

**PENGEMBANGAN APLIKASI *ONE DAY ONE JUZ* (ODOJ) BERBASIS  
ANDROID**

**(Skripsi)**

**Oleh**

**MUHAMMAD ARIEF HIDAYAT  
1517051086**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2022**

**PENGEMBANGAN APLIKASI *ONE DAY ONE JUZ* (ODOJ) BERBASIS  
ANDROID**

Oleh

**MUHAMMAD ARIEF HIDAYAT**

Skripsi

Sebagai Salah satu Syarat Untuk Mencapai Gelar  
**SARJANA KOMPUTER**

Pada

**Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam**



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2022**

## **ABSTRAK**

### **PEGEMBANGAN APLIKASI ONE DAY ONE JUZ (ODOJ) BERBASIS ANDROID**

**Oleh**

**Muhammad Arief Hidayat**

Program ODOJ yang memanfaatkan aplikasi instant messaging seperti Whatsapp dan Telegram tidak dapat menangani perekapan "tilawatil Quran" tiap juz perhari. Penelitian ini berupaya untuk mengembangkan aplikasi ODOJ berbasis Android yang dapat membantu umat muslim dalam melakukan program ODOJ khususnya dalam proses pelaporan dan perekapan tilawah. Data yang digunakan dalam penelitian ini didapatkan dengan cara observasi, wawancara dan studi litelatur. Metode pengembangan sistem yang dipilih dalam penelitian ini adalah metode pengembangan Extreme Programming. Sistem ini dibangun berbasis android, proses pembuatan program (coding) menggunakan bahasa pemrograman kotlin dengan firebase sebagai databasenya. Pengujian dalam penelitian ini menggunakan pengujian Black-Box dengan metode *Equivalence Partitioning* (EP) Hasil pengujian yang telah dilakukan menunjukkan bahwa sistem dapat berfungsi sesuai kebutuhan pengguna dan pada hasil *user acceptance test* sistem masuk dalam kategori sangat baik.

Kata kunci: ODOJ, Android, Kotlin, Firebase, *Equivalence Partitioning*

## **ABSTRACT**

### **ONE DAY ONE JUZ (ODOJ) APPLICATION DEVELOPMENT BASED ON ANDROID**

**By**

**Muhammad Arief Hidayat**

The ODOJ program that utilizes instant messaging applications such as Whatsapp and Telegram cannot handle the recording of "Tilawatil Quran" every juz per day. This study seeks to develop an Android-based ODOJ application that can assist Muslims in conducting the ODOJ program, especially in the process of reporting and recording recitations. Data used in this study was obtained by means of observation, interviews, and literature studies. The system development method chosen in this research is the Extreme Programming development method. This system is built based on android, the programming process (coding) uses the Kotlin programming language with firebase as the database. The test in this study uses Black-Box testing with the Equivalence Partitioning (EP) method. The results of the tests that have been carried out show that the system can function according to user needs and the results of the user acceptance test of the system are in the very good category.

**Keywords:** ODOJ, Android, Kotlin, Firebase, *Equivalence Partitioning*.

Judul Skripsi : PENGEMBANGAN APLIKASI *ONE DAY ONE JUZ* (ODOJ) BERBASIS ANDROID

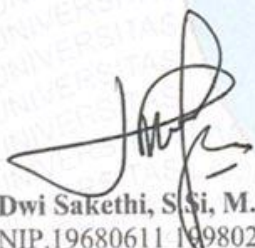
Nama Mahasiswa : Muhammad Arief Hidayat

Nomor Pokok Mahasiswa : 1517051086

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam




  
**Dwi Sakethi, S.Si, M.Kom**  
NIP.19680611 19802 1 001

  
**Ardiansyah, S.Kom., M.Kom**  
NIP. 19870128 201803 1 001

2. Mengetahui

Ketua Jurusan Ilmu Komputer  
FMIPA Universitas Lampung

  
**Didik Kurniawan, M.T.**  
NIP. 19800419 200501 1 004

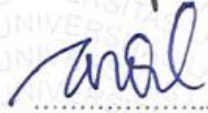
**Mengesahkan**

**1. Tim Penguji**

**Ketua : Dwi Sakethi, S.Si, M.Kom**



**Sekretaris : Ardiansyah, S.Kom., M.Kom**



**Penguji  
Bukan Pembimbing : Febi Eka Febriansyah, M.T**



**2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam**



**Dr. Eng. Suripto Dwi Yuwono, M.T**  
NIP. 197407052000031001

**Tanggal Lulus Ujian Skripsi: 25 Januari 2022**

## PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi saya yang berjudul “Pengembangan Aplikasi *One Day One Juz* (Odoj) Berbasis Android” Merupakan karya saya sendiri bukan hasil karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila terbukti di kemudian hari bahwa skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima

Bandar Lampung, 10 Juni 2022



Muhammad Arief Hidayat  
NPM. 1517051086

## **Persembahan**

Bismillahirrahmanirrahim Dengan rasa syukur kepada Allah SWT atas segala ridho dan berkat-Nya serta kerendahan hati ku

Skripsi ini kupersembahkan untuk

Ibu dan Ayah, terima kasih selalu atas doa dan kasih sayang yang tiada terhingga. Terimakasih atas segala dukungan baik moril ataupun materi. Terimakasih atas doa yang tidak pernah lelah diberikan, terimakasih atas segala keringat dan air mata. Terimakasih untuk selalu percaya

Keluarga dan sahabat- sahabat tersayang yang selalu memberikan dukungan dan keceriaan tanpa henti.

Terima kasih juga untuk semua pihak yang terlibat dalam pembuatan skripsi ini. Semoga Allah senantiasa membalas setiap kebaikan kalian.

Terima kasih saya ucapkan juga untuk teman-teman Com(B)Nation dan Almamater Tercinta, Universitas Lampung



## SANWACANA

Puji syukur penulis ucapkan kehadiran Tuhan Yang Maha Esa, karena atas rahmat dan hidayah-Nya skripsi ini dapat diselesaikan. Skripsi dengan judul **“Pengembangan Aplikasi *One Day One Juz* (ODOJ) Berbasis Android”** adalah salah satu syarat untuk memperoleh gelar sarjana komputer pada Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung

Dalam penyusunan skripsi ini, penulis mendapat bantuan, dukungan dan motivasi dari berbagai pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak yang telah membantu dan berperan besar dalam penyusunan skripsi ini, antara lain.

1. Kedua orang tuaku tercinta, Ayah dan Ibu yang selalu mendoakan mendukung, membimbing dan memotivasi penulis selama ini.
2. Bapak Dwi Sakethi, S.Si, M.Kom selaku dosen pembimbing utama yang telah sabar dalam memberikan bimbingan, saran, dan motivasi dalam pembuatan laporan skripsi ini.
3. Ardiansyah, S.Kom., M.Kom selaku dosen pembimbing kedua yang telah sabar dalam memberikan bimbingan, saran, dan motivasi dalam pembuatan laporan skripsi ini.
4. Bapak Febi Eka Febriansyah, M.T., selaku pembahas yang telah memberikan saran dan masukan yang bermanfaat dalam perbaikan skripsi ini.
5. Bapak Dr. Eng. Suropto Dwi Yuwono, M.T., selaku dekan FMIPA Universitas Lampung.

6. Bapak Didik Kurniawan, S.Si., M.T., Selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung.
7. Ibu Astria Hijriani, S.Kom., M.Kom., Selaku Sekretaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Bapak Drs. Rd. Irwan Adi Pribadi, M.Kom., selaku dosen Pembimbing Akademik yang telah membimbing dan memberi nasihat selama perkuliahan.
9. bapak dan Ibu dosen Jurusan Ilmu Komputer yang telah memberikan ilmu yang bermanfaat dan pengalaman selama penulis menjadi mahasiswi.
10. Para staff dan pegawai Jurusan Ilmu Komputer yang telah membantu memudahkan segala urusan administrasi penulis di Jurusan Ilmu Komputer.
11. Beasiswa BIDIKMISI, terimakasih karena telah memberikan kesempatan sehingga penulis dapat merasakan belajar di perguruan tinggi
12. Keluarga ODOJ yang telah bersedia memberikan dan memfasilitasi untuk dijadikan sebagai penelitian skripsi ini.
13. Keluarga besar ComBnation yang sudah memberikan doa dan semangat
14. Teman-teman Ilmu Komputer angkatan 2015 yang menjadi teman satu angkatan selama menjalankan masa studi
15. Noverina Rahmaniyanti yang selalu memberikan suport dan doa selama masa studi dan penelitian ini berlangsung.

Akhir kata, penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi penulis berharap skripsi ini dapat berguna dan bermanfaat bagi semua. Semoga dengan bantuan dan dukungan yang diberikan mendapat balasan pahala di sisi Allah Subhanahu Wa Ta'ala. Aamiin.

## DAFTAR ISI

	Halaman
<b>DAFTAR TABEL .....</b>	vii
<b>DAFTAR GAMBAR.....</b>	viii
<b>I. PENDAHULUAN.....</b>	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	3
1.3. Tujuan Penelitian.....	4
1.4. Manfaat .....	4
1.4.1 Manfaat Bagi Mahasiswa .....	4
1.4.2 Manfaat Bagi Pengguna.....	5
1.5. Ruang Lingkup.....	5
<b>II. TINJAUAN PUSTAKA.....</b>	6
2.1 Pengertian Alquran.....	6
2.2 Membaca Alquran.....	6
2.3 Sejarah ODOJ.....	7
2.4 Sistem Operasi Android .....	8
2.5 Versi Android.....	8
2.6 Arsitektur Android .....	9
2.6.1 <i>System Apps</i> .....	10
2.6.2 Kerangka Kerja API Java.....	10
2.6.3 Pustaka dan Waktu Proses Android.....	11
2.6.4 <i>Hardware Abstraction Layer</i> .....	11
2.6.5 Kernel Linux.....	12
2.7 Android Activity Lifecycle.....	12
2.8 Android Studio.....	13

2.9 Instans Messaging.....	14
2.10 Kotlin.....	15
2.11 Firebase.....	15
2.12 XML (Extensible Markup Language) .....	16
2.13 <i>Unified Modeling Language</i> .....	16
2.14 <i>Extreme programming (XP)</i> .....	21
2.15 <i>Black-box Testing</i> .....	23
<b>III. METODOLOGI PENELITIAN.....</b>	<b>24</b>
3.1. Tempat dan Waktu .....	24
3.2. Alat dan Bahan.....	24
3.2.1 Alat.....	24
3.2.2 Perangkat Lunak.....	25
3.3. Metode Penelitian.....	26
3.3.1 Metode Pengumpulan Data.....	26
3.3.2 Sumber Data.....	27
3.2.3 Metode Pengembangan Sistem.....	27
3.4. Analisis.....	28
3.3.1 Analisis Sistem.....	29
3.3.2 <i>User Requirement</i> .....	29
3.5. Perancangan Sistem.....	30
3.5.1 <i>Activity Diagram</i> .....	31
3.5.2 <i>Sequence Diagram</i> .....	37
3.6. Perancangan Antarmuka ( <i>Design Interface</i> ) .....	41
3.6.1. <i>Layout splash Screen</i> .....	42
3.6.2. <i>Layout Login</i> .....	43
3.6.3. <i>Layout Daftar</i> .....	43
3.6.4. <i>Layout Chat Admin</i> .....	44
3.6.5. <i>Layout Menu Utama (Dashboard)</i> .....	44
3.6.6. <i>Layout Room Chat</i> .....	45
3.6.7. <i>Layout Profil</i> .....	46
3.6.8. <i>Layout Laporan versi (“Odojer”)</i> .....	46
3.6.9. <i>Layout Grup – Admin</i> .....	47

3.6.10. <i>Layout</i> Laporan-Admin.....	48
3.6.11. <i>Layout</i> Grup Laporan-Admin.....	49
3.6.12. <i>Layout</i> Kontak.....	49
<b>3.7.</b> Metode Pengujian Sistem.....	50
<b>IV. HASIL DAN PEMBAHASAN</b> .....	54
4.1. Hasil Implementasi Sistem.....	54
4.1.1. <i>Layout Splash Screen</i> .....	54
4.1.2. <i>Layout Login</i> .....	55
4.1.3. <i>Layout</i> Daftar.....	56
4.1.4. <i>Layout Dashboard</i> .....	57
4.1.5. <i>Layout</i> Pesan Terakhir.....	58
4.1.6. <i>Layout</i> Kontak.....	59
4.1.7. <i>Layout</i> Riwayat <i>Pesan</i> .....	60
4.1.8. <i>Layout</i> Laporan Tilawah.....	61
4.1.9. <i>Layout</i> Menu Navigasi.....	62
4.1.10. <i>Layout History</i> Laporan Tilawah.....	63
4.1.11. <i>Layout</i> Grup.....	64
4.1.12. <i>Layout</i> Admin Grup.....	65
4.2. Pengujian Sistem.....	66
4.2.1 Pengujian Fungsional.....	67
4.2.2 Pengujian Non Fungsional .....	72
4.3. Pembahasan .....	74
<b>V. SIMPULAN DAN SARAN</b> .....	75
5.1 Simpulan .....	75
5.2 Saran.....	75
DAFTAR PUSTAKA.....	77
LAMPIRAN.....	79

## DAFTAR TABEL

Tabel	Halaman
2.1 <i>Android Activity Lifecycle</i> .....	12
2.2 Simbol <i>Use case Diagram</i> .....	17
2.3 Simbol <i>Activity Diagram</i> .....	19
2.4 Simbol <i>Sequence Diagram</i> .....	20
3.1 Rancangan Pengujian .....	50
4.1 Hasil Pengujian Versi Android .....	67
4.2 Interval dan Kategori Penilaian.....	73
4.3 Interval dan Kategori Penilaian.....	73

