

**SISTEM PENCARIAN PENCERAMAH UNTUK WILAYAH
KOTA MAKASSAR MENGGUNAKAN ALGORITMA FLOYD
WARSHALL DAN HAVERSINE FORMULA BERBASIS
ANDROID**

**Searching System For Reservation For Makassar City Area Using
Floyd Warshall And Haversine Formula Algorithm Android
Based**

Oleh :

RIFALDY RAMADHAN L

2017130013



Dosen Pembimbing 1 : Prof. Dr. Ir.Andani Achmad, M.T

Dosen Pembimbing 2 : Dr.IT. Supriadi Sahibu, S.Kom., M.T.

**PROGRAM PASCASARJANA
STMIK HANDAYANI
MAKASSAR
2019**

PASCASARJANA
STMK HANDAYANI
PROGRAM STUDI SISTEM KOMPUTER
SISTEM Pencarian Penceramah Untuk Wilayah Kota
MAKASSAR MENGGUNAKAN ALGORITMA FLOYD WARSHALL
DAN HAVERSINE FORMULA BERBASIS ANDROID

Oleh :

RIFALDY RAMADHAN L

2017130013

Telah Di Pertahankan Di Depan Dewan Penguji Ujian Tesis Pada Hari Sabtu, 5
Oktober 2019 Dan Di Nyatakan Telah Memenuhi Syarat.

Mengetahui
Komisi penasehat

Pembimbing I

Prof. Dr. Ir. Andani Achmad, M.T
Ketua

Pembimbing II

Dr. IT. Supriadi Salibu, S.Kom., M.T.
Anggota

Mengetahui

Ketua Program Studi
Magister Sistem Komputer



Dr. Ir. Zahri Zainuddin, M.Sc

Direktur Program Pascasarjana
STMK HANDAYANI Makassar



Dr. Eng. Yuyun, S.Kom., M.T

PERNYATAAN KEASLIAN TESIS

Dengan ini saya yang bertanda tangan di bawah ini,

Nama : Rifaldy Ramadhan L
NIM : 2017130013
Program Studi : Pascasarjana Sistem Komputer
Judul Tesis : SISTEM PENCARIAN PENCERAMAH UNTUK WILAYAH
KOTA MAKASSAR MENGGUNAKAN ALGORITMA FLOYD
WARSHALL DAN HAVERSINE FORMULA BERBASIS
ANDROID

Menyatakan bahwa tesis dengan judul di atas beserta keseluruhan isi adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan. Atas pernyataan ini, saya siap menanggung resiko/sanksi yang dijatuhkan kepada saya apabila kemudian ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya ini, atau ada klaim dari pihak lain terhadap keaslian karya saya ini.

Makassar, 05 Oktober 2019

Yang Membuat Pernyataan,



Rifaldy Ramadhan L

NIM : 2017130013

KATA PENGANTAR

Puji dan syukur terhadap kehadiran ALLAH SWT atas rahmat dan karunia-Nya sehingga tesis dalam penelitian yang berjudul Sistem Penjadwalan Mata Kuliah Menggunakan Algoritma Genetika Pada STIMIK Handayani Makassar dapat terselesaikan sesuai dengan target, penelitian ini merupakan salah satu syarat untuk memperoleh gelar magister (S2) pada program studi Sistem Komputer STIMIK Handayani Makassar

Dalam penelitian ini penulis mengambil beberapa sumber baik dari internet dalam bentuk jurnal, skripsi, tesis maupun penelitian terdahulu yang berkaitan dengan topik penelitian ini yang dijadikan sebagai referensi. Penulis menyadari bahwa laporan penelitian ini masih jauh dari kesempurnaan oleh karena itu penulis sangat mengharapkan saran, masukan, dan kritik dari para pembaca untuk pengembangan sistem lebih lanjut.

Pada kesempatan ini penulis menghanturkan terima kasih sebesar besarnya kepada semua pihak yang telah membantu dalam penyusunan laporan penelitian khususnya kepada:

1. Dr. Eng. Yuyun, MT Selaku Direktur Pascasarjana STIMIK Handayani.
2. Dr.ir.Zahir Zainuddin, M.Sc Selaku Ketua Program Studi Sistem Komputer.
3. Prof. Dr. Ir.Andani Achmad, M.T dan Dr.IT. Supriadi Sahibu, S.Kom., M.T. Selaku Ketua Komisi Penasehat dan Anggota atas bantuan, arahan, bimbingan yang telah diberikan mulai dari tahap pembimbingan proposal penelitian, pelaksana penelitian sampai dengan penulisan hasil penelitian ini.
4. Bapak/Ibu dosen pada Program Pascasarjana STIMIK Handayani Makassar yang telah memberikan dan mentransformasikan ilmunya kepada penulis.
5. Orang Tua kami Ir. Abd. Latief dan Dra. Sumarni yang telah berjuang dalam mendidik dan membesarkan kami.

6. Fierda Amalia Hamzah, S.Kom. yang telah memberikan doa dan motivasi serta dukungan moril selama proses perkuliahan dan menyelesaikan tugas akhir ini.
7. Kepada teman teman mahasiswa Program Pascasarjana STIMIK Handayani Program Sistem Komputer yang telah berjuang Bersama sama dalam proses awal hingga akhir proses perkuliahan.

Dengan segala kerendahan hati penulis berharap semoga apa yang disusun dalam penelitian ini dapat bermanfaat bagi semua pihak.

Makassar, 05 Oktober 2019

Penulis

Rifaldy Ramadhan L
NIM : 2017130013

ABSTRAK

Rifaldy Ramadhan I. “Sistem Pencarian Penceramah Untuk Wilayah Kota Makassar Menggunakan Algoritma Floyd Warshall dan Haversine Formula Berbasis Android”, dibimbing oleh **Andani Achmad** dan **Supriadi Sahibu**.

Dakwah adalah kegiatan yang bersifat menyeru, mengajak dan memanggil orang untuk beriman dan taat kepada Allah sesuai garis aqidah, syariat dan akhlak islam. Kegiatan berdakwah sering dilaksanakan oleh umat Islam dalam berbagai kegiatan keislaman yang dilakukan oleh para pendakwah/penceramah. Namun masih sulit untuk menentukan jadwal dan lokasinya. Di kota Makassar sendiri, terdapat banyak Masjid dan lokasi ceramah yang masih minim informasi alamatnya, sehingga penceramah sering kali kesulitan menemukan lokasi ceramah.

Tujuan dari penelitian ini adalah untuk memudahkan masyarakat mencari penceramah dan memudahkan penceramah mencari lokasi ceramah dengan merancang suatu sistem pencarian sebagai media informasi penyedia jasa dakwah untuk masyarakat. Dalam melakukan penelitian ini, jenis penelitian yang digunakan adalah metode penelitian kualitatif dengan menggunakan strategi Design dan creation sebagai konsep yang sangat tepat untuk mengolah penelitian ini. Hasil dari penelitian ini berupa aplikasi mobile dimana masyarakat dapat mencari penceramah dan penceramah dapat menentukan lokasi ceramah dengan lebih mudah.

Berdasarkan pernyataan diatas, bahwa dakwah Sangat berperan penting dalam kehidupan sehari-hari. Untuk itu perlu dibuatkan sebuah aplikasi berbasis Android yang dapat membantu masyarakat muslim kota Makassar untuk mencari penceramah disetiap kegiatan keislaman untuk mengisi acara dakwah.

Kata kunci : **Pencarian, Dakwah, Ustadz, Android**

ABSTRACT

Rifaldy Ramadhan L "Lecturer Search System for Makassar City Using the Floyd Warshall Algorithm and Haversine Formula Android Based ", guided by **Andani Achmad** and **Supriadi Sahibu..**

Da'wah is an activity that is called for, invites and calls people to believe and obey God in accordance with the lines of aqeedah, syariah and Islamic morals. Preaching activities are often carried out by Muslims in various Islamic activities carried out by preachers / preachers. But it is still difficult to determine the schedule and location. In the city of Makassar itself, there are many mosques and lecture locations that still lack address information, so preachers often have difficulty finding the location of lectures.

The purpose of this research is to make it easier for the public to look for lecturers and make it easier for lecturers to find the location of lectures by designing a search system as a medium for preaching service information for the public. In conducting this research, the type of research used is a qualitative research method using the Design and creation strategy as a very appropriate concept for processing this research. The results of this study are in the form of a mobile application where people can look for lecturers and lecturers can determine the location of lectures more easily.

Based on the statement above, that preaching is very important role in everyday life. For this reason it is necessary to make an Android-based application that can help the Muslim community in Makassar to look for preachers in every Islamic activity to fill da'wah events.

Keywords: Search, Da'wah, Ustadz, Android

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGAJUAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PERNYATAN KEASLIAN.....	iv
KATA PENGANTAR.....	v
ABSTRAK	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xiv
BAB I PENDAHULUAN.....	1
A. Latar Belakang Masalah.....	1
B. Rumusan Masalah	4
C. Tujuan Penelitian.....	4
D. Manfaat Penelitian.....	5
E. Ruang Lingkup Dan Batas Penelitian	5
F. Sistematika Penulisan.....	6
BAB II TINJAUAN PUSTAKA.....	8
A. Tinjauan Teori Dan Konsep	8
1. Lembaga Dakwah Islam Indonesia	8
2. Penjadwalan Khatib dan Penceramah	9
a. Khatib dan Penceramah	9
b. Penjadwalan Khatib dan Penceramah	10
3. Sistem Informasi.....	10
4. Aplikasi	12
5. Tentang Android.....	13
a. Pengertian Android.....	13
b. Komponen Android	13
c. Komponen Kebutuhan Aplikasi	14
B. RoadMap Penelitian	16
C. Kerangka Berfikir.....	19

BAB III METODOLOGI PENELITIAN	20
A. Jenis dan Lokasi Penelitian	20
B. Pendekatan Penelitian.....	20
C. Sumber Data	20
D. Metode Pengumpulan Data	20
E. Alat dan Bahan Penelitian	21
1. Perangkat Keras.....	21
2. Perangkat Lunak.....	21
F. Teknik Pengolahan dan Analisi Data	22
G. Metode Dan Perancangan Sistem.....	22
1. <i>Requirement System</i>	23
2. <i>Analysis</i>	24
3. <i>Design</i>	24
4. <i>Coding</i>	24
5. <i>Implementation</i>	24
6. <i>Evaluation</i>	25
BAB IV HASIL DAN PEMBAHASAN	26
A. Analisis Sistem yang sedang berjalan	26
B. Analisis Sistem yang diusulkan.....	27
1. Analisis Masalah	27
2. Analisa Kebutuhan	27
3. Analisis kelemahan.....	29
C. Perancangan Sistem.....	29
1. <i>Use Case Diagram</i>	29
2. <i>Class Diagram</i>	30
3. <i>Sequence Diagram</i>	31
4. <i>Activity Diagram</i>	35
5. Perancangan <i>Eternity Relationship Diagram (ERD)</i>	36
6. Perancangan Tabel	36
7. Struktur Navigasi.....	38
8. Perancangan Antarmuka (<i>Interface</i>)	39
D. Implementasi Sistem	49
E. Hasil Pengujian.....	60
1. Prosedur Pengujian.....	60

2. Hasil Pengujian Metode	61
3. Hasil Pengujian Sistem.....	63
BAB V PENUTUP.....	68
A. Kesimpulan.....	68
B. Saran	69
DAFTAR PUSTAKA	70
LAMPIRAN.....	74

DAFTAR GAMBAR

Gambar

Gambar II. 1 Kerangka Berfikir	19
Gambar III. 1 Analisis Sistem.....	23
Gambar III. 2 Model <i>Waterfall</i>	24
Gambar IV. 1 <i>Flowmap</i> Sistem yang Sedang Berjalan.....	26
Gambar IV. 2 <i>Use Case Diagram</i>	29
Gambar IV. 3 <i>Class Diagram</i>	30
Gambar IV. 4 <i>Sequence Diagram</i> Menu Utama/ <i>Home</i>	31
Gambar IV. 5 <i>Sequence Diagram</i> Pencarian Penceramah.....	32
Gambar IV. 6 <i>Sequence Diagram</i> Pendaftaran Jadwal.....	33
Gambar IV. 7 <i>Sequence Diagram</i> Persetujuan Atur Jadwal	34
Gambar IV. 8 <i>Activity Diagram</i>	35
Gambar IV. 9 <i>Entity Relationship Diagram</i> (ERD).....	36
Gambar IV. 10 Struktur Navigasi	38
Gambar IV. 11 Desain Antarmuka <i>Splashscreen</i>	39
Gambar IV. 12 Desain Antarmuka <i>Login</i>	40
Gambar IV. 13 Desain Antarmuka Pendaftaran <i>User</i>	41
Gambar IV. 14 Desain Antarmuka Menu Utama	42
Gambar IV. 15 Desain Antarmuka Menu Lokasi	43
Gambar IV. 16 Desain Antarmuka Menu Pencarian	44
Gambar IV. 17 Desain Antarmuka Jadwal Ustadz	45
Gambar IV. 18 Desain Antarmuka Permintaan Jadwal Ceramah.....	46
Gambar IV. 19 Desain Antarmuka Menu Akun <i>User</i>	47
Gambar IV. 20 Desain Antarmuka Menu Akun Ustadz	48
Gambar IV. 21 <i>Splash Screen</i>	49

Gambar IV. 22 <i>Login</i>	50
Gambar IV. 23 Pendaftaran <i>User</i>	51
Gambar IV. 24 Menu Utama.....	53
Gambar IV. 25 Antarmuka Menu Lokasi	54
Gambar IV. 26 Antarmuka Menu Pencarian.....	55
Gambar IV. 27 Antarmuka Menu Jadwal Ustadz	56
Gambar IV. 28 Antarmuka Permintaan Jadwal Ceramah.....	57
Gambar IV. 29 Antarmuka Menu Akun <i>User</i>	58
Gambar IV. 30 Antarmuka Menu Akun Ustadz	59

DAFTAR TABEL

Tabel

Tabel IV. 1 Tabel User.....	36
Tabel IV. 2 Tabel Ustadz	37
Tabel IV. 3 Tabel Jadwal	37
Tabel IV. 4 Tabel Approval	38
Tabel IV. 5 Pengujian Metode	61
Tabel IV. 6 Pengujian Form Login	63
Tabel IV. 7 Pengujian Menu Daftar User	64
Tabel IV. 8 Pengujian Menu Lokasi	64
Tabel IV. 9 Pengujian Menu Pencarian	65
Tabel IV. 10 Pengujian Menu Jadwal Ustadz.....	66
Tabel IV. 11 Pengujian Menu Permintaan Jadwal Ceramah	66
Tabel IV. 12 Pengujian Menu Akun User.....	67
Tabel IV. 13 Pengujian Menu Akun Ustadz	67

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Dakwah yang dipersepsikan oleh masyarakat pada umumnya adalah kegiatan penyampaian pesan verbal yang berisi nasehat-nasehat agama beserta dalil-dalil yang dijadikan rujukan. Mereka yang menyampaikan dakwah sering disebut sebagai ustadz, kiyai, ulama, atau mereka yang dinilai sebagai ahli agama (Akhyar, 2014).

Di Indonesia, ada berbagai macam cara penceramah menyampaikan isi ceramahnya sesuai bahasa dan dialek daerah tempat ia berceramah. Semakin dekat bahasa yang digunakan dengan bahasa daerah setempat, maka lebih mudah isi dan pesan ceramahnya diterima. Oleh karena itu, terkadang ada beberapa daerah yang hanya memprioritaskan penceramah tertentu untuk berceramah di tempatnya.

Dengan adanya kendala bahasa dan banyaknya penceramah-penceramah di Indonesia, maka dibentuk sebuah organisasi yang bergerak di bidang dakwah untuk mendata dan mengkoordinir setiap penceramah. Organisasi-organisasi ini disebut Lembaga Dakwah. Hampir disetiap kota di Indonesia terdapat lembaga dakwah. Namun, meskipun sudah dibentuk lembaga dakwah, terkadang masih sering terjadi kendala seperti miskomunikasi, penceramah-penceramah baru yang belum terdata oleh lembaga dakwah.

Di kota Makassar terdapat ratusan masjid dan lokasi ceramah yang masih minim informasi alamatnya sehingga khatib atau penceramah seringkali kesulitan menemukan lokasi yang ditempati khutbah atau ceramah. Kesulitan ini

menyebabkan khatib atau penceramah kadang tidak melaksanakan tugasnya di lokasi yang telah dijadwalkan. Alamat atau lokasi menjadi informasi agar khatib dan penceramah dapat menemukan lokasi yang ditempati khutbah atau ceramah. Meskipun pada penelitian terdahulu, sudah dibuat sebuah sistem terintegrasi manajemen penjadwalan khatib pada masjid-masjid yang terdaftar pada lembaga dakwah, akan tetapi sistem tersebut kurang fleksibel. Hal ini dikarenakan, sistem terdahulu hanya dimanfaatkan oleh Lembaga Dakwah dan beberapa masjid yang telah terdaftar. Sehingga, terbatasnya informasi untuk ustadz yang belum terdaftar.

Perkembangan teknologi merupakan salah satu bidang ilmu pengetahuan yang berkembang pesat saat ini. Adapun *handphone* merupakan salah satu teknologi yang kebanyakan masyarakat telah gunakan dan manfaatkan. *Handphone* berevolusi menjadi *smartphone*, yang dulu hanya terbatas kegunaannya untuk menelfon dan mengirim pesan singkat, kini telah banyak memberi manfaat dan kemudahan bagi penggunanya dalam segala bidang. *Smartphone* dengan sistem operasi Android memungkinkan banyak aplikasi yang bisa dipasang dan dimanfaatkan pada *smartphone* itu sendiri.

Smartphone merupakan telepon genggam yang mempunyai kemampuan tingkat tinggi, bahkan dengan fungsi yang menyerupai komputer. Tentunya banyak sekali fasilitas-fasilitas yang diberikan oleh *smartphone* sehingga saat ini banyak sekali peminatnya. Jika awalnya pengguna *smartphone* ini adalah para pejabat atau karyawan swata, wanita karir juga eksekutif muda. Namun saat ini pengguna *smartphone* sudah meluas ke semua kalangan (Lutfi Parmuarip, 2012).

Android merupakan salah satu platform dari perangkat smartphone. Salah satu keutamaan dari Android yaitu lisensinya yang bersifat terbuka (*open source*) dan gratis (*free*) sehingga bebas untuk dikembangkan karena tidak ada biaya royalti maupun didistribusikan dalam bentuk apapun. Hal ini memudahkan para programmer untuk membuat aplikasi baru di dalamnya (Gunita Mustika Hati, 2013).

Adapun penelitian yang dilakukan oleh *Hakim* pada 2014. Penelitian ini berjudul “*Sistem Informasi Penjadwalan Kunjungan Perpustakaan Keliling Berbasis Web dengan SMS Gateway pada Kantor Perpustakaan dan Arsip Daerah Kabupaten Kudus*”. Tujuan dari tesis ini adalah untuk merancang dan membangun suatu sistem penjadwalan kunjungan perpustakaan keliling beserta pemberian informasi jadwal kunjungan melalui sms guna memudahkan koordinator perpustakaan keliling dalam mengelola jadwal kunjungan perpustakaan keliling beserta informasinya. Konsep yang diterapkan dalam tahap perancangan Sistem Penjadwalan Kunjungan Perpustakaan Keliling Berbasis Web dengan SMS Gateway pada Kantor Perpustakaan dan Arsip Daerah Kabupaten Kudus adalah dengan menggunakan Waterfall. Sedangkan bahasa pemrograman yang digunakan adalah PHP.

