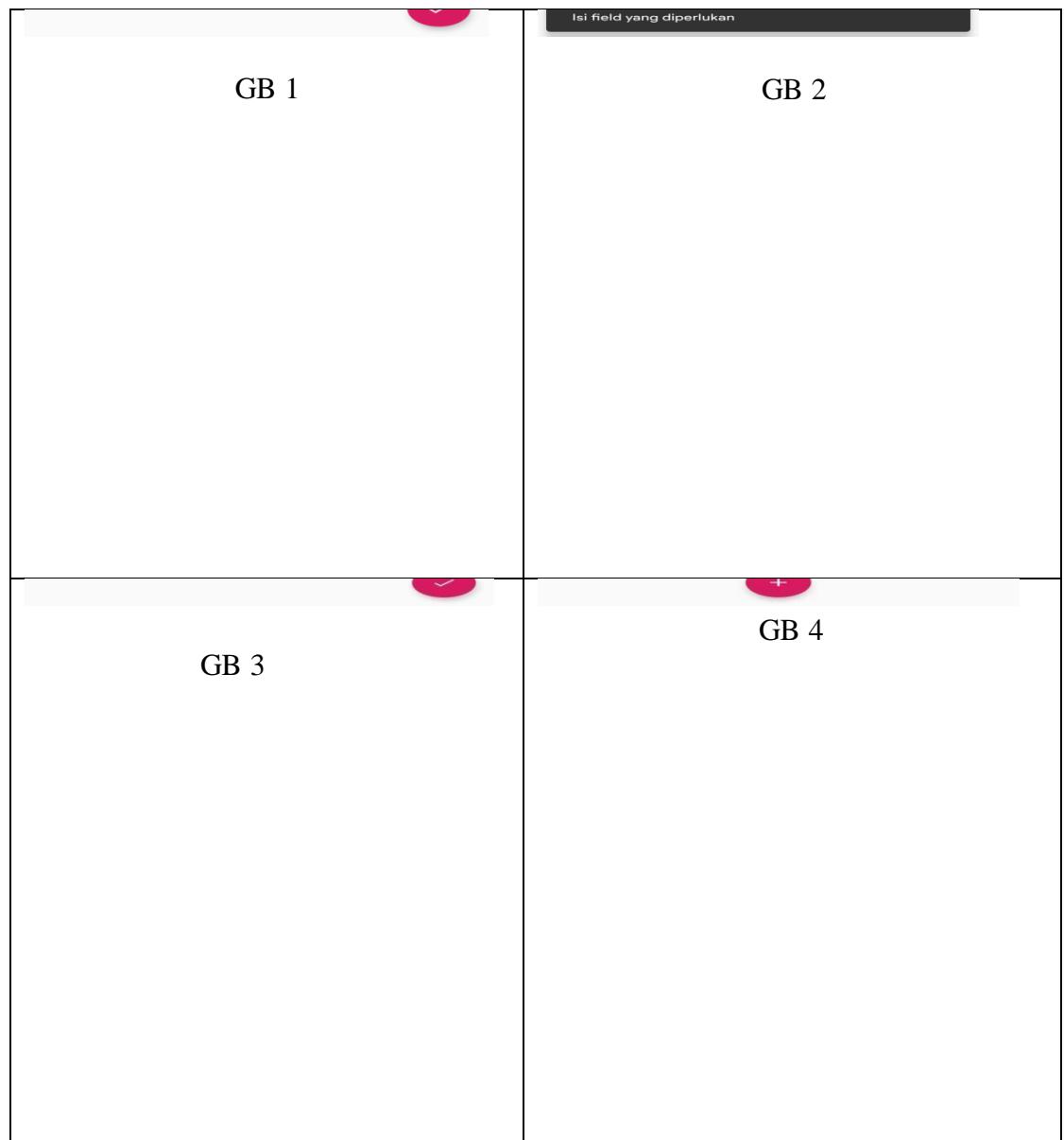
Gambar 5.38 Hasil pengujian halaman Tugas Dose

### 5.2.9 Pengujian Halaman *Self Reminder* Dosen

Pengujian halaman *Self Reminder* Dosen adalah pengujian yang dilakukan terhadap dua kolom masukan yang dilakukan untuk melakukan *Self Reminder*. Pengujian halaman *Self Reminder* dapat dilihat pada tabel 5.6 di bawah ini.

