

APLIKASI PA'LAUNDRY BERBASIS ANDROID

TUGAS AKHIR

**Karya tulis sebagai salah satu syarat
untuk memperoleh gelar Sarjana dari
Universitas Fajar**

Oleh

AZHARI USI LALOANG

1620221110



PROGRAM STUDI TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS FAJAR

2022

HALAMAN PENGESAHAN

APLIKASI PA'LAUNDRY BERBASIS ANDROID

Disusun Oleh

AZHARI USI LALOANG

1620221110

Menyetujui

Tim Pembimbing

Makassar, 15 Mei 2023

Pembimbing I



Andita Dani Achmad, S.T., M.T.
NIDN: 0913029001

Pembimbing II



Febriansyah, S.Kom., M.T.
NIDN: 0921029003

Mengetahui

Dekan Fakultas Teknik



Prof. Dr. Ir. Ermiati, S.T., M.T.
NIDN: 0906107701

Ketua Program Studi Elektro



Safaruddin, S.Si., M.T.
NIDN: 0909106901

PERNYATAAN ORISINALITAS

Penulis dengan ini menyatakan bahwa Tugas Akhir:

“Aplikasi Pa'*Laundry* Berbasis Android” adalah karya orisinal saya dan setiap serta seluruh sumber acuan telah ditulis sesuai dengan Panduan Penulisan Ilmiah yang berlaku di Fakultas Teknik Universitas Fajar.

Makassar, 15 Mei 2023

Yang menyatakan



AZHARI USI LALOANG

ABSTRAK

Aplikasi Pa'Laundry Berbasis Android, Azhari Usi Laloang. Dibalik perkembangan teknologi dan informasi masih banyak usaha *laundry* yang kesulitan dalam mempromosikan jasanya dan untuk mendapatkan informasi atau pelayanan tanpa survei ke lokasi *laundry* akan terasa sulit, sedangkan bisnis *laundry* di Kota Makassar tumbuh dengan sangat pesat sehingga membuat masyarakat menjadi selektif dalam memilih *laundry* terdekat. Maka dari itu dilakukan penelitian dengan judul Aplikasi Pa'Laundry Berbasis Android yang di mana aplikasi dapat mewadahi usaha *laundry* yang ada di Kota Makassar dalam penyediaan informasi dan membantu masyarakat memilih *laundry* yang sesuai dan terdekat berdasarkan lokasi. Metode yang digunakan dalam penelitian ini adalah *haversine formula* di mana metode ini menghitung jarak antara dua buah titik berdasarkan garis lintang (*latitude*) dan garis bujur (*longitude*). Hasil perhitungan akurasi dari penerapan metode *haversine formula* berdasarkan perbandingan antara dua studi kasus, di mana perhitungan akurasi oleh sistem dan perhitungan akurasi secara manual (mandiri) yang dilakukan dengan menghitung jarak di kedua titik tersebut lalu dibandingkan dengan hasil dari sistem dan didapatkan nilai akurasi 90% akurat dalam pencarian *laundry* berdasarkan lokasi. Hasil pengujian fungsionalitas berfungsi dengan baik berdasarkan pengujian efektifitas responden dengan nilai rata-rata 89,6% sangat setuju dengan kualitas aplikasi.

Kata Kunci : *Laundry, Haversine Formula*

ABSTRACT

Android-Based Pa'Laundry Application, Azhari Usi Laloang. Behind the development of technology and information, there are still many laundry businesses that have difficulty promoting their services and getting information or services without surveying laundry locations will be difficult, while the laundry business in Makassar City is growing very rapidly, making people selective in choosing the nearest laundry. Therefore, a research was conducted with the title Android-Based Pa'Laundry Application in which the application can accommodate laundry businesses in Makassar City in providing information and helping people choose the appropriate and closest laundry based on location. The method used in this research is the haversine formula, in which this method calculates the distance between two points based on latitude and longitude. The results of the accuracy calculation from the application of the haversine formula method are based on a comparison between two case studies, where the accuracy calculation by the system and the accuracy calculation manually (independently) are carried out by calculating the distance at the two points and then compared with the results from the system and an accuracy value of 90% is obtained. accurate in laundry search based on location. The results of functionality testing function properly based on testing the effectiveness of respondents with an average value of 89.6% strongly agree with the quality of the application.

Keywords: *Laundry, Haversine Formula*

KATA PENGANTAR

Bismillahirrahmannirrahim. Assalamu'alaikum Wr. Wb.

Puji dan syukur penulis panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nyalah sehingga penulis dapat menyelesaikan tugas akhir yang berjudul “**APLIKASI PA'LAUNDRY BERBASIS ANDROID**”. Pada kesempatan ini, penulis hendak menyampaikan terima kasih kepada semua pihak yang telah memberikan dukungan moril maupun materil sehingga tugas akhir ini dapat selesai. Ucapan terima kasih ini penulis tujukan kepada:

1. Dekan Fakultas Teknik Universitas Fajar Makassar, Ibu Prof. Dr. Ir. Erniati, S.T., M.T.
2. Ketua Program Studi Teknik Elektro Fakultas Teknik Universitas Fajar Makassar, Bapak Safaruddin, S.Si., M.T.
3. Dosen Pembimbing Ibu Andita Dani Achmad, S.T., M.T. selaku pembimbing I serta Bapak Febriansyah, S.Kom., M.T. selaku pembimbing II di Program Studi Teknik Elektro Fakultas Teknik Universitas Fajar Makassar.
4. Orang tua dan keluarga besar yang senantiasa mendoakan agar penulis selalu diberi kekuatan dan kesabaran.
5. Teman-teman dari program studi Teknik Elektro yang telah berjuang bersama-sama penulis dalam menyelesaikan tugas akhir.

Meskipun telah berusaha menyelesaikan tugas akhir ini sebaik mungkin, penulis menyadari bahwa tugas akhir ini masih ada kekurangan. Oleh karena itu, penulis mengharapkan kritik dan saran yang membangun dari para pembaca, Akhir kata, penulis berharap semoga tugas akhir ini berguna bagi para pembaca dan pihak-pihak lain yang berkepentingan.

