

**PENGEMBANGAN SISTEM INFORMASI JADWAL
KULIAH UNTUK FAKULTAS KEDOKTERAN
UNIVERSITAS ISLAM INDONESIA**



Disusun Oleh:

N a m a : Moh. Dwi Chaidier Rizaldi
NIM : 17523157

**PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2021

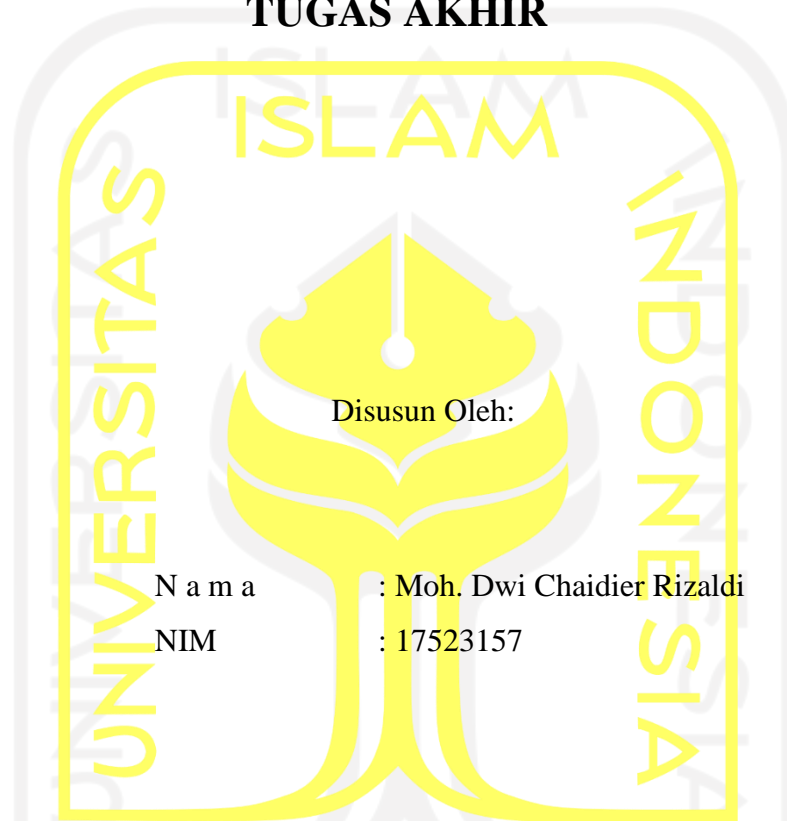
HALAMAN PENGESAHAN DOSEN PEMBIMBING

PENGEMBANGAN SISTEM INFORMASI JADWAL

KULIAH UNTUK FAKULTAS KEDOKTERAN

UNIVERSITAS ISLAM INDONESIA

TUGAS AKHIR



الجمعة المباركة الأندونيسية

Yogyakarta, 30 Juli 2021

Pembimbing,

(Ahmad Munasir Raf'ie Pratama, S.T., M.I.T., PH.D.)

HALAMAN PENGESAHAN DOSEN PENGUJI

**Pengembangan Sistem Informasi Jadwal Kuliah Untuk Fakultas
Kedokteran Universitas Islam Indonesia**

TUGAS AKHIR

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk
memperoleh gelar Sarjana Komputer dari Program Studi Teknik Informatika
di Fakultas Teknologi Industri Universitas Islam Indonesia

Yogyakarta, 30 Juli 2021

Tim Penguji

Ahmad Rafie Pratama, S.T., M.I.T., PH.D.

Anggota 1

Hanson Prihantoro Putro, S.T., M.T.

Anggota 2

Kurniawan Dwi Irianto, S.T., M.Sc.

Mengetahui,

Ketua Program Studi Informatika – Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia



(Dr. Raden Teduh Dirgahayu, S.T., M.Sc.)

HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan di bawah ini:

Nama : Moh. Dwi Chaidier Rizaldi

NIM : 17523157

Tugas akhir dengan judul:

**PENGEMBANGAN SISTEM INFORMASI JADWAL
KULIAH UNTUK FAKULTAS KEDOKTERAN
UNIVERSITAS ISLAM INDONESIA**

Menyatakan bahwa seluruh komponen dan isi dalam tugas akhir ini adalah hasil karya saya sendiri. Apabila dikemudian hari terbukti ada beberapa bagian dari karya ini adalah bukan hasil karya sendiri, tugas akhir yang diajukan sebagai hasil karya sendiri ini siap ditarik kembali dan siap menanggung resiko dan konsekuensi apapun.

Demikian surat pernyataan ini dibuat, semoga dapat dipergunakan sebagaimana mestinya.

Yogyakarta, 30 Juli 2021



(Moh. Dwi Chaidier Rizaldi)

HALAMAN PERSEMBAHAN

Puji syukur saya berikan kepada Allah Subhanallahu Wa Ta'ala atas rahmat, karunia, dan kekuatan yang diberikan kepada saya sehingga saya dapat menyelesaikan tugas akhir ini. Tugas akhir ini saya persembahkan untuk orang tua saya yang saya sangat cintai dan saya sayangi, yaitu bapak saya Ruslan Effendy dan ibu saya Nurbani Yusup yang selalu menampung keluh kesah saya dalam proses belajar serta memberikan dukungan verbal maupun non verbal untuk menyelesaikan studi saya, tidak luput juga saudara kandung saya yang sudah mendukung saya hingga detik ini dan juga kepada Silvi Rahmawati yang sudah membantu saya dalam mencari data-data yang diperlukan dalam menulis tugas akhir ini.

Terima kasih kepada seluruh dosen-dosen Informatika yang telah membantu saya, memberi saya pelajaran serta ilmu dalam membentuk pribadi yang lebih baik. Terutama kepada dosen pembimbing saya Pak Ahmad Raf'ie yang selalu sedia membimbing saya serta Bu Arrie Kurniawardhani selaku dosen wali yang memberikan dukungan selama menjadi mahasiswa Informatika UII.

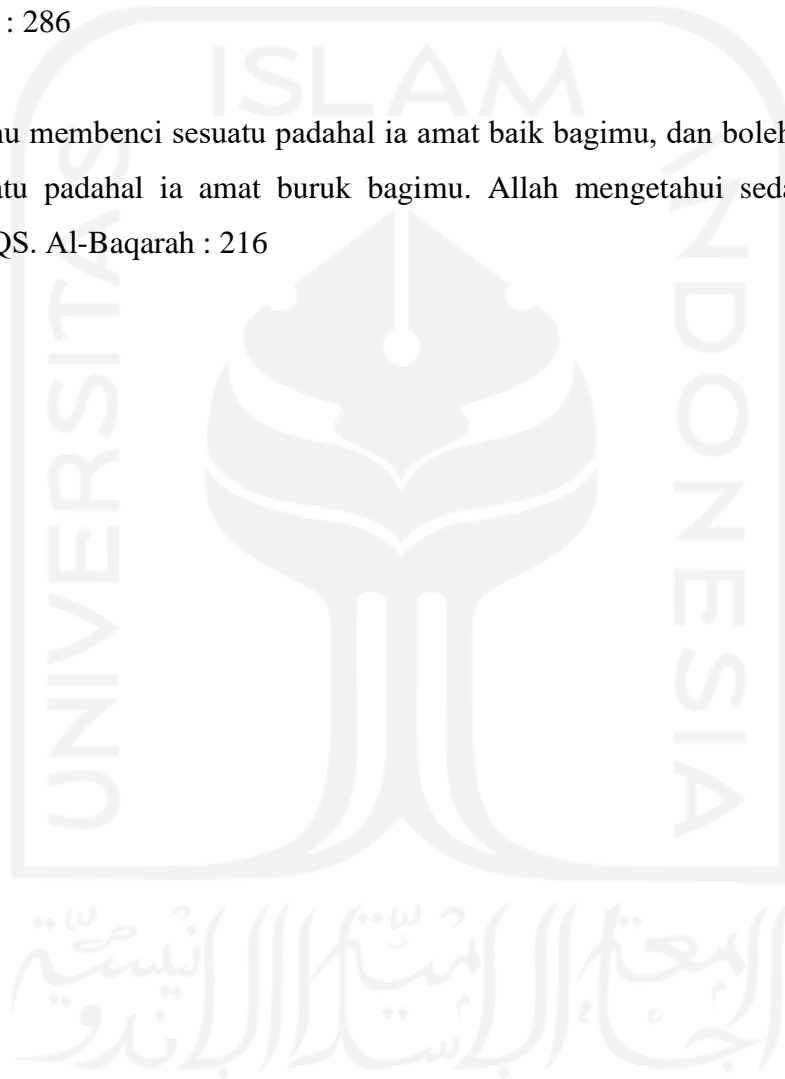
Tak lupa pula berterima kasih kepada teman-teman seperjuangan di Informatika. Terutama kepada Khukuh Anugrah yang sudah membantu menyelesaikan program aplikasi Android, Muhammad Yasin yang sudah membantu menyelesaikan program aplikasi *website*, Nur Jati Lantang dan Aditya Mahavira yang sudah membantu mengoreksi hasil tugas akhir saya. Serta saya sangat berterima kasih kepada semua teman-teman saya yang saya tidak bisa menyebutkannya satu persatu. Sukses selalu untuk kita semua.

HALAMAN MOTO

Jangan lupa 5 waktu, puasa sunnah, mengaji dan juga sedekah, mau sesibuk apapun urusan dunia jangan lupa libatkan Allah disetiap aktivitasmu ~ Ruslan Effendy, Nurbani Yusup

“Allah tidak akan membebani seseorang melainkan sesuai dengan kadar kesanggupannya” ~ QS. Al-Baqarah : 286

“Boleh jadi kamu membenci sesuatu padahal ia amat baik bagimu, dan boleh jadi pula kamu menyukai sesuatu padahal ia amat buruk bagimu. Allah mengetahui sedang kamu tidak mengetahui” ~ QS. Al-Baqarah : 216



KATA PENGANTAR

Bismillahirrahmanirrahim, Allhamdulillah puji syukur kehadiran Allah Subhanallahu Wa Ta'ala yang telah memberikan segala nikmat, rezeki, kesejahteraan, dan kelancaran dalam segala hal yang membantu saya dalam menyelesaikan tugas akhir sebagai syarat untuk menyelesaikan program pendidikan Strata Satu di Jurusan Informatika, Universitas Islam Indonesia.

Dalam proses penulisan penelitian ini, saya menyadari betul jika terdapat banyak kesulitan yang saya temui, seperti pengumpulan data, pencarian referensi, hingga pembuatan program. Saya juga menyadari bahwa tugas akhir ini dapat diselesaikan berkat dukungan dan bantuan dari berbagai pihak.

Oleh karena itu, pada kesempatan ini saya ingin mengucapkan terima kasih sebesar-besarnya kepada:


1. Allah Subhanallahu Wa Ta'ala, yang telah memberikan saya kemudahan dan petunjuk dalam menyelesaikan penelitian tugas akhir dengan baik.
2. Orang tua dan keluarga penulis atas segala doa dan dukungan selama saya melakukan penelitian ini untuk menyelesaikan tugas akhir.
3. Bapak Ahmad Rafi'e Pratama, S.T., MIT., PH.D. selaku Dosen Pembimbing yang telah bersedia meluangkan waktu untuk memberikan arahan selama penyusunan tugas akhir.
4. Ibu Arrie Kurniawardhani, S.Si.,M.Kom. selaku dosen wali yang telah memberikan dukungan pengarahan selama masa perkuliahan.
5. Silvi Rahamawati, selaku pendamping yang selalu memberi dukungan dan doa selama saya melakukan penelitian ini untuk menyelesaikan tugas akhir.
6. Teman-teman Universitas Islam Indonesia angkatan 2017 dan 2016 sebagai teman seperjuangan dan juga mentor yang baik dalam mengarahkan saya pada proses pembuatan tugas akhir ini.
7. Semua pihak yang tidak dapat disebutkan satu persatu yang telah membantu memberikan dukungan.

Semoga segala bantuan yang sudah diberikan sehingga saya mampu menyelesaikan tugas akhir ini dicatat oleh Allah Subhanallahu Wa Ta'ala sebagai bentuk amal ibadah. Aaamiiinnn.

Saya menyadari bahwa tugas akhir ini masih jauh dari kata sempurna, karena keterbatasan kemampuan, ilmu, dan juga pengalaman. Oleh karena itu, saya mengharap kritik dan saran yang membangun demi kesempurnaan tugas akhir ini. Semoga tugas akhir ini dapat memberikan manfaat dan mampu menjadikan referensi untuk penelitian sejenis sehingga dapat melakukan perbaikan-perbaikan yang ada pada penelitian sebelumnya. Akhir kata, saya ucapkan terima kasih untuk semua pihak yang telah membantu saya dalam menyelesaikan tugas akhir ini.

Wassalamu'alaykum Warrahmatullahi Wabarakatuh

Yogyakarta, 30 Juli 2021



(Moh. Dwi Chaidier Rizaldi)

الجمهورية الإسلامية اندونيسية

SARI

Selama ini, proses penyampaian informasi perubahan jadwal di Fakultas Kedokteran Universitas Islam Indonesia (FK UII) masih belum dilakukan secara *real time*. Di sisi lain, sering terjadi perubahan jadwal kuliah yang memaksa mahasiswa untuk senantiasa mengikuti perkembangan jadwal terkini yang hanya bisa didapatkan melalui Biro Administrasi Akademik dan Kemahasiswaan (BAAK). Hal ini tidak hanya memakan waktu lebih lama, namun juga membutuhkan banyak tempat dan menghabiskan banyak kertas. Penelitian ini bertujuan untuk memberikan solusi alternatif bagi mahasiswa, dosen/dosen, dan BAAK di lingkungan FK UII dalam hal manajemen dan penyampaian informasi jadwal kuliah secara *real time*. Dengan menggunakan metode *Waterfall SDLC*, aplikasi Android MedSch yang dikembangkan akan digunakan oleh mahasiswa dan dokter/dosen, sementara aplikasi berbasis *website* akan digunakan oleh pihak BAAK. Berdasar hasil pengujian fungsionalitas, aplikasi yang dibangun berjalan dengan lancar dan sudah menjalani seluruh skenario pengujian dengan metode *black box* dan mampu berjalan di *smartphone* pada sistem operasi Android versi 8.0 ke atas dan semua *web browser* utama di berbagai sistem operasi. Keberadaan aplikasi ini bisa membantu pencegahan terjadinya *miscommunication* terkait jadwal kuliah di lingkungan FK UII.

Kata kunci: sistem informasi, jadwal kuliah, Android, MedSch, *website*, mahasiswa, dosen.

GLOSARIUM

<i>Tools</i>	sebuah alat yang melakukan sebuah perintah pada sebuah aplikasi tertentu.
<i>Platform</i>	merupakan kombinasi antara sebuah arsitektur perangkat keras dengan sebuah kerangka kerja perangkat lunak. Kombinasi tersebut memungkinkan sebuah perangkat lunak, khusus perangkat lunak aplikasi, dapat berjalan.
<i>Code Editor</i>	sebuah perangkat lunak penyunting teks yang dirancang khusus untuk menyunting kode sumber program komputer oleh pemrogram. Aplikasi ini dapat berupa aplikasi yang berdiri sendiri atau dapat juga tergabung ke dalam sebuah <i>Integrated Development Environment</i> .
<i>Middleware</i>	perangkat lunak komputer yang menyediakan layanan bagi aplikasi perangkat lunak yang tersedia di luar sistem operasi. Peranti tengah dapat digambarkan sebagai "lem perangkat lunak".
<i>Mobile</i>	perangkat telekomunikasi elektronik yang mempunyai kemampuan dasar yang sama dengan telepon konvensional saluran tetap, tetapi dapat dibawa ke mana-mana dan tidak perlu disambungkan dengan jaringan telepon menggunakan kabel.
<i>Smartphone</i>	kelas ponsel dari ponsel dan perangkat komputasi bergerak multiguna. Mereka dibedakan dari ponsel berfitur dengan kemampuan perangkat keras yang lebih kuat dan sistem operasi seluler, yang memfasilitasi perangkat lunak yang lebih luas, bersama fungsi-fungsi inti ponsel seperti panggilan suara dan pesan teks.
<i>Programmer</i>	sebuah jenis profesi atau pekerjaan yang bertujuan untuk membuat sebuah sistem menggunakan bahasa pemrograman.
<i>Server-side Scripting</i>	sebuah teknik dalam perancangan desain web yang melibatkan <i>embedding script</i> dalam dokumen HTML yang diminta oleh client dari sebuah <i>server</i> , dimana permintaan akan diproses di sisi <i>server</i> melalui aplikasi <i>server-side</i> yang tersedia di <i>server</i> .
<i>Software</i>	istilah khusus untuk data yang diformat dan disimpan secara digital, termasuk program komputer, dokumentasinya, dan berbagai informasi yang bisa dibaca, dan ditulis oleh komputer.

<i>Hyperlink</i>	sebuah acuan dalam dokumen hiperteks ke dokumen yang lain atau sumber lain.
<i>Server</i>	program komputer atau perangkat yang menyediakan fungsionalitas untuk program atau perangkat lain, yang disebut "klien".
<i>Wireframing</i>	merupakan cara desainer UI/UX membuat rancangan <i>website</i> maupun aplikasi pada level struktural. Wireframe sendiri dapat diartikan dengan sederhana sebagai kerangka gambar.
<i>Handler</i>	program yang fungsinya hanya mengendalikan <i>input</i> , <i>output</i> , piranti penyimpanan, <i>file</i> , atau fasilitas untuk melakukan interupsi tertentu.
<i>Website</i>	Sekumpulan halaman <i>web</i> yang saling berhubungan yang umumnya berada pada peladen yang sama berisikan kumpulan informasi yang disediakan secara perorangan, kelompok, atau organisasi.
<i>REST</i>	singkatan bahasa Inggris dari <i>Representational State Transfer</i> atau transfer keadaan representasi, adalah suatu gaya arsitektur perangkat lunak untuk untuk pendistribusian sistem hipermedia.
<i>Miss-communication</i>	salah paham yang kerap terjadi saat kita melakukan interaksi atau komunikasi dengan lawan bicara.
<i>WhatsApp</i>	aplikasi pesan untuk ponsel cerdas.
<i>Dashboard</i>	aplikasi sistem informasi yang menyajikan informasi mengenai indikator utama dari aktifitas organisasi secara sekilas dalam layar tunggal.
<i>Database</i>	kumpulan data yang terorganisir, yang umumnya disimpan dan diakses secara elektronik dari suatu sistem komputer.
<i>Waterfall</i>	metode pengembangan perangkat lunak yang memungkinkan pembuatan sistem dilakukan secara terstruktur
<i>Black Box Testing</i>	merupakan metode pengujian perangkat lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program.
<i>Gradle</i>	sebuah program yang berfungsi untuk melakukan build secara otomatis.
<i>Google Cloud Platform</i>	adalah kumpulan layanan komputasi awan yang ditawarkan oleh Google.
<i>Google Cloud Messaging</i>	layanan gratis dari google tanpa ada pembatasan tidak seperti <i>google map service</i> dan <i>google translate service</i> .

<i>App Engine</i>	platform yang memungkinkan pengembang untuk membuat dan menjalankan aplikasi-aplikasi web.
<i>Emulator</i>	memungkinkan suatu program atau peranti lunak yang dibuat pada awalnya oleh suatu sistem komputer dan untuk dijalankan dalam sistem itu, dapat dijalankan dalam sistem komputer yang sama sekali berbeda.
<i>Library</i>	koleksi dari rutin-rutin program yang digunakan untuk membangun dan mengembangkan perangkat lunak. Pustaka umumnya mengandung kode program dan data pembantu, yang menyediakan layanan-layanan kepada program-program independen.
<i>Object Oriented</i>	paradigma pemrograman berdasarkan konsep "objek", yang dapat berisi data, dalam bentuk field atau dikenal juga sebagai atribut; serta kode, dalam bentuk fungsi/prosedur atau dikenal juga sebagai method.
<i>Spreadsheet</i>	tabel informasi/data berbentuk kotak dengan baris dan kolom yang berisi penghitungan-penghitungan yang digunakan untuk melakukan analisis komparatif.
<i>Classic Life Cycle</i>	suatu metode/proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian
<i>Flowchart</i>	jenis diagram yang mewakili algoritme, alir kerja atau proses, yang menampilkan langkah-langkah dalam bentuk simbol-simbol grafis, dan urutannya dihubungkan dengan panah.
<i>Web Browser</i>	perangkat lunak yang berfungsi untuk menerima dan menyajikan sumber informasi dari Internet.
<i>Login</i>	proses untuk mengakses komputer dengan memasukkan identitas dari akun pengguna dan kata sandi guna mendapatkan hak akses menggunakan sumber daya komputer tujuan.
<i>Use Case</i>	gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya.
<i>PC</i>	<i>personal computer</i> yang merujuk pada komputer yang dapat digunakan dan diperoleh orang dengan mudah.
<i>Web Service</i>	mekanisme komunikasi dua aplikasi/mesin terlepas dari arsitektur dan teknologi yang digaribawahi.

- Activity Diagram* bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan *concurrency*.
- Request* meminta sesuatu dari seseorang atau kita meminta seseorang melakukan sesuatu.
- ERD* *entity relationship diagram* merupakan model data berupa notasi grafis dalam pemodelan data konseptual yang menggambarkan hubungan antara penyimpan.
- User Interface* bentuk tampilan grafis yang berhubungan langsung dengan pengguna.
- Wireframe* sebuah kerangka untuk menata suatu item di laman *website* atau aplikasi.



DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN DOSEN PEMBIMBING.....	ii
HALAMAN PENGESAHAN DOSEN PENGUJI	iii
HALAMAN PERNYATAAN KEASLIAN TUGAS AKHIR	iv
HALAMAN PERSEMBAHAN	v
HALAMAN MOTO	vi
KATA PENGANTAR	vii
SARI.....	ix
GLOSARIUM.....	x
DAFTAR ISI	xiv
DAFTAR TABEL	xvi
DAFTAR GAMBAR.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	2
1.4 Tujuan Penelitian	2
1.5 Manfaat Penelitian	3
1.6 Langkah Penelitian.....	3
1.7 Sistematika Penulisan	4
BAB II LANDASAN TEORI	5
2.1 Sistem Informasi Perkuliahan	5
2.1.1 Android.....	6
2.1.2 Java	7
2.1.3 Website	7
2.1.4 PHP.....	8
2.1.5 MySQL.....	8
2.1.6 Node.js.....	8
2.1.7 API.....	9
2.2 Metode <i>Waterfall</i> SDLC	9
2.3 <i>Black Box Testing</i>	10
BAB III METODOLOGI PENELITIAN	11
3.1 Pengumpulan Data	12
3.2 Analisis Kebutuhan <i>User</i>	14
3.2.1 Proses Bisnis.....	14
3.2.2 Kebutuhan Fungsional & Non Fungsional.....	16
3.2.3 Identifikasi Aktor	17
3.2.4 Usecase Diagram.....	18
3.3 Perancangan	20
3.3.1 Activity Diagram.....	21
3.3.2 Percancangan <i>Database</i>	30
3.4 Implementasi	31
3.5 Pengujian.....	32
3.5.1 Black Box Testing	32
BAB IV HASIL DAN PEMBAHASAN	34
4.1 Hasil Implementasi	34
4.1.1 <i>Database</i>	34

4.1.2	User Interface	37
4.1.3	Desain Tampilan Aplikasi MedSch.....	57
4.2	Hasil Pengujian	79
	BAB V KESIMPULAN DAN SARAN.....	83
5.1	Kesimpulan	83
5.2	Saran.....	83
	DAFTAR PUSTAKA	85
	LAMPIRAN	88



DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional MedSch.....	16
Tabel 3.2 Kebutuhan Non Fungsional MedSch.....	17
Tabel 3.3 Tabel Identifikasi Aktor pada MedSch.....	17
Tabel 4.1 Hasil Pengujian Fungsionalitas Pada Aplikasi Berbasis Android	80
Tabel 4.2 Hasil Pengujian Fungsionalitas Pada Aplikasi Berbasis <i>Website</i>	81



DAFTAR GAMBAR

Gambar 3.1 Flowchart Perancangan Aplikasi MedSch	11
Gambar 3.2 Proses Bisnis Penjadwalan Kuliah FK UII MedSch.....	15
Gambar 3.3 <i>Use Case</i> Diagram MedSch Berbasis Android	18
Gambar 3.4 <i>Use Case</i> Diagram MedSch Berbasis <i>Website</i>	19
Gambar 3.5 Arsitektur Sistem Aplikasi MedSch.....	20
Gambar 3.6 <i>Activity Diagram</i> UC-01	21
Gambar 3.7 <i>Activity Diagram</i> UC-02	22
Gambar 3.8 <i>Activity Diagram</i> UC-03	22
Gambar 3.9 <i>Activity Diagram</i> UC-04	23
Gambar 3.10 <i>Activity Diagram</i> UC-05	24
Gambar 3.11 <i>Activity Diagram</i> UC-06	24
Gambar 3.12 <i>Activity Diagram</i> UC-07	25
Gambar 3.13 <i>Activity Diagram</i> UC-08	26
Gambar 3.14 <i>Activity Diagram</i> UC-09	26
Gambar 3.15 <i>Activity Diagram</i> UC-10	27
Gambar 3.16 <i>Activity Diagram</i> UC-11	28
Gambar 3.17 <i>Activity Diagram</i> UC-012	29
Gambar 3.18 <i>Activity Diagram</i> UC-013	30
Gambar 3.19 <i>Entity Relationship Diagram</i>	31
Gambar 4.1 <i>Layout Database</i> MedSch.....	34
Gambar 4.2 Tabel Jadwal	35
Gambar 4.3 Tabel Mahasiswa.....	35
Gambar 4.4 Tabel Pengajar.....	36
Gambar 4.5 Tabel Kelompok.....	36
Gambar 4.6 Tabel Blok.....	36
Gambar 4.7 Tabel Admin	37
Gambar 4.8 Tabel blok_tiap_mahasiswa	37
Gambar 4.9 Tabel blok_tiap_pengajar.....	37
Gambar 4.10 Halaman Registrasi Akun Mahasiswa	38
Gambar 4.11 Halaman <i>Login</i> Mahasiswa.....	39
Gambar 4.12 Halaman Beranda Mahasiswa	40
Gambar 4.13 Halaman Jadwal Kuliah Mahasiswa	41

Gambar 4.14 Halaman Daftar Dokter Mahasiswa.....	42
Gambar 4.15 Halaman <i>Login</i> Dosen.....	43
Gambar 4.16 Halaman Beranda Dosen.....	44
Gambar 4.17 Halaman Jadwal Kuliah Dosen.....	45
Gambar 4.18 Halaman <i>Update</i> Status Kuliah Dosen.....	46
Gambar 4.19 Halaman Konfirmasi Jadwal Dosen.....	47
Gambar 4.20 Halaman Konfirmasi Jadwal Dokter.....	48
Gambar 4.21 Halaman <i>Login</i> Admin.....	49
Gambar 4.22 Halaman Beranda Admin.....	50
Gambar 4.23 Halaman Jadwal Kuliah Admin.....	50
Gambar 4.24 Halaman Lihat Jadwal Admin.....	51
Gambar 4.25 Halaman Edit Jadwal Admin.....	52
Gambar 4.26 Halaman Hapus Jadwal Admin.....	52
Gambar 4.27 Halaman <i>Assign</i> Kelompok Mahasiswa.....	53
Gambar 4.28 Halaman Tambah Jadwal Admin.....	54
Gambar 4.29 Halaman Daftar Dokter Admin.....	54
Gambar 4.30 Halaman Lihat Dokter Admin.....	55
Gambar 4.31 Halaman Edit Dokter Admin.....	55
Gambar 4.32 Halaman Hapus Dokter Admin.....	56
Gambar 4.33 Halaman Tambah Dokter Admin.....	57
Gambar 4.34 Tampilan <i>Splash Screen</i> Aplikasi Android MedSch Mahasiswa.....	59
Gambar 4.35 Tampilan <i>Landing Page</i> Aplikasi Android MedSch Mahasiswa.....	60
Gambar 4.36 Tampilan <i>Login</i> Aplikasi Android MedSch Mahasiswa.....	61
Gambar 4.37 Tampilan <i>Sign Up</i> Aplikasi Android MedSch Mahasiswa.....	62
Gambar 4.38 Tampilan <i>Dashboard</i> Aplikasi Android MedSch Mahasiswa.....	63
Gambar 4.39 Tampilan Jadwal Kuliah Aplikasi Android MedSch Mahasiswa.....	64
Gambar 4.40 Tampilan Daftar Dokter Aplikasi Android MedSch Mahasiswa.....	65
Gambar 4.41 Tampilan Profile Aplikasi Android MedSch Mahasiswa.....	66
Gambar 4.42 Tampilan <i>Splash Screen</i> Aplikasi Android MedSch Dosen.....	68
Gambar 4.43 Tampilan <i>Login</i> Aplikasi Android MedSch Dosen.....	69
Gambar 4.44 Tampilan <i>Dashboard</i> Aplikasi Android MedSch Dosen.....	70
Gambar 4.45 Tampilan Jadwal Kuliah Aplikasi Android MedSch Dosen.....	71
Gambar 4.46 Tampilan Konfirmasi Jadwal Aplikasi Android MedSch Dosen.....	72
Gambar 4.47 Tampilan <i>Login</i> Aplikasi <i>Website</i> MedSch Admin.....	74

Gambar 4.48 Tampilan <i>Dashboard</i> Kuliah Aplikasi <i>Website</i> MedSch Admin.....	75
Gambar 4.49 Tampilan Lihat Jadwal Kuliah Aplikasi <i>Website</i> MedSch Admin	75
Gambar 4.50 Tampilan Hapus Jadwal Kuliah Aplikasi <i>Website</i> MedSch Admin	76
Gambar 4.51 Tampilan Assign Kelompok Mahasiswa Aplikasi <i>Website</i> MedSch Admin.....	76
Gambar 4.52 Tampilan Tambah Jadwal Aplikasi <i>Website</i> MedSch Admin	77
Gambar 4.53 Tampilan Lihat Dokter Aplikasi <i>Website</i> MedSch Admin	77
Gambar 4.54 Tampilan Edit Dokter Aplikasi <i>Website</i> MedSch Admin.....	78
Gambar 4.55 Tampilan Tambah Dokter Aplikasi <i>Website</i> MedSch Admin	78
Gambar 4.56 Tampilan Profile Aplikasi <i>Website</i> MedSch Admin.....	79



BAB I PENDAHULUAN

1.1 Latar Belakang

Fakultas Kedokteran Universitas Islam Indonesia merupakan salah satu fakultas di perguruan tinggi swasta yang berada di Kabupaten Sleman. Fakultas Kedokteran UII memiliki mahasiswa yang hampir sebagian besarnya berasal dari luar kota Sleman dan luar provinsi Jawa Tengah. Jumlah mahasiswa yang semakin meningkat mendorong Fakultas Kedokteran UII untuk terus meningkatkan kualitas pelayanan baik dalam bidang akademik maupun non-akademik.