Dengan ilmu pengetahuan, manusia dapat mengetahui benda-benda langit, menjelajahi angkasa raya, dan menembus sekat-sekat yang selama ini belum terkuak. Manusia diberi potensi oleh Allah swt. berupa akal. Akal ini harus terus diasah, diberdayakan dengan cara belajar dan berkarya. Dengan belajar, manusia

bisa mendapatkan ilmu dan wawasan yang baru. Dengan ilmu, manusia dapat berkarya untuk kehidupan yang lebih baik.

B. Rumusan Masalah

Berdasarkan pada latar belakang masalah diatas, maka rumusan permasalahan dari penelitian ini sebagai berikut :

1. Bagaimana mengimplementasikan sebuah aplikasi yang berjalan pada sistem operasi berbasis android minimal versi android 4.4: kitkat.?
2. Bagaimana mengimplementasikan aplikasi pencarian lokasi rumah penceramah dengan memanfaatkan memanfaatkan *Global Positioning System* (GPS) dan *Google Map* API?
3. Bagaimana menerapkan metode Floyd Warshall dan Haversine Formula dalam menentukan jarak terdekat pada aplikasi?
4. Bagaimana menerapkan radius dalam menampilkan jarak terdekat pada aplikasi?

C. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk merancang dan membangun suatu aplikasi berbasis android yang dapat memudahkan masyarakat untuk mencari penceramah dan penceramah dapat mencari lokasi ceramah.

1. Mengimplementasikan sebuah aplikasi yang berjalan pada sistem operasi berbasis android minimal versi android 4.4: kitkat.

2. Mengimplementasikan aplikasi pencarian lokasi rumah penceramah dengan memanfaatkan memanfaatkan *Global Positioning System* (GPS) dan Google *Map* API.
3. Menerapkan metode Floyd Warshall dan Haversine Formula dalam menentukan jarak terdekat pada aplikasi?
4. Menerapkan radius dalam menampilkan jarak terdekat pada aplikasi?

D. Manfaat Penelitian

a. Kegunaan Teoritis

Secara teoritis, hasil dari penelitian ini dapat menjadi referensi atau masukan bagi perkembangan teknologi informasi dan menambah kajian teknologi informasi khususnya dalam pencarian suatu objek.

b. Kegunaan Praktis

Hasil penelitian ini secara praktis diharapkan dapat memberi manfaat bagi masyarakat dalam pencarian penceramah dan informasi terkait.

E. Ruang Lingkup dan Batasan Penelitian

Dalam penyusunan tugas akhir ini perlu adanya pengertian pada pembahasan yang terfokus sehingga permasalahan tidak melebar. Adapun fokus penelitian ini adalah :

1. Aplikasi ini dapat digunakan masyarakat untuk mencari penceramah menggunakan *smartphone* Android minimal versi android 4.4: kitkat.
2. Aplikasi ini menjelaskan profil penceramah.
3. Aplikasi ini menampilkan lokasi penceramah.

4. Aplikasi ini mengurutkan berdasarkan lokasi penceramah terdekat.
5. Aplikasi ini merupakan sistem penjadwalan khatib dan penceramah.
6. Aplikasi ini menggunakan algoritma Floyd Warshall dan Haversine Formula.

Untuk mempermudah pemahaman dan memberikan gambaran serta menyamakan persepsi antara peneliti dan pembaca, maka dikemukakan penjelasan yang sesuai dalam penelitian ini. Adapun yang dijelaskan dalam penelitian ini adalah:

1. Aplikasi ini akan memudahkan masyarakat dalam mencari penceramah.
2. Pengguna akan mengetahui informasi lengkap mengenai profil dan jadwal penceramah.
3. Pengguna akan mengetahui informasi jadwal ceramah pada setiap khatib.
4. Aplikasi ini merupakan pengembangan dari sistem penjadwalan khatib dan penceramah yang telah di terapkan kementrian agama di setiap masjid.

F. Sistematika Penulisan

Dalam penulisan tesis ini dibagi menjadi enam bab dengan beberapa sub pokok bahasan. Adapun sistematika dari tesis ini adalah sebagai berikut:

BAB I : PENDAHULUAN

Dalam bab ini akan diuraikan latar belakang penulisan, ruang lingkup atau batasan dalam penelitian ini, tujuan yang hendak dicapai, manfaat yang diharapkan dan metodologi yang digunakan dalam penelitian ini secara sistematis.

BAB II : TINJAUAN PUSTAKA

Dalam bab ini akan diuraikan tentang teori-teori yang digunakan dalam pembahasan penulisan tesis ini dan sumber teori-teori tersebut.

BAB III : METODE PENELITIAN

Dalam bab ini akan menjelaskan tentang tata cara yang digunakan dalam penelitian berdasarkan metodologi penelitian yang dipilih. Metode yang digunakan antara lain metode pengumpulan data, menggunakan metode wawancara dan studi literatur.

BAB IV : HASIL PENELITIAN DAN PEMBAHASAN

Dalam bab ini akan dibahas tentang analisis sistem, dan perancangan sistem serta implementasi sistem dan hasil pengujian sistem.

BAB VI : PENUTUP

Bab ini merupakan akhir penulisan tesis, dimana berdasarkan uraian-uraian yang telah dibahas akan dituangkan dalam suatu bentuk kesimpulan akhir serta saran-saran.

BAB II

TINJAUAN PUSTAKA

A. Tinjauan Teori dan Konsep

1. Lembaga Dakwah Islam Indonesia

Lembaga Dakwah Indonesia merupakan organisasi kemasyarakatan yang independen, resmi dan legal yang mengikuti ketentuan UU No. 8 tahun 1985 tentang Organisasi Kemasyarakatan, Pasal 9 ayat (2), tanggal 4 April 1986 (Lembara Negara RI 1986 nomor 24), serta pelaksanaannya meliputi PP No. 18 tahun 1986 dan peraturan Menteri Dalam Negeri No 5 tahun 1986 dan Aturan Hukum lainnya. Lembaga Dakwah Islam Indonesia memiliki Anggaran Dasar (AD) dan Anggaran Rumah Tangga (ART), program kerja dan pengurus mulai dari tingkat pusat sampai tingkat Desa. Lembaga Dakwah Islam Indonesia sudah tercatat di Badan Kesatuan Bangsa dan Perlindungan Masyarakat (Bakesbang & Linmas) Departemen Dalam Negeri.

Pada umumnya suatu lembaga dakwah memiliki peran penting didalam masyarakat, adanya suatu lembaga dakwah dikarenakan memiliki tujuan untuk kepentingan bersama. Secara garis besar suatu lembaga dakwah berperan sebagai pengontrol di dalam masyarakat guna perbaikan bangsa.

Adapun peran lainnya sebagai lembaga dakwah adalah:

- a. Menebar pemikiran dan dakwah
- b. Mengembangkan kemampuan SDM para kader dakwah
- c. Pelembagaan yang professional dan kompeten pada bidangnya

- d. Menghimpun tokoh dan pakar yang siap memberikan kontribusi dan pemikiran serta pengaruh bagi kepentingan dakwah
- e. Melayani, melindungi, serta memberdayakan masyarakat.

2. Penjadwalan Khatib dan Penceramah

Penjadwalan adalah kegiatan pengalokasian sumber-sumber atau mesin-mesin yang ada untuk menjalankan sekumpulan tugas dalam jangka waktu tertentu. (Baker,1974). Penjadwalan produksi adalah suatu kegiatan memasukkan sejumlah produk yang telah direncanakan ke dalam proses pengerjaannya (Biegel,1992). Penjadwalan adalah proses pengurutan pembuatan produk secara menyeluruh pada beberapa mesin (Conway,et,al,1967). Penjadwalan juga didefinisikan sebagai rencana pengaturan urutan kerja serta pengalokasian sumber, baik waktu maupun fasilitas untuk setiap operasi yang harus diselesaikan (Vollman,1998). Dari beberapa definisi yang telah disebutkan maka dapat ditarik satu definisi “Penjadwalan adalah suatu kegiatan perancangan berupa pengalokasian sumber daya baik mesin maupun tenaga kerja untuk menjalankan sekumpulan tugas sesuai prosesnya dalam jangka waktu tertentu”.(Onikom, 2006)

a. Khatib dan Penceramah

Dalam pengertian umum, khatib merupakan sebutan untuk orang-orang yang berpidato. Dalam pengertian khusus adalah sebutan untuk orang yang khutbah pada saat shalat Jumat dan shalat Id. Kata khatib selain dipakai jabatan, dipakai pula sebagai gelar seseorang. Pada masa pra-Islam, khatib mempunyai kedudukan tinggi di kalangan masyarakat Arab. Pada masa itu banyak khatib yang mampu menciptakan prosa bersajak (annasr al-masju') secara alami,

sehingga kehadiran khatib di kalangan mereka sama dengan penyair yang mempunyai kedudukan tinggi dalam masyarakat.

Khatib merupakan bagian penting dalam pelaksanaan shalat jumat. Bisa dikatakan keberadaan khatib ini memang wajib ada saat menjalankan ibadah bagi kaum pria pada hari Jumat tersebut. Penyebutan istilah khatib berbeda dengan penceramah, kalau khatib adalah penceramah yang bukan hanya pandai mengolah kata secara lisan, namun dia harus mengetahui seluk beluk agama islam dan menjadi ahli ibadah. Kalau ungkapan penceramah adalah mereka yang sering melakukan ceramah sekalipun tidak mengetahui seluk beluk apa yang diceramahkan. Khatib adalah panggilan bagi seorang pria yang memimpin shalat jumat sekaligus memberikan ceramah kepada jamaahnya.

b. Penjadwalan khatib dan penceramah

suatu kegiatan perancangan berupa pengalokasian atau penempatan khatib dan penceramah pada masjid untuk menjalankan tugas dakwah dari lembaga dakwah dalam jangka waktu tertentu.

3. Sistem Informasi

Sistem Informasi (SI) adalah kombinasi dari teknologi informasi dan aktivitas orang yang menggunakan teknologi itu untuk mendukung operasi dan manajemen. Dalam arti yang sangat luas, istilah sistem informasi yang sering digunakan merujuk kepada interaksi antara orang, proses algoritmik, data, dan teknologi. Dalam pengertian ini, istilah ini digunakan untuk merujuk tidak hanya pada penggunaan organisasi teknologi informasi dan komunikasi (TIK), tetapi

juga untuk cara di mana orang berinteraksi dengan teknologi ini dalam mendukung proses bisnis.

Ada yang membuat perbedaan yang jelas antara sistem informasi, dan komputer sistem TIK, dan proses bisnis. Sistem informasi yang berbeda dari teknologi informasi dalam sistem informasi biasanya terlihat seperti memiliki komponen TIK. Hal ini terutama berkaitan dengan tujuan pemanfaatan teknologi informasi. Sistem informasi juga berbeda dari proses bisnis. Sistem informasi membantu untuk mengontrol kinerja proses bisnis.

(Alter,2009) berpendapat untuk sistem informasi sebagai tipe khusus dari sistem kerja. Sistem kerja adalah suatu sistem di mana manusia dan/atau mesin melakukan pekerjaan dengan menggunakan sumber daya untuk memproduksi produk tertentu dan/atau jasa bagi pelanggan. Sistem informasi adalah suatu sistem kerja yang kegiatannya ditujukan untuk pengolahan (menangkap, transmisi, menyimpan, mengambil, memanipulasi dan menampilkan) informasi. Dengan demikian, sistem informasi antar-berhubungan dengan sistem data di satu sisi dan sistem aktivitas di sisi lain. Sistem informasi adalah suatu bentuk komunikasi sistem di mana data yang mewakili dan diproses sebagai bentuk dari memori sosial. Sistem informasi juga dapat dianggap sebagai bahasa semi formal yang mendukung manusia dalam pengambilan keputusan dan tindakan. Sistem informasi merupakan fokus utama dari studi untuk disiplin sistem informasi dan organisasi informatika.

Sistem informasi adalah gabungan yang terorganisasi dari manusia, perangkat lunak, perangkat keras, jaringan komunikasi dan sumber data dalam mengumpulkan, mengubah, dan menyebarkan informasi dalam organisasi.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

4. Aplikasi

Menurut Dhanta (2009:32), Aplikasi (*application*) adalah *software* yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya *Microsoft word*, *Microsoft excel*, sedangkan menurut anisyah (2000:30), aplikasi merupakan *software* yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan, dan penambahan data.

Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut suatu paket atau suite aplikasi (*application suite*). Contohnya adalah *Microsoft office* dan *openoffice.org*, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan dan mempelajari tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan

pengguna. Contohnya, suatu lembar kerja dapat dinamakan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

5. *Android*

a. **Pengertian *Android***

Android merupakan sistem operasi untuk telepon seluler yang berbasis *linux*. *android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam peranti bergerak. Awalnya, *Google Inc.* membeli *android Inc.*, pendatang baru yang membuat peranti lunak untuk *smartphone*. Kemudian untuk mengembangkan *android*, dibentuklah *open handset alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

b. **Komponen *Android***

Berikut ini adalah komponen pada aplikasi *android* yaitu:

- 1) *Activities*, Suatu *activity* akan menyajikan *user interface* (UI) kepada pengguna sehingga pengguna dapat melakukan interaksi untuk menjalankan fungsi tertentu. Sebuah aplikasi *Android* bisa jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut.
- 2) *Service*, *Service* tidak memiliki *Graphic User Interface* (GUI), tetapi *service* berjalan secara *background* untuk melakukan operasi-

operasi yang *longrunning* (proses yang memakan waktu cukup lama) atau melakukan operasi untuk proses *remote*.

- 3) *Broadcast Receiver*, *Broadcast Receiver* berfungsi menerima dan bereaksi untuk menyiapkan *notifikasi*. *Broadcast Receiver* tidak memiliki *user interface (UI)* tapi memiliki sebuah *activity* untuk merespon informasi yang diterima atau kepada pengguna. *Broadcast Receiver* hanyalah pintu gerbang menuju komponen lain dan memang dirancang untuk hanya melakukan kerja seminimal mungkin.
- 4) *Content Provider*, *Content Provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. *Content providers* juga berguna untuk membaca dan menulis data yang berstatus *private* dan tidak dibagikan ke suatu aplikasi.

c. **Komponen Kebutuhan Aplikasi**

1) *SQLite*

SQLite (Structured Query Lite) merupakan sebuah sistem manajemen basis data relasional yang bersifat *ACID-compliant* dan memiliki ukuran pustaka kode yang relatif kecil, ditulis dalam bahasa C. *SQLite* merupakan proyek yang bersifat *public domain* yang dikerjakan oleh D. Richard Hipp.

2) *Eclipse*

Eclipse adalah sebuah IDE (*Integrated Development Environment*) untuk mengembangkan perangkat lunak dan dapat dijalankan di

semua *platform (platform-independent)*. Berikut ini adalah sifat dari *eclipse*: *Multi-platform*: Target sistem operasi *eclipse* adalah *microsoft windows, linux, solaris, AIX, HP-UX dan mac OS X*; *Multi-language*: *eclipse* dikembangkan dengan bahasa pemrograman *java*, akan tetapi *eclipse* mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti *C/C++, cobol, python, perl, PHP*, dan lain sebagainya. *multi-role*: Selain sebagai *IDE* untuk pengembangan aplikasi, *eclipse* pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, test perangkat lunak, pengembangan *web*, dan lain sebagainya.

3) ADT (*Android Development Tools*)

Android Development Tools (ADT) adalah *plug-in* untuk *eclipse IDE* yang dirancang untuk memberikan lingkungan yang *powerfull* dan terpadu untuk membangun aplikasi *android*. *ADT* memperluas kemampuan *eclipse* untuk mempercepat dalam pembuatan project *android* baru, membuat aplikasi UI, menambahkan komponen berdasarkan *android framework API*, *debug* aplikasi menggunakan *android tool SDK*, dan bahkan ekspor *unsigned.apk file* dalam rangka untuk mendistribusikan Aplikasi. Mengembangkan Aplikasi *android* di *eclipse* dengan *ADT* sangat dianjurkan dan merupakan cara tercepat untuk memulai membuat Aplikasi *android*. Dengan dipandu *set-up project*, serta integrasi peralatan, *custom XML editor*, dan *debug panel*

ouput, ADT dapat memberikan dorongan luar biasa dalam mengembangkan aplikasi *android*.

4) SDK (*Software Development Kit*)

Android SDK adalah *tools API (Application Programming Interface)* yang diperlukan untuk memulai mengembangkan aplikasi pada platform *android* menggunakan menggunakan bahasa pemrograman *java*. *Android* merupakan subset perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang dirilis oleh *google*. Saat ini SDK disediakan untuk alat bantu dan API untuk memulai mengembangkan aplikasi pada platform *android* menggunakan bahasa pemrograman *java* (*Safaat, 2012*).

B. RoadMap Penelitian

Roadmap ini digunakan sebagai pembanding antara penelitian yang sudah dilakukan dan yang akan dilakukan peneliti. Penelitian tersebut diantaranya sebagai berikut:

Dirman (2017) dalam penelitiannya yang berjudul “sistem terintegrasi manajemen penjadwalan khatib pada masjid-masjid yang terdaftar pada lembaga dakwah”. Penelitian ini bertujuan untuk membuat sebuah sistem yang terintegrasi terhadap manajemen penjadwalan khatib.

Terdapat kesamaan pada penelitian di atas yaitu sama-sama memanfaatkan sistem terintegrasi manajemen penjadwalan khatib, namun perbedaannya penelitian diatas hanya fokus untuk perancangan dan pembuatan sistem yang akan digunakan

lembaga dakwah untuk mengorganisir khatib-khatib. Sedangkan, pada penelitian ini akan berfokus pada pembuatan aplikasi berbasis Android.

Kedua, penelitian yang dilakukan oleh *Rody* pada 2011. yang berjudul “*Program Aplikasi Penjadwalan Khotib Pada Lembaga Corps Mubaligh Muhammadiyah Malang Berbasis Jsp Dan Xml*“. Dimana aplikasi ini bertujuan untuk merancang aplikasi penjadwalan khatib pada lembaga Corps Muballigh Muhammadiyah dengan menggunakan metode *Round Robin*. Perancangan dan implementasi akan dilakukan dengan Bahasa pemrograman *JavaScript* dan *Xml* serta basis data yang digunakan adalah *MySQL*.