Penulis

AZHARI USI LALOANG

DAFTAR ISI

	Halaman
HALAMAN SAMPUL	i
HALAMAN PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
ABSTRAK	iv
ABSTRACT	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	xi
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	2
I.3 Tujuan Penelitian.....	3
I.4 Batasan Masalah	3
BAB II TINJAUAN PUSTAKA	4
II.1 Tinjauan Teori	4
II.1.1 Aplikasi	4
II.1.2 Jasa <i>Laundry</i>	5
II.1.3 Android	5
II.1.4 Java	6
II.1.5 Android Studio	6
II.1.6 <i>Framework</i>	7
II.1.7 <i>Database</i>	8
II.1.8 Mapbox	8
II.1.9 <i>Waterfall</i>	9
II.1.10 <i>Haversine Formula</i>	10
II.1.11 Perancangan Sistem	11

II.1.12 Pengujian Sistem	15
II.2 Penelitian Terdahulu (<i>State of The Art</i>)	18
II.3 Kerangka Pemikiran	20
BAB III METODE PENELITIAN	21
III.1 Tahapan Penelitian.....	21
III.2 Perancangan Penelitian	23
III.2.1 Sistem yang Sedang Berjalan	23
III.2.2 Sistem yang Diusulkan	24
III.2.3 <i>Use Case Diagram</i>	24
III.2.4 <i>Activity Diagram</i>	27
III.2.5 <i>Sequence Diagram</i>	36
III.2.6 <i>Class Diagram</i>	44
III.2.7 Pencarian <i>Laundry</i> Terdekat Menggunakan <i>Haversine Formula</i> ...	48
III.2.8 Perancangan <i>Interface</i>	49
III.3 Waktu dan Tempat Penelitian	54
III.4 Alat dan Bahan	54
III.5 Metode Pengumpulan Data	54
III.6 Metode Pengujian Sistem.....	55
BAB IV HASIL DAN PEMBAHASAN.....	56
IV.1 Hasil	56
IV.1.1 Implementasi <i>Haversine Formula</i>	56
IV.1.2 Implementasi <i>Interface</i>	68
IV.2 Pembahasan	78
IV.2.1 Pengujian <i>White Box</i>	78
IV.2.2 Pengujian <i>Black Box</i>	99
BAB V PENUTUP	108
V.1 Kesimpulan.....	108
V.2 Saran	108
DAFTAR PUSTAKA	109

DAFTAR TABEL

	Halaman
Tabel II. 1 <i>State of The Art</i>	18
Tabel III. 1 Definisi Aktor.....	25
Tabel III. 3 Definisi <i>Use Case</i> Pada Aktor Admin.....	25
Tabel III. 4 Definisi <i>Use Case</i> Pada Pemilik <i>Laundry</i>	26
Tabel III. 5 Definisi <i>Use Case</i> Pada Pengguna/Pelanggan	26
Tabel III. 6 Admin	45
Tabel III. 7 Kelola <i>Laundry</i>	45
Tabel III. 8 Kelola Pengguna	45
Tabel III. 9 Registrasi	45
Tabel III. 10 Pemilik <i>Laundry</i>	46
Tabel III. 11 Pelanggan	46
Tabel III. 12 Kelola Penjualan.....	46
Tabel III. 13 Proses Pesanan	47
Tabel III. 14 Cari <i>Laundry</i>	47
Tabel III. 15 Pesan	47
Tabel III. 16 Lihat Proses Pesanan	48
Tabel IV. 1 Perhitungan Jarak	65
Tabel IV. 2 <i>Test Case Login</i>	79
Tabel IV. 3 <i>Test Case</i> Registrasi	81
Tabel IV. 4 <i>Test Case</i> Kelola <i>Laundry</i>	83
Tabel IV. 5 <i>Test Case</i> Kelola Pengguna	86
Tabel IV. 6 <i>Test Case</i> Kelola Penjualan	88
Tabel IV. 7 <i>Test Case</i> Proses Pesanan	91
Tabel IV. 8 <i>Test Case</i> Cari <i>Laundry</i>	94
Tabel IV. 9 <i>Test Case</i> Pesan.....	96
Tabel IV. 10 <i>Test Case</i> Lihat Pesanan	98
Tabel IV. 11 Rekapitulasi Hasil Pengujian	99
Tabel IV. 12 Pengujian Fungsional Sistem.....	99

Tabel IV. 13 Tabel Pengujian Fungsional.....	102
Tabel IV. 14 Nilai Skor Maksimum	104
Tabel IV. 15 Kriteria Skor.....	104
Tabel IV. 16 Hasil Kuesioner Pertanyaan 1	105
Tabel IV. 17 Hasil Kuesioner Pertanyaan 2	105
Tabel IV. 18 Hasil Kuesioner Pertanyaan 3	106
Tabel IV. 19 Hasil Kuesioner Pertanyaan 4	106
Tabel IV. 20 Hasil Kuesioner Pertanyaan 5	107
Tabel IV. 21 Hasil Rata-Rata Persentase	107