Tabel 5.6 Pengujian Halaman *SelfReminder* Dosen

No	Skenario	Hasil yang diharapkan	Kesimpulan
1	Mengosongkan deskripsi dan jadwal reminder. Kemudian menekan tombol simpan (GB1)	System akan menolak hak akses untuk menyimpan data dan menampilkan eror (GB2)	Valid
2	Mengisi deskripsi dan jadwal reminer dengan sesuai kemudian menekan tombol simpan (GB3)	System akan menerima hak akses Simpan data dan menampilkan detail (GB4)	Valid



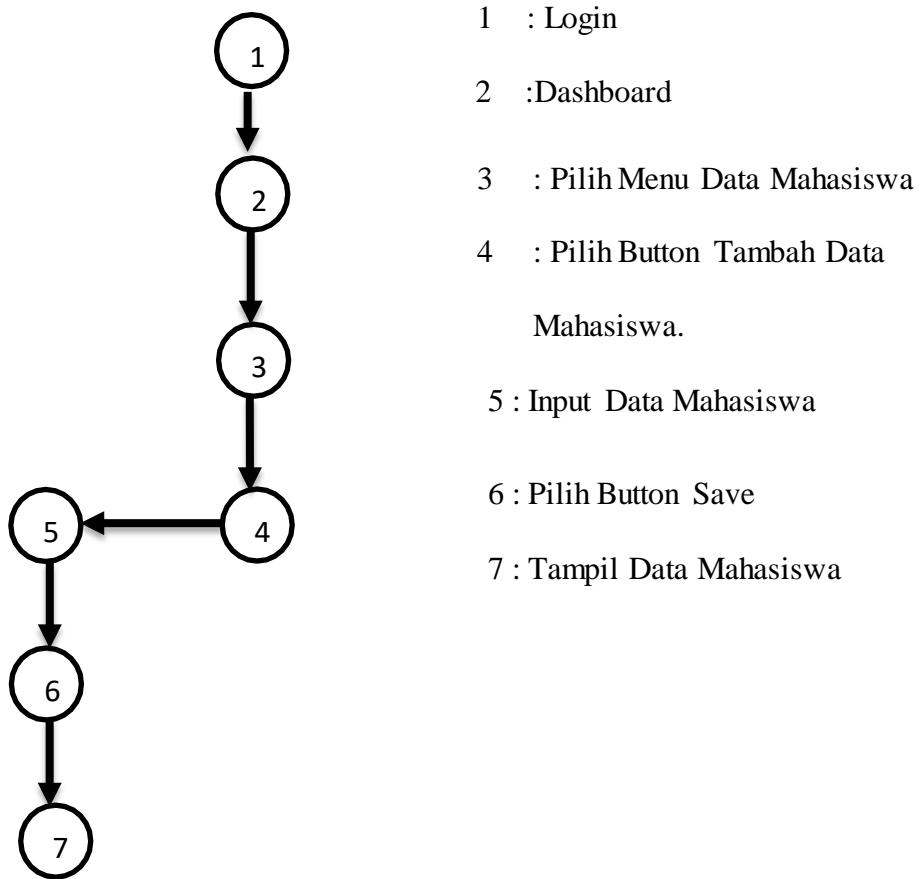
Gambar 5.39Hasil pengujian halaman *SelfReminder Dosen*

### **5.2.10 Pengujian Sistem *White Box Testing***

Pengujian *white box* dilakukan dengan metode *basis path testing*. Pengujian menggunakan metode *basis path testing* dilakukan dengan mengeksekusi seluruh jalur independen aplikasi minimal satu kali. Jalur independen program tersebut ditentukan melalui analisa pada notasi diagram alir. Jumlah jalur independen ditentukan melalui metode perhitungan *Cyclomatic Complexity*.