Selama ini proses penyampaian informasi akademik termasuk di Fakultas Kedokteran Universitas Islam Indonesia (UII) dilakukan dengan memakan waktu yang lama dan membutuhkan banyak tempat karena menggunakan banyak kertas. Di dalam satu blok masa perkuliahan berlangsung sering terjadi *misscommunication* antara dosen dan mahasiswa perihal jadwal kuliah yang sudah ditentukan, karena seringnya terjadi perubahan jadwal kuliah, baik dibatalkan, diundur, atau adanya kuliah pengganti membuat mahasiswa harus dapat mengikuti perkembangan perubahan jadwal, sedangkan informasi perubahan jadwal hanya didapatkan melalui Biro Administrasi Akademik dan Kemahasiswaan (BAAK) Fakultas Kedokteran UII (kecuali jika dosen memberi informasi ke grup mahasiswa lewat aplikasi seperti *WhatsApp*). Universitas Islam Indonesia memiliki layanan informasi akademik sendiri yaitu UII Gateway. UII Gateway memiliki fitur jadwal kuliah yang bisa dilihat oleh mahasiswa. Namun jadwal kuliah yang ada di UII Gateway tersebut tidak bisa diubah lagi jika ada perubahan jadwal kuliah sedangkan bagi mahasiswa FK UII sendiri memerlukan jadwal kuliah yang mampu menyampaikan perubahan jadwal kuliah secara *real time*. Selain UII Gateway, Google Classroom merupakan pilihan lain dari UII Gateway untuk menyampaikan jadwal kuliah, namun beberapa dosen di FK UII jarang menggunakan Google Classroom dikarenakan sulitnya dalam menggunakan fitur yang ada pada Google Classroom. Hal ini membuat mahasiswa harus sering memeriksa jadwal ke BAAK.

Berdasarkan permasalahan diatas maka dibutuhkan suatu *tools* atau aplikasi yang dapat memberikan informasi yang cepat dan mampu mendistribusikan perubahan jadwal kuliah secara *real time* dari BAAK kepada mahasiswa, dan membuat aplikasi dengan fitur yang mudah digunakan oleh penggunanya. Oleh karena itu aplikasi pemberitahuan jadwal kuliah akan dikembangkan pada *platform* Android agar dapat diakses oleh mahasiswa dimanapun dan

kapanpun selama aplikasi sudah terinstal di perangkat Android yang terhubung dengan jaringan internet. Sedangkan untuk sisi BAAK akan dikembangkan aplikasi berbasis *website*. Aplikasi Android yang digunakan oleh mahasiswa akan dikembangkan dengan menggunakan bahasa pemrograman java. Sedangkan aplikasi *website* untuk pihak BAAK akan dibuat dalam bahasa pemrograman PHP.

Dengan dirancang dan dikembangkannya “Sistem Pemberitahuan Jadwal Kuliah” ini harapannya dapat membantu pihak BAAK untuk menyampaikan informasi jadwal kuliah kepada mahasiswa Fakultas Kedokteran UII. Adanya sistem seperti ini diharapkan dapat membuat proses penyampaian informasi lebih efektif dan efisien, serta dapat menghemat tempat dan tidak perlu menggunakan banyak kertas.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang sudah dijelaskan sebelumnya, maka perumusan masalah dapat diuraikan sebagai berikut:

- a. Bagaimana merancang sistem pemberitahuan jadwal kuliah yang sesuai dengan sistem perkuliahan yang berjalan di FK UII?
- b. Bagaimana mengimplementasikan rancangan sistem pemberitahuan jadwal kuliah FK UII berbasis *website* dan Android?

1.3 Batasan Masalah

Penelitian ini memerlukan batasan-batasan agar tidak menyimpang dari permasalahan yang sudah dirumuskan. Batasan masalah pada penelitian ini sebagai berikut:

- a. Sistem menampilkan jadwal kuliah mahasiswa Fakultas Kedokteran UII sesuai dengan kurikulum 2016 yang berjalan pada tahun akademik 2020/2021.
- b. Pada sistem ini terdapat tiga aktor, yaitu mahasiswa, dosen, dan BAAK.
- c. Mahasiswa dan dosen sebagai pengguna aplikasi Android.
- d. BAAK sebagai admin yang mengelola *dashboard* berbasis *website*.

1.4 Tujuan Penelitian

Penelitian ini memiliki beberapa tujuan yang nantinya bisa menjawab permasalahan yang sudah dirumuskan di atas, yaitu:

- a. Merancang sistem pemberitahuan jadwal kuliah sesuai dengan kebutuhan sistem perkuliahan yang sedang berjalan di Fakultas Kedokteran UII.

- b. Mengimplementasi dan melakukan pengujian aplikasi yang akan dibuat, diharapkan dapat berguna bagi para pengguna khususnya mahasiswa yang bersangkutan serta proses pembuatan jadwal mata kuliah di Fakultas Kedokteran UII akan dapat dilakukan lebih cepat dan sesuai dengan kebutuhan.

1.5 Manfaat Penelitian

Dalam pengembangan penelitian ini, diharapkan mampu memberikan manfaat kepada pihak yang membutuhkannya, yaitu:

- a. Untuk memudahkan mahasiswa dalam mengakses informasi akademik kapan saja dan dimana saja menggunakan *smartphone* yang mereka miliki.
- b. Menyingkat waktu bagi mahasiswa karena mampu melihat informasi akademik secara daring melalui *smartphone*.
- c. Memudahkan pihak Universitas khususnya Fakultas Kedokteran Universitas Islam Indonesia untuk memberi informasi tentang jadwal mata kuliah kepada mahasiswanya.

1.6 Langkah Penelitian

Dalam melakukan penelitian ini, peneliti menyiapkan beberapa tahapan yang meliputi:

- a. Kajian Pustaka.
Melakukan kajian lebih lanjut untuk beberapa literatur yang bersangkutan dengan penelitian supaya dijadikan acuan atau landasan.
- b. Pengumpulan Data.
Mengumpulkan data-data yang diperlukan dalam pengembangan sistem.
- c. Analisis Kebutuhan.
Menganalisis kebutuhan apa saja yang diperlukan oleh sistem.
- d. Perancangan.
Merancang perencanaan, pembuatan, dan penggambaran sketsa atau penyelarasan dari elemen - elemen yang terpisah setelah itu dibentuk menjadi kesatuan yang lengkap.
- e. Implementasi.
Mengimplementasikan seluruh rancangan yang telah dilakukan untuk dikembangkan menjadi sebuah aplikasi
- f. Pengujian.
Menguji setiap fungsionalitas yang sudah ditentukan pada aplikasi.

1.7 Sistematika Penulisan

Untuk menjelaskan bagaimana topik yang dibahas agar memudahkan para pembaca dalam memahami laporan penelitian ini, maka disusunlah laporan penelitian ini dengan sistematika penulisan sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang penelitian, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, langkah penelitian, serta sistematika penulisan yang digunakan dalam laporan penelitian.

BAB II KAJIAN PUSTAKA

Pada bab ini menjelaskan tentang sistem informasi perkuliahan atau akademik, Android, Java, *website*, PHP, MySQL, Node.js, API, metode *Waterfall SDLC*, dan *Black box testing*.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan metodologi laporan penelitian yang berisi langkah-langkah dalam penelitian ini yaitu pengumpulan data, analisis kebutuhan user, perancangan penelitian, implementasi, dan pengujian.

BAB IV HASIL DAN PEMBAHASAN

Pada bab ini menjelaskan hasil dan pembahasan dari implementasi serta pengujian yang sudah dilakukan pada tahap sebelumnya.

BAB V KESIMPULAN DAN SARAN

Bagian ini berisikan beberapa kesimpulan serta saran dari hasil penelitian yang sudah dilakukan.

BAB II LANDASAN TEORI

2.1 Sistem Informasi Perkuliahan

Sistem Informasi merupakan suatu sistem yang terdiri atas subsistem yang terintegrasi antara satu dengan lainnya sehingga mampu memberikan informasi yang bermanfaat bagi penggunanya atau pihak manajemen di dalam suatu organisasi untuk memudahkan pengambilan sebuah keputusan. Menurut Robert A Leitch: “Sistem informasi adalah sistem dalam organisasi yang mengatur kebutuhan pemrosesan transaksi sehari-hari, mendukung operasi, mengatur manajemen organisasi dan operasi strategis, serta menyediakan laporan tertentu yang dibutuhkan pihak ketiga.” (Setiawan et al., 2018).

Beberapa kajian literatur membahas tentang sistem informasi perkuliahan atau akademik berbasis website dan Android dimana dua *platform* tersebut bisa dijadikan sarana penyampaian informasi seperti jadwal kuliah dan lain-lainnya. Dalam pengembangan sistem informasi perkuliahan atau akademik berbasis *website* menurut Susanti (2016) adalah sebuah rancangan sistem yang mampu meringankan masyarakat dalam mencari informasi sekolah, karena masyarakat tidak perlu lagi pergi ke sekolah. Masyarakat cukup membuka layanan internet ke *website* sekolah yang dituju. Hal ini lebih efektif karena masyarakat tidak perlu menyisihkan waktu dan merogoh ongkos perjalanan untuk pergi ke sekolah.

Menurut Masturoh et al. (2019) sistem informasi akademik berbasis *website* merupakan sarana komunikasi bagi guru, siswa dan pihak sekolah untuk mendapatkan segala informasi mengenai sesuatu yang berkaitan dengan sistem proses belajar mengajar. Memberikan kemudahan kepada siswa untuk memperoleh informasi perihal kesiswaan, seperti jadwal belajar, skor penilaian dan informasi terbaru yang diperoleh dari pihak sekolah. Meningkatkan efektivitas dan efisiensi dalam menyebarkan informasi akademik kepada guru.

Menurut Dzihni et al. (2019) sistem informasi akademik berbasis *website* merupakan sebuah sistem yang dirancang untuk memenuhi kebutuhan akademik bagi masyarakat yang terlibat di dalamnya. Sistem informasi akademik menyediakan segala bentuk informasi dalam lingkup akademik yang dapat diakses dan digunakan kapanpun dan dimanapun melalui internet.

Untuk pengembangan sistem informasi perkuliahan atau akademik berbasis Android menurut Fitryani & Ariantini (2020) yaitu aplikasi sistem informasi yang mampu memberikan simplifikasi dalam proses pendistribusian jadwal kuliah. Pengembangan aplikasi sistem

informasi ini sangat bermanfaat untuk memberikan kemudahan dalam mengakses dan menyampaikan informasi terkait seperti jadwal kuliah dengan cepat, tepat, dan akurat. Karena dengan memanfaatkan platform Android, proses pengembangan sistem informasi akademik sangat efektif, efisien dan memudahkan dalam penyampaian informasi.

Menurut Rolly & Hakiem (2015), sistem informasi akademik berbasis Android merupakan sebuah aplikasi yang mampu mengoptimalkan dalam mengakses dan mengoperasikan sebuah sistem informasi akademik pada perangkat *smartphone* khususnya bagi dosen dan mahasiswa jika memerlukan informasi yang akurat dan bisa diakses melalui perangkat *mobile*. Sistem ini mampu memberikan menu dan antarmuka yang bisa menyesuaikan dengan kebutuhan dosen untuk mengakses KRS mahasiswa dan penginputan nilai, sedangkan mahasiswa sendiri bisa digunakan untuk melihat jadwal kuliah dan skor penilaian. Perancangan sistem informasi perkuliahan atau akademik disini memiliki beberapa komponen yang digunakan sebagai subbab-subbab tersendiri yang akan dibahas yaitu:

2.1.1 Android

Menurut Nazruddin (2012) Android adalah aplikasi sistem operasi ponsel dan inti OS Android adalah kernel Linux 2.6 yang diperbarui untuk perangkat seluler. Android adalah sistem operasi yang mencakup *middleware* dan aplikasi yang mendasarinya. Android menyediakan *platform* terbuka bagi pengembang untuk membangun aplikasi mereka sendiri, nantinya akan digunakan di berbagai perangkat seluler. Namun, untuk merancang sebuah aplikasi Android dibutuhkan sebuah editor kode dan *tools* utama untuk membangunnya, yaitu Android Studio dan Android SDK.

Menurut Yati (2018) Android Studio adalah sebuah Lingkungan Pengembangan Terpadu - Integrated Development Environment (IDE) untuk Android Development yang diperkenalkan google pada acara Google I/O 2013. Android Studio merupakan hasil pengembangan *code editor* dari Eclipse IDE dan berbasis Java IDE, khususnya IntelliJ IDE. Android Studio adalah IDE resmi untuk perancangan aplikasi berbasis Android. Sebagai evolusi dari Eclipse, Android Studio memperkenalkan banyak fitur baru dibandingkan dengan Eclipse IDE. Tidak seperti Eclipse yang menggunakan Ant, Android Studio menggunakan *Gradle* sebagai lingkungan binaan. Fitur lainnya adalah sebagai berikut:

1. Memakai *Gradle-based build system* yang fleksibel.
2. Mampu mem-*build multiple* APK .
3. Template support untuk *Google Services* dan beragam tipe perangkat.

4. Layout editor yang lebih memadai.
5. Bangun dukungan untuk *Google Cloud Platform* dan fasilitasi integrasi dengan *Google Cloud Messaging* dan *App Engine*.
6. Mengimpor library langsung dari repositori Maven.

Dalam hal membangun sebuah aplikasi, Android Studio menggunakan *Gradle*, jadi Android Studio lebih baik daripada Eclipse. Perbedaan lainnya antara Android Studio dengan Eclipse yaitu kita tidak perlu lagi dipusingkan dengan dependencies package pada Android Studio. Selain itu yang membuat Android Studio unggul adalah dukungan layout xml editor secara visual yang jauh lebih baik dari pada Eclipse (Android Studio, 2021).

Dalam pengembangan aplikasi untuk *platform* Android, telah disediakan sebuah SDK (*Software Development Kit*) yaitu Android SDK. Android SDK adalah alat untuk *programmer* yang ingin membangun aplikasi berbasis Google Android. Android SDK mencakup seperangkat alat pengembangan atau paket yang komprehensif (Yati, 2018). Di dalam paket Android SDK sudah disediakan contoh kode sumber *project*, *development tools*, *emulator* dan *library* yang dibutuhkan untuk membangun aplikasi Android. Pengembangan aplikasi Android menggunakan bahasa pemrograman Java, dimana konsep-konsep pemrograman Java berhubungan dengan Pemrograman Berbasis Objek (OOP).

2.1.2 Java

Menurut Andrian (2019), Java merupakan bahasa pemrograman untuk membuat isi yang aktif dalam halaman web, juga dapat digerakkan pada semua jenis komputer. Java adalah bahasa pemrograman berorientasi objek murni karena semua kode program dienkapsulasi dalam kelas. Java adalah bahasa pemrograman yang berkembang dengan orientasi objek. Teknologi *object oriented* melihat *software* sebagai sebuah interaksi antar bagian dalam sebuah sistem dan menjabarkan bagian tersebut ke dalam satu objek yang mempunyai sifat/properti/data dan memiliki kemampuan untuk melakukan suatu tugas tertentu.

2.1.3 Website

Menurut Utama (2011) *website* atau situs bisa diartikan seperti kumpulan halaman-halaman yang dipakai untuk menampilkan informasi, teks, gambar diam atau bergerak, animasi, suara, atau gabungan dari semuanya itu, baik yang bersifat dinamis maupun statis yang membentuk satu rangkaian bangunan yang saling berkaitan dimana masing-masing dihubungkan dengan jaringan halaman atau *hyperlink*.

2.1.4 PHP

Menurut Novendri et.al (2019) Bahasa pemrograman PHP merupakan bahasa pemrograman yang digunakan untuk membuat *website* yang bersifat *server-side scripting*. PHP memiliki sifat yang dinamis karena mampu dijalankan pada berbagai macam sistem operasi seperti Windows, Linux, dan Mac Os. PHP dapat menggunakan *database* untuk membuat halaman web dinamis. Sistem manajemen basis data yang paling umum digunakan dalam PHP adalah MySQL. Namun, PHP juga mampu memberi dukungan sistem manajemen *Database Oracle, Microsoft Access, Interbase, d-Base, dan PostgreSQL*.

2.1.5 MySQL

Menurut Novendri et.al (2019) MySQL adalah sebuah *software* atau sistem manajemen *database* yang bersifat relasional. Artinya, data yang dikelola dalam *database* kemudian akan dimasukkan ke dalam beberapa tabel terpisah, sehingga manipulasi data menjadi lebih cepat. MySQL mampu digunakan untuk pengelolaan *database* dimulai dari yang kecil sampai dengan yang sangat besar.

SQL bisa diartikan sebagai bahasa perintah standar untuk sistem manajemen relasional, termasuk sistem yang dioperasikan pada komputer pribadi. SQL memungkinkan seorang pengguna untuk mengetahui dimana lokasi data tersebut, atau bagaimana informasi terkait akan disusun. SQL lebih mudah digunakan daripada bahasa pemrograman, tetapi jauh lebih kompleks daripada *spreadsheet* dan perangkat lunak pengolah data. Kumpulan data SQL sederhana yang dapat memenuhi kumpulan informasi yang diperlukan disimpan di komputer yang berbeda di lokasi terdistribusi, yang membutuhkan banyak waktu dan sumber daya komputasi.

2.1.6 Node.js

Menurut Saputra & Santoso (2021) Node.js merupakan *software* yang dikembangkan untuk membangun aplikasi berbasis web dan ditulis dalam sintaks bahasa pemrograman JavaScript. Node.js dibuat untuk meningkatkan peran JavaScript sebagai bahasa pemrograman sisi *server*. Node.js memiliki pustaka *server* HTTP bawaan yang memungkinkan Anda menjadi *server* web tanpa menggunakan perangkat lunak lain seperti Apache atau Nginx. Ini menjadikan Node.js lingkungan eksekusi bersyarat untuk mengeksekusi, mengeksekusi, dan mengeksekusi fungsi, dan juga merupakan jantung dari bahasa pemrograman.

2.1.7 API

Menurut Lubis (2017) API merupakan singkatan dari *Application Programming Interface*, atau bisa disebut dengan Antarmuka Pemrograman aplikasi. Lubis (2017) juga menuturkan jika API adalah seperangkat perintah, fungsi, serta protokol yang mampu dipakai oleh programmer saat mengembangkan perangkat lunak. API memungkinkan pemrogram untuk berinteraksi dengan sistem operasi lain menggunakan fungsi standar. Sesuai dengan fungsinya yang bisa digunakan sebagai penerjemah dan sebagai jembatan antar perangkat lunak, teknologi bernama API ini membuat persoalan sulit menjadi lebih mudah.

API bisa difungsikan untuk bahasa pemrograman atau sistem operasi apapun, selama paket yang dibutuhkan sudah terinstal. Karena dalam API terdapat fungsi-fungsi/perintah- perintah yang menggantikan bahasa untuk digunakan dalam *system calls* (membedakan antara satu sistem operasi dengan yang lainnya) dengan bahasa yang lebih teratur dan mudah dipahami oleh programmer, termasuk programmer paling pemula sekalipun. Fungsi yang dibuat oleh API kemudian akan dipanggil sesuai dengan sistem operasi.

2.2 Metode *Waterfall* SDLC

Menurut Wahid (2020) Metode *Waterfall* SDLC adalah suatu metode yang dipakai untuk melakukan pengembangan sistem informasi berbasis komputer. Pendekatan SDLC menggunakan apa yang disebut pendekatan air terjun, yang digunakan pada berbagai tahap pengembangan sistem. Fase ini disebut air terjun karena sistem beroperasi dalam urutan menurun dari fase ke fase di setiap prosesnya.

Kelebihan metode *Waterfall* antara lain model yang mudah dipahami, mudah digunakan, kebutuhan sistem yang stabil, baik dalam manajemen kontrol, dan bekerja dengan baik ketika kualitas diprioritaskan daripada biaya dan jadwal (tenggat waktu). Kekurangannya adalah semua persyaratan sistem harus diketahui terlebih dahulu, dapat memberikan kesan palsu tentang kemajuan, pelanggan hanya memiliki sedikit kesempatan untuk melihat dan meninjau sistem (yaitu di akhir proyek), risikonya sangat tinggi karena pengujian hanya dilakukan pada akhir setiap fase. pengujian hanya dilakukan pada akhir proyek, membutuhkan waktu yang lama (walaupun proyeknya tidak terlalu besar), perubahan persyaratan dapat mengubah seluruh proses yang telah diimplementasikan.

2.3 *Black Box Testing*

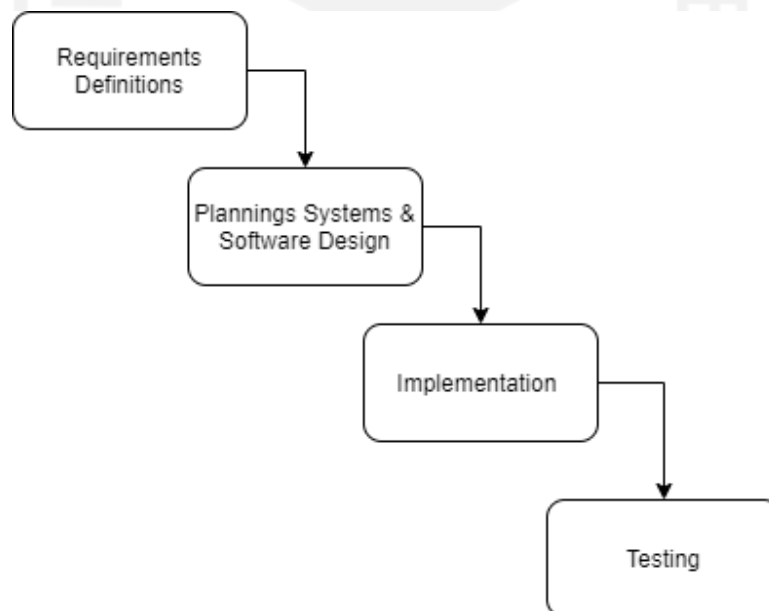
Menurut Nigrum et al. (2019) Metode *Black box testing* merupakan sebuah metode yang digunakan untuk menguji sebuah *software* tanpa harus mengamati detail *software*. Pemeriksaan pada pengujian ini hanya memvalidasi setiap nilai keluaran terhadap nilai masukan. Tidak ada usaha untuk mengetahui bagaimana kode program yang digunakan dan seperti apa *output* yang dipakai. Pengujian *black box* dilakukan dengan menguji program yang dihasilkan dan mencoba mengisi setiap modul yang telah ditentukan.

Sama halnya dengan penyampaian dari Febiharsa et.al (2018) yaitu pengujian *black box* atau disebut uji fungsional adalah pengujian yang mengabaikan mekanisme internal sistem atau komponen *software* yang sudah dibuat, namun hanya berfokus pada *output* yang sudah dilakukan dalam menanggapi input yang disesuaikan dan kondisi eksekusi. Maka dapat disimpulkan jika *black box testing* adalah sebuah riset yang berorientasi pada fungsionalitas yaitu perilaku dari perangkat lunak dari *input* yang diberikan pengguna sehingga membuahkan hasil *output* yang diharapkan tanpa mengetahui proses internal atau kode program yang dijalankan pada perangkat lunak.

BAB III

METODOLOGI PENELITIAN

Pada bab ini akan membahas proses perancangan penelitian, dimana proses tersebut dapat didefinisikan sebagai perencanaan, pembuatan, dan penggambaran sketsa atau penyelarasan dari elemen - elemen yang terpisah setelah itu dibentuk menjadi kesatuan yang lengkap dan mampu difungsikan sebagai perancangan sistem. Disini peneliti memilih metode penelitian dengan merancang sebuah aplikasi MedSch menggunakan metode *Waterfall* SDLC. Metode *Waterfall* sering juga disebut dengan *classic life cycle* adalah suatu metode/proses pengembangan perangkat lunak berurutan, di mana kemajuan dipandang sebagai terus mengalir ke bawah (seperti air terjun) melewati fase-fase perencanaan, pemodelan, implementasi (konstruksi), dan pengujian (Setiawan et al., 2018). Metode *waterfall* merupakan sebuah metode model lama namun matang dalam perancangannya, sehingga peneliti banyak yang memilih metode ini dalam penelitian pengembangan sebuah sistem. Gambar 3.1 menjelaskan bagaimana tahapan pengembangan perangkat lunak yang memanfaatkan metode *waterfall*.



Gambar 3.1 Flowchart Perancangan Aplikasi MedSch

- a. *Requirement Definitions* disini menjelaskan bagaimana peneliti melakukan pengembangan perangkat lunak, peneliti harus mengetahui dan memahami bagaimana informasi kebutuhan pengguna terhadap sebuah perangkat lunak. Metode yang diakumulasikan

dalam bentuk informasi ini diperoleh dengan berbagai cara yaitu observasi, wawancara, survei, diskusi, dan lainnya. Setelah informasi yang dibutuhkan sudah cukup selanjutnya akan dianalisis dan diolah oleh peneliti terkait perangkat lunak yang akan dikembangkan.

- b. *Planning Systems & Software Design* disini menjelaskan tentang informasi mengenai spesifikasi kebutuhan dari tahap *Requirement Definitions* dan selanjutnya di analisa, pada tahap ini akan dilanjutkan dengan pengimplementasian ke desain pengembangan. Perancangan desain memiliki tujuan untuk membantu menyampaikan gambaran besar perihal perancangan yang harus dikerjakan. Tahap ini mampu menyokong peneliti untuk mempersiapkan kebutuhan pengguna dalam penyusunan struktur sistem perangkat lunak yang akan dibuat secara menyeluruh. Tujuannya untuk lebih memahami gambaran besar dari apa yang akan dikerjakan.
- c. *Implementation* merupakan tahap pemrograman, dalam pembuatan perangkat lunak dibagi menjadi modul-modul yang lebih kecil agar mudah dibangun dan hasil akhirnya akan digabungkan dalam tahap berikutnya. Tahap ini akan dilanjutkan ke proses pengujian dan pengecekan perihal fungsionalitas modul yang sudah dikerjakan, apakah sudah memenuhi tolok ukur yang diharapkan atau belum. Tahapan ini bisa dibilang sebagai tahapan proses penerjemahan dari bentuk desain menjadi kode/program atau bentuk bahasa yang mampu dibaca oleh mesin. Setelah proses pemrograman selesai, maka dilanjutkan ke pemeriksaan terhadap sistem dan juga program yang sudah dibuat. Tujuannya yaitu mencari kesalahan yang mungkin terjadi dan kedepannya bisa diperbaiki melalui *update* berkala.
- d. *Testing* adalah tahap yang dilakukan untuk pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah pengembangan aplikasi tersebut memenuhi kriteria yang sudah ditentukan atau belum. Setelah proses implementasi selesai, selanjutnya melakukan pengujian terhadap sistem dan juga kode yang sudah dibangun. Tujuannya untuk menemukan kesalahan yang mungkin terjadi disaat tahap pengemplementasian agar nantinya diperbaiki. Tahap terakhir yaitu melakukan pengecekan dan pengujian sistem secara menyeluruh untuk mencari kemungkinan adanya kegagalan dan kesalahan sistem.