DAFTAR GAMBAR

	Halaman
Gambar II. 1 Android.....	5
Gambar II. 2 Java.....	6
Gambar II. 3 Android Studio.....	7
Gambar II. 4 Mapbox.....	8
Gambar II. 5 <i>Waterfall</i>	9
Gambar II. 6 Simbol <i>Use Case</i>	12
Gambar II. 7 Simbol <i>Activity</i>	13
Gambar II. 8 Simbol <i>Sequence</i>	14
Gambar II. 9 Simbol <i>Class</i>	15
Gambar II. 10 Kerangka Pikir.....	20
Gambar III. 1 Tahapan Penelitian.....	21
Gambar III. 2 Sistem Yang Sedang Berjalan.....	23
Gambar III. 3 Sistem yang Diusulkan.....	24
Gambar III. 4 <i>Activity Login</i> Admin, Pemilik <i>Laundry</i> dan Pengguna/Pelanggan	27
Gambar III. 5 <i>Activity</i> Registrasi Pemilik <i>Laundry</i> dan Pengguna/Pelanggan.....	28
Gambar III. 6 <i>Activity</i> Admin Mengelola <i>Laundry</i>	29
Gambar III. 7 <i>Activity</i> Admin Mengelola Pengguna/Pelanggan.....	30
Gambar III. 8 <i>Activity</i> Pemilik <i>Laundry</i> Mengelola Penjualan.....	31
Gambar III. 9 <i>Activity</i> Pemilik <i>Laundry</i> Memproses Pesanan.....	32
Gambar III. 10 <i>Activity</i> Pengguna/Pelanggan Mencari <i>Laundry</i>	33
Gambar III. 11 <i>Activity</i> Pelanggan Memesan Jasa.....	34
Gambar III. 12 <i>Activity</i> Pengguna/Pelanggan Melihat Proses Pemesanan.....	35
Gambar III. 13 <i>Sequence Login</i> Admin, Pemilik <i>Laundry</i> dan Pengguna/Pelanggan.....	36
Gambar III. 14 <i>Sequence</i> Registrasi Pemilik <i>Laundry</i> dan Pengguna/Pelanggan.....	36
Gambar III. 15 <i>Sequence</i> Admin Mengelola <i>Laundry</i>	37
Gambar III. 16 <i>Sequence</i> Admin Mengelola Pengguna/Pelanggan.....	38
Gambar III. 17 <i>Sequence</i> Pemilik <i>Laundry</i> Mengelola Penjualan.....	39

Gambar III. 18 <i>Sequence</i> Pemilik <i>Laundry</i> Memproses Pesanan	40
Gambar III. 19 <i>Sequence</i> Pengguna/Pelanggan Mencari <i>Laundry</i>	41
Gambar III. 20 <i>Sequence</i> Pengguna/Pelanggan Memesan Jasa	42
Gambar III. 21 <i>Sequence</i> Pengguna/Pelanggan Melihat Proses Pesanan	43
Gambar III. 22 <i>Class Diagram</i>	44
Gambar III. 23 <i>Flowchart</i> Tahapan <i>Haversine Formula</i>	48
Gambar III. 24 Tampilan <i>Form Login</i>	49
Gambar III. 25 Tampilan Registrasi	50
Gambar III. 26 Tampilan Admin Mengelola <i>Laundry</i>	50
Gambar III. 27 Tampilan Admin Mengelola Pengguna/Pelanggan	51
Gambar III. 28 Tampilan Pemilik <i>Laundry</i> Mengelola Penjualan	51
Gambar III. 29 Tampilan Pemilik <i>Laundry</i> Proses Pesanan	52
Gambar III. 30 Tampilan Pengguna/Pelanggan Mencari <i>Laundry</i>	52
Gambar III. 31 Tampilan Pengguna/Pelanggan Memesan Jasa	53
Gambar III. 32 Tampilan Pengguna/Pelanggan Melihat Proses Pesanan	53
Gambar IV. 1 Hasil perhitungan Jarak Terdekat pada aplikasi di Kecamatan Biringkanaya	56
Gambar IV. 2 Hasil Perhitungan Jarak Terdekat Pada Aplikasi di Kecamatan Tamalanrea	57
Gambar IV. 3 Hasil Tampilan Aplikasi	68
Gambar IV. 4 Hasil Tampilan <i>Login</i>	69
Gambar IV. 5 Hasil Tampilan Registrasi	70
Gambar IV. 6 Hasil Tampilan Admin Mengelola <i>Laundry</i>	71
Gambar IV. 7 Hasil Tampilan Admin Mengelola Pengguna/Pelanggan	72
Gambar IV. 8 Hasil Tampilan Pemilik <i>Laundry</i> Mengelola Penjualan	73
Gambar IV. 9 Hasil Tampilan Pemilik <i>Laundry</i> Memproses Pesanan	74
Gambar IV. 10 Hasil Tampilan Pengguna/Pelanggan Mencari <i>Laundry</i>	75
Gambar IV. 11 Hasil Tampilan Pengguna/Pelanggan Memesan Jasa	76
Gambar IV. 12 Hasil Tampilan Pengguna/Pelanggan Melihat Proses Pesanan	77
Gambar IV. 13 <i>Flowchart Login</i>	78
Gambar IV. 14 <i>Flow graph Login</i>	79

Gambar IV. 15 <i>Flowchart</i> Registrasi.....	80
Gambar IV. 16 <i>Flow graph</i> Registrasi.....	81
Gambar IV. 17 <i>Flowchart</i> Kelola Laundry.....	82
Gambar IV. 18 <i>Flow graph</i> Kelola Laundry.....	83
Gambar IV. 19 <i>Flowchart</i> Kelola Pengguna.....	85
Gambar IV. 20 <i>Flow graph</i> Kelola Pengguna.....	85
Gambar IV. 21 <i>Flowchart</i> Kelola Penjualan.....	87
Gambar IV. 22 <i>Flow graph</i> Kelola Penjualan.....	87
Gambar IV. 23 <i>Flowchart</i> Proses Pesanan.....	90
Gambar IV. 24 <i>Flow graph</i> Proses Pesanan	91
Gambar IV. 25 <i>Flowchart</i> Cari Laundry	93
Gambar IV. 26 <i>Flow graph</i> Cari Laundry	93
Gambar IV. 27 <i>Flowchart</i> Pesan	95
Gambar IV. 28 <i>Flow graph</i> Pesan	96
Gambar IV. 29 <i>Flowchart</i> Lihat Pesanan	97
Gambar IV. 30 <i>Flow graph</i> Lihat Pesanan	98
Gambar IV. 31 Grafik Hasil Pengujian Fungsional.....	103

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Kuesioner <i>Laundry</i> Responden 1.....	112
Lampiran 2 Kuesioner <i>Laundry</i> Responden 2.....	112
Lampiran 3 Kuesioner <i>Laundry</i> Responden 3.....	113
Lampiran 4 Kuesioner <i>Laundry</i> Responden 4.....	113
Lampiran 5 Kuesioner <i>Laundry</i> Responden 5.....	114
Lampiran 6 Kuesioner Pengguna 1.....	114
Lampiran 7 Kuesioner Pengguna 2.....	115
Lampiran 8 Kuesioner Pengguna 3.....	115
Lampiran 9 Kuesioner Pengguna 4.....	116
Lampiran 10 Kuesioner Pengguna 5.....	116