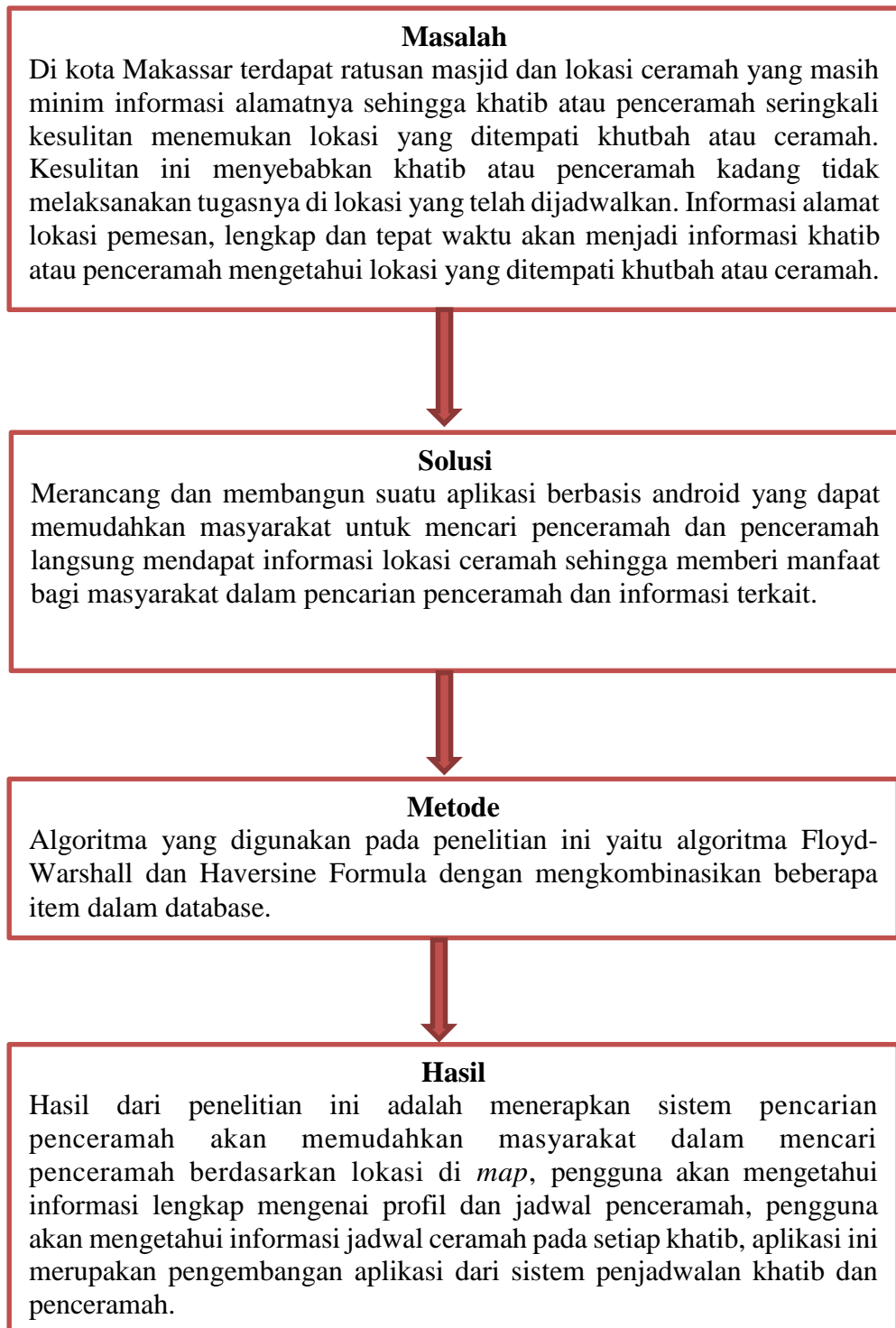
Program Aplikasi Penjadwalan Khatib Pada Lembaga Corps Mubaligh Muhammadiyah memiliki persamaan dan perbedaan dengan yang akan dibuat penulis. Adapun persamaannya adalah sama-sama melakukan penjadwalan khatib dengan menggunakan metode *Round Robin*. Adapun perbedaan dengan aplikasi yang akan dibuat oleh penulis ialah sistem pencarian penceramah berdasarkan lokasi di *map*.

Ketiga, penelitian yang dilakukan oleh *Hakim* pada 2014. Penelitian ini berjudul “*Sistem Informasi Penjadwalan Kunjungan Perpustakaan Keliling Berbasis Web dengan SMS Gateway pada Kantor Perpustakaan dan Arsip Daerah Kabupaten Kudus*”. Tujuan dari tesis ini adalah untuk merancang dan membangun suatu sistem penjadwalan kunjungan perpustakaan keliling beserta pemberian informasi jadwal kunjungan melalui sms guna memudahkan koordinator perpustakaan keliling dalam mengelola jadwal kunjungan perpustakaan keliling beserta informasinya. Konsep yang diterapkan dalam tahap perancangan Sistem

Penjadwalan Kunjungan Perpustakaan Keliling Berbasis Web dengan SMS *Gateway* pada Kantor Perpustakaan dan Arsip Daerah Kabupaten Kudus adalah dengan menggunakan Waterfall. Sedangkan bahasa pemrograman yang digunakan adalah PHP.

Penelitian ini memiliki kesamaan dengan sistem yang akan dibuat. Persamaannya adalah sama-sama memanfaatkan notifikasi untuk memberikan informasi kepada pengguna. Sedangkan perbedaannya adalah penelitian ini menggunakan bahasa pemrograman java (android).

C. Kerangka Berfikir



Gambar II.1 Kerangka Fikir

BAB III

METODE PENELITIAN

A. Jenis dan Lokasi Penelitian

Dalam melakukan penelitian ini, jenis penelitian yang di gunakan adalah penelitian kualitatif dimana strategi yang digunakan adalah *Design and Creation*. Dipilihnya jenis penelitian ini oleh penulis dikarenakan konsep dari *Design and Creation* sangat tepat untuk mengelolah penelitian ini.

Adapun lokasi penelitian ini dilakukan di laboratorium penelitian STMIK Handayani dan Pondok Pesantren Moderen Pendidikan Al Quran IMMIM Putra Makassar.

B. Pendekatan Penelitian

Penelitian ini menggunakan pendekatan penelitian saintifik yaitu pendekatan berdasarkan wawancara, observasi, ilmu pengetahuan dan teknologi.

C. Sumber Data

Sumber data pada penelitian ini adalah hasil observasi dari Pondok Pesantren Moderen Pendidikan Al Quran IMMIM Putra Makassar. Selain itu, dengan cara mengumpulkan data dari beberapa buku, jurnal, tesis, tesis maupun literatur lainnya yang dapat dijadikan acuan pembahasan dalam masalah ini. Sumber data juga diperoleh dari data *online* atau internet.

D. Metode pengumpulan Data

Metode pengumpulan data yang digunakan pada penelitian untuk aplikasi ini yaitu:

1. Observasi yaitu melakukan pengamatan langsung dilokasi-lokasi yang masih sering digunakan sebagai tempat-tempat diskusi baik diruang-ruang kuliah, secretariat, maupun warung kopi.
2. Wawancara merupakan teknik pengumpulan data yang dilakukan dengan pihak-pihak yang terkait dengan masalah pengolahan data, maupun dengan masyarakat.

E. Alat dan Bahan Penelitian

1. Perangkat Keras

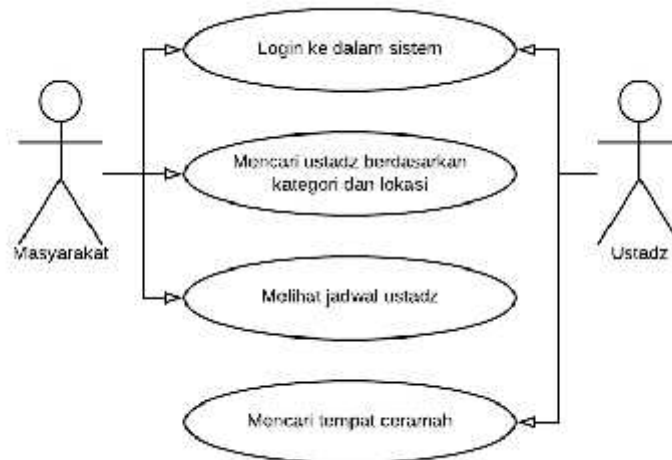
- 1) *Handphon eoperating system Android versi 2.3 (Gingerbread).*
- 2) Laptop ACER dengan spesifikasi sebagai berikut :
 - a) *Processor AMD FX-9800 RADEON R7, 12 COMPUTE CORES 4C+8G 2.70GHz.*
 - b) *RAM 8.00 GB (7.46 GB usable).*
 - c) *Harddisk 128 GB SSD + 1 TB HDD*

2. Perangkat Lunak

- 1) *Windows 10*
- 2) *Java*
- 3) *Maptile*
- 4) *Eclipse SDK 3.6.2*
- 5) *Android SDK Versi 2.3.3 (Gingerbread)*
- 6) *Android Development Tool(ADT)15.0.1*
- 7) *SQLite Manager*

F. Teknik Pengolahan dan Analisis Sistem

Analisis data terbagi menjadi dua yaitu, metode analisis kuantitatif dan metode analisis kualitatif. Analisis kuantitatif ini menggunakan data statistik dan dapat dilakukan dengan cepat. Sedangkan metode analisis kualitatif adalah dimana data yang diperoleh melalui hasil pengamatan dan hasil wawancara yang semuanya disusun dilokasi penelitian dengan melakukan analisis data dengan memperkaya informasi, mencari hubungan, membandingkan, menemukan pola atas dasar data aslinya. Dalam penelitian ini, penulis menggunakan metode analisis kualitatif. Analisis kualitatif adalah prosedur penelitian yang menghasilkan data deskriptif berupa kata-kata tertulis atau lisan dari orang-orang dan perilaku yang dapat diamati.



Gambar III.1 Analisis Sistem

G. Metode dan Perancangan Sistem

Pada penelitian ini, algoritma floyd-warshall adalah salah satu varian dari pemrograman dinamis, yaitu suatu metode yang melakukan pemecahan masalah dengan memandang solusi yang akan diperoleh sebagai suatu keputusan yang saling terkait. Artinya solusi-solusi tersebut dibentuk dari solusi yang berasal dari

tahap sebelumnya dan ada kemungkinan solusi lebih dari satu. Dalam usaha untuk mencari lintasan terpendek, algoritma floyd-warshall memulai iterasi dari titik awalnya kemudian memperpanjang lintasan dengan mengevaluasi titik demi titik hingga mencapai titik tujuan dengan jumlah bobot yang seminimum mungkin. Misalkan W_0 adalah matriks hubung graf berarah berlabel mula-mula. W^* adalah matriks hubung minimal dengan W_{ij}^* = lintasan terpendek dari titik v_i ke v_j . Algoritma floyd-warshall untuk mencari lintasan terpendek adalah sebagai berikut.

1. $W = W_0$
2. Untuk $k = 1$ hingga n , lakukan:
 Untuk $i = 1$ hingga n , lakukan:
 Untuk $j = 1$ hingga n lakukan;
 Jika $W[i, j] > W[i, k] + W[k, j]$ maka
 Tukar $W[i, j]$ dengan $W[i, k] + W[k, j]$
3. $W^* = W$

Keterangan:

W = matriks

W_0 = Matriks hubung graf mula-mula

k = iterasi 1 sampai ke- n

i = titik awal pada v_i

j = titik akhir pada v_j

W^* = hasil matriks setelah perbandingan

Haversine formula adalah persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (Bumi) berdasarkan bujur dan lintang.

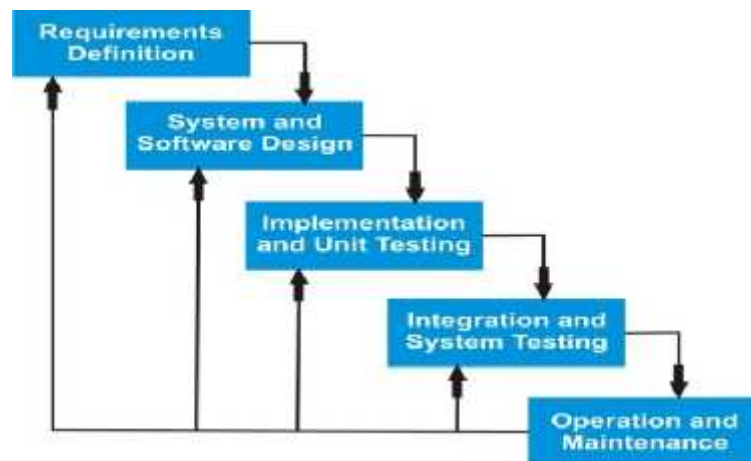
$$D = R \cdot \arccos(\sin(\text{lat1}) \cdot \sin(\text{lat2}) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \cos(\text{long2} - \text{long1}))$$

Keterangan :

R = jari-jari bumi sebesar 6371 (km)

d = jarak (km)

Dan penelitian ini, menggunakan metode perencanaan aplikasi yang digunakan adalah *Waterfall*. Model *Waterfall* adalah model klasik yang bersifat sistematis, berurutan dalam membangun *software*, dimana proses pengerjaannya bertahap dan harus menunggu tahap sebelumnya selesai dilaksanakan kemudian memulai tahap selanjutnya.



Gambar III.2. Model *Waterfall*

Berikut ini adalah detesis dari tahap model *Waterfall* :

1. *Requirement System*

Tahap dimana menentukan kebutuhan-kebutuhan bagi seluruh elemen-elemen sistem, kemudian mengalokasikan beberapa subset dari kebutuhan-kebutuhan tersebut bagi perangkat. Gambaran sistem merupakan hal yang penting pada saat perangkat lunak harus berinteraksi dengan elemen sistem lain

seperti perangkat keras, manusia dan database *Requitment System* mencakup kumpulan kebutuhan pada setiap tingkat teratas perancangan dan analisis.

2. *Analysis*

Tahap dimana kita menterjemahkan kebutuhan pengguna kedalam spesifikasi kebutuhan sistem atau *SRS (System Requirement Spesification)*. Spesifikasi kebutuhan sistem ini bersifat menangkap semua yang dibutuhkan sistem dan dapat terus diperbaharui secara *iterative* selama berjalannya proses pengembangan sistem.

3. *Design*

Tahap dimana dimulai dengan pernyataan masalah dan diakhiri dengan rincian perancangan yang dapat ditransformasikan ke sistem operasional. Transformasi ini mencakup seluruh aktivitas pengembangan perancangan.

4. *Coding*

Melakukan penghalusan rincian perancangan ke penyebaran sistem yang sesuai dengan kebutuhan pengguna. Transformasi ini juga mencakup perancangan peralatan yang digunakan, prosedur-prosedur pengoperasian, detesis orang-orang yang akan menggunakan sistem dan sebagainya.

5. *Implementation*

Implementasi yang akan digunakan meliputi proses pengaplikasian aplikasi yang sesuai dengan perancangan awal, dan membuat *prototype* untuk mengetahui kekurangan atau masalah yang dihadapi.

6. *Evaluation*

Evaluasi yang digunakan dalam pembuatan aplikasi tersebut yaitu evaluasi sistem. Evaluasi sistem dengan melakukan percobaan–percobaan kepada aplikasi tersebut dan mencari kekurangan–kekurangan yang ada serta memperbaikinya.

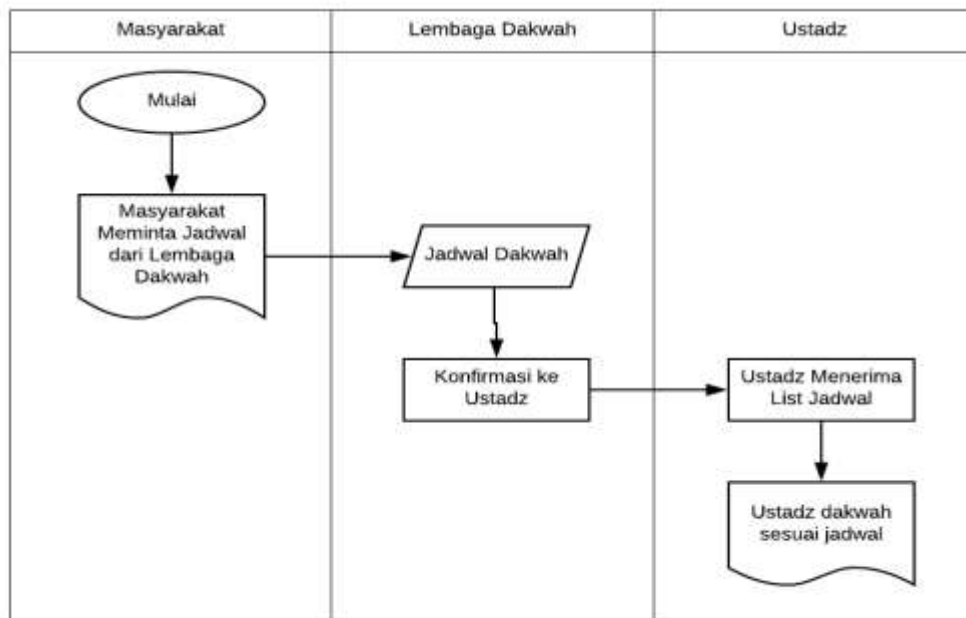
BAB IV

HASIL DAN PEMBAHASAN

A. Analisis Sistem yang Sedang Berjalan

Sistem yang sedang berjalan di masyarakat saat ini untuk mencari seorang penceramah adalah selain komunikasi langsung dengan ustadz kenalan juga memperhitungkan untuk mencari ke lembaga-lembaga dakwah yang memang mendata dan mengkoordinir langsung jadwal para ustadz dan ustadzah penceramah agar sesuai dengan tujuan pencarian.

Untuk mengetahui jadwal dakwah para penceramah tersebut, lembaga dakwah perlu mengkonfirmasi kepada ustadz dan memberikan list jadwal untuk kemudian disepakati. Seperti tahap yang terlihat pada *Flowmap* berikut :



Gambar IV.1 *flowmap* sistem yang sedang berjalan

B. Analisis Sistem yang diusulkan

Analisis sistem merupakan penguraian dari suatu sistem yang utuh ke dalam bagian-bagian komponennya untuk mengidentifikasi dan mengevaluasi permasalahan. Bagian analisis terdiri dari analisis masalah, analisis kebutuhan dan analisis kelemahan.

1. Analisis Masalah

Dalam lingkungan masyarakat luas terdapat ratusan masjid dan lokasi ceramah yang masih minim informasi alamatnya sehingga khatib atau penceramah seringkali kesulitan menemukan lokasi yang ditempati khutbah atau ceramah. Kesulitan ini menyebabkan khatib atau penceramah kadang tidak melaksanakan tugasnya di lokasi yang telah dijadwalkan. Informasi alamat lokasi yang akurat, lengkap dan tepat waktu akan memudahkan khatib dan penceramah menemukan lokasi yang telah dijadwalkan untuk khutbah atau ceramah.

Kadang juga meskipun sudah dibentuk oleh lembaga dakwah, terkadang masih sering terjadi kendala seperti miskomunikasi, untuk mencari seorang penceramah yang sesuai dengan gaya bahasa dan pola pikir masyarakat, agar makna dan isi dakwah dapat tersampaikan dengan jelas dan dapat dimengerti.

2. Analisis Kebutuhan Sistem

a. Kebutuhan Antar Muka (*Interface*)

Kebutuhan-kebutuhan antarmuka untuk pembangunan aplikasi ini yaitu sebagai berikut :

- 1) Aplikasi yang dibangun akan memiliki antarmuka yang mudah digunakan (*user friendly*).
- 2) Aplikasi ini akan menampilkan daftar nama dan profil penceramah
- 3) Aplikasi ini akan menampilkan daftar jadwal penceramah
- 4) Aplikasi ini akan menampilkan lokasi penceramah
- 5) Aplikasi ini akan menampilkan lokasi ceramah

b. Kebutuhan Data

Data yang dibutuhkan aplikasi ini yaitu sebagai berikut:

- 1) Data daftar nama-nama penceramah
- 2) Data daftar profil penceramah
- 3) Data daftar lokasi penceramah

c. Kebutuhan Fungsional

Kebutuhan fungsional merupakan penjelasan proses fungsi yang berupa penjelasan terinci setiap fungsi yang digunakan untuk menyelesaikan masalah.