### 1. Pengujian Tambah Data Mahasiswa

#### a. Menentukan Notasi Diagram Alir



Gambar 5.40 Notasi Diagram Alir Proses tambah data mahasiswa

a. Jalur 1 : 1,2,3,4,5,6,7

Pada jalur ini *user* melakukan proses login dengan username dan password (node 1). System akan mengarahkan pengguna ke menu dashboard. Selanjutnya pilih menu data mahasiswa (3). System akan mengarahkan pengguna jika ingin menambah data

mahasiswa(node4). System akan mengarahkan pengguna untuk memasukan data mahasiswa baru(node5). Setelah pengguna telah memasukan data mahasiswa baru pengguna akan diarahkan untuk menekah tombol simpan(6). Setelah pengguna melakukan proses simpan data mahasiswa baru system akan menampilkan hasil data yang telah diinputkan(node7).

b. Menghitung *Cyclomatic Complexity*

Berdasarkan hasil analisis terhadap notasi diagram alir aplikasi diketahui jumlah *Edge* (E) yaitu 7 dan jumlah *Node* (N) yaitu 7.

Jumlah *cyclomatic complexity* adalah :

$$V(G) = E - N + 2 \quad V(G) = 6 - 7 + 2$$

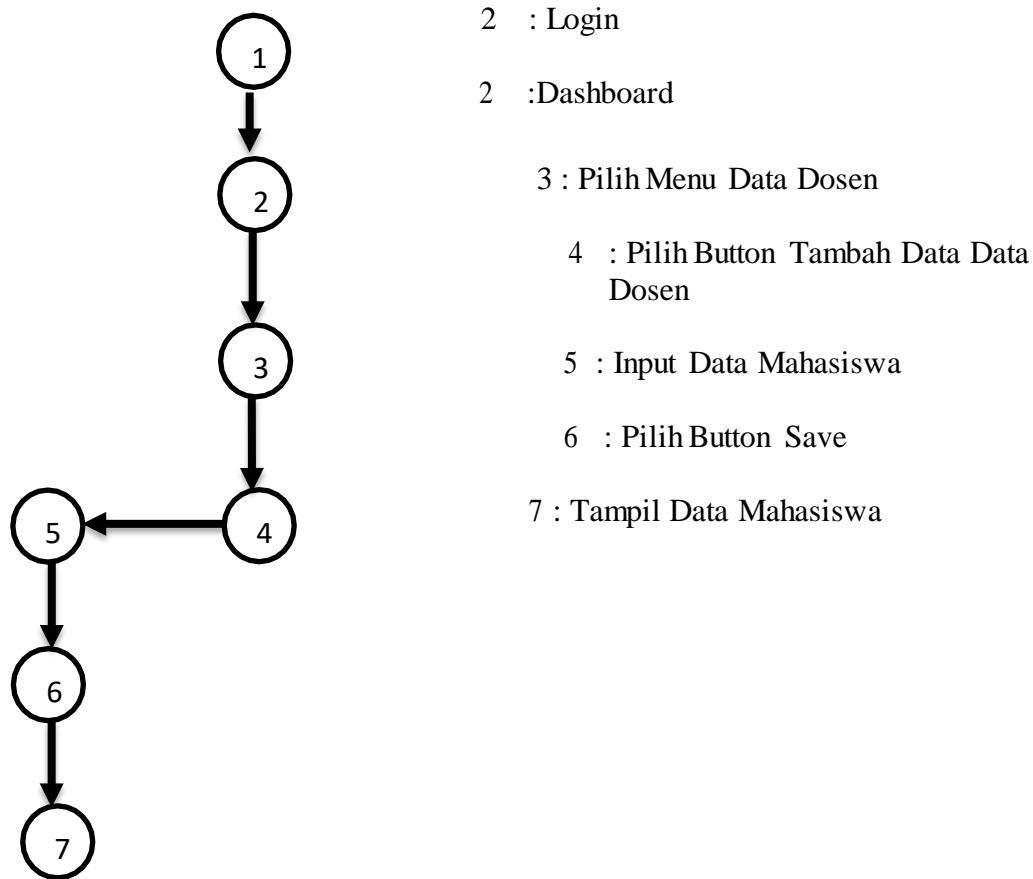
$$V(G) = 1$$

c. Pengujian

1. Dari jalur 1 sistem dapat melakukan login dan system dapat mengarahkan ke menu dashboard selanjutnya diarahkan ke menu tambah data mahasiswa baru dan system dapat menyimpan data mahasiswa baru.