3.1 Pengumpulan Data

Berdasarkan data atau informasi yang berhasil dikumpulkan, maka dua hal tersebut bisa digunakan untuk memulai tahapan analisis dan membangun *software*. Dalam penelitian ini, menurut cara mendapatkan data yang digunakan dapat dikategorikan sebagai data primer.

Pengertian data primer menurut Sugiyono (2015) adalah sumber data yang langsung memberikan data kepada pengumpul data. Data primer diperoleh dari hasil wawancara ke salah satu staf Biro Administrasi Akademik dan Kemahasiswaan Fakultas Kedokteran UII mengenai bagaimana alur penjadwalan kuliah yang berlangsung disana. Berikut pertanyaan yang disampaikan untuk sesi wawancara ke salah satu staf BAAK FK UII, yaitu:

1. Bagaimana proses penyusunan jadwal kuliah yang ada di FK UII?
2. Siapa saja yang terlibat dalam penyusunan jadwal kuliah di FK UII?
3. Apa peran dosen dan BAAK dalam penyusunan jadwal kuliah di FK UII?
4. Berdasarkan proses penyusunan jadwal kuliah, apa saja kendala yang sering dialami oleh pihak BAAK maupun dosen?

Berdasarkan hasil wawancara dan observasi dengan pihak bersangkutan, maka ditemukan beberapa permasalahan yang diuraikan antara lain:

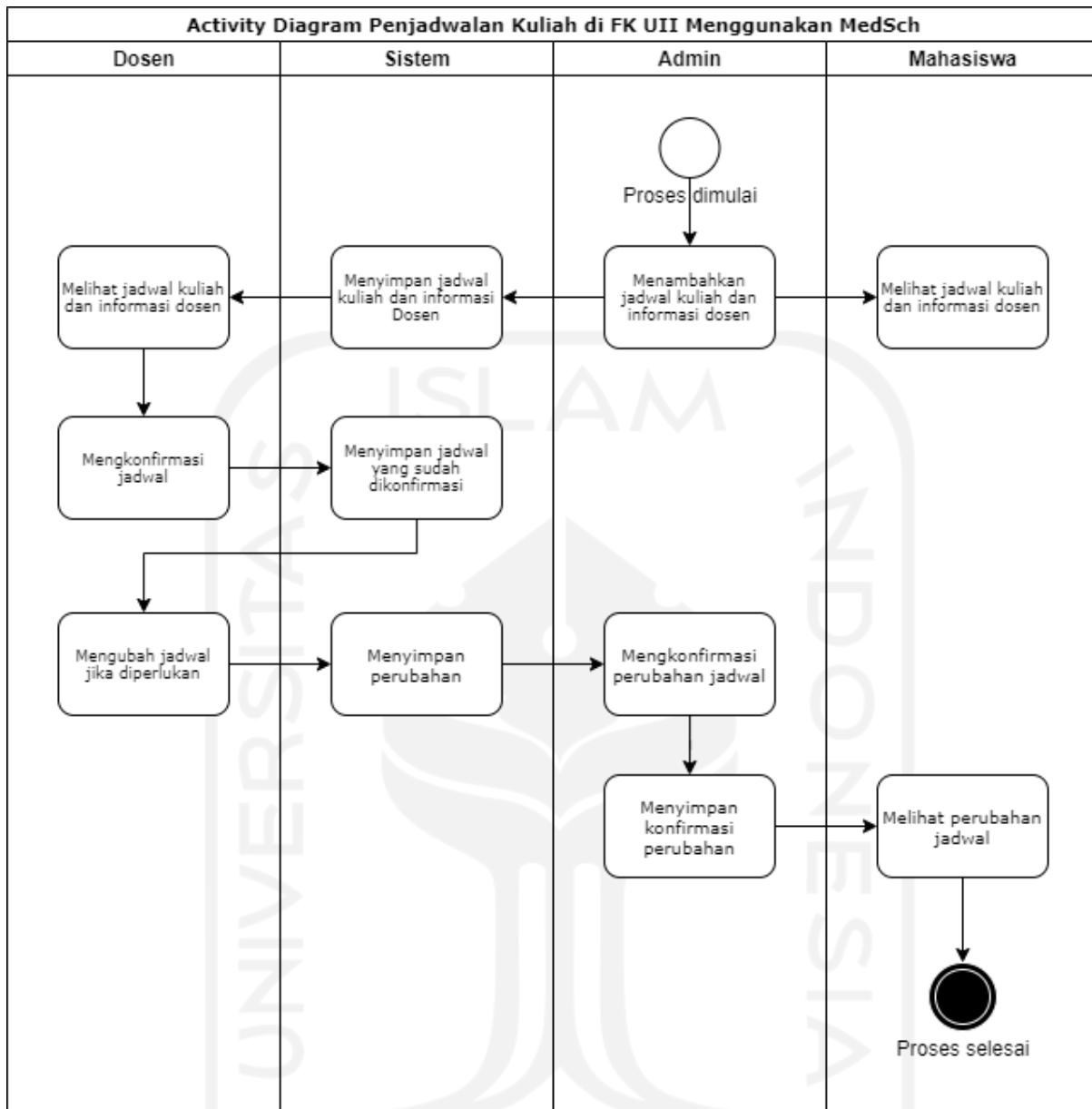
1. Penjadwalan yang ada di Fakultas Kedokteran UII sangat berbeda dengan sistem pada Fakultas lain yang ada di UII. Kalender akademik FK UII yang telah dibagikan tiap semester memiliki struktur jadwal kuliah tersendiri berbeda dengan Fakultas lain. Contohnya pada tiap semester, ada mata kuliah (kalau di FK yaitu blok) untuk Fakultas lain berjalan bersamaan, di FK hanya menjalankan 1 blok yang tidak berjalan bersamaan. Jadwal kuliah tidak disusun pada awal semester, tetapi disusun sebelum memulai perkuliahan. Contohnya blok yang dimulai pada tanggal 2 Maret 2020, akan disusun 2 minggu sebelum dimulai perkuliahan.
2. Penyusunan jadwal kuliah yang ada di FK untuk kuliah kuliah pakar, praktikum, medik, dan tutorial akan diajarkan oleh dosen sesuai bidang kepakarannya dan disusun oleh BAAK agar tidak terjadi tabrakan selama berlangsungnya proses perkuliahan.
3. Penentuan dalam penyampaian materi – materi yang akan disampaikan pada proses perkuliahan akan dibentuk oleh para dosen dan dibantu dengan BAAK sesuai dengan unit kurikulum.
4. Masalah pada distribusi penjadwalan yaitu jika terdapat kuliah yang dibatalkan, kuliah yang diundur, dan kuliah pengganti, mahasiswa akan mendapatkan pemberitahuan setelah berada di kampus atau dari BAAK (kecuali jika dosen menginfokan ke grup mahasiswa lewat aplikasi seperti *WhatsApp*). Jika terjadi pergantian jadwal kuliah, pembaharuan tersebut tidak bisa diakses secara daring oleh mahasiswa. Ketika ada dosen yang ingin membatalkan kuliah, jam perkuliahan diundur, serta pergantian jadwal perkuliahan harus melewati BAAK dahulu. Jadwal Kegiatan Kuliah masih menggunakan selebaran kertas

yang di *print* setelah itu dibagikan kepada para mahasiswa, ada juga versi daringnya berupa gambar yang dikonversi dengan format *pdf*.

3.2 Analilis Kebutuhan *User*

3.2.1 Proses Bisnis

Secara umum pada proses penyusunan jadwal kuliah di Fakultas Kedokteran (FK) Universitas Islam Indonesia masih memakan waktu yang lama, dan juga media selain BAAK yaitu melalui UII Gateway dan Classroom saja. UII Gateway hanya menampilkan jadwal tetap yang ada di FK UII, sedangkan media Classroom jarang digunakan oleh dosen yang ada di FK UII. Selama perkuliahan berlangsung sering terjadi *misscommunication* antara dosen dan mahasiswa perihal jadwal kuliah sedangkan mahasiswa mendapatkan informasi jadwal kuliah hanya bisa didapatkan melalui Biro Administrasi Akademik dan Kemahasiswaan (BAAK) Fakultas Kedokteran UII, terkecuali jika dosen menginfokan ke grup mahasiswa lewat aplikasi seperti *WhatsApp*. Hal ini membuat mahasiswa harus sering memeriksa jadwal ke BAAK. Pada tahap inilah proses akan diperbaiki, dalam penyusunan dan pendistribusian jadwal kuliah yang memakan waktu lama dan manual akan disederhanakan dengan adanya sistem pemberitahuan jadwal kuliah. Gambar 3.2 menjelaskan bagaimana alur proses bisnis MedSch bekerja.



Gambar 3.2 Proses Bisnis Penjadwalan Kuliah FK UII MedSch.

Proses bisnis diatas menjabarkan hubungan aktivitas – aktivitas pengguna dengan aplikasi, yaitu:

- a. Mahasiswa menggunakan aplikasi ini untuk melihat jadwal kuliah serta melihat daftar dokter yang mengajar di Fakultas Kedokteran UII.
- b. Admin menggunakan *web browser* untuk mengelola data jadwal kuliah dan juga data dosen dengan adanya aktivitas edit dan hapus, juga admin dapat melakukan aktivitas konfirmasi jadwal kuliah dan *update* status jadwal jika ada perubahan jadwal dari dosen yang akan mengajar. Admin akan menambah jadwal kuliah baru dan daftar dokter yang baru, setelah

itu dikirim ke sistem dan sistem menyimpan apa yang sudah dibuat dari admin, nantinya jika mahasiswa ingin melihat jadwal kuliah dan daftar dokter, sistem akan menampilkan data-data yang sudah diinput oleh admin.

- c. Dosen menggunakan aplikasi ini dengan cara *login*, dimana data *login* diberi langsung oleh admin agar bisa masuk ke halaman beranda sehingga dapat melihat jadwal kuliah serta dapat mengonfirmasi jadwal kuliah yang sudah dibuat oleh admin. Dosen dapat memperbaharui status kelas jika dosen tersebut mengalami halangan untuk mengajar mahasiswanya dan akan dikirim melalui *server*, *server* mengirim ke admin agar admin dapat mengubah status kelas yang akan berlangsung.

3.2.2 Kebutuhan Fungsional & Non Fungsional

Dari analisis permasalahan yang telah dilakukan, maka diperoleh beberapa hal utama yang dibutuhkan oleh pengguna dalam sebuah sistem pemberitahuan jadwal kuliah. Tabel 3.1 memperlihatkan 5 kebutuhan fungsionalitas dari MedSch, sebuah aplikasi informasi jadwal kuliah. Dan tabel 3.2 menunjukkan kebutuhan non fungsionalitas dari MedSch. 4 kebutuhan pendukung sistem pemberitahuan jadwal kuliah ini juga diperoleh dari hasil analisis permasalahan sebelumnya.

Tabel 3.1 Kebutuhan Fungsional MedSch

Kode	Tujuan
MOF-01	Aplikasi harus dapat menyimpan data diri mahasiswa.
MOF-02	Aplikasi harus dapat menyediakan fasilitas <i>sign-in</i> .
MOF-03	Aplikasi harus dapat menyimpan data informasi akademik (jadwal kuliah, nama dosen, dll) untuk ditampilkan kepada mahasiswa.
MOF-04	Aplikasi harus dapat menyediakan fasilitas kepada admin untuk mengubah data informasi akademik yang berupa status perkuliahan, konfirmasi antara dosen dengan admin, alasan cancel jika terjadi perubahan jadwal kuliah, dan mampu mengelompokkan mahasiswa untuk pendistribusian jadwal kuliah.

MOF-05	Aplikasi harus dapat memberikan informasi kepada mahasiswa berupa status perkuliahan, tampilan jadwal perkuliahan, dan alasan cancel jika terjadi perubahan jadwal kuliah.
--------	--

Tabel 3.2 Kebutuhan Non Fungsional MedSch

Kode	Tujuan
MON-01	Aplikasi harus memiliki usability yang tinggi sehingga mudah digunakan oleh mahasiswa, dosen maupun admin.
MON-02	Aplikasi harus memiliki tampilan yang menarik, dengan tabel jadwal kuliah yang menarik sehingga mahasiswa dapat mengakses aplikasi dengan dengan nyaman.
MON-03	Aplikasi harus menyediakan informasi yang jelas terkait jadwal kuliah.
MON-04	Aplikasi harus dapat menyediakan informasi data diri mahasiswa.

3.2.3 Identifikasi Aktor

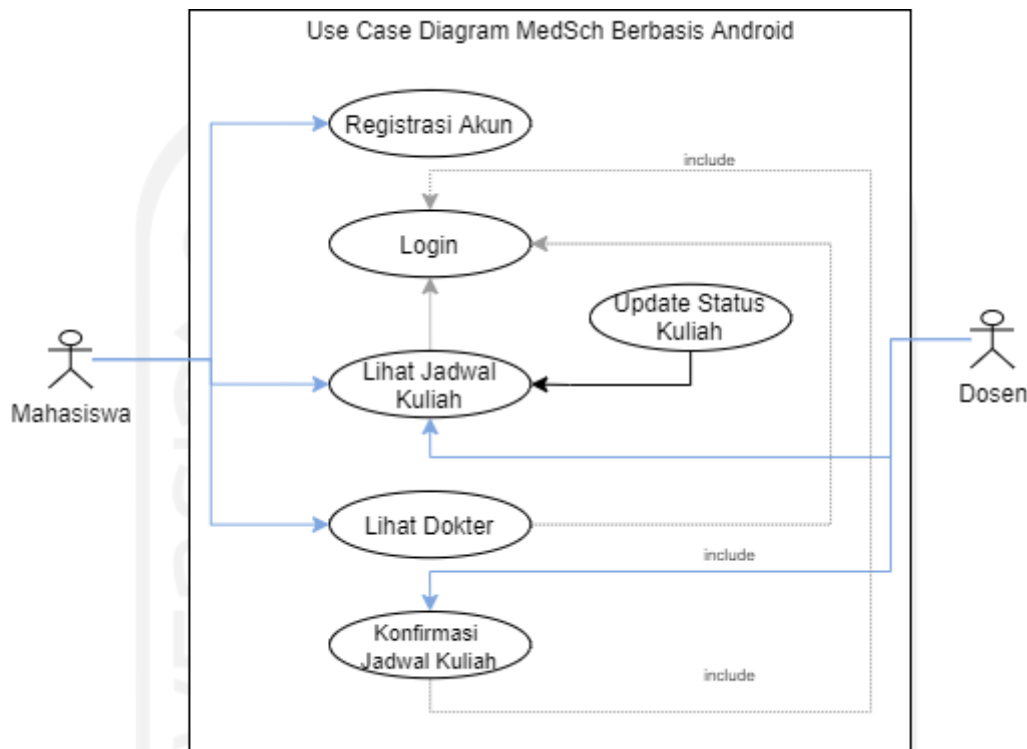
Dari kebutuhan fungsionalitas yang telah disampaikan sebelumnya, maka diidentifikasi tiga aktor dalam MedSch. Tabel 3.3 menjelaskan ketiga aktor dalam sistem ini.

Tabel 3.3 Tabel Identifikasi Aktor pada MedSch

Nama Aktor	Deskripsi
Admin	Admin adalah BAAK, merupakan orang yang bertugas dan memiliki hak akses untuk melakukan pengelolaan jadwal kuliah, data mahasiswa, dan data dosen. Admin akan memberikan informasi mengenai jadwal kuliah dan informasi tentang dosen yang mengajar di Fakultas Kedokteran UII.
Mahasiswa	Mahasiswa adalah sebutan orang yang sedang menempuh pendidikan di sebuah perguruan tinggi. Mahasiswa disini diberi hak untuk melihat informasi mengenai jadwal kuliah dan daftar dokter yang mengajar di Fakultas Kedokteran UII
Dosen	Dosen adalah dokter, merupakan pendidik profesional dalam bidangnya. Dosen disini diberi hak untuk melihat informasi mengenai jadwal kuliah, mengkonfirmasi jadwal kuliah serta mengupdate status kelas.

3.2.4 Usecase Diagram

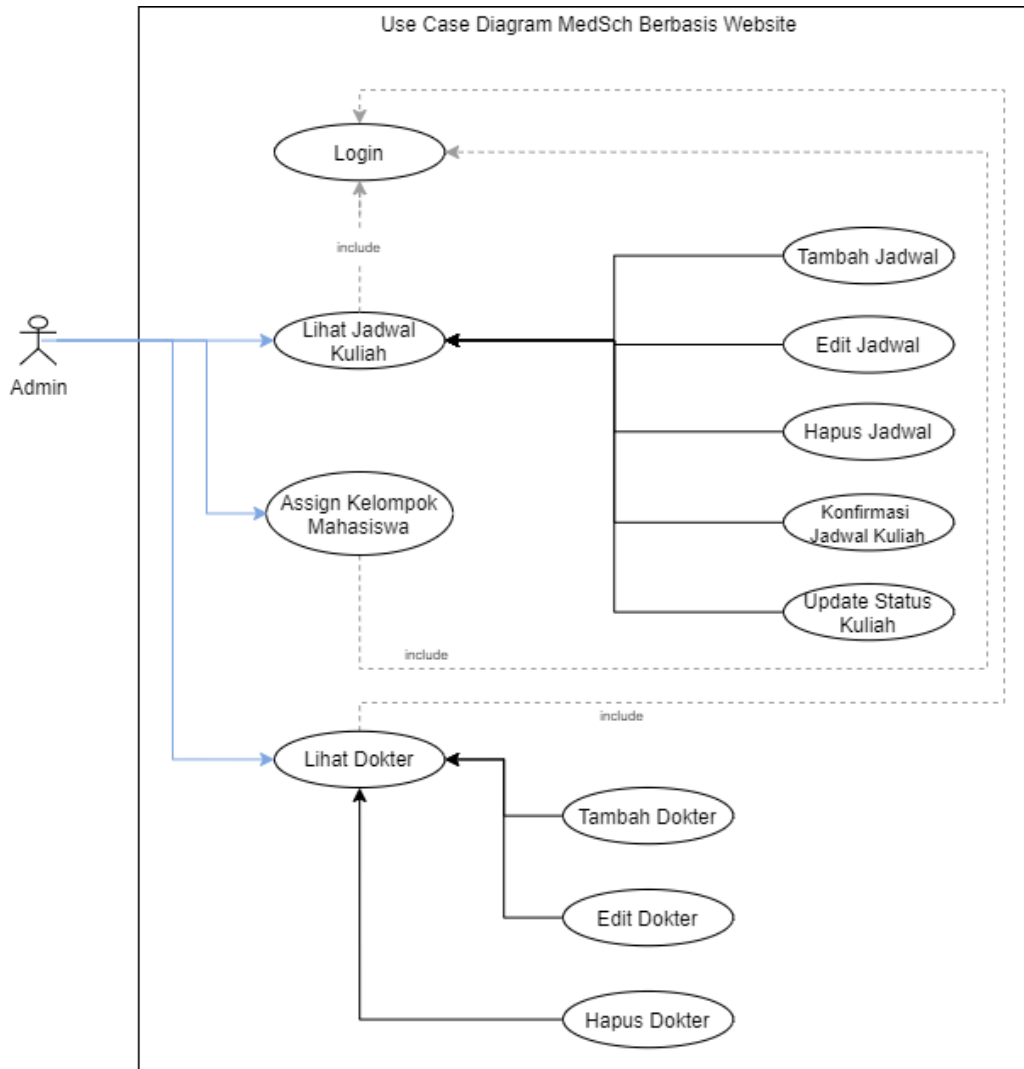
Gambar 3.3 menjelaskan *use case* diagram MedSch berbasis Android yang akan digunakan oleh mahasiswa dan dosen sebagai aktor dari sistem pemberitahuan jadwal kuliah. *Use case* tersebut dipetakan dari kebutuhan fungsionalitas aplikasi MedSch ini.



Gambar 3.3 *Use Case* Diagram MedSch Berbasis Android

Terdapat 6 *use case* diagram MedSch berbasis Android yang sudah dirancang berdasarkan dari 5 kebutuhan fungsional yaitu Registrasi Akun, *Login*, Lihat Jadwal, *Update* Status Kuliah, Lihat Dokter, Konfirmasi Jadwal Kuliah. Untuk menggunakan aplikasi MedSch berbasis Android ini, mahasiswa melakukan registrasi akun terlebih dahulu setelah itu *login* untuk menggunakan fitur (*use case*) yang ada pada aplikasi. Untuk dosen sendiri tidak perlu melakukan registrasi akun karena sudah dibuatkan akun secara langsung oleh admin, dosen melakukan *login* langsung untuk menggunakan fitur (*use case*) yang ada pada aplikasi. Untuk *use case Update* Status Kuliah hanya bisa digunakan oleh dosen saja karena *use case* ini bertujuan untuk memperbaharui status kuliah yang diajarkan oleh dosen jika dosen tersebut memiliki halangan ataupun kendala pada proses perkuliahan berlangsung.

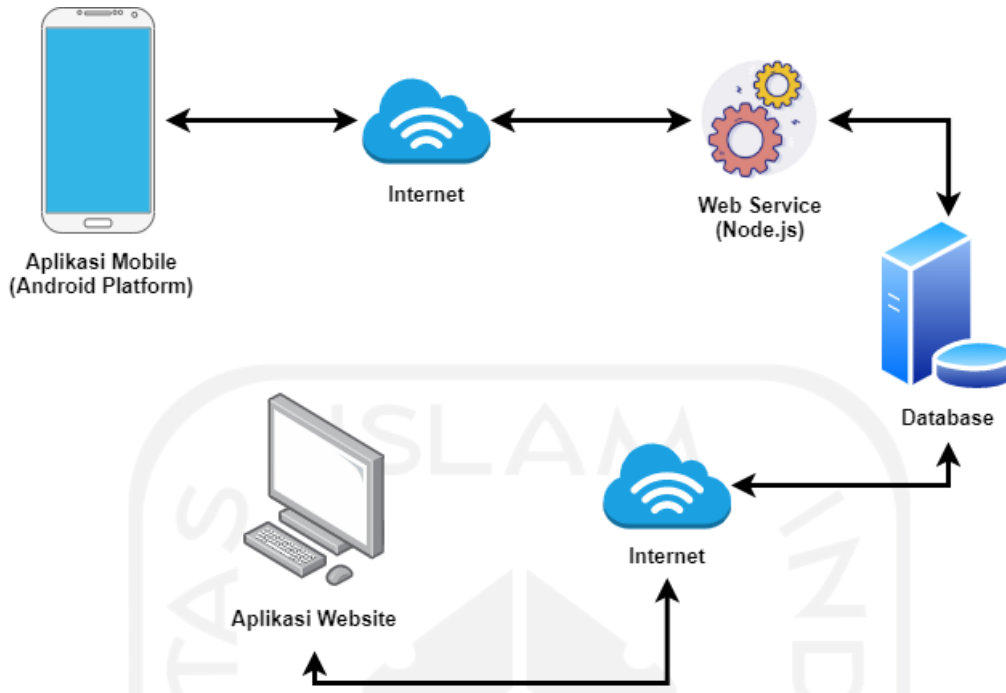
Gambar 3.4 menjelaskan *use case* diagram MedSch berbasis *website* yang akan digunakan oleh admin sebagai aktor dari sistem pemberitahuan jadwal kuliah. *Use case* tersebut dipetakan dari kebutuhan fungsionalitas aplikasi MedSch ini.



Gambar 3.4 *Use Case* Diagram MedSch Berbasis *Website*

Terdapat 12 *use case* diagram MedSch berbasis *website* yang sudah dirancang berdasarkan dari 5 kebutuhan fungsional yaitu *Login*, *Lihat Jadwal*, *Tambah Jadwal*, *Edit Jadwal*, *Hapus Jadwal*, *Konfirmasi Jadwal Kuliah*, *Update Status Kuliah*, *Assign Kelompok Mahasiswa*, *Lihat Dokter*, *Tambah Dokter*, *Edit Dokter*, *Hapus Dokter*.

Berikut rancangan arsitektur sistem aplikasi jadwal kuliah MedSch. Arsitektur sistem pada aplikasi ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Arsitektur Sistem Aplikasi MedSch

Pada Gambar 3.5 terdapat 2 aplikasi dalam arsitektur ini yaitu 1 aplikasi *mobile* dan 1 aplikasi *website*. Aplikasi *mobile* diimplementasikan ke perangkat *mobile* bertujuan untuk digunakan oleh pengguna, yaitu mahasiswa dan dosen FK UII. Aplikasi *website* diimplementasikan pada laptop/PC dan ditujukan kepada admin yaitu BAAK untuk mengelola data-data mengenai jadwal kuliah, data mahasiswa, dan data dosen. Untuk menjalankan kedua aplikasi ini, semua perangkat harus terhubung dengan koneksi internet. Dengan adanya internet, aplikasi *mobile* yang digunakan oleh mahasiswa dan dosen dapat mengirimkan dan menerima data ke dan atau dari *database* melalui *web service* (Node.js).

3.3 Perancangan

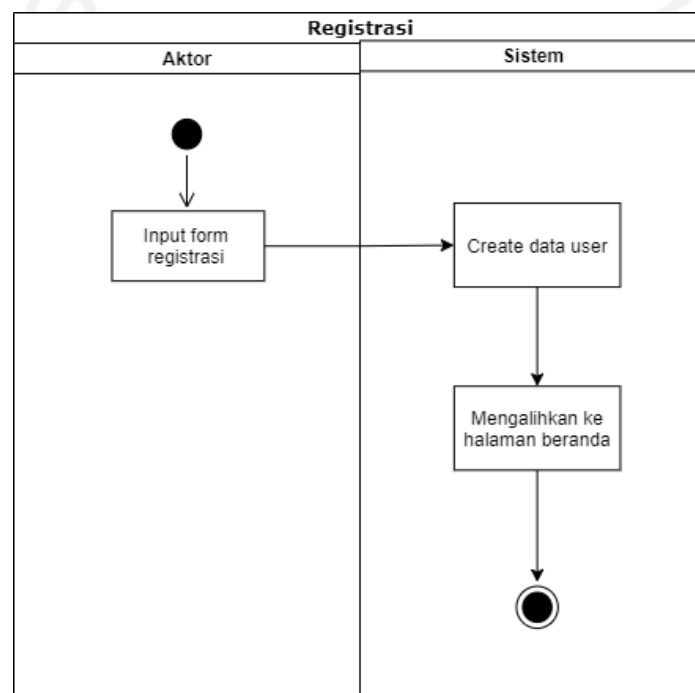
Perancangan dari sistem aplikasi jadwal kuliah sendiri untuk *platform* Android akan dibangun dengan Android Studio dan menggunakan bahasa pemrograman Java. Penetapan dalam batas minimal perangkat yang didukung pada aplikasi berbasis Android ini yaitu perangkat Android dengan versi 8.0 (nama kode nya yaitu *Oreo*). Untuk *platform website* akan dibangun dengan aplikasi besutan *Microsoft* yaitu *Visual Studio Code* dengan menggunakan bahasa pemrograman *native* PHP. Sebelum melakukan perancangan, peneliti akan menentukan *Activity diagram* terlebih dahulu, selanjutnya merancang desain *database* untuk sistem aplikasi jadwal kuliah.

3.3.1 Activity Diagram

Diagram aktivitas sering kali diibaratkan sebagai cara untuk menggambarkan logika prosedural, proses bisnis, dan alur kerja. *Activity diagram* disini menjabarkan bagaimana prosedur aplikasi tersebut dimulai dari awal penggunaan hingga aplikasi tersebut selesai digunakan. Berikut diagram aktivitas yang akan dijabarkan dalam setiap *use case* (UC).