## DAFTAR GAMBAR

Gambar	Halaman
2.1 Arsitektur Android .....	10
2.2 Kerangka Kerja XP .....	21
3.1 <i>Use case Diagram</i> Aplikasi ODOJ .....	30
3.2 <i>Activity</i> Aplikasi ODOJ .....	31
3.3 <i>Activity Diagram</i> Menerima Pesan .....	32
3.4 <i>Activity Diagram</i> Mengubah Profil .....	33
3.5 <i>Activity Diagram</i> Membuat Grup .....	33
3.6 <i>Activity Diagram</i> Menambahkan “Odojer” ke dalam Grup .....	34
3.7 <i>Activity Diagram</i> Menambahkan Laporan .....	35
3.8 <i>Activity Diagram</i> Melihat Laporan .....	35
3.9 <i>Activity Diagram</i> Mengakses Statistik Laporan Grup .....	36
3.10 <i>Activity Diagram</i> Mengakses Statistik Laporan Grup .....	36
3.11 <i>Sequence Diagram</i> Mengirim Pesan .....	37
3.12 <i>Sequence Diagram</i> Menerima Pesan .....	38
3.13 <i>Sequence Diagram</i> Mengubah Profil .....	38
3.14 <i>Sequence Diagram</i> Membuat Grup .....	39
3.15 <i>Sequence Diagram</i> Memasukkan “Odojer” ke dalam grup .....	40
3.16 <i>Sequence Diagram</i> Menambahkan Laporan .....	40
3.17 <i>Sequence Diagram</i> Melihat Laporan .....	41
3.18 <i>Layout Splash Screen</i> .....	42
3.19 <i>Layout Login</i> .....	43
3.20 <i>Layout Daftar</i> .....	43
3.21 <i>Layout Chat Admin</i> .....	44
3.22 <i>Menu Utama</i> .....	45
3.23 <i>Layout Room Chat</i> .....	45
3.24 <i>Layout Profil</i> .....	46

3.25 <i>Layout Laporan “Odojer”</i> .....	47
3.26 <i>Layout Grup – Admin</i> .....	47
3.27 <i>Layout Laporan – Admin</i> .....	48
3.28 <i>Layout Grup Laporan – Admin</i> .....	49
3.29 <i>Layout Lihat Grup Laporan – Admin</i> .....	49
4.1 <i>Layout Splash Screen</i> .....	55
4.2 <i>Layout Login</i> .....	56
4.3 <i>Layout Daftar</i> .....	57
4.4. <i>Layout Dashboard</i> .....	58
4.5 <i>Layout Pesan Terakhir</i> .....	59
4.6 <i>Layout Kontak</i> .....	60
4.7 <i>Layout Riwayat Pesan</i> .....	61
4.8 <i>Layout Laporan Tilawah</i> .....	62
4.9 <i>Layout Menu Navigasi</i> .....	63
4.10 <i>Layout History Laporan tilawah</i> .....	64
4.11 <i>Layout Grup</i> .....	65
4.12 <i>Layout Admin Grup</i> .....	66



## I. PENDAHULUAN

### 1.1 Latar Belakang

Alquran sebagai pegangan hidup dari umat muslim merupakan hal yang wajib untuk dibaca dan dipelajari. Membaca Alquran bagi umat muslim merupakan perintah langsung dari Allah *Subhanahu wa ta'ala*. Perintah tersebut tertuang pada ayat pertama yang turun kepada Nabi Muhammad *Shalallahu Alaihi Wasallam* yang memiliki arti “Bacalah dengan menyebut nama Tuhanmu yang menciptakan” (Q.S Al-Alaq, 1).

“Telah menceritakan kepadaku “Abu Umamah Al Bahili ia “berkata Saya mendengar Rasulullah shallallahu 'alaihi wasallam bersabda: Bacalah Alquran karena sesungguhnya ia akan menjadi syafa'at bagi para pembacanya di hari kiamat ...”” (H.R Muslim, 1337).

Hadis yang diriwayatkan oleh Imam Muslim tersebut menerangkan bahwa para pembaca Alquran diberikan keistimewaan untuk mendapatkan syafaat di hari kiamat. Ayat-ayat Alquran dan hadis-hadis dari Rasulullah *Shalallahu Alaihi Wasallam* tersebut merupakan latar belakang lahirnya rumah-rumah Alquran di Indonesia.

Rumah Quran merupakan salah satu program yang di dalamnya melakukan aktivitas untuk membumikan Alquran baik dengan cara menghafalnya maupun dengan membacanya. *One Day One Juz* (ODOJ) adalah program yang diinisiasi oleh alumni Rumah Quran dengan tujuan meringankan dan

memfasilitasi masyarakat yang ingin membaca Alquran dengan target satu juz per hari (Onedayonejuz, 2017).

Meningkatnya jumlah Rumah Quran diiringi dengan meningkatnya jumlah pengguna ponsel pintar di Indonesia, tercatat dari tahun 2016 sampai 2018 pengguna ponsel pintar di Indonesia selalu meningkat dari tahun ke tahun. Pengguna ponsel pintar diprediksi akan mencapai angka 92 juta di tahun 2019 (Databoks, 2016). Berdasarkan data sementara yang ada, pada tahun 2018 Rumah Quran yang dirintis oleh Seorang Ulama tiap tahun bertambah dan kini jumlahnya sudah mencapai seratus unit lebih yang tersebar di 17 kabupaten dan kota di provinsi Sumatera Selatan (Yayan, 2018).

Ponsel cerdas dengan sistem operasi Android yang *open source* dimanfaatkan oleh berbagai pihak dalam bidang agama. Muludi dkk (2015) yang membuat aplikasi Iqro' berbasis Android sebagai media pembelajaran alternatif untuk mempelajari Iqra agar lebih interaktif dan memberikan kemudahan dalam mempelajarinya. Komunitas ODOJ juga memanfaatkan aplikasi Android *instant messaging* seperti WhatsApp dan Telegram untuk melakukan pelaporan *tilawah* satu juz Alquran setiap harinya.

*Instant Messaging* saat ini mengalami perkembangan yang cukup pesat pada jaringan *user*, karena kemampuannya mengirimkan pesan secara singkat dan cepat antara pengguna telekomunikasi. *Instant messaging* menjadi perangkat yang sangat penting untuk industri di seluruh dunia (Zuliarso dan Februriyanti, 2013).

Program ODOJ yang diminati berbagai kalangan di Indonesia maupun mancanegara. Jumlah anggota ODOJ yang mencapai 125.000 pada tahun 2014 setelah satu tahun diluncurkan membuktikan tingginya peminat ODOJ di Indonesia. ODOJ juga berkembang di berbagai negara antara lain di Qatar terbentuk empat grup, Australia tiga grup, di Hongkong tiga grup, di Malaysia

dua grup dan grup campuran yang berasal dari negara Singapura, Mesir, Korea, Jepang, Jerman dan berbagai negara lainnya (Onedayonejuz, 2017).

Komunitas ODOJ yang terus meningkat dari tahun ke tahun tidak selalu berjalan dengan baik. Terdapat sedikit kendala seperti admin yang terkadang tidak *up to date* dalam melakukan perekapan "tilawatil Quran" tiap juz per harinya. Hal ini tentu saja belum dapat ditangani oleh aplikasi *instant messaging* yang tidak memiliki fitur tersebut.

Admin ODOJ yang kurang *up to date* dapat ditangani dengan melakukan pengembangan aplikasi ODOJ berbasis Android untuk melakukan penyetoran tilawah dan perekapan tilawah. Pengembangan aplikasi tersebut diharapkan juga dapat menarik minat masyarakat Indonesia agar mengikuti program ODOJ sehingga giat dalam membaca Alquran.

Berdasarkan latar belakang tersebut peneliti mengajukan penelitian berbasis Android yang dapat membantu proses perekapan "tilawatil Quran" per hari pada program ODOJ. Penelitian tersebut diberi judul "Pengembangan Aplikasi *One Day One Juz* Berbasis Android".

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang masalah yang telah diuraikan di atas, maka pokok permasalahan yang dihadapi adalah "Bagaimana mengembangkan aplikasi *One Day One Juz* berbasis Android?".

### **1.3 Tujuan Penelitian**

Adapun tujuan dari penelitian ini adalah untuk mengembangkan aplikasi ODOJ berbasis Android yang dapat membantu umat muslim dalam melakukan program ODOJ khususnya dalam proses pelaporan dan perekapan tilawah.

### **1.4 Manfaat**

Penelitian ini memberikan manfaat bagi pihak mahasiswa dan juga pengguna.

#### **1.4.1 Manfaat Bagi Mahasiswa**

Manfaat yang dapat diperoleh mahasiswa dari penelitian ini adalah untuk memperoleh gelar sarjana pada jurusan Ilmu Komputer Universitas Lampung serta untuk mengembangkan dan menerapkan ilmu yang telah dipelajari sebagai persiapan dalam dunia kerja.

#### **1.4.2 Manfaat Bagi Pengguna**

Manfaat yang dapat diperoleh pengguna dari penelitian ini adalah memudahkan proses perekapan laporan tilawah harian dalam program *One Day One Juz*.

## 1.5 Ruang Lingkup

Aplikasi *One Day One Juz* Berbasis Android ini memiliki ruang lingkup sebagai berikut.

1. Aplikasi ini menggunakan *platform* Android hanya dapat diakses menggunakan koneksi internet.
2. Aplikasi ini dapat melakukan proses pelaporan dalam program ODOJ.
3. Aplikasi ini dapat melakukan proses perekapan laporan harian dalam program ODOJ.
4. Aplikasi ini memiliki dua user yaitu Admin dan “Odojer”.
5. Aplikasi ini tidak dapat memvalidasi pelaporan tilawah.

## II TINJAUAN PUSTAKA

### 2.1 Pengertian Alquran

Alquran dapat didefinisikan melalui dua pendekatan yaitu pendekatan secara etimologi dan terminologi. Alquran secara bahasa berasal dari kata *qara'a*, *qira'atan*, *wa qur'an* yang berarti menghimpun atau mengumpulkan. Alquran secara bahasa berarti bacaan atau kumpulan huruf-huruf yang terstruktur dengan rapi. Sedangkan secara istilah para ulama *Ushul Fiqh* mendefinisikan Alquran sebagai kalam Allah yang diturunkan kepada Nabi Muhammad *Shalallahu Alaihi Wasallam* secara bertahap melalui perantara Malaikat Jibril dan merupakan sebuah pahala dengan membacanya, yang diawali surat Al-Fatihah dan diakhiri dengan surat An-Nas (Syarbini dan Jamhari, 2012).

### 2.2 Membaca Alquran

*“Telah menceritakan kepada kami ‘Abu Nu’aim Telah menceritakan kepada kami ‘Sufyan dari Alqamah bin Martsad dari Abu Abdurrahman As Sulami dari Utsman bin ‘Affan ia berkata ‘Nabi shallallahu ‘alaihi wasallam bersabda: ‘Orang yang paling utama di antara kalian adalah seorang yang belajar Alquran dan mengajarkannya.’””””” (HR. Bukhari, 4640)*

Membaca Alquran tidak saja sangat besar keutamaannya bahkan merupakan investasi yang ringan modalnya, tidak perlu uang yang banyak, hanya cukup berkeinginan kuat untuk mencari keridhaan Allah dengan membaca Alquran (Arifin dan Faqih, 2010)

Allah Ta'ala telah memberitahukan tentang kebaikan yang akan didapatkan oleh orang yang membaca Alquran. “*Dari Ibnu Mas'ud ra berkata “Rasulullah Shalallahu Alaihi Wasallam bersabda: “Barang siapa yang membaca satu huruf dari kitab Allah maka dia akan mendapatkan kebaikan dan kebaikan yang akan dia dapatkan akan dilipatgandakan sehingga mencapai sepuluh kali lipat, dan aku tidak mengatakan (alif lam mim) satu huruf, akan tetapi alif satu huruf, lam satu huruf dan mim satu huruf”””*” (HR. At Tirmidzi, 2910).

### 2.3 Sejarah ODOJ

Bhayu Subrata dan Pratama Widodo pada awalnya memperkenalkan program *One Day One Juz* (ODOJ) menggunakan fasilitas *short message service* (sms). Bhayu mengirimkan SMS *broadcast* berupa nasihat tentang Alquran untuk mengaji satu hari satu juz dan membuat buletin untuk disebar (Onedayonejuz, 2017).

Sedangkan Widodo, partner Bhayu, di tahun yang sama hingga 2009, membangun *fanspage* ODOJ di Facebook dengan harapan program ODOJ dapat menjangkau seluruh pelosok Indonesia dan seisi Dunia. Teknik mudah baca Alquran harian yaitu dengan menggunakan rumus  $2 \times 5$ , membaca dua lembar setelah salat fardu (lima waktu) maka *insya Allah* akan khatam satu juz dalam satu hari (Onedayonejuz, 2017).

## 2.4 Sistem Operasi Android

Sistem operasi adalah *software* yang mengontrol *hardware*. Sistem operasi dapat juga disebut sebagai program pengendali, yaitu program yang digunakan untuk mengontrol program lain. Tujuan dari sistem operasi adalah untuk mengontrol eksekusi program-program aplikasi dan berfungsi sebagai *interface* antara *user* dengan *hardware* (Pangera dan Ariyus, 2005).

Sistem operasi Android lahir pertama kali dari anak perusahaan Android Inc di tahun 2003. Pada tahun 2005 sistem operasi Android diakuisisi oleh Google LLC kemudian dikembangkan untuk ponsel cerdas dan perangkat selular lainnya. Platform Android, berdasarkan pada *Kernel Linux*, terutama didesain untuk perangkat seluler layar sentuh seperti ponsel cerdas dan tablet (Spath, 2018).