Fungsi-fungsi yang dimiliki oleh aplikasi ini adalah sebagai berikut:

- 1) Menampilkan jadwal penceramah
- 2) Menampilkan notifikasi konfirmasi kesiapan penceramah

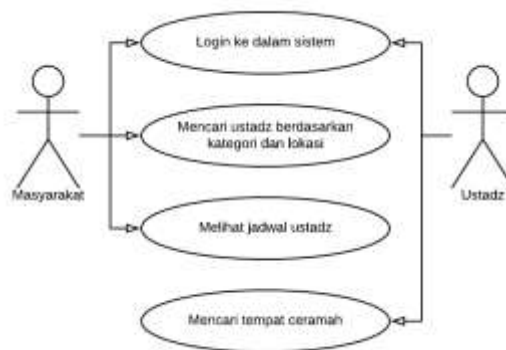
3. Analisis kelemahan

Aplikasi ini merupakan aplikasi yang berjalan di lingkungan sistem operasi android yang memberikan kemudahan pengguna untuk mencari dan mendapatkan informasi tentang penceramah. aplikasi ini memberikan info nama, jadwal, kategori dan lokasi penceramah yang akan diterima melalui notifikasi setelah melakukan order dengan membuka aplikasi secara berkala. Karena aplikasi ini belum dapat menerima notifikasi secara *real time*.

C. Perancangan Sistem

1. Use Case Diagram

Use case diagram merupakan gambaran skenario dari interaksi antara pengguna dengan sistem. *Use case diagram* menggambarkan hubungan antara pengguna dan kegiatan yang dapat dilakukannya terhadap aplikasi. Seperti ketika melihat menu utama, penjelasan item-item yang terdapat pada menu utama. Jadi *use case diagram* sangat membantu dalam memulai merancang sebuah aplikasi karena dengan hal tersebut hasil dari pembuatan aplikasi sudah dipahami.



Gambar IV.2: Use Case Diagram

2. Class Diagram

Class diagram merupakan diagram yang menggambarkan struktur system dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun system. *Class diagram* ini memberikan perincian alur sebuah aplikasi yang akan dibuat.



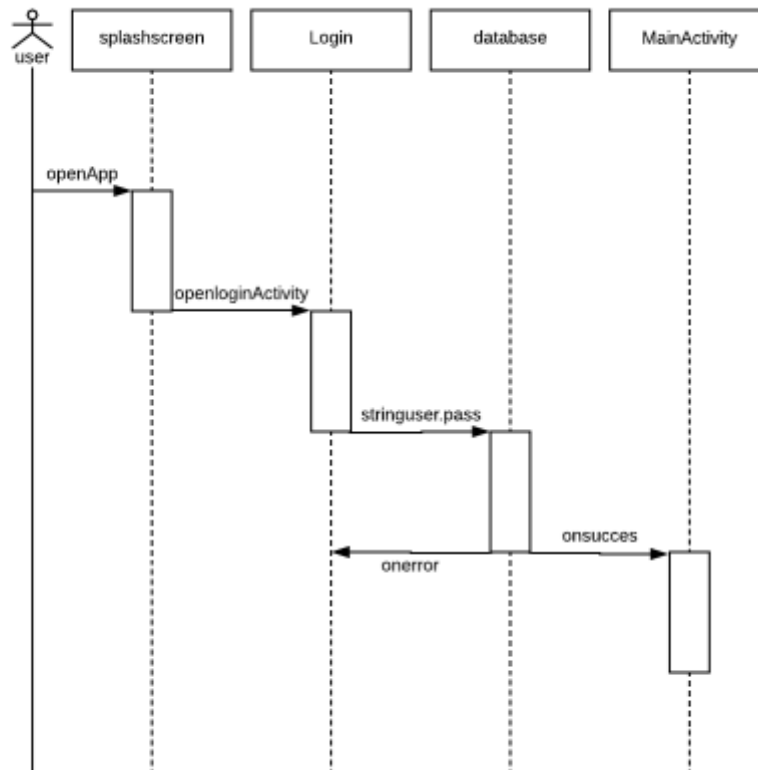
Gambar IV.3 Class Diagram

3. Sequence Diagram

Sequens Diagram menggambarkan interaksi antara obejek di dalam dan di sekitar system berupa *messaget* terhadap waktu. Pembuatan *sequence diagram* bertujuan agar perancangan aplikasi lebih mudah dan terarah. Interaksi-interaksi yang terjadi dalam aplikasi monitoring ini adalah

a. *Sequence Diagram* Melihat Tampilan login

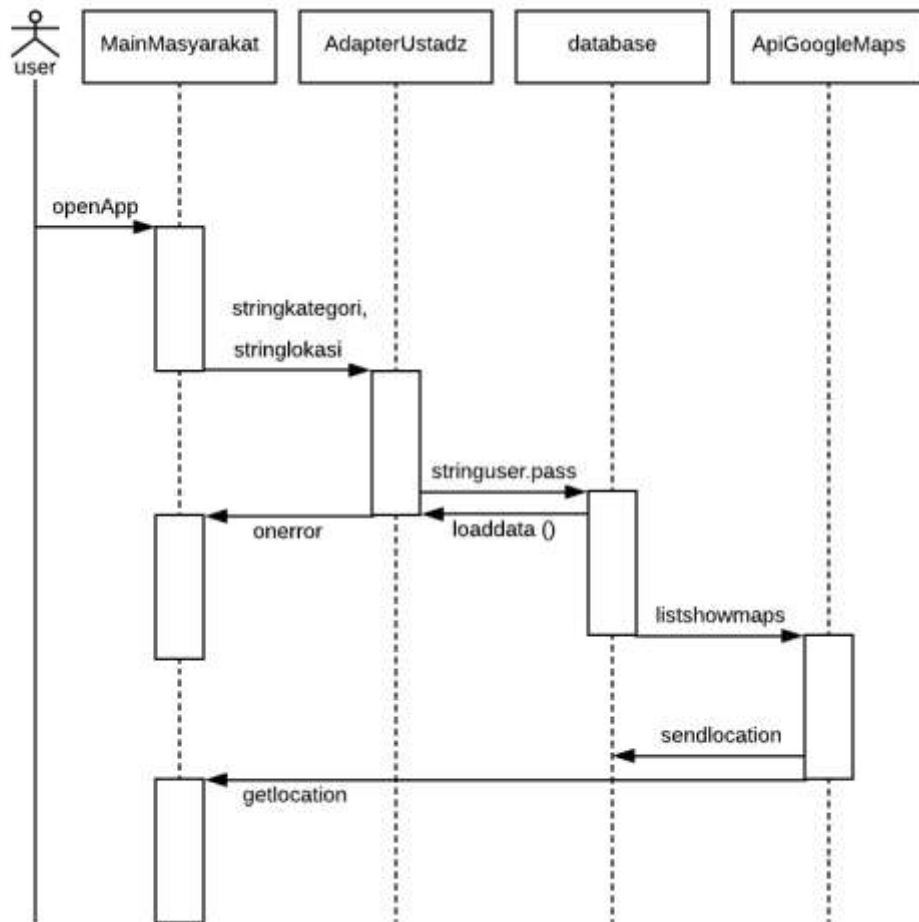
Dalam hal ini ketika pengguna menggunakan aplikasi dan setelah membuka aplikasi maka akan diarahkan oleh system ke menu utama yakni login.



Gambar IV.4 *Sequence Diagram* Menu Utama/Home

b. *Sequence Diagram* Pencarian Penceramah

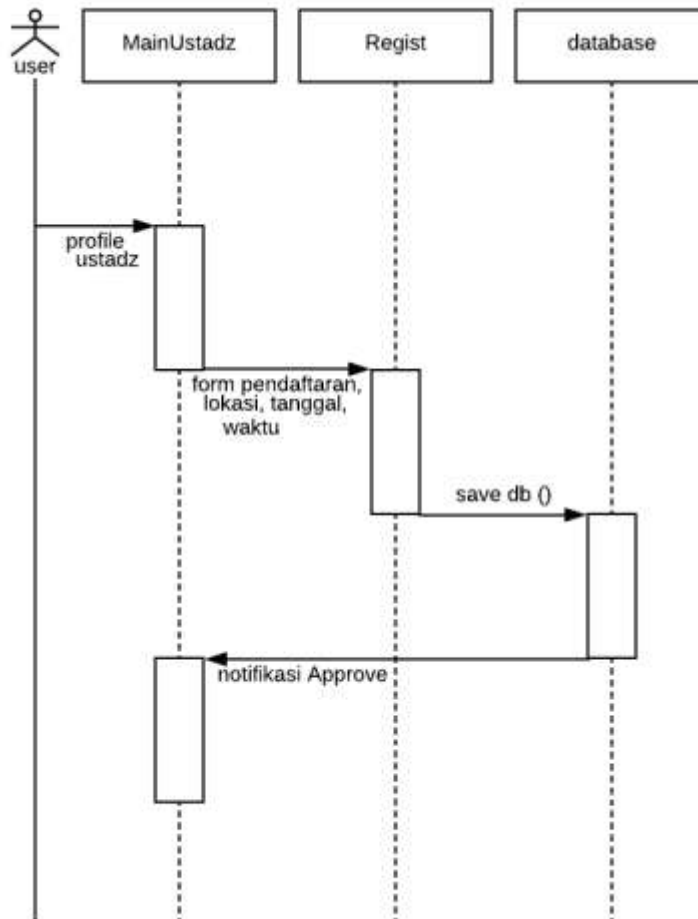
Dalam hal ini ketika pengguna menggunakan aplikasi maka akan dihadapkan dengan pencarian penceramah.



Gambar IV.5 *Sequence Diagram* Pencarian Penceramah

c. *Sequence Diagram* Pendaftaran Jadwal

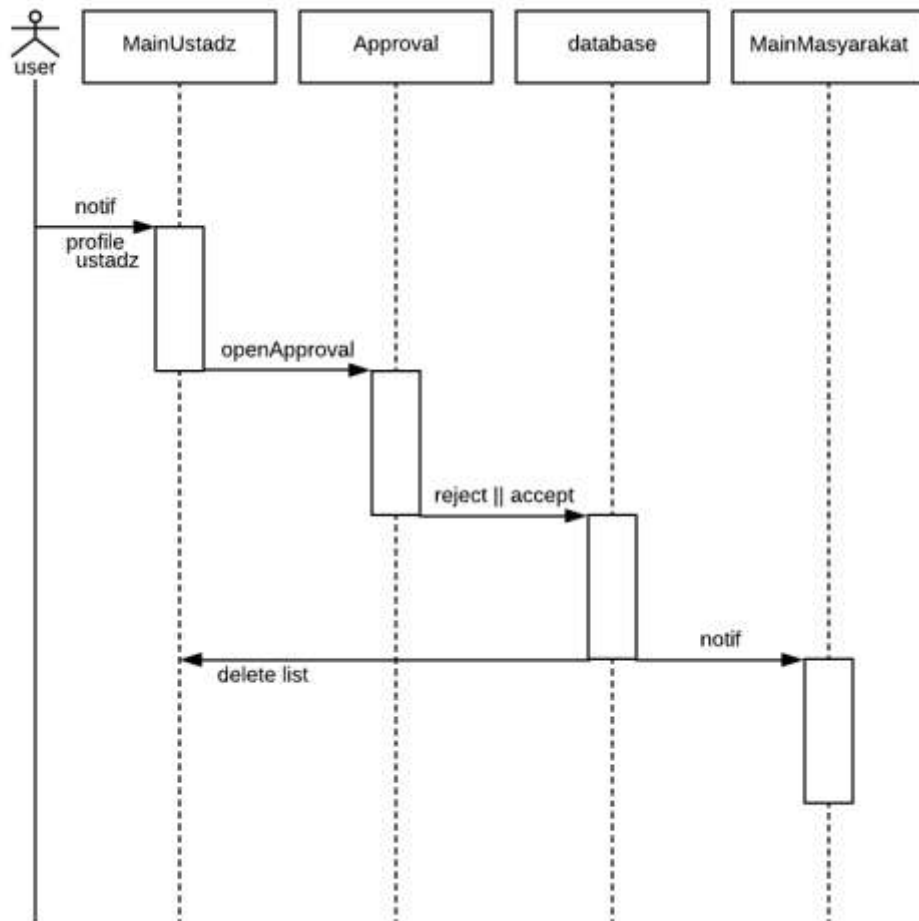
Dalam hal ini pengguna meminta jadwal ceramah kepada ustadz.



Gambar IV.6 *Sequence Diagram* Pendaftaran Jadwal

d. *Sequence Diagram* Persetujuan Atur Jadwal

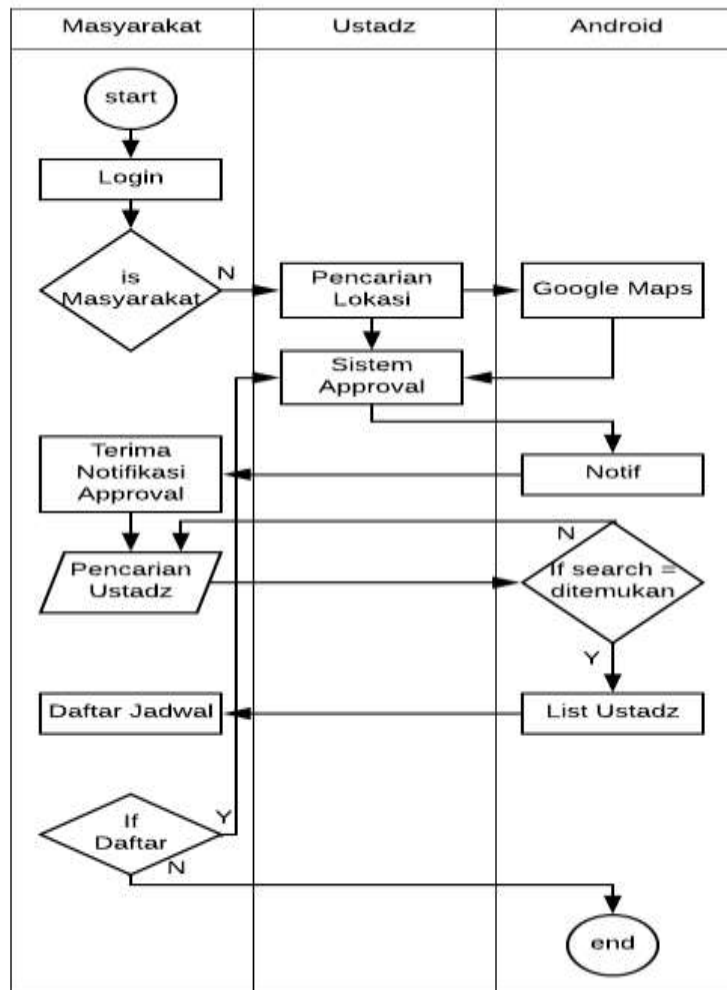
Dalam hal ini setelah *user* meminta jadwal kepada ustadz, maka notifikasi akan muncul kepada ustadz, kemudian ustadz akan menyetujui atau tidak menyetujui permintaan tersebut.



Gambar IV.7 *Sequence Diagram* Persetujuan Atur Jadwal

4. Activity Diagram

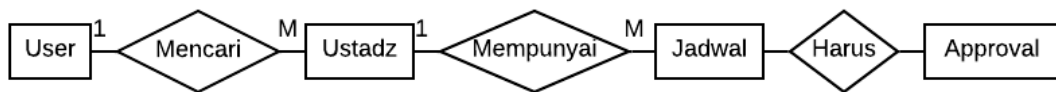
Activity diagram adalah representasi grafis dari seluruh tahapan alur kerja yang mengandung aktifitas, pilihan tindakan, perulangan, dan hasil dari aktifitas tersebut. Diagram ini dapat digunakan untuk menjelaskan proses bisnis dan alur kerja operasional secara langkah demi langkah dari komponen suatu sistem. Adapun activity diagram dari sistem ini adalah sebagai berikut :



Gambar IV.8 Activity Diagram

5. Perancangan *Entity Relationship Diagram* (ERD)

Entity Relationship Diagram (ERD) merupakan suatu model untuk menjelaskan hubungan antar data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi. Adapun *Entity Relationship Diagram*(ERD) dari sistem ini sebagai berikut :



Gambar IV.9 *Entity Relationship Diagram* (ERD)

6. Perancangan Tabel

Perancangan tabel dalam aplikasi ini yaitu untuk menampung data tentang pengguna aplikasi, nama ustadz, dan jadwal ustadz. Berikut rincian tabel dalam aplikasi ini:

a. Tabel User

Tabel IV.1 Tabel User

No	Nama_field	Tipe data	Panjang data	Keterangan
1.	id_user	Int		Primary_key
2.	user	Varchar	25	
3.	email	Varchar	35	
4.	pass	Varchar	25	
5.	level	Varchar	10	

b. Tabel Ustadz

Tabel IV.2 Tabel Ustadz

No	Nama_field	Tipe data	Panjang data	Keterangan
1.	id_ustadz	Int		Primary_key
2.	nama	Varchar	35	
3.	kategori	Varchar	20	
4.	lokasi	Varchar	35	
5.	lat	Double		
6.	Long	Double		

c. Tabel Jadwal

Tabel IV.3 Tabel Jadwal

No	Nama_field	Tipe data	Panjang data	Keterangan
1.	id	Int		Primary_key
2.	lokasi	Varchar	35	
3.	waktu	Varchar	15	
4.	tanggal	Date		
5.	lat	Double		
6.	long	Double		

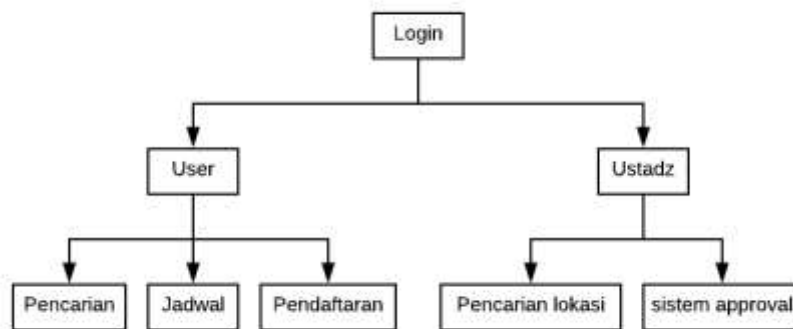
d. Tabel Approval

Tabel IV.4 Tabel Approval

No	Nama_field	Tipe data	Panjang data	Keterangan
1.	id	Int		Primary_key
2.	id_user	Int		
3.	id_ustadz	Int		
4.	lokasi	Varchar	35	
5.	waktu	Varchar	15	
6.	tanggal	Date		

7. Struktur Navigasi

Aplikasi monitoring Berbasis Mobile ini menggunakan Struktur Navigasi *Composite* (campuran), dimana ketika membuka aplikasi akan menampilkan berbagai button dan beberapa button bercabang pada aplikasi.



