

**APLIKASI PENYEDIA JASA ONLINE BERBASIS MOBILE ANDROID**

**(Skripsi)**

**Oleh :**

**M. SHANDY PUTRA**



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG**

**2019**

## **ABSTRAK**

### **APLIKASI PENYEDIA JASA ONLINE BERBASIS MOBILE ANDROID**

**Oleh**

**M. SHANDY PUTRA**

Layanan penyedia jasa merupakan layanan yang sangat penting dan sangat dibutuhkan saat ini. Keinginan orang untuk membuat, membeli, menjual, dan memperbaiki sesuatu membuat penyedia jasa sangat dibutuhkan. Banyaknya penyedia jasa saat ini membuat kita terkadang bingung untuk memilih dan menentukan penyedia jasa mana yang dapat membantu dan cocok dengan keinginan kita sebagai pengguna jasa. Dalam penelitian ini, peneliti merancang dan membangun aplikasi penyedia jasa *online* berbasis Android dengan nama Cari Tukang. Aplikasi berguna untuk mencari dan menampilkan penyedia jasa dan juga pengguna jasa yang ada di Bandar Lampung. Sistem ini dirancang menggunakan metode *Unified Process*, dan diuji menggunakan metode *Black Box*. Kesimpulan yang didapat dari penelitian ini adalah aplikasi Cari Tukang berhasil dibangun dan berfungsi dengan baik dibuktikan dengan pengujian yang telah dilakukan.

Kata kunci: Cari tukang, Penyedia Jasa, Android.

## **ABSTRACT**

**ANDROID-BASED ONLINE LABOR SERVICE PROVIDER APPLICATION**

**Oleh**

**M. SHANDY PUTRA**

Labor service provider is one of the most required service nowadays. Consumer's wish to create, sell, buy, and fix something is one of the main reason of the need for this service. The vast amount of labor service provider today proved to be a problem for us, especially in choosing whether they are able suitable to our request and able to complete our request as consumer. In this research, we create and build android-based online service provider named Cari Tukang. The application purpose is to find and display list of labor service provider located in Bandar Lampung. The system is designed using Unified Process method, and tested using Black Box method. The conclusion we obtained from the research is Cari Tukang application have been built and functioning properly proven by the testing we have done.

***Keywords : Android, Cari Tukang, Service Provider .***

**APLIKASI PENYEDIA JASA ONLINE  
BERBASIS MOBILE ANDROID**

**Oleh**

**M. SHANDY PUTRA**

**Skripsi**

Sebagai salah satu syarat untuk memperoleh gelar  
SARJANA ILMU KOMPUTER

Pada

Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2019**

Judul Skripsi : **APLIKASI PENYEDIA JASA ONLINE BERBASIS  
MOBILE ANDROID**

Nama Mahasiswa : **M. Shandy putra**

No. Pokok Mahasiswa : 1217051041

Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam

**MENYETUJUI**

1. Komisi Pembimbing



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



**Drs. Rd. Irwan Adi Pribadi, M.Kom.**  
NIP. 19630110 198902 1 002

2. Mengetahui  
Ketua Jurusan Ilmu Komputer



**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
NIP. 19640616 198902 1 001

**MENGESAHKAN**

1. Tim Penguji

Ketua : **Didik Kurniawan, S.Si., M.T.**



Sekretaris : **Drs. Rd. Irwan Adi Pribadi, M.Kom.**



Penguji  
Bukan pembimbing : **Tristiyanto, S.Kom., M.I.S., Ph.D.**



2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Drs. Suratman, M.Sc.**  
Telp. 19640604 199003 1 002

Tanggal lulus Ujian Skripsi : **10 Desember 2019**

## PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul "APLIKASI PENYEDIA JASA ONLINE BERBASIS MOBILE ANDROID" merupakan karya saya sendiri dan bukan karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti Kaidah Penulisan Karya Ilmiah Universitas Lampung. Apabila dikemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.



NPM. 1217051041

## RIWAYAT HIDUP



Penulis dilahirkan pada tanggal 9 Juni 1994 di Bandar Lampung, sebagai anak ketiga dari empat bersaudara dari ayah yang bernama Syahril, dan ibu yang bernama Elly Wahyuni. Penulis menyelesaikan pendidikan pertamanya di TK Dharma Wanita pada Tahun 2000, melanjutkan sekolah

dasar di SD Kartika II-5 Bandar Lampung lulus pada Tahun 2006, Sekolah Menengah Pertama di SMP Negeri 4 Bandar Lampung dan lulus pada tahun 2009. Kemudian, penulis melanjutkan pendidikan menengah atas di SMA Negeri 10 Bandar Lampung dan lulus pada tahun 2012. Pada tahun 2012 penulis terdaftar sebagai mahasiswa di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

Selama masa perkuliahan, penulis aktif mengikuti organisasi di dalam kampus. Adapun organisasi di dalam kampus yang diikuti penulis adalah organisasi HIMAKOM (Himpunan Mahasiswa Komputer).



Selain aktif organisasi, selama menjadi mahasiswa penulis juga aktif mengikuti kegiatan antara lain:

1. Pada bulan Januari Tahun 2013 penulis mengikuti kegiatan Karya Wisata Ilmiah (KWI) FMIPA Unila di Desa Lebakharjo IV, Kabupaten Pringsewu, Lampung.
2. Pada bulan Februari Tahun 2015 penulis melaksanakan kegiatan Kerja Praktik di Badan Pusat Statistik Provinsi Lampung.
3. Pada bulan Juli Tahun 2015 penulis melaksanakan Kegiatan KKN Tematik Unila di Desa Indraloka I, Kabupaten Tulang Bawang barat, Lampung

## SANWACANA

*Assalamualaikum Wr. Wb*

*Alhamdulillah* *rabbil'alamin*, puji syukur kehadiran Allah SWT atas berkat rahmat, hidayah, dan karunia-Nya sehingga penulis dapat menyelesaikan penulisan skripsi ini yang berjudul “ Aplikasi Penyedia Jasa Online Berbasis Mobile Android”. Skripsi ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Universitas Lampung.

Penyelesaian skripsi ini tidak terlepas dari bantuan banyak pihak, baik secara materi maupun moril dan juga saran serta bimbingan dari banyak pihak. Penulis mengucapkan terimakasih kepada:

1. Kedua orang tua tercinta, Ayahanda Alm. Syahril, dan Ibunda Elly Wahyuni Soegiarto, Kakakku Elisa Putri, Gita Kurniawan Putra serta adikku Nurul Putri, dan juga keluarga besar yang selalu memberi doa, dukungan, motivasi, dan kasih sayang yang tak terhingga sehingga penulis mampu menyelesaikan skripsi ini.
2. Bapak Didik Kurniawan, S.Si., M.T. sebagai Pembimbing Utama dan Pembimbing Akademik, yang telah membimbing penulis dan memberikan kritik serta saran sehingga penulisan skripsi ini dapat diselesaikan.
3. Bapak Drs. Rd. Irwan Adi Pribadi, M.Kom. sebagai Pembimbing Kedua, yang telah memberikan saran, bantuan, dan membimbing penulis sehingga penulisan skripsi ini dapat diselesaikan.

4. Bapak Tristiyanto, S.Kom., M.I.S., Ph.D. sebagai Pembahas , yang telah memberikan masukan yang bermanfaat dalam penulisan skripsi ini.
5. Bapak Drs. Suratman, M.Sc. selaku Dekan FMIPA Unila.
6. Bapak Dr. Ir. Kurnia Muludi, M.S.Sc. selaku Ketua Jurusan Ilmu Komputer.
7. Bapak Didik Kurniawan, S.Si., M.T. selaku Sekretaris Jurusan Ilmu Komputer FMIPA Unila.
8. Bapak dan Ibu Dosen Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan ilmu yang bermanfaat.
9. Ibu Ade Nora Maela dan Ibu Lusiana yang telah membantu segala urusan administrasi Penulis di Jurusan Ilmu Komputer.
10. Teman-teman seperjuangan : Qiqin, Debi, Cindy, Uchi, Febrianti, Nurul, Indah Ayu Atika, Erika, Dipa, Rahman, Yobi, Afrizka Amidya, Nurmayanti, Aprizal Arief, Hendire, M. Rahman, Juan Syahputra, Yuda, dan teman-teman Jurusan Ilmu Komputer 2012.
11. Mas Naufal dan Mba Lia yang telah memberikan fasilitas dan menyiapkan ruang seminar MIPA Terpadu.
12. Almamater Tercinta.

Bandar Lampung, Desember 2019

Penulis

M. Shandy Putra

## **PERSEMBAHAN**

“Yakinlah bahwa setiap usaha yang kita kerjakan dengan kesungguhan hati pasti akan kita dapatkan hanya dengan sebuah kesabaran”

Dengan mengucap puji dan syukur kehadirat Allah SWT  
kupersembahkan karya kecilku ini untuk:

Orang Tua ku tercinta Alm. Syahril dan Ibunda Elly Wahyuni Soegiarto, kedua kakakku dan adikku yang tak pernah henti memberikan doa, motivasi serta kesabaran yang luar biasa.

Untuk

Semua temanku yang selalu memberikan motivasi, perhatian dan juga bantuan kepada penulis yang tiada hentinya.

## **MOTTO**

*“Jalani hidup dengan penuh senyuman” (M. Shandy Putra)*

*“Jadikanlah Sabar dan Shalat sebagai penolongmu, sesungguhnya Allah bersama orang-orang yang sabar “ (Q.S Al-Baqarah: 153)*

*“ Maka sesungguhnya bersama kesulitan ada kemudahan, Sesungguhnya bersama kesulitan ada kemudahan” (Q.S Al-Insyrah:5-6)*

## DAFTAR ISI

	Halaman
DAFTAR ISI.....	i
DAFTAR GAMBAR.....	vi
DAFTAR TABEL.....	ix
DAFTAR KODE.....	x
BAB I PENDAHULUAN	
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan.....	3
1.5 Manfaat.....	4
BAB II TINJAUAN PUSTAKA	
2.1 Pengertian Penyedia Jasa.....	5
2.1.1 Pengertian Jasa.....	5
2.1.2 Pengertian <i>Online</i> .....	6
2.2 Pengertian <i>Chat Messenger</i> .....	6

2.3 Android .....	6
2.3.1 Sejarah Android .....	7
2.3.2 Arsitektur Android .....	9
2.3.2.1 <i>Application</i> dan <i>Widgets</i> .....	10
2.3.2.2 <i>Application Frameworks</i> .....	10
2.3.2.3 <i>Libraries</i> .....	11
2.3.2.4 <i>Android Run Time</i> .....	12
2.3.2.5 <i>Linux Kernel</i> .....	12
2.3.3 Komponen Aplikasi Android .....	12
2.3.3.1 <i>Activities</i> .....	13
2.3.3.2 <i>Service</i> .....	13
2.3.3.3 <i>Broadcast Receiver</i> .....	14
2.3.3.4 <i>Content Provider</i> .....	14
2.3.4 Distribusi Android .....	15
2.4 <i>Android Development Tools</i> .....	16
2.4.1 <i>Android Software Development Kit (SDK)</i> .....	16
2.4.2 <i>Android Development Tools (ADT)</i> .....	16
2.4.3 <i>Android Studio</i> .....	17
2.4.3.1 <i>Fitur Android Studio</i> .....	17
2.4.3.2 <i>Kebutuhan Installasi Android Studio</i> .....	18
2.5 Metodologi Pengembangan Sistem .....	19
2.5.1 <i>Unified Modeling Language (UML)</i> .....	19
2.5.1.1 <i>Use Case Diagram</i> .....	20
2.5.1.2 <i>Activity Diagram</i> .....	21

2.5.1.3	<i>Class Diagram</i>	22
2.5.1.4	<i>Sequence Diagram</i>	23
2.6	Firestore	25
2.6.1	<i>Firestore Cloud Messaging and Notifications</i>	25
2.6.2	<i>Firestore Real Time Database</i>	26
2.7	Pengujian Perangkat Lunak	26
2.7.1	<i>Equivalence Partitioning</i>	27

### BAB III METODOLOGI PENELITIAN

3.1	Waktu dan Tempat Penelitian	28
3.2	Metodologi Penelitian	28
3.2.1	Alur Penelitian	28
3.2.2	Metode Pengembangan Sistem	32
3.2.3	Metode Pengumpulan Data	35
3.3	Analisis dan Perancangan Sistem	35
3.3.1	Identifikasi Masalah	35
3.3.2	Analisis Sistem	36
3.3.3	<i>Analisis User Requirement</i>	36
3.3.4	Perancangan Sistem	36
3.3.4.1	Desain Arsitektur	37
3.3.4.2	Perancangan UML ( <i>Unified Modelling Language</i> )	38
3.3.4.3	Perancangan Antarmuka	52
3.4	Metode Pengujian Sistem	62

### BAB IV HASIL DAN PEMBAHASAN



4.1 Pembahasan.....	65
4.2 Implementasi Kode Program.....	65
4.2.1 Kode Program <i>Splash screen</i> .....	66
4.2.2 Kode Program Menu Utama.....	66
4.2.3 Kode Program Notifikasi.....	68
4.2.4 Kode Program Cari Tukang.....	71
4.2.5 Kode Program Cari Order.....	72
4.2.6 Kode Program My Jasa.....	74
4.2.7 Kode Program My Order.....	76
4.3 Hasil.....	78
4.4 Implementasi Sistem.....	79
4.5 <i>Layout</i> Apilkasi Cari Tukang.....	81
4.5.1 <i>Layout</i> Halaman <i>Splash Screen</i> .....	82
4.5.2 <i>Layout</i> Halaman Menu Utama.....	82
4.5.3 <i>Layout</i> Halaman Notifikasi.....	84
4.5.4 <i>Layout</i> Halaman Cari Tukang.....	85
4.5.5 <i>Layout</i> Halaman Cari Order.....	88
4.5.6 <i>Layout</i> Halaman My Jasa.....	89
4.5.7 <i>Layout</i> Halaman My Order.....	91
4.5.8 <i>Layout</i> Halaman Bantuan.....	92
4.5.9 <i>Layout</i> Halaman Tentang Aplikasi.....	93
4.6 Hasil Pengujian.....	94
4.6.1 Pengujian <i>Black Box</i> .....	94
4.6.1.1 Pengujian Fungsional.....	94

4.6.1.1.1 Pengujian Versi Android.....	95
4.6.1.1.2 Pengujian Resolusi Layar dan Densitas Layar.....	96
4.6.1.1.3 Pengujian <i>User Interface</i> .....	97
4.6.1.2 Pengujian Non Fungsional.....	99
4.7 Ulasan Pengguna.....	103

## BAB V SIMPULAN DAN SARAN

5.1 Simpulan.....	106
5.2 Saran.....	107

DAFTAR PUSTAKA.....	108
---------------------	-----

## LAMPIRAN

## DAFTAR GAMBAR

	Halaman
Gambar 2.1. Arsitektur <i>Android</i> .....	9
Gambar 3.1. Diagram Alir Metodologi Penelitian.....	29
Gambar 3.2. Diagram <i>Unified Process</i> (UP).....	34
Gambar 3.3. Desain Arsitektur.....	37
Gambar 3.4. <i>Use Case Diagram</i> .....	38
Gambar 3.5. <i>Activity Diagram</i> Menu Login.....	39
Gambar 3.6. <i>Activity Diagram</i> Notifikasi.....	40
Gambar 3.7. <i>Activity Diagram</i> Cari Tukang.....	40
Gambar 3.8. <i>Activity Diagram</i> Cari Order.....	41
Gambar 3.9. <i>Activity Diagram</i> Menu My Jasa.....	42
Gambar 3.10. <i>Activity Diagram</i> My Order.....	42
Gambar 3.11. <i>Activity Diagram</i> Bantuan.....	43
Gambar 3.12. <i>Activity Diagram</i> Tentang Aplikasi.....	44
Gambar 3.13. <i>Sequence Diagram</i> Login.....	45
Gambar 3.14. <i>Sequence Diagram</i> Notifikasi.....	46
Gambar 3.15. <i>Sequence Diagram</i> Cari Tukang.....	47
Gambar 3.16. <i>Sequence Diagram</i> Cari Order.....	48