3.3.1.1 UC-01 Registrasi Akun

Use case registrasi akun berisi proses mengenai pendaftaran akun yang dilakukan oleh aktor.

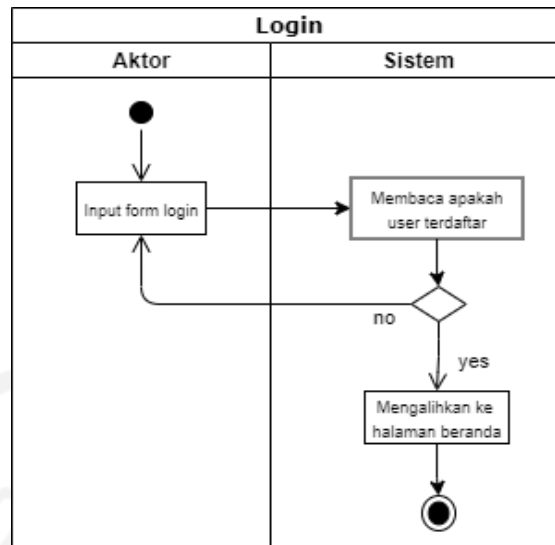


Gambar 3.6 Activity Diagram UC-01

Gambar 3.6 menjelaskan bahwa aktor mendaftar dengan menginput data diri untuk membuat akun, nantinya akan disimpan oleh sistem. Setelah membuat akun, aktor akan diarahkan ke halaman beranda aplikasi MedSch.

3.3.1.2 UC-02 Login Akun

Use case login akun berisi proses untuk mengakses aplikasi MedSch dengan memasukkan data diri yang sudah terdaftar.

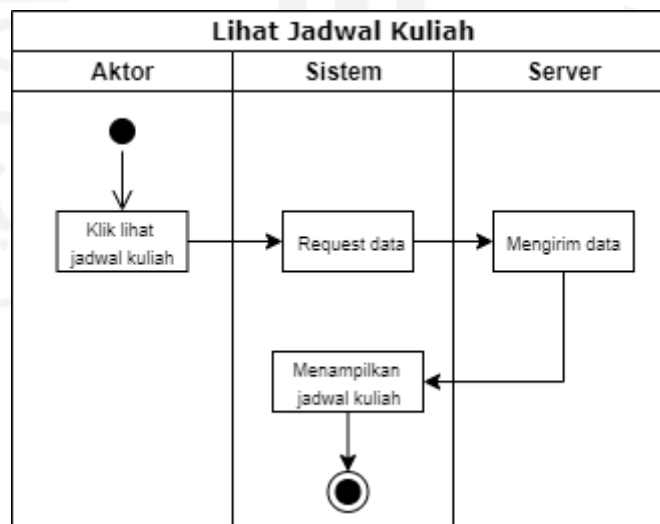


Gambar 3.7 Activity Diagram UC-02

Gambar 3.7 menjelaskan bahwa aktor menginput data diri yang sudah terdaftar oleh sistem. Setelah menginputkan data diri, maka aktor akan diarahkan ke halaman beranda aplikasi MedSch.

3.3.1.3 UC-03 Lihat Jadwal Kuliah

Use case lihat jadwal kuliah berisi proses untuk melihat jadwal kuliah yang ada pada aplikasi MedSch.

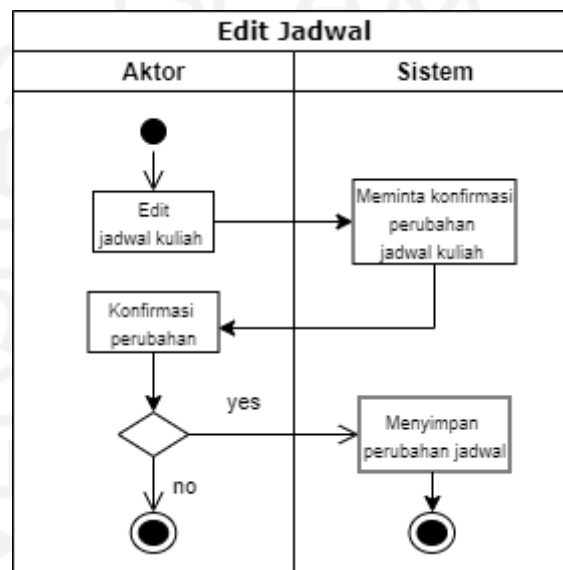


Gambar 3.8 Activity Diagram UC-03

Gambar 3.8 menjelaskan bahwa aktor dapat melihat informasi jadwal kuliah. Ketika aktor sudah masuk ke halaman beranda aplikasi MedSch, aktor dapat memilih lihat jadwal kuliah untuk melihat informasi jadwal kuliah yang berlangsung, aplikasi akan *request* data dan *server* akan mengirim request data yang diminta dan ditampilkan ke aktor.

3.3.1.4 UC-01 Edit Jadwal

Use case edit jadwal berisi proses untuk mengedit jadwal kuliah yang ada pada aplikasi MedSch.

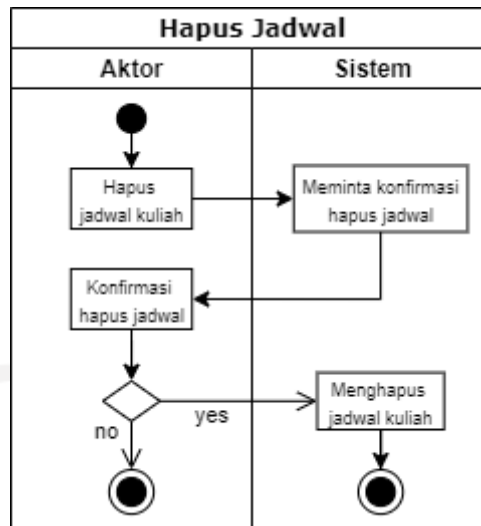


Gambar 3.9 Activity Diagram UC-04

Gambar 3.9 menjelaskan bahwa aktor dapat mengedit jadwal kuliah jika ada perubahan dan kekeliruan. Ketika aktor sudah masuk ke halaman lihat jadwal kuliah aplikasi MedSch, aktor dapat memilih opsi untuk mengedit jadwal kuliah. Setelah diedit, maka perubahan tersebut dikonfirmasi dahulu untuk memastikan perubahan yang sudah dibuat, terakhir yaitu menyimpan perubahan yang sudah dibuat.

3.3.1.5 UC-05 Hapus Jadwal

Use case hapus jadwal berisi proses untuk menghapus jadwal kuliah yang ada pada aplikasi MedSch.

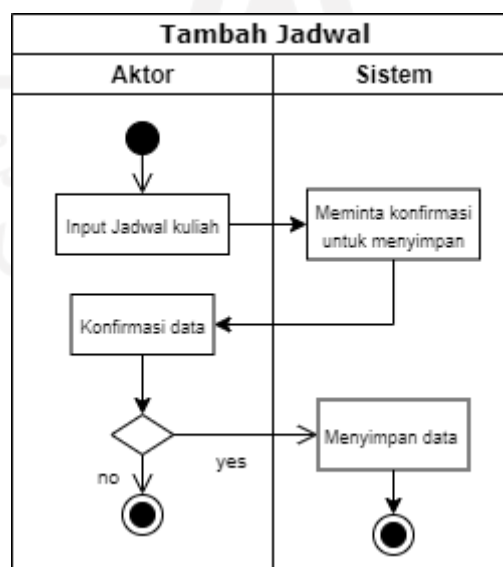


Gambar 3.10 Activity Diagram UC-05

Gambar 3.10 menjelaskan bahwa aktor dapat menghapus jadwal kuliah jika sudah tidak berlaku atau kesalahan data. Ketika aktor sudah masuk ke halaman lihat jadwal kuliah aplikasi MedSch, aktor dapat memilih opsi untuk menghapus jadwal kuliah. Setelah memilih data yang ingin dihapus tersebut perlu dikonfirmasi dahulu untuk memastikan jika ingin menghapus data, jika sudah dikonfirmasi, maka data tersebut akan dihapus.

3.3.1.6 UC-06 Tambah Jadwal

Use case tambah jadwal berisi proses untuk menambah jadwal kuliah yang ada pada aplikasi MedSch.

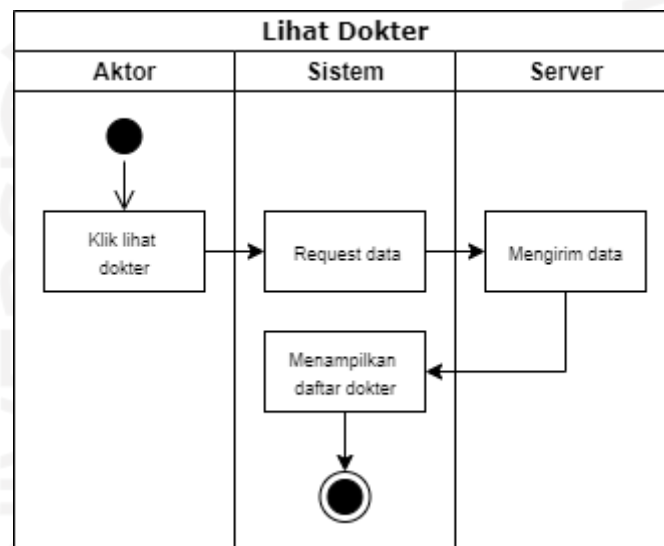


Gambar 3.11 Activity Diagram UC-06

Gambar 3.11 menjelaskan bahwa aktor dapat menambahkan jadwal baru. Aktor masuk ke halaman tambah jadwal kuliah aplikasi MedSch, setelah itu aktor mengisi data-data atau informasi terkait jadwal kuliah. Ketika sudah mengisi data atau informasi jadwal perlu dikonfirmasi dahulu untuk memastikan data atau informasi yang dituliskan sudah benar. Jika sudah dikonfirmasi maka data tersebut akan disimpan.

3.3.1.7 UC-07 Lihat Dokter

Use case lihat dokter berisi proses untuk melihat daftar dosen yang ada pada aplikasi MedSch.

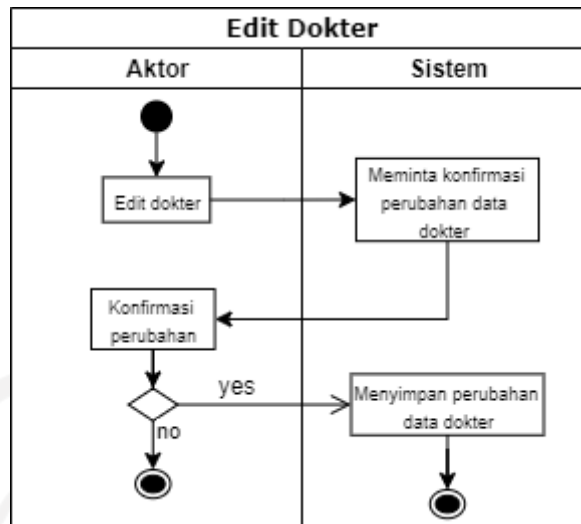


Gambar 3.12 Activity Diagram UC-07

Gambar 3.12 menjelaskan bahwa aktor dapat melihat informasi dosen yang mengajar. Ketika aktor sudah masuk ke halaman beranda aplikasi MedSch, aktor dapat memilih lihat dokter untuk melihat informasi mengenai dosen yang mengajar di FK UII, aplikasi akan merequest data dan *server* akan mengirim request data yang diminta dan ditampilkan ke aktor.

3.3.1.8 UC-08 Edit Dokter

Use case edit dokter berisi proses untuk mengedit data dosen yang ada pada aplikasi MedSch.

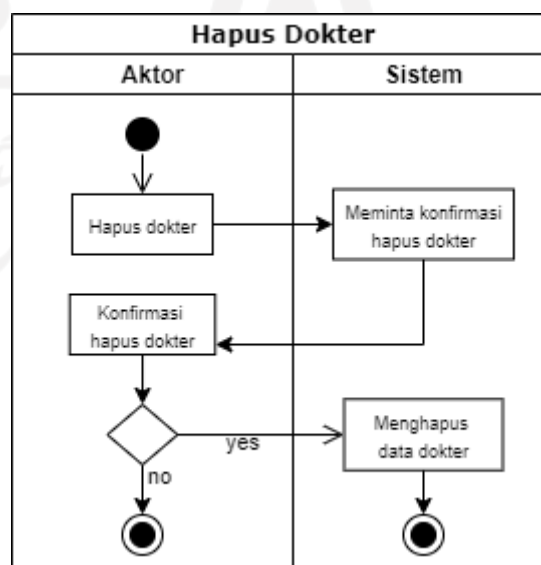


Gambar 3.13 Activity Diagram UC-08

Gambar 3.13 menjelaskan bahwa aktor dapat mengedit data dosen jika ada perubahan dan kekeliruan. Ketika aktor sudah masuk ke halaman lihat dokter aplikasi MedSch, aktor dapat memilih opsi untuk mengedit data dosen. Setelah diedit, maka perubahan tersebut dikonfirmasi dahulu untuk memastikan perubahan yang sudah dibuat, terakhir yaitu menyimpan perubahan yang sudah dibuat.

3.3.1.9 UC-09 Hapus Dokter

Use case hapus dokter berisi proses untuk menghapus data dosen yang ada pada aplikasi MedSch.

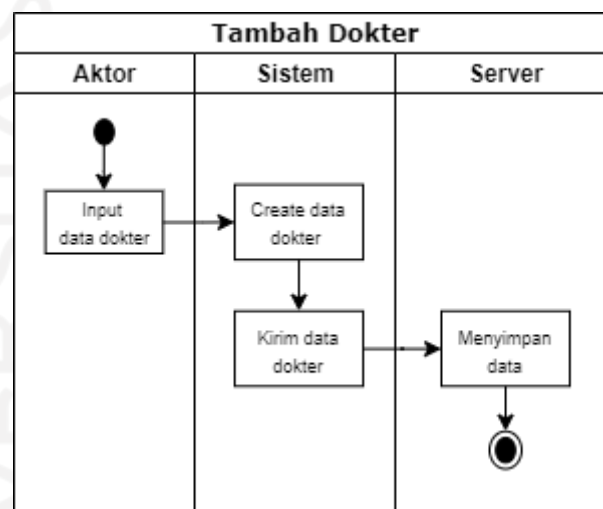


Gambar 3.14 Activity Diagram UC-09

Gambar 3.14 menjelaskan bahwa aktor dapat menghapus data dosen jika sudah tidak berlaku atau kesalahan data. Ketika aktor sudah masuk ke halaman lihat dokter aplikasi MedSch, aktor dapat memilih opsi untuk menghapus data dosen. Setelah memilih data yang ingin dihapus tersebut perlu dikonfirmasi dahulu untuk memastikan jika ingin menghapus data, jika sudah dikonfirmasi, maka data tersebut akan dihapus.

3.3.1.10 UC-10 Tambah Dokter

Use case tambah dokter berisi proses untuk menambah data dosen yang mengajar di FK UII pada aplikasi MedSch.

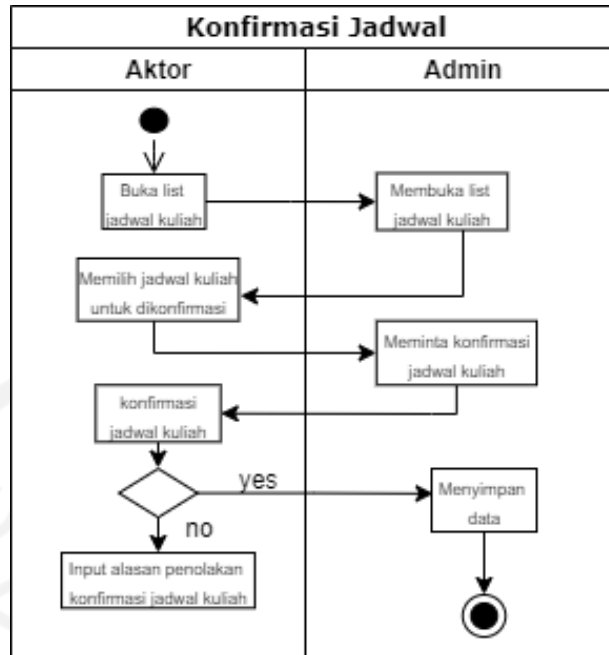


Gambar 3.15 Activity Diagram UC-10

Gambar 3.15 menjelaskan bahwa aktor dapat menambahkan data dosen yang terbaru. Aktor masuk ke halaman tambah dokter aplikasi MedSch, setelah itu aktor mengisi data-data atau informasi terkait dosen yang mengajar di FK UII. Ketika sudah mengisi data atau informasi, perlu dikonfirmasi dahulu untuk memastikan data atau informasi yang dituliskan sudah benar. Jika sudah dikonfirmasi maka data tersebut akan disimpan.

3.3.1.11 UC-11 Konfirmasi Jadwal Kuliah

Use case konfirmasi jadwal kuliah berisi proses untuk mengonfirmasi jadwal kuliah pada aplikasi MedSch.

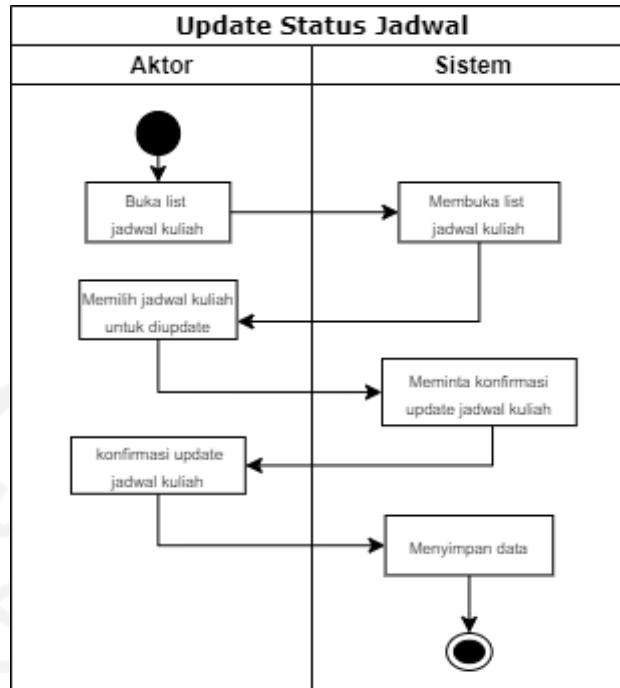


Gambar 3.16 Activity Diagram UC-11

Gambar 3.16 menjelaskan bahwa aktor dapat mengonfirmasi jadwal kuliah. Aktor masuk ke halaman konfirmasi jadwal aplikasi MedSch, setelah itu aktor dapat melihat list rancangan jadwal kuliah yang sudah dibuat oleh admin. Aktor dapat memilih jadwal yang sekiranya bersedia untuk mengajar pada tanggal yang sudah ditentukan, aktor akan mengonfirmasi rancangan jadwal kuliah. Setelah itu hasil jadwal yang sudah konfirmasi akan tersimpan pada sistem. Jika aktor tidak bisa mengajar di tanggal yang sudah ditentukan, maka aktor akan memberikan alasan mengapa aktor tidak bisa mengajar.

3.3.1.12 UC-12 Update Status Kelas

Use case update status kelas berisi proses untuk memperbaharui jadwal kuliah pada aplikasi MedSch.

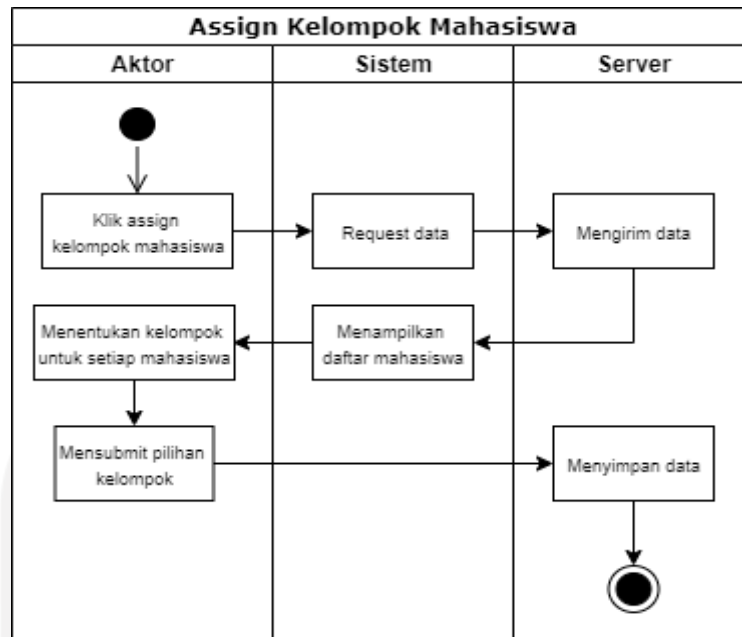


Gambar 3.17 Activity Diagram UC-012

Gambar 3.17 menjelaskan bahwa aktor dapat memperbaharui status kelas jika ada perubahan. Aktor masuk ke halaman lihat jadwal aplikasi MedSch, setelah itu aktor dapat melihat list jadwal kuliah yang sudah dikonfirmasi oleh dosen dan dibuat oleh admin. Aktor dapat merubah status jadwal yang sekiranya berhalangan untuk mengajar pada tanggal yang sudah dikonfirmasi. Setelah status jadwal sudah diperbaharui, maka data jadwal akan tersimpan pada sistem.

3.3.1.13 UC-13 Assign Kelompok Mahasiswa

Use case assign kelompok mahasiswa berisi proses untuk mengelompokkan tiap mahasiswa untuk pembagian jadwal kuliah pada aplikasi MedSch.

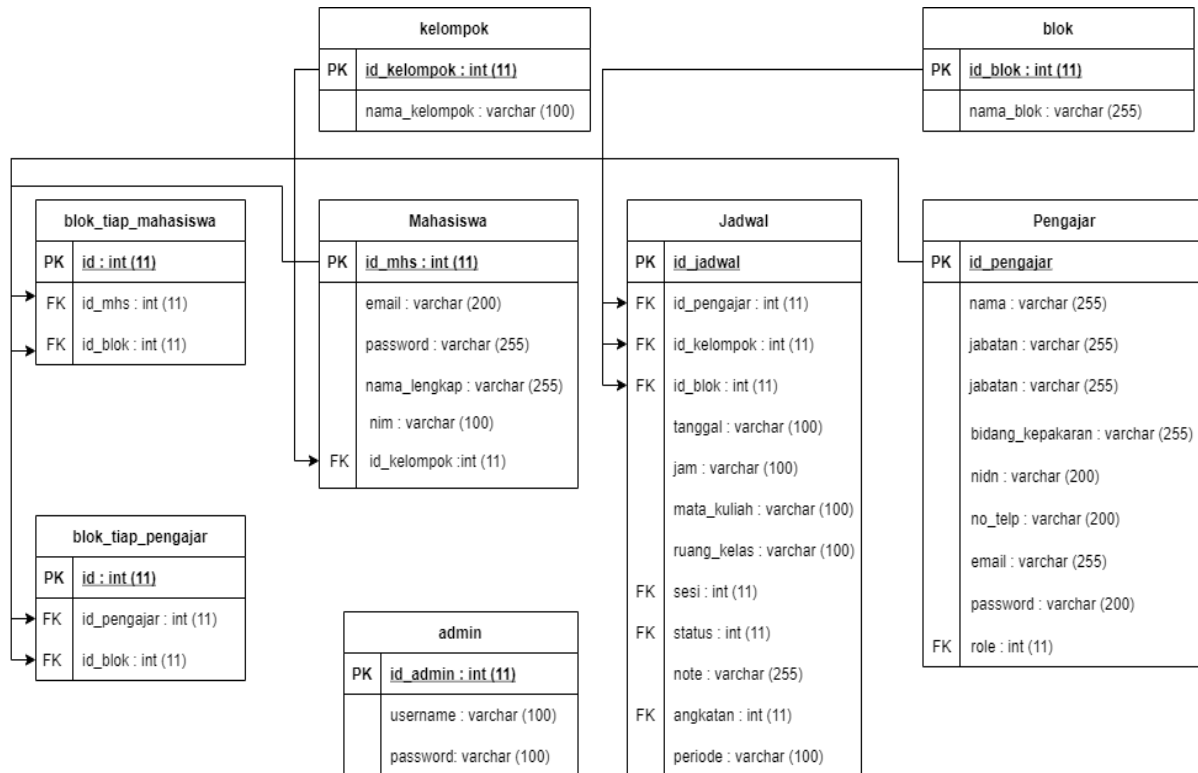


Gambar 3.18 *Activity Diagram* UC-013

Gambar 3.18 menjelaskan bahwa aktor menentukan kelompok pada tiap-tiap mahasiswa untuk pembagian jadwal kuliah. Aktor masuk ke halaman *assign* kelompok mahasiswa aplikasi MedSch, aktor dapat melihat daftar mahasiswa yang sudah terdaftar dari aplikasi MedSch. Aktor akan mengelompokkan tiap mahasiswa untuk mempermudah dalam pendistribusian jadwal kuliah yang sudah ditentukan. Setelah pengelompokkan selesai maka data tersebut akan tersimpan.

3.3.2 Perancangan *Database*

Perancangan *Database* adalah rancangan untuk memastikan data apa saja yang diperlukan untuk pengembangan sebuah sistem, serta memudahkan sistem dalam menyimpan, membaca, dan mengubah data. Dalam merancang *database*, penulis menggunakan basis data yang digunakan untuk merancang aplikasi jadwal kuliah ini menggunakan MySQL. Terdapat 8 entitas yang akan ada pada rancangan *database* yaitu Jadwal, Mahasiswa, Pengajar, Kelompok, Blok, Admin, blok_tiap_mahasiswa, dan blok_tiap_pengajar. Keterangan relasi antara entitas satu dengan lainnya pada aplikasi yang akan dibangun bisa dilihat pada diagram *Entity Relationship Diagram* (ERD) Gambar 3.19.



Gambar 3.19 Entity Relationship Diagram

Pada Gambar 3.19 merupakan ilustrasi gambar kumpulan tabel dari basis data yang akan dipakai oleh penulis sebagai struktur data yang akan digunakan dalam sistem aplikasi jadwal kuliah ini.

3.4 Implementasi

Tahapan ini dilakukan untuk mengimplementasikan hasil rancangan yang telah diperoleh pada tahap sebelumnya untuk dikembangkan menjadi sebuah aplikasi dengan bahasa pemrograman, dalam pembuatan perangkat lunak dibagi menjadi modul-modul yang lebih kecil agar mudah dibangun dan hasil akhirnya akan digabungkan dalam tahap berikutnya. Tahap ini akan dilanjutkan ke proses pengujian dan pengecekan perihal fungsionalitas modul yang sudah dikerjakan, apakah sudah memenuhi tolok ukur yang diharapkan atau belum. Akan ada 3 aplikasi yang akan dikembangkan, sesuai dengan aktor yang sudah ditentukan yaitu aplikasi berbasis Android akan dilakukan oleh aktor yaitu mahasiswa dan dosen, sedangkan aplikasi berbasis *website* akan dilakukan oleh aktor yaitu BAAK (atau admin).

Untuk pengembangan Android akan dibangun dengan bahasa pemrograman java menggunakan perangkat lunak Android Studio, karena Android Studio merupakan IDE

pemrograman resmi dari Google yang dikembangkan oleh IntelliJ. Mampu membuild, debug, dan testing langsung tanpa perangkat lainnya. Arsitektur sendiri sangat jelas membuat program lebih terstruktur sehingga dikemudian hari jika program tersebut ingin dikembangkan lebih lanjut bisa dilanjutkan di perangkat lunak Android Studio.

Untuk pengembangan *website* sendiri akan dibangun dengan bahasa PHP karena PHP paling banyak digunakan dan dikembangkan oleh pengembang. PHP sendiri mudah dipelajari karena PHP banyak menyediakan dokumentasi lengkap tentang fungsinya beserta contoh-contohnya. Peneliti akan membangun *website* menggunakan perangkat lunak dari *Microsoft* yaitu *Visual Studio Code*.

Untuk pengembangan *database* akan dibangun dengan MySQL, namun MySQL hanya bisa diakses oleh *website*, sedangkan Android tidak bisa membaca *query* MySQL secara langsung karena Android membutuhkan *query* khusus *Endpoint* berupa API. Agar *database* MySQL juga bisa diakses oleh Android, maka dibutuhkan *middleware* sebagai perantara Android dan MySQL yaitu menggunakan *Representational State Transfer* API (REST API) berupa Node.js. Node.js bertugas untuk *handler server* antara aplikasi Android dan *website*. Node.js merupakan salah satu platform yang sering digunakan para pengembang untuk membangun *website* dan memiliki kemampuan lebih dalam pengembangan aplikasi.