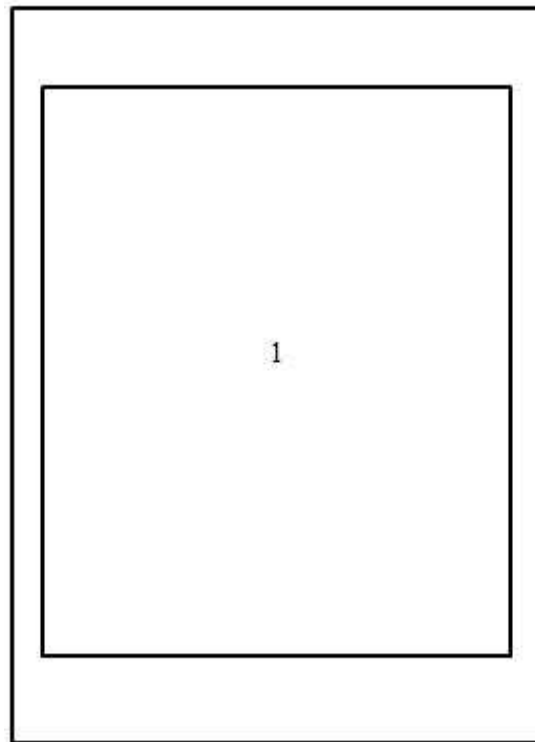
Gambar IV.10 Struktur Navigasi

Dari Struktur Navigasi ini perpindahan antar fitur yang tersedia dapat dilakukan melalui menu-menu aplikasi. Struktur Navigasi adalah solusi cerdas untuk memudahkan mendesain layoutnya.

8. Perancangan Antarmuka (*Interface*)

Perancangan antarmuka (*interface*) merupakan bagian penting dalam perancangan aplikasi, karena berhubungan dengan tampilan dan interaksi pengguna dengan aplikasi. Adapun perancangan pada aplikasi ini yaitu sebagai berikut:

a. Perancangan Antarmuka *Splash Screen*



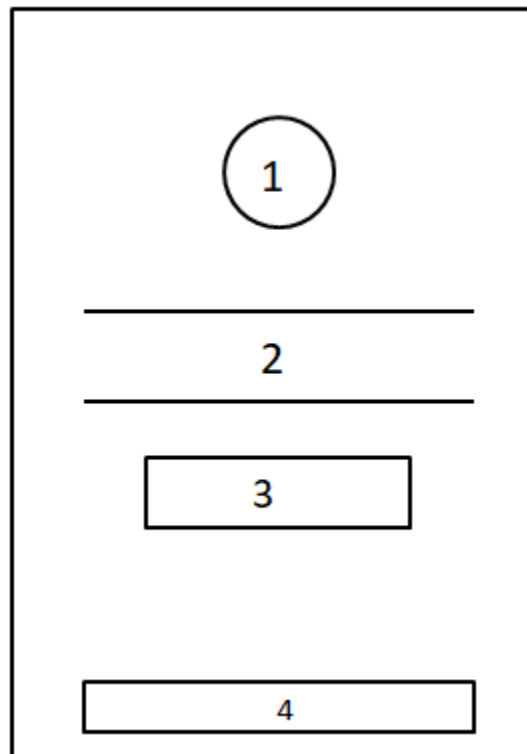
Gambar IV.11 Desain Antarmuka *Splashscreen*

Keterangan Gambar:

1) Gambar

Akan dibuat dengan berisikan gambar *splashscreen*

b. Perancangan Antarmuka *login*



Gambar IV.12 Desain antarmuka *login*

Keterangan Gambar:

1) Gambar

Akan dibuat untuk menampilkan logo aplikasi

2) *Edit text*

Akan dibuat untuk memasukkan *username* dan *password*

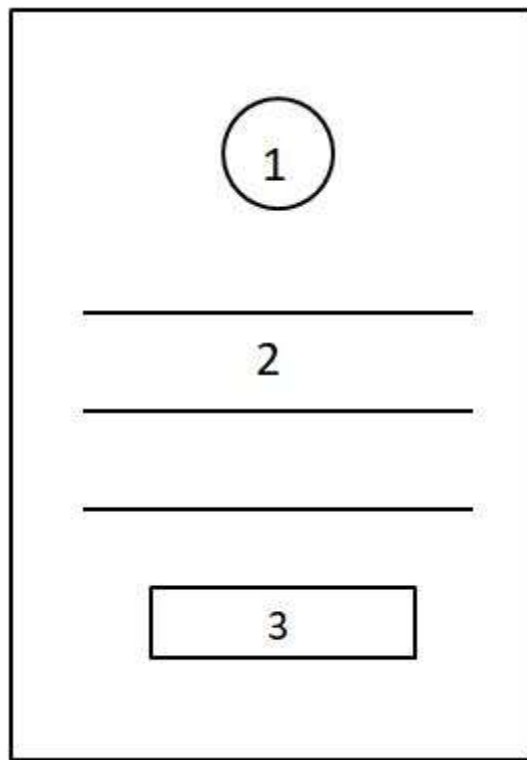
3) *Button*

Akan dibuat untuk masuk ke menu utama

4) *Text View*

Akan dibuat untuk pendaftaran *user*

c. Perancangan Antarmuka Pendaftaran *User*



Gambar IV.13 Desain Antarmuka Pendaftaran *User*

Keterangan Gambar:

1. Gambar

Akan dibuat untuk menampilkan logo aplikasi

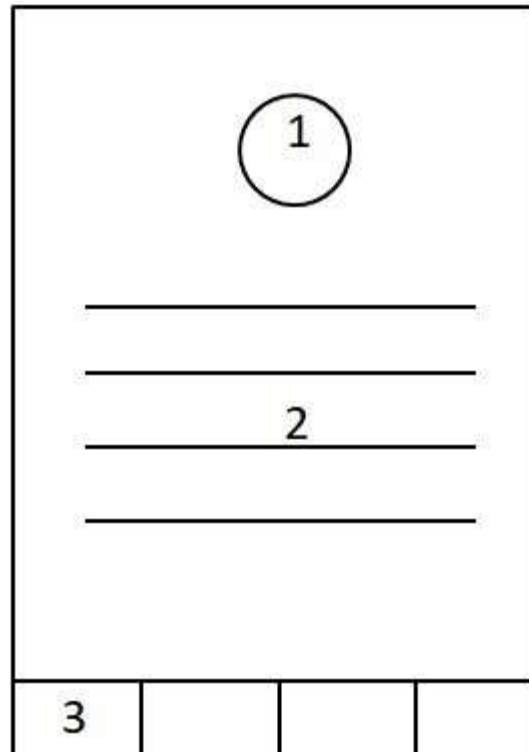
2. *Edit Text*

Akan dibuat untuk *form* pendaftaran *user*

3. *Button*

Akan dibuat untuk melakukan pendaftaran *user*

d. Perancangan Antarmuka Menu Utama



Gambar IV.14 Desain Antarmuka Menu Utama

Keterangan Gambar:

1) *Gambar*

Akan dibuat untuk menampilkan logo aplikasi

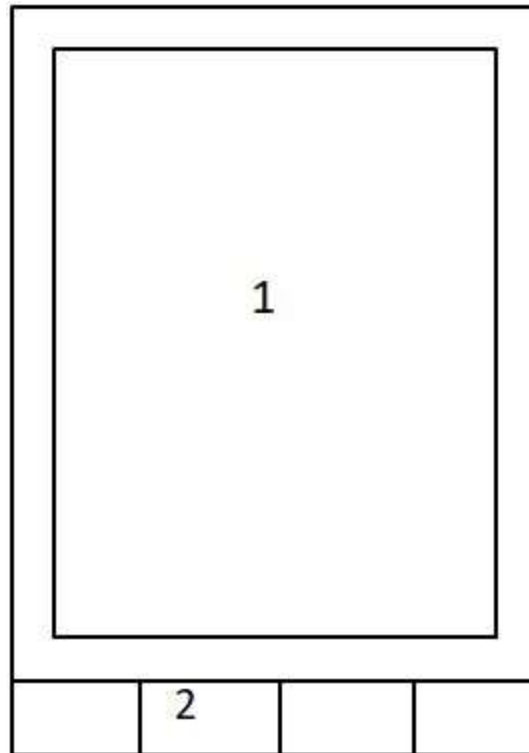
2) *Text View*

Akan dibuat untuk menampilkan informasi aplikasi

3) *Button*

Akan dibuat untuk pemilihan menu utama

e. Perancangan Antarmuka Menu Lokasi



Gambar IV.15 Desain Antarmuka Menu Lokasi

Keterangan Gambar:

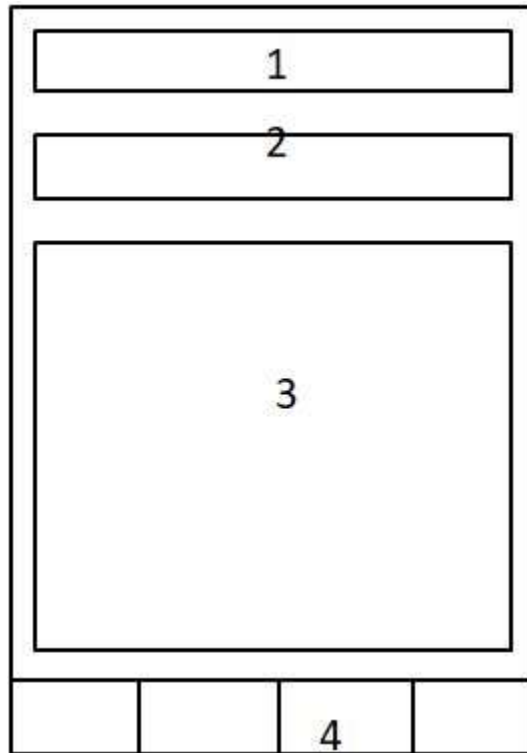
1) *Map*

Akan dibuat untuk menampilkan lokasi ustadz

2) *Button*

Akan dibuat untuk pemilihan menu lokasi

f. Perancangan Antarmuka Menu Pencarian



Gambar IV.16 Desain Antarmuka Menu Pencarian

Keterangan Gambar:

1) *Edit Text*

Akan dibuat untuk mencari nama ustadz

2) *Spinner*

Akan dibuat untuk mencari kategori ustadz

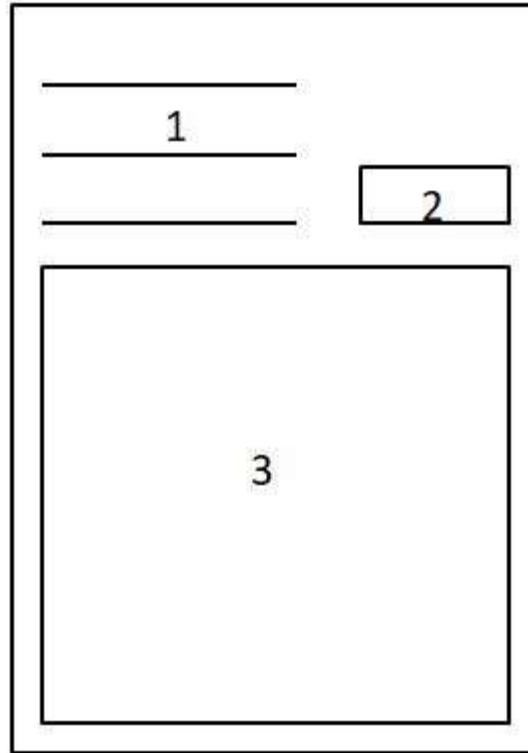
3) *List View*

Akan dibuat untuk menampilkan daftar ustadz

4) *Button*

Akan dibuat untuk pemilihan menu pencarian

g. Perancangan Antarmuka Jadwal Ustadz



Gambar IV.17 Desain Antarmuka Jadwal Ustadz

Keterangan Gambar :

1) *Text View*

Akan dibuat untuk profil ustadz

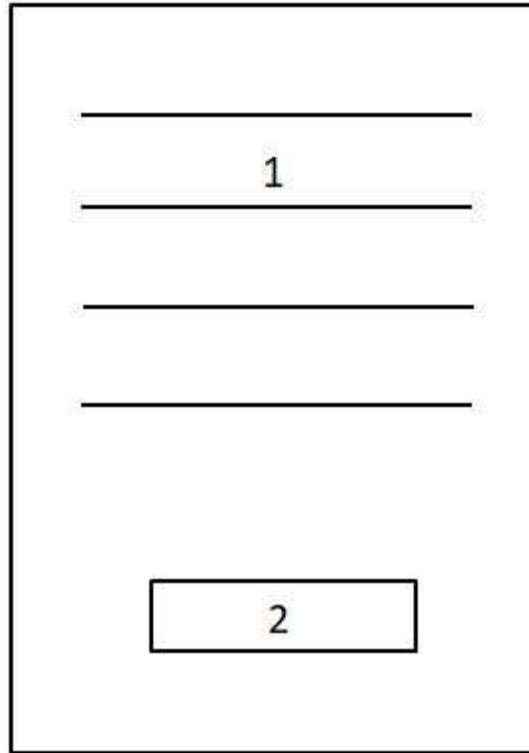
2) *Button*

Akan dibuat untuk melakukan permintaan jadwal ustadz

3) *List View*

Akan dibuat untuk menampilkan *list* jadwal ustadz

h. Perancangan Antarmuka Permintaan Jadwal Ceramah



Gambar IV.18 Desain Antarmuka Permintaan Jadwal Ceramah

Keterangan gambar :

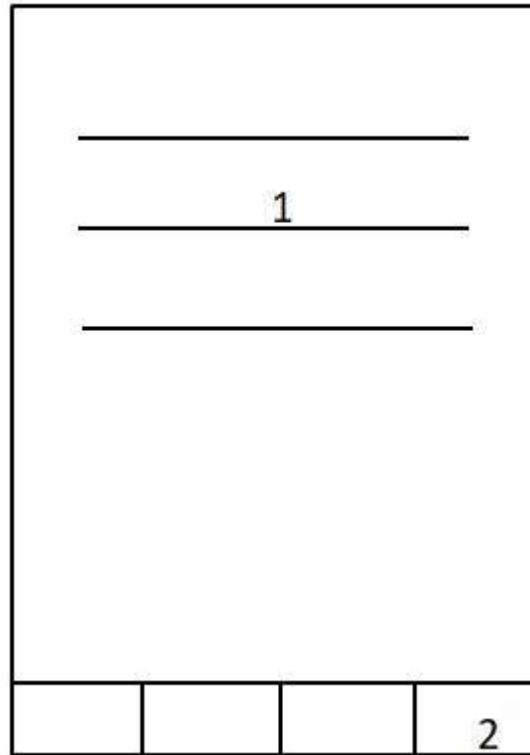
1) *Edit Text*

Akan dibuat untuk *form* permintaan jadwal ceramah

2) *Button*

Akan dibuat untuk melakukan permintaan ceramah

i. Perancangan Antarmuka Menu Akun *User*



Gambar IV.19 Desain Antarmuka Menu Akun *User*

Keterangan gambar :

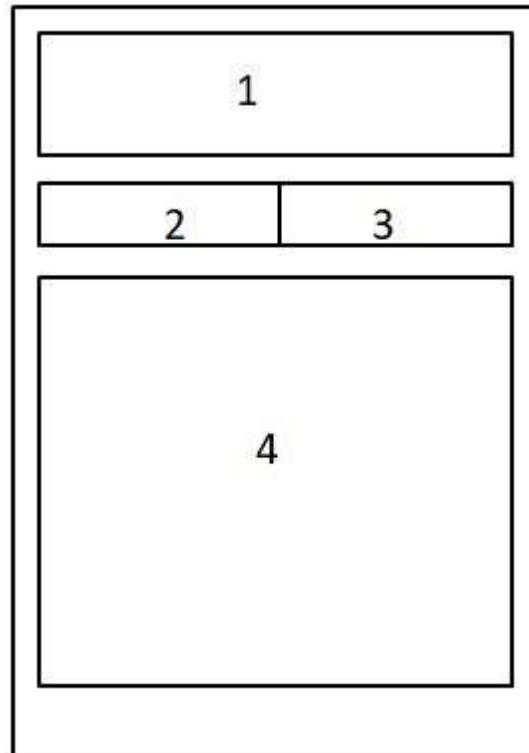
1) *Text View*

Akan dibuat untuk menampilkan profil *user*

2) *Button*

Akan dibuat untuk pemilihan menu akun

j. Perancangan Antarmuka Menu Akun Ustadz



Gambar IV.20 Desain Antarmuka Menu Akun Ustadz

Keterangan gambar :

1) *Text View*

Akan dibuat untuk menampilkan profil ustadz

2) *Button*

Akan dibuat untuk pemilihan menampilkan permintaan jadwal

3) *Button*

Akan dibuat untuk pemilihan menampilkan jadwal ceramah

4) *List View*

Akan dibuat untuk menampilkan *list* jadwal penceramah

D. Implementasi Sistem

1. Interface

a. Antarmuka *Splash Screen*

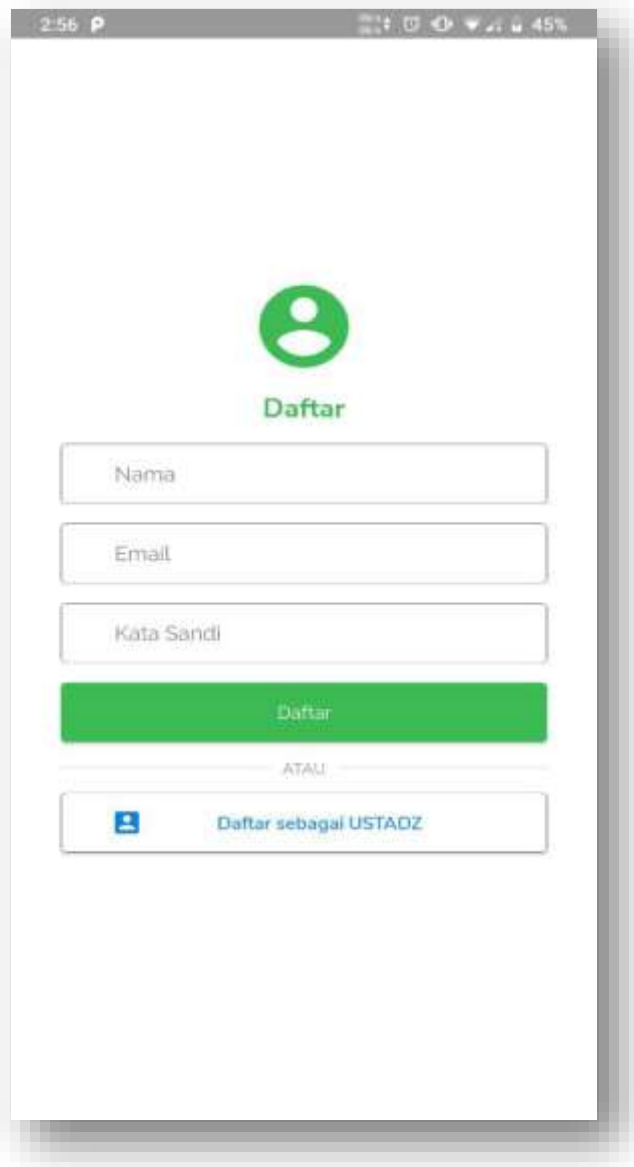
Antarmuka *Splash Screen* akan menampilkan gambar *splash screen* aplikasi selama 1 detik.