BAB I

PENDAHULUAN

I.1 Latar Belakang

Perkembangan teknologi dan informasi seiring dengan waktu yang sangat cepat, membuat manusia harus memahami fenomena tersebut, karena dengan adanya fenomena ini sangat membantu dan dapat mempermudah segala aktifitas yang dilakukan oleh manusia khususnya dalam layanan jasa *online* yang mudah dan cepat. Meningkatnya layanan jasa *online* saat ini seperti Go-Jek maupun Grab yang merupakan jasa pemesanan *online* berupa jasa ojek antar jemput penumpang, jasa antar jemput makanan, dan jasa antar jemput barang kini sudah sangat diminati oleh masyarakat Indonesia dikarenakan jasa online yang mudah didapati cepat dan aman, (Kusumastuti, A., 2020). Salah satu aktifitas masyarakat yang paling sering dilakukan juga adalah mencuci pakaian tentunya aktifitas ini merupakan kegiatan yang harus dilakukan dikarenakan kebutuhan masyarakat akan pakaian bersih untuk digunakan sehari-hari, tetapi dengan padatnya kegiatan masyarakat seperti bekerja atau rutinitas lain terkadang membuat masyarakat tidak memiliki waktu untuk mencuci pakaian sehingga memakai alternatif jasa cuci pakaian atau jasa *laundry*. *Laundry* sendiri merupakan bisnis yang bergerak dibidang jasa pencucian pakaian maupun kain yang menggunakan mesin cuci otomatis, pengeringan, penyetrikaan, cairan pembersih noda dan pengharum khusus. Usaha jenis ini sekarang sangat marak dijumpai di Kota-kota besar dan peminat penggunaan jasa *laundry* sendiri kebanyakan masyarakat yang tidak sempat atau tidak mempunyai waktu untuk mencuci pakaiannya sendiri seperti pekerja, pelajar, mahasiswa atau orang berkesibukan lain, (Felicia Sylviana, dkk, 2019).

Dengan maraknya usaha *laundry* ini membuat masyarakat menjadi selektif untuk memilih tempat *laundry* terdekat dengan kualitas terbaik. Namun dibalik perkembangan teknologi dan informasi masih banyak usaha *laundry* yang kesulitan dalam mempromosikan tempat atau jasanya sehingga untuk mendapatkan informasi dan pelayanan tanpa survei ke lokasi *laundry* akan terasa sulit. Sedangkan bisnis *laundry* di kota Makassar tumbuh sangat pesat. Kebanyakan dapat dijumpai di setiap

tempat di kota Makassar terutama di daerah dekat kampus, pabrik, atau kost, tetapi belum adanya aplikasi yang dapat mewadahi usaha *laundry* yang ada di Kota Makassar dalam hal penyediaan informasi dan pelayanan jasa *laundry*.

Penelitian sebelumnya yang berjudul “Sistem Informasi Pemesanan *Laundry* Berbasis Android di Kota Palembang” pada penelitian ini menggunakan metode *Rational Unified Process* (RUP) dan Sistem Informasi Geografis (SIG) dengan hasil sistem ini dapat mempermudah pelanggan dalam mencari tempat *laundry* terdekat dengan kualitas terbaik berdasarkan dari hasil penilaian, (Susanto, E., dkk, 2019). Penelitian selanjutnya yang berjudul “Penerapan Metode *Haversine Formula* Pada Sistem Informasi Geografis Pencarian Lokasi *Laundry* Terdekat di Kota Makassar” pada penelitian ini menggunakan Sistem Informasi Geografis (SIG) dan Metode *Haversine Formula* dengan hasil terbentuknya aplikasi pencarian yang dapat membantu masyarakat sekitar dalam mencari *laundry* terdekat dari lokasi mereka berada, (M, Iqbal., dkk, 2021).

Oleh karena itu berdasarkan latar belakang diatas maka dilakukan penelitian dengan judul “**Aplikasi Pa’*Laundry* Berbasis Android**” yang dimana aplikasi dapat mewadahi usaha *laundry* yang ada di Kota Makassar dalam penyediaan informasi dan pelayanan jasa *laundry*.

I.2 Rumusan Masalah

Berdasarkan uraian di atas, maka permasalahan yang akan dibahas dalam tugas akhir ini adalah:

1. Bagaimana cara membuat aplikasi pencarian *laundry* berdasarkan lokasi yang dapat mewadahi usaha *laundry* dalam penyediaan informasi dan pelayanan jasa *laundry*?
2. Bagaimana cara mengimplementasikan metode *haversine formula* pada aplikasi pencarian *laundry* berdasarkan lokasi?

I.3 Tujuan Penelitian

Penelitian ini bertujuan untuk:

1. Membuat aplikasi pencarian *laundry* berdasarkan lokasi berbasis Android yang dapat mewadahi usaha *laundry* dalam penyediaan informasi dan pelayanan jasa *laundry*
2. Mengimplementasikan metode *haversine formula* pada aplikasi dalam pencarian *laundry* berdasarkan lokasi.

I.4 Batasan Masalah

Dalam tugas akhir ini terdapat batasan:

1. Aplikasi ini belum memiliki fitur pembayaran secara *online*.
2. Aplikasi ini hanya mencari *laundry* berdasarkan lokasi.
3. Pengujian aplikasi ini hanya dilakukan di Kecamatan Biringkanaya dan Tamalanrea
4. Aplikasi ini hanya menyediakan informasi dan pelayanan jasa *laundry*

BAB II

TINJAUAN PUSTAKA

II.1 Tinjauan Teori

II.1.1 Aplikasi

Aplikasi merupakan sebuah program berbentuk perangkat lunak yang bekerja pada sistem untuk membantu pengguna dalam kegiatan yang dilakukan, (Abdurahman, H., Riswayah, R, A., 2014) dalam karya ilmiah (Simargolang dan Nasution, 2018). Fungsi dari aplikasi secara umum dapat mempermudah pekerjaan seseorang, sebagai media hiburan, media komunikasi dan pertemanan, dan pembaharuan kabar terkini. Aplikasi juga memiliki beberapa pengelompokan atau jenis-jenis seperti:

1. *Real Time Software*

Merupakan klasifikasi aplikasi yang dapat berguna dalam mengamati, mengendalikan, maupun menganalisa sebuah keadaan di dunia nyata secara langsung.