Gambar 3.17. <i>Sequence Diagram</i> My Jasa .....	49
Gambar 3.18. <i>Sequence Diagram</i> My Order .....	50
Gambar 3.19. <i>Sequence Diagram</i> Bantuan.....	51
Gambar 3.20. <i>Sequence Diagram</i> Tentang Aplikasi.....	51
Gambar 3.21 <i>Class Diagram</i> .....	52
Gambar 3.22. <i>Design Layout Splash Screen</i> .....	53
Gambar 3.23. <i>Design Layout Layout</i> Menu Utama.....	54
Gambar 3.24. <i>Layout menu</i> Notifikasi .....	55
Gambar 3.25. <i>Design Layout</i> Menu Cari Tukang.....	56
Gambar 3.26. <i>Design Layout Layout</i> Detail Tukang.....	56
Gambar 3.27. <i>Design Layout Messenger</i> .....	57
Gambar 3.28. <i>Design Layout</i> Menu Cari Order.....	58
Gambar 3.29. <i>Design Layout</i> Detail Order.....	58
Gambar 3.30. <i>Layout</i> Menu My Jasa.....	59
Gambar 3.31. <i>Layout</i> Tambah Jasa.....	59
Gambar 3.32. <i>Design Layout</i> Menu My Order.....	60
Gambar 3.33. <i>Layout</i> Tambah Order.....	61
Gambar 3.34. <i>Design Layout</i> Menu Bantuan.....	61
Gambar 3.35. <i>Design Layout</i> Menu Tentang Aplikasi.....	62
Gambar 4.1 <i>Layout Splash Screen</i> .....	82
Gambar 4.2 <i>Layout</i> Menu Utama.....	83
Gambar 4.3 <i>Layout</i> Menu Navigasi.....	84

Gambar 4.4 <i>Layout</i> Halaman Notifikasi .....	85
Gambar 4.5 <i>Layout</i> Halaman Cari Tukang .....	86
Gambar 4.6 <i>Layout</i> Halaman Detail Tukang .....	87
Gambar 4.7 <i>Layout</i> Halaman <i>Messenger</i> .....	87
Gambar 4.8 <i>Layout</i> Halaman Cari Order .....	88
Gambar 4.9 <i>Layout</i> Halaman Detail Order .....	89
Gambar 4.10 <i>Layout</i> Halaman My Jasa .....	90
Gambar 4.11 <i>Layout</i> Halaman Tambah Jasa .....	90
Gambar 4.12 <i>Layout</i> Halaman My Order .....	91
Gambar 4.13 <i>Layout</i> Halaman Tambah Order .....	92
Gambar 4.14 <i>Layout</i> Menu Bantuan .....	93
Gambar 4.15 <i>Layout</i> Menu Tentang Aplikasi .....	94
Gambar 4.16 Grafik Presentasi Rata-rata Jawaban Responden per Kategori Penilaian pada Variabel <i>User Friendly</i> .....	102
Gambar 4.17 Grafik Presentasi Rata-rata Jawaban Responden per Kategori Penilaian pada Variabel Interaktif .....	103
Gambar 4.18 Aplikasi Cari Tukang Pada Play Store .....	104
Gambar 4.19 Komentar Aplikasi Cari Tukang Pada Play Store .....	105

## DAFTAR TABEL

	Halaman
Tabel 2.1 Kebutuhan Instalasi Android Studio.....	18
Tabel 2.2. Simbol dan Keterangan <i>Use Case</i> .....	20
Tabel 2.3. Simbol dan Keterangan <i>Activity Diagram</i> .....	21
Tabel 2.4. Simbol dan keterangan <i>Class Diagram</i> .....	22
Tabel 2.5. Simbol dan keterangan <i>Sequence Diagram</i> .....	23
Tabel 3.1 Daftar Pengujian <i>Equivalence Partitioning</i> (EP).....	63
Tabel 4.1 Daftar <i>Class</i> pada Aplikasi Cari Tukang.....	79
Tabel 4.2 Daftar <i>Layout</i> pada Aplikasi Cari Tukang.....	81
Tabel 4.3 Tabel Pengujian Versi Android.....	95
Tabel 4.4 Pengujian Resolusi Layar dan Densitas Layar.....	96
Tabel 4.5 Pengujian <i>User Interface</i> .....	97
Tabel 4.6 Interval dan Kategori Penilaian.....	100
Tabel 4.7 Hasil Penilaian Variabel <i>User Friendly</i> .....	100
Tabel 4.8 Hasil Penilaian Variabel Interaktif.....	102

## DAFTAR KODE

	Halaman
Kode 4.1 Potongan Kode Program <i>Splashscreen</i> .....	66
Kode 4.2 Potongan Kode Program Menu Utama.....	67
Kode 4.3 Potongan Kode Program XML Menu Utama.....	68
Kode 4.4 Potongan Kode Program Menu Notifikasi.....	69
Kode 4.5 Potongan Kode Program XML Menu Notifikasi.....	70
Kode 4.6 Potongan Kode Program Menu Cari Tukang.....	71
Kode 4.7 Potongan Kode Program XML Menu Cari Tukang.....	72
Kode 4.8 Potongan Kode Program Cari Order.....	73
Kode 4.9 Potongan Kode Program XML Cari Order.....	74
Kode 4.10 Potongan Kode Program My Jasa.....	75
Kode 4.11 Potongan Kode Program XML My Jasa.....	76
Kode 4.12 Potongan Kode Program My Order.....	77
Kode 4.13 Potongan Kode Program XML My Order.....	78

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Layanan penyedia jasa merupakan layanan yang sangat penting dan sangat dibutuhkan saat ini. Keinginan orang untuk membuat, membeli, menjual, dan memperbaiki sesuatu membuat penyedia jasa sangat dibutuhkan. Dengan adanya penyedia jasa, orang tidak perlu terlalu repot untuk mengerjakan sesuatu. Dengan menghubungi penyedia jasa yang diperlukan, orang tersebut akan lebih merasa dimudahkan dikarenakan pihak penyedia jasa yang akan menyelesaikan masalahnya.

Banyaknya penyedia jasa saat ini membuat kita terkadang bingung untuk memilih dan menentukan penyedia jasa mana yang dapat membantu dan cocok dengan keinginan kita sebagai pengguna jasa. Dikarenakan terkadang belum tentu yang kita inginkan dapat dilaksanakan oleh pihak penyedia jasa, dan juga apakah tempat penyedia jasa yang akan kita gunakan masih menyediakan jasanya atau tidak. Dengan mempertimbangkan hal tersebut, penulis akan merancang dan membangun aplikasi yang dapat membantu masyarakat untuk mencari penyedia jasa yang tepat untuk membantu menyelesaikan masalah mereka.



Dengan memanfaatkan perkembangan teknologi, tentu saja hal tersebut dapat dilakukan. Berdasarkan hal tersebut, pada penelitian ini akan dirancang dan dibangun sebuah aplikasi Penyedia Jasa Online berbasis Android yang dapat membantu masyarakat untuk mencari penyedia jasa yang tepat untuk membantu menyelesaikan masalah mereka. Dimana pengguna dapat meng-*install* aplikasi ini ke *handphone* pengguna, dan dapat meng-*order* serta melihat informasi terkait penyedia jasa mana yang dapat membantu menyelesaikan masalah yang sedang pengguna hadapi. Selain itu, untuk para penyedia jasa sendiri dapat memasang informasi tentang layanan jasa yang mereka sediakan. Aplikasi ini juga mampu membuat pengguna dapat berkomunikasi langsung antar pihak penyedia jasa dan pihak konsumen melalui *chat messenger* yang ada pada aplikasi ini. Selain itu, aplikasi ini dapat menampilkan informasi layanan apa saja yang disediakan oleh pihak penyedia jasa tersebut dan juga menampilkan informasi kontak yang dapat dihubungi oleh pihak pemesan.

Hal-hal yang telah dijabarkan tersebut menjadi latar belakang dalam melakukan penelitian yang disajikan dalam skripsi yang berjudul “Aplikasi Penyedia Jasa Online Berbasis Mobile Android”, dengan harapan membantu setiap pengguna *smartphone* Android untuk dapat mencari dan melihat informasi layanan jasa yang disediakan oleh para penyedia jasa serta dapat meng-*order* jasa tersebut untuk membantu menyelesaikan masalah pengguna.

## 1.2 Rumusan Masalah

Rumusan masalah dalam penelitian ini adalah “Bagaimana merancang dan membangun aplikasi penyedia jasa *online* yang dapat mencari, menampilkan, dan meng-*order* layanan penyedia jasa dengan berbasis Android”.

## 1.3 Batasan Masalah

Aplikasi penyedia jasa *online* untuk mencari, menampilkan, dan meng-*order* layanan jasa dengan berbasis Android ini memiliki batasan sebagai berikut:

1. Pada penelitian ini aplikasi hanya menampilkan informasi jasa layanan yang berada di kota Bandar Lampung.
2. Aplikasi ini hanya menyediakan Jasa Kontraktor, Jasa Pindahan, Jasa Pembersihan, Jasa Perbaikan Peralatan, dan Jasa Sewa Barang.
3. Aplikasi ini hanya dapat meng-*order* dan menerima *order*-an jasa sesuai dengan kategori yang tersedia di dalam aplikasi.
4. Aplikasi ini hanya menyediakan fitur *chatting* antar pemesan jasa dan penyedia jasa untuk saling berkomunikasi.

## 1.4 Tujuan

Penelitian ini bertujuan untuk merancang dan membangun aplikasi yang dapat mencari, menampilkan, dan meng-*order* layanan jasa yang ada dengan berbasis Android.

## **1.5 Manfaat**

Manfaat yang diperoleh dari penelitian ini diharapkan dapat membantu dan memudahkan pengguna *smartphone* Android sebagai pengguna jasa untuk mencari penyedia jasa yang ada serta memudahkan pihak penyedia jasa untuk memasarkan jasa yang disediakan.

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Pengertian Penyedia Jasa**

Penyedia Barang/Jasa adalah badan usaha atau orang perseorangan yang menyediakan Barang/Pekerjaan Konstruksi/Jasa Konsultansi/Jasa Lainnya (Perpres No.54, 2010).

##### **2.1.1 Pengertian Jasa**

Ada beberapa definisi jasa yang dikemukakan oleh beberapa ahli. Diantaranya ialah menurut Philip Kotler seperti yang dikutip J. Supranto (2001, p227) yang mendefinisikan jasa sebagai berikut:

“Jasa ialah setiap tindakan atau kegiatan yang dapat ditawarkan oleh satu pihak kepada pihak lain yang pada dasarnya tidak berwujud dan tidak mengakibatkan kepemilikan apapun. Produksinya dapat dikaitkan atau tidak dikaitkan pada satu produk fisik”.

Dari definisi tersebut dapat disimpulkan bahwa jasa merupakan suatu kegiatan yang tidak berwujud dan cepat hilang serta tidak dapat dimiliki. Secara umum jasa adalah setiap tindakan atau perbuatan yang dapat ditawarkan oleh suatu pihak kepada pihak lain dimana produk yang ditawarkan bisa berupa produk fisik maupun tidak dimana jika produk itu berupa produk fisik yang didalam

tahapannya akan melalui beberapa perubahan sehingga nantinya akan memuaskan keinginan konsumen/ pelanggan tersebut.

### **2.1.2 Pengertian *Online***

*Online* adalah istilah saat kita sedang terhubung dengan internet atau dunia maya, baik itu terhubung dengan akun media sosial kita, *email* dan berbagai jenis akun lainnya yang kita pakai atau gunakan lewat internet.

## **2.2 Pengertian *Chat Messenger***

*Chat Messenger* adalah suatu teknologi jaringan yang memungkinkan penggunanya mengirimkan pesan secara *realtime* ke pengguna lain yang tersambung dalam sebuah jaringan LAN atau *Local Area Network* ataupun internet (Jumadi, 2004).

Pengertian *Chat Messenger* atau *Chatting* adalah mengobrol jika diterjemahkan langsung dari Bahasa Inggris. Dalam dunia komputer dan internet, pengertian *Chat Messenger* adalah suatu fasilitas dalam internet untuk berkomunikasi sesama pengguna internet yang sedang *on-line*. Komunikasi dapat berupa teks. Sekarang banyak sudah banyak sekali *software* atau aplikasi yang memfasilitasi pengguna untuk memungkinkan melakukan *Chat Messenger*, misalnya *Yahoo Messenger*, *Google Talk*, *Skype*, dan sebagainya.

## **2.3 Android**

Pada awal peluncurannya, Google meyakini bahwa *platform* perangkat *mobile Android* memiliki kesempatan yang sangat besar dalam pengembangan aplikasi.

*Google* mengumumkan *Open Handset Alliance* (OHA) dan *platform Android* pada November 2007, dan meluncurkan *Android Software Development Kit* (SDK) pertama yang masih dalam versi *beta* di waktu yang sama. Dalam waktu yang tidak lama, lebih dari satu juta orang mengunduh *Android SDK* dari *website Google*. Di Amerika Serikat, *T-Mobile* mengumumkan perangkat *mobile Android* bergelar *G1* pada Oktober 2008, dan diperkirakan ratusan ribu perangkat *G1* terjual pada akhir tahun yang sama. *Android* memiliki potensi yang besar untuk menghilangkan batasan dan kendala yang selama ini muncul dalam mengembangkan suatu perangkat lunak versi *mobile phone*. Dari berbagai hal yang telah disiapkan *Google*, *Android* berhasil mengembangkan pasar aplikasi perangkat *mobile*, serta memberikan kesempatan besar untuk para pengembang perangkat lunak untuk mengambil keuntungan dari aplikasi yang dibuatnya (Roger. 2009).

### **2.3.1 Sejarah Android**

*Android* adalah sebuah sistem operasi yang sengaja diciptakan untuk perangkat *mobile* berbasis *Linux* yang mencakup sistem operasi, *middleware*, dan aplikasi. *Android* menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat piranti lunak untuk perangkat ponsel pintar atau *smartphone*. Kemudian untuk mengembangkan *Android*, dibentuklah *Open Handset Alliance* (OHA), sebuah konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

Pada saat perilisan perdana *Android*, 5 November 2007, *Android* bersama OHA menyatakan mendukung pengembangan *open source* pada perangkat *mobile*. Dilain pihak, *Google* merilis kode-kode *Android* di bawah lisensi *Apache*, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. Di dunia ini terdapat dua jenis distributor sistem operasi *Android*. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services* (GMS), dan yang kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal dengan *Open Handset Distribution* (OHD).

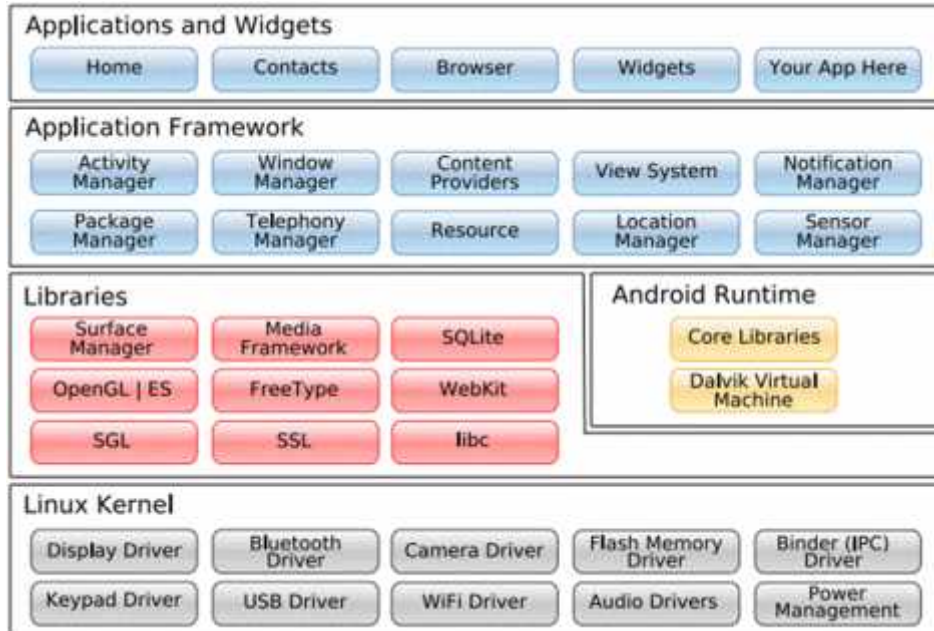
Pada Juli 2005, *Google* bekerjasama dengan *Android Inc.*, perusahaan yang berada di Palo Alto, California Amerika Serikat. Para pendiri *Android Inc.* bekerjasama pada *Google*, diantaranya Andy Rubin, Nick Sears, dan Chris White. Saat itu banyak yang menganggap fungsi *Android* hanyalah sebagai perangkat lunak pada telepon seluler. Sejak saat itu muncul rumor bahwa *Google* hendak memasuki pasar telepon seluler. Pada internal perusahaan *Google*, tim yang dipimpin Rubin bertugas mengembangkan program perangkat seluler yang didukung oleh *kernel Linux*. Hal ini menunjukkan bahwa *Google* sedang bersiap menghadapi persaingan dalam pasar telepon seluler.