Gambar IV.21 *Splash Screen*

b. Antarmuka Menu *Login*

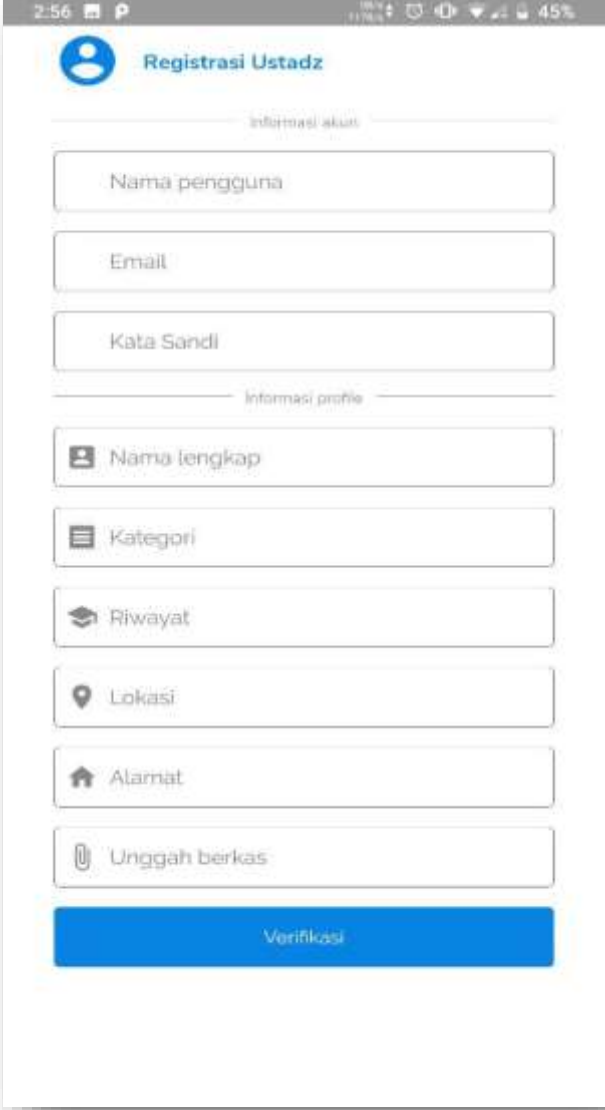
Antarmuka menu *login* berupa menu yang akan menampilkan halaman untuk verifikasi akses. Apakah akan *login* sebagai *user* atau *ustadz*.



Gambar IV.22 *Login*

c. Antarmuka Pendaftaran *User*

Antarmuka pendaftaran *user* akan menampilkan *form* pengisian data *user* seperti nama *user*, email, dan *password*.

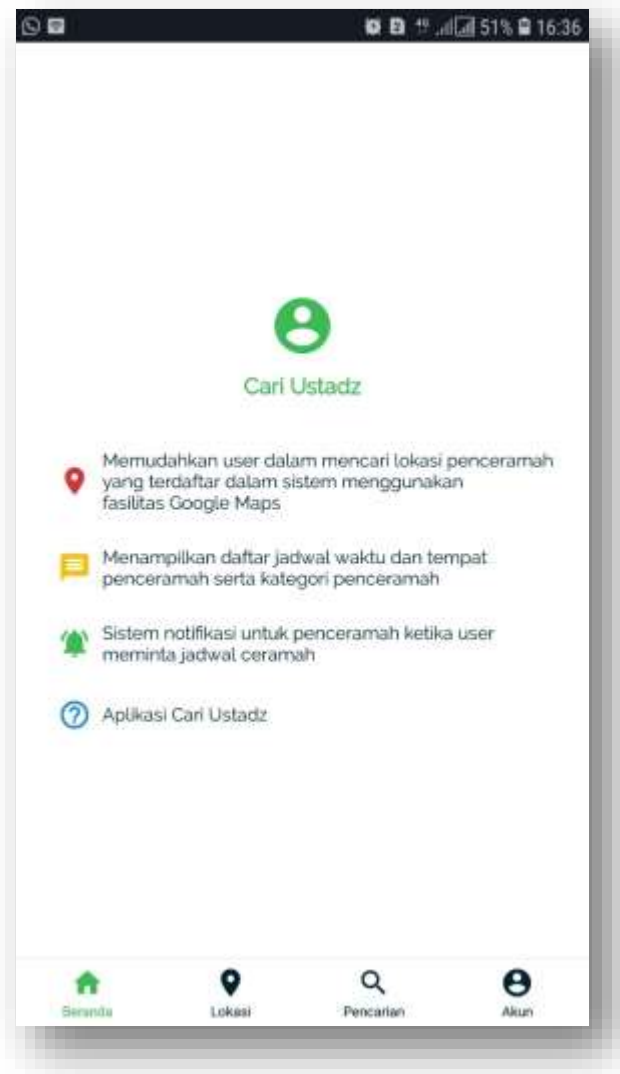


The image shows a mobile application interface for user registration. At the top, the status bar displays the time 2:56, signal strength, Wi-Fi, and 45% battery. The app title is 'Registrasi Ustadz' with a blue profile icon. The form is organized into two sections: 'Informasi akun' and 'Informasi profil'. The 'Informasi akun' section contains three input fields: 'Nama pengguna', 'Email', and 'Kata Sandi'. The 'Informasi profil' section contains six input fields: 'Nama lengkap' (with a person icon), 'Kategori' (with a list icon), 'Riwayat' (with a graduation cap icon), 'Lokasi' (with a location pin icon), 'Alamat' (with a house icon), and 'Unggah berkas' (with a paperclip icon). A prominent blue button labeled 'Verifikasi' is positioned at the bottom of the form.

Gambar IV.23 Pendaftaran *User*

d. Antarmuka Beranda

Antarmuka beranda merupakan menu yang berisi tentang informasi aplikasi secara singkat.

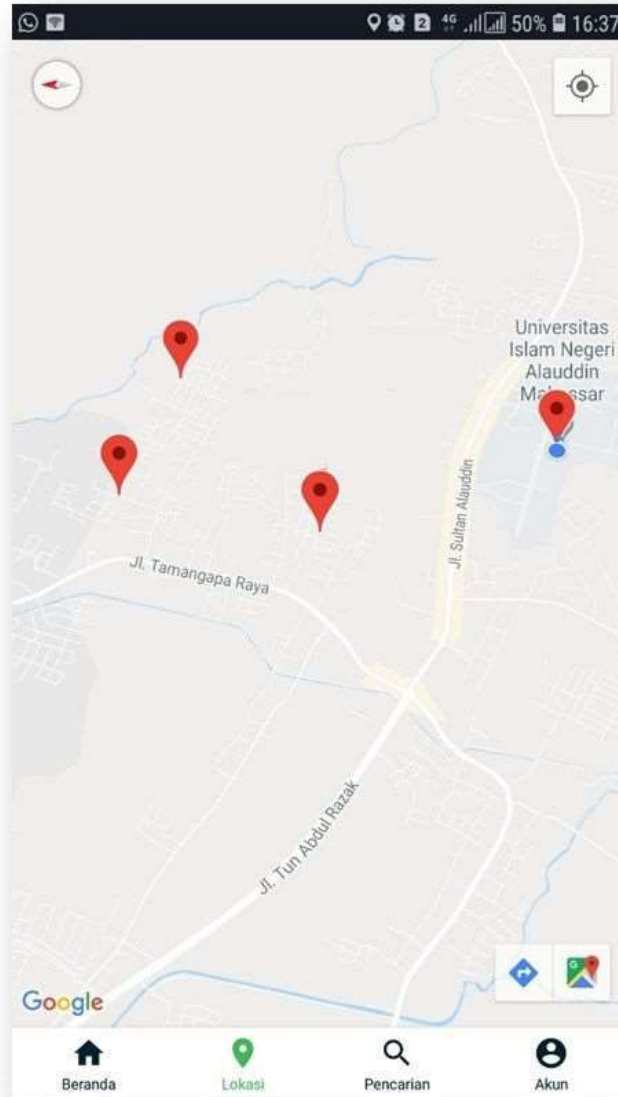




Gambar IV.24 Menu Utama

e. Antarmuka Menu Lokasi

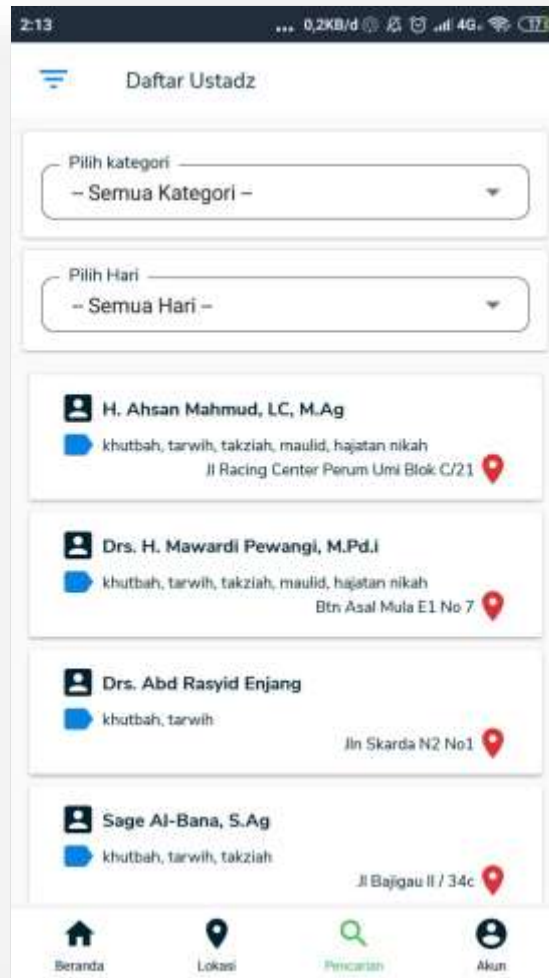
Antarmuka menu lokasi berupa *Maps* yang akan menampilkan lokasi para penceramah.



Gambar IV.25 Antarmuka Menu Lokasi

f. Antarmuka Menu Pencarian

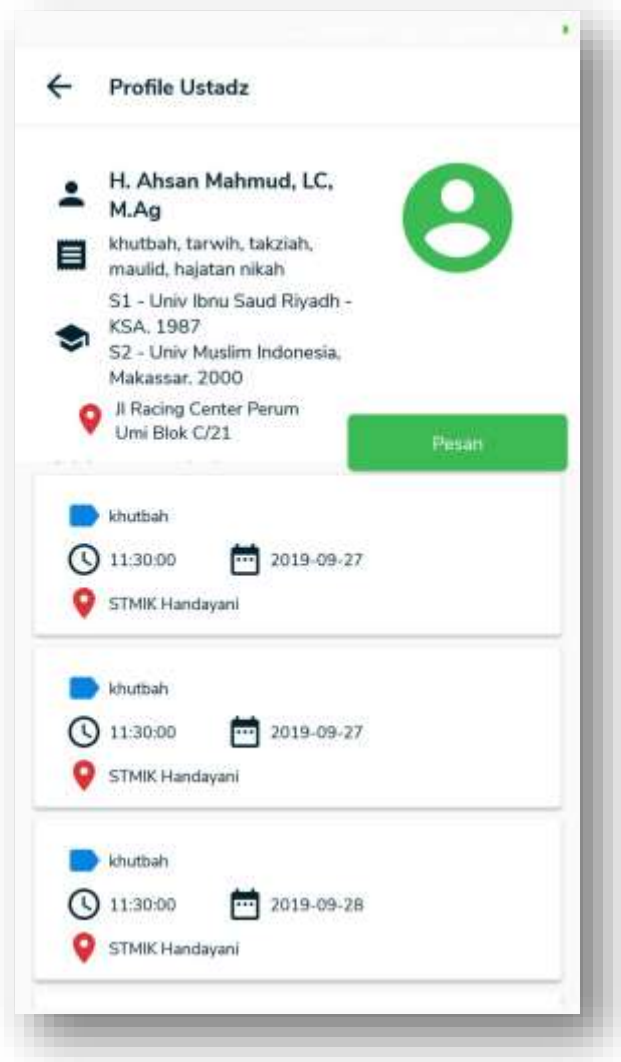
Antarmuka menu pencarian berisi menu inputan dan List nama penceramah berdasarkan nama dan kategori.



Gambar IV.26 Antarmuka Menu Pencarian

g. Antarmuka Menu Jadwal Ustadz

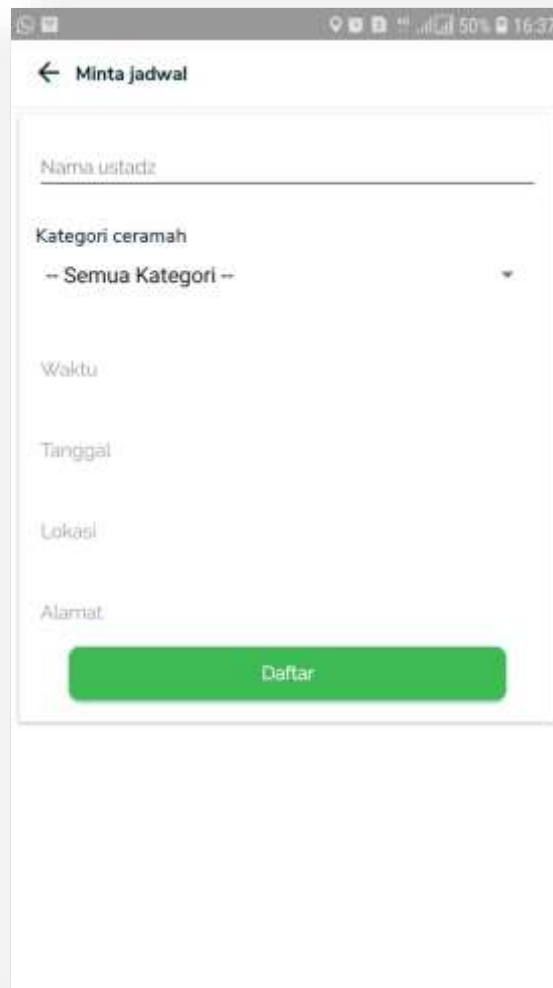
Antarmuka menu jadwal ustadz merupakan menu yang akan menampilkan profil dan jadwal ustadz.



Gambar IV.27 Antarmuka Menu Jadwal Ustadz

h. Antarmuka Permintaan Jadwal Ceramah

Antarmuka permintaan jadwal ceramah merupakan *form* pengisian permintaan jadwal ceramah seperti kategori ceramah, waktu, tanggal, lokasi dan alamat.



The image shows a mobile application interface for requesting a lecture schedule. The screen is titled "Minta jadwal" and contains several input fields: "Nama ustadz", "Kategori ceramah" (a dropdown menu currently showing "-- Semua Kategori --"), "Waktu", "Tanggal", "Lokasi", and "Alamat". A green "Daftar" button is located at the bottom of the form.

Gambar IV.28 Antarmuka Permintaan Jadwal Ceramah

i. Antarmuka Menu Akun *User*

Antarmuka menu akun *user* akan menampilkan profil *user* dan riwayat permintaan jadwal ceramah.



Gambar IV.29 Antarmuka Menu Akun *User*

j. Antarmuka Menu Akun Ustadz

Antarmuka menu akun ustadz merupakan menu yang akan menampilkan profil ustadz, permintaan jadwal ceramah, dan jadwal ceramah ustadz.



Gambar IV.30 Antarmuka Menu Akun Ustadz

E. Hasil Pengujian

Pengujian sistem merupakan proses pengekseskuan sistem perangkat lunak untuk menentukan apakah sistem tersebut cocok dengan spesifikasi sistem dan berjalan di lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian bug, ketidaksempurnaan program, kesalahan pada program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak.

Pengujian dilakukan dengan menguji setiap proses dan kemudian kesalahan yang terjadi untuk setiap proses. Adapun pengujian sistem yang digunakan adalah *Black box*. Pengujian *Black box* yaitu menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program. Pengujian dimaksudkan untuk mengetahui apakah fungsi-fungsi, masukan dan keluaran dari perangkat lunak sesuai dengan spesifikasi yang dibutuhkan.

1. Prosedur Pengujian

Persiapan yang dibutuhkan dalam melakukan pengujian ini adalah sebagai berikut:

- a. Menyiapkan sebuah *Smartphone* dengan sistem operasi Android
- b. Menginstall aplikasi Cari Ustadz pada perangkat tersebut
- c. Melakukan proses pengujian
- d. Mencatat hasil pengujian

2. Hasil Pengujian Metode

Metode ini menggunakan Haversine formula adalah persamaan yang penting pada navigasi, memberikan jarak lingkaran besar antara dua titik pada permukaan bola (Bumi) berdasarkan bujur dan lintang.

$$D = \text{Acos}(\sin(\text{lat1}) \cdot \sin(\text{lat2}) + \cos(\text{lat1}) \cdot \cos(\text{lat2}) \cdot \cos(\text{long2} - \text{long1})) \cdot R$$

Keterangan :

R = jari-jari bumi sebesar 6371(km)

D = jarak (km)

Tabel IV. 5. Pengujian Metode Haversine Formula

Nama Ustadz	Latitude	Logitude	Radian Lat	Radian Long	Jarak(Km)
H. Ahsan Mahmud, LC, M.Ag	0.5148461	119.45006	0.008986	2.0847968	1.424377716
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Drs. H. Mawardi Pewangi, M.Pd.i	0.5134804	119.479051	0.008962	2.0853028	2.015017309
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Drs. Abd Rasyid Enjang	0.5176851	119.44394	0.009035	2.0846912	2.008895642
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Sage Al-Bana, S.Ag	0.517404	119.412089	0.009031	2.0841341	5.552009062
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Drs. Aman Setia R	0.5106516	119.538046	0.008912	2.0863324	8.515992786
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Drs. H. Muh Thahir Taro	0.596554	119.461805	0.010411	2.0850018	8.556274345
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Drs. Amirullah Amri, M.A	0.514302	119.433708	0.008976	2.0845114	3.197504203
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Dr. H. Suf Kasman , M.Ag	0.5154252	119.48717	0.008995	2.0854445	2.839560633
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	
Drs. Muh Basri Wallang	0.5169302	119.497159	0.009022	2.0856188	3.922597113
Rifaldy Ramadhan L. (Pemesan)	0.5190919	119.461979	0.009068	2.0850048	

Pembahasan rumus haversine jarak ustadz dengan pemesan dari table diatas maka hasil perhitungan sebagai berikut:

1. Jarak H. Ahsan Mahmud, LC, M.Ag dengan pemesan

$$\text{Acos}(\text{Sin}(0.008986)*\text{Sin}(0.009068)+\text{Cos}(0.008986)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0847968))*6371= 1,42 \text{ Km}$$

2. Jarak Drs. H. Mawardi Pewangi, M.Pd.i dengan pemesan

$$\text{Acos}(\text{Sin}(0.008962)*\text{Sin}(0.009068)+\text{Cos}(0.008962)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0853028))*6371= 2,01 \text{ Km}$$

3. Jarak Drs. Abd Rasyid Enjang dengan pemesan

$$\text{Acos}(\text{Sin}(0.009035)*\text{Sin}(0.009068)+\text{Cos}(0.009035)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0846912))*6371= 2 \text{ Km}$$

4. Jarak Sage Al-Bana, S.Ag dengan pemesan

$$\text{Acos}(\text{Sin}(0.009031)*\text{Sin}(0.009068)+\text{Cos}(0.009031)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0841341))*6371= 5,55 \text{ Km}$$

5. Jarak Drs. Aman Setia R dengan pemesan

$$\text{Acos}(\text{Sin}(0.008912)*\text{Sin}(0.009068)+\text{Cos}(0.008912)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0863324))*6371= 8,51 \text{ Km}$$

6. Jarak Drs. H. Muh Thahir Taro dengan pemesan

$$\text{Acos}(\text{Sin}(0.010411)*\text{Sin}(0.009068)+\text{Cos}(0.010411)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0850018))*6371= 8,55 \text{ Km}$$

7. Jarak Drs. Amirullah Amri, M.A dengan pemesan

$$\text{Acos}(\text{Sin}(0.008976)*\text{Sin}(0.009068)+\text{Cos}(0.008976)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0845114))*6371= 3,19 \text{ Km}$$

8. Jarak Dr. H. Suf Kasman , M.Agdengan pemesan

$$\text{Acos}(\text{Sin}(0.008995)*\text{Sin}(0.009068)+\text{Cos}(0.008995)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0854445))*6371= 2,83 \text{ Km}$$

9. Jarak Drs. Muh Basri Wallang dengan pemesan

$$\text{Acos}(\text{Sin}(0.009022)*\text{Sin}(0.009068)+\text{Cos}(0.009022)*\text{Cos}(0.009068)*\text{Cos}(2.0850048-2.0856188))*6371= 3,92 \text{ Km}$$

3. Hasil Pengujian Sistem

a. Pengujian Form Login

Tabel pengujian Form Login digunakan untuk mengetahui apakah pengguna dapat memverifikasi data login.