3.5 Pengujian

Pengujian dari aplikasi akan menggunakan metode pengujian black box untuk menguji fungsionalitas aplikasi. Salah satu metode pengujian yang berfokus pada spesifikasi fungsionalitas dari perangkat lunak disebut *Black Box Testing* (Nurudin, 2019). Pengujian ini memberikan gambaran atas sekumpulan kondisi masukan dan melakukan pengujian pada uraian fungsional program. Untuk menguji kesalahan yang tidak dapat dicakup oleh *White Box Testing*, maka solusi lainnya dapat menggunakan *Black Box Testing* (Vikasari, 2018).

3.5.1 Black Box Testing

Fungsionalitas dari aplikasi diuji menggunakan metode *Black Box Testing*. Metode ini dipilih untuk membantu validasi fungsionalitas keseluruhan dari sistem. Pengujian *black box* didasarkan pada kebutuhan pengguna, sehingga memudahkan untuk mengidentifikasi persyaratan yang tidak lengkap atau tidak diinginkan dan mengatasinya lebih lanjut. Pengujian *black box* dilakukan berdasarkan perspektif pengguna akhir. Pentingnya pengujian kotak hitam adalah menangani masukan yang valid dan tidak valid dari sudut pandang *user*. Selain itu,

Manfaat utama dari pengujian *black box* adalah penguji tidak perlu mengetahui bahasa pemrograman dan implementasinya. Dalam pengujian *black box*, baik program maupun tester tidak bergantung satu sama lain. Keuntungan lainnya adalah pengujian dilakukan dari sudut pandang pengguna. Keuntungan substansial dari pengujian *black box* adalah membantu mengungkap ambiguitas atau inkonsistensi dalam spesifikasi persyaratan. Fungsionalitas yang diujikan berupa fitur-fitur dalam aplikasi.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil Implementasi

Penelitian ini telah berhasil melakukan perancangan dan pengembangan aplikasi Android dan *Website* MedSch pada Sistem Informasi Jadwal Kuliah. Peneliti telah membuat *database* yang akan digunakan untuk menampung semua data-data atau informasi yang digunakan dari aplikasi MedSch. Desain *user interface* berupa wireframe yang digunakan untuk acuan dalam pembuatan tampilan aplikasi berbasis Android maupun *website* MedSch.

4.1.1 Database

Database MedSch menggunakan MySQL dan diberi nama 'med_db'. *Database* digunakan untuk menampung semua data-data atau informasi yang digunakan dari aplikasi MedSch. Hasil *database* MedSch dapat dilihat pada Gambar 4.1.

Table	Action
<input type="checkbox"/> admin	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> blok	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> blok_tiap_mahasiswa	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> blok_tiap_pengajar	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> jadwal	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> kelompok	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> mahasiswa	★ Browse Structure Search Insert Empty Drop
<input type="checkbox"/> pengajar	★ Browse Structure Search Insert Empty Drop
8 tables	Sum

Gambar 4.1 *Layout Database* MedSch

a. Tabel Jadwal

Rincian pada tabel jadwal yaitu id_jadwal, id_pengajar, id_kelompok, id_blok, tanggal, jam, mata_kuliah, ruang_kelas, sesi, status, *note*, angkatan, periode. Tabel jadwal pada *database* MedSch dapat dilihat pada Gambar 4.2.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_jadwal 🔑	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	id_pengajar	int(11)		No	None		
<input type="checkbox"/>	3	id_kelompok	int(11)		No	None		
<input type="checkbox"/>	4	id_blok	int(11)		No	None		
<input type="checkbox"/>	5	tanggal	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	6	jam	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	7	mata_kuliah	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	8	ruang_kelas	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	9	sesi	int(11)		No	None		
<input type="checkbox"/>	10	status	int(11)		No	None		
<input type="checkbox"/>	11	note	varchar(255)	utf8mb4_general_ci	Yes	NULL		
<input type="checkbox"/>	12	angkatan	int(11)		No	None		
<input type="checkbox"/>	13	periode	varchar(100)	utf8mb4_general_ci	No	None		

Gambar 4.2 Tabel Jadwal

b. Mahasiswa

Rincian pada tabel mahasiswa yaitu *id_mhs*, *email*, *password*, *nama_lengkap*, *nim*, *id_kelompok*. Tabel mahasiswa pada *database* MedSch dapat dilihat pada Gambar 4.3.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_mhs 🔑	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	email	varchar(200)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	3	password	varchar(255)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	4	nama_lengkap	varchar(255)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	5	nim	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	6	id_kelompok	int(11)		Yes	NULL		

Gambar 4.3 Tabel Mahasiswa

c. Pengajar

Rincian pada tabel pengajar yaitu *id_pengajar*, *nama*, *jabatan*, *bidang_kepakaran*, *nidn*, *no_telp*, *email*, *password*, *role*. Tabel pengajar pada *database* MedSch dapat dilihat pada Gambar 4.4.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_pengajar 🔑	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	nama	varchar(255) utf8mb4_general_ci		No	None		
<input type="checkbox"/>	3	jabatan	varchar(255) utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	4	bidang_kepakaran	varchar(255) utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	5	nidn	varchar(200) utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	6	no_telp	varchar(200) utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	7	email	varchar(255) utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	8	password	varchar(200) utf8mb4_general_ci		Yes	NULL		
<input type="checkbox"/>	9	role	int(11)		No	None		

Gambar 4.4 Tabel Pengajar

d. Kelompok

Rincian pada tabel kelompok yaitu `id_kelompok` dan `nama_kelompok`. Tabel kelompok pada *database* MedSch dapat dilihat pada Gambar 4.5.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_kelompok 🔑	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	nama_kelompok	varchar(100) utf8mb4_general_ci		No	None		

Gambar 4.5 Tabel Kelompok

e. Blok

Rincian pada tabel blok yaitu `id_blok`, `nama_blok`, `masa_blok`. Tabel blok pada *database* MedSch dapat dilihat pada Gambar 4.6.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_blok 🔑	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	nama_blok	varchar(255) utf8mb4_general_ci		No	None		
<input type="checkbox"/>	3	masa_blok	varchar(255) utf8mb4_general_ci		No	None		

Gambar 4.6 Tabel Blok

f. Admin

Rincian pada tabel admin yaitu `id_admin`, `username`, `password`. Tabel admin pada *database* MedSch dapat dilihat pada Gambar 4.7.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id_admin	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	username	varchar(100)	utf8mb4_general_ci	No	None		
<input type="checkbox"/>	3	password	varchar(100)	utf8mb4_general_ci	No	None		

Gambar 4.7 Tabel Admin

g. blok_tiap_mahasiswa

Rincian pada tabel blok_tiap_mahasiswa yaitu id, id_mhs, id_blok. Tabel blok_tiap_mahasiswa pada *database* MedSch dapat dilihat pada Gambar 4.8.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	id_mhs	int(11)		No	None		
<input type="checkbox"/>	3	id_blok	int(11)		No	None		

Gambar 4.8 Tabel blok_tiap_mahasiswa

h. blok_tiap_pengajar

Rincian pada tabel blok_tiap_pengajar yaitu id, id_pengajar, id_blok. Tabel blok_tiap_pengajar pada *database* MedSch dapat dilihat pada Gambar 4.9.

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
<input type="checkbox"/>	1	id	int(11)		No	None		AUTO_INCREMENT
<input type="checkbox"/>	2	id_pengajar	int(11)		No	None		
<input type="checkbox"/>	3	id_blok	int(11)		No	None		

Gambar 4.9 Tabel blok_tiap_pengajar

4.1.2 User Interface

User interface MedSch yang pertama dibagi menjadi 2 yaitu *user interface* untuk mahasiswa dan untuk dosen, *user interface* ini dapat diakses menggunakan *smartphone* Android dan yang kedua yaitu *user interface* MedSch untuk admin berbasis *website* yang diakses melalui *browser* di komputer. Untuk tahapan yang dilakukan dalam perancangan desain antarmuka pada penelitian ini yaitu *wireframe*, yang menentukan komposisi desain

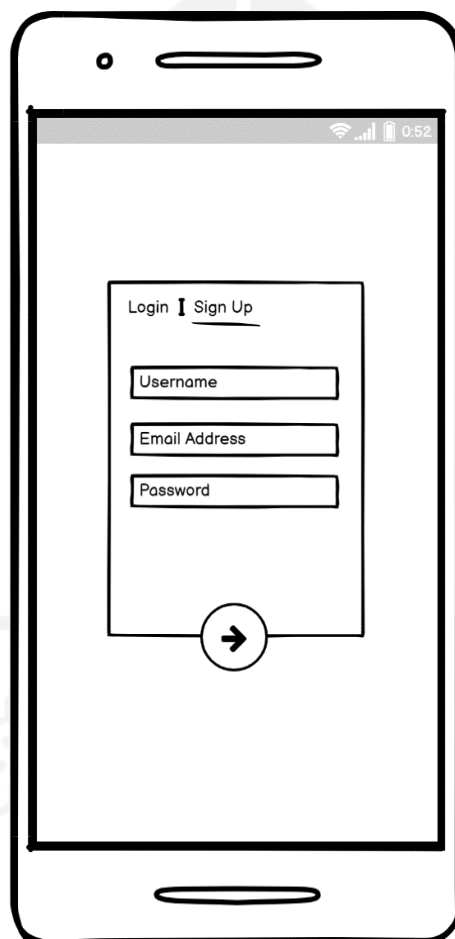
aplikasi dan elemen apa saja yang digunakan. Adapun gambaran yang telah dibuat sebagai berikut:

4.1.2.1 *User Interface MedSch Berbasis Android Mahasiswa*

Perancangan *user interface* MedSch berbasis *android* mahasiswa bertujuan memudahkan peneliti untuk membuat aplikasi. Berikut beberapa halaman perancangan *user interface* untuk aplikasi:

a. Halaman Registrasi Akun Mahasiswa

Halaman registrasi akun mahasiswa berisi tampilan dan elemen yang digunakan untuk menggambarakan halaman registrasi akun.

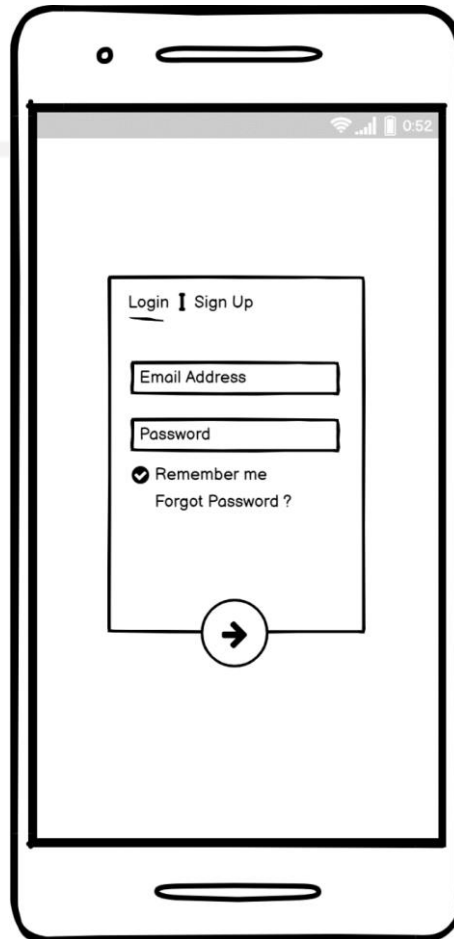


Gambar 4.10 Halaman Registrasi Akun Mahasiswa

Pada Gambar 4.10 halaman registrasi akun berbasis Android untuk mahasiswa bertujuan untuk mendaftarkan data diri mahasiswa agar bisa menggunakan fitur yang ada dalam aplikasi MedSch.

b. Halaman *Login* Mahasiswa

Halaman *login* mahasiswa berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman *login*.

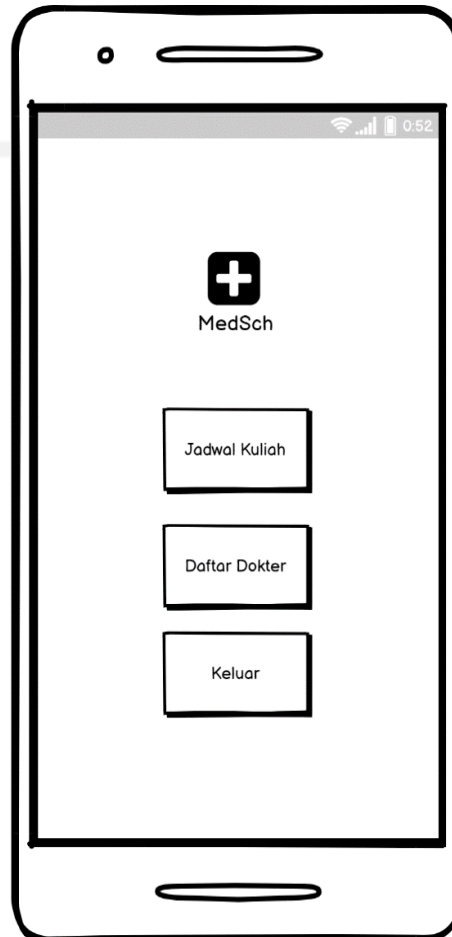


Gambar 4.11 Halaman *Login* Mahasiswa

Pada Gambar 4.11 halaman *login* untuk mahasiswa disini bertujuan untuk memverifikasi data sebelum menggunakan aplikasi MedSch dengan memasukkan data diri mahasiswa yang sudah didaftarkan pada halaman registrasi akun agar bisa menggunakan fitur yang ada didalam aplikasi MedSch.

c. Halaman Beranda Mahasiswa

Halaman beranda mahasiswa berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman beranda.

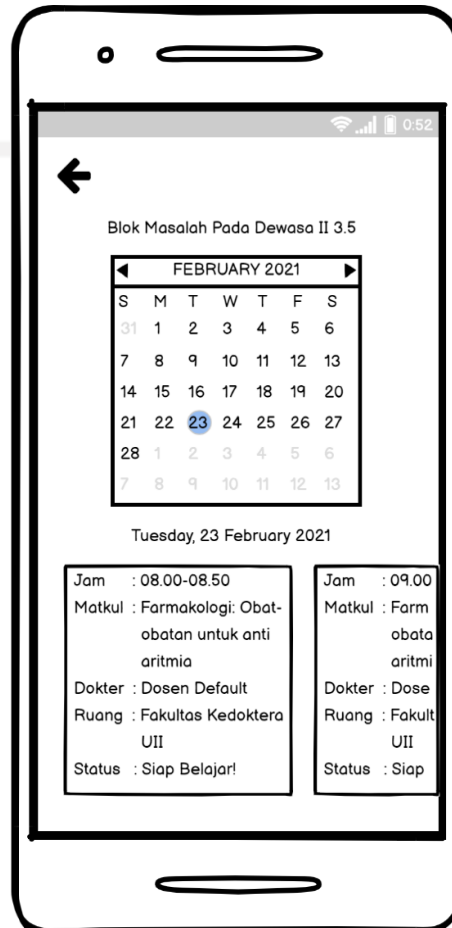


Gambar 4.12 Halaman Beranda Mahasiswa

Pada halaman beranda aplikasi untuk mahasiswa menampilkan fitur yang tertera pada Gambar 4.12 yaitu Jadwal Kuliah dan Daftar Dokter

d. Halaman Jadwal Kuliah

Halaman jadwal kuliah mahasiswa berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman jadwal kuliah.

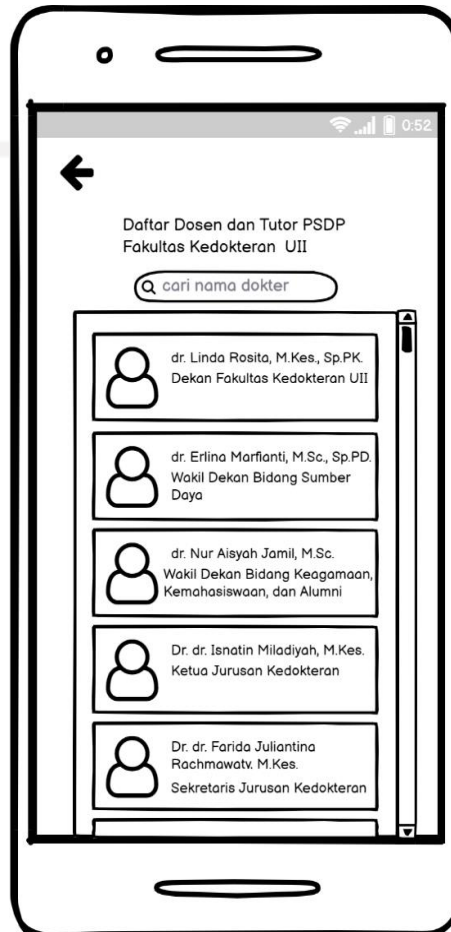


Gambar 4.13 Halaman Jadwal Kuliah Mahasiswa

Pada Gambar 4.13 adalah halaman jadwal kuliah mahasiswa. Halaman ini menampilkan jadwal kuliah berdasarkan sistem Blok yang sedang ditempuh oleh mahasiswa bersangkutan.

e. Halaman Daftar Dokter

Halaman daftar dokter mahasiswa berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman daftar dokter.



Gambar 4.14 Halaman Daftar Dokter Mahasiswa

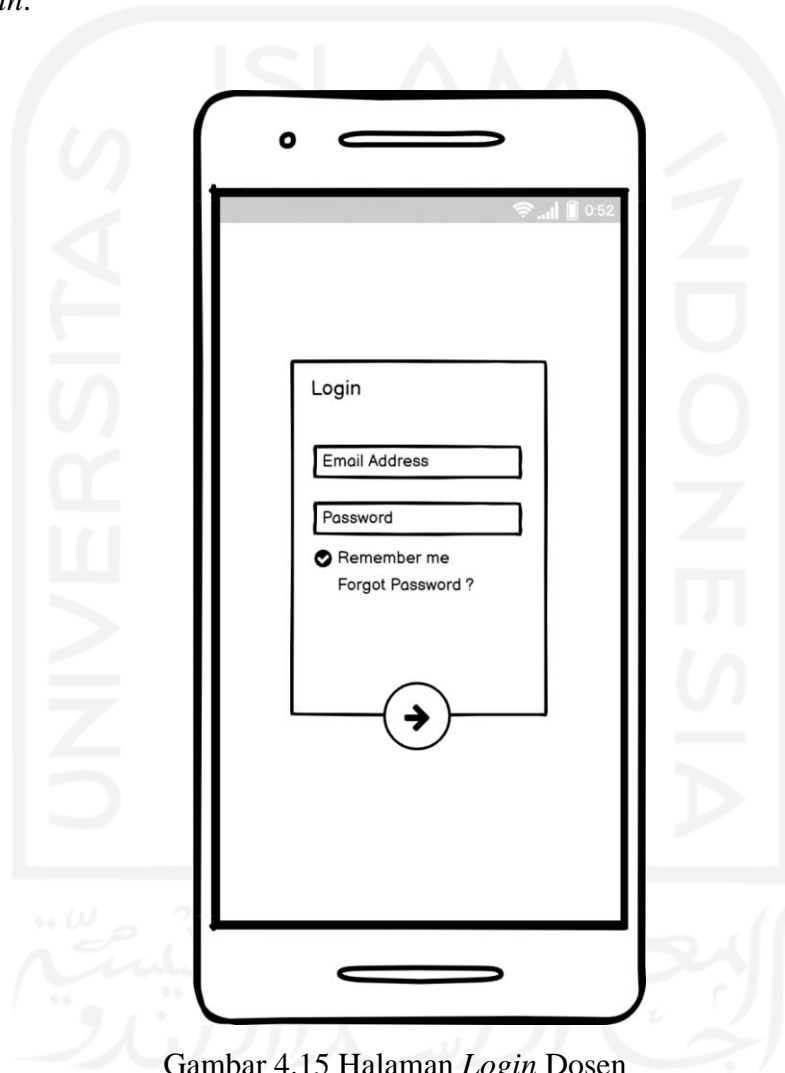
Pada Gambar 4.14 adalah halaman daftar dokter mahasiswa. Halaman ini menampilkan daftar dokter yang mengajar di Fakultas Kedokteran UII.

4.1.2.2 *User Interface MedSch Berbasis Android Dosen*

Perancangan *user interface* MedSch berbasis Android dosen bertujuan memudahkan peneliti untuk membuat aplikasi. Berikut beberapa halaman perancangan *user interface* untuk aplikasi:

a. Halaman *Login* Dosen

Halaman *login* dosen berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman *login*.

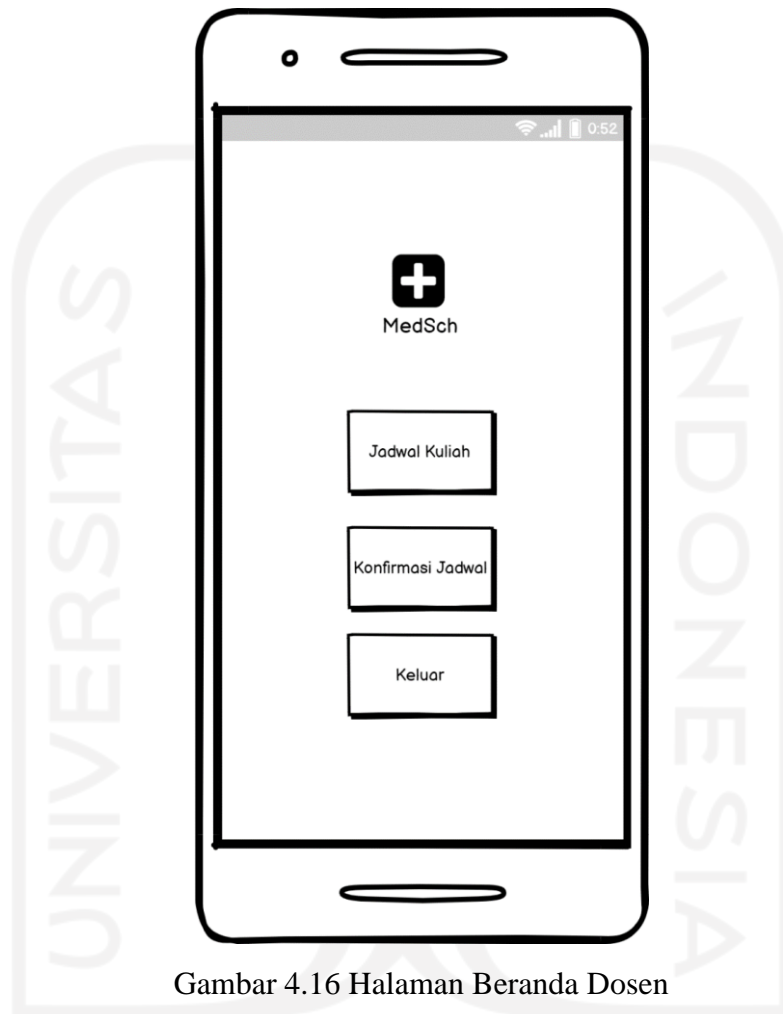


Gambar 4.15 Halaman *Login* Dosen

Pada Gambar 4.15 Halaman *login* Android untuk dosen disini bertujuan untuk memverifikasi data sebelum menggunakan aplikasi MedSch dengan memasukkan data diri dosen yang sudah didaftarkan oleh admin agar bisa menggunakan fitur yang ada didalam aplikasi MedSch

b. Halaman Beranda Dosen

Halaman beranda dosen berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman beranda.

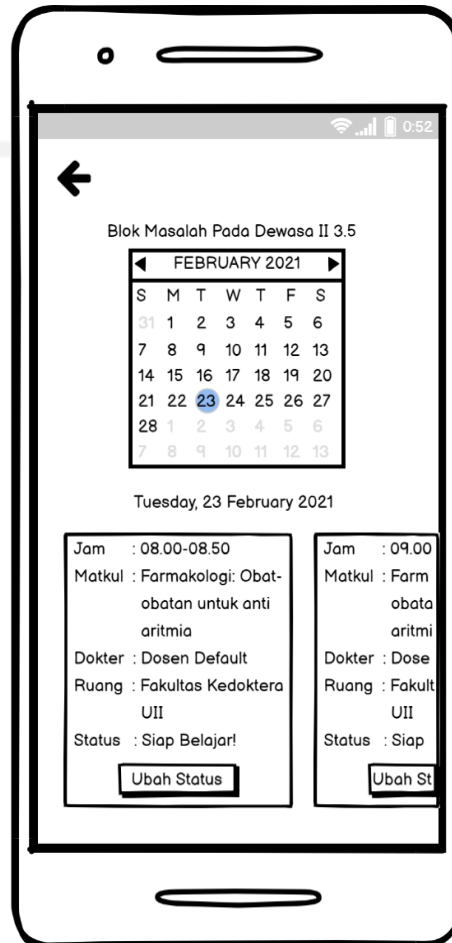


Gambar 4.16 Halaman Beranda Dosen

Pada halaman beranda aplikasi untuk dosen menampilkan fitur yang tertera pada Gambar 4.16 yaitu Jadwal Kuliah dan Konfirmasi Jadwal.

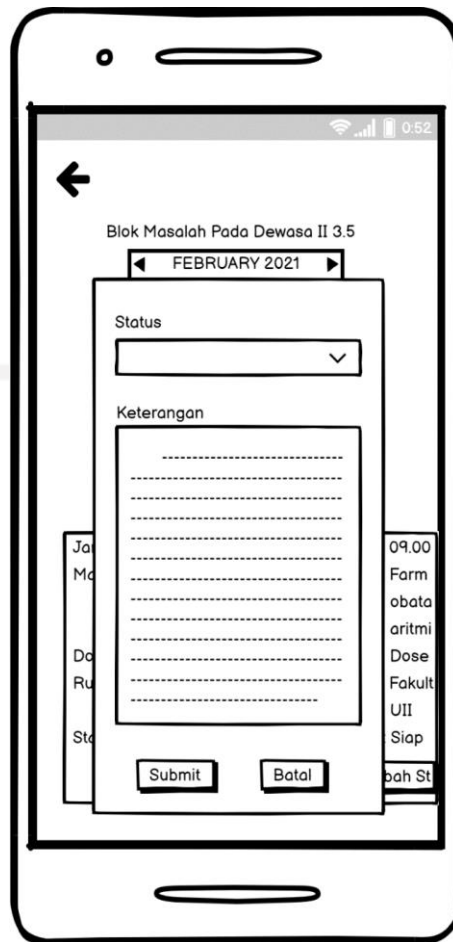
c. Halaman Jadwal Kuliah

Halaman jadwal kuliah dosen berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman jadwal kuliah.



Gambar 4.17 Halaman Jadwal Kuliah Dosen

Pada Gambar 4.17 adalah halaman jadwal kuliah dosen. Halaman ini menampilkan jadwal kuliah berdasarkan sistem Blok yang sedang ditugaskan oleh dosen bersangkutan.

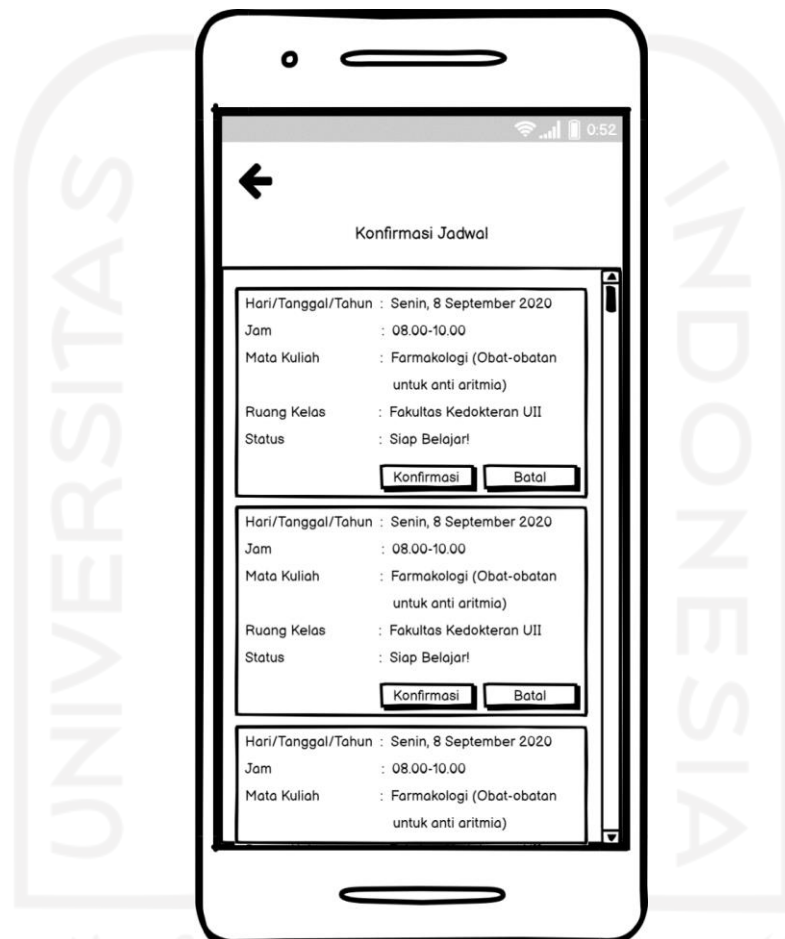


Gambar 4.18 Halaman *Update* Status Kuliah Dosen

Pada Gambar 4.18 halaman *update* status kuliah dosen, diberikan fitur *update* status kuliah dengan tujuan jika dosen memiliki kendala untuk mengajar pada hari yang sudah dijadwalkan, dosen juga bisa memberikan keterangan pada ubah status untuk memberi penjelasan mengapa kuliah tersebut ditunda.