Sistem operasi Android memiliki beberapa kelebihan. Kelebihan sistem operasi Android adalah sebagai berikut:

1. Sistem operasi Android merupakan *open source*
2. Sistem operasi Android dijalankan di banyak spesifikasi *hardware*
3. Dapat diaplikasikan di banyak peralatan *software*
4. Dikembangkan oleh Google

## 2.5 Versi Android

Sejak pertama kali muncul sampai sekarang, Android telah memiliki sejumlah pembaharuan. Pembaharuan ini dilakukan untuk memperbaiki *bug* dan menambah fitur-fitur yang baru. Versi-versi yang ada pada Android, yaitu (Developers, 2022):

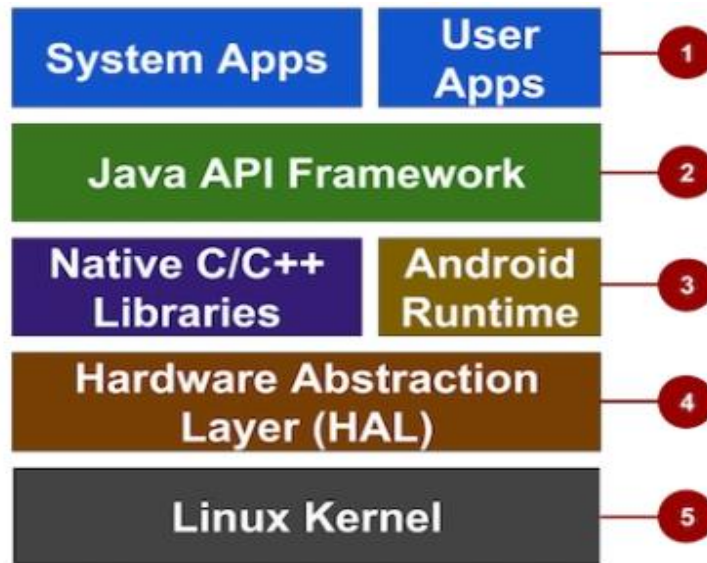
1. Android versi 1.1
2. Android versi 1.5 (*Cupcake*)



3. Android versi 1.6 (*Dounut*)
4. Android versi 2.0/2.1 (*Eclair*)
5. Android versi 2.2 (*Froyo: Frozen Yoghurt*)
6. Android versi 2.3 (*Gingerbread*)
7. Android versi 3.0/3.1 (*Honeycomb*)
8. Android versi 4.0 (*ICS: Ice Cream Sandwich*)
9. Android versi 4.1 (*Jelly Bean*)
10. Android versi 4.4 (*Kit kat*)
11. Android versi 5.0 (*Lollipop*)
12. Android versi 6.0 (*Marshmallow*)
13. Android versi 7.0 (*Nougat*)
14. Android versi 8.0 (*Oreo*)
15. Android versi 9.0 (*Pie*)
16. Android versi 10.0 (Q)
17. Android versi 11.0 (R)
18. Android versi 12.0 (S)

## 2.6 Arsitektur Android

Android menyediakan Arsitektur *development* yang kaya, arsitektur ini memberikan informasi tentang apa saja yang digunakan dalam sebuah sistem aplikasi. Arsitektur Android dapat dilihat pada Gambar 1 (Developer, 2017).



Gambar 1. Arsitektur Android (Developer, 2017).

### 2.6.1 *System Apps*

*System Apps* berada pada tingkat satu, bersama dengan aplikasi sistem inti untuk *e-mail* perpesanan SMS, kalender, penjelajahan internet, dan kontak.

### 2.6.2 Kerangka Kerja API Java

Semua fitur Android tersedia untuk developer melalui antarmuka pemrograman aplikasi (API) yang ditulis dalam bahasa Java. Berikut ini adalah API yang berguna untuk membuat aplikasi:

- a. Sistem Tampilan  
Sistem tampilan digunakan untuk membangun UI aplikasi, termasuk daftar, tombol dan menu
- b. Pengelola Referensi

Pengelola referensi digunakan untuk mengakses sumber daya *non-code* seperti *string*, grafik, dan *file layout* yang dilokalkan

c. Pengelola Notifikasi

Pengelola notifikasi digunakan untuk menampilkan peringatan khusus di bilah status

d. Pengelola Aktivitas

Pengelola aktivitas yang mengelola daur hidup aplikasi

e. Penyedia Materi

Penyedia materi yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain

f. Semua API kerangka kerja yang digunakan aplikasi sistem Android

### 2.6.3 Pustaka dan Waktu Proses Android

Setiap aplikasi berjalan dalam prosesnya sendiri dan dengan *instance Android runtime* sendiri, yang memungkinkan beberapa aplikasi berjalan dalam satu waktu pada perangkat bermemori rendah. Banyak layanan dan komponen sistem Android inti dibangun dari kode asli yang memerlukan pustaka asli yang ditulis dalam C dan C++. Pustaka asli tersebut tersedia untuk aplikasi *Java API Framework*.

### 2.6.4 Hardware Abstraction Layer

Lapisan ini menyediakan antarmuka yang menunjukkan kemampuan perangkat keras di perangkat ke *Java API Framework* yang lebih tinggi. *Hardware Abstraction Layer* terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau bluetooth.

### 2.6.5 Kernel Linux

Pondasi *platform* Android adalah *Kernel Linux*. Lapisan pada bagian atas mengandalkan *Kernel Linux* untuk fungsionalitas pokok seperti *threading* dan manajemen memori tingkat rendah. Menggunakan *Kernel Linux* memungkinkan Android memanfaatkan fitur keamanan utama dan memungkinkan produsen perangkat mengembangkan *driver* perangkat keras untuk *Kernel* yang cukup dikenal.

### 2.7 Android Activity Lifecycle

Setiap *user interface* diwakili oleh kelas *Activity* (*Activity class*), setiap *activity* mempunyai siklus yang dapat dilihat pada Tabel 2.1 berikut (Burnette, 2015).

Tabel 2.1 *Android Activity Lifecycle* (Burnette, 2015)

No	State	Penjelasan
1	<i>OnCreate (Bundle)</i>	<i>Method</i> ini dipanggil saat pertama kali aplikasi dijalankan. <i>Developer</i> dapat menggunakan ini untuk deklarasi variabel atau membuat <i>user interface</i> .
2	<i>OnStart ()</i>	<i>Method</i> ini mengindikasikan <i>activity</i> yang ditampilkan ke pengguna ( <i>user</i> ).
3	<i>OnResume ()</i>	<i>Method</i> ini dipanggil saat aplikasi mulai berinteraksi dengan pengguna. Di sini sangat cocok untuk meletakkan animasi ataupun musik.

Tabel 2.1 *Android Activity Lifecycle* (lanjutan)

No	State	Penjelasan
4	<i>OnPause ()</i>	<i>Method</i> ini dipanggil saat aplikasi yang berjalan kembali ke halaman sebelumnya atau biasanya karena ada <i>activity</i> baru yang dijalankan. <i>OnPause</i> cocok digunakan untuk meletakkan algoritma penyimpanan ( <i>save</i> ).
5	<i>OnStop ()</i>	<i>Method</i> ini dipanggil saat aplikasi yang berjalan di belakang layar dalam waktu cukup lama.
6	<i>OnRestart ()</i>	<i>Method</i> terjadi saat <i>Activity</i> kembali menampilkan <i>user interface</i> setelah status <i>stop</i> .
7	<i>OnDestroy ()</i>	Dipanggil saat aplikasi berhenti.
8	<i>OnSaveInstanceState (Bundle)</i>	metode ini mengijinkan <i>activity</i> untuk menyimpan setiap status <i>instance</i> . Misalnya dalam mengedit teks, kursor bergerak dari kiri ke kanan.
9	<i>OnRestoreInstanceState (Bundle)</i>	Dipanggil saat <i>activity</i> kembali menginisialisasi dari status sebelumnya yang disimpan oleh <i>onSaveInstanceState (Bundle)</i>

## 2.8 Android Studio

Android studio adalah IDE Google untuk aplikasi Android. Android studio menyediakan *editor* kode tingkat lanjut dan serangkaian template aplikasi. Selain itu, Android studio juga memiliki alat untuk *development*, *men-debug*, menguji dan meningkatkan kinerja yang membuat pengembangan aplikasi lebih

cepat dan mudah. Selain merupakan editor kode IntelliJ dan alat pengembang yang berdaya guna, Android studio menawarkan fitur lebih banyak untuk meningkatkan produktivitas saat membuat aplikasi Android, misalnya (Developers, 2022):

- a. Sistem versi berbasis *Gradle* yang *fleksibel*.
- b. *Emulator* yang cepat dan kaya fitur.
- c. Lingkungan yang menyatu untuk pengembangan bagi semua perangkat Android.
- d. *Instant run* untuk mendorong perubahan ke aplikasi yang berjalan tanpa membuat APK baru.
- e. Template kode dan integrasi GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode contoh.
- f. Alat pengujian dan kerangka kerja yang ekstensif.
- g. Alat lain untuk meningkatkan kinerja, kegunaan, kompatibilitas versi, dan masalah-masalah lain.
- h. Dukungan C++ dan NDK.
- i. Dukungan bawaan untuk *Google Cloud Platform*, mempermudah pengintegrasian Google Cloud Messaging dan App Engine.

## **2.9 Instant Messaging**

*Instant messaging* merupakan fasilitas komunikasi *chatting* untuk para pengguna internet. *Instant messaging* memudahkan *user* agar dapat berkomunikasi dengan cara mengirimkan pesan berupa *text* dengan *user* lain. Selain itu, *instant messaging* juga berfungsi untuk tukar menukar *file* secara *peer to peer* (Zuliarso dan Februariyanti, 2013).

## 2.10 Kotlin

Kotlin adalah sebuah bahasa pemrograman dengan pengetikan statis yang berjalan pada mesin virtual java ataupun menggunakan kompiler LLVM. Kotlin dapat juga dikompilasikan ke dalam bentuk kode sumber *JavaScript*.

Pengembang utama kotlin berasal dari tim *programmer* JetBrains yang bermarkas di Saint Petersburg, Rusia. JetBrains terkenal dengan produk IntelliJ IDEA, sebuah *platform* IDE yang sangat *powerful* dan banyak digunakan oleh *developer java*. Android studio merupakan sebuah IDE resmi serta khusus dibuat untuk *Android development*. Pembuatan IDE ini berdasarkan IntelliJ IDEA.

Kotlin didesain berdasarkan pemahaman pemrograman dari *java developer* serta IntelliJ sebagai IDE utamanya. Tim Pengembang memutuskan untuk memberi nama “Kotlin”, yakni nama sebuah pulau di Rusia, sebagaimana “Java” yang berasal dari nama pulau Jawa (atau “Java” dalam bahasa Inggris) di Indonesia.

Pada Google I/O 2017 Kotlin resmi diumumkan sebagai bahasa pemrograman yang dapat digunakan untuk membuat aplikasi Android. Google menetapkan Kotlin sebagai bahasa kelas satu bagi Android. Maka bersama Java dan C++, Kotlin merupakan bahasa resmi untuk pengembangan aplikasi-aplikasi Android (Rohman dan Rendra, 2019).

## 2.11 Firebase

*Android firebase* adalah *cloud service provider* dan *backend as service* yang dimiliki oleh google. *Firebase* dapat digunakan untuk membangun aplikasi Android, iOS dan aplikasi berbasis Web. *Firebase* memiliki banyak SDK yang

memungkinkan untuk mengintegrasikan layanan antara Android, iOS, Javascript, C++ hingga Unity (Stonehem, 2016).

*Firestore* memiliki tujuh teknologi yang dibangun untuk meningkatkan pengembangan aplikasi seperti *Firestore Authentication*, *The Realtime Database*, *Cloud Storage for Firestore*, *Firestore Hosting*, *Firestore Test Lab*, *Firestore Crash Reporting* dan *Cloud Functions for Firestore* (Moroney, 2017).

### **2.12 XML (*Extensible Markup Language*)**

XML merupakan dasar terbentuknya *web service* yang digunakan untuk mendeskripsikan data. XML membentuk *web service* secara keseluruhan pada level paling detail. Fungsi utama dari XML adalah komunikasi antar aplikasi, dan integrasi data. Dengan standarisasi XML, aplikasi-aplikasi yang berbeda dapat dengan mudah berkomunikasi antara satu dengan yang lain (Deviana, 2011).

### **2.13 *Unified Modeling Language***






UML merupakan singkatan dari “*Unified Modeling Language*” yaitu suatu metode pemodelan secara visual untuk sarana perancangan sistem berorientasi objek, atau definisi UML adalah sebagai bahasa yang sudah menjadi standar pada visualisasi, perancangan dan juga pendokumentasian sistem. Saat ini UML sudah menjadi bahasa standar dalam penulisan *blue print software* (Al Fatta, 2007).



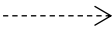
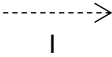
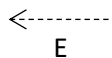



### 2.13.1 Use case Diagram

*Use case diagram* adalah salah satu jenis diagram pada UML yang menggambarkan interaksi antara sistem dan aktor. *Use case diagram* dapat mendeskripsikan tipe interaksi antara si pemakai sistem dengan sistemnya. Simbol *Use case diagram* dapat dilihat pada Tabel 2.2 berikut (Al Fatta, 2007).

Tabel 2.2 Simbol *Use case Diagram* (Al Fatta, 2007)

Gambar	Nama	Keterangan
	<i>Use case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Actor</i>	Menjelaskan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Association</i>	Menghubungkan antara objek satu dengan objek lainnya.
	<i>Directed Association</i>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, <i>directed association</i> biasanya juga disertai dengan <i>multiplicity</i> .
	<i>Generalization</i>	Hubungan antara objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ) (umum-khusus).