Tabel IV. 6. Pengujian Form Login

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan ID User/Ustadz dan Password	Login data	berhasil Login	[√] Diterima [] Ditolak

b. Pengujian Menu Daftar User

Tabel Pengujian Menu Daftar User digunakan pendaftaran user baru.

Tabel IV. 7. Pengujian Menu Daftar User

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Menginput data user	Menyimpan data user berdasarkan Inputan	Data berhasil disimpan	[√] Diterima [] Ditolak

c. Pengujian Menu Lokasi

Tabel Menu Lokasi digunakan untuk menampilkan lokasi para penceramah

Tabel IV. 8. Pengujian Menu Lokasi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih Lokasi penceramah	Menampilkan Lokasi penceramah	Lokasi berhasil ditampilkan	[√] Diterima [] Ditolak

d. Pengujian Menu Pencarian

Tabel pengujian Menu Pencarian digunakan untuk menampilkan daftar nama penceramah berdasarkan nama dan kategori.

Tabel IV. 9. Pengujian Menu Pencarian

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan Inputan	Menampilkan daftar nama penceramah	Daftar nama penceramah berhasil ditampilkan	[√] Diterima [] Ditolak
Memilih Kategori	Menampilkan List Kategori	List kategori Berhasil ditampilkan	[√] Diterima [] Ditolak

e. Pengujian Menu Jadwal Ustadz

Tabel Pengujian Menu Jadwal Ustadz digunakan mengetahui Profil dan daftar riwayat jadwal ceramah ustadz.

Tabel IV. 10. Pengujian Menu Jadwal Ustadz

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih Nama Ustadz	Menampilkan profil dan Jadwal	Data berhasil disimpan	[√] Diterima [] Ditolak

f. Pengujian Menu Permintaan Jadwal Ceramah

Tabel Pengujian Menu Permintaan Jadwal Ceramah digunakan untuk menginput permintaan jadwal seperti kategori ceramah, waktu, lokasi dan alamat.

Tabel IV. 11. Pengujian Menu Permintaan Jadwal Ceramah

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi Form Permintaan	Menampilkan Data Form sesuai permintaan	Data berhasil disimpan	[√] Diterima [] Ditolak

g. Pengujian Menu Akun user

Tabel Pengujian Menu Akun User digunakan untuk menampilkan profil user dan Riwaya permintaan jadwal.

Tabel IV. 12. Pengujian Menu Akun User

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Pilih menu user	Menampilkan Data Profil dan List Permintaan jadwal	Data berhasil ditampilkan	[√] Diterima [] Ditolak

h. Pengujian Menu Akun Ustadz

Tabel Pengujian Menu Akun Ustadz digunakan untuk menampilkan profil ustadz, jadwal ceramah dan permintaan jadwal

Tabel IV. 13. Pengujian Menu Akun Ustadz

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Pilih Akun Ustadz	Menampilkan Data Profil, jadwal ceramah dan permintaan jadwal	Data berhasil ditampilkan	[√] Diterima [] Ditolak

BAB V

PENUTUP

A. Kesimpulan

Sistem Pencarian Penceramah ini dibuat berdasarkan kebutuhan masyarakat Makassar untuk mempermudah pencarian penceramah berdasarkan jadwal dan lokasinya. Sistem ini digunakan dua arah langsung oleh penceramah/ustadz dan yang mencari ustadz. Dimana pencari menggunakan sistem untuk mencari penceramah dan sistem mengarahkan kepada ustadz yang sesuai dengan kebutuhan pencari berdasarkan nama, alamat, kategori dan jadwal penceramah tersebut. Kemudian mengisi permintaan jadwal untuk di konfirmasi kepada ustadz. Sistem ini berjalan di sistem operasi Android. Adapun rumusan masalah yang terselesaikan dari penelitian sebagai berikut :

1. Aplikasi yang berjalan pada system operasi berbasis android.
2. Aplikasi pencarian lokasi rumah penceramah dengan memanfaatkan *Global Positioning System (GPS)* dan *Google Map API*
3. Menggunakan metode *haversine formula* dalam menentukan jarak terdekat pada aplikasi
4. Menerapkan radius dalam menampilkan jarak terdekat pada aplikasi

B. Saran

Aplikasi pencarian penceramah ini masih sangat jauh dari sempurna. Masih memiliki beberapa kelemahan diantaranya belum dapat menerima notifikasi secara real time. Masih harus membuka aplikasi terlebih dahulu untuk mendapatkan notifikasi. Diharapkan kedepannya untuk keperluan pengembangan sistem, kiranya bisa di perhatikan lagi penambahan beberapa fitur untuk memaksimalkan manfaat aplikasi.

DAFTAR PUSTAKA

- Akhyar, “*Dakwah Dan Perubahan Sosial*”. *Jurnal*. Surabaya, 2014.
- Al-Mahalliy, As-Suyuthi. *Tafsir Jalalain*, Sinar Baru Bandung, 1990.
- Departemen Agama RI, *Al-Qur'an Al-Karim Dan Terjemahannya*. Bandung: PT. Sygma Examedia Arkanleema, 2007.
- Dirman, “*Sistem Terintegrasi Manajemen Penjadwalan Khatib Pada Masjid-Masjid Yang Terdaftar Pada Lembaga Dakwah*”. *Tesis*. Makassar: Fakultas Sains Dan Teknologi UIN Alauddin Makassar. 2017.
- Gunita Mustika Hati, “*Aplikasi Penanda Lokasi Peta Digital Berbasis Mobile GIS Pada Smartphone Android*”. *Tesis*. Semarang: Teknik Geodesi Universitas Diponegoro. 2013.
- Hakim, Luqman. “*Sistem Informasi Penjadwalan Kunjungan Perpustakaan Keliling Berbasis Web dengan SMS Gateway pada Kantor Perpustakaan dan Arsip Daerah Kabupaten Kudus*”. *Tesis*. Kudus: Fakultas Teknik Universitas Muria. 2014.
- Alter, *Sistem dan teknologi informasi merupakan fokus utama*, <https://mikatorin.wordpress.com/2015/09/22/sistem-dan-teknologi-informasi/>, 2019.
- Lutfi Parmuarip, Wildan Muslim, Yeni Mulyani, “*Alasan Penggunaan Smartphone di Kalangan Mahasiswa Politeknik Negeri Bandung*”. *Tesis*.

Bandung: Jurusan Teknik Komputer dan Informatika Politeknik Negeri Bandung. 2012.

Baker, Biegel, Conway, et al, Vollman, Onikom. “Pengertian Penjadwalan (Scheduling) dalam Proses Produksi”, <https://ilmumanajemenindustri.com/pengertian-penjadwalan-scheduling-dalam-proses-produksi>. 2018.

Dhanta, “Aplikasi”, https://elib.unikom.ac.id/files/disk1/631/jbptunikompp-gdl-elsaramdha-31533-8-unikom_e-i.pdf. 2009.

Anisyah, “Aplikasi”, https://elib.unikom.ac.id/files/disk1/624/jbptunikompp-gdl-hilmanmaul-31181-10-unikom_h-i.pdf. 2009.

Safaat, “Pemrograman aplikasi mobile smartphone dan tablet PC berbasis android”, https://elib.unikom.ac.id/files/disk1/707/jbptunikompp-gdl-mastoniher-35304-7-unikom_m-i.pdf, 2012.

Dirman, “Sistem Terintegrasi Manajemen Penjadwalan Khatib pada Masjid-masjid yang Terdaftar pada Lembaga Dakwah (Studi Kasus : Lembaga Dakwah Al-Misriyah)”, , 2017.

Rody. “Program Aplikasi Penjadwalan Khotib Pada Lembaga Corps Mubaligh Muhammadiyah Malang Berbasis Jsp Dan Xml. Tesis. Malang”: Fakultas Sains dan Teknologi UIN Maulana Malik Malang, 2011.

- Roth, dkk. *Information Integration : A new generation of information technology*.
ProQuest, 2, 2002.
- Rosa, dan M. Shalahuddin. *Modul Pembelajaran Rekayasa Perangkat Lunak
(Terstruktur dan Berorientasi Objek)*. Bandung: Modula, 2011.
- Saluky. "Pengertian MYSQL", <http://www.etunas.com/web/pengertian-mysql.html>. 2006.
- Setiadi, Ade. "Pembangunan notifikasi system jatuh tempo tunggakan pembayaran sekolah berbasis sms gateway dan web pada SMK Al-Amanah". *Tesis*. Tangerang: Teknik informatika Sekolah Tinggi Manajemen Dan Ilmu Komputer (STMIK) Raharja, 2014.
- Shihab. M. Quraish. *Tafsir Al Misbah: Pesan, Kesan dan Keserasian Al-Quran*. Jakarta: Lentera Hati, 2009.
- Saputra. Wahidin, *Pengantar Ilmu Dakwah*. Jakarta: PT. Rajagrafindo Persada, 2011.
- Wikipedia. *Lembaga dakwah*. https://id.wikipedia.org/wiki/Lembaga_dakwah (12 Mei 2016).

LAMPIRAN

```

package id.faldy.ustads.adapter;

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.Button;
import android.widget.LinearLayout;
import android.widget.TextView;

import java.util.ArrayList;
import java.util.List;

import id.faldy.ustads.masyarakat.MapActivity;
import id.faldy.ustads.R;
import id.faldy.ustads.model.Approval;
import id.faldy.ustads.rest.ApiClient;
import id.faldy.ustads.rest.ApiInterface;
import id.faldy.ustads.tools.getNama;
import id.faldy.ustads.ustads.ApprovalNotif;
import id.faldy.ustads.ustads.JadwalActivity;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class AdapterApproval extends RecyclerView.Adapter<AdapterApproval.MyViewHolder> {
    private List<Approval> data;
    private List mListData;
    private boolean mIsLoading = false;
    private String mNamam, mKategoim;
    private int mCode;

    public AdapterApproval() {
        data = new ArrayList<>();
    }

    public void updateAdapter(List<Approval> list, String nama, String kategori, int code) {
        data = new ArrayList<>(list);
        mNamam = nama;
        mCode = code;
        mKategoim = kategori;
        notifyDataSetChanged();
    }

    public void showItemLoading() {
        if (mIsLoading) {
            return;
        }

        int postition = data.size();
        mListData.add(postition, "LOADER");
        mIsLoading = true;
    }

```

```

}

@SuppressWarnings("unchecked")
public void clearList() {
    data.clear();
}

public void removeItemLoading() {
    if (!mIsLoading) {
        return;
    }
    int position = mListData.size() - 1;
    data.remove(position);
    mIsLoading = false;
}

@NonNull
@Override
public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {
    View mView;
    LayoutInflater inflater = LayoutInflater.from(parent.getContext());
    if (mCode == 1) mView = inflater.inflate(R.layout.list_approval, parent, false);
    else mView = inflater.inflate(R.layout.list_approval1, parent, false);
    return new AdapterApproval.MyViewHolder(mView);
}

@Override
public void onBindViewHolder(@NonNull final MyViewHolder holder, final int position) {
    ApiInterface mApiInterface = ApiClient.getClient().create(ApiInterface.class);
    String use;
    if (mCode == 1) {
        getNama test=new getNama();
        use=test.nama(data.get(position).getId_ustadz());
    }
    else {
        getNama test=new getNama();
        use=test.nama(data.get(position).getId_user());
    }
    holder.mKategori.setText(data.get(position).getKategori() + " ( " + use + " )");
    holder.mTanggal.setText(data.get(position).getTanggal());
    holder.mWaktu.setText(data.get(position).getWaktu());
    holder.mLokasi.setText(data.get(position).getLokasi());
    if (mCode == 1) {
        holder.mStatus.setVisibility(View.VISIBLE);
        holder.mStatus.setText(data.get(position).getStatus());
    }
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
//            Intent i = new Intent(v.getContext(),ProfileActivity.class);
//            i.putExtra("nama",data.get(position).getNama());
//            i.putExtra("kategori",data.get(position).getKategori());
//            i.putExtra("riwayat",data.get(position).getRiwayat());
//            i.putExtra("lokasi",data.get(position).getLokasi());
//            i.putExtra("id",data.get(position).getId_ustadz());
//            v.getContext().startActivity(i);
//            ((AppCompatActivity) v.getContext()).overridePendingTransition(R.anim.enter, R.anim.exit);

```

```

    }
});
holder.mLokasi.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (mCode != 1) {
            Intent i = new Intent(v.getContext(), MapActivity.class);
            i.putExtra("lat", data.get(position).getLat());
            i.putExtra("longi", data.get(position).getLongi());
            i.putExtra("lokasi", data.get(position).getLokasi());
            i.putExtra("code", "1");
            v.getContext().startActivity(i);
            ((AppCompatActivity) v.getContext()).overridePendingTransition(R.anim.enter, R.anim.exit);
        }
    }
});

holder.mTolak.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        final ApiInterface mApiInterface=ApiClient.getClient().create(ApiInterface.class);
        Call<ResponseBody> set=mApiInterface.SetStatus(String.valueOf(data.get(position).getId()),"Ditolak");
        set.enqueue(new Callback<ResponseBody>() {
            @Override
            public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
                Log.d("retro","status ganti code = "+response.code());
                notifyItemChanged(position);
                ApprovalNotif.ma.refresh();
                JadwalActivity.ma.refresh();
            }

            @Override
            public void onFailure(Call<ResponseBody> call, Throwable t) {

            }
        });
    }
});

holder.mTerima.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        final ApiInterface mApiInterface=ApiClient.getClient().create(ApiInterface.class);
        Call<ResponseBody> set=mApiInterface.SetStatus(String.valueOf(data.get(position).getId()),"Diterima");
        set.enqueue(new Callback<ResponseBody>() {
            @Override
            public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
                Log.d("retro","status ganti code = "+response.code());
                notifyItemChanged(position);
                Call<ResponseBody>
                sett=mApiInterface.pushJadwal(String.valueOf(data.get(position).getId_ustadz()),data.get(position).getKategori(),data
                .get(position).getLokasi(),data.get(position).getWaktu(),data.get(position).getTanggal());
                sett.enqueue(new Callback<ResponseBody>() {
                    @Override
                    public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
                        Log.d("retro","tambah jadwal code = "+response.code());
                        ApprovalNotif.ma.refresh();
                        JadwalActivity.ma.refresh();
                    }
                });
            }
        });
    }
});

```

```

    }

    @Override
    public void onFailure(Call<ResponseBody> call, Throwable t) {

    }
});
}

@Override
public void onFailure(Call<ResponseBody> call, Throwable t) {

}
});
}
});
}

@Override
public int getItemCount() {
    return data.size();
}

public class MyViewHolder extends RecyclerView.ViewHolder {
    public TextView mName;
    public TextView mKategori;
    public TextView mLokasi;
    public TextView mWaktu;
    public TextView mTanggal;
    public Button mTolak;
    public Button mTerima;
    public TextView mStatus;
    public LinearLayout mKonfirmasi;

    public MyViewHolder(View itemView) {
        super(itemView);
        mStatus = itemView.findViewById(R.id.status);
        mKonfirmasi = itemView.findViewById(R.id.konfirmasi);
        mTerima = itemView.findViewById(R.id.terima);
        mTolak = itemView.findViewById(R.id.tolak);
        mName = itemView.findViewById(R.id.nama);
        mKategori = itemView.findViewById(R.id.kategori);
        mLokasi = itemView.findViewById(R.id.lokasi);
        mWaktu = itemView.findViewById(R.id.waktu);
        mTanggal = itemView.findViewById(R.id.tanggal);
    }
}
}
}

```

```

package id.faldy.ustads.adapter;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;

```

```

import android.widget.TextView;

import java.util.ArrayList;
import java.util.Calendar;
import java.util.List;

import id.faldy.ustads.R;
import id.faldy.ustads.model.Jadwal;

public class AdapterJadwal extends RecyclerView.Adapter<AdapterJadwal.MyViewHolder> {
    private List<Jadwal> data;
    private List mListData;
    private boolean mIsLoading = false;
    public static Integer id_us;
    private String mNamam, mKategoim;
    private int mCode;

    public AdapterJadwal(){
        data = new ArrayList<>();
    }

    public void updateAdapter(List<Jadwal> list,String nama,String kategori,int code){
        data = new ArrayList<>(list);
        mNamam=nama;
        mCode=code;
        mKategoim=kategori;
        notifyDataSetChanged();
    }

    public void showItemLoading(){
        if(mIsLoading){
            return;
        }

        int postition = data.size();
        mListData.add(postition,"LOADER");
        mIsLoading = true;
        notifyDataSetChanged();
    }

    @SuppressWarnings("unchecked")
    public void clearList(){
        data.clear();
        notifyDataSetChanged();
    }

    public void removeItemLoading(){
        if(!mIsLoading){
            return;
        }
        int position = mListData.size() - 1;
        data.remove(position);
        mIsLoading = false;
        notifyDataSetChanged();
    }
}

```

```

@NonNull
@Override
public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {
    View mView ;
    LayoutInflater inflater=LayoutInflater.from(parent.getContext());
    if(mCode==1)
        mView=inflater.inflate(R.layout.list_approval,parent,false);
    else mView=inflater.inflate(R.layout.list_approval1,parent,false);
    return new AdapterJadwal.MyViewHolder(mView);
}

@Override
public void onBindViewHolder(@NonNull MyViewHolder holder, final int position) {
    holder.mKategori.setText(data.get(position).getKategori());
//    holder.mNama.setText(mNamam);
    Calendar c = Calendar.getInstance();

    holder.mTanggal.setText(data.get(position).getTanggal());
    holder.mWaktu.setText(data.get(position).getWaktu());
    holder.mLokasi.setText(data.get(position).getLokasi());
    if(mCode==7 || mCode==1) holder.mKonfirmasi.setVisibility(View.GONE);
    else holder.mKonfirmasi.setVisibility(View.VISIBLE);
    if(mCode==7) holder.mKategori.setVisibility(View.GONE);
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
//            Intent i = new Intent(v.getContext(),ProfileActivity.class);
//            i.putExtra("nama",data.get(position).getNama());
//            i.putExtra("kategori",data.get(position).getKategori());
//            i.putExtra("riwayat",data.get(position).getRiwayat());
//            i.putExtra("lokasi",data.get(position).getLokasi());
//            i.putExtra("id",data.get(position).getId_ustadz());
//            v.getContext().startActivity(i);
//            ((AppCompatActivity) v.getContext()).overridePendingTransition(R.anim.enter, R.anim.exit);
        }
    });
}

@Override
public int getItemCount() {
    return data.size();
}

public class MyViewHolder extends RecyclerView.ViewHolder{
    public TextView mName;
    public TextView mKategori;
    public TextView mLokasi;
    public TextView mWaktu;
    public TextView mTanggal;
    public LinearLayout mKonfirmasi;
    public MyViewHolder(View itemView) {
        super(itemView);
        mKonfirmasi=itemView.findViewById(R.id.konfirmasi);
        mName=itemView.findViewById(R.id.nama);
        mKategori=itemView.findViewById(R.id.kategori);
        mLokasi=itemView.findViewById(R.id.lokasi);
        mWaktu=itemView.findViewById(R.id.waktu);
        mTanggal=itemView.findViewById(R.id.tanggal);
    }
}