## 2. Pengujian Tambah Data Dosen

### a. Menentukan Notasi Diagram Alir



Gambar 5.41 Notasi Diagram Alir Proses tambah data dosen

b. Jalur 1 : 1,2,3,4,5,6,7

Pada jalur ini *user* melakukan proses login dengan username dan password (node 1). System akan mengarahkan pengguna ke menu dashboard. Selanjutnya pilih menu data dosen (3). System akan

mengarahkan pengguna jika ingin menambah data dosen(node4). System akan mengarahkan pengguna untuk memasukan data dosen baru(node5). Setelah pengguna telah memasukan data dosenbaru pengguna akan diarahkan untuk menekah tombol simpan(6). Setelah pengguna melakukan proses simpan data dosen baru system akan menampilkan hasil data yang telah diinputkan(node7).

c. Menghitung *Cyclomatic Complexity*

Berdasarkan hasil analisis terhadap notasi diagram alir aplikasi diketahui jumlah *Edge* (E) yaitu 7 dan jumlah *Node* (N) yaitu 7. Jumlah *cyclomatic complexity* adalah :

$$V(G) = E - N + 2 \quad V(G) = 6 - 7 + 2$$

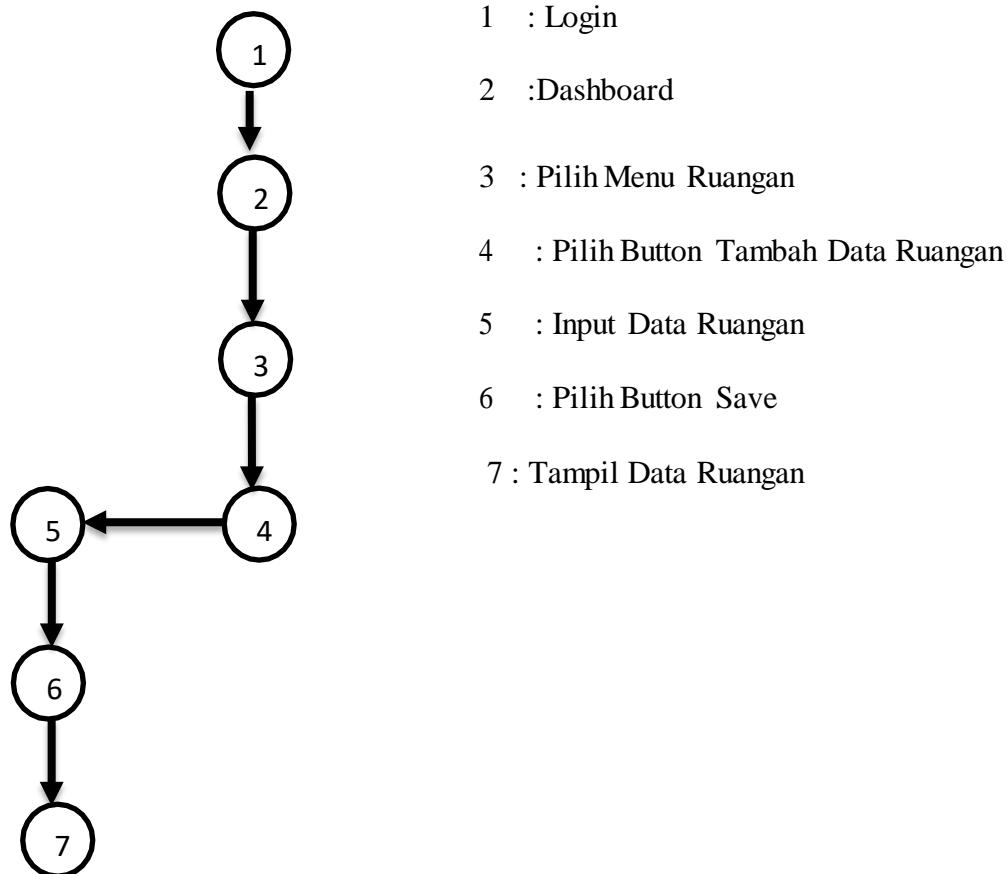
$$V(G) = 1$$

d. Pengujian

Dari jalur 1 sistem dapat melakukan login dan system dapat mengarahkan ke menu dashboard selanjutnya diarahkan ke menu tambah data mahasiswa baru dan system dapat menyimpan data mahasiswa baru.