Sekitar September 2007, sebuah studi melaporkan bahwa *Google* mengajukan hak paten aplikasi telepon seluler. Akhirnya *Google* mengenalkan *Nexus One*, salah satu jenis telepon pintar GSM yang menggunakan *Android* pada sistem operasinya. Telepon seluler ini diproduksi oleh *HTC Corporation* dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holding*, *Atheros Communications*, diproduksi oleh *Asustek Computer Inc*, *Garmin Ltd*,

Softbank, Sony Ericsson, Toshiba Corp., dan Vodafone Group Plc. Seiring pembentukan OHA (*Open Handset Alliance*), OHA mengumumkan produk perdananya, *Android*, perangkat *mobile* yang merupakan modifikasi dari *kernel Linux2.6*. Sejak *Android* dirilis, telah dilakukan pembaruan berupa pembaruan *bug* dan penambahan fitur baru. Pada penghujung tahun 2009, diperkirakan terdapat 18 jenis telepon seluler yang menggunakan sistem operasi *Android* (Safaat, 2012).

### 2.3.2 Arsitektur Android

Secara garis besar, Arsitektur *Android* dapat dijelaskan dan ditunjukkan seperti pada Gambar 2.1.



Gambar 2.1. Arsitektur *Android*



### **2.3.2.1 Application dan Widgets**

*Applications and Widgets* ini adalah *layer* yang membatasi penggunaannya hanya untuk berhubungan dengan aplikasi saja, seperti halnya saat pengguna mengunduh aplikasi kemudian melakukan instalasi dan menjalankan aplikasi tersebut. Pada *layer* terdapat aplikasi ini termasuk klien *email*, program SMS, kalender, peta, *browser*, atau kontak. Semua aplikasi ditulis dengan menggunakan Bahasa Pemrograman *Java*.

### **2.3.2.2 Application Frameworks**

*Android* adalah “*Open Development Platform*”, yaitu *Android* menawarkan kepada pengembang atau memberi kemampuan kepada pengembang untuk membangun aplikasi yang baik dan inovatif. Pengembang bebas untuk mengakses perangkat keras, akses informasi *resources*, menjalankan *service background*, mengatur alarm, dan menambahkan status *notifications*, serta lain sebagainya. Pengembang memiliki akses penuh menuju *API framework* seperti yang dilakukan oleh aplikasi yang berkategori inti. Arsitektur aplikasi dirancang supaya pengguna dapat dengan mudah menggunakan kembali komponen yang sudah digunakan (*reuse*).

Sehingga dapat dikatakan bahwa *Applications Framework* ini merupakan *layer* yang dapat digunakan pembuat aplikasi melakukan pengembangan atau pembuatan aplikasi yang dijalankan di sistem operasi *Android*, karena pada *layer* inilah aplikasi dapat dirancang dan dibuat, seperti *content-provider* yang berupa

SMS dan panggilan telepon.

Komponen yang termasuk di dalam *Applications Frameworks* adalah sebagai berikut:

- a. *Views*.
- b. *Content Provider*.
- c. *Resource Manager*.
- d. *Notifications Manager*.
- e. *Activity Manager*.

### **2.3.2.3 Libraries**

*Libraries* adalah *layer* yang merupakan tempat fitur-fitur *Android* berada. Para pembuat aplikasi mengakses *libraries* untuk menjalankan aplikasinya. Berjalan di atas *kernel Linux*, *layer* ini meliputi berbagai *library C* atau *C++* inti seperti *Libe* dan *SSL*, serta:

- a. *Libraries media* untuk pemutaran media audio dan video.
- b. *Libraries* untuk manajemen tampilan.
- c. *Libraries graphics* mencakup *SGL* dan *OpenSGL* untuk grafis *2D* dan *3D*.
- d. *Libraries SQLite* untuk dukungan *database*.
- e. *Libraries SSL* dan *WebKit* terintegrasi dengan *web browser* dan *security*.
- f. *Libraries Live Webcote* mencakup modern *web browser* dengan *engine embedded web view*.
- g. *Libraries 3D* yang mencakup implementasi *OpenGL ES 1.0 API's*.

#### **2.3.2.4 Android Run Time**

*Android Run Time* merupakan *layer* yang membuat aplikasi *Android* dapat dijalankan yang dalam prosesnya menggunakan implementasi *Linux*. *Dalvik Virtual Machine* (DVM) merupakan mesin yang membentuk dasar kerangka aplikasi *Android*. *Android Run Time* dibagi menjadi dua bagian, yaitu :

- a. *Core Libraries*, yaitu aplikasi *Android* dibangun dalam bahasa *Java*, sementara *Dalvik* sebagai mesin virtualnya bukan *Virtual Machine Java*, sehingga diperlukan sebuah *libraries* yang berfungsi untuk menerjemahkan bahasa *Java* atau *C* yang ditangani oleh *Core Libraries*.
- b. *Dalvik Virtual Machine*, yaitu mesin virtual berbasis *register* yang dioptimalkan untuk menjalankan fungsi-fungsi secara efisien, dan merupakan pengembangan yang mampu membuat *Linux kernel* untuk melakukan *threading* dan manajemen tingkat rendah.

#### **2.3.2.5 Linux Kernel**

*Linux Kernel* adalah *layer* inti dari system operasi *Android* itu berada. Berisi *file-file system* yang mengatur *system processing*, *memory*, *resource*, *drivers*, dan sistem-sistem operasi *Android* lainnya. *Linux kernel* yang digunakan *Android* adalah *Linux Kernel release 2.6* (Safaat, 2012).

### **2.3.3 Komponen Aplikasi Android**

Aplikasi *Android* ditulis dalam Bahasa Pemrograman *Java*. Kode *Java* dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi,

prosesnya di-*package* oleh *tools* yang dinamakan “*apt tools*” kedalam paket *Android* sehingga menghasilkan *file* dengan ekstensi *.apk (dot apk)*. *File apk* ini yang disebut dengan aplikasi, dan kemudian dapat di-*install* di perangkat *Android*.

### **2.3.3.1 Activities**

Suatu *activity* menyajikan *user interface (UI)* kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi *Android* mungkin hanya terdapat satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface (UI)* saat aplikasi diperlihatkan oleh pengguna. Untuk pindah dari satu *activity* ke *activity* yang lainnya, dapat dilakukan dengan satu *even*, misalnya klik tombol, memilih opsi, atau menggunakan *trigger* tertentu. Secara hirarki sebuah *windows activity* dinyatakan dengan *method* `Activity setContentView()`. `ContentView` adalah objek yang berada pada *root* hirarki.

### **2.3.3.2 Service**

*Service* tidak memiliki *Graphic User Interface (GUI)*, tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* harus berada dalam kelas induknya. Misalnya *media player* sedang memutar lagu dari *list* yang ada, aplikasi ini memiliki dua atau lebih *activity* yang memungkinkan pengguna untuk memilih lagu, atau menulis SMS sambil *media player* tetap

berjalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service*. *Service* dijalankan pada *thread* utama dari proses aplikasi.

### **2.3.3.3 Broadcast Receiver**

*Broadcast Provider* berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. Sebagai contoh *broadcast* seperti notifikasi zona waktu berubah, baterai *low*, gambar telah selesai diambil oleh kamera, atau perubahan referensi bahasa yang digunakan.

*Broadcast receiver* tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang diterima, atau mungkin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau getaran perangkat.

### **2.3.3.4 Content Provider**

*Content Provider* membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file system* seperti *database SQLite*. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*, misalnya ketika menggunakan aplikasi yang membutuhkan peta (*map*), atau aplikasi yang membutuhkan untuk mengakses data kontak dan navigasi, maka inilah fungsi *Content Provider* untuk menyediakan data yang diperlukan oleh aplikasi (Safaat, 2012).

### 2.3.4 Distribusi Android

Perangkat *mobile* yang pertama menggunakan sistem operasi *Android* untuk menjalankan perangkatnya adalah *HTC Dream*, yang dirilis pada 22 Oktober 2008. Saat ini, 900 juta orang diseluruh dunia telah menggunakan perangkat *mobile* berbasis *Android*, baik itu *smartphone* maupun *tablet PC* (Lam, 2013).

*Android* menggunakan penamaan yang cukup unik dan menarik bagi setiap versinya, yaitu menggunakan nama-nama makanan penutup (*desert*) yang bercita rasa manis. Berikut versi-versi *Android* yang dapat menjalankan aplikasi Penyedia Jasa Online, antara lain:

1. Android versi 4.0 (*ICS: Ice Cream Sandwich*)

Pada 19 Oktober 2011 Android versi ini diluncurkan untuk *smartphone* dan menambahkan fitur baru, yaitu membuka kunci dengan pengenalan wajah, jaringan data pemantauan penggunaan dan kontrol, terpadu kontak jaringan sosial, perangkat tambahan fotografi, mencari email secara offline, dan berbagi informasi dengan menggunakan NFC.

2. Android versi 4.1 (*Jelly Bean*)

Penambahan fitur baru pada versi ini adalah peningkatan *input keyboard*, desain baru fitur pencarian, *user interface* yang baru dan pencarian melalui *voice search*.

3. Android versi 4.4 (*Kitkat*)

Pembaharuan fitur baru pada versi *Kitkat* ini adalah peningkatan *user interface* yang baru dalam notifikasi ponsel, fitur NFC, fitur mencetak secara online serta penambahan fitur yang lainnya.

#### 4. Android Versi 5.0 (*Lollipop*)

Pembaruan utama terbaru versi Android adalah *Lollipop* 5.0, yang dirilis pada 3 November 2014. *Lollipop* adalah update Android paling besar dan ambisius dengan lebih dari 5.000 API baru untuk para *developer*. Perangkat yang menggunakan OS Android L ini akan mampu berintegrasi antar perangkat seperti *smartphone*, tablet, dan *smartwatch* berbasis Android (Developers, 2014).

### 2.4 *Android Development Tools*

Dalam proses merancang dan membangun aplikasi sandi *semaphore* para perangkat *mobile* berbasis *Android*, diperlukan beberapa *tools* sebagai berikut:

#### 2.4.1 *Android Software Development Kit (SDK)*

*Android* SDK merupakan *tools* API (*Application Programming Interface*) yang diperlukan untuk mengembangkan aplikasi *Android* dengan menggunakan Bahasa Pemrograman *Java* (Meier, 2009).

#### 2.4.2 *Android Development Tools (ADT)*

*Android Development Tools* merupakan suatu *plugin* yang dirancang untuk *IDE Eclipse* agar mempermudah dalam membangun dan mengembangkan suatu aplikasi *Android* pada *IDE Eclipse*. Dengan *ADT*, pengembang dapat mem-*package source* menjadi satu *file installer Android*, yaitu *file* dengan ekstensi *.apk* (*dot apk*). Semakin tinggi *platform* atau versi *Android* yang

digunakan, maka semakin tinggi pula ADT yang digunakan (Meier, 2009).

### **2.4.3 *Android Studio***

Android Studio adalah sebuah *Integrated Development Environment*(IDE) untuk mengembangkan aplikasi pada *platform* Android. Android Studio diumumkan pada 16 Mei 2013 pada konferensi Google I/O oleh Product Manager Google, Ellie Powers. Android Studio tersedia secara bebas di bawah Apache License 2.0. Android Studio berada di awal tahap akses *preview* mulai dari versi 0.1 pada Mei 2013, kemudian memasuki tahap beta mulai dari versi 0.8 yang dirilis pada Bulan Juni 2014. Android Studio dengan kemampuan yang stabil dirilis pada Bulan Desember 2014, mulai dari versi 1.0. Android Studio tersedia untuk diunduh pada Windows, Mac OS X dan Linux.

#### **2.4.3.1 Fitur Android Studio**

Android Studio memiliki beberapa fitur yang akan memudahkan pengembang aplikasi pada *platform* Android.

- 1) Tata Letak Langsung yaitu *Editor - Live Coding - Real-time App Rendering*.
- 2) *Developer Console* yang meliputi tips optimasi, bantuan untuk penerjemahan, pelacakan rujukan, metrik penggunaan.
- 3) Penyisihan beta rilis dan peluncuran.
- 4) *Gradle-based build support*.
- 5) *Android-specific refactoring* dan perbaikan cepat.



- 6) *Lint tools* untuk mengoptimalkan kinerja, kegunaan, versi kompatibilitas dan masalah lainnya.
- 7) Kemampuan *ProGuard* dan *app-signing*.
- 8) *Template-based wizards* untuk membuat desain dan komponen dasar Android.
- 9) *Layout editor* yang memungkinkan pengguna untuk melakukan *drag and drop* komponen UI, pilihan untuk melihat tata letak pada beberapa konfigurasi layar.
- 10) Mendukung pengembangan aplikasi pada Android Wear
- 11) *Built-in* pendukung untuk Google Cloud Platform yang memungkinkan integrasi dengan Google Cloud Messaging dan App Engine.

#### 2.4.3.2 Kebutuhan Instalasi Android Studio

Kebutuhan yang harus dipenuhi pada perangkat komputer yang digunakan untuk dapat menjalankan Android Studio.

Tabel 2.1 Kebutuhan Instalasi Android Studio

	Windows	OS X	GNU/Linux
Versi Sistem Operasi	Microsoft Windows 8/7/Vista/2003 (32 atau 64 bit)	Mac OS X 10.8.5 atau versi yang lebih baru (Yosemite)	GNOME atau KDE atau Unity desktop dalam Ubuntu atau Fedora
RAM	2 GB RAM minimum, 4 GB RAM direkomendasikan	2 GB RAM minimum, 4 GB RAM direkomendasikan	2 GB RAM minimum, 4 GB RAM Direkomendasikan
Kapasitas Penyimpanan	500 MB	500 MB	500 MB

Tabel 2.1 Kebutuhan Instalasi Android Studio (lanjutan)

Kebutuhan Memori	Paling sedikit 16 GB untuk Android SDK, <i>emulator system images</i> , dan <i>caches</i>	Paling sedikit 16 GB untuk Android SDK, <i>emulator system images</i> , dan <i>caches</i>	Paling sedikit 16 GB untuk Android SDK, <i>emulator system images</i> , dan <i>caches</i>
Versi Java	Java Development Kit (JDK) 7 atau yang lebih baru	Java Development Kit (JDK) 7 atau yang lebih baru	Java Development Kit (JDK) 7 atau yang lebih baru
Resolusi Layar	1280x800 resolusi layar minimum	1280x800 resolusi layar minimum	1280x800 resolusi layar minimum

(<http://developer.android.com/tools/studio/index.html>, diakses tanggal 14 November 2016)

## 2.5 Metodologi Pengembangan Sistem

Metodologi pengembangan sistem yang digunakan yaitu *Unified Process* (UP) dan desain menggunakan *Unified Modeling Language* (UML).

### 2.5.1 *Unified Modeling Language* (UML)

*Unified Modeling Language* (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OOP). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang-orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang-bangun perangkat lunak efektif.


*Unified Modeling Language* merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat standar-standar yang mendukung interoperabilitas, khususnya interoperabilitas sistem berorientasi objek. OMG mungkin lebih dikenal dengan standar-standar COBRA (*Common Object Request Broker Architecture*).

UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corp.* UML menyediakan notasi-notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Fowler, 2004).





### 2.5.1.1 Use Case Diagram

*Use Case Diagram* adalah diagram yang mendeskripsikan interaksi antara pengguna dengan aplikasi. Kesimpulannya *use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem (Adi, 2015). Simbol dan keterangan *use case* pada Tabel 2.2.

Tabel 2.2. Simbol dan Keterangan *Use Case*

SIMBOL	KETERANGAN
Aktor 	Mewakili peran orang, system yang lain atau alat ketika berkomunikasi dengan <i>use case</i> .


Tabel 2.2. Simbol dan Keterangan *Use Case* (lanjutan)

<b>SIMBOL</b>	<b>KETERANGAN</b>
	Abstraksi dari interaksi antara sistem dan actor.
Association	Abstraksi dari penghubung antara actor dan <i>use case</i> .
Generalisasi 	Menunjukkan spessialisasi actor untuk dapat berpartisipasi dalam <i>use case</i> .
Extend 	Mempesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
Include 	Menunjukkan bahwa suatu <i>use case</i> seluruhnya merupakan fungsionalitas dari <i>use case</i> lainnya.




### 2.5.1.2 Activity Diagram

*Activity Diagram* atau Diagram Aktivitas menggambarkan alur aktivitas dalam aplikasi, menjelaskan proses masing-masing alur berawal dan proses aplikasi berakhir. Diagram aktivitas juga menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi (Adi, 2015). Simbol dan keterangan *activity diagram* seperti pada Tabel 2.3.

Tabel 2.3. Simbol dan Keterangan *Activity Diagram*

<b>SIMBOL</b>	<b>KETERANGAN</b>
Status Awal 	Status awal aktivitas system, sebuah diagram aktivitas memiliki sebuah status awal.