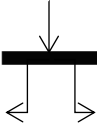
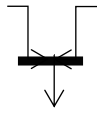

Tabel 2.2 Simbol *Use case Diagram* (lanjutan)

Gambar	Nama	Keterangan
	<i>Dependency</i>	Hubungan saat perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri ( <i>independent</i> ).
	<i>Include</i>	Menjelaskan bahwa <i>Use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menjelaskan bahwa <i>Use case</i> target memperluas perilaku dari <i>Use case</i> sumber pada suatu titik yang diberikan.
	<i>System</i>	Menjelaskan paket yang menampilkan sistem secara terbatas.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

### 2.13.2 Activity Diagram

*Activity diagram* atau diagram aktivitas yaitu salah satu jenis diagram pada UML yang dapat memodelkan proses-proses apa saja yang terjadi pada sistem. Simbol *activity diagram* dapat dilihat pada Tabel 2.3 berikut (Al Fatta, 2007).


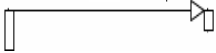


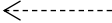


Tabel 2.3 Simbol *Activity Diagram* (Al Fatta, 2007)

Gambar	Nama	Keterangan
	<i>Activity</i>	Menandakan sebuah aktivitas. Bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<i>Action</i>	<i>State</i> dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<i>Initial Node</i>	Titik awal, untuk memulai suatu aktivitas.
	<i>Activity Final Node</i>	Objek dibentuk dan diselesaikan atau di akhiri.
	<i>Fork</i>	Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran.
	<i>Join</i>	Beberapa aliran yang pada tahap tertentu berubah menjadi satu aliran.
	<i>Decision</i>	Pilihan untuk mengambil keputusan.

### 2.13.3 Sequence diagram

*Sequence diagram* yaitu salah satu jenis pada UML yang menjelaskan interaksi objek yang berdasarkan urutan waktu, *sequence diagram* juga dapat menggambarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu *Use case diagram*. Simbol Sequence diagram dapat dilihat pada Tabel 2.4 berikut (Al Fatta, 2007).

Tabel 2.4 Simbol *Sequence Diagram* (Al Fatta, 2007)

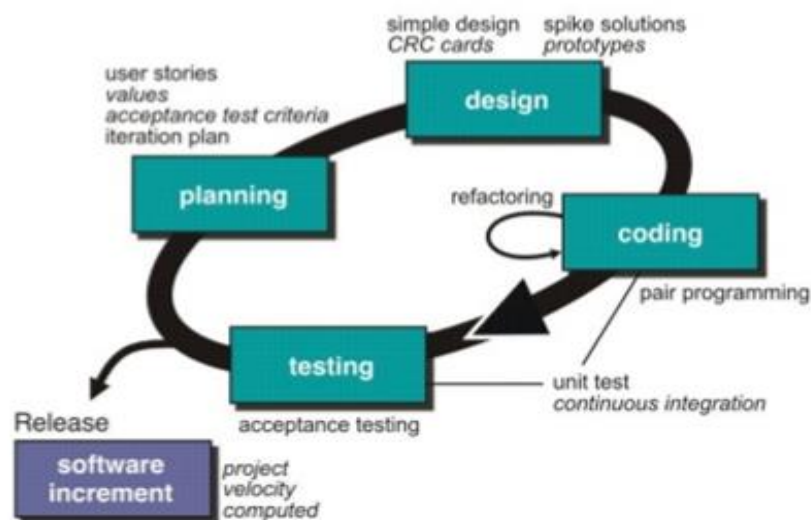
Gambar	Nama	Keterangan
	<i>Lifeline</i>	Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktivitas yang terjadi
	<i>Self Message</i>	<i>Message</i> yang diberikan oleh sebuah objek kepada dirinya sendiri.
	<i>Async Message</i>	Menghubungkan dua objek, pesan asinkron tidak perlu balasan agar interaksi berlanjut.
	<i>Replay Message</i>	<i>Message</i> yang diberikan oleh sebuah objek untuk membalas <i>message</i> objek lain.
	<i>Create Message</i>	<i>Message</i> yang diberikan oleh sebuah objek untuk membuat objek baru.
	<i>Delete Message</i>	<i>Message</i> yang diberikan oleh sebuah objek untuk menghapus objek.

## 2.14 *Extreme Programming (XP)*

XP merupakan metodologi rekayasa perangkat lunak yang digunakan untuk mengembangkan proyek dengan adaptasi cepat dalam perubahan-perubahan yang terjadi selama pengembangan sebuah aplikasi. XP diperkenalkan oleh Kent Beck pada tahun 1996 dengan tujuan untuk menghasilkan perangkat lunak yang berkualitas tinggi dan lebih produktif. Xp juga bertujuan untuk mengurangi biaya perubahan selama pengembangan aplikasi dengan menggunakan siklus yang singkat (Suryantara, 2017).

### 2.14.1 Kerangka Kerja *Extreme Programming*

Para pengembang perangkat lunak menggunakan metodologi XP untuk mengembangkan perangkat lunak dengan cepat. Tahapan pengembangan perangkat lunak dengan XP meliputi *planning*, *design*, *coding*, *testing* dan *release*. Kerangka kerja dari XP dapat dilihat pada Gambar 2.2 berikut.



Gambar 2.2 Kerangka Kerja XP (Suryantara, 2017).

#### 2.14.2 *Planning*

Tahap ini dimulai dengan pemahaman konteks bisnis dari aplikasi, mendefinisikan keluaran(*output*) fitur yang ada pada aplikasi, fungsi dari aplikasi yang dikembangkan, penentuan waktu dan biaya pengembangan aplikasi, serta alur pengembangan aplikasi.

#### 2.14.3 *Design*

Tahap desain menekankan pada desain aplikasi secara sederhana berdasarkan *planning*

#### 2.14.4 *Coding*

Tahap *coding* adalah tahapan pengembangan sebuah *design* aplikasi menjadi code program

#### 2.14.5 *Testing*

Tahapan *Testing* memfokuskan pada pengujian fitur-fitur yang ada ada aplikasi sehingga tidak ada kesalahan (*error*) dan aplikasi yang dibuat sesuai dengan proses bisnis pada *client* (pelanggan).

#### 2.14.6 Release

Tahapan rilis adalah tahap akhir dan jembatan menuju tahapan *planning* dari metodologi XP. Pada tahapan ini sistem akan dipublikasikan dan mendapatkan *feedback* dari user. jika sistem belum dianggap sesuai dengan kebutuhan pengguna maka akan dilakukan proses *software increment* dan memasuki tahapan *planning*. Apabila sistem sudah sesuai dengan kebutuhan pengguna maka tahapan pengembangan sistem akan berakhir

#### 2.1.5 Black-box Testing

*Black-box Testing* berfokus pada spesifikasi fungsional dari perangkat lunak. *Tester* dapat mendefinisikan kumpulan kondisi *input* dan melakukan pengetesan pada spesifikasi fungsional program. *Black-box Testing* bukanlah solusi alternatif dari *White-Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White-Box Testing* (Nidhra dan Dondeti, 2012).

### **III. METODOLOGI PENELITIAN**

#### **3.1. Tempat dan Waktu**

Penelitian ini dilakukan di Program Studi Ilmu Komputer, Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung. Penelitian dimulai pada bulan November tahun 2018 sampai dengan bulan Januari tahun 2022.

#### **3.2. Alat dan Bahan**

Alat dan bahan merupakan perangkat untuk membangun aplikasi ini. Alat meliputi perangkat keras (*hardware*). Bahan meliputi perangkat lunak (*software*).

##### **3.2.1 Alat**

Alat yang digunakan dalam pengembangan aplikasi ODOJ ini berupa Laptop dan *gadget* dengan spesifikasi sebagai berikut:



### 3.2.1.1. Laptop

Laptop yang digunakan memiliki spesifikasi sebagai berikut:

- a. *Processor Intel Skylake Core i5-7200U @ 2.50 GHz*
- b. *DDR 4 RAM 8192 MB*
- c. *Penyimpanan 1 TB HDD*

### 3.2.1.2. Smartphone

*Smartphone* yang digunakan memiliki spesifikasi sebagai berikut:

- a. *Sistem operasi Android 7.1.2 (nougat)*
- b. *CPU Octa-core 1.4 Ghz Cortex-A53*
- c. *Layar 5.0 inches, 720 x 1280 pixels (~296 ppi pixel density)*
- d. *Sensor Fingerprint, accelerometer, gyro, proximity, compass*
- e. *RAM 2GB*
- f. *Memory 16 GB*
- g. *Konektivitas Wi-Fi 802.11 GPS (A-GPS, GLONASS, BDS), MicroUSB 2.0 dan Internet (HSPA, LTE)*

### 3.2.2 Perangkat Lunak

Perangkat Lunak yang digunakan dalam pengembangan aplikasi ODOJ ini adalah:

- a. *Sistem operasi Windows 10 home single language 64-bit (10.0, build 17134)*
- b. *Android Studio 3.4.1 win64-bit*
- c. *Nox Emulator*
- d. *StarUML*

- e. *Corel draw*
- f. *Adobe Experience Design*

### **3.3. Metode Penelitian**

Dalam metode penelitian ini akan dilakukan metode pengumpulan data menggunakan metode observasi, metode wawancara dan studi literatur. Sedangkan metode pengembangan sistem yang digunakan adalah metode *Extreme Programming*.

#### **3.3.1 Metode Pengumpulan Data**

Untuk memperoleh data yang sesuai dengan permasalahan dalam kegiatan ODOJ, maka dalam penelitian ini peneliti menggunakan metode pengumpulan data sebagai berikut:

a. Metode Observasi

Teknik observasi dilakukan dengan mengikuti langsung program *one day one juz* dan melakukan pengamatan terhadap proses-proses yang sedang berjalan.

b. Metode Wawancara

Metode wawancara digunakan untuk mendapatkan informasi dengan mewawancarai langsung ketua bidang koordinator admin ODOJ pusat.

c. Studi Literatur

Studi literatur yang dilakukan dengan cara pencarian informasi melalui media internet tentang bahan-bahan yang berhubungan dengan masalah yang dibahas.

### 3.3.2 Sumber Data

Sumber data yang digunakan dalam penelitian ini yaitu data primer dan sekunder:

- a. Sumber data primer dalam penelitian ini adalah Ketua Departemen Komunikasi Internal ODOJ pusat Tony Dwi Prasetyo.
- b. Sumber data sekunder kegiatan ini adalah situs internet ODOJ dan juga situs internet yang relevan.

### 3.2.3 Metode Pengembangan Sistem

Pengembangan sistem yang dilakukan pada penelitian ini menggunakan metode *eXtreme Programming* (XP). Tahap pengembangan sistem yang dilakukan dalam pengembangan aplikasi *one day one juz* dalam penelitian ini sebagai berikut:

- a. *Planning*

Pada tahap *planning* dilakukan pengumpulan data serta mendefinisikan apa yang dibutuhkan oleh sistem. Pengumpulan data dalam tahap ini dilakukan dengan wawancara, observasi dan studi literatur.

- b. *Design*

Tahapan ini adalah tahap pembuatan desain dari sebuah sistem. Desain yang digunakan dalam penelitian ini menggunakan model UML dan juga perancangan antarmuka. Desain UML yang dibuat berupa *Use case diagram*, *activity diagram*, dan *sequence diagram*. Setelah desain UML terbuat maka dibangunlah perancangan antarmuka dari sistem. Desain yang dibuat merupakan representasi dari sistem yang dikembangkan.

c. *Coding*

Tahap *coding* adalah Tahapan keseluruhan desain diubah menjadi kode-kode program. Kode program akan diimplementasikan ke dalam desain sistem, sehingga nantinya desain dari aplikasi ODOJ dapat berjalan dengan lancar dan benar. Bahasa pemrograman yang digunakan pada tahap implementasi pengembangan aplikasi ODOJ adalah Kotlin dan XML. Bahasa pemrograman kotlin digunakan untuk fungsionalitas dari aplikasi, sedangkan XML digunakan untuk *interface* dari aplikasi ODOJ.

d. *Testing*

Setelah semua *Requirement Analysis* dan *System Design* selesai diimplementasikan maka pengujian dapat segera dimulai. Tahap pengujian aplikasi ODOJ dilakukan dengan melihat kesesuaian *output*, *input* dan juga fungsionalitas dari aplikasi. Pengujian yang dilakukan menggunakan *Black-box testing*.

e. *Release*

Setelah tahap pengujian selesai maka aplikasi ODOJ dapat dipublikasikan sehingga mendapatkan *feedback* dari pada “Odojer” yang nantinya dapat digunakan untuk proses *software increment*.

### 3.4. Analisis

Pada tahap ini akan dilakukan analisis sistem dan analisis *user requirement* untuk mempermudah proses pengembangan sistem.

### 3.3.1 Analisis Sistem

Berdasarkan latar belakang yang telah dijelaskan maka sistem yang dibutuhkan adalah sebagai berikut.