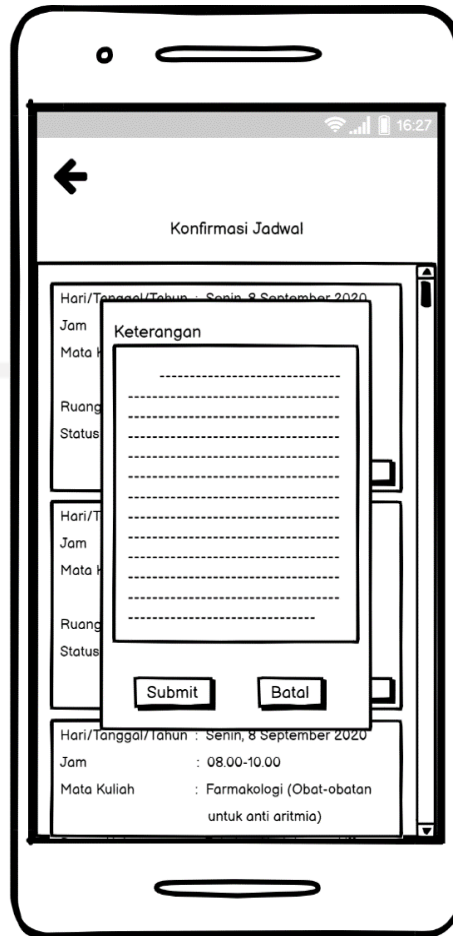
d. Halaman Konfirmasi Jadwal

Halaman konfirmasi jadwal dosen berisi tampilan dan elemen yang digunakan untuk menggamabarkan halaman konfirmasi jadwal.



Gambar 4.19 Halaman Konfirmasi Jadwal Dosen

Pada Gambar 4.19 adalah halaman konfirmasi jadwal dosen. Halaman ini menampilkan rancangan jadwal kuliah yang sudah dibuat oleh admin, untuk memastikan dosen bisa mengajar pada waktu yang sudah ditentukan oleh admin.



Gambar 4.20 Halaman Konfirmasi Jadwal Dokter

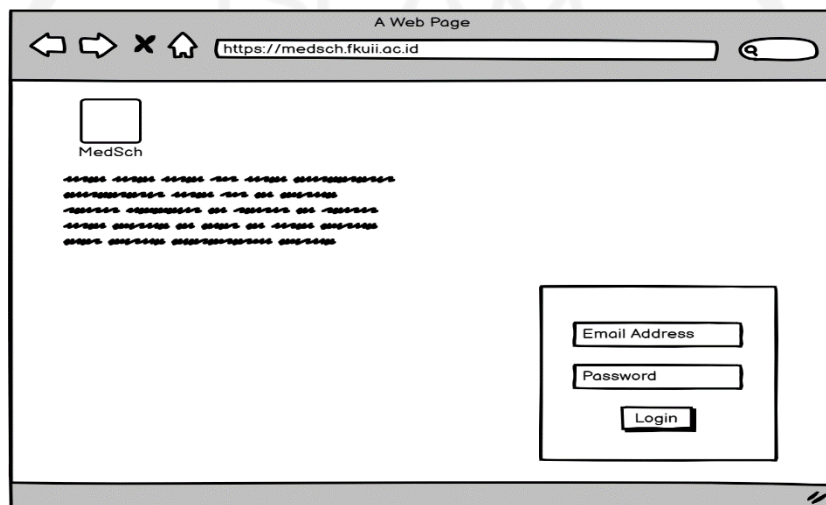
Pada Gambar 4.20 ini, diberikan fitur konfirmasi dan juga batal dengan tujuan jika dosen mampu untuk mengajar pada jadwal yang sudah ditentukan, dosen bisa menekan ikon konfirmasi. Namun dosen bisa menekan ikon batal jika dosen tidak bisa mengajar pada waktu yang sudah ditentukan dengan memberikan keterangan yang jelas.

4.1.2.3 User Interface MedSch Berbasis Website Admin

Perancangan *user interface* MedSch berbasis *website* admin bertujuan memudahkan peneliti untuk membuat *website*. Berikut beberapa halaman perancangan *user interface* untuk *website* :

a. Halaman *Login* Admin

Halaman *login* admin berisi tampilan dan elemen yang digunakan untuk menggambar halaman *login*.

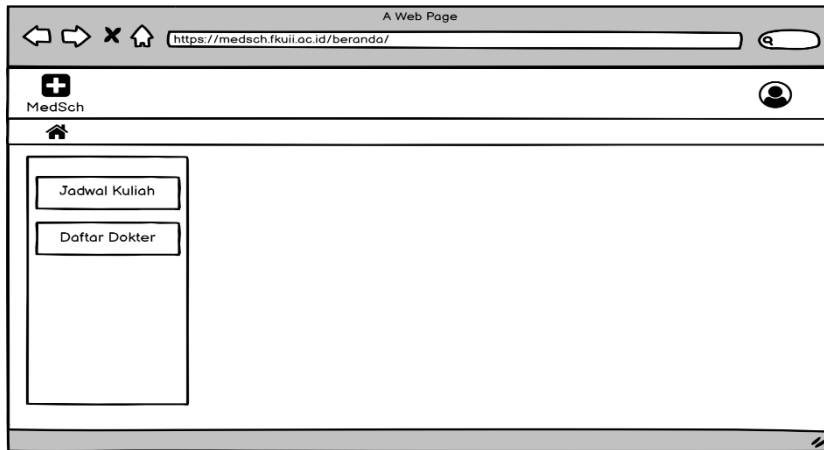


Gambar 4.21 Halaman *Login* Admin

Pada Gambar 4.21 halaman *login website* untuk admin disini bertujuan untuk memverifikasi data sebelum menggunakan aplikasi MedSch dengan memasukkan data diri admin yang sudah terdaftar agar bisa menggunakan fitur yang ada di dalam *website* MedSch.

b. Halaman Beranda Admin

Halaman beranda admin berisi tampilan dan elemen yang digunakan untuk menggambar halaman beranda.

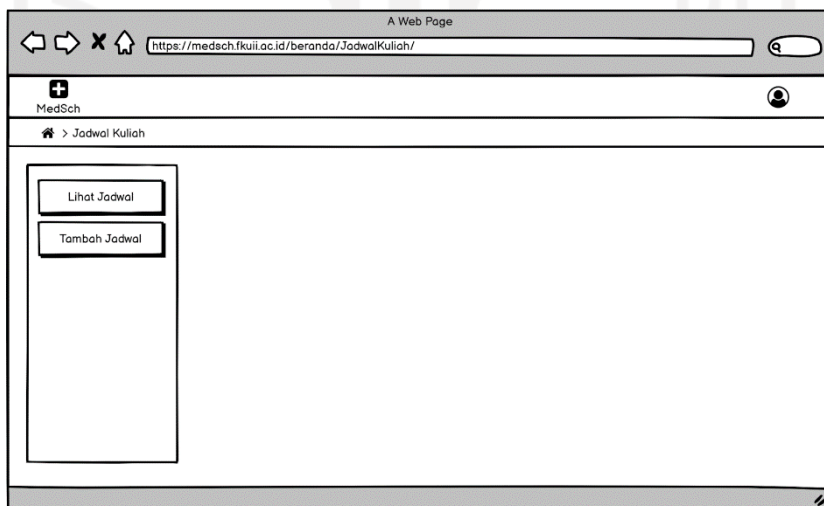


Gambar 4.22 Halaman Beranda Admin

Pada halaman beranda *website* untuk admin menampilkan fitur yang tertera pada Gambar 4.22 yaitu Jadwal Kuliah dan Daftar Dokter.

c. Halaman Jadwal Kuliah Admin

Halaman jadwal kuliah admin berisi tampilan dan elemen yang digunakan untuk menggamabarkan halaman jadwal kuliah.

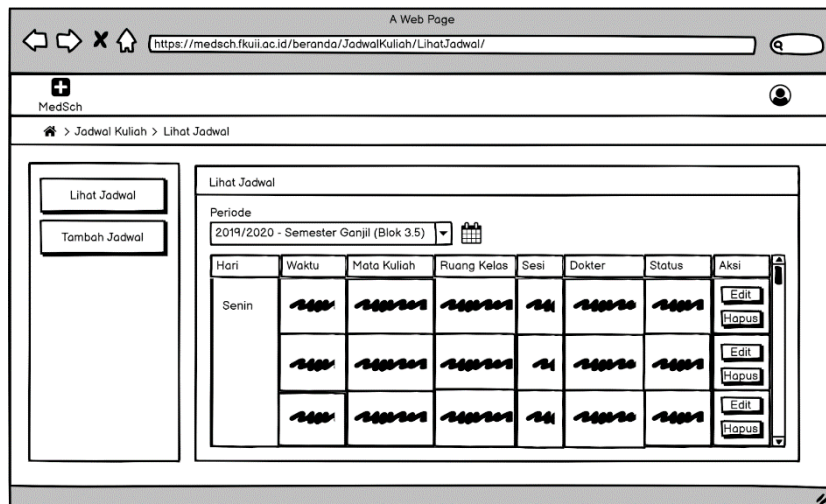


Gambar 4.23 Halaman Jadwal Kuliah Admin

Pada halaman jadwal kuliah *website* untuk admin menampilkan fitur yang tertera pada Gambar 4.23 yaitu Lihat Jadwal dan Tambah Jadwal.

d. Halaman Lihat Jadwal Admin

Halaman lihat jadwal kuliah admin berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman lihat jadwal kuliah.

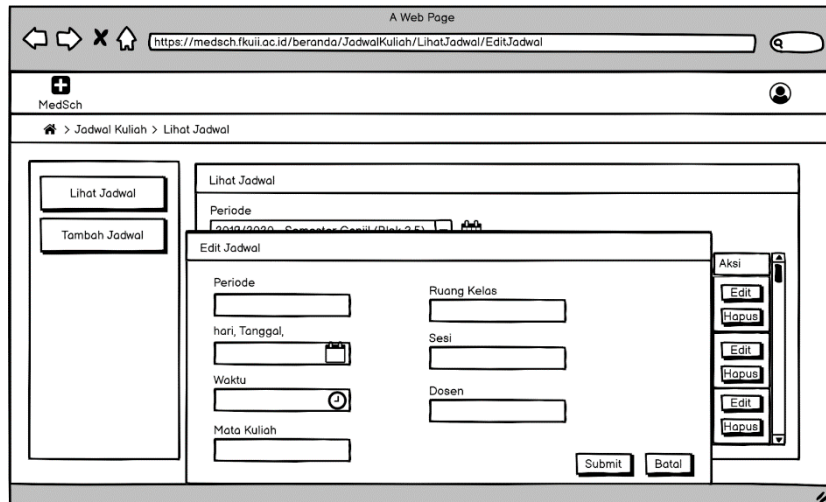


Gambar 4.24 Halaman Lihat Jadwal Admin

Pada halaman lihat jadwal *website* untuk admin menampilkan jadwal kuliah yang sudah diinput oleh admin, fitur yang tertera pada Gambar 4.24 yaitu Edit Jadwal dan Hapus Jadwal.

e. Halaman Edit Jadwal Admin

Halaman edit jadwal kuliah admin berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman edit jadwal kuliah.

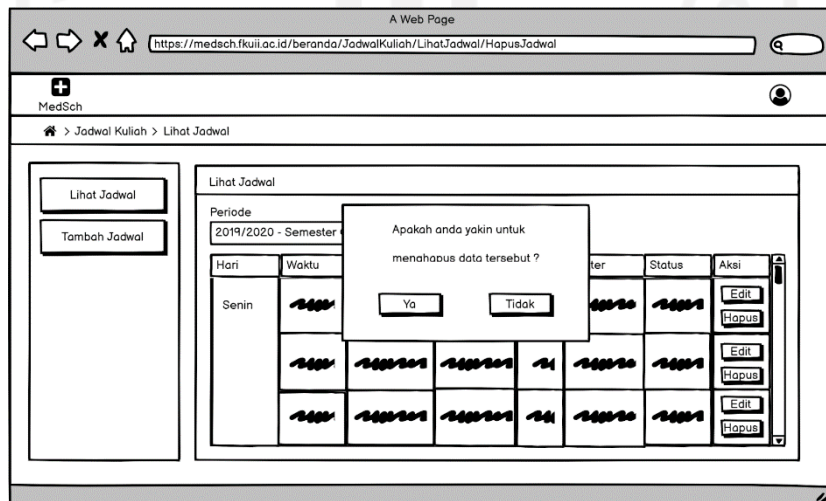


Gambar 4.25 Halaman Edit Jadwal Admin

Pada Gambar 4.25 halaman edit jadwal *website* admin menampilkan *form* untuk mengedit jadwal kuliah jika ada kesalahan atau perubahan.

f. Halaman Hapus Jadwal Admin

Halaman hapus jadwal kuliah admin berisi tampilan dan elemen yang digunakan untuk menggambarakan halaman hapus jadwal kuliah.



Gambar 4.26 Halaman Hapus Jadwal Admin

Pada gambar 4.26 halaman hapus jadwal *website* admin bertujuan untuk menghapus jadwal kuliah.

g. Halaman *Assign* Kelompok Mahasiswa

Halaman *assign* kelompok mahasiswa admin berisi tampilan dan elemen yang digunakan untuk menggamabarkan halaman *assign* kelompok mahasiswa.

The screenshot shows a web browser window with the URL <https://medsch.fkuii.ac.id/beranda/JadwalKuliah/AssignKelompokMahasiswa>. The page title is "Assign Kelompok Mahasiswa". On the left, there is a sidebar with two buttons: "Jadwal Kuliah" and "Daftar Dokter". The main content area contains a table with the following data:

No.	Nama Mahasiswa	NIM	Kelompok
1	Silvi Rahmawati	17711162	Kelompok 2
2	Rizal Ahsan Rizqi	17711033	Kelompok 2
3	Syifa'a Maharani	17711097	Kelompok 2
4	Fadila Rusdi AS	18711154	Kelompok 2

At the bottom right of the table, there is a "Submit" button.

Gambar 4.27 Halaman *Assign* Kelompok Mahasiswa

Pada Gambar 4.27 halaman *assign* kelompok mahasiswa admin bertujuan untuk mengelompokkan mahasiswa baru agar memudahkan admin dalam mendistribusikan jadwal kuliah.

h. Halaman Tambah Jadwal Admin

Halaman tambah jadwal admin berisi tampilan dan elemen yang digunakan untuk menggamabarkan halaman tambah jadwal.

Gambar 4.28 Halaman Tambah Jadwal Admin

Pada Gambar 4.28 halaman tambah jadwal admin menampilkan *form* untuk menambah jadwal kuliah.

i. Halaman Daftar Dokter Admin

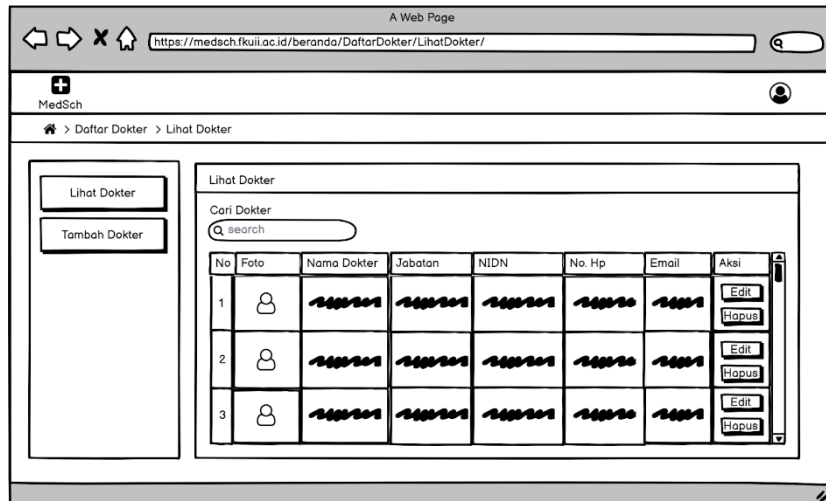
Halaman daftar dokter admin berisi tampilan dan elemen yang digunakan untuk menggambar halaman daftar dokter.

Gambar 4.29 Halaman Daftar Dokter Admin

Pada halaman daftar dokter admin menampilkan fitur yang tertera pada Gambar 4.29 yaitu Lihat Dokter dan Tambah Dokter.

j. Halaman Lihat Dokter Admin

Halaman lihat dokter admin berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman lihat dokter.

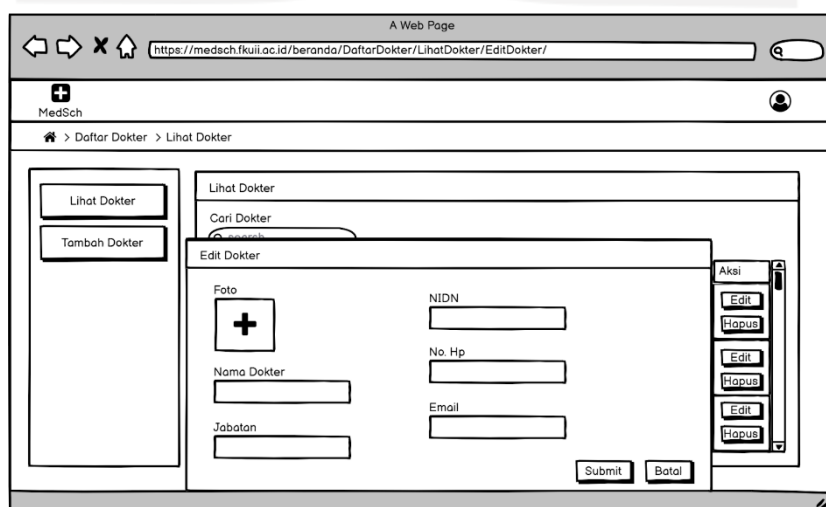


Gambar 4.30 Halaman Lihat Dokter Admin

Pada halaman lihat dokter *website* untuk admin menampilkan daftar dokter yang sudah diinput oleh admin, fitur yang tertera pada Gambar 4.30 yaitu Edit dan Hapus Dokter.

k. Halaman Edit Dokter Admin

Halaman edit dokter admin berisi tampilan dan elemen yang digunakan untuk menggambarkan halaman edit dokter.

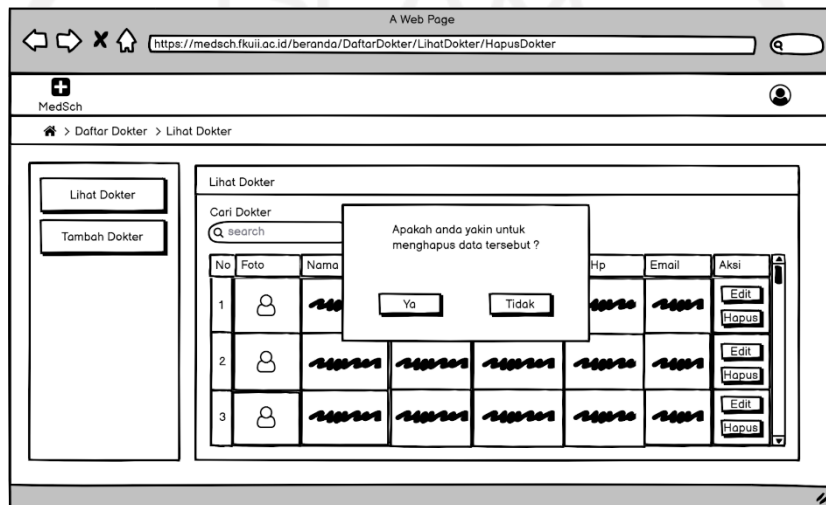


Gambar 4.31 Halaman Edit Dokter Admin

Pada Gambar 4.31 halaman edit dokter admin menampilkan *form* untuk mengedit data dosen jika ada kesalahan atau perubahan.

l. Halaman Hapus Dokter Admin

Halaman hapus dokter admin berisi tampilan dan elemen yang digunakan untuk menggamabarkan halaman hapus dokter.



Gambar 4.32 Halaman Hapus Dokter Admin

Pada Gambar 4.32 halaman hapus dokter admin bertujuan untuk menghapus data dokter.

m. Halaman Tambah Dokter Admin

Halaman tambah dokter admin berisi tampilan dan elemen yang digunakan untuk menggamabarkan halaman tambah dokter.

The screenshot shows a web browser window with the URL <https://medsch.fkuii.ac.id/beranda/DaftarDokter/TambahDokter/>. The page title is 'A Web Page'. The browser's address bar shows the URL. The page content includes a sidebar on the left with two buttons: 'Lihat Dokter' and 'Tambah Dokter'. The main content area is titled 'Tambah Dokter' and contains a form with the following fields: 'Foto' (with a plus icon), 'NIDN', 'Nama Dokter', 'No. Hp', 'Jabatan', and 'Email'. A 'Submit' button is located at the bottom right of the form.

Gambar 4.33 Halaman Tambah Dokter Admin

Pada Gambar 4.33 halaman tambah dokter admin menampilkan *form* untuk menambah data dosen.

4.1.3 Desain Tampilan Aplikasi MedSch

Berikut hasil implementasi desain tampilan dari aplikasi MedSch berdasarkan *user interface* yang telah dibuat. Pada tahap disini akan dibagi menjadi tiga skenario yaitu mahasiswa dan dosen menggunakan aplikasi berbasis Android dan Admin menggunakan aplikasi berbasis *Website*. Pada tahap ini bisa dikatakan bahwa tahapan dari penerapan hasil rancangan sistem yang sudah dibangun sebelumnya agar menjadi sistem yang siap untuk dioperasikan. Untuk hasil implementasi dari Sistem Informasi Jadwal Kuliah berbasis Android dan *website*, adalah sebagai berikut:

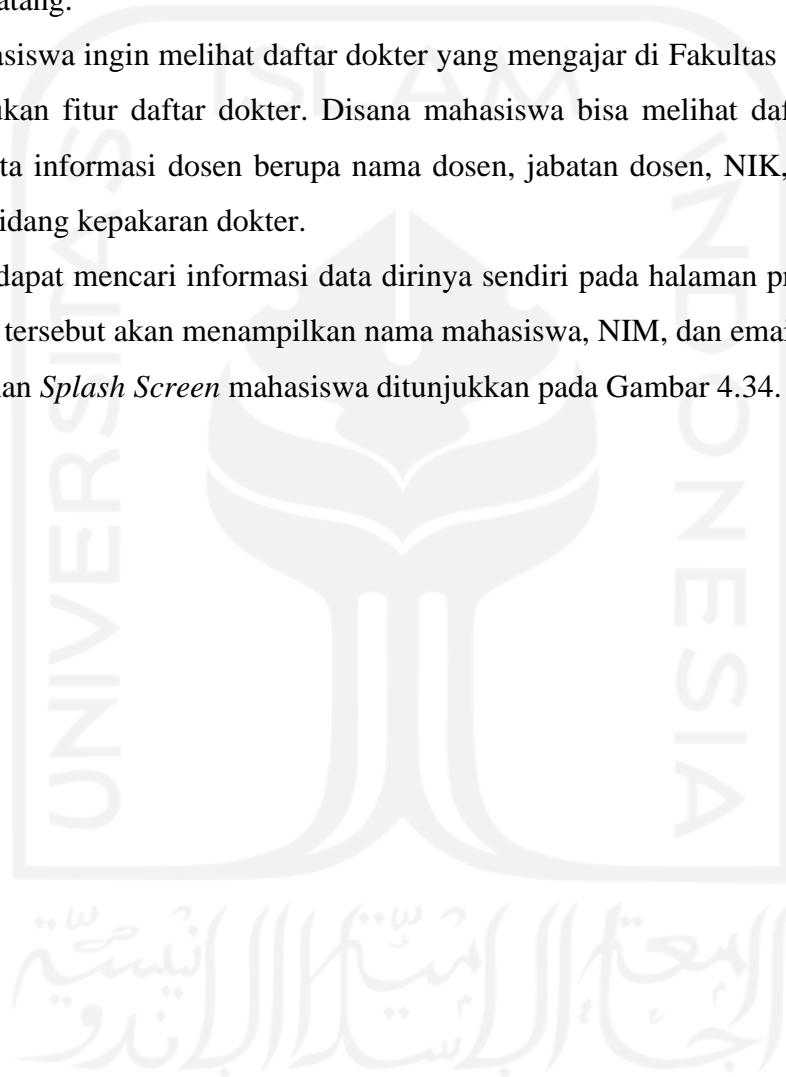
4.1.3.1 Skenario Mahasiswa

Pada sistem aplikasi berbasis Android mahasiswa ini memiliki halaman *Splash Screen*, *Landing Page*, halaman *Login & Sign Up*, halaman *Dashboard*, halaman Jadwal Kuliah, halaman Daftar Dokter, dan halaman Profile. Skenario dari mahasiswa yaitu:

1. Mahasiswa membuka aplikasi Android MedSch, lalu mendaftarkan di halaman *Sign Up*, kemudian memasukkan email, password, nama lengkap, dan NIM (disini penulis membuat akun dengan email rizaldiBali@gmail.com, password bali123, nama lengkap Moh. Dwi Chaidier Rizaldi, dan NIM 17523157). Selanjutnya mahasiswa 'masuk' ke sistem dengan alamat email rizaldiBali@gmail.com dan kata sandi bali123 dan diarahkan ke halaman dashboard.

2. Mahasiswa dapat melihat halaman dashboard yang memiliki dua fitur yaitu jadwal kuliah dan daftar dokter.
3. Jika mahasiswa yang ingin melihat jadwal dapat diakses dari fungsi atau fitur jadwal kuliah. Pada fitur ini mahasiswa bisa melihat jadwal kuliah yang berisikan informasi terkait jam, mata kuliah, pengajar, ruang kelas, dan status. Halaman jadwal kuliah juga bisa memilih blok yang sedang berlangsung untuk melihat jadwal terkini, jadwal sebelumnya, dan jadwal yang akan datang.
4. Ketika mahasiswa ingin melihat daftar dokter yang mengajar di Fakultas Kedokteran UII, dapat dilakukan fitur daftar dokter. Disana mahasiswa bisa melihat daftar dokter yang berisikan data informasi dosen berupa nama dosen, jabatan dosen, NIK, nomor telepon, email, dan bidang kepakaran dokter.
5. Mahasiswa dapat mencari informasi data dirinya sendiri pada halaman profil mahasiswa, dimana data tersebut akan menampilkan nama mahasiswa, NIM, dan email mahasiswa.

Tampilan halaman *Splash Screen* mahasiswa ditunjukkan pada Gambar 4.34.