Tabel 2.3. Simbol dan Keterangan *Activity Diagram* (lanjutan)

SIMBOL	KETERANGAN
Aktivitas 	Aktivitas yang dilakukan system, aktivitas biasanya diawali dengan kata kerja.
Percabangan /decision 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Status akhir 	Status akhir yang dilakukan system, sebuah diagram aktivitas memiliki sebuah status akhir.



### 2.5.1.3 Class Diagram

*Class Diagram* atau Diagram Kelas merupakan diagram yang memodelkan sekumpulan kelas, *interface*, kolaborasi, dan relasinya. Diagram kelas digambarkan dengan bentuk kotak (Adi, 2005). Simbol dan keterangan *class diagram* seperti pada Tabel 2.4.

Tabel 2.4. Simbol dan keterangan *Class Diagram*

SIMBOL	KETERANGAN
CLASS 	Himpunan dari objek-objek yang berbagai atribut serta operasi yang sama.
Nary Association 	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.

Tabel 2.4. Simbol dan keterangan *Class Diagram* (lanjutan)


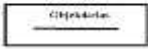
SIMBOL	KETERANGAN
	Hubungan dimana objek anak (descendent) berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk(ancestor).
	Operasi yang benar-benar dilakukan oleh suatu objek.

#### 2.5.1.4 Sequence Diagram








*Sequence Diagram* biasanya digunakan untuk tujuan analisa dan desain, memfokuskan pada identifikasi metode di dalam sebuah sistem (Adi, 2015).

Simbol dan keterangan *sequence diagram* seperti pada Tabel 2.5.

Tabel 2.5. Simbol dan keterangan *Sequence Diagram*

SIMBOL	KETERANGAN
	Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.
<b>OBJEK</b> 	Sebuah objek: Berpartisipasi secara berurutan dengan mengirimkan atau menerima pesan dan ditempatkan di bagian atas diagram.

Tabel 2.5. Simbol dan keterangan *Sequence Diagram* (lanjutan)

SIMBOL	KETERANGAN
<p>Garis hidup objek</p> 	<p>Menandakan kehidupan obyek selama urutan dan diakhiri tanda X pada titik dimana kelas tidak lagi berinteraksi.</p>
<p>Objek sedang aktif berinteraksi</p> 	<p>Fokus kontrol: Persegi panjang yang sempit panjang ditempatkan diatas sebuah garis hidup dan menandakan ketika suatu objek mengirim atau menerima pesan.</p>
<p>pesan</p> 	<p>Objek mengirim satu pesan ke objek lainnya.</p>
<p>Create</p> 	<p>Menyatakan suatu objek membuat objek yang lain,arah panah mengarah pada objek yang dibuat.</p>
<p>Masukan</p> 	<p>Menyatakan bahwa suatu objek mengirimkan masukan ke objek lainnya arah panah mengarah pada objek yang dikirim.</p>
<p>Keluaran</p> 	<p>Objek metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian.</p>
<p>Destroy</p> 	<p>Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah objek yang diakhiri.</p>

## 2.6 Firebase

Firestore adalah penyedia layanan *cloud* dengan *backend* sebagai servis yang berbasis di San Fransisco, California. Perusahaan ini membuat sejumlah produk untuk pengembangan aplikasi *mobile* ataupun *web*. Firestore didirikan oleh Andrew Lee dan James Tamplin pada tahun 2011 dan diluncurkan dengan *cloud database* secara *realtime* di tahun 2012. Produk utama dari Firestore yakni suatu database yang menyediakan API untuk memungkinkan pengembang menyimpan dan mensinkronisasi data lewat *multiple client*. Perusahaan ini diakuisisi oleh Google pada Oktober 2014.

Beberapa fitur yang dimiliki oleh Firestore adalah sebagai berikut:

1. *Firestore Analytics.*
2. *Firestore Cloud Messaging dan Notifications.*
3. *Firestore Authentication.*
4. *Firestore Remote Config.*
5. *Firestore Real Time Database.*
6. *Firestore Crash Reporting*

### 2.6.1 *Firestore Cloud Messaging and Notifications*

*Firestore Cloud Messaging/FCM*, adalah layanan yang diberikan oleh Firestore untuk menggantikan *Google Cloud Messaging (GCM)*. Pihak Google menyarankan untuk aplikasi yang masih menggunakan GCM untuk segera migrasi ke FCM.



Fitur-fitur yang diberikan oleh GCM sebenarnya tidak terlalu jauh berbeda dengan GCM. Dengan FCM kita bisa memberikan push notification dan membuat komunikasi dua arah antara *device*.

### **2.6.2 *Firestore Real Time Database***

Firestore menyediakan *realtime database* dan *backend* sebagai layanan. Suatu aplikasi layanan yang memungkinkan pengembang membuat API untuk disinkronisasikan untuk *client* yang berbeda – beda dan disimpan pada ‘cloud’-nya Firestore. Firestore memiliki banyak *library* yang memungkinkan untuk mengintegrasikan layanan ini dengan Android, iOS, Javascript, Java, Objective-C, dan Node.JS. Database Firestore juga bisa diakses lewat REST API dan data *binding* untuk beberapa framework Javascript seperti halnya AngularJS, ReactJS, Ember.JS, dan Backbone.JS. REST API tersebut menggunakan protokol *Server-Sent Event* dengan membuat koneksi HTTP untuk menerima *push notification* dari server. Pengembang juga bisa menggunakan database ini untuk mengamankan data mereka menggunakan server Firestore dengan *rules* yang ada.

## **2.7 Pengujian Perangkat Lunak**

Pengujian perangkat lunak adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum (Clune, 2011). Singkat kata, pengujian adalah aktivitas untuk menemukan dan menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya.

Pendekatan *Black-Box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Teknik pengujian yang digunakan dalam penelitian ini terbagi atas pengujian fungsional dengan menggunakan metode *Black Box* yaitu *Equivalence Partitioning*.

### **2.7.1 *Equivalence Partitioning***

*Equivalence Partitioning*(EP) merupakan metode *Black Box testing* yang membagi domain masukan dari program kedalam kelas-kelas sehingga *test case* dapat diperoleh. *Equivalence Partitioning* berusaha untuk mendefinisikan kasus uji yang menemukan sejumlah jenis kesalahan, dan mengurangi jumlah kasus uji yang harus dibuat. Kasus uji yang didesain untuk *Equivalence Partitioning* berdasarkan pada evaluasi dari kelas ekuivalensi untuk kondisi masukan yang menggambarkan kumpulan keadaan yang valid atau tidak. Kondisi masukan dapat berupa spesifikasi nilai numerik, kisaran nilai, kumpulan nilai yang berhubungan atau kondisi *Boolean*. Kesetaraan kelas dapat didefinisikan menurut panduan berikut (Pressman, 2001):

1. Jika masukan kondisi menentukan kisaran, satu sah dan dua diartikan tidak valid kesetaraan kelas.
2. Jika masukan membutuhkan nilai, kondisi tertentu satu sah dan dua tidak valid kesetaraan kelas diartikan.
3. Jika masukan kondisi menentukan anggota dari set, satu sah dan satu tidak valid kesetaraan kelas diartikan.
4. Jika kondisi yang input, *boolean* satu sah dan satu tidak valid kelas diartikan.

## **BAB III METODOLOGI PENELITIAN**

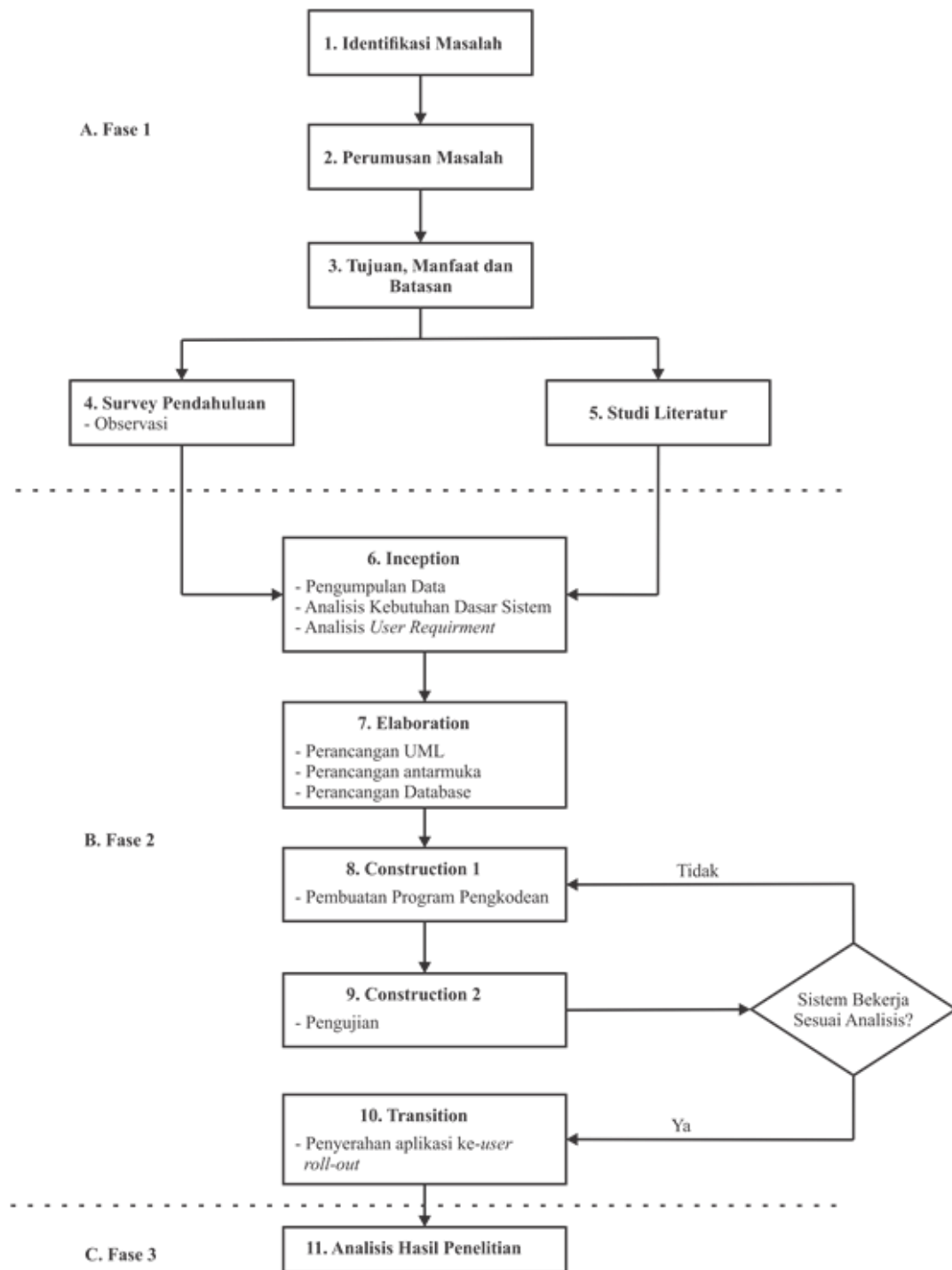
### **3.1 Waktu dan Tempat Penelitian**

Penelitian dilaksanakan di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Lampung yang berada di jalan Soemantri Brojonegoro No.1 Gedung Meneng, Bandar Lampung. Penelitian ini dilakukan pada Semester Ganjil Tahun Ajaran 2016-2017.

### **3.2 Metodologi Penelitian**

#### **3.2.1 Alur Penelitian**

Penelitian dilakukan berdasarkan diagram alir metodologi penelitian yang terdapat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

Penjelasan dari diagram alir Metodologi Penelitian pada Gambar 3.1 adalah sebagai berikut:

#### **A. Fase 1**

1. Tahap pertama pada langkah pertama adalah mengidentifikasi masalah, yaitu pengenalan masalah untuk menentukan masalah yang akan dipecahkan dalam penelitian.
2. Tahap ke-dua adalah perumusan masalah, yaitu pertanyaan penelitian yang membutuhkan jawaban melalui pengumpulan data yang umumnya disusun dalam bentuk kalimat tanya.
3. Tahap ke-tiga adalah menentukan tujuan, manfaat dan batasan. Target pencapaian dalam tahap ini adalah diketahuinya tujuan dan manfaat dari aplikasi penyedia jasa *online* berbasis mobile android. Sedangkan batasan digunakan untuk membatasi pembahasan dan ruang lingkup penelitian.
4. Tahap ke-empat adalah survei pendahuluan, yaitu untuk mengetahui hal-hal penting yang berhubungan dengan penelitian yang dikaji.
5. Tahap ke-lima adalah studi literatur, yaitu untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan sebagai teori-teori yang akan dijadikan landasan penelitian.

#### **B. Fase 2**

6. Tahap ke-enam pada fase ke-dua merupakan pengembangan sistem. Metode pengembangan sistem yang digunakan dalam penelitian ini adalah *Unified Process*. Tahap pertama yang dilakukan dalam

pengembangan sistem ini adalah *inception*. Pada tahap ini akan dilakukan pengumpulan data berupa data mengenai kategori-kategori maupun jenis-jenis jasa yang ada di kota Bandar Lampung. Pada tahap ini juga akan dilakukan analisis kebutuhan dasar sistem dan analisis *user requirement*.

7. Tahap ke-tujuh adalah *Elaboration*. Pada tahap ini akan dilakukan perancangan sistem dan perancangan *interface* yang menggunakan *Unified Modelling Language (UML)*, pada tahap ini akan diketahui semua entitas luar, input dan output yang terlibat dalam sistem serta *usecase diagram*, *classdiagram*, *activitydiagram*, *sequencediagram* yang digunakan dalam analisis sistem. Selanjutnya akan dilakukan perancangan *interface*, perancangan ini dilakukan untuk merancang tata letak sistem sesuai dengan analisis kebutuhan sistem.
8. Tahap ke-delapan adalah *construction 1*. Pada tahap ini akan dilakukan pembuatan program.
9. Tahap ke-sembilan adalah *construction 2*. Program yang telah selesai akan diuji menggunakan *Black Box* dengan metode *Equivalence Partitioning (EP)*. Jika sistem tidak bekerja sesuai analisis, maka kembali ke tahap *construction 1*, namun jika sistem bekerja sesuai analisis maka dilakukan tahap selanjutnya.
10. Tahap ke-sepuluh adalah *transition*. Pada tahap ini akan dilakukan penyerahan sistem aplikasi ke-*user (roll-out)* melalui PlayStore.

### C. Fase 3

11. Tahap ke-sebelas adalah analisis hasil penelitian. Pada tahap ini menjelaskan berhasil atau tidaknya dalam suatu penelitian dengan permasalahan yang telah dideskripsikan di awal.

#### 3.2.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang dipilih dalam penelitian ini adalah *Unified Process* (UP). Tahap-tahap yang dilakukan dalam pengembangan sistem aplikasi ini adalah:

##### 1) *Inception*

Pada tahap ini dilakukan pengumpulan data yang berupa data jenis-jenis layanan jasa di kota Bandar Lampung. Data ini diperoleh langsung dilapangan dengan melakukan observasi untuk mendapatkan jenis-jenis layanan jasa yang terdapat di kota Bandar Lampung. Data terdiri dari beberapa jenis layanan jasa. Adapun jenis-jenis layanan jasa di kota Bandar Lampung sebagai berikut:

1. Jasa Kontraktor
2. Jasa Pindahan
3. Jasa Kebersihan
4. Jasa Perbaikan
5. Jasa Sewa Barang

(Observasi Lapangan, Tahun 2016)

Pada tahap ini juga dilakukan analisis kebutuhan dasar sistem dan analisis *user requirement*.

## **2) Elaboration**

Perancangan sistem dan perancangan *interface* berlangsung pada tahap ini. Perancangan atau desain sistem dalam penelitian ini menggunakan *Unified Modelling Language* (UML). Diagram-diagram UML yang dibuat dalam penelitian ini adalah sebagai berikut:

- a. *Use case diagram*.
- b. *Activity diagram*.
- c. *Sequence diagram*.
- d. *Class diagram*.

Perancangan *interface* (antarmuka) yang dirancang dalam aplikasi ini adalah sebagai berikut:

- a. *Layout Splash Screen*.
- b. *Layout Menu Utama*.
- c. *Layout Notifikasi*
- d. *Layout Menu Cari Jasa*.
- e. *Layout Menu Cari Order*.
- f. *Layout Menu My Jasa*.
- g. *Layout Menu My Order*.

## **3) Construction 1**

Proses pembuatan program (*coding*) dilakukan pada tahap ini menggunakan Bahasa Pemrograman Java dengan bantuan aplikasi



Android Studio. Pada aplikasi ini dibuat beberapa *class* java yang menjelaskan jalannya aplikasi khususnya pada fungsi pencarian penyedia jasa dan order. Aplikasi ini juga membutuhkan akses internet pada perangkat Android. Setelah semua terpenuhi didalam *class* java, maka selanjutnya aplikasi akan menampilkan informasi mengenai penyedia jasa di Bandar Lampung untuk pengguna, yaitu dengan mengambil data yang telah disimpan dalam *database*.