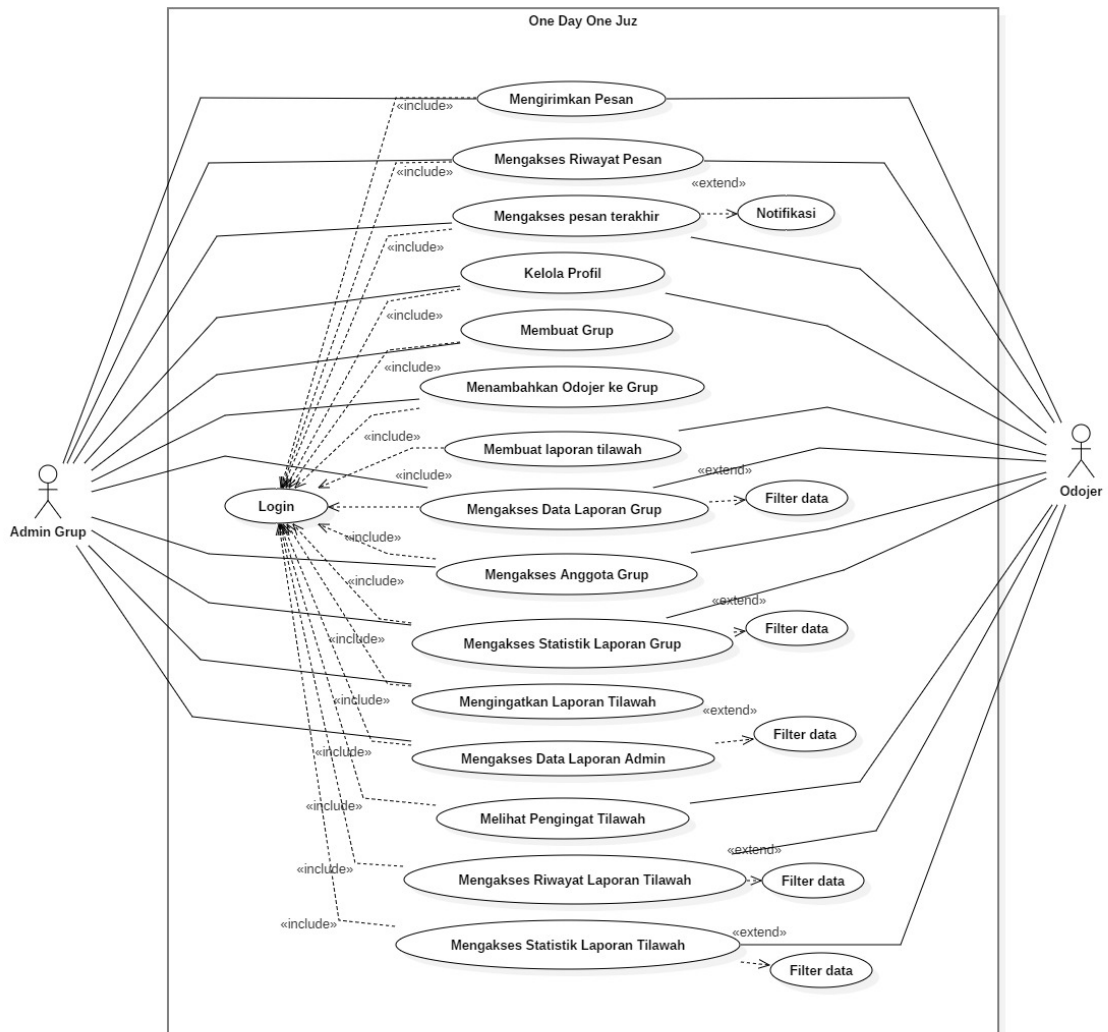
- a. Sistem menggunakan perangkat *smartphone*.
- b. Sistem membutuhkan akses internet.
- c. Sistem membutuhkan database penyimpanan
- d. Sistem membutuhkan *Google mobile service*
- e. Sistem memiliki dua aplikasi yaitu aplikasi admin dan aplikasi “Odojer”.

### 3.3.2 User Requirement

Kebutuhan dasar aplikasi ini adalah sebagai berikut

- a. Sistem dapat memisahkan antara *user* laki laki dan *user* perempuan.
- b. Sistem dapat menerima dan mengirim pesan.
- c. Sistem dari admin dapat menambahkan “Odojer” ke dalam grup.
- d. Sistem dari admin dapat membuat grup ODOJ.
- e. Sistem dapat mengubah profil.
- f. Sistem dapat melakukan “lapor tilawah”.
- g. Sistem dapat melakukan rekapan tilawah setiap harinya.

Kebutuhan dasar tersebut dapat dijelaskan oleh *Use Case Diagram* yang menggambarkan fungsionalitas dari aplikasi. *Use case diagram* dari aplikasi ODOJ ditunjukkan pada Gambar 3.1.



Gambar 3.1 Use case Diagram Aplikasi ODOJ.

### 3.5. Perancangan Sistem

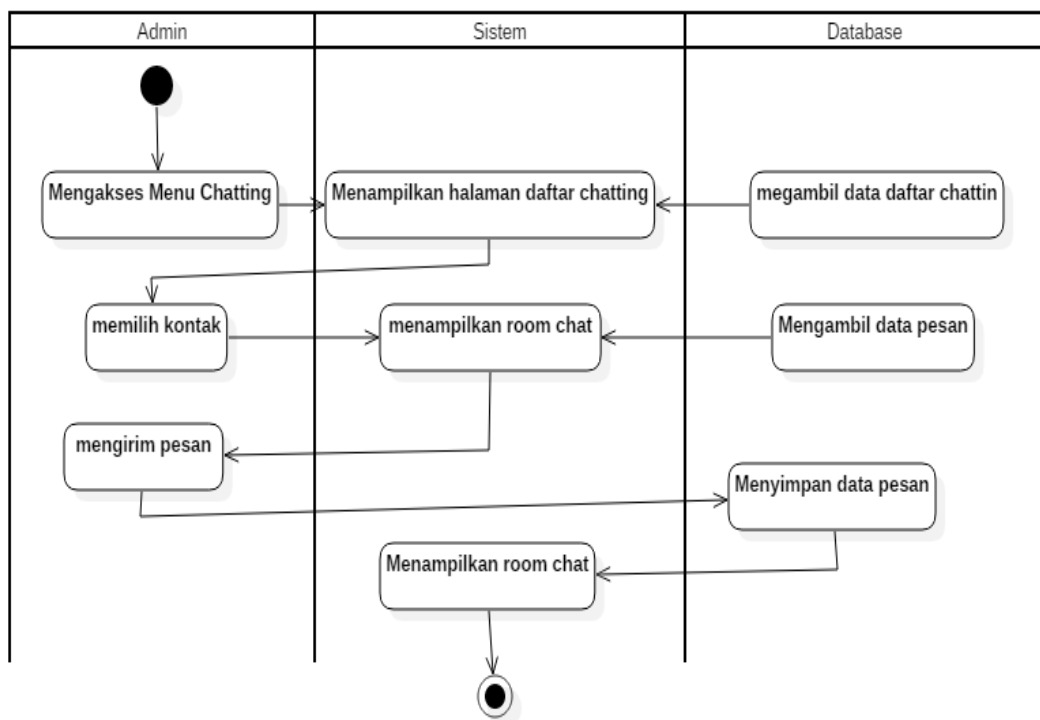
Perancangan sistem yang dilakukan pada penelitian ini menggunakan UML yang meliputi *Activity diagram*, dan *sequence diagram*. Kemudian UML tersebut diubah menjadi *design layout* atau *user interface* dari sistem.

### 3.5.1. Activity Diagram

*Activity diagram* menggambarkan proses-proses yang terjadi dari dimulainya aktivitas hingga aktivitas berhenti. *Activity diagram* dari Aplikasi ODOJ sebagai berikut:

#### 3.5.2.1. Activity Diagram Mengirim Pesan

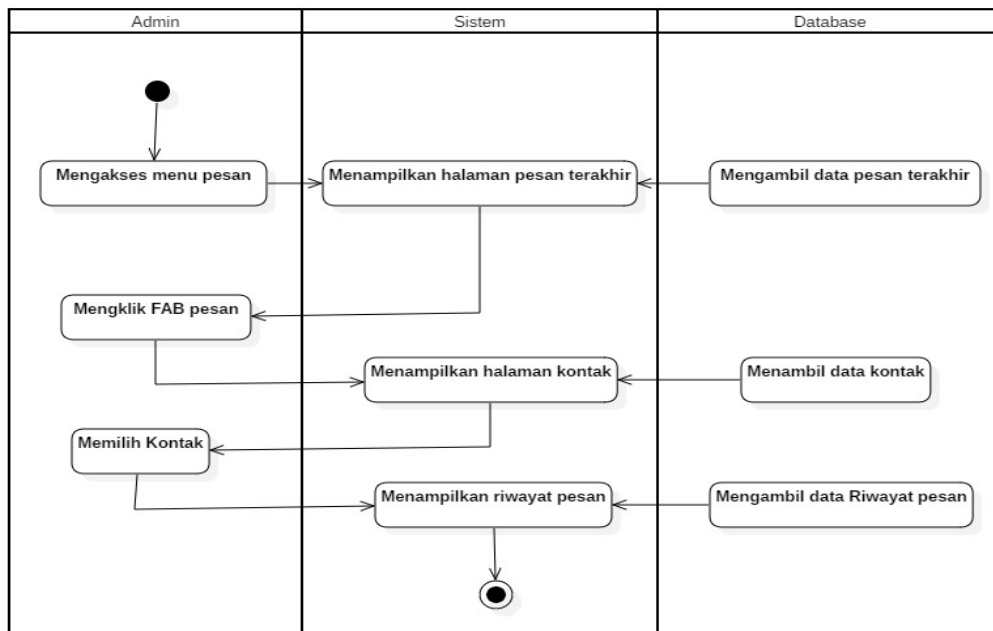
*Activity* mengirim pesan menunjukkan alur aktivitas yang dilakukan *user* dan admin untuk mengirim pesan ke sesama anggota “Odojer” dan menyimpannya ke dalam *database* sistem. *Activity diagram* mengirim pesan dapat dilihat pada Gambar 3.2



Gambar 3.2 Activity Aplikasi ODOJ.

### 3.5.2.2. Activity Diagram Mengakses Riwayat Pesan

Activity riwayat pesan menunjukkan alur aktivitas yang dilakukan *user* maupun admin untuk mengakses riwayat pesan yang telah dikirim dan diterima oleh anggota “Odojer”. Activity diagram riwayat pesan dapat dilihat pada Gambar 3.3.

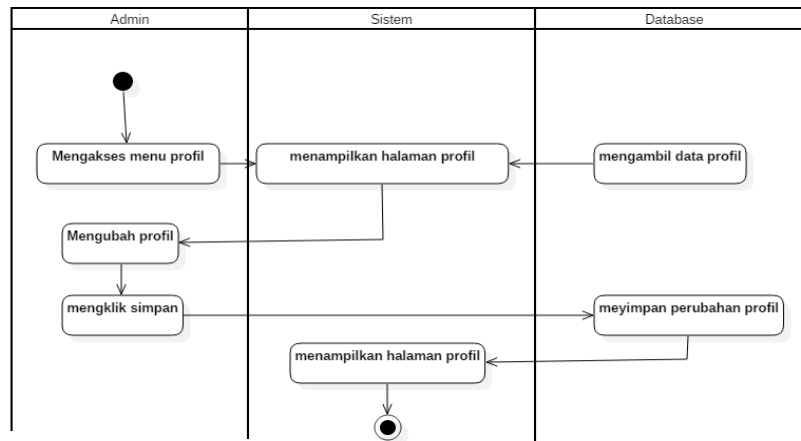


Gambar 3.3 Activity Diagram Menerima Pesan.

### 3.5.2.3. Activity Diagram Kelola Profil

Activity kelola profil menunjukkan alur aktivitas yang dilakukan *user* maupun admin untuk melihat dan mengubah profil kemudian menyimpannya kembali pada *database*. Activity diagram kelola profi dapat dilihat pada Gambar 3.4.

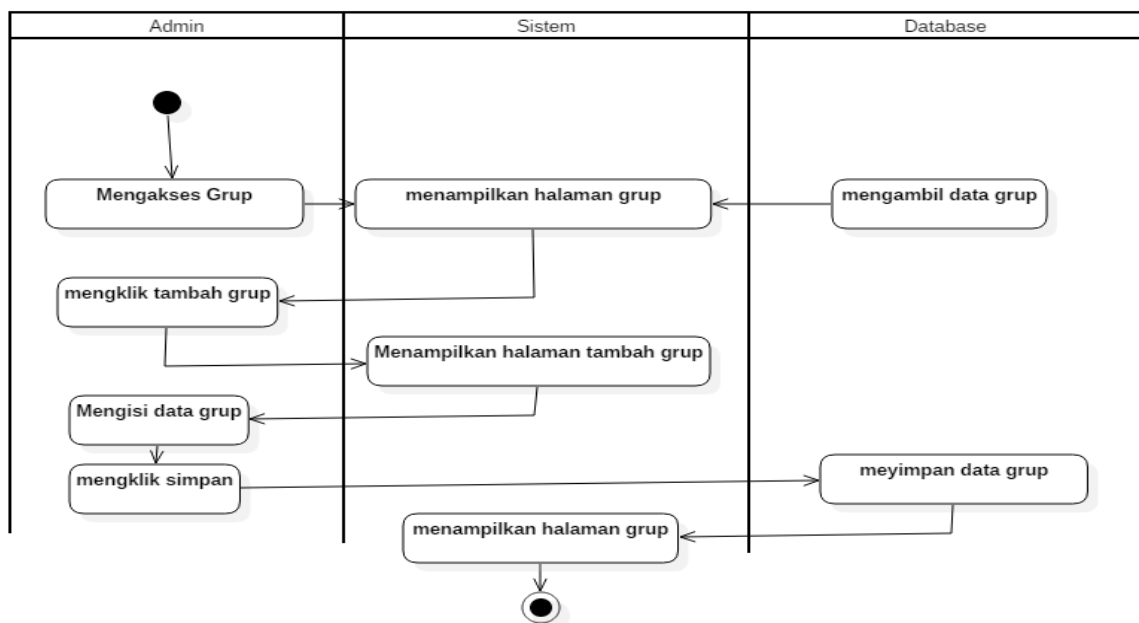




Gambar 3.4 Activity Diagram Mengubah Profil.