### 3. Pengujian Tambah Data Ruangan

#### a. Menentukan Notasi Diagram Alir



Gambar 5.42 Notasi Diagram Alir Proses tambah data Ruangan

#### b. Jalur 1 : 1,2,3,4,5,6,7

Pada jalur ini *user* melakukan proses login dengan username dan password (node 1). System akan mengarahkan pengguna ke menu dashboard. Selanjutnya pilih menu data ruangan (3). System akan mengarahkan pengguna jika ingin menambah data ruangan(node4)

System akan mengarahkan pengguna untuk memasukan data ruangan baru(node5). Setelah pengguna telah memasukan data ruangan baru pengguna akan diarahkan untuk menekah tombol simpan(6). Setelah pengguna melakukan proses simpan data ruangan baru system akan menampilkan hasil data yang telah diinputkan(node7).

c. Menghitung *Cyclomatic Complexity*

Berdasarkan hasil analisis terhadap notasi diagram alir aplikasi diketahui jumlah *Edge* (E) yaitu 7 dan jumlah *Node* (N) yaitu 7. Jumlah *cyclomatic complexity* adalah :

$$V(G) = E - N + 2 \quad V(G) = 6 - 7 + 2$$

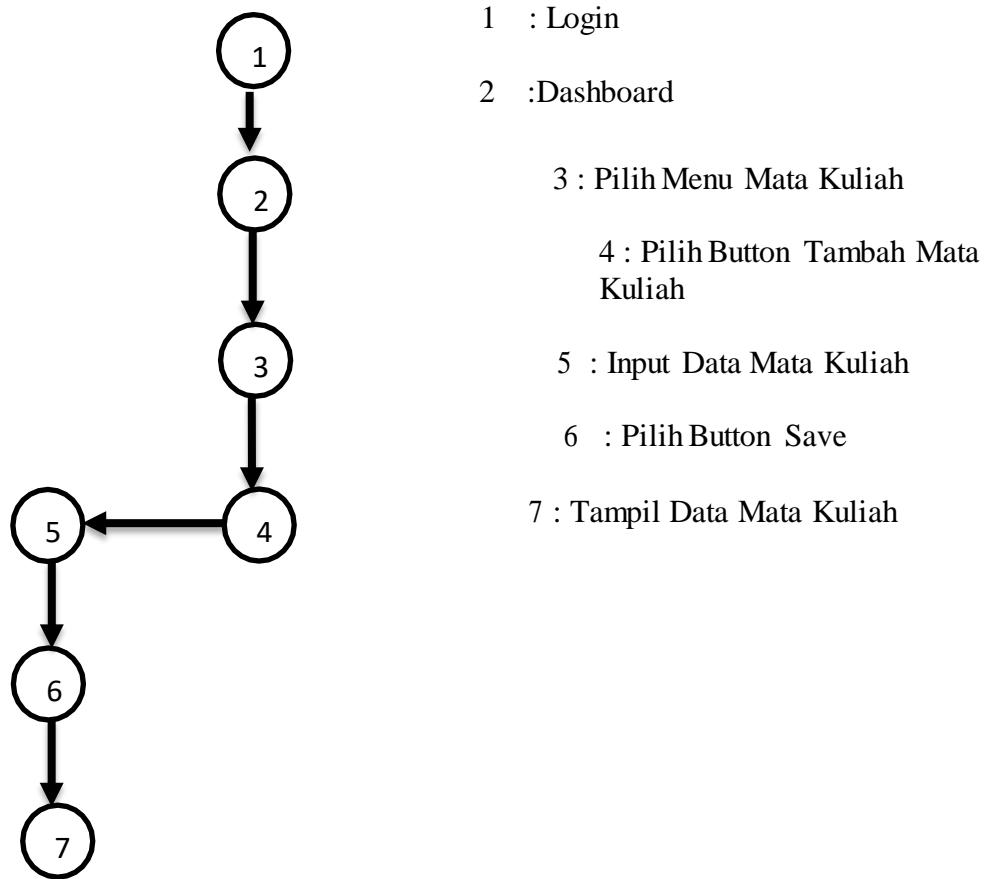
$$V(G) = 1$$

d. Pengujian

Dari jalur 1 sistem dapat melakukan login dan system dapat mengarahkan ke menu dashboard selanjutnya diarahkan ke menu ruangan dan system akan menampilkan detail ruangan selanjutnya system akan mengarahkan ke menu tambah ruangan yang selanjutnya system dapat menyimpan data ruangan dan menampilkannya.

4. Pengujian Tambah Data Mata Kuliah

a. Menentukan Notasi  
Diagram Alir



Gambar 5.43 Notasi Diagram Alir Proses tambah data mata kuliah

b. Jalur 1 : 1,2,3,4,5,6,7

Pada jalur ini *user* melakukan proses login dengan username dan password (node 1). System akan mengarahkan pengguna ke menu dashboard. Selanjutnya pilih menu mata kuliah (3). System akan mengarahkan pengguna jika ingin menambah data

(node4). System akan mengarahkan pengguna untuk memasukan data mata kuliah baru (node5). Setelah pengguna telah memasukan data mata kuliah baru pengguna akan diarahkan untuk menekan tombol simpan(6). Setelah pengguna melakukan proses simpan data mahasiswa baru system akan menampilkan hasil data yang telah diinputkan(node7).

c. Menghitung *Cyclomatic Complexity*

Berdasarkan hasil analisis terhadap notasi diagram alir aplikasi diketahui jumlah *Edge* (E) yaitu 7 dan jumlah *Node* (N) yaitu 7. Jumlah *cyclomatic complexity* adalah :