2. *System Software*

Adalah aplikasi yang berguna untuk mengendalikan serta mengelola proses operasi internal yang ada pada sebuah sistem komputer.

3. *Business Software*

Merupakan aplikasi yang dibuat untuk membantu keperluan bisnis seseorang. Salah satu contohnya adalah mengatur sistem keuangan, mendata jumlah pesanan, dan masih banyak lagi.

4. *Personal Computer Software*

Adalah aplikasi yang dapat digunakan pengguna resmi maupun pribadi.

5. *Web Based Software*

Adalah aplikasi dengan fungsi sebagai media penghubung pengguna pada internet secara langsung.

6. *Engginering and Scientific Software*

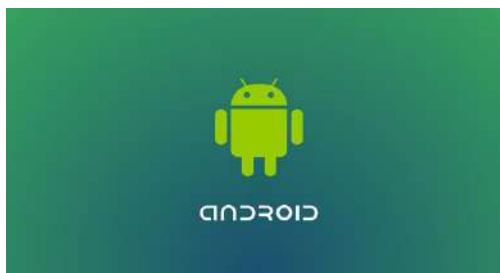
Merupakan aplikasi yang dibuat dan dikembangkan guna membantu manusia menyelesaikan berbagai masalah yang ada. Hanya saja masalah tersebut bersifat secara khusus, yakni non algoritmik.

II.1.2 *Jasa Laundry*

Jasa laundry merupakan salah satu pelayanan jasa dibidang cuci mencuci pakaian, kain, selimut, dan lain-lain. Pelanggan bisa memakai jasa tersebut dengan memilih jenis cucian yang telah ditetapkan harga oleh pihak penyedia jasa dan waktu lama cucian biasanya ditentukan oleh penyedia dengan batas minimal dan maksimal selesainya cucian yang dipesan oleh pelanggan, (Abdurahman. H, Riswayah. R. A, 2014) dalam karya ilmiah (Simargolang dan Nasution, 2018).

II.1.3 *Android*

Android adalah sistem operasi yang dirancang oleh Google dengan basis kernel Linux untuk mendukung kinerja perangkat elektronik layar sentuh, seperti *tablet* atau *smartphone*. Sistem operasi adalah aplikasi yang mengatur seluruh sumber daya dalam perangkat elektronik, mulai dari manajemen memori, aplikasi, antarmuka tampilan dan sebagainya. Android bersifat *open source* atau bebas digunakan, dimodifikasi, diperbaiki dan didistribusikan oleh para pembuat ataupun pengembang perangkat lunak sehingga dengan sifat *open source* perusahaan teknologi bebas menggunakan *operating system* (OS) ini diperangkat tanpa lisensi/gratis, (Nazruddin dan Safaat, 2012) dalam karya ilmiah (Susanto, dkk, 2019).



Gambar II. 1 Android
(Sumber: Google)

II.1.4 Java

Java merupakan bahasa pemrograman tingkat tinggi yang dapat diterapkan diberbagai *platform*. Java memiliki ciri-ciri sebagai bahasa yang sederhana, arsitektural netral berorientasi objek, memiliki kinerja tinggi, *multithreaded*, kuat, dinamis dan aman. Java juga dikenal memiliki moto “*Write Once, Run Anywhere*” artinya, Java mampu dijalankan diberbagai platform tanpa perlu disusun ulang menyesuaikan platformnya. Misalnya, berjalan di Android, Linux, Windows, dan lainnya. Bahasa pemrograman Java diciptakan sebagai pengembangan dari bahasa pemrograman C dan C++ yang kurang fleksibel. Java menjadi bahasa pemrograman yang telah banyak dipakai oleh para *developer* karena keunggulannya. Di antaranya, mudah dijalankan dibeberapa *platform*, memiliki *library* yang lengkap, berorientasi pada objek, hingga mudah dipelajari karena mirip dengan bahasa C++ (Windu dan Grace, 2013).



Gambar II. 2 Java

(Sumber: Google)

II.1.5 Android Studio

Android Studio adalah *Integrated Development Environment* (IDE) yakni *software* yang bisa digunakan untuk mengembangkan aplikasi Android. Android Studio merupakan pengembangan dari Eclipse IDE, dan dibuat berdasarkan IDE Java populer. Dalam Android Studio terdapat banyak fitur yang bisa diandalkan para programmer baik pemula maupun profesional untuk mengembangkan aplikasi Android, (Noviana, E., 2018) dalam karya ilmiah (Kusumastuti, 2020). Adapun fitur-fitur tersebut adalah sebagai berikut:

1. Sistem versi *gradle* yang fleksibel.
2. Emulator yang cepat dan dilengkapi kaya fitur.

3. Lingkungan yang menyatu untuk mengembangkan aplikasi Android bagi semua perangkat Android (*smartphone, tablet, smarttv, dan smartwatch*).
4. Template kode dan integrasi dengan GitHub untuk membuat fitur aplikasi yang sama dan mengimpor kode-kode contoh.
5. Alat pengujian dan kerangka kerja yang ekstensif.
6. Dukungan C++ dan NDK.
7. Dukungan *Google Cloud Platform* sehingga mempermudah saat pengintegrasian *Google Cloud Messaging* dan *App Engine*.