#### 3.5.2.4. Activity Diagram Membuat Grup

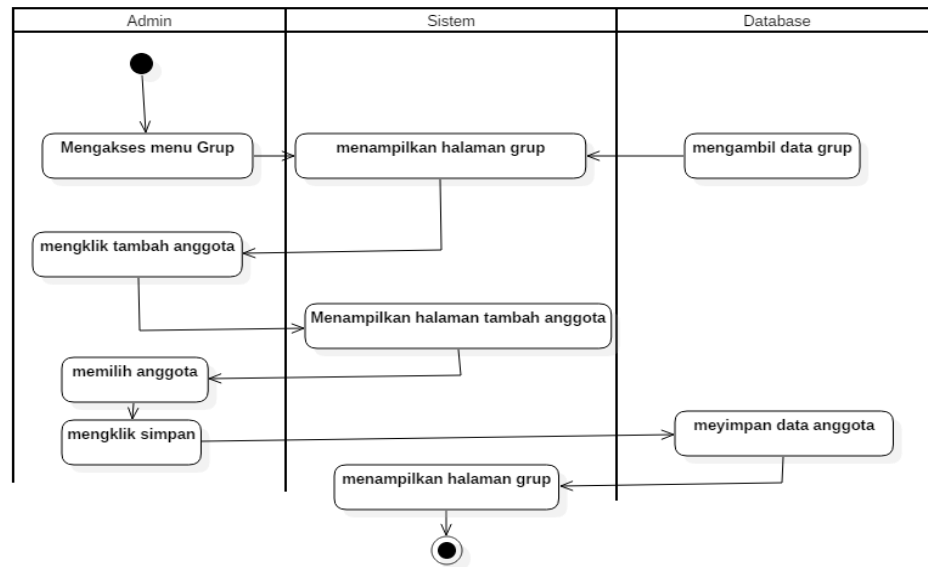
Activity membuat grup menunjukkan alur aktivitas yang dilakukan admin untuk membuat sebuah grup ODOJ. Activity diagram membuat grup dapat dilihat pada Gambar 3.5.



Gambar 3.5 Activity Diagram Membuat Grup.

### 3.5.2.5. Activity Diagram Menambahkan “Odojer” ke Grup

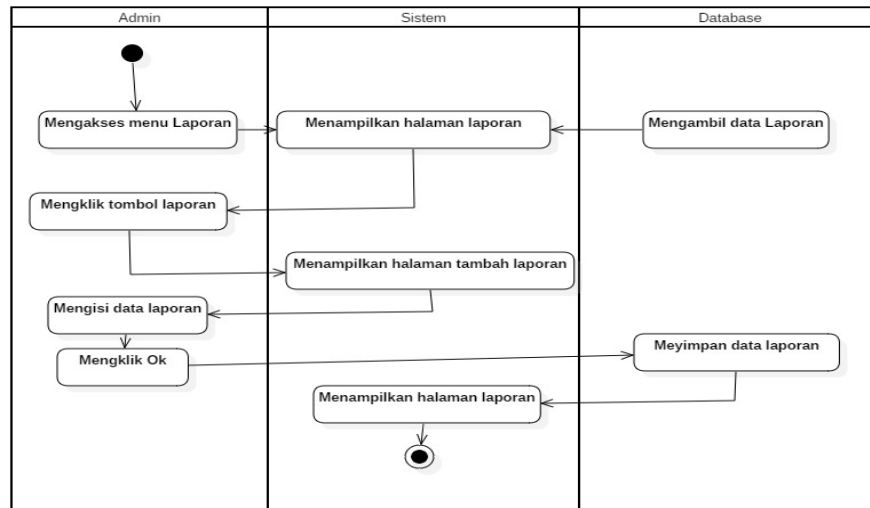
Activity ini menunjukkan alur aktivitas yang dilakukan admin untuk menambahkan “Odojer” lain yang belum memiliki grup dalam aplikasi. Activity diagram menambahkan “Odojer” ke grup dapat dilihat pada Gambar 3.6.



Gambar 3.6 Activity Diagram Menambahkan “Odojer” ke dalam Grup.

### 3.5.2.6. Activity Diagram Menambahkan Laporan

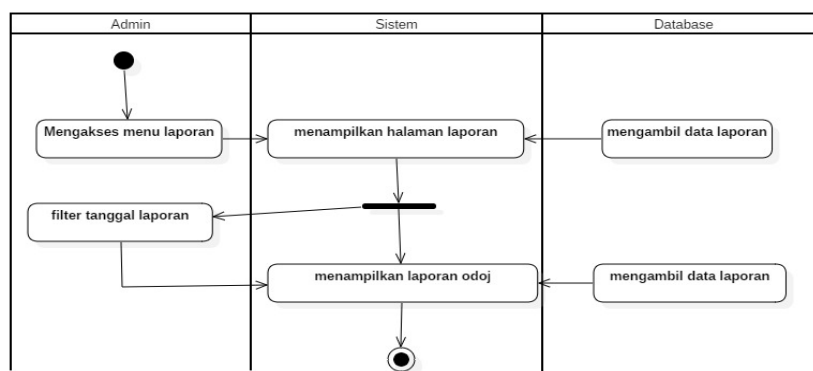
Activity menambahkan laporan menunjukkan alur aktivitas yang dilakukan *user* untuk melaporkan hasil tilawahnya setiap hari ke dalam halaman laporan yang dapat diakses oleh admin dan “Odojer”. Activity diagram menambahkan laporan dapat dilihat pada Gambar 3.7.



Gambar 3.7 Activity Diagram Menambahkan Laporan.

### 3.5.2.7. Activity Diagram Mengakses Data Laporan

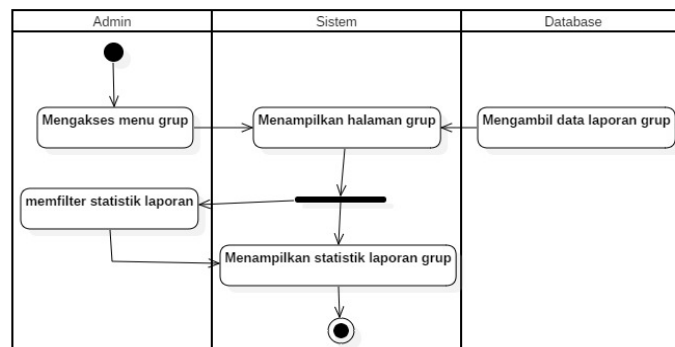
Activity ini menunjukkan alur aktivitas yang dilakukan user dan admin untuk melihat laporan tilawah semua user dalam satu grup setiap harinya. Activity diagram melihat laporan dapat dilihat pada Gambar 3.8.



Gambar 3.8 Activity Diagram Melihat Laporan.

### 3.5.2.8. Activity Diagram Mengakses Statistik Laporan Grup

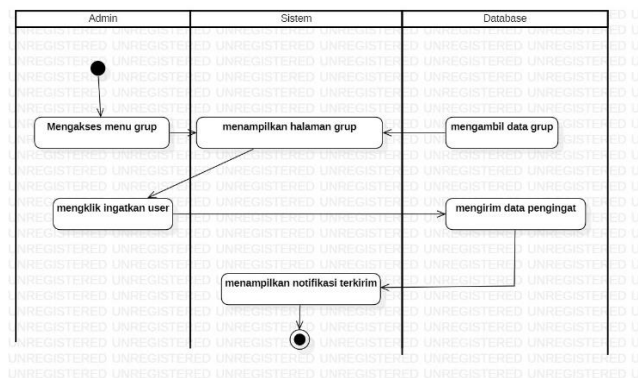
Activity mengakses statistik laporan grup menunjukkan alur aktivitas untuk melihat statistik laporan tilawah semua user dalam satu grup, Activity diagram mengakses statistik laporan grup Gambar 3.9.



Gambar 3.9 Activity Diagram Mengakses Statistik Laporan Grup.

### 3.5.2.9. Activity Diagram Mengingatkan Laporan Tilawah

Activity ini menunjukkan alur aktivitas untuk memberikan pengingat tilawah kepada semua anggota grup, Activity diagram mengingatkan laporan tilawah dapat dilihat pada Gambar 3.10.



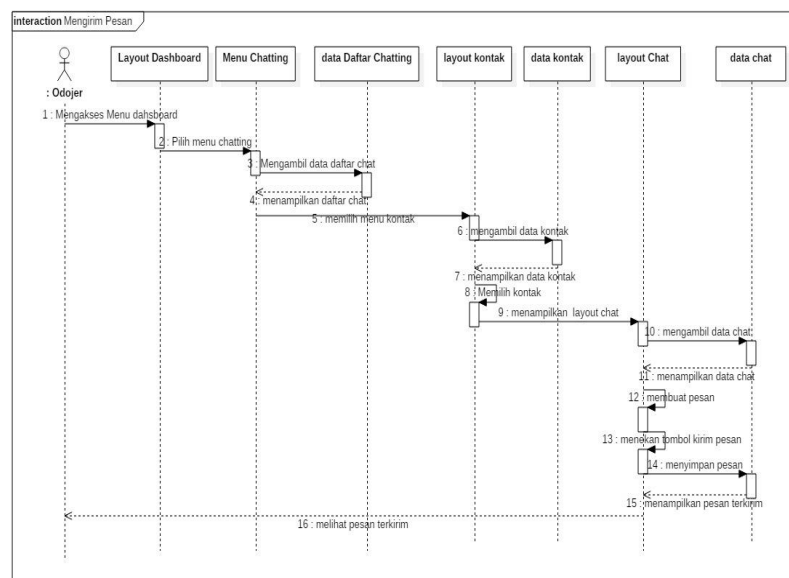
Gambar 3.10 *Activity Diagram* Mengingat Laporan Tilawah

### 3.5.2. *Sequence Diagram*

*Sequence diagram* digunakan untuk menggambarkan alur fungsionalitas dari *Use case diagram*. *Sequence diagram* dari aplikasi ODOJ adalah sebagai berikut.

#### 3.5.3.1. *Sequence Diagram* Mengirim Pesan.

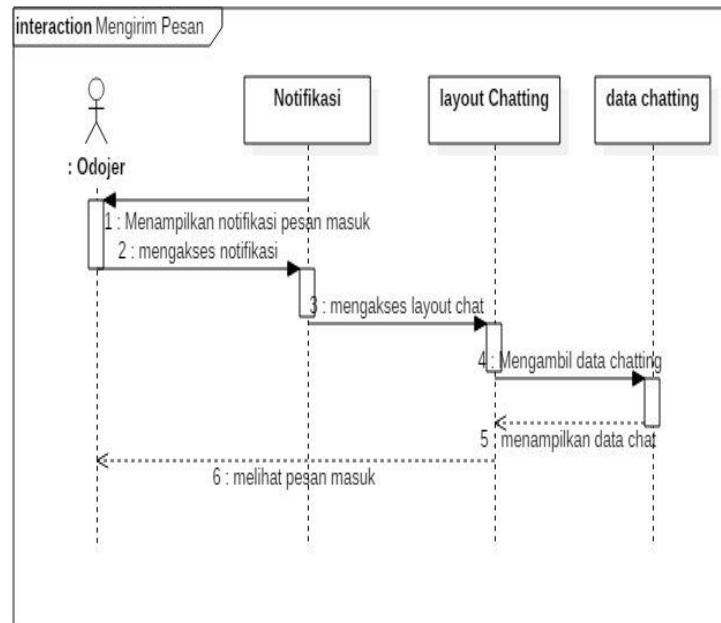
*Sequence diagram* mengirim pesan dapat dilihat pada Gambar 3.11



Gambar 3.11 *Sequence Diagram* Mengirim Pesan.

#### 3.5.3.2. *Sequence Diagram* Menerima Pesan.

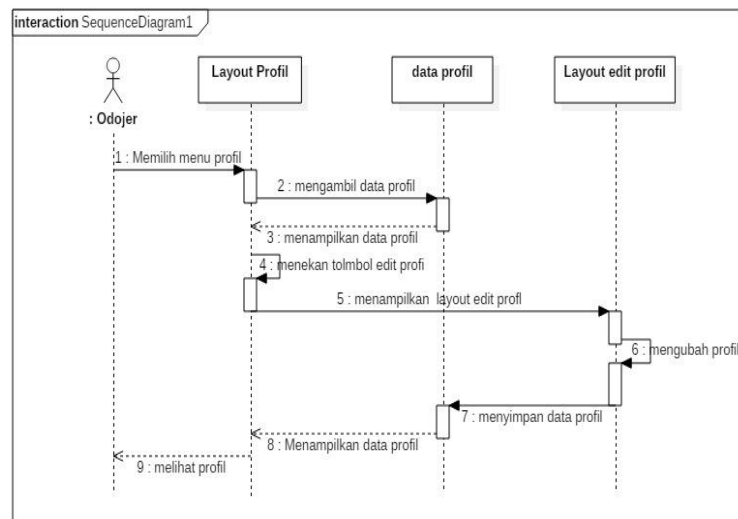
*Sequence diagram* menerima pesan dapat dilihat pada Gambar 3.12



Gambar 3.12 *Sequence Diagram* Menerima Pesan.

### 3.5.3.3. *Sequence Diagram* Mengubah Profil.

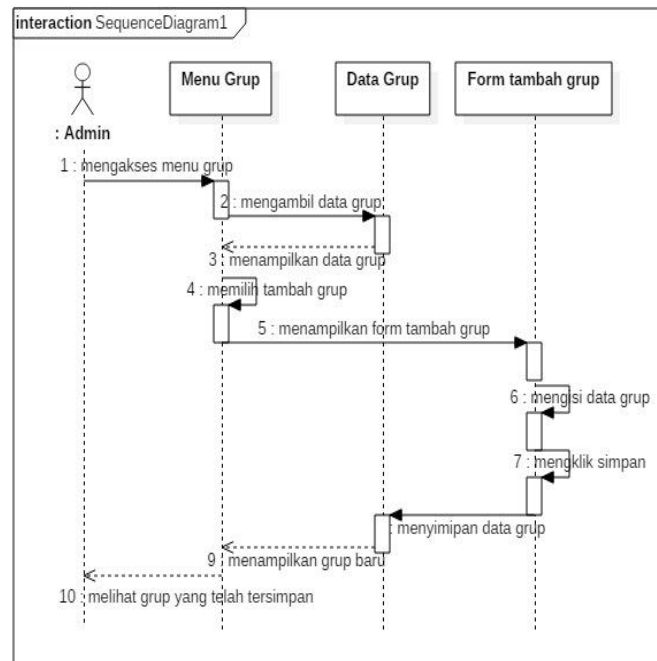
*Sequence diagram* mengubah profil dapat dilihat pada Gambar 3.13 berikut.