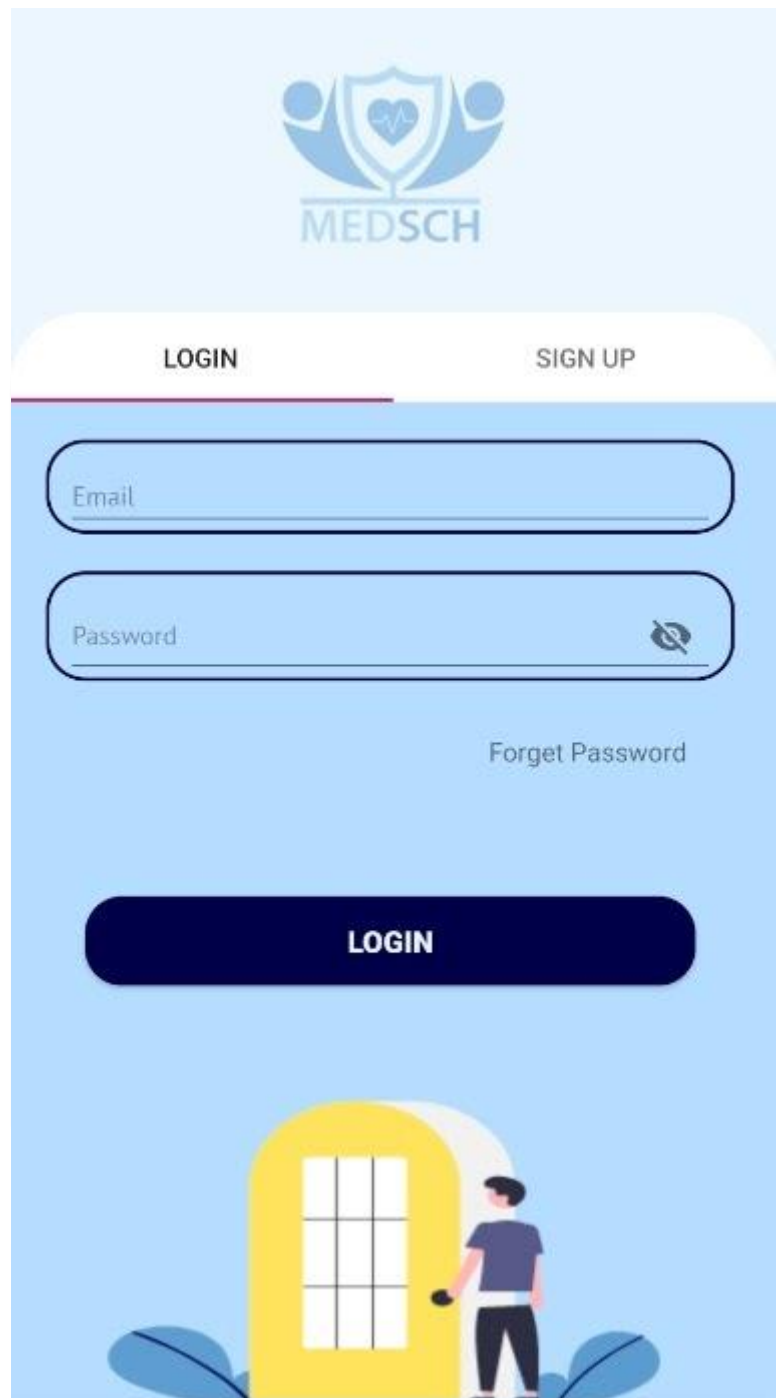
Gambar 4.34 Tampilan *Splash Screen* Aplikasi Android MedSch Mahasiswa

Tampilan halaman *Landing Page* mahasiswa ditunjukkan pada Gambar 4.35.



Gambar 4.35 Tampilan *Landing Page* Aplikasi Android MedSch Mahasiswa

Tampilan halaman *Login* mahasiswa ditunjukkan pada Gambar 4.36.



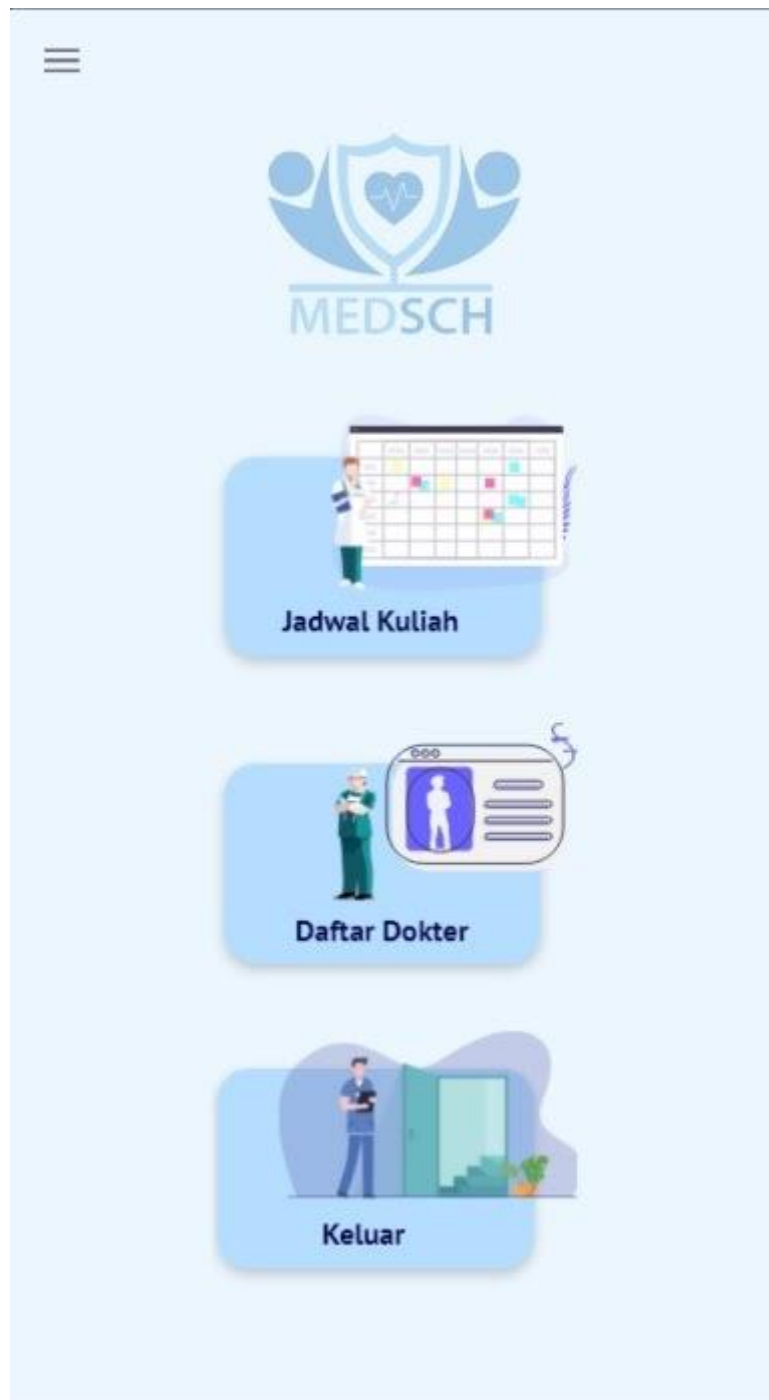
Gambar 4.36 Tampilan *Login* Aplikasi Android MedSch Mahasiswa

Tampilan halaman *Sign Up* mahasiswa ditunjukkan pada Gambar 4.37.

The image shows the sign-up interface of the MedSch application. At the top center is the MedSch logo, which consists of a shield with a heart and a pulse line, flanked by two stylized figures, with the text 'MEDSCH' below it. Below the logo is a white navigation bar with two tabs: 'LOGIN' and 'SIGN UP'. The 'SIGN UP' tab is currently selected, indicated by a red underline. Below the navigation bar are four rounded rectangular input fields stacked vertically. The first field is labeled 'Email', the second 'Password' (with a toggle icon on the right), the third 'Nama Lengkap', and the fourth 'NIM'. Below these fields is a large, dark blue button with the text 'SIGN UP' in white. At the bottom of the screen, there is a stylized illustration of a person sitting on a ledge, looking thoughtful, with some trees and a dark rectangular shape with white lines below them.

Gambar 4.37 Tampilan *Sign Up* Aplikasi Android MedSch Mahasiswa

Tampilan halaman *Dashboard* mahasiswa ditunjukkan pada Gambar 4.38.



Gambar 4.38 Tampilan *Dashboard* Aplikasi Android MedSch Mahasiswa

Tampilan halaman Jadwal Kuliah mahasiswa ditunjukkan pada Gambar 4.39.



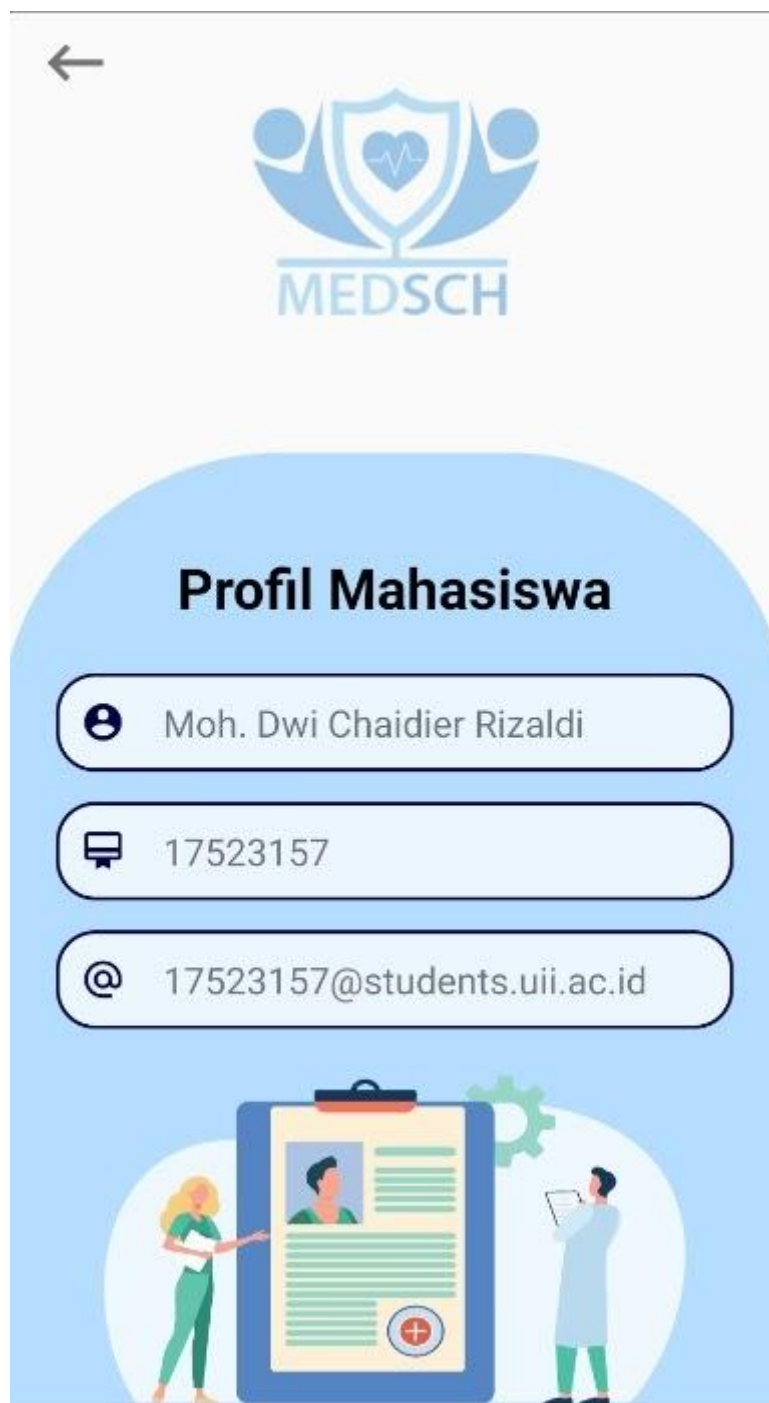
Gambar 4.39 Tampilan Jadwal Kuliah Aplikasi Android MedSch Mahasiswa

Tampilan halaman Daftar Dokter mahasiswa ditunjukkan pada Gambar 4.40.



Gambar 4.40 Tampilan Daftar Dokter Aplikasi Android MedSch Mahasiswa

Tampilan halaman *Profile* mahasiswa ditunjukkan pada Gambar 4.41.



Gambar 4.41 Tampilan Profile Aplikasi Android MedSch Mahasiswa

4.1.3.2 Skenario Dosen

Pada sistem aplikasi berbasis Android mahasiswa ini memiliki halaman *Splash Screen*, halaman *Sign In*, halaman *Dashboard*, halaman Jadwal Kuliah, dan halaman Konfirmasi Jadwal. Skenario dari dosen yaitu:

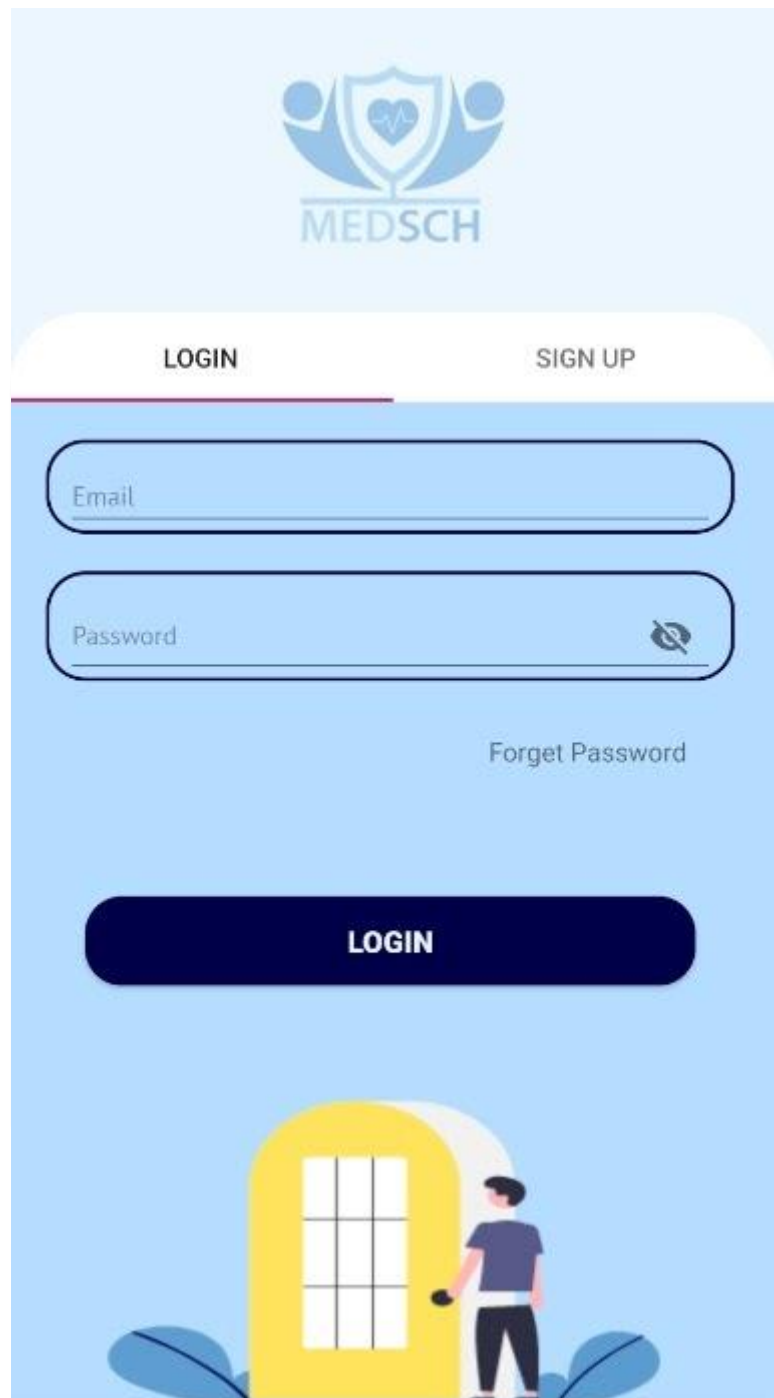
1. Dosen membuka aplikasi Android MedSch, setelah itu menuliskan email dan password akun di halaman *Sign In* (disini penulis membuat akun dengan email isnatin@uii.ac.id, password isnatin123). Selanjutnya dosen 'masuk' ke sistem dengan alamat email isnatin@uii.ac.id dan kata sandi isnatin123 dan diarahkan ke halaman dashboard.
2. Pada halaman dashboard memiliki pilihan fitur yang disediakan yaitu jadwal kuliah dan konfirmasi jadwal.
3. Jika ingin melihat jadwal perkuliahan, dosen bisa memilih fitur jadwal kuliah. Disana dosen dapat melihat jadwal kuliah yang berisikan informasi berupa jam, mata kuliah, pengajar, ruang kelas, status, dan juga dosen bisa mengubah status jadwal lewat fitur *update* status. Halaman jadwal kuliah juga dapat memilih blok yang sedang berlangsung untuk melihat jadwal yang sudah dikonfirmasi sebelumnya pada halaman konfirmasi jadwal.
4. Jika ingin mengonfirmasi jadwal kuliah, dosen dapat memilih fitur konfirmasi jadwal. Disana dosen dapat melihat rancangan jadwal kuliah yang sudah disiapkan oleh admin. Dosen diberikan pilihan konfirmasi jika dosen bisa mengajar di waktu yang sudah disiapkan oleh admin, atau dosen dapat menolak jadwal tersebut.

Tampilan halaman *Splash Screen* dosen ditunjukkan pada Gambar 4.42.



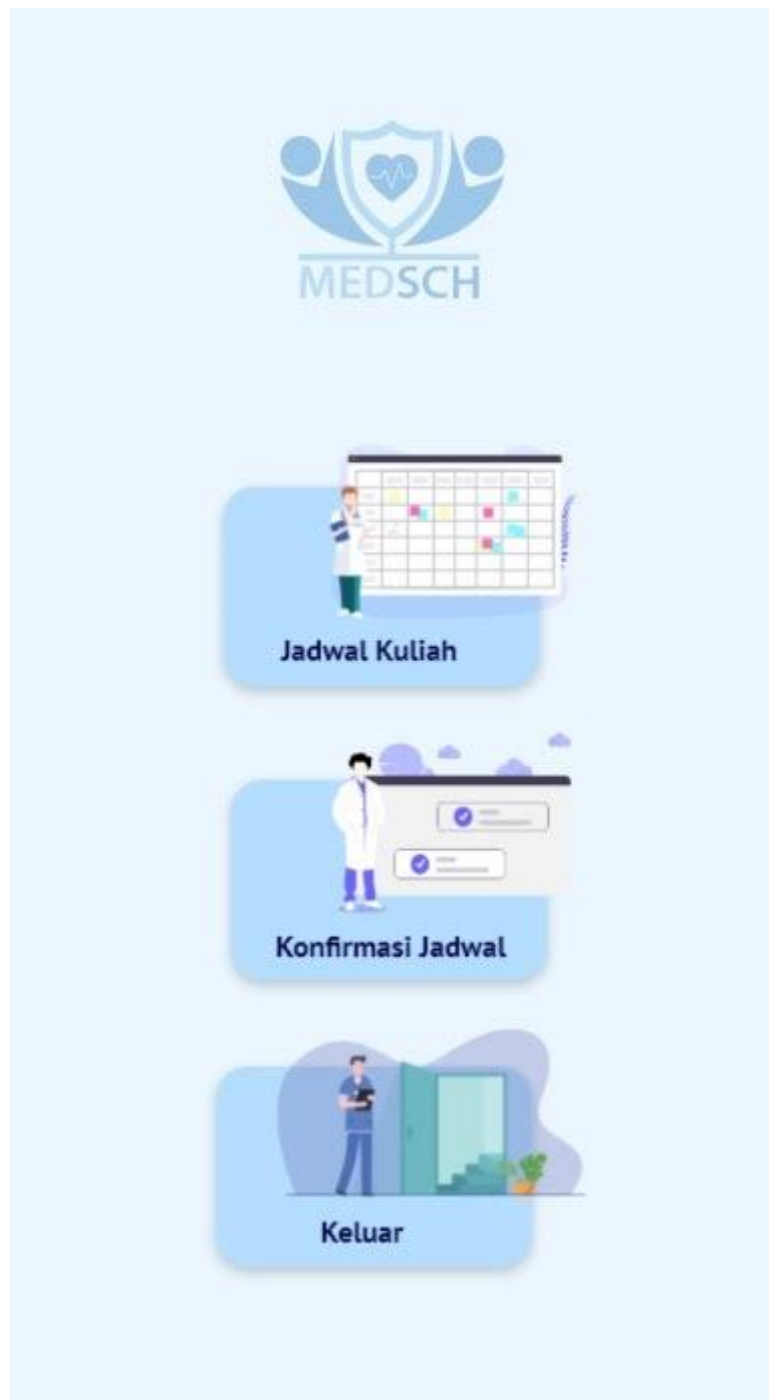
Gambar 4.42 Tampilan *Splash Screen* Aplikasi Android MedSch Dosen

Tampilan halaman *Login* dosen ditunjukkan pada Gambar 4.43.



Gambar 4.43 Tampilan *Login* Aplikasi Android MedSch Dosen

Tampilan halaman *Dashboard* dosen ditunjukkan pada Gambar 4.44.



Gambar 4.44 Tampilan *Dashboard* Aplikasi Android MedSch Dosen

Tampilan halaman Jadwal Kuliah dosen ditunjukkan pada Gambar 4.45.



Gambar 4.45 Tampilan Jadwal Kuliah Aplikasi Android MedSch Dosen

Tampilan halaman Konfirmasi jadwal dosen ditunjukkan pada Gambar 4.46.

The screenshot displays the 'Konfirmasi Jadwal' (Schedule Confirmation) screen. At the top, there is a back arrow and the title 'Konfirmasi Jadwal'. Below the title, there are three schedule items, each with a list of details and two buttons: 'Tolak' (Reject) and 'Konfirmasi' (Confirm).

Item	Tanggal/Tahun	Jam	Nama Blok	Mata Kuliah	Ruang Kelas	Status
1	2019/10/21	13.00 - 13.50 14.00 - 14.50	Blok 3.2 Masalah Pada Anak	EBM : Telaah Kritis Jurnal Harm dan Etiology	Gedung Kuliah Umum (Prof. Dr. Sardjito)	PENDING
2	2019/10/22	10.00 - 10.50 11.00 - 11.50	Blok 3.2 Masalah Pada Anak	IKA : Gangguan Pertumbuhan Pada Anak	Gedung Kuliah Umum (Prof. Dr. Sardjito)	PENDING
3	2019/10/23	10.00 - 10.50				

Gambar 4.46 Tampilan Konfirmasi Jadwal Aplikasi Android MedSch Dosen

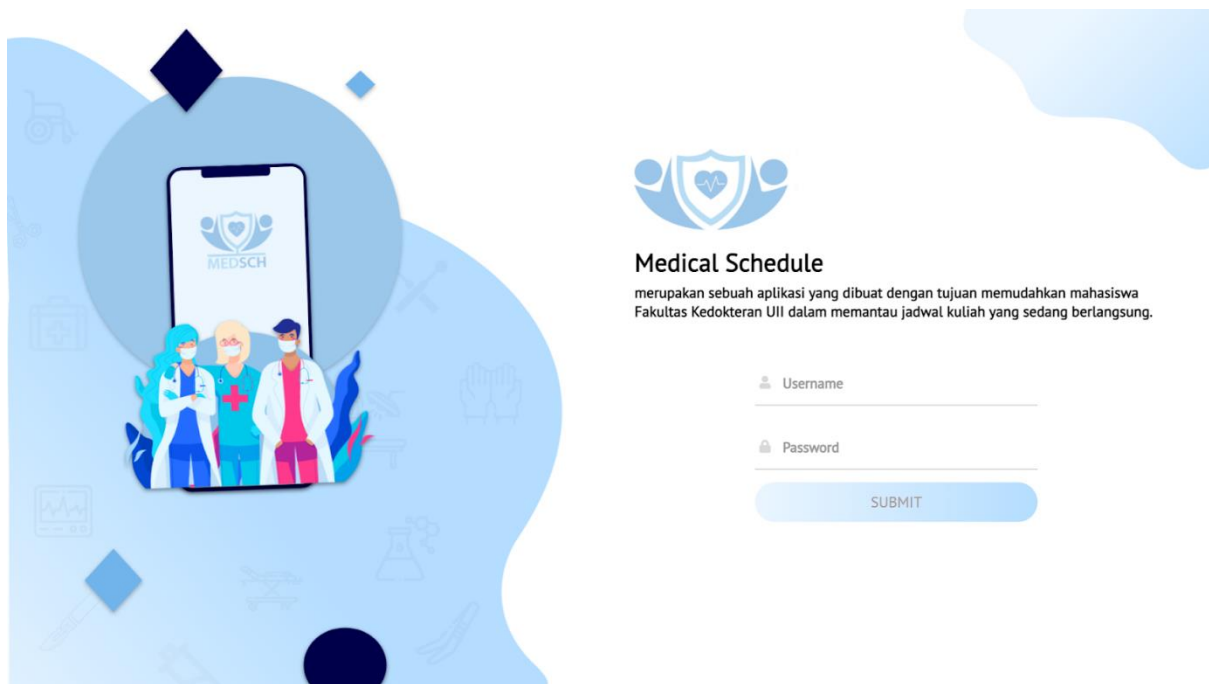
4.1.3.3 Skenario Admin

Pada sistem aplikasi berbasis Android mahasiswa ini memiliki halaman *Login*, halaman *Dashboard*, halaman Jadwal Kuliah, halaman Lihat Jadwal, halaman Assign Kelompok Mahasiswa, halaman Tambah Jadwal, halaman Daftar Dokter, halaman Lihat Dokter, halaman Tambah Dokter, dan halaman Profile, Skenario dari admin yaitu:

1. Admin membuka aplikasi *website* MedSch, setelah itu menuliskan email dan password akun di halaman *Login* (disini penulis membuat akun dengan email admin, password 123435). Selanjutnya admin ‘masuk’ ke sistem dengan alamat email admin dan kata sandi 12345 dan diarahkan ke halaman dashboard.
2. Admin dapat melihat halaman dashboard yang memiliki pilihan menu pada sisi kiri navbar yaitu jadwal kuliah dan daftar dokter.
3. Admin dapat mengelola jadwal perkuliahan mahasiswa pada menu jadwal kuliah. Jadwal kuliah ini memiliki fitur yaitu:
 - a. Lihat jadwal
Admin dapat melihat jadwal kuliah mahasiswa serta dapat mengedit dan menghapus jadwal kuliah. Admin juga dapat melihat jadwal kuliah yang ditolak oleh dosen dan dapat memfilter jadwal kuliah yang diinginkan.
 - b. Assign kelompok mahasiswa
Admin dapat mengelompokkan para mahasiswa yang baru mendaftar melalui aplikasi Android, gunanya mengelompokkan mahasiswa disini adalah untuk memudahkan admin dalam mendistribusikan jadwal kuliah yang sudah ditentukan.
 - c. Tambah jadwal
Admin dapat menambahkan jadwal kuliah, lalu sistem akan menyimpan dan ditampilkan pada halaman lihat jadwal.
4. Admin dapat mengelola data dosen pada menu daftar dokter. Daftar dokter ini memiliki fitur yaitu:
 - a. Lihat dokter
Admin dapat melihat list data dokter serta dapat mengedit dan menghapus data dokter tersebut.
 - b. Tambah dokter
Admin dapat menambahkan data diri dokter, lalu sistem akan menyimpan dan ditampilkan pada halaman lihat dokter.

5. Ketika admin ingin mengedit profil, dapat dilakukan pada bagian sisi atas navbar lalu pilih menu profile (admin disini hanya bisa mengedit password saja tidak dengan namanya). Selanjutnya pada kolom password diubah menjadi 123456 lalu klik tombol submit untuk menyimpan.

Tampilan halaman *Login* admin ditunjukkan pada Gambar 4.47.



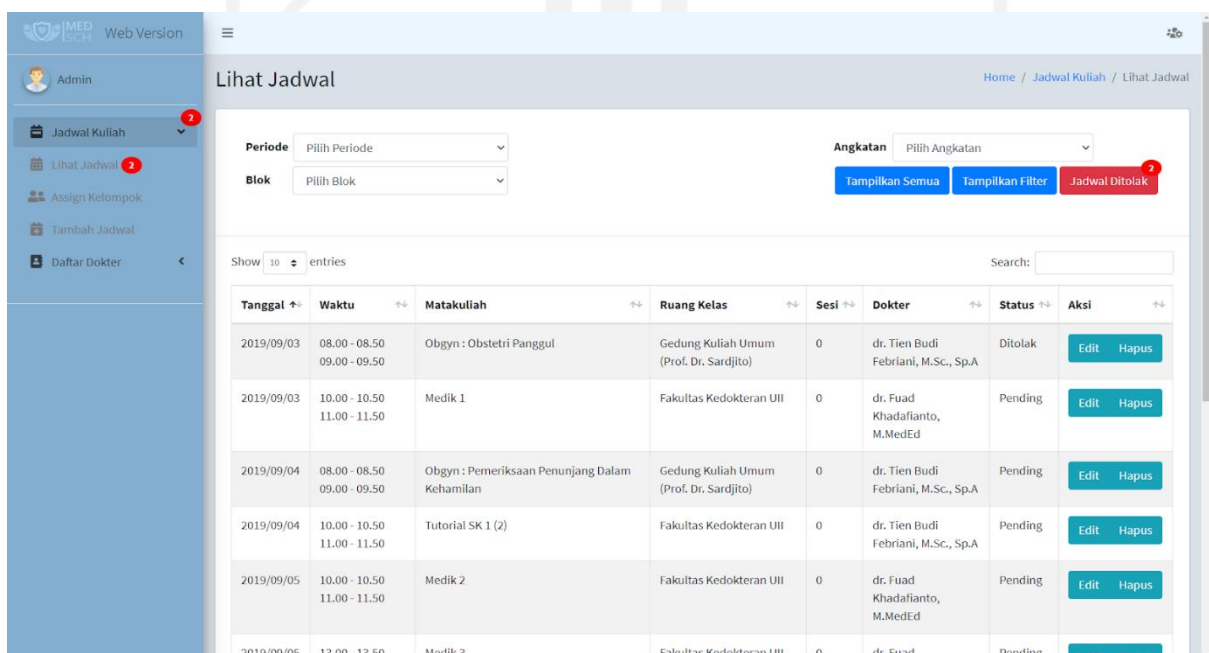
Gambar 4.47 Tampilan *Login* Aplikasi *Website* MedSch Admin

Tampilan halaman *Dashboard* admin ditunjukkan pada Gambar 4.48.