Gambar II. 3 Android Studio
(Sumber: Google)

II.1.6 *Framework*

Framework adalah sebuah *software* yang digunakan untuk memudahkan para *programmer* membuat aplikasi atau *web* yang isinya adalah berbagai fungsi, *plugin*, dan konsep sehingga membentuk suatu sistem tertentu. Dengan menggunakan *framework*, sebuah aplikasi akan tersusun dan terstruktur dengan rapi. *Framework* juga dapat diartikan sebagai komponen pemrograman yang telah jadi dan siap digunakan kapan saja sehingga *programmer* tidak perlu lagi membuat *script* yang sama untuk *project* yang sama, (Kasman, 2015) dalam karya ilmiah (Purnama Sari dan Wijanarko, 2020). *Framework* memiliki beberapa fungsi seperti:

1. Menghasilkan aplikasi dengan uji stabilitas yang baik.
2. Meningkatkan keamanan dan mengantisipasi masalah keamanan yang akan muncul.
3. Mempermudah membaca kode program dan mencari *bug*.
4. Menghemat waktu pengempangan.
5. Mempermudah mendokumentasi aplikasi yang dibangun.

II.1.7 Database

Database atau basis data adalah kumpulan data yang dikelola sedemikian rupa berdasarkan ketentuan tertentu yang saling berhubungan sehingga mudah dalam pengelolaannya. Melalui pengelolaan tersebut pengguna dapat memperoleh kemudahan dalam mencari informasi, menyimpan informasi dan membuang informasi. Secara konsep basis data atau database adalah kumpulan dari data-data yang membentuk suatu berkas (*file*) yang saling berhubungan (*relation*) dengan tata cara yang tertentu untuk membentuk data baru atau informasi, (Sutarman, 2012) dalam karya ilmiah (Mur, dkk, 2019).

II.1.8 Mapbox

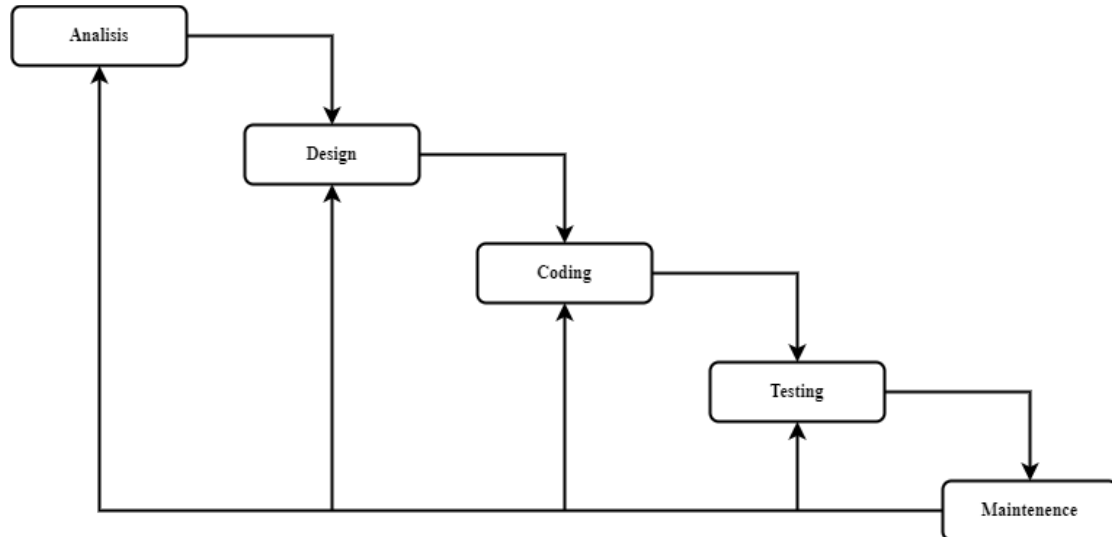
Mapbox adalah salah satu penyedia peta daring kustom terbesar di situs-situs ternama seperti *foursquare*, *pinterest*, *evernote*, *financial times*, dan *uber technologies*. Mapbox memperbanyak pilihan peta kustomnya untuk mengisi keterbatasan yang dimiliki penyedia peta seperti Google Maps. Mapbox merupakan pencipta atau kontributor sejumlah pustaka dan aplikasi peta bebas terkenal, misalnya spesifikasi MBTiles, kartografi TileMill IDE, pustaka JavaScript *leaflet*, bahasa gaya dan parser peta CartoCSS, dan pustaka JavaScript *mapbox.js*. Data Mapbox diambil dari sumber-sumber data terbuka seperti *Open Street Map* dan *NAS*, dan sumber-sumber data berbayar seperti *Digital Globe*. Teknologinya dibangun menggunakan *Node.js*, *CouchDB*, *Mapnik*, *GDAL* dan *Leaflet.js*. MapBox, yaitu dukungan terhadap *GeoJSON – encoding* obyek geografis dalam bentuk *JSON*. Mapbox juga merupakan turunan dari *Open Street Map*, (Aryando, dkk, 2019).



Gambar II. 4 Mapbox
(Sumber: Google)

II.1.9 Waterfall

Waterfall atau air terjun merupakan metode dalam pengembangan sistem yang dilakukan untuk membuat pembaruan pada sistem yang berjalan, (Felicia Sylviana, dkk, 2019). Berikut gambar yang dapat diilustrasikan pada pengembangan sistem model *waterfall*.



Gambar II. 5 *Waterfall*

(Sumber: Google)

1. Analisis

Sebelum melakukan pengembangan perangkat lunak, seorang pengembang harus mengetahui dan memahami bagaimana informasi kebutuhan pengguna terhadap sebuah perangkat lunak. Metode pengumpulan informasi ini dapat diperoleh dengan berbagai macam cara diantaranya, diskusi, observasi, survei, wawancara, dan sebagainya.

2. *Design*

Perancangan desain dilakukan dengan tujuan membantu memberikan gambaran lengkap mengenai apa yang harus dikerjakan. Tahap ini juga akan membantu pengembang untuk menyiapkan kebutuhan *hardware* dalam pembuatan arsitektur sistem perangkat lunak yang akan dibuat secara keseluruhan.

3. *Coding*

Tahap implementasi pemrograman merupakan tahap dimana pembuatan perangkat lunak dibagi menjadi modul-modul kecil yang nantinya akan digabungkan dalam tahap berikutnya. Disamping itu, pada fase ini juga dilakukan pengujian dan pemeriksaan terhadap fungsionalitas modul yang sudah dibuat, apakah sudah memenuhi kriteria yang diinginkan atau belum.