$$V(G) = E - N + 2 \quad V(G) = 6 - 7 + 2$$

$$V(G) = 1$$

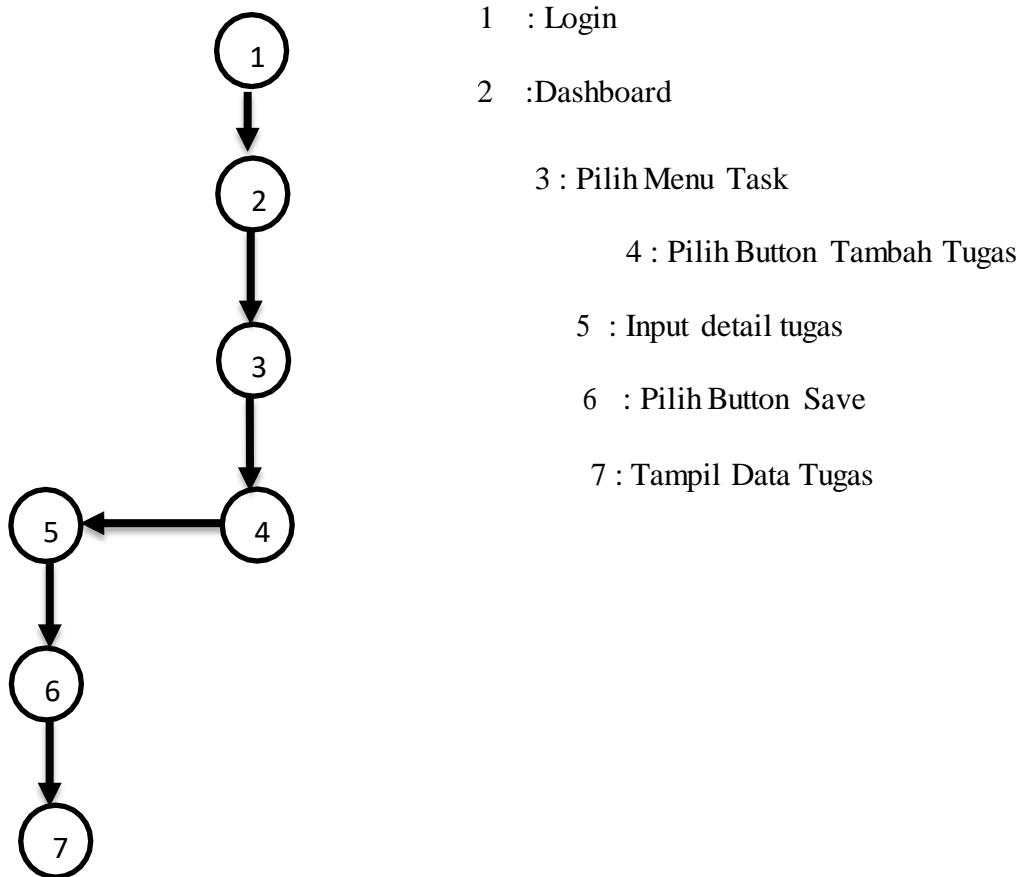
d. Pengujian

i. Dari jalur 1 sistem dapat melakukan login dan system dapat mengarahkan ke menu dashboard selanjutnya di arahkan ke menu tambah data mata kuliah dan system dapat menyimpan data mata kuliah baru.

## 5. Pengujian Tambah Tugas

### a. Menentukan Notasi

#### Diagram Alir



Gambar 5.44 Notasi Diagram Alir Proses tambah tugas

b. Jalur 1 : 1,2,3,4,5,6,7

Pada jalur ini *user* melakukan proses login dengan username dan password (node 1). System akan mengarahkan pengguna ke menu dashboard. Selanjutnya pilih menu task (3). System akan mengarahkan pengguna jika ingin menambah

task(node4). System akan mengarahkan pengguna untuk memasukan tugas baru(node5). Setelah pengguna telah memasukan detail tugas baru pengguna akan di arahkan untuk menekah tombol simpan(6). Setelah pengguna melakukan proses simpan data tugas baru system akan menampilkan hasil data yang telah diinputkan(node7).

i. Menghitung *Cyclomatic Complexity*

Berdasarkan hasil analisis terhadap notasi diagram alir aplikasi diketahui jumlah *Edge* (E) yaitu 7 dan jumlah *Node* (N) yaitu 7. Jumlah *cyclomatic complexity* adalah :

$$V(G) = E - N + 2 \quad V(G) = 6 - 7 + 2$$

$$V(G) = 1$$

j. Pengujian