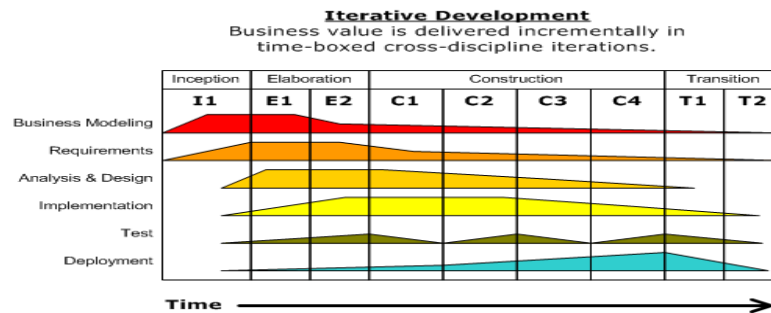
#### 4) *Construction 2*

Pada tahap ini, dilakukan pengujian *black box* dengan metode *Equivalence Partitioning*(EP).

#### 5) *Transition*

Play Store adalah layanan konten digital milik Google yang mencakup toko untuk produk-produk seperti musik/lagu, buku, aplikasi, permainan, ataupun pemutar media berbasis *cloud*. Pada tahap ini adalah tahap *upload* aplikasi penyedia jasa onloe di Bandar Lampung ke dalam Play Store, proses ini dilakukan agar pengguna dapat mengunduh aplikasi melalui Play Store secara gratis.

Diagram proses pengembangan sistem menggunakan *Unified Process* (UP) dapat dilihat pada gambar 3.2



Gambar 3.2 Diagram *Unified Process*

### **3.2.3 Metode Pengumpulan Data**

Metode pengumpulan data yang digunakan pada penelitian ini adalah:

1. Studi Literatur

Studi literatur yang digunakan yaitu buku-buku, jurnal, prosiding dan internet yang menyajikan informasi tentang *Android Studio*, *Android*, dan Bahasa Pemrograman Java.

2. Metode Observasi

Metode ini digunakan untuk mendapatkan informasi pengamatan yang bertujuan untuk mengumpulkan data berupa data kategori dan jenis-jenis layanan jasa yang ada di kota Bandar Lampung.

## **3.3 Analisis dan Perancangan Sistem**

### **3.3.1 Identifikasi Masalah**

Banyaknya penyedia jasa dan juga perusahaan saat ini membuat kita terkadang bingung untuk memilih dan menentukan penyedia jasa mana yang dapat membantu dan cocok dengan keinginan kita sebagai pengguna jasa. Disebabkan kadang kala belum tentu yang kita inginkan dapat dilaksanakan oleh pihak penyedia jasa, dan juga apakah tempat penyedia jasa yang akan kita gunakan masih menyediakan jasanya atau tidak. Dengan mempertimbangkan hal tersebut, penulis akan merancang dan membangun aplikasi yang dapat membantu masyarakat untuk mencari penyedia jasa yang tepat untuk membantu menyelesaikan masalah mereka, yaitu:

“Aplikasi Penyedia Jasa Online Berbasis Mobile Android”, dengan harapan membantu setiap masyarakat untuk mencari penyedia jasa yang tepat untuk menyelesaikan masalah mereka.

### **3.3.2 Analisis Sistem**

Berdasarkan identifikasi masalah tersebut, maka data yang dibutuhkan sistem adalah:

1. Data penyedia jasa di kota Bandar Lampung.
2. Data jasa perusahaan di kota Bandar Lampung.

### **3.3.3 Analisis *User Requirement***

Kebutuhan dasar aplikasi ini adalah sebagai berikut:

1. Sistem dapat menampilkan informasi layanan jasa yang berada di kota Bandar Lampung.
2. Sistem dapat meng-*order* dan menerima *order*-an jasa sesuai dengan kategori yang tersedia di dalam aplikasi.
3. Sistem menyediakan fitur *chatting* antar pemesan jasa dan penyedia jasa untuk saling berkomunikasi.
4. Sistem dapat merespon beberapa aksi yang diberikan oleh *user*.

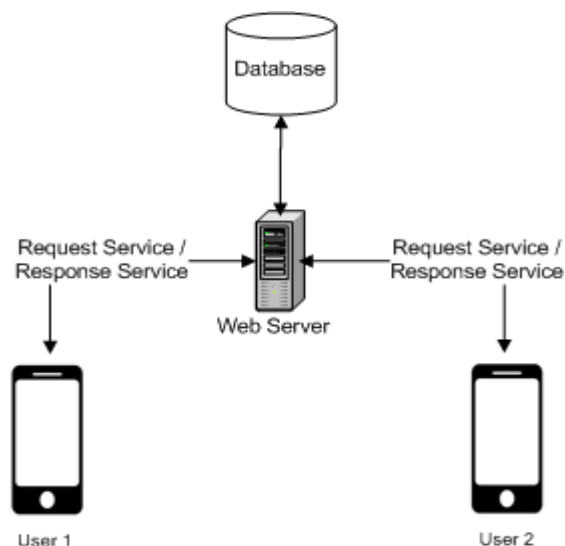
### **3.3.4 Perancangan Sistem**

Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem. Perancangan sistem berupa penggambaran, perencanaan, dan pembuatan

sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan dapat berfungsi. Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang harus diselesaikan. Tahap ini termasuk mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah dilakukan instalasi akan benar-benar sesuai dengan rancangan awal sistem.

#### 3.3.4.1 Desain Arsitektur

Teknologi yang digunakan dalam pembuatan sistem adalah teknologi *web service*. *Web Service* merupakan teknologi yang menyediakan integrasi proses dan data. Pada penelitian ini *web service* dibangun untuk menghubungkan aplikasi *mobile* android dan aplikasi web sebagai *client* dengan *database server*. Web service dan database data base pada aplikasi ini menggunakan Firebase. Dan data yang dikirim merupakan data dalam format JSON. Desain arsitektur web service dapat dilihat pada Gambar 3.3.



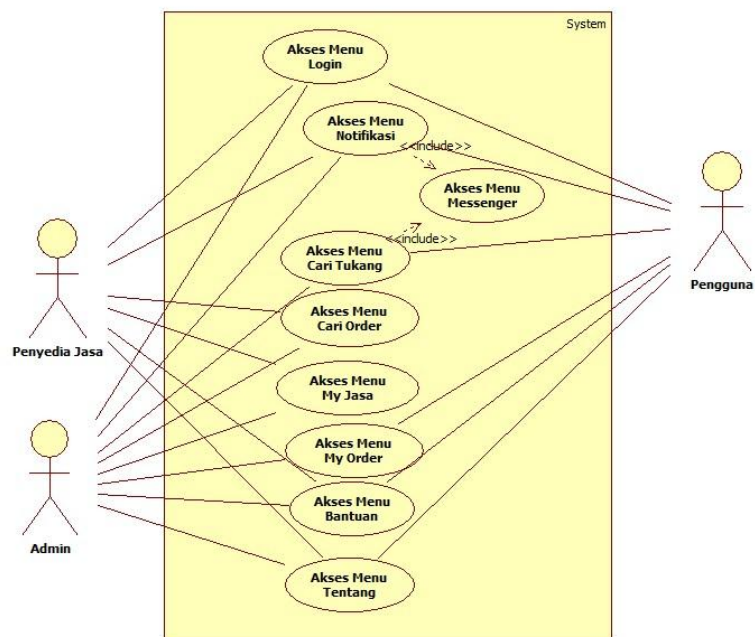
Gambar 3.3 Desain Arsitektur

### 3.3.4.2 Perancangan UML (*Unified Modelling Language*)

Perancangan sistem atau desain sistem dilakukan dengan memodelkan permasalahan dalam bentuk diagram-diagram UML sebagai berikut:

#### 1. *Use Case Diagram*

*Use case diagram* dibawah ini menjelaskan fungsionalitas dari aplikasi penyedia jasa online di Bandar Lampung. Pada aplikasi ini, pengguna dapat melakukan sembilan interaksi antara lain aksesmenu login, akses menu notifikasi, akses menu *messenger*, akses menu cari tukang, akses menu cari order, akses menu my jasa, akses menu my order, akses menu bantuan dan akses menu tentang aplikasi. *Use case diagram* aplikasi Cari Tukang dapat dilihat pada Gambar 3.4.



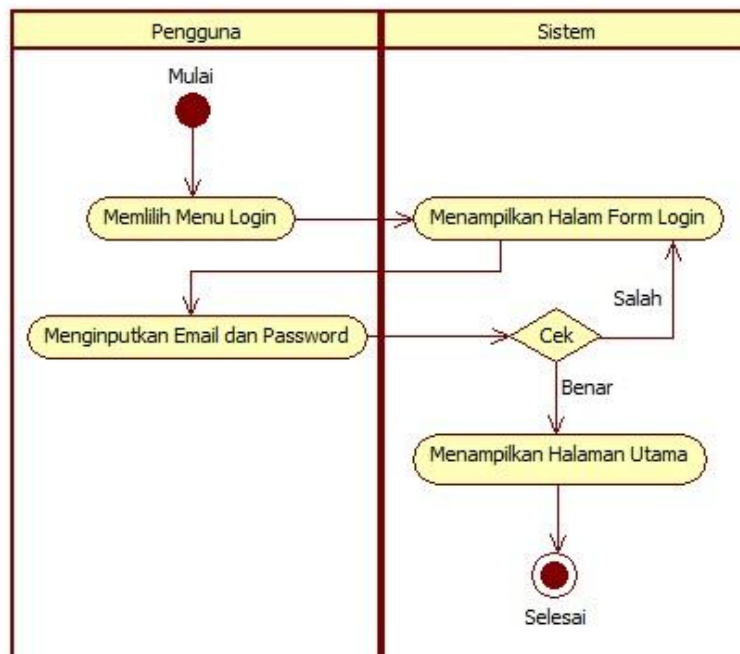
Gambar 3.4 *Use Case Diagram*

## 2. Activity Diagram

Diagram aktivitas atau *Activity Diagram* menggambarkan aliran fungsionalitas sistem. Pada aplikasi Cari Tukang terdapat delapan *activity diagram*, yaitu sebagai berikut:

### a) *Activity Diagram* Menu Login

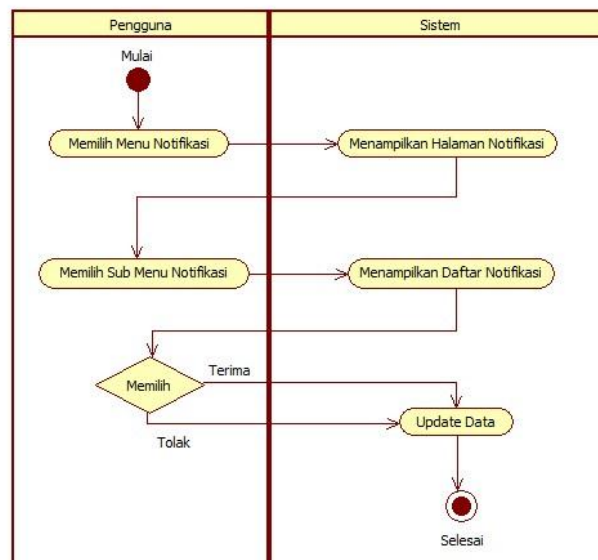
*Activity Diagram* Menu Login dimulai dengan pengguna memilih menu “Login” pada antarmuka aplikasi yang kemudian sistem akan menampilkan halaman *login*. Setelah itu pengguna diminta untuk memasukkan *email* dan *password*, yang kemudian sistem akan mengecek apakah id pengguna jika sesuai maka akan tampil halaman utama aplikasi. *Activity Diagram* Menu Login disajikan pada gambar Gambar 3.5.



Gambar 3.5. *Activity Diagram* Menu Login

b) *Activity Diagram* Notifikasi

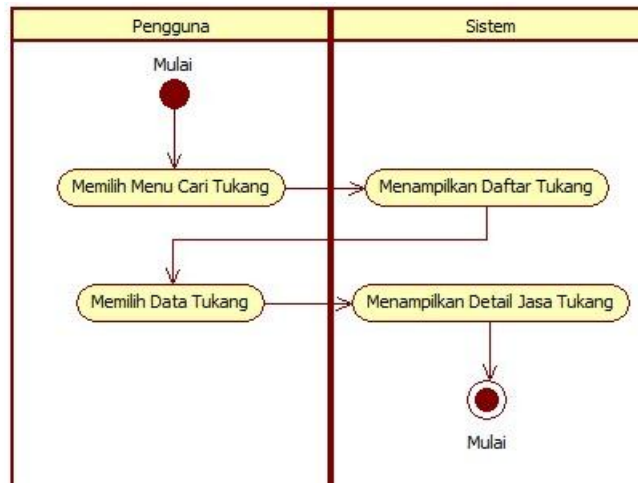
*Activity Diagram* Notifikasi dimulai dengan pengguna memilih menu “Notifikasi” kemudian sistem akan menampilkan halaman notifikasi. Setelah pengguna memilih submenu pada halaman notifikasi selanjutnya sistem akan menampilkan daftar notifikasi yang sesuai berdasarkan submenu yang dipilih sebelumnya. Pengguna dapat menerima dan menolak order. *Activity Diagram* Notifikasi disajikan pada Gambar 3.6.



Gambar 3.6. *Activity Diagram* Notifikasi

c) *Activity Diagram* Cari Tukang

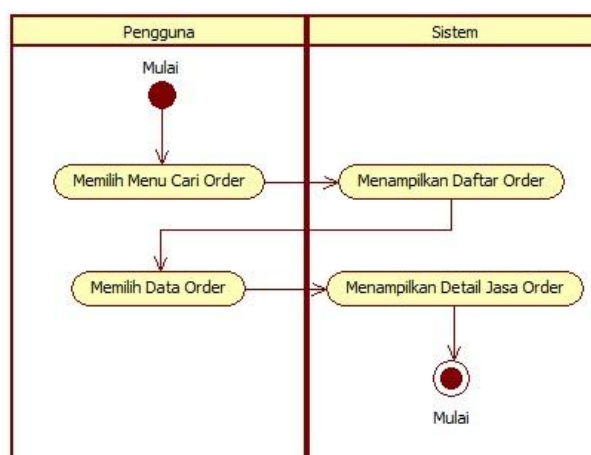
*Activity Diagram* Cari Tukang dimulai dengan pengguna memilih menu “Cari Tukang” kemudian sistem akan menampilkan daftar tukang. Pengguna dapat melihat detail informasi tukang dengan memilih salah satu data tukang. *Activity Diagram* Cari Tukang disajikan pada Gambar 3.7.



Gambar 3.7. *Activity Diagram* Cari Tukang

d) *Activity Diagram* Cari Order

*Activity Diagram* Cari Order dimulai dengan pengguna memilih menu “Cari Order” kemudian sistem akan menampilkan daftar order. Pengguna dapat melihat detail informasi order dengan memilih salah satu data order. *Activity Diagram* Cari Order disajikan pada Gambar 3.8.

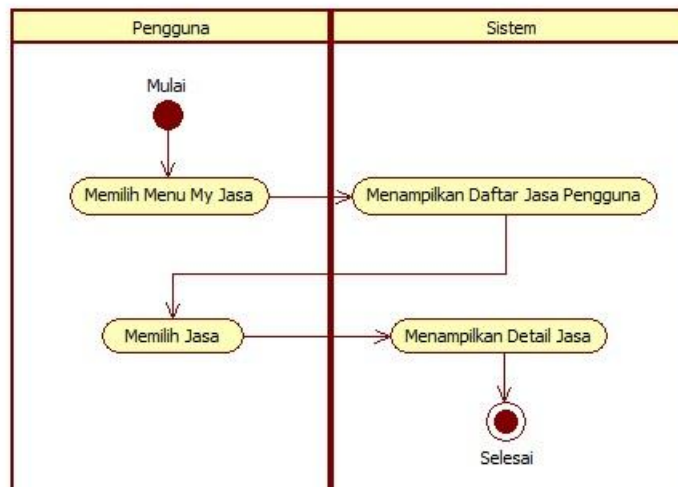


Gambar 3.8. *Activity Diagram* Cari Order



e) *Activity Diagram My Jasa*

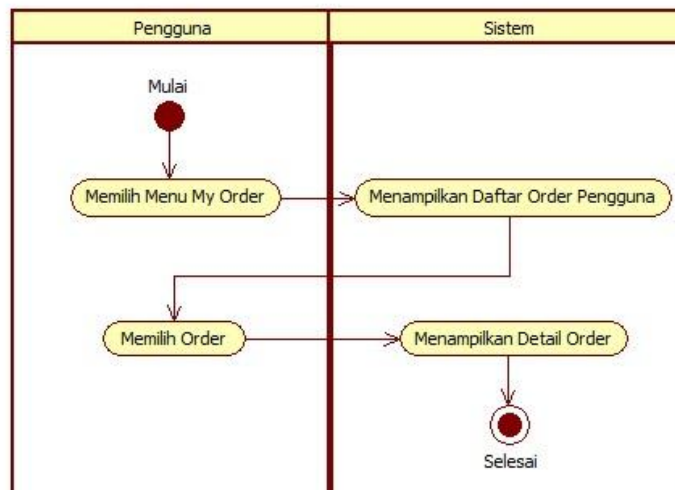
*Activity Diagram My Jasa* dimulai dengan pengguna memilih menu “My Jasa” yang kemudian sistem akan menampilkan daftar jasa pengguna. Pengguna dapat melihat detail jasa pengguna dengan memilih salah satu data jasa. *Activity Diagram My Jasa* disajikan dalam Gambar 3.9.