```

```
}  
}  
}
```

```
package id.faldy.ustads.adapter;
```

```
import android.support.annotation.NonNull;  
import android.support.v7.widget.RecyclerView;  
import android.text.TextUtils;  
import android.util.Log;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.Button;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;
```

```
import com.squareup.picasso.Picasso;
```

```
import java.util.ArrayList;  
import java.util.List;
```

```
import id.faldy.ustads.R;  
import id.faldy.ustads.model.ModelApprovalUstadz;
```

```
public class AdapterList extends RecyclerView.Adapter<AdapterList.MyViewHolder> {  
    List<ModelApprovalUstadz.listUstadz> mData = new ArrayList<>();  
    OnClickKonfirmasi mListener;
```

```
    public interface OnClickKonfirmasi {  
        void onReject(String id);  
        void onAccept(ModelApprovalUstadz.listUstadz data);  
    }
```

```
    public void deleteList(String id){  
        for(int a = 0; a<getItemCount(); a++){  
            if(TextUtils.equals(id, mData.get(a).getId())){  
                mData.remove(a);  
                notifyDataSetChanged();  
                return;  
            }  
        }  
    }
```

```
    public void setOnPageChange(OnClickKonfirmasi listener) {  
        mListener = listener;  
    }
```

```
    public void updateAdapter(List<ModelApprovalUstadz.listUstadz> list) {  
        if(!mData.isEmpty())  
            mData.clear();  
        mData.addAll(list);  
        notifyDataSetChanged();  
    }
```

```

@NonNull
@Override
public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {
    View mView;
    LayoutInflater inflater = LayoutInflater.from(parent.getContext());
    mView = inflater.inflate(R.layout.list_ustadz_approval, parent, false);
    return new AdapterList.MyViewHolder(mView);
}

@Override
public void onBindViewHolder(@NonNull final MyViewHolder holder, final int position) {
    holder.mNama.setText(mData.get(position).getNama());
    holder.mKategori.setText(mData.get(position).getKategori());
    holder.mLokasi.setText(mData.get(position).getLokasi());
    holder.mRiwayat.setText(mData.get(position).getRiwayat());
    Log.d("data", mData.get(position).getRiwayat());
    if (!TextUtils.isEmpty(mData.get(position).getDokumen())) {
        Picasso.get().load(mData.get(position).getDokumen()).into(holder.mDokumen);
    }
    holder.mTerima.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mListener != null) mListener.onAccept(mData.get(position));
        }
    });

    holder.mTolak.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            if (mListener != null) mListener.onReject(mData.get(position).getId());
        }
    });
}
@Override
public int getItemCount() {
    return mData.size();
}

public class MyViewHolder extends RecyclerView.ViewHolder {
    public TextView mName;
    public TextView mKategori;
    public TextView mLokasi;
    public TextView mRiwayat;
    public Button mTolak;
    public Button mTerima;
    public TextView mStatus;
    public LinearLayout mKonfirmasi;
    public ImageView mDokumen;

    public MyViewHolder(View itemView) {
        super(itemView);
        mStatus = itemView.findViewById(R.id.status);
        mRiwayat = itemView.findViewById(R.id.riwayat);
        mKonfirmasi = itemView.findViewById(R.id.konfirmasi);
        mTerima = itemView.findViewById(R.id.terima);
        mTolak = itemView.findViewById(R.id.tolak);
        mName = itemView.findViewById(R.id.nama);
    }
}

```

```

        mKategori = itemView.findViewById(R.id.kategori);
        mLokasi = itemView.findViewById(R.id.lokasi);
        mDokumen = itemView.findViewById(R.id.dokumen);
    }
}
}

```

```
package id.faldy.ustads.adapter;
```

```

import android.graphics.Color;
import android.support.annotation.NonNull;
import android.support.v4.content.ContextCompat;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

```

```

import com.github.mikephil.charting.charts.PieChart;
import com.github.mikephil.charting.data.PieData;
import com.github.mikephil.charting.data.PieDataSet;
import com.github.mikephil.charting.data.PieEntry;

```

```

import java.util.ArrayList;
import java.util.List;

```

```
import id.faldy.ustads.R;
```

```

public class AdapterPersentase extends RecyclerView.Adapter<AdapterPersentase.MyViewHolder> {
    private List<Float> data;

    public AdapterPersentase(){
        data = new ArrayList<>();
    }

    public void updateAdapter(List<Float> mData){
        data=mData;
        notifyDataSetChanged();
    }

    @SuppressWarnings("unchecked")
    public void clearList(){
        data.clear();
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public AdapterPersentase.MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {
        View mView ;
        LayoutInflater inflater=LayoutInflater.from(parent.getContext());
        mView=inflater.inflate(R.layout.persentase_item,parent,false);
        return new AdapterPersentase.MyViewHolder(mView);
    }
}

```

```

@Override
public void onBindViewHolder(@NonNull final AdapterPersentase.MyViewHolder holder, int position) {
    //code di bawah sepertinya akan berubah apabila telah menggunakan data yang sebenarnya
    float persen1, persen2, jumlah;
    int temp;
    if(data.size(>0){
        switch(position){
            case 0 : {
                persen1=data.get(0);
                holder.mPersen.setText((int)persen1+"%");
                jumlah=data.get(8);
                temp= (int) jumlah;
                holder.mTitle.setText("KHUTBAH (" +temp+""));
                persen2=data.get(4);
                holder.mPencapaian.setText("Permintaan Jadwal (" +(int)persen2+""));

                holder.bindPieData(ContextCompat.getColor(holder.itemView.getContext(),R.color.aaa_common_color_purple),persen1,100-persen1);
                } break;
            case 1 : {
                persen1=data.get(1);
                holder.mPersen.setText((int)persen1+"%");
                jumlah=data.get(9);
                temp= (int) jumlah;
                holder.mTitle.setText("TARWIH (" +temp+""));
                persen2=data.get(5);
                holder.mPencapaian.setText("Permintaan Jadwal (" +(int)persen2+""));

                holder.bindPieData(ContextCompat.getColor(holder.itemView.getContext(),R.color.aaa_common_color_orange),persen1,100-persen1);
                } break;
            case 2 : {
                persen1=data.get(2);
                holder.mPersen.setText((int)persen1+"%");
                jumlah=data.get(10);
                temp= (int) jumlah;
                holder.mTitle.setText("TAKZIAH (" +temp+""));
                persen2=data.get(6);
                holder.mPencapaian.setText("Permintaan Jadwal (" +(int)persen2+""));

                holder.bindPieData(ContextCompat.getColor(holder.itemView.getContext(),R.color.aaa_common_color_green),persen1,100-persen1);
                } break;
            case 3 : {
                persen1=data.get(3);
                holder.mPersen.setText((int)persen1+"%");
                jumlah=data.get(11);
                temp= (int) jumlah;
                holder.mTitle.setText("MAULID (" +temp+""));
                persen2=data.get(7);
                holder.mPencapaian.setText("Permintaan Jadwal (" +(int)persen2+""));

                holder.bindPieData(ContextCompat.getColor(holder.itemView.getContext(),R.color.aaa_common_color_blue),persen1,100-persen1);
                RecyclerView.LayoutParams layoutParams = (RecyclerView.LayoutParams)
                holder.mContainerItem.getLayoutParams();
                layoutParams.rightMargin =
                holder.itemView.getContext().getResources().getDimensionPixelSize(R.dimen.aaa_card_pie_margin);
            }
        }
    }
}

```

```

        holder.mContainerItem.setLayoutParams(layoutParams);
    } break;
}
}
}

```

```

@Override
public int getItemCount() {
    return 4;
}

```

```

public class MyViewHolder extends RecyclerView.ViewHolder{
    private PieChart mPieChart;
    private TextView mTitle;
    private TextView mPersen;
    private TextView mPencapaian;
    private CardView mContainerItem;
    private MyViewHolder(View itemView) {
        super(itemView);
        mContainerItem = itemView.findViewById(R.id.aaa_container_item);
        mPieChart=itemView.findViewById(R.id.item_piechart);
        mTitle=itemView.findViewById(R.id.title);
        mPersen=itemView.findViewById(R.id.persen);
        mPencapaian=itemView.findViewById(R.id.pencapaian);
    }
}

```

```

private void bindPieData(int mColor, float percent, float offset){
    mPieChart.setDrawEntryLabels(false);
    mPieChart.getLegend().setEnabled(false);
    mPieChart.setDrawHoleEnabled(false);
    mPieChart.setTouchEnabled(false);
    mPieChart.setDescription(null);
    PieEntry entryAbsen = new PieEntry(percent);
    PieEntry entryEmpty = new PieEntry(offset);
    List<PieEntry> mEntries = new ArrayList<>();
    mEntries.add(0,entryAbsen);
    mEntries.add(1,entryEmpty);
    PieDataSet pieDataSet = new PieDataSet(mEntries,mTitle.getText().toString());
    pieDataSet.setSliceSpace(4f);
}

```

```

    pieDataSet.setColors(mColor,Color.parseColor("#B323313B"));
    pieDataSet.setDrawValues(false);
    PieData pieData = new PieData(pieDataSet);
    mPieChart.setData(pieData);
    mPieChart.animate();
    mPieChart.invalidate();
}
}
}

```

```

package id.faldy.ustads.adapter;

```

```

import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.RecyclerView;

```

```
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
```

```
import java.util.ArrayList;
import java.util.List;
```

```
import id.faldy.ustads.R;
import id.faldy.ustads.model.Ustadz;
import id.faldy.ustads.ustads.ProfileActivity;
```

```
public class AdapterUstadz extends RecyclerView.Adapter<AdapterUstadz.MyViewHolder> {
    private List<Ustadz> data;
    private List mListData;
    private boolean mIsLoading = false;
    public static Integer id_us;

    public AdapterUstadz(){
        data = new ArrayList<>();
    }
    public void updateAdapter(List<Ustadz> list){
        data = new ArrayList<>(list);
        notifyDataSetChanged();
    }

    public void showItemLoading(){
        if(mIsLoading){
            return;
        }

        int position = data.size();
        mListData.add(position,"LOADER");
        mIsLoading = true;
        notifyDataSetChanged();
    }

    @SuppressWarnings("unchecked")
    public void clearList(){
        data.clear();
        notifyDataSetChanged();
    }

    public void removeItemLoading(){
        if(!mIsLoading){
            return;
        }
        int position = mListData.size() - 1;
        data.remove(position);
        mIsLoading = false;
        notifyDataSetChanged();
    }

    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int i) {
        View mView = LayoutInflater.from(parent.getContext()).inflate(R.layout.list_ustadz, parent, false);
```

```
    return new AdapterUstadz.MyViewHolder(mView);
}
```

```
@Override
```

```
public void onBindViewHolder(@NonNull MyViewHolder holder, final int position) {
    holder.mKategori.setText(data.get(position).getKategori());
    holder.mNama.setText(data.get(position).getNama());
    holder.mLokasi.setText(data.get(position).getLokasi());
    holder.itemView.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent i = new Intent(v.getContext(),ProfileActivity.class);
            id_us=data.get(position).getId_ustadz();
            i.putExtra("nama",data.get(position).getNama());
            i.putExtra("kategori",data.get(position).getKategori());
            i.putExtra("riwayat",data.get(position).getRiwayat());
            i.putExtra("lokasi",data.get(position).getLokasi());
            i.putExtra("id",data.get(position).getId_ustadz());
            v.getContext().startActivity(i);
            ((AppCompatActivity) v.getContext()).overridePendingTransition(R.anim.enter, R.anim.exit);
        }
    });
}
```

```
@Override
```

```
public int getItemCount() {
    return data.size();
}
```

```
public class MyViewHolder extends RecyclerView.ViewHolder{
    public TextView mName;
    public TextView mKategori;
    public TextView mLokasi;
    public MyViewHolder(View itemView) {
        super(itemView);
        mName=itemView.findViewById(R.id.nama);
        mKategori=itemView.findViewById(R.id.kategori);
        mLokasi=itemView.findViewById(R.id.lokasi);
    }
}
```

```
package id.faldy.ustads.admin;
```

```
import android.annotation.SuppressLint;
import android.content.Intent;
import android.graphics.Color;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.design.widget.AppBarLayout;
import android.support.design.widget.TabLayout;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.view.ViewPager;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.LinearLayoutManager;
```

```

import android.support.v7.widget.RecyclerView;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.widget.FrameLayout;
import android.widget.ImageView;
import android.widget.RelativeLayout;
import android.widget.TextView;

import com.transitionseverywhere.TransitionManager;

import java.util.ArrayList;
import java.util.List;

import id.faldy.ustads.R;
import id.faldy.ustads.adapter.AdapterList;
import id.faldy.ustads.adapter.AdapterUstadz;
import id.faldy.ustads.model.ModelApprovalUstadz;
import id.faldy.ustads.rest.ApiClient;
import id.faldy.ustads.rest.ApiInterface;
import id.faldy.ustads.user.LoginActivity;
import okhttp3.ResponseBody;
import retrofit2.Call;
import retrofit2.Callback;
import retrofit2.Response;

public class AdminMain extends AppCompatActivity {
    TextView textToolbarTitle;
    boolean visible;
    public static AdminMain ma;
    FrameLayout search;
    RelativeLayout menu;
    FrameLayout container_cover;
    RecyclerView mRecyclerView;
    AdapterList mRecyclerAdapter;
    public static Toolbar mToolbar;
    public FrameLayout container_view;
    public int staticColor= R.color.colorActionRipple;
    @Override
    protected void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main_admin);
        menu =findViewById(R.id.menu);
        ma=this;
        container_view=findViewById(R.id.container);
        final ImageView logout=findViewById(R.id.exit);
        container_cover=findViewById(R.id.container_cover);
        mRecyclerView = findViewById(R.id.listapproval);
        final LinearLayoutManager mLayoutManager = new LinearLayoutManager(this);
        mRecyclerAdapter = new AdapterList();
        mRecyclerView.setHasFixedSize(true);
        mRecyclerView.setNestedScrollingEnabled(true);
        mRecyclerView.setLayoutManager(mLayoutManager);
        mRecyclerView.setAdapter(mRecyclerAdapter);
        logout.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }
}

```

```

        Intent i=new Intent(getApplicationContext(), LoginActivity.class);
        startActivity(i);
        overridePendingTransition(R.anim.enter, R.anim.exit);
    }
});
final Intent intent=getIntent();

mToolbar = findViewById(R.id.toolbar);
setSupportActionBar(mToolbar);
getSupportActionBar().setDisplayShowTitleEnabled(false);
textToolbarTitle = findViewById(R.id.toolbar_title);
textToolbarTitle.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        TransitionManager.beginDelayedTransition(mToolbar);
        visible = !visible;
        search.setVisibility(visible ? View.VISIBLE : View.GONE);
        if(search.getVisibility()==View.VISIBLE) {
            textToolbarTitle.setText("");
        } else{
            textToolbarTitle.setText(" Portal Admin");
        }
    }
});

AppBarLayout appBarLayout = findViewById(R.id.container_toolbar);
appBarLayout.addOnOffsetChangedListener(new AppBarLayout.OnOffsetChangedListener() {
    @SuppressWarnings("ResourceAsColor")
    @Override
    public void onOffsetChanged(AppBarLayout appBarLayout, int verticalOffset) {
        int heightDiff = appBarLayout.getTotalScrollRange() - mToolbar.getHeight();
        verticalOffset = Math.abs(verticalOffset);
        if (verticalOffset >= heightDiff && !textToolbarTitle.isShown()) {
            textToolbarTitle.setVisibility(View.VISIBLE);
            logout.setVisibility(View.VISIBLE);
            mToolbar.setVisibility(View.VISIBLE);
            mToolbar.setBackgroundColor(getResources().getColor(R.color.color_green_alert));
        } else if (verticalOffset <= heightDiff && textToolbarTitle.isShown()) {
            textToolbarTitle.setVisibility(View.GONE);
            mToolbar.setVisibility(View.GONE);
            logout.setVisibility(View.GONE);
        }
    }
});

final ApiInterface mApiInterface= ApiClient.getClient().create(ApiInterface.class);
Call<ModelApprovalUstadz> modelCall = mApiInterface.getUstadzApproval();
modelCall.enqueue(new Callback<ModelApprovalUstadz>() {
    @Override
    public void onResponse(Call<ModelApprovalUstadz> call, Response<ModelApprovalUstadz> response) {
        if(response.isSuccessful()){
            Log.d("akses", "sukses "+ response.body().getResult().size());
            mRecyclerAdapter.updateAdapter(response.body().getResult());
        }
    }
}

@Override
public void onFailure(Call<ModelApprovalUstadz> call, Throwable t) {

```

```

    }
  });

  mRecyclerAdapter.setOnPageChange(new AdapterList.OnCLickKonfirmasi() {
    @Override
    public void onReject(final String id) {
      Call<ResponseBody> modelCall = mApiInterface.rejectUstadz(id);
      modelCall.enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
          if(response.isSuccessful())
            mRecyclerAdapter.deleteList(id);
        }

        @Override
        public void onFailure(Call<ResponseBody> call, Throwable t) {

        }
      });
    }

    @Override
    public void onAccept(final ModelApprovalUstadz.listUstadz data) {
      Call<ResponseBody> modelCall = mApiInterface.acceptUser(data.getNama(), data.getPassword(),
data.getEmail(), data.getRiwayat(), data.getKategori(), data.getLokasi(), data.getLat(), data.getLongi());
      modelCall.enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
          if (response.isSuccessful())
            mRecyclerAdapter.deleteList(data.getId());
        }

        @Override
        public void onFailure(Call<ResponseBody> call, Throwable t) {

        }
      });
    }
  });
}

@Override
public void onBackPressed() {

}

}

```

```
package id.faldy.ustads.common;
```

```
public class AboutActivity {
```

```
package id.faldy.ustads.common;
```

```
import android.content.Intent;
```

```

import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.widget.ProgressBar;

import id.faldy.ustads.user.LoginActivity;
import id.faldy.ustads.R;

public class SplashScreen extends AppCompatActivity {
    ProgressBar mProgress;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.splashscreen);
        mProgress = (ProgressBar) findViewById(R.id.progress);

        // Start lengthy operation in a background thread
        new Thread(new Runnable() {
            public void run() {
                doWork();
                openMainActivity();
            }
        }).start();
    }

    private void doWork() {
        for (int progress=0; progress<100; progress+=10) {
            try {
                Thread.sleep(300);
                mProgress.setProgress(progress);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
    }

    private void openMainActivity(){
        startActivity(new Intent(getApplicationContext(), LoginActivity.class));
        overridePendingTransition(R.anim.fade_in,R.anim.fade_out);
        finish();
    }
}

```