4. *Testing*

Setelah seluruh unit atau modul yang dikembangkan dan diuji di tahap implementasi selanjutnya diintegrasikan dalam sistem secara keseluruhan. Setelah proses integrasi selesai, selanjutnya dilakukan pemeriksaan dan pengujian sistem secara keseluruhan untuk mengidentifikasi kemungkinan adanya kegagalan dan kesalahan sistem.

5. *Maintenance*

Pada tahap terakhir dalam Metode Waterfall, perangkat lunak yang sudah jadi dioperasikan pengguna dan dilakukan pemeliharaan. Pemeliharaan memungkinkan pengembang untuk melakukan perbaikan atas kesalahan yang tidak terdeteksi pada tahap-tahap sebelumnya.

II.1.10 Haversine Formula

Metode *haversine formula* merupakan rumus yang berdasarkan dari bentuk bumi yang tidak elips melainkan bulat. Metode ini digunakan untuk memperhitungkan besarnya jarak yang ada diantara dua buah titik yang diambil dari titik *latitude* (lintang) dan *longitude* (bujur), Rumus perhitungan *haversine formula* dapat dilihat dibawah:

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$y = (\text{lat2} - \text{lat1})$$

$$d = \text{sqrt}(x * x + y * y) * r$$

Keterangan

x = *Longitude* (Lintang)

y = *Latitude* (Bujur)

d = Jarak

$r = \text{Radius bumi} = 6371 \text{ Km}$

1 derajat = 0.0174532925 radian

(D, Prasetyo dan K, Hastuti, 2015) dalam karya ilmiah (Abadi Nugroho, dkk, 2020)

II.1.11 Perancangan Sistem




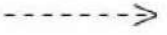



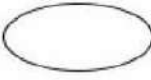


Perancangan sistem merupakan suatu penentuan proses dan data yang akan diperlukan oleh sistem baru. Tujuan dari perancangan sistem adalah untuk memenuhi kebutuhan pemakai sistem serta untuk memberikan gambaran jelas dan rancang bangun yang lengkap. Perancangan sistem juga mempunyai beberapa metode perancangan permodelan seperti *Data Flow Diagram (DFD)* dan *Unified Modelling Language (UML)*, dalam karya ilmiah (Mur, dkk, 2019).

II.1.11.1 Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah suatu metode untuk permodelan visual untuk digunakan sebagai sarana perancangan sistem yang berorientasi pada objek. Sejarah UML diciptakan oleh *object management group* pada tahun 1997, UML juga dapat didefinisikan sebagai bahasa standar visualisasi, perancangan, dan pendokumentasian sistem atau biasa dikenal sebagai bahasa standar penulisan *blueprint* sebuah *software*. UML juga dapat digunakan sebagai alat transfer ilmu tentang sistem yang dikembangkan dari pengembang ke pengembang yang lain, UML juga berfungsi sebagai jembatan penerjemah antara pengembang sistem dan pengguna, dalam karya ilmiah (Mur, dkk, 2019). UML memiliki beberapa jenis diagram yang sering digunakan seperti:

1. *Use Case Diagram*






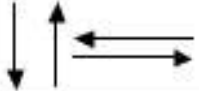
Jenis diagram ini menggambarkan hubungan interaksi antara sistem dan aktor, *Use case* merupakan langkah awal untuk melakukan permodelan pada sistem. Berikut penjelasan nama dan simbol yang dapat dilihat pada gambar II.6:

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak (<i>descendent</i>) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (<i>ancestor</i>).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara <i>eksplisit</i> .
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya (<i>sinergi</i>).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

Gambar II. 6 Simbol *Use Case*

2. Activity Diagram


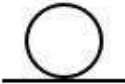
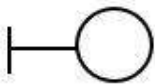



Jenis diagram ini merupakan sebuah diagram yang dapat memodelkan berbagai proses pada sistem yang digambarkan secara vertical. Berikut penjelasan nama dan simbol yang dapat dilihat pada gambar II.7:

GAMBAR	NAMA	KETERANGAN
	Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
	Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
	Initial Node	Bagaimana objek dibentuk atau diawali.
	Activity Final Node	Bagaimana objek dibentuk dan diakhiri
	Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
	Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

Gambar II. 7 Simbol Activity

3. Sequence Diagram







diagram ini menjelaskan interaksi objek berdasarkan urutan atau tahapan yang harus dilakukan untuk dapat menghasilkan sesuatu. Berikut penjelasan nama dan simbol yang dapat dilihat pada gambar II.8:

GAMBAR	NAMA	KETERANGAN
	<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sisitem.
	<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
	<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari foem
	<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
	<i>A focus of Control & A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
	<i>A message</i>	Menggambarkan Pengiriman Pesan

Gambar II. 8 Simbol *Sequence*

4. *Class Diagram*

Jenis diagram ini digunakan untuk menampilkan kelas-kelas berupa paket untuk memenuhi salah satu kebutuhan paket yang akan digunakan nantinya. Berikut penjelasan nama dan simbol yang dapat dilihat pada gambar II.9:

GAMBAR	NAMA	KETERANGAN
	<i>Generalization</i>	Relasi antarkelas dengan makna generalisasi-spesialisasi (umum-khusus)
	<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
	<i>Class</i>	Kelas pada struktur sistem
	<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
	<i>Dependency</i>	Relasi antarkelas dengan makna kebergantungan antarkelas
	<i>Association</i>	Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan multiplicity

Gambar II. 9 Simbol *Class*

II.1.12 Pengujian Sistem

Pengujian sistem merupakan proses pengeksekusian sistem perangkat lunak untuk menentukan apakah sistem perangkat lunak tersebut cocok dengan spesifikasi sistem dan berjalan sesuai dengan yang diinginkan. Pengujian sistem sering ditautkan dengan pencarian bagian-bagian, ketidak sempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak. Perancangan sistem memiliki beberapa metode pengujian seperti *white box* dan *black box*, dalam karya ilmiah (Ninik Sri Lestari, 2020).