Gambar 3.9. *Activity Diagram Menu My Jasa*

f) *Activity Diagram My Order*

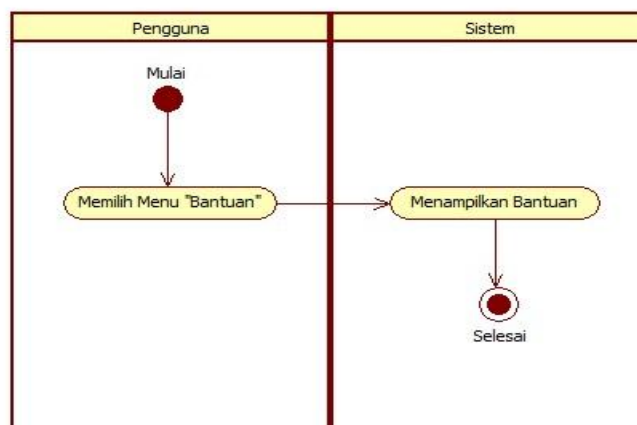
*Activity Diagram My Order* dimulai dengan pengguna memilih menu “My Order” yang kemudian sistem akan menampilkan daftar order pengguna. Pengguna dapat melihat detail order pengguna dengan memilih salah satu data order. *Activity Diagram My Order* disajikan dalam Gambar 3.10.



Gambar 3.10. *Activity Diagram My Order*

g) *Activity Diagram Bantuan*

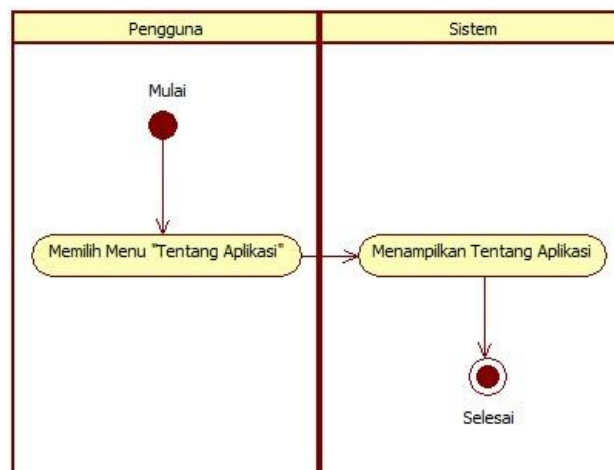
*Activity Diagram* Bantuan dimulai dengan pengguna memilih menu “Bantuan” yang kemudian sistem akan menampilkan bantuan mengenai penggunaan aplikasi Cari Tukang. *Activity Diagram* Bantuan disajikan dalam Gambar 3.11.



Gambar 3.11. *Activity Diagram Bantuan*

h) *Activity Diagram* Tentang Aplikasi

Menu “Tentang” memberikan informasi yang berkaitan dengan aplikasi Cari Tukang, yaitu untuk mengetahui tujuan dibuatnya aplikasi. *Activity diagram* menu “Tentang Aplikasi” disajikan pada Gambar 3.12.



Gambar 3.12. *Activity Diagram* Tentang Aplikasi

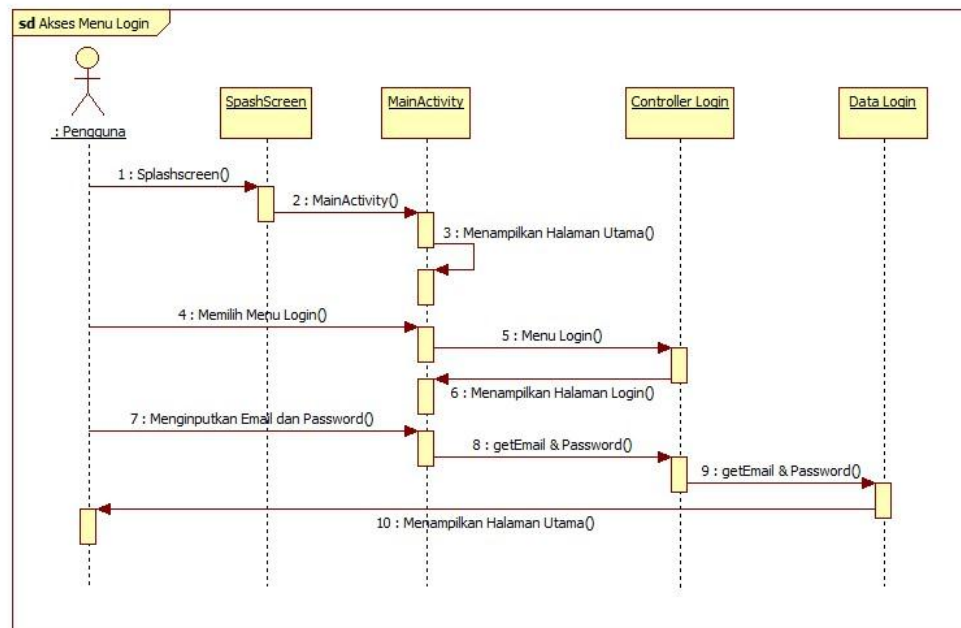
### 3. *Sequence Diagram*

Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Pada aplikasi Cari Tukang terdapat delapan *sequence diagram*, yaitu sebagai berikut.

a) *Sequence Diagram* Menu Login

Untuk dapat memilih menu “Login” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul halaman awal, kemudian pengguna dapat memilih menu “Login”, sistem akan menampilkan halaman login. Setelah itu pengguna diminta untuk

memasukkan *email* dan *password*, yang kemudian sistem akan mengecek apakah id pengguna jika sesuai maka akan tampil halaman utama aplikasi. Untuk lebih jelasnya *sequence diagram* menu “Login” disajikan pada Gambar 3.13.

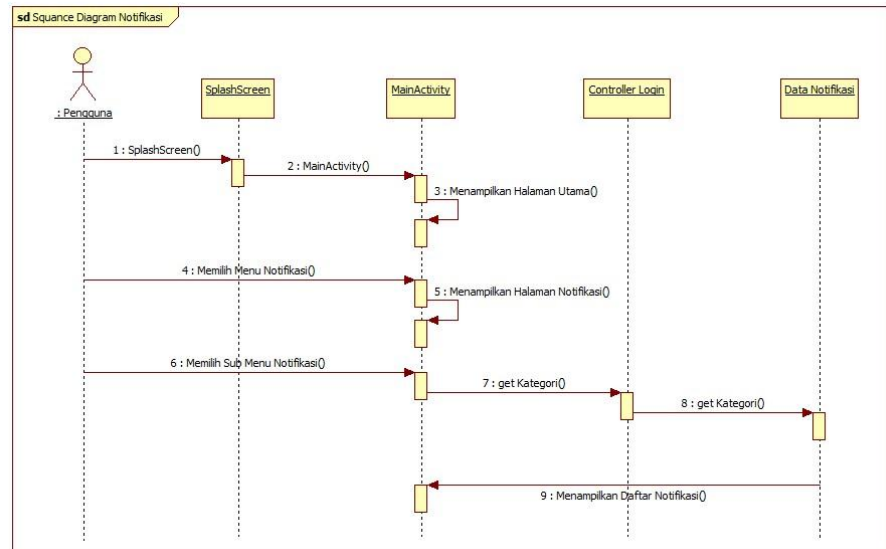


Gambar 3.13. *Sequence Diagram* Login

b) *Sequence Diagram* Notifikasi

Untuk dapat memilih menu “Notifikasi” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *plash screen*, selanjutnya akan muncul halaman utama. Kemudian pengguna memilih menu “Notifikasi” pada menu navigasi, selanjutnya sistem akan menampilkan halaman notifikasi. Setelah pengguna memilih submenu pada halaman notifikasi selanjutnya sistem akan menampilkan daftar notifikasi yang sesuai berdasarkan submenu yang dipilih sebelumnya. Pengguna dapat

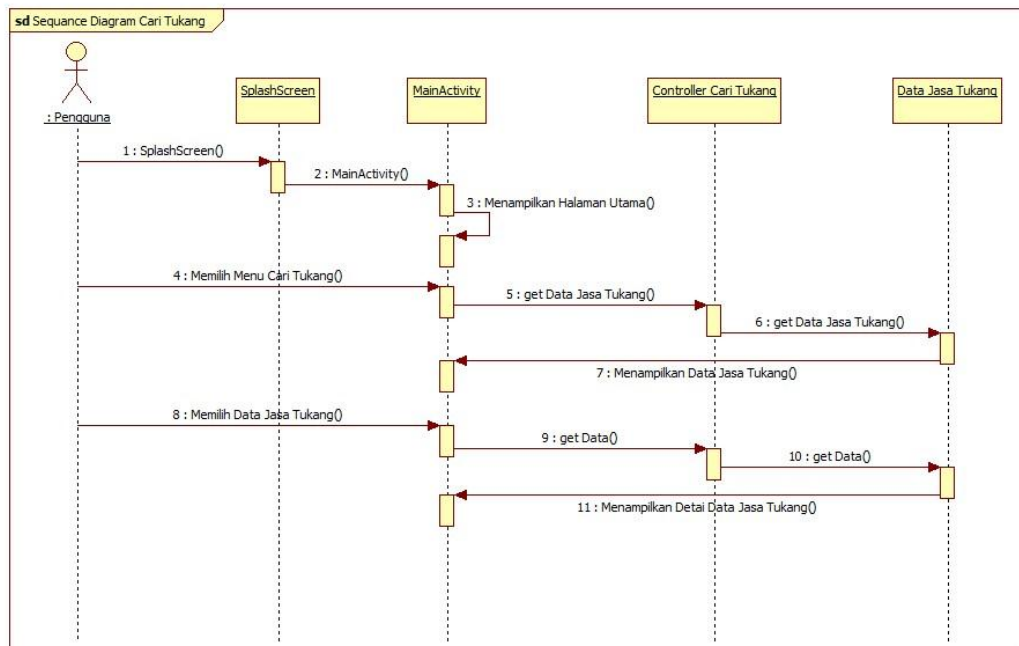
menerima dan menolak order. Untuk lebih jelasnya *sequence diagram* menu “Cari Jasa” disajikan pada Gambar 3.14.



Gambar 3.14. *Sequence Diagram* Notifikasi

c) *Sequence Diagram* Cari Tukang

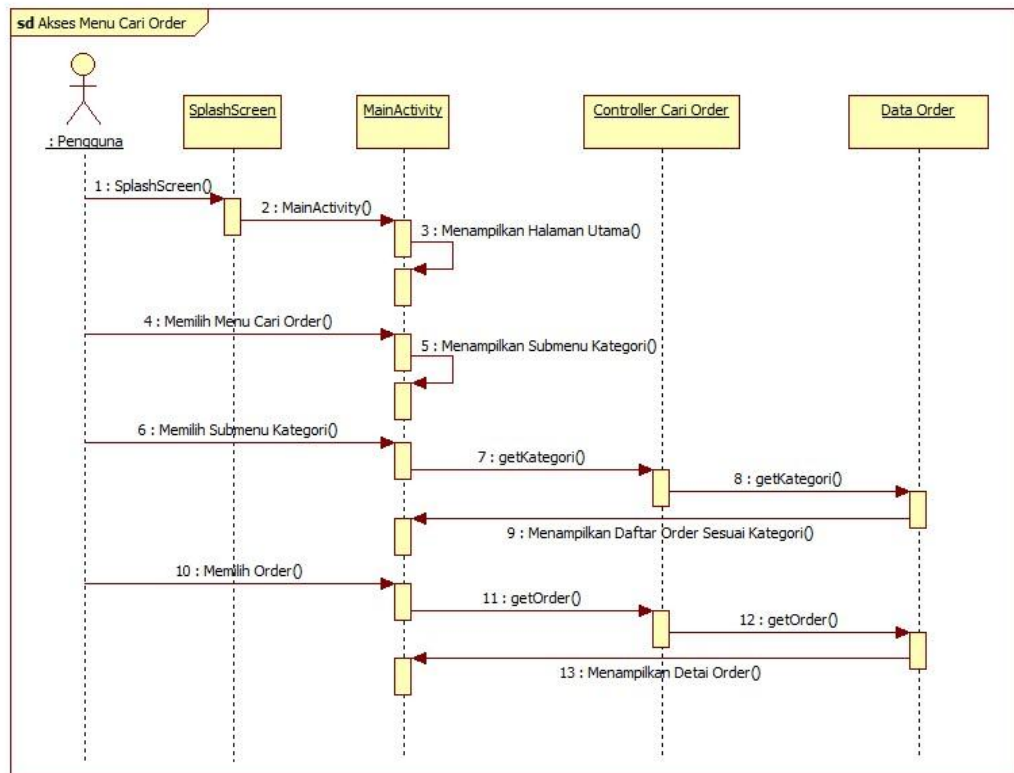
Untuk dapat memilih menu “Login” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul halaman utama. Kemudian pengguna memilih menu “Cari Tukang” kemudian sistem akan menampilkan daftar data jasa tukang. Pengguna dapat melihat detail informasi tukang dengan memilih salah satu data tukang. Untuk lebih jelasnya *sequence diagram* menu “Cari Tukang” disajikan pada Gambar 3.15.



Gambar 3.15. *Sequence Diagram* Cari Tukang

d) *Sequence Diagram* Cari Order

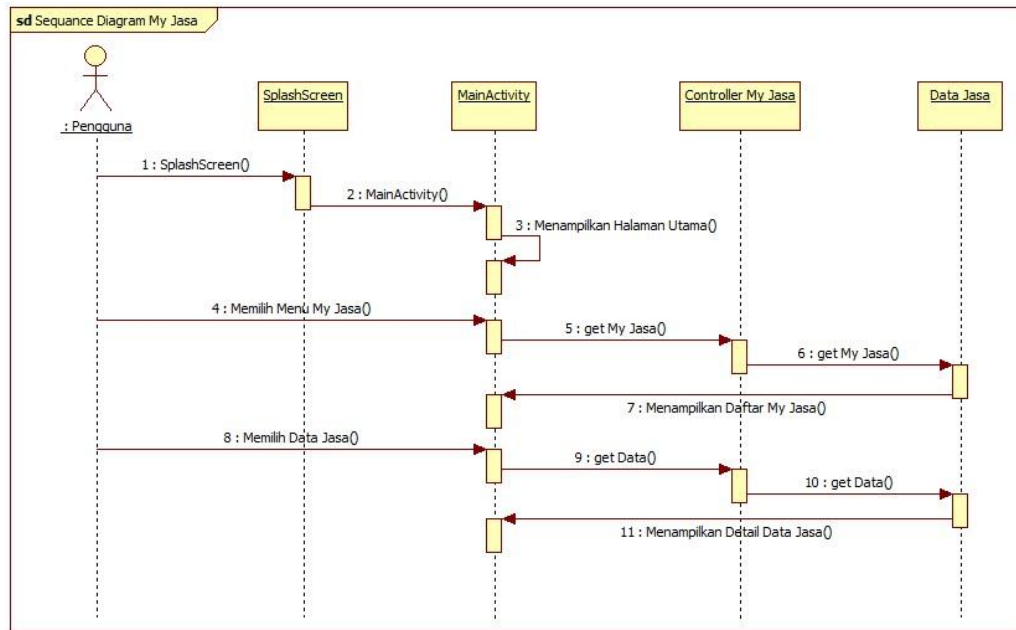
Untuk dapat memilih menu “Cari Order” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *plash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “Cari Order” pada menu navigasi, selanjutnya sistem akan menampilkan daftar order. Pengguna dapat melihat detail informasi order dengan memilih salah satu data order. Untuk lebih jelasnya *sequence diagram* menu “Cari Order” disajikan pada Gambar 3.16.