Dari jalur 1 sistem dapat melakukan login dan system dapat mengarahkan ke menu dashboard selanjutnya diarahkan ke menu tugas selanjutnya system akan mengarahkan ke menu tambah ruangan yang selanjutnya system dapat menyimpan data ruangan dan menampilkannya.

## **BAB VI**

### **PENUTUP**

#### **6.1 Kesimpulan**

Kesimpulan Berdasarkan penelitian yang telah dilaksanakan, dapat disimpulkan bahwa tujuan penelitian yaitu merancang dan membuat aplikasi pengingat tugas berbasis Android sebagai sarana penunjang yang mampu mengingatkan mahasiswa tentang tugas dan batas waktu pengumpulan tugas telah terpenuhi. Hal ini dibuktikan berdasarkan pengujian whitebox dan blackbox. Hasil pengujian whitebox dengan menggunakan metode Cyclomatic Complexity (CC) membuktikan bahwa sistem telah benar dan tidak memiliki kesalahan baik dari segi logika maupun fungsi. Hasil pengujian blackbox membuktikan bahwa proses aplikasi yang dilakukan sudah benar sehingga fungsional sistem sudah dapat menghasilkan output yang diharapkan.

#### **6.2 Saran**

Saran Aplikasi Pengingat Tugas ini masih memiliki fungsi yang sangat terbatas, untuk membangun sebuah aplikasi yang baik tentu perlu dilakukan pengembangan baik dari sisi manfaat maupun sistem kerjanya. Berikut beberapa saran bagi yang ingin mengembangkan aplikasi yang mungkin dapat menambah nilai dari aplikasi nantinya yaitu fitur *booking* ruangan untuk jadwal tambahan.