Gambar 4.48 Tampilan *Dashboard* Kuliah Aplikasi *Website* MedSch Admin

Tampilan halaman Lihat Jadwal admin ditunjukkan pada Gambar 4.49.



Gambar 4.49 Tampilan Lihat Jadwal Kuliah Aplikasi *Website* MedSch Admin

Tampilan halaman Edit Jadwal admin ditunjukkan pada Gambar 4.50.

Gambar 4.50 Tampilan Hapus Jadwal Kuliah Aplikasi *Website MedSch Admin*

Tampilan halaman Assign Kelompok Mahasiswa admin ditunjukkan pada Gambar 4.51.

No	Nama Mahasiswa	NIM	Kelompok
1	testing	17523001	kelompok 2
2	kukuh Anugrah	17523002	kelompok 1
3	Rizaldi Bali	17523003	kelompok 1
4	Yanti Purnama Sari	17711181	kelompok 1
5	Silvi Rahmawati	17711162	kelompok 2
6	Syifa'a Maharani I	17711097	kelompok 2
7	Rizal Ahsan Rizqi	17711033	kelompok 2
8	Fadila Rusdi AS	18711154	Kelompok 3

Gambar 4.51 Tampilan Assign Kelompok Mahasiswa Aplikasi *Website MedSch Admin*

Tampilan halaman Tambah Jadwal admin ditunjukkan pada Gambar 4.52.

MedSch Web Admin Version 1.0.0

Gambar 4.52 Tampilan Tambah Jadwal Aplikasi *Website* MedSch Admin

Tampilan halaman Lihat Dokter admin ditunjukkan pada Gambar 4.53.

No	Nama Dokter	Jabatan	Bidang Kepekaran	NIDN	No. HP	Email	Aksi
1	Dr. dr. Isnatin Miladiyah, M.Kes	Ketua Jurusan Kedokteran	Farmakologi	4009097004	(0274) 898444	isnatin@uii.ac.id	Edit Hapus
2	Dr. dr. Farida Juliantina Rachmawaty, M.Kes	Sekretaris Jurusan Kedokteran	Mikrobiologi	4009097005	081328584137	017110101@uii.ac.id	Edit Hapus
3	dr. Ana Fauziyati, M.Sc., Sp.PD.	Ketua Program Studi Profesi Dokter	Ilmu Penyakit Dalam	4009097006	(0274) 898444	anafauziyati@uii.ac.id	Edit Hapus
4	dr. Umatul Khoiriyah, M.Med.Ed, Ph.D	Ketua Program Studi Kedokteran – Program Sarjana	Medical Education	4009097007	(0274) 898444	047110101@uii.ac.id	Edit Hapus
5	dr. Fuad Khadafianto, M.MedEd	Pengajar	Medical Education	4009097009	(0274) 898444	097110403@uii.ac.id	Edit Hapus
6	Prof. Dr. dr. H. Soewadi, MPH, Sp.KJ(K)	Pengajar	Ilmu Kesehatan Jiwa	4009097010	(0274) 898444	077110437@uii.ac.id	Edit Hapus
7	dr.g. Andy Yok Siswosaputro, M.Kes	Pengajar	Ilmu Forensik	4009097011	(0274) 898444	107110401@uii.ac.id	Edit Hapus

Gambar 4.53 Tampilan Lihat Dokter Aplikasi *Website* MedSch Admin

Tampilan halaman Edit Dokter admin ditunjukkan pada Gambar 4.54

Form Edit Dokter Dr. dr. Isnatin Miladiyah, M.Kes

Nama Dokter	<input type="text" value="Dr. dr. Isnatin Miladiyah, M.Kes"/>	Bidang Kepakaran	<input type="text" value="Farmakologi"/>
Jabatan	<input type="text" value="Ketua Jurusan Kedokteran"/>	No. HP	<input type="text" value="(0274) 898444"/>
NIDN	<input type="text" value="4009097004"/>	Email	<input type="text" value="isnatin@uii.ac.id"/>

No	Nama Dokter	Jabatan	Bidang	NIDN	No. HP	Email	Aksi
4	dr. Umatai Khoiriyah, M.Med.Ed, Ph.D	Ketua Program Studi Kedokteran – Program Sarjana	Medical Education	4009097007	(0274) 898444	047110101@uii.ac.id	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
5	dr. Fuad Khadafianto, M.MedEd	Pengajar	Medical Education	4009097009	(0274) 898444	097110403@uii.ac.id	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
6	Prof. Dr. dr. H. Soewadi, MPH, Sp.KJ(K)	Pengajar	Ilmu Kesehatan Jiwa	4009097010	(0274) 898444	077110437@uii.ac.id	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>
7	drg. Andy Yok Siswosaputro, M.Kes	Pengajar	Ilmu Forensik	4009097011	(0274) 898444	107110401@uii.ac.id	<input type="button" value="Edit"/> <input type="button" value="Hapus"/>

Gambar 4.54 Tampilan Edit Dokter Aplikasi *Website MedSch Admin*

Tampilan halaman Tambah Dokter admin ditunjukkan pada Gambar 4.55.

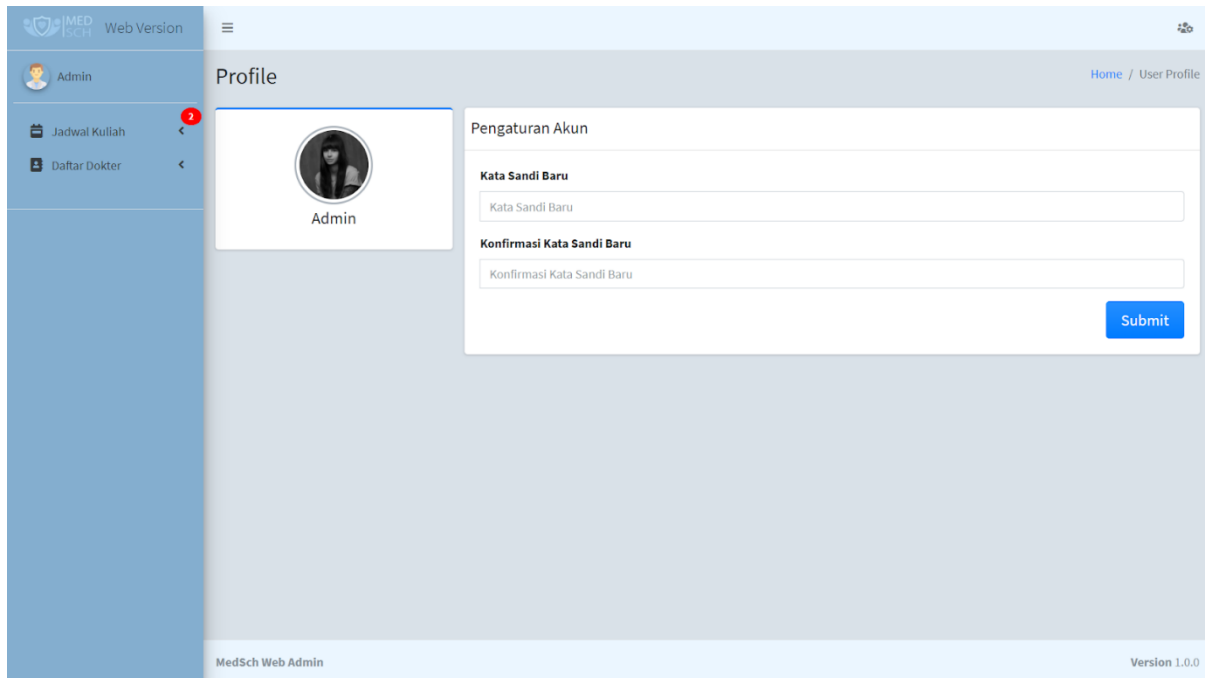
Form Tambah Dokter

Nama Dokter	<input type="text" value="Nama Dokter"/>	Bidang Kepakaran	<input type="text" value="Bidang Kepakaran"/>
Jabatan	<input type="text" value="Jabatan"/>	No. HP	<input type="text" value="No. HP"/>
NIDN	<input type="text" value="NIDN"/>	Email	<input type="text" value="Email"/>

MedSch Web Admin Version 1.0.0

Gambar 4.55 Tampilan Tambah Dokter Aplikasi *Website MedSch Admin*

Tampilan halaman Profile admin ditunjukkan pada Gambar 4.56.



Gambar 4.56 Tampilan Profile Aplikasi *Website* MedSch Admin

4.2 Hasil Pengujian

Hasil dari pengujian fungsionalitas aplikasi melalui metode Black Box testing seperti yang tersaji pada Tabel 4.1 dan Tabel 4.2 memberikan hasil yang memuaskan di mana fitur-fitur yang terdapat dalam aplikasi telah bekerja sesuai dengan apa yang telah diharapkan. Pengujian fungsionalitas dengan metode *black box* yang sudah diterapkan pada aplikasi memberikan hasil bahwa semua fitur dan juga fungsi yang ada didalam aplikasi mulai dari pertama kali digunakan sampai selesai dapat berjalan dengan baik juga sesuai dengan apa yang telah diharapkan. Masing – masing pengujian dari aplikasi berbasis Android dan berbasis *website*, untuk berbasis Android memiliki 11 fungsionalitas yang telah diujikan, semua berjalan dengan baik dan sesuai harapan. Begitu pula untuk yang berbasis *website*, memiliki 16 fungsionalitas yang telah diujikan dan semua berjalan dengan baik dan sesuai yang direncanakan. Untuk hasil pengujian disajikan pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Hasil Pengujian Fungsionalitas Pada Aplikasi Berbasis Android

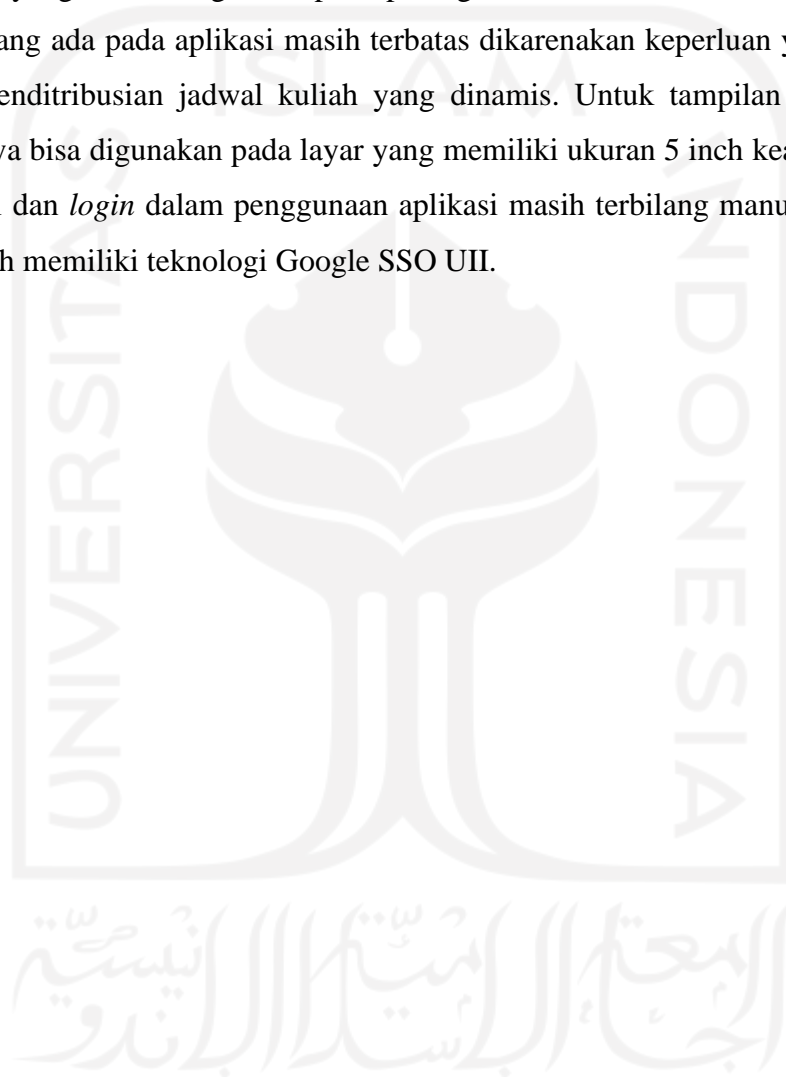
No.	Use Case	Keterangan	Hasil Pengujian
1	<i>Login</i>	Melakukan <i>login</i> untuk mahasiswa dan dosen	Berhasil
2	Daftar akun	Melakukan pendaftaran akun pengguna baru untuk mahasiswa	Berhasil
3	Dashboard pada aplikasi	Menampilkan fitur pada aplikasi yang akan digunakan untuk mahasiswa dan dosen	Berhasil
4	Menu jadwal kuliah	Menampilkan data jadwal kuliah berupa kalender untuk memilih hari pada jadwal kuliah, jam kuliah, nama kuliah, pengajar, ruang kelas dan status jadwal kuliah.	Berhasil
5	Menu daftar dokter	Menampilkan data dosen berupa nama dosen, jabatan dosen, bidang kepakaran, NIDN, nomor telepon, dan email.	Berhasil
6	Menu konfirmasi jadwal	Melakukan konfirmasi atau membatalkan jadwal kuliah yang sudah dirancang oleh admin untuk dosen.	Berhasil
7	Searchbox pada daftar dokter	Menampilan filter keyword untuk menampilkan data dosen yang ingin ditampilkan.	Berhasil
8	Memilih blok pada jadwal kuliah	Memilih blok yang ingin dipilih untuk menampilkan jadwal kuliah yang ingin dilihat.	Berhasil
9	Ubah status jadwal kuliah	Mengubah status jadwal kuliah dari dosen.	Berhasil
10	Menolak rancangan jadwal pada konfirmasi jadwal	Menolak rancangan jadwal yang sudah dibuat oleh admin, dan jika ada catatan akan tersampaikan langsung ke <i>website</i> .	Berhasil
11	Menu profil	Menampilkan informasi mahasiswa berupa nama, NIM dan email.	Berhasil

Tabel 4.2 Hasil Pengujian Fungsionalitas Pada Aplikasi Berbasis *Website*

No.	Use Case	Keterangan	Hasil Pengujian
1	<i>Login</i>	Melakukan <i>login</i> untuk mahasiswa dan dosen	Berhasil
2	Dashboard pada web	Menampilkan fitur pada web yang akan digunakan untuk admin.	Berhasil
3	Menu lihat jadwal	Menampilkan data jadwal kuliah berupa tanggal, waktu, mata kuliah, ruang kelas, sesi, dosen, status, dan aksi untuk mengedit atau menghapus data jadwal kuliah.	Berhasil
4	Memilih periode, blok, dan angkatan pada lihat jadwal	Memilih dan menampilkan data jadwal kuliah yang sudah diatur pada filter.	Berhasil
5	Tampilkan semua data jadwal kuliah pada lihat jadwal	Menampilkan semua jadwal kuliah.	Berhasil
6	Tampilkan jadwal yang ditolak pada lihat jadwal	Menampilkan jadwal yang ditolak oleh dosen pada aplikasi berbasis Android yang dikirim ke <i>website</i> .	Berhasil
7	Edit jadwal kuliah	Mengedit jadwal kuliah yang ingin diubah data dari sebelumnya.	Berhasil
8	Hapus jadwal kuliah	Menghapus data jadwal dari jadwal kuliah.	Berhasil
9	Menu assign kelompok mahasiswa	Menampilkan data mahasiswa berupa nama mahasiswa, NIM, dan pilihan kelompok.	Berhasil
10	Menu tambah jadwal	Menambahkan data jadwal ke menu lihat jadwal.	Berhasil
11	Menu lihat dokter	Menampilkan data dosen berupa nama dosen, jabatan dosen, bidang kepakaran, NIDN, nomor telepon, email dan aksi untuk mengedit atau menghapus data dosen.	Berhasil
12	Edit data dokter	Mengedit data dosen yang ingin diubah data dari sebelumnya.	Berhasil
13	Hapus data dokter	Menghapus data dosen dari daftar dokter.	Berhasil

14	Menu tambah dokter	Menambahkan data dosen ke menu lihat dokter.	Berhasil
15	Menu profil	Menampilkan informasi pengguna berupa foto profil, nama admin, dan password.	Berhasil
16	Ubah profil	Mengubah data password admin.	Berhasil

Pada sistem yang dikembangkan pada perangkat mobile masih sebatas basis Android dan fitur-fitur yang ada pada aplikasi masih terbatas dikarenakan keperluan yang dibutuhkan hanya proses pendistribusian jadwal kuliah yang dinamis. Untuk tampilan pada perangkat *mobile* pun hanya bisa digunakan pada layar yang memiliki ukuran 5 inch keatas. Di sisi lain, proses registrasi dan *login* dalam penggunaan aplikasi masih terbilang manual, dimana pada UII sendiri sudah memiliki teknologi Google SSO UII.



BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengembangan Sistem Informasi Jadwal Kuliah untuk Fakultas Kedokteran UII (MedSch) menggunakan metode Waterfall SDLC maka dapat disimpulkan sebagai berikut:

- a. Metode Waterfall SDLC berhasil diterapkan pada pengembangan Sistem Informasi Jadwal Kuliah MedSch, mulai dari tahapan *Requirements Definition, Plannings System & Software Design, Implementation*, hingga *Testing*.
- b. Aplikasi *mobile* yang digunakan mahasiswa pada Sistem Informasi Jadwal Kuliah MedSch berhasil dikembangkan. Pengembangan Aplikasi MedSch menggunakan bahasa pemrograman Java, untuk *server-side* menggunakan REST API berupa Node.js, dan untuk *database* menggunakan MySQL.
- c. Aplikasi *Website* yang digunakan admin BAAK pada Sistem Informasi Jadwal Kuliah MedSch berhasil dikembangkan menggunakan bahasa pemrograman native PHP dan *database* MySQL.
- d. *Tools* yang digunakan dalam pengembangan aplikasi MedSch berbasis Android yaitu Android Studio dan berbasis *Website* Visual Studio Code.
- e. Metode pengujian yang dilakukan pada sistem Informasi Jadwal Kuliah untuk FK UII (MedSch) yaitu dengan metode *black box testing*. Masing – masing pengujian dari aplikasi berbasis Android memiliki 11 fungsionalitas dan berbasis *website* memiliki 16 fungsionalitas, dimana pada pengujian tersebut berjalan dengan baik dan sesuai harapan.

5.2 Saran

Sistem yang dikembangkan ini masih memiliki beberapa kekurangan. Maka dari itu, terdapat beberapa saran pengembangan aplikasi lebih lanjut, seperti:

- a. Aplikasi yang dikembangkan baru sebatas berbasis Android. Diharapkan dapat dikembangkan untuk perangkat iOS juga.
- b. Adanya perbaikan jika suatu saat aplikasi MedSch mengalami *error*.
- c. Adanya perawatan secara berkala untuk menjaga agar aplikasi tetap berjalan dengan baik.
- d. Penambahan fitur baru apabila nanti dibutuhkan sehingga aplikasi lebih informatif dan lebih mudah digunakan oleh pengguna.

- e. Perbaiki tampilan agar lebih responsif dan menyesuaikan semua jenis perangkat.
- f. Fitur *login* diganti dengan Google SSO UII agar semua data mahasiswa maupun dosen dapat terintegrasi dan aktor tidak perlu melakukan registrasi.



DAFTAR PUSTAKA

- Andrian, M. A. W. (2019). Perancangan Sistem Pengolahan Data Nilai Siswa berbasis Java di SMP At-Taqwa Kec. Sawah Besar Jakarta. *STRING (Satuan Tulisan Riset dan Inovasi Teknologi)*, 3(3), 267-273.
- Android Studio. Mengenal Android Studio. Diakses pada Mei 23, 2021, dari <https://developer.android.com/studio/intro>
- Dewi, F. K. S., Indriasari, T. D., & Prayogo, Y. (2016). Rancang Bangun Aplikasi Pengingat Kegiatan Akademik Berbasis Mobile. *Jurnal Buana Informatika*, 7(4), 303–312. <https://doi.org/10.24002/jbi.v7i4.771>
- Dzihni, A. S., Andreswari, R., & Hasibuan, M. A. (2019). Business process analysis and academic information system audit of helpdesk application using genetic algorithms a process mining approach. *Procedia Computer Science*, 161, 903–909. <https://doi.org/10.1016/j.procs.2019.11.198>
- Febiharsa, D., Sudana, I. M., & Hudallah, N. (2018). Uji fungsionalitas (blackbox testing) sistem informasi lembaga sertifikasi profesi (silsp) batik dengan appperfect web test dan uji pengguna. *Joined Journal (Journal of Informatics Education)*, 1(2), 117-126.
- Fittryani, Y. P., & Ariantini, M. S. (2020). Perancangan Distribusi Sistem Informasi Untuk Optimalisasi Perubahan Jadwal Kuliah Berbasis Mobile Android (Studi Kasus: Stiki Indonesia). *Jurnal Teknologi Informasi Dan Komputer*, 6(2), 208–217.
- Hidayat, M., Soegiarto, H., Informatika, S. T., Selatan, K., Mahasiswa, A. P., & Android, B. (2019). Aplikasi Asisten Perkuliahan Mahasiswa STMIK Banjarbaru Berbasis Android. *Jutisi: Jurnal Ilmiah Teknik Informatika Dan Sistem Informasi*, 8(3), 151–160.
- Lastiansah, S. (2012). Pengertian User Interface. *Jakarta: PT. Elex Media Komputindo*.
- Lubis, S. (2017). *Implementasi Application Programming Interface (API) Dalam Upaya Peningkatan Pengelolaan dan Pelayanan Informasi Publik Pada Kantor KPU Kabupaten Tapanuli Selatan* (Doctoral dissertation, Universitas Medan Area).
- Malhotra, R., Kumar, D., & Gupta, D. P. (2020). An android application for campus information system. *Procedia Computer Science*, 172, 863–868. <https://doi.org/10.1016/j.procs.2020.05.124>
- Masturoh, S., Wijayanti, D., & Prasetyo, A. (2019). Sistem Informasi Akademik Berbasis Web Menggunakan Model Waterfall Pada SMK ITENAS Karawang. *Jurnal Informatika*, 6(1), 62-68.
- Morton, T. E., & Pentico, D. W. (1993). *Heuristic scheduling systems: with applications to*
- Nazruddin Safaat, H. (2012). Android: Pemrograman Aplikasi Mobile Smartphone dan Tablet PC (Edisi Revisi). *Informatika, Bandung*.

- Ningrum, F. C., Suherman, D., Aryanti, S., Prasetya, H. A., & Saifudin, A. (2019). Pengujian Black Box pada Aplikasi Sistem Seleksi Sales Terbaik Menggunakan Teknik Equivalence Partitions. *Jurnal Informatika Universitas Pamulang*, 4(4), 125-130.
- Novendri, M. S., Saputra, A., & Firman, C. E. (2019). Aplikasi Inventaris Barang Pada Mts Nurul Islam Dumai Menggunakan Php Dan Mysql. *lentera dumai*, 10(2).
- Nurudin, M., Jayanti, W., Saputro, R. D., Saputra, M. P., & Yulianti, Y. (2019). Pengujian Black Box pada Aplikasi Penjualan Berbasis Web Menggunakan Teknik Boundary Value Analysis. *Jurnal Informatika Universitas Pamulang*, 4(4), 143.
<https://doi.org/10.32493/informatika.v4i4.3841>
- Ong, S. P., Cholia, S., Jain, A., Brafman, M., Gunter, D., Ceder, G., & Persson, K. A. (2015). The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. *Computational Materials Science*, 97, 209–215.
<https://doi.org/10.1016/j.commatsci.2014.10.037> *production systems and project management*. New York: Wiley.
- Rahmawati, N. A., & Bachtiar, A. C. (2018). Analisis dan perancangan sistem informasi perpustakaan sekolah berdasarkan kebutuhan sistem. *Berkala Ilmu Perpustakaan Dan Informasi*, 14(1), 76. <https://doi.org/10.22146/bip.28943>
- Ratna Indrawati, Rudy Dwi Nyoto, T. M. (2017). Rancang Bangun Aplikasi Jadwal Kegiatan Akademik Berbasis Android. *Sistem Dan Teknologi Informasi*, 1(02), 1–5.
- Rolly, N., & Hakiem, N. (2015). PENGEMBANGAN APLIKASI MOBILE ACADEMIC INFORMATION SYSTEM (AIS) BERBASIS ANDROID UNTUK PENGGUNA DOSEN DAN MAHASISWA (Studi Kasus : Pusat Teknologi Informasi dan Pangkalan Data (Pustipanda) UIN Syarif Hidayatullah Jakarta). *Jurnal Teknik Informatika*, 8(1), 16–21. <https://doi.org/10.15408/jti.v8i1.1932>
- Sanjaya, G. T., & Sumboro, B. (2015). Rancang Bangun Aplikasi Penjadwalan Kuliah STMIK AUB SURAKARTA Berbasis Web. *Jurnal Ilmiah Go Infotech*, 21(1), 37–42.
- Saputra, R. A., & Santoso, S. (2021). Aplikasi Multimedia Untuk Karaoke Online. *Jurnal STRATEGI-Jurnal Maranatha*, 3(1), 94-108.
- Setiawan, D., Suratno, T., & Lutfi, L. (2018). Analisis, Desain dan Rancang Bangun Sistem Informasi Akademik Mobile Berbasis Android. *Elkha*, 10(2), 73.
<https://doi.org/10.26418/elkha.v10i2.28272>
- Sugiyono, P. D. (2015). Metode Penelitian Pendidikan Pendekatan Kuantitatif, Kualitatif, dan R&D Bagian III.
- Susanti, A. R., & Hidayattullah, M. F. (2016). SISTEM PENDUKUNG KEPUTUSAN BEASISWA PPA DAN BBP-PPA PADA POLITEKNIK MUHAMMADIYAH PEKALONGAN BERBASIS WEB. *Jurnal Surya Informatika: Membangun Informasi dan Profesionalisme*, 2(1).

- Syahlan, M. (2019). Perancangan Aplikasi Papan Informasi Berbasis Android Pada STMIK Dipanegara Makassar. *Prosiding Seminar Ilmiah Sistem Informasi Dan Teknologi Informasi, VIII(2)*, 27–40.
- Utama, Y. (2011). Sistem informasi berbasis web jurusan sistem informasi fakultas ilmu komputer universitas sriwijaya. *JSI: Jurnal Sistem Informasi (E-Journal)*, 3(2).
- Vikasari, C. (2018). Pengujian Sistem Informasi Magang Industri dengan Metode Blackbox Testing Boundary Value Analysis. *Syntax : Jurnal Informatika*, 7(1), 44–51.
<https://journal.unsika.ac.id/index.php/syntax/article/view/1291>
- Wahid, A. A. (2020). Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi. *Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK, October*, 1-5.
- Whitten, J. L., & Bentley, L. D. (2007). *Systems analysis and design methods*. New York: McGraw-Hill.
- Wijaya, I. W. G. S. (2012). Penerapan Web Service pada Aplikasi Sistem Akademik pada Platform Sistem Operasi Mobile Android. *Teknik Informatika, STIKOM PGRI Banyuwangi*, 1–6.
- Yati, M. N. (2018). APLIKASI BELAJAR MENGAJI BERBASIS ANDROID. *Jurnal Manajemen Informatika*, 8(2).

LAMPIRAN