Gambar 3.13 *Sequence Diagram* Mengubah Profil.

### 3.5.3.4. *Sequence Diagram* Membuat Grup.

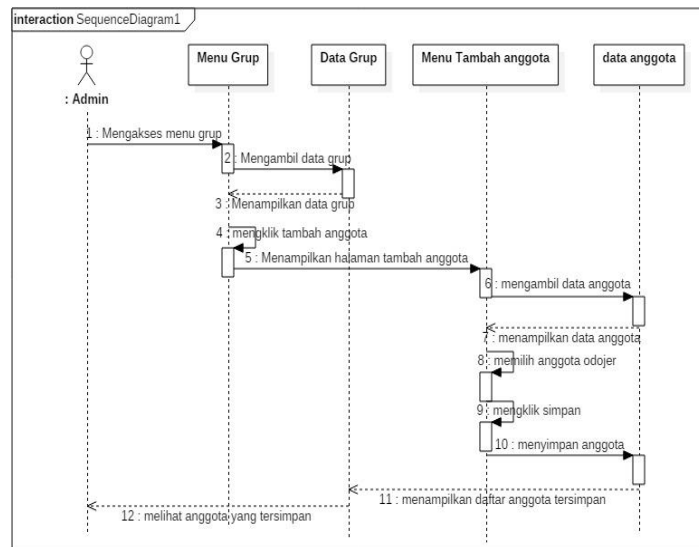
*Sequence diagram* membuat grup dapat dilihat pada Gambar 3.14.



Gambar 3.14 *Sequence Diagram* Membuat Grup.

### 3.5.3.5. *Sequence Diagram* Memasukkan “Odojer” ke dalam Grup.

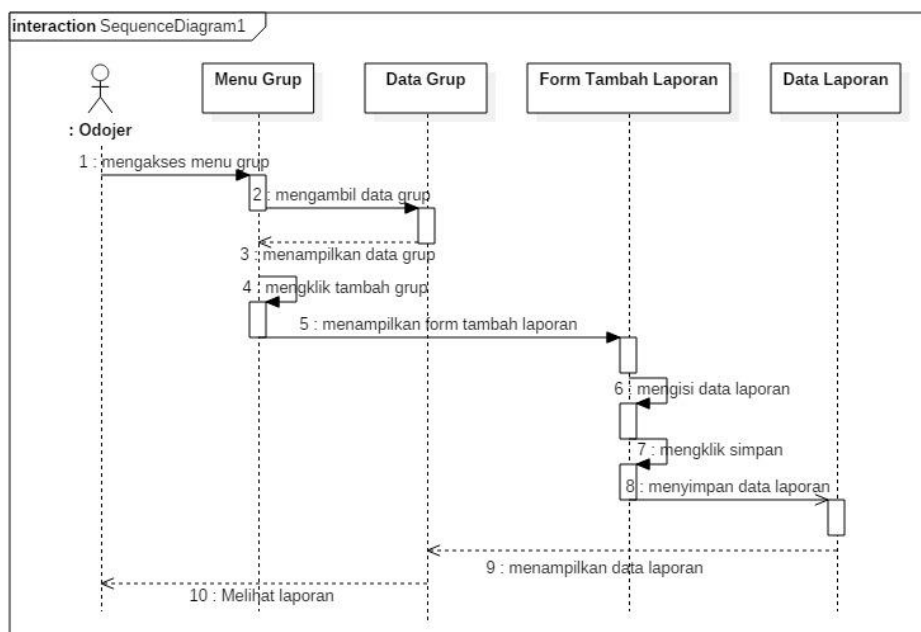
*Sequence diagram* memasukkan “Odojer” ke grup dapat dilihat pada Gambar 3.15.



Gambar 3.15 *Sequence Diagram* Memasukkan “Odojer” ke dalam grup.

### 3.5.3.6. *Sequence Diagram* Menambahkan Laporan.

*Sequence diagram* menambahkan laporan dapat dilihat pada Gambar 3.16 berikut.

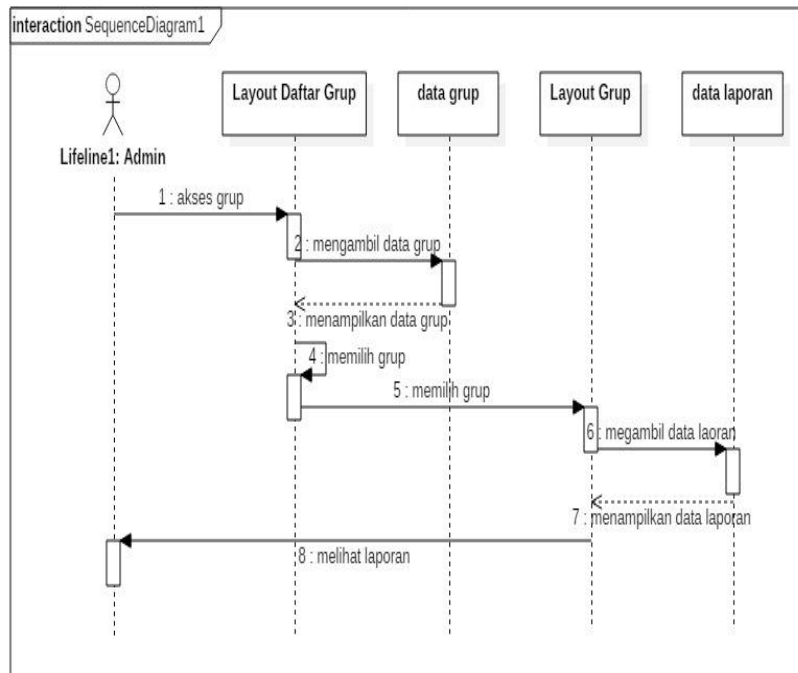


Gambar 3.16 *Sequence Diagram* Menambahkan Laporan.



### 3.5.3.7. Sequence Diagram Melihat Laporan.

Sequence diagram melihat laporan dapat dilihat pada Gambar 3.17.



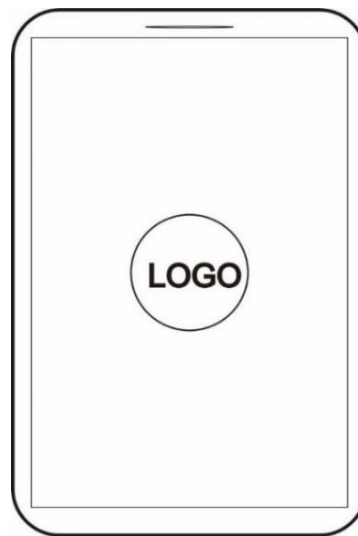
Gambar 3.17 Sequence Diagram Melihat Laporan.

### 3.6. Perancangan Antarmuka (*Design Interface*)

Perancangan antarmuka merupakan gambaran dari *interface* sebuah sistem yang akan dibentuk. Perancangan antarmuka disesuaikan dengan kebutuhan dari *user* terhadap sistem. Layout yang dibuat dalam pengembangan “ODOJ” ini adalah sebagai berikut.

### 3.6.1. *Layout Splash Screen*

*Layout* ini digunakan sebagai tampilan awal saat pertama kali menggunakan aplikasi ODOJ. *Layout Splash screen* berisi logo ODOJ yang dapat dilihat pada Gambar 3.18.



Gambar 3.18 *Layout Splash Screen*.

### 3.6.2. *Layout Login*

*Layout* ini digunakan “Odojer” untuk masuk ke dalam aplikasi. *Layout login* memiliki dua *edit text* yang berfungsi sebagai *input e-mail* dan *password*, *layout login* juga memiliki satu tombol *Login*, dua *textview* yaitu *Daftar* dan *Lupa Password*. *Layout Login* dapat dilihat pada Gambar 3.19.

The image shows a wireframe of a mobile login screen. At the top, the title "Login" is centered. Below it are two input fields: "Email" and "Password". A rounded rectangular button labeled "Login" is positioned below the password field. Underneath the button are two links: "Daftar" and "Lupa Password". At the bottom of the screen, there is a link that says "Masuk Sebagai Admin".

Gambar 3.19 *Layout Login.*

### 3.6.3. *Layout Daftar*

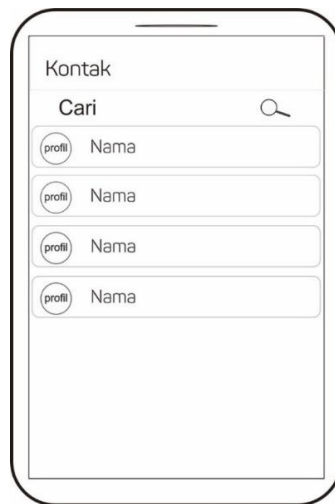
Layout ini digunakan *user* untuk mendaftarkan program ODOJ. *Layout Daftar* memiliki empat *edit text* sebagai peng-input nama, *e-mail*, nomor dan *password*. *Layout* daftar memiliki dua tombol yaitu tombol daftar dan daftar menggunakan G+. *Layout* daftar dapat dilihat pada Gambar 3.20.

The image shows a wireframe of a mobile registration screen. At the top, the title "Daftar" is centered. Below it are four input fields: "Nama", "Email", "Nomor", and "Password". Below the "Password" field are two rounded rectangular buttons: "Daftar" and "Daftar menggunakan G+".

Gambar 3.20 *Layout Daftar.*

#### 3.6.4. *Layout Chat Admin*

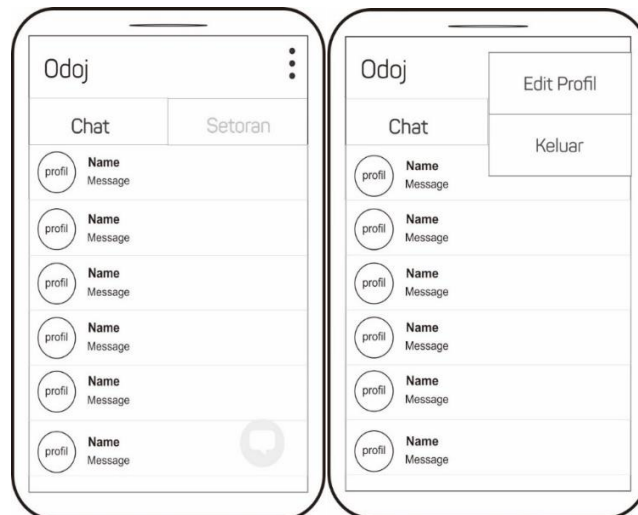
*Layout* ini digunakan bagi user yang akan bergabung ke dalam grup. Halaman chat admin digunakan juga oleh user yang saat mendaftarkan diri sebagai anggota namun tidak ada grup yang kosong, maka user dapat menghubungi Admin untuk mengkonfirmasi bahwa user belum memiliki grup. *Layout Chat Admin* dapat dilihat pada Gambar 3.21.



Gambar 3.21 *Layout Chat Admin.*

#### 3.6.5. *Layout Menu Utama (Dashboard)*

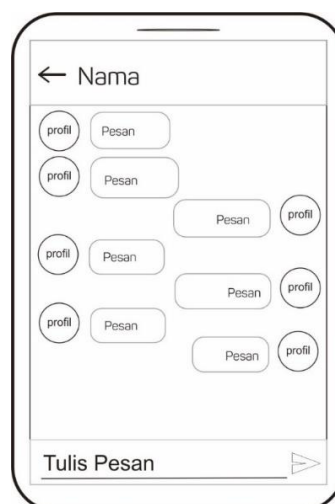
*Layout* ini merupakan menu utama dari aplikasi ODOJ. *Layout* menu utama “Odojer” memiliki dua *tab layout*, *layout* pertama digunakan “Odojer” untuk menampilkan daftar pesan terakhir dari sesama pengguna aplikasi ODOJ, sedangkan *layout* kedua digunakan “Odojer” untuk melakukan “laporan tilawah”. *Layout Menu Utama* dapat dilihat pada Gambar 3.22.



Gambar 3.22 Menu Utama.

### 3.6.6. *Layout Room Chat*

*Layout* ini merupakan *room chat* dari aplikasi ODOJ. *Layout room chat* digunakan untuk menampilkan pesan terkirim dan pesan yang diterima dari sesama pengguna aplikasi ODOJ. *Layout room chat* dapat dilihat pada Gambar 3.23.



Gambar 3.23 *Layout Room Chat*.

### 3.6.7. *Layout Profil*

*Layout* ini merupakan *tempat bagi user* maupun admin untuk menampilkan dan mengubah foto. *Layout Edit profil* dapat dilihat pada Gambar 3.24.



Gambar 3.24 *Layout Profil*

### 3.6.8. *Layout Laporan versi “Odojer”*

*Layout* laporan merupakan grup yang berisi “Odojer” dan orang admin. *Layout* ini digunakan untuk memberikan laporan tilawah dari aplikasi ODOJ. *Layout* laporan dapat dilihat pada Gambar 3.25.



Gambar 3.25 *Layout Laporan “Odojer”*.