Gambar 3.16. *Sequence Diagram* Cari Order

e) *Sequence Diagram* My Jasa

Untuk dapat memilih menu “My Jasa” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “My Jasa” pada menu navigasi, selanjutnya sistem akan menampilkan daftar jasa pengguna. Pengguna dapat melihat detail jasa pengguna dengan memilih salah satu data jasa. Untuk lebih jelasnya *sequence diagram* menu “My Jasa” disajikan pada Gambar 3.17.

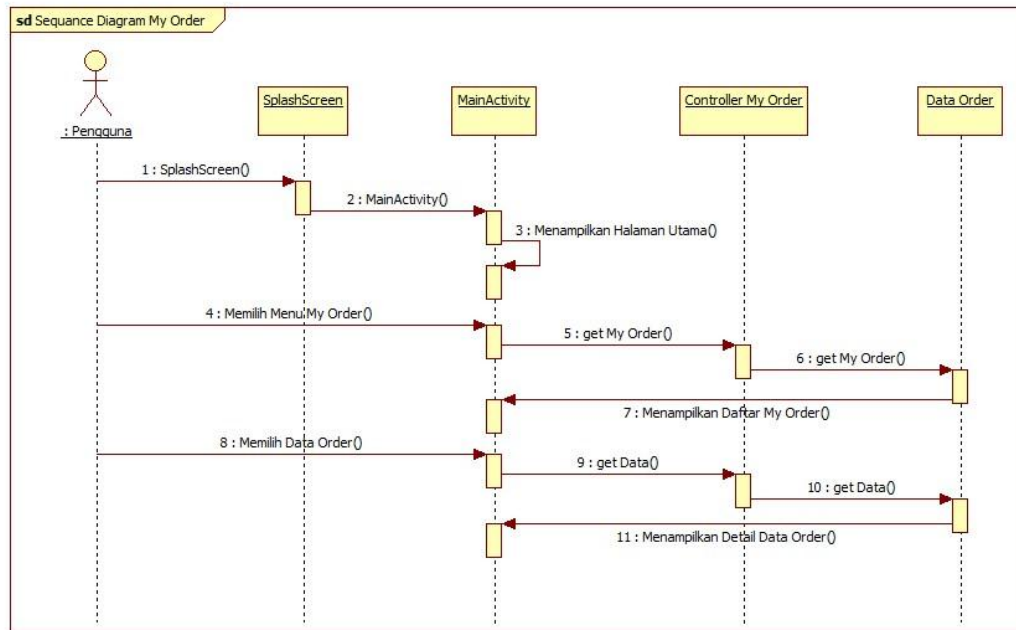


Gambar 3.17. *Sequence Diagram My Jasa*

f) *Sequence Diagram My Order*

Untuk dapat memilih menu “My Order” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, kemudian pengguna dapat memilih menu “My Order” pada menu navigasi, selanjutnya sistem akan menampilkan daftar order pengguna. Pengguna dapat melihat detail order pengguna dengan memilih salah satu data order. Untuk lebih jelasnya *sequence diagram* menu “My Order” disajikan pada Gambar 3.18.

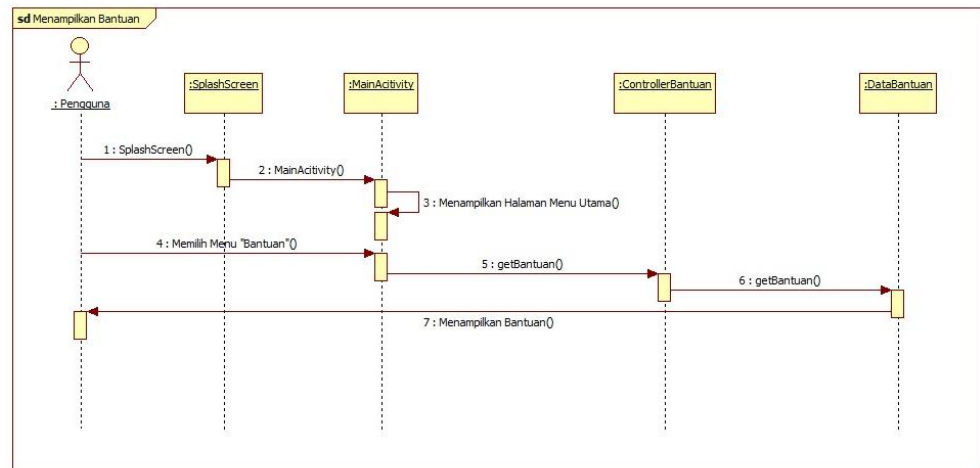




Gambar 3.18. *Sequence Diagram My Order*

g) *Sequence Diagram Bantuan*

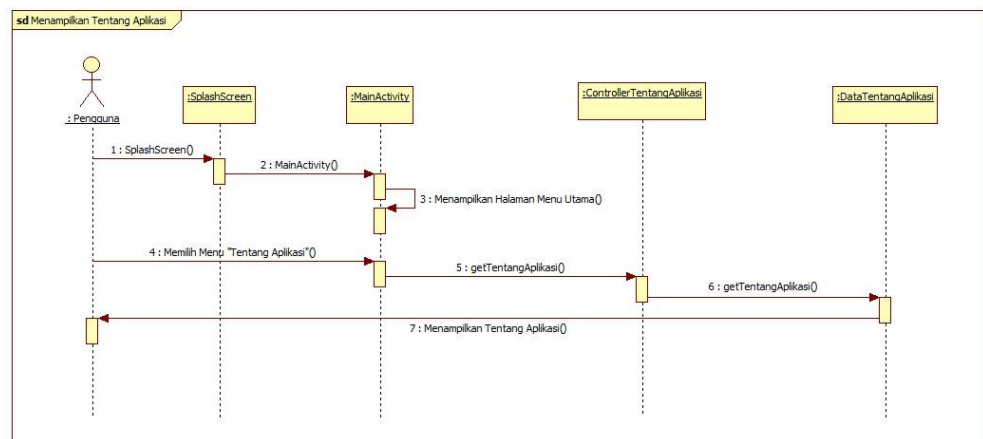
Untuk dapat memilih menu “Bantuan” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Bantuan”, maka sistem akan menampilkan bantuan tentang penggunaan aplikasi Cari Tukang. *Sequence diagram* menu “Bantuan” disajikan pada Gambar 3.19.



Gambar 3.19. *Sequence Diagram* Bantuan

h) *Sequence Diagram* Tentang Aplikasi

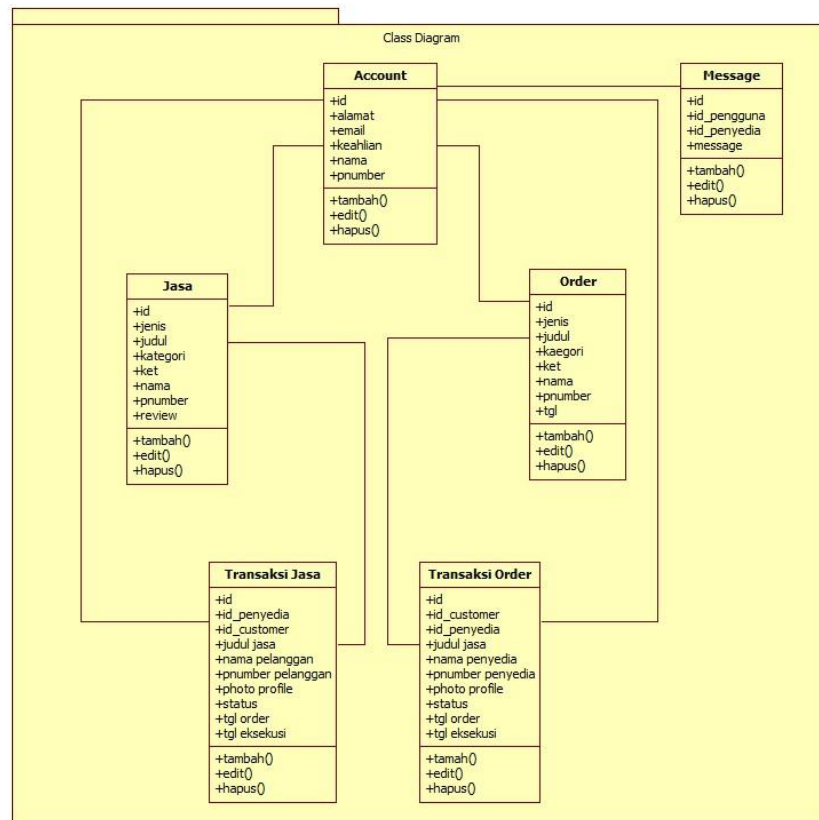
Untuk dapat memilih menu “Tentang Aplikasi” maka pengguna terlebih dahulu menjalankan aplikasi Cari Tukang kemudian masuk ke halaman *splash screen*, selanjutnya akan muncul menu utama aplikasi, dan pengguna dapat memilih menu “Tentang Aplikasi”, maka sistem akan menampilkan informasi tentang aplikasi Cari Tukang. *Sequence diagram* menu “Tentang Aplikasi” disajikan pada Gambar 3.20.



Gambar 3.20. *Sequence Diagram* Tentang Aplikasi

#### 4. Class Diagram

*Class Diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain. *Class diagram* pada aplikasi Cari Tukang yaitu sebagai berikut:



Gambar 3.21 *Class Diagram*

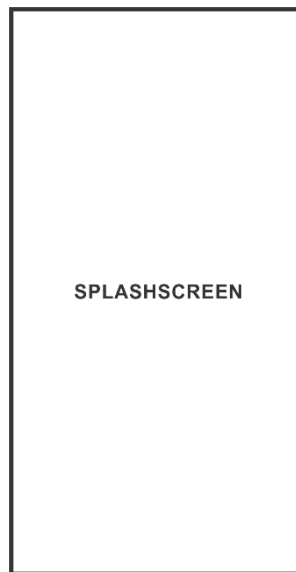
#### 3.3.4.3 Perancangan Antarmuka

Perancangan antarmuka merupakan proses penggambaran bagaimana sebuah tampilan (*interface*) sistem dibentuk. Aplikasi Cari Tukang dirancang dengan tampilan yang *user friendly*, sehingga diharapkan dapat mempermudah pengguna

dalam menggunakan aplikasi ini. Dalam Aplikasi ini terdapat beberapa *layout* atau *form*, antara lain:

1. *Layout Splash Screen*

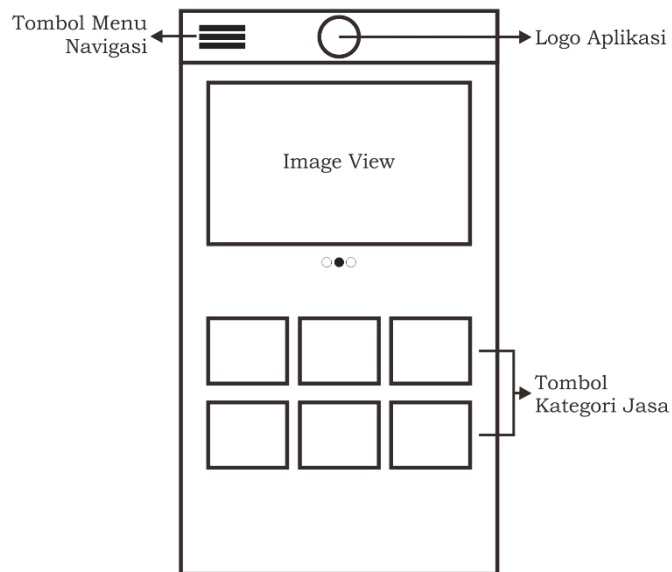
*Splash Screen* adalah tampilan awal ketika pengguna menjalankan aplikasi, *Splash screen* digunakan sebagai identitas aplikasi. Perancangan *layout splash screen* aplikasi Cari Tukang disajikan pada Gambar 3.22.



Gambar 3.22. *Design Layout Splash Screen*

2. *Layout Menu Utama*

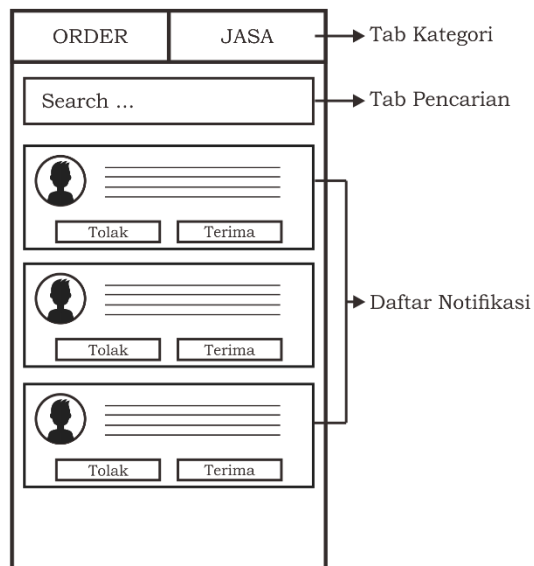
Menu utama pada aplikasi ini menampilkan menu-menu untuk pengguna, menu tersebut adalah menu cari tukang berdasarkan kategori, dan menu navigasi yang berisikan menu notifikasi, cari tukang, cari order, my jasa, dan my order. Perancangan *layout* menu utama aplikasi Cari Tukang disajikan pada Gambar 3.23.



Gambar 3.23. *Design Layout* Menu Utama

### 3. *Layout menu* Notifikasi

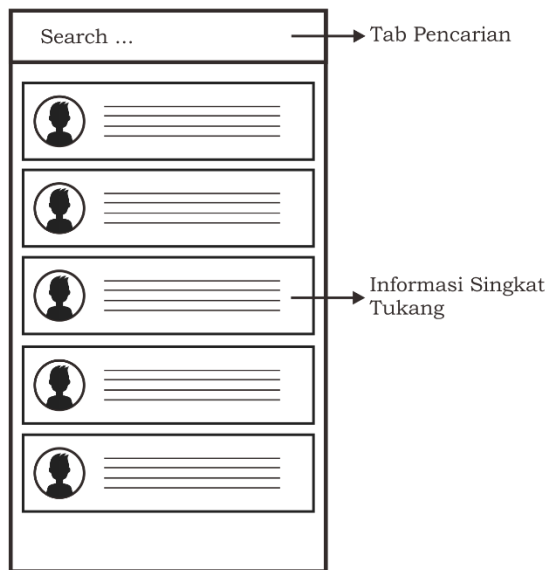
Pada halaman menu notifikasi daftar notifikasi pengguna yang dapat dikelompokkan berdasarkan kategori order atau jasa. Pengguna dapat menyaring daftar notifikasi berdasarkan kategori dengan memilih kategori yang diinginkan pada tab kategori. Dalam halaman ini juga tersedia tab pencarian untuk memudahkan pengguna dalam menemukan notifikasi yang diinginkan. Perancangan *layout tmenu* Notifikasi aplikasi Cari Tukang disajikan pada Gambar 3.24.



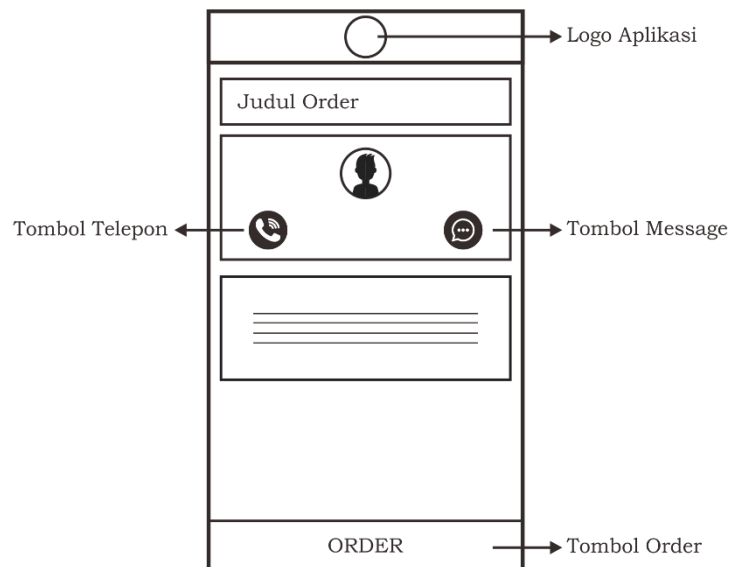
Gambar 3.24. *Layout menu* Notifikasi

#### 4. *Layout Menu* Cari Tukang

Pada menu ini akan ditampilkan daftar tukang yang tersedia pada aplikasi Cari Tukang. Pada tampilan ini pengguna juga dapat melihat detail jasa dengan cara memilih salah satu jasa. Dalam halaman ini juga tersedia tab pencarian untuk memudahkan pengguna dalam menemukan data tukang yang diinginkan. Perancangan *layout* menu “Cari Tukang” disajikan pada Gambar 3.25.



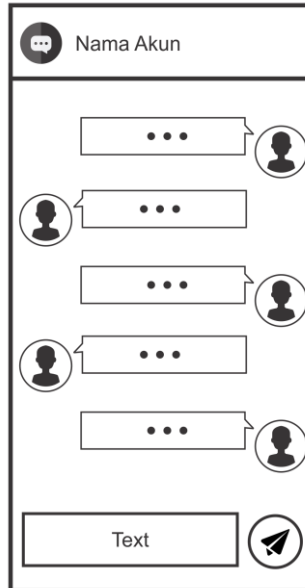
Gambar 3.25. *Design Layout* Menu Cari Tukang



Gambar 3.26. *Design Layout* Detail Tukang

Pada perancangan antarmuka detail tukang, pengguna dapat saling berkomunikasi dengan penyedia jasa dengan menggunakan fitur *Messenger* yang ada pada aplikasi. Pengguna juga dapat meng-order tukang dengan

menekan tombol “Order” Perancangan *layout* “*Messenger*” disajikan pada Gambar 3.27.

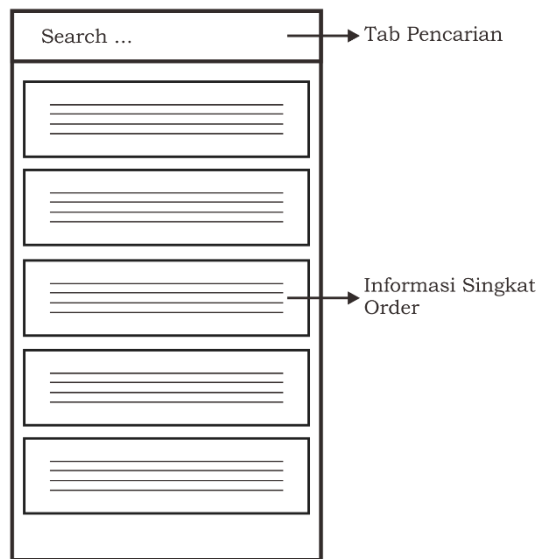


Gambar 3.27. *Design Layout Messenger*

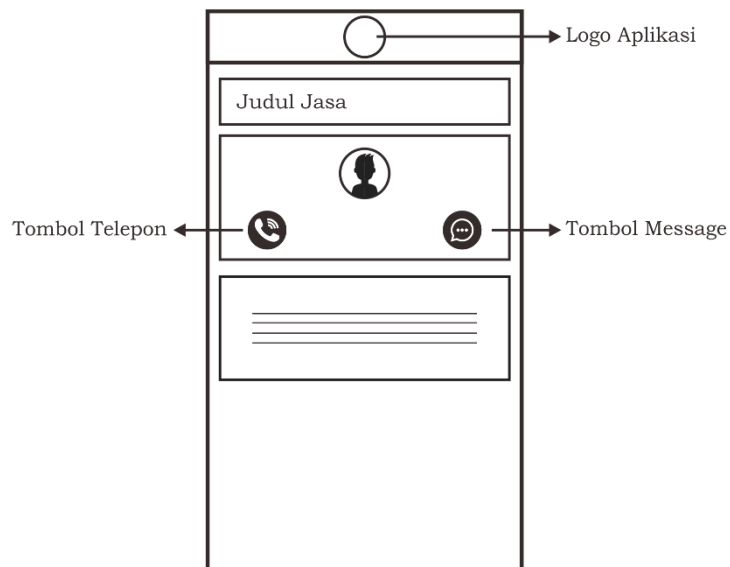
#### 5. *Layout* Menu Cari Order

Pada menu ini akan ditampilkan daftar order yang tersedia pada aplikasi Cari Tukang. Pada tampilan ini pengguna juga dapat melihat detail order dengan cara memilih salah satu data order. Dalam halaman ini juga tersedia tab pencarian untuk memudahkan pengguna dalam menemukan data order yang diinginkan. Perancangan *layout* menu “Cari Order” disajikan pada Gambar 3.28.





Gambar 3.28. *Design Layout* Menu Cari Order

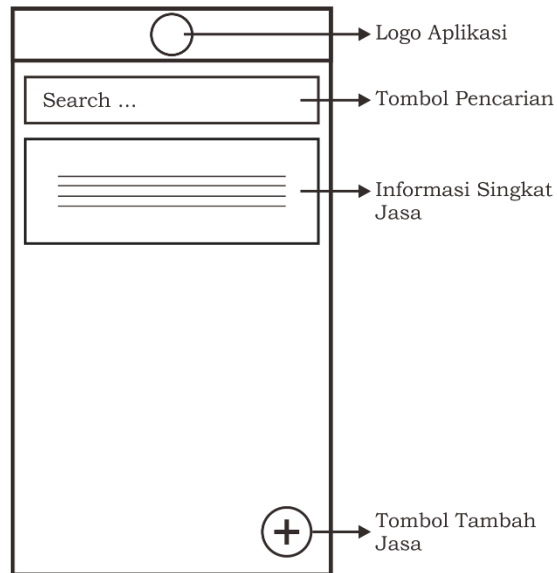


Gambar 3.29. *Design Layout* Detail Order

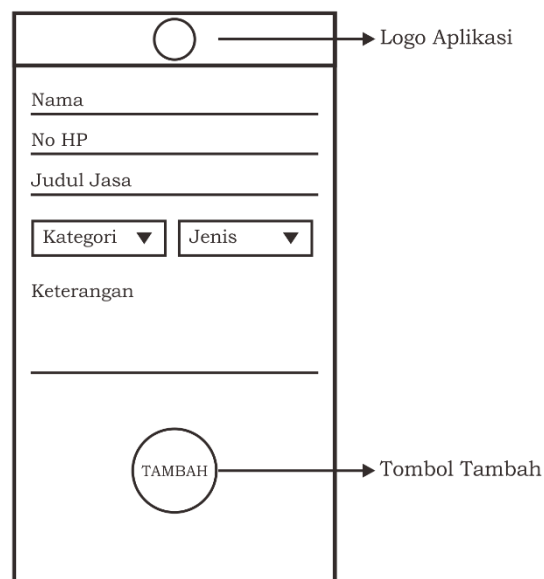
#### 6. *Layout* Menu My Jasa

Pada halaman “My Jasa” menampilkan daftar jasa pengguna. Pengguna dapat melihat detail jasa pengguna dengan memilih salah satu data jasa. Pengguna juga dapat menambahkan data jasa dengan menekan tombol

“Tambah Jasa”. Dalam halaman ini juga tersedia tab pencarian untuk memudahkan pengguna dalam menemukan data jasa pengguna yang diinginkan. Perancangan *layout* menu “My Jasa” disajikan pada Gambar 3.30.



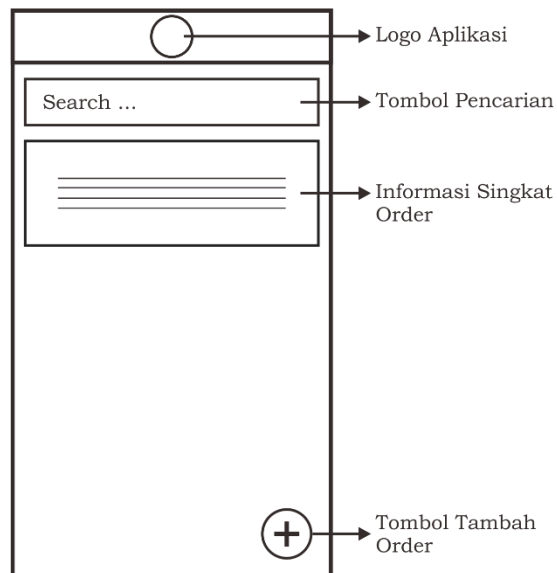
Gambar 3.30. *Layout* Menu My Jasa



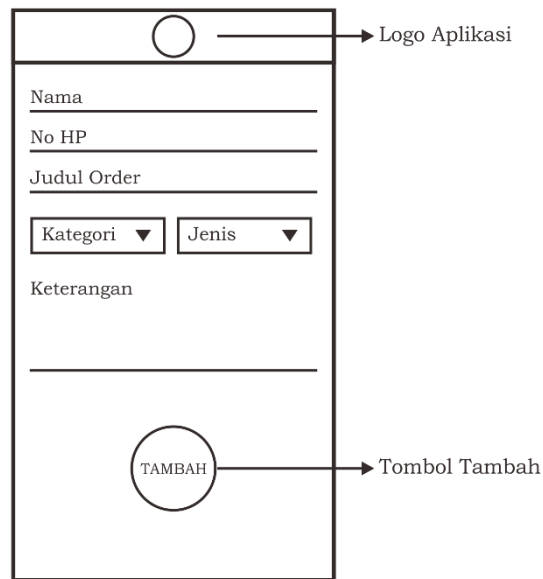
Gambar 3.31. *Layout* Tambah Jasa

## 7. *Layout Menu My Order*

Pada halaman my order menampilkan daftar order pengguna. Pengguna dapat melihat detail order pengguna dengan memilih salah satu data order. Pengguna juga dapat menambahkan data order dengan menekan tombol “Tambah Order”. Dalam halaman ini juga tersedia tab pencarian untuk memudahkan pengguna dalam menemukan data order pengguna yang diinginkan. Perancangan *layout* menu “My Order” disajikan pada Gambar 3.32.



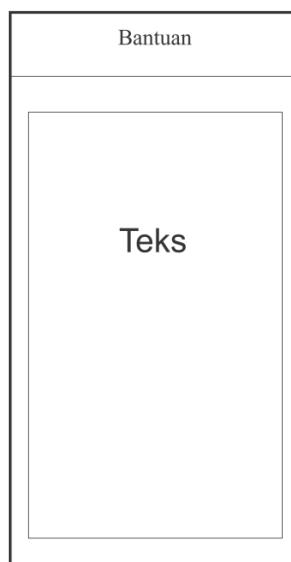
Gambar 3.32. *Design Layout Menu My Order*



Gambar 3.33. *Layout* Tambah Order

#### 8. *Layout* Menu Bantuan

Menu Bantuan berguna untuk memberikan informasi kepada pengguna tentang bagaimana menggunakan aplikasi Cari Tukang. Perancangan *layout* menu Bantuan disajikan pada Gambar 3.34.



Gambar 3.34. *Design Layout* Menu Bantuan

9. *Layout* menu Tentang aplikasi.

Menu Tentang Aplikasi berisi tentang manfaat aplikasi Cari Tukang serta informasi pengembang. Perancangan *layout* menu Tentang Aplikasi disajikan pada Gambar 3.35.



Gambar 3.35. *Design Layout* Menu Tentang Aplikasi

### 3.4 Metode Pengujian Sistem

Metode pengujian system dalam penelitian ini adalah pengujian *Black Box* dengan metode *Equivalence Partitioning* (EP). Pengujian ini berguna untuk membuktikan semua fungsi-fungsi pada aplikasi berjalan dengan baik.

Pada pengujian ini diyakinkan bahwa masukan dan respon yang diterima sama sehingga terjadi kecocokan antara aplikasi dan pengguna. Metode ini dipilih karena metode ini dapat mencari kesalahan pada fungsi aplikasi, *interface* aplikasi dan kesalahan pada struktur data aplikasi. Rancangan daftar pengujian disajikan pada Tabel 3.1.

Tabel 3.1 Daftar Pengujian *Equivalence Partitioning*(EP)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Versi Android	Pengujian kompatibilitas versi operatif sistem android	Pengujian pada android Versi 5.0 (Lollipop)	Kompatibel dengan android Versi 5.0 (Lollipop)
			Pengujian pada android Versi 6.0 (Marshmallow)	Kompatibel dengan android Versi 6.0 (Marshmallow)
2	Resolusi Layar dan Densitas Layar	Pengujian Resolusi Layar dan Densitas Layar pada Android	Pengujian pada android dengan resolusi 4 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 4 inch
			Pengujian pada android dengan resolusi 5 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 5 inch
			Pengujian pada android dengan resolusi 6 inch	Tampilan terlihat sesuai atau baik pada android dengan resolusi 6 inch

Tabel 3.1 Daftar Pengujian *Equivalence Partitioning*(EP) (Lanjutan)

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
3	<i>User Interface</i>	Pengujian Pada <i>Icon</i> Cari Tukang	Klik <i>icon</i> Cari Tukang pada perangkat android pengguna	Menampilkan layout <i>splash screen</i>
		Pengujian Pada Menu Utama Aplikasi Cari Tukang	Klik tombol “Kontraktor”	Menampilkan Daftar Tukang Kategori Kontraktor
			Klik tombol “Pindahan”	Menampilkan Daftar Tukang Kategori Pindahan

			Klik tombol “Kebersihan”	Menampilkan Daftar Tukang Kategori Kebersihan
			Klik tombol “Perbaikan”	Menampilkan Daftar Tukang Kategori Perbaikan
			Klik tombol “Sewa Barang”	Menampilkan Daftar Tukang Kategori Sewa Barang
			Klik tombol “Navigation Bar”	Menampilkan <i>Layout</i> Menu Navigasi
		Pengujian Pada Menu Navigasi	Klik Menu “Notifikasi”	Menampilkan <i>Layout</i> Notifikasi
			Klik Menu “Cari Tukang”	Menampilkan <i>Layout</i> Cari Tukang
			Klik Menu “Cari Order”	Menampilkan <i>Layout</i> Cari Order
			Klik Menu “My Jasa”	Menampilkan <i>Layout</i> My Jasa
			Klik Menu “My Order”	Menampilkan <i>Layout</i> My Order
			Klik Menu “Bantuan”	Menampilkan <i>Layout</i> Bantuan
			Klik Menu “Tentang Aplikasi”	Menampilkan <i>Layout</i> Tentang Aplikasi
			Klik “Log Out”	Keluar dari Aplikasi

## **BAB V**

### **SIMPULAN DAN SARAN**

#### **5.1 Simpulan**

Dari hasil penelitian yang dilakukan, penulis dapat mengambil simpulan sebagai berikut.

1. Aplikasi Cari Tukang telah berhasil dibangun, yang digunakan sebagai media pencarian penyedia jasa *online* berbasis *Mobile Android*.
2. Teknologi *Firebase* telah berhasil diterapkan untuk menyimpan dan mensinkronisasi data lewat *multiple client*.
3. Dari pengujian resolusi layar dan densitas layar yang dilakukan pada *Android* dengan resolusi 4 inci, 5 inci, dan 6 inci berdasarkan *test case* yang diberikan, sistem ini dapat berjalan dengan baik pada android yang memiliki ukuran layar 4 inci, 5 inci, dan 6 inci.
4. Dari pengujian user *interface* yang dilakukan, berdasarkan *test case* yang diberikan, dapat disimpulkan bahwa tampilan yang disediakan untuk pengguna (*user*) dapat berjalan dengan baik dan sesuai yang diharapkan dan dapat disimpulkan bahwa fungsi dari masing-masing menu aplikasi dapat berjalan dengan baik sesuai dengan intruksi yang diberikan.



## **5.2 Saran**

1. Meningkatkan sistem keamanan agar user merasa lebih nyaman dalam melakukan transaksi.
2. Memperluas jangkauan penyedia jasa dengan menambahkan jenis-jenis penyedia jasa yang belum ada.

## DAFTAR PUSTAKA

- Android Studio. 2016. *Android Studio*. [Online]. Tersedia:<http://developer.android.com/tools/studio/index.html>. Diakses pada tanggal 14 Januari 2017.
- Clune, T.L., R.B. Rood. 2011. *Software Testing and Verification In Climate Model Development*. *IEEE Journal, Focus: Climate Change Software*. September-October, pp. 49-55.
- Developers, Android. 2014. *Android Developers*. [Online]. Tersedia:<http://developer.android.com/index.html>. Diakses pada tanggal 20 Februari 2017.
- Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3*. Andi Publishing, Yogyakarta.
- J. Supranto, 2001. *Statistik Teori dan Aplikasi, Cetakan Kedua*, Jakarta: Penerbit Erlangga
- Jiang, F., Y. Lu. 2012. *Software Testing Model Selection Research based on Yin-Yang Testing Theory*. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.
- Meier, Reto. 2009. *Professional Android Application Development*. Wiley Publishing, Inc: Canada.
- Nasution, H.M., 1996. *Management Transportasi*. Jakarta: Ghalia Indonesia.

Nugroho, Adi. 2015. *Analisis dan Perancangan Sistem Informasi Dengan Metodologi Berorientasi Objek*. Bandung: Informatika.

Peraturan Presiden Republik Indonesia Nomor 54 Tahun 2010 Tentang Pengadaan Barang/Jasa Pemerintah

Pressman, Roger S. 2001. *Software Engineering A Practitioner's Approach Fifth Edition*. McGraw-Hill Companies, Inc, New York.

Roger, Rick, John Lombardo, et al. 2009. *Android Application Development*. O'Reilly Media: USA.

Safaat, Nazruddin. 2012. *Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android Edisi Revisi*. Informatika: Bandung.