### 3.6.9. *Layout Grup – Admin*

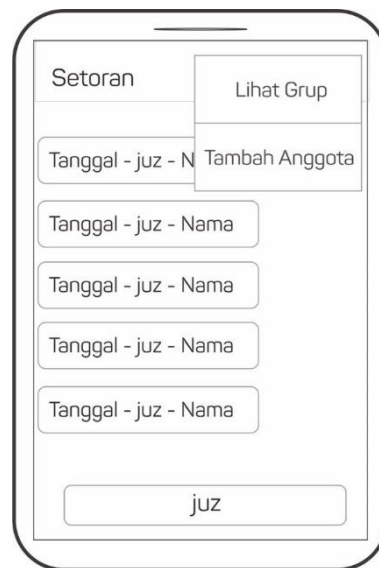
*Layout* utama grup admin merupakan grup yang berisi semua admin. *Layout* grup versi admin berfungsi untuk melaporkan rekapan dari tiap grup yang mereka pegang. *Layout grup* versi admin dapat dilihat pada Gambar 3.26.



Gambar 3.26 *Layout Grup – Admin*.

### 3.6.10. *Layout Laporan – Admin*

*Layout* laporan admin merupakan grup yang berisi “Odojer” dan seorang admin. *Layout* laporan versi admin digunakan untuk menambah anggota yang merupakan seorang “Odojer”. *Layout* laporan versi admin dapat dilihat pada Gambar 3.27.

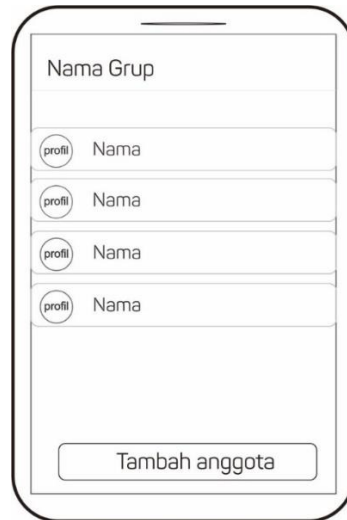


Gambar 3.27 *Layout* Laporan – Admin.

### 3.6.11. *Layout Grup Laporan – Admin*

*Layout* lihat grup laporan versi admin merupakan lanjutan dari *layout* laporan versi admin. *Layout* ini Digunakan untuk melihat anggota dan menambah anggota grup. *Layout* lihat laporan versi admin dapat dilihat pada Gambar 3.28.

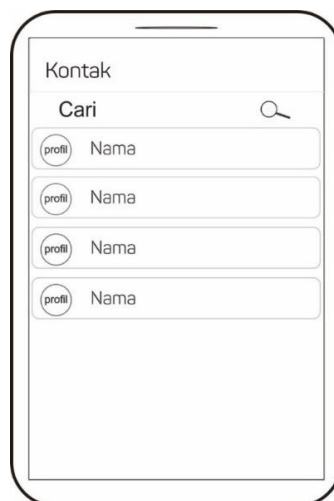




Gambar 3.28 *Layout Grup Laporan – Admin.*

### 3.6.12. *Layout Kontak*

*Layout Menu Kontak* merupakan *layout* yang digunakan user untuk melihat daftar kontak atau pengguna dari aplikasi ODOJ. *Layout* kontak dapat dilihat pada Gambar 3.29.



Gambar 3.29 *Layout Lihat Grup Laporan – Admin.*

### 3.7. Metode Pengujian Sistem

Metode pengujian yang dilakukan pada penelitian ini adalah pengujian *Black-box Testing* menggunakan metode *Equivalence Partitioning* (EP). Pengujian dilakukan dengan memastikan fungsionalitas dan kesesuaian antara input dan output dari aplikasi. Pengujian juga dilakukan dengan membagi domain masukan dan program ke dalam kelas kelas sehingga testcase diperoleh. Metode EP digunakan karena dapat mencari kesalahan pada fungsi aplikasi, interface serta kesalahan pada struktur aplikasi. Rancangan pengujian dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Rancangan Pengujian

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang diharapkan
1	Versi Android	Pengujian kompatibilitas versi operating system Android	Pengujian pada Android versi 5.0 ( <i>Lollipop</i> )  Pengujian pada Android versi 6.0 ( <i>Marshmallow</i> )  Pengujian pada Android versi 7.0 ( <i>Nougat</i> )  Pengujian pada Android versi 8.0 ( <i>Oreo</i> )  Pengujian pada Android versi 9.0 ( <i>Pie</i> )  Pengujian pada Android versi 10.0	Kompatibel dengan Android versi 5.0 ( <i>Lollipop</i> )  Kompatibel dengan Android versi 6.0 ( <i>Marshmallow</i> )  Kompatibel dengan Android versi 7.0 ( <i>Nougat</i> )  Kompatibel dengan Android versi 8.0 ( <i>Oreo</i> )  Kompatibel dengan Android versi 9.0 ( <i>Pie</i> )  Pengujian pada Android versi 10.0

Tabel 3.1 Rancangan Pengujian (lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang diharapkan
2	Ukuran Layar	Pengujian Ukuran Layar pada Android	Pengujian pada Android dengan ukuran layar 5 inch	Kompatibel pada Android dengan ukuran layar 5 inch
			Pengujian pada Android dengan ukuran layar 5.5 inch	Kompatibel pada Android dengan ukuran layar 5.5 inch
			Pengujian pada Android dengan ukuran layar 6 inch	Kompatibel pada Android dengan ukuran layar 6 inch
			Pengujian pada Android dengan ukuran layar 6.5 inch	Kompatibel pada Android dengan ukuran layar 6.5 inch
3	User Interface	Pengujian pada Layout Menu Utama	Klik button "login"	Menampilkan <i>Layout Login</i>
			Klik button "daftar"	Menampilkan <i>Layout Daftar</i>
		Pengujian pada Layout Menu Login	Klik button "Login"	Menampilkan <i>Layout Dashboard</i>
			Klik button "Daftar di sini"	Menampilkan <i>Layout Daftar</i>
		Pengujian pada Layout Menu Daftar	Klik button "Daftar"	Menampilkan <i>Layout Dashboard</i>
			Klik button "Login di sini"	Menampilkan <i>Layout Login</i>

Tabel 3.1 Rancangan Pengujian (lanjutan)

<b>No</b>	<b>Kelas Uji</b>	<b>Daftar Pengujian</b>	<b>Skenario Uji</b>	<b>Hasil yang diharapkan</b>
3	<i>User Interface</i>	Pengujian pada <i>Layout Menu Dashboard</i>	Klik Menu “ <i>About</i> ”	Menampilkan <i>Layout About</i>
			Klik Menu “ <i>Setting</i> ”	Menampilkan <i>Layout Setting</i>
			Menekan <i>Tab layout Laporan</i>	Menampilkan <i>Layout Laporan</i>
			Menekan <i>Tab layout Pesan</i>	Menampilkan <i>Layout Pesan</i>
			Klik Menu <i>icon</i>	Menampilkan <i>layout</i>
			Buat Pesan baru	Buat Pesan baru
			Klik <i>button</i> “ <i>Lapor tilawah</i> ”	Menampilkan Dialog “ <i>Lapor tilawah</i> ”
			Klik Menu <i>Logout</i>	Menampilkan <i>Layout Menu Login</i>
			Klik <i>button</i> “ <i>Edit Profil</i> ”	Menampilkan <i>Layout Menu Edit Profil</i>
			Klik <i>button</i> “ <i>Edit Image</i> ”	Menampilkan <i>Layout Picture</i>
4	Fungsi menu <i>Daftar</i>	Pengujian pada <i>Menu Daftar</i>	Klik tombol “ <i>daftar</i> ”	Menyimpan data-data pendaftar
5	Fungsi menu <i>Login</i>	Pengujian Pada <i>Menu Login</i>	Klik <i>button</i> “ <i>Login</i> ”	Melakukan <i>Autentifikasi User</i>
6	Fungsi menu <i>Dashboard</i>	Pengujian pada <i>Menu Dashboard</i>	Klik Menu “ <i>Setting</i> ”	Menampilkan data data <i>User</i>

Tabel 3.1 Rancangan Pengujian Versi Android (lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang diharapkan
6	Fungsi menu Dashboard	Pengujian pada Menu Dashboard	<p>Klik <i>button</i> “Lapor tilawah”</p> <p>Klik <i>button</i> “Kirim Pesan”</p> <p>Menekan <i>Tab layout Pesan</i></p> <p>Menekan <i>Tab layout Laporan tilawah</i></p> <p>Klik Menu <i>icon</i> Buat Pesan baru</p> <p>Klik Menu “About”</p>	<p>Menyimpan data laporan dan menampilkannya</p> <p>Mengirim dan menyimpan pesan pada <i>database</i> dan menampilkannya</p> <p>Menampilkan daftar data pesan terakhir</p> <p>Menampilkan daftar laporan tilawah</p> <p>Menampilkan daftar kontak</p> <p>Menampilkan data <i>about</i></p>
7	Fungsi Menu <i>Setting</i>	Pengujian pada Menu <i>Setting</i>	<p>Klik <i>button Edit Profil</i></p> <p>Klik <i>button Edit Image</i></p> <p>Klik Menu <i>Logout</i></p>	<p>Menampilkan <i>form edit</i> dan data <i>user</i></p> <p>Mengubah dan menyimpan foto <i>user</i></p> <p>Melakukan <i>Logout</i></p>
8	Fungsi Menu <i>Edit Profil</i>	Pengujian pada Menu <i>Edit Profil</i>	<p>Klik <i>button</i> “Simpan”</p>	<p>Menyimpan data data <i>user</i></p>

## V. SIMPULAN DAN SARAN

### 5.1. Simpulan

Dari hasil penelitian yang telah dilakukan dapat ditarik kesimpulan sebagai berikut:

1. Aplikasi One One Juz berbasis Android telah berhasil dikembangkan.
2. Berdasarkan hasil pengujian fungsional semua fungsi pada sistem berhasil dijalankan dengan baik secara online berdasarkan koneksi internet.
3. Berdasarkan hasil pengujian non fungsional aplikasi dikategorikan *user friendly* dengan penilaian 87% yang berarti sangat baik.

### 5.2. Saran

Aplikasi One One Juz berbasis Android masih memiliki beberapa kekurangan. Berdasarkan hasil implementasi sistem yang dilakukan, maka terdapat beberapa saran yang perlu diperhatikan dalam pengembangan aplikasi ini adalah sebagai berikut.

1. Aplikasi dapat dikembangkan kembali sehingga kompatibel dengan *platform* iOS.

2. Aplikasi dapat dikembangkan kembali sehingga kompatibel dengan platform Android yang tidak support *Google Mobile Services* (GMS).
3. Sistem menyediakan fitur *read chat* ketika pesan sudah dibaca oleh penerima pesan.
4. Sistem dapat memvalidasi pelaporan tilawah yang dilakukan oleh “Odojer”.

## **DAFTAR PUSTAKA**



## DAFTAR PUSTAKA

- Al Fatta, H., 2007. *Analisis dan Perancangan Sistem Informasi untuk keunggulan bersaing perusahaan dan organisasi modern*. Penerbit Andi.
- Arifin, G. dan Faqih, S.A., 2010. *Al-Qur'an Sang Mahkota Cahaya*. Elex Media Komputindo.
- Burnette, E. 2015. *Hello, Android: introducing Google's mobile development platform*. United States of America: Pragmatic Bookshelf.
- Databoks. 2016. Pengguna Smartphone di Indonesia 2016-2019. [Online] Tersedia: <https://databoks.katadata.co.id/datapublish/2016/08/08/pengguna-smartphone-di-indonesia-2016-2019>. Diakses pada tanggal 7 Februari 2019
- Developers, *Mengenal Android Studio*. Diakses pada tanggal 7 Februari 2022.
- Deviana, H. 2011. *Penerapan XML Web service Pada Sistem Distribusi Barang*. Jurnal Generic, vol.6, no. 2, hal. 61-70.
- Developer, G. 2017. *Android Developer Fundamentals Course*. Google Developer Training Team. [Online]. Tersedia: <https://google-developer-training.github.io/android-developer-fundamentals-course-concepts/>
- Moroney, L. 2017. *The Definitive guide to firebase: build Android apps on Google's mobile platform*. Apress.
- Muludi, K., Didik, K., dan Ahmad, H. 2015. *Rancang Bangun Aplikasi Iqro' Berbasis Android*. Jurnal Komputasi. vol. 3.

Nidhra, S., dan Dondeti, J. 2012. *Black-box and white-box testing techniques-a literature review*. International Journal of Embedded Systems and Applications (IJESA), vol. 2, no.2, hal. 29-50.

Onedayonejuz, 2017. Sejarah One day one juz. [Online]. Tersedia: <http://onedayonejuz.org>.

Pangera, A. A., dan Ariyus, D. 2005. *Sistem Operasi*. Yogyakarta: Penerbit Andi.

Rohman, N. dan Rendra, T. 2019. *Kotlin Android Developer Expert*. Bandung: Presentologics

Spath, P. 2018. *Pro Android with Kotlin: Developing Modern Mobile Apps*. Apress.

Stonehem, B. 2016. *Google Android Firebase: Learning the Basics*. First Rank Publishing.

Suryantara, I. G. N. 2017. *Merancang Aplikasi dengan Metodologi Extreme Programming*. Jakarta: Elex Media Komputindo.

Syarbini, A. dan Jamhari, S. 2012. *Kedahsyatan Membaca Alquran*. Bandung: Ruang Kata.

Yayan, F. 2018. Jumlah Rumah Tahfiz di Sumsel Terus Bertambah. [Online]. Tersedia: <https://www.republika.co.id/berita/dunia-islam/islam-nusantara/18/02/13/p41zk4396-jumlah-rumah-tahfiz-di-sumsel-terus-bertambah>. Diakses pada 5 Februari 2019

Zuliarso, E., dan Februariyanti, H., 2013. *Pemanfaatan instant messaging untuk aplikasi layanan akademik*. Dinamik, vol. 18, no. 2, hal. 112-121