II.1.12.1 *White Box*

White box testing adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan melihat modul untuk memeriksa dan menganalisis kode program

ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam *output* yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan singkatnya *white box testing* ini menguji dengan cara melihat *pure code* dari suatu aplikasi/*software* yang diuji tanpa memperdulikan tampilan atau *User Interface* (UI) dari aplikasi tersebut, dalam karya ilmiah (Ninik Sri Lestari, 2020). Berikut teknik yang digunakan *white box testing*:

1. *Basis Path Testing*

Basis path testing merupakan metode yang memungkinkan perancang *testcase* untuk membuat pengukuran kompleks logikal dari rancangan prosedur dan menggunakan pengukuran ini sebagai panduan untuk mendefinisikan himpunan basis dari jalur eksekusi.

2. *Flow graph*

Flow Graph adalah notasi sederhana untuk merepresentasi *control flow*.

3. *Cyclomatic Clompexity*

Cyclomatic complexity digunakan untuk mengetahui jumlah jalur yang perlu dicari. *Cyclomatic complexity* adalah *metricsoftware* yang menyediakan ukuran kuantitatif dari kompleksitas logikal program.

II.1.12.2 Black Box

Black box testing adalah pengujian yang didasarkan pada detail aplikasi seperti tampilan aplikasi, fungsi-fungsi yang ada pada aplikasi, dan kesesuaian alur fungsi dengan bisnis proses yang diinginkan oleh *customer*. *Black box testing* ini lebih menguji ke tampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh *customer*. Pengujian ini tidak melihat dan menguji *souce code* program. *Black-box testing* bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya hanya terfokus pada informasi domain. Berikut teknik yang digunakan *black box testing*:

1. *Equivalence Partitioning*

Cara kerja teknik ini adalah dengan melakukan *partition* atau pembagian menjadi beberapa partisi dari input data.

2. *Boundary Value Analysis*

Teknik ini lebih fokus kepada *boundary*, di mana adakah *error* dari luar atau sisi dalam *software*, minimum, maupun maksimum nilai dari *error* yang didapat.

3. *Fuzzing*

Fuzz merupakan teknik untuk mencari *bug/gangguan* dari *software* dengan menggunakan injeksi data yang terbilang cacat ataupun sesi semi otomatis.

4. *Cause Effect Graph*

Penggunaan teknik *testing* ini menggunakan grafik sebagai pacuannya di mana dalam grafik ini menggambarkan relasi diantara efek dan penyebab dari *error* tersebut.

5. *Orthogonal Array Testing*

Dapat digunakan jika *input* domain yang relatif terbilang kecil ukurannya, tetapi cukup berat untuk digunakan dalam skala besar.

6. *All Pair Testing*

Dalam teknik ini, semua pasangan dari *testcase* didesain sedemikian rupa agar dapat dieksekusi semua kemungkinan kombinasi diskrit dari seluruh pasangan berdasar *input* parameteranya.

7. *State Transition*

Testing ini berguna untuk melakukan pengetesan terhadap kondisi dari mesin dan navigasi dari UI dalam bentuk grafik.

II.2 Penelitian Terdahulu (*State of The Art*)

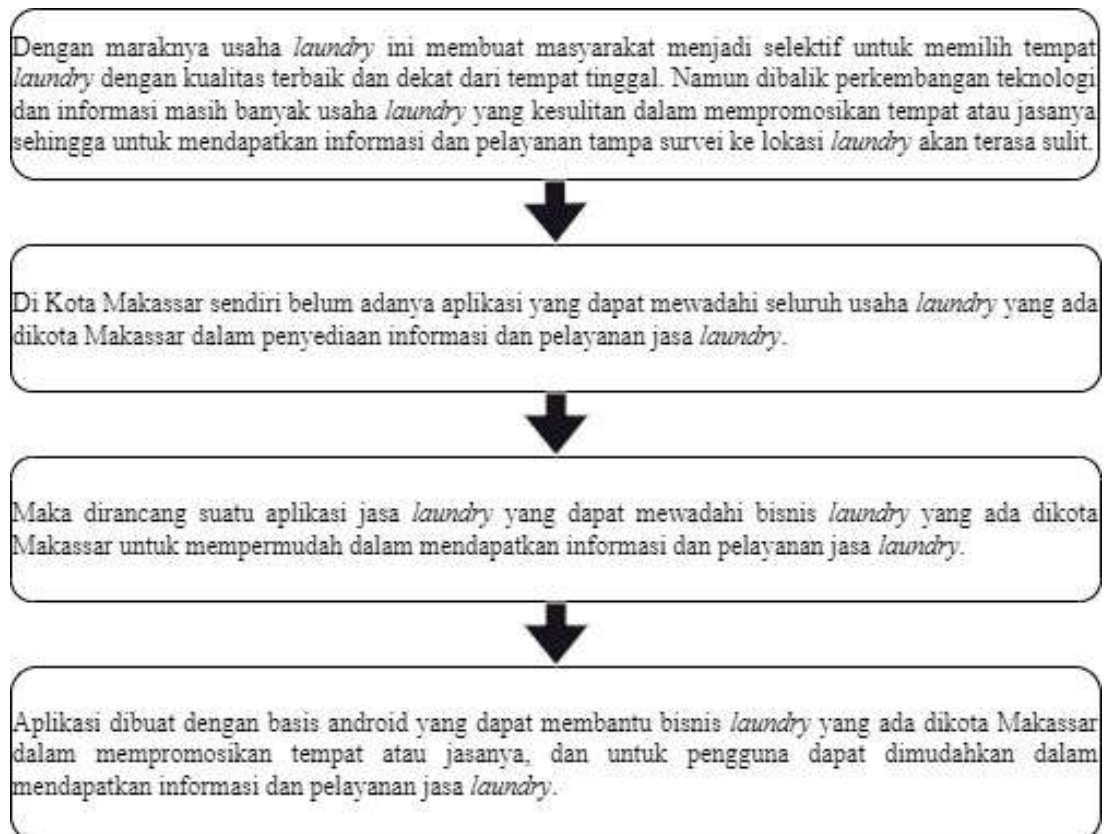
Tabel II. 1 *State of The Art*

No	Nama Peneliti	Judul Penelitian	Tahun Penelitian	Metode Penelitian	Hasil Penelitian
1	Kamil. H, Duhani. A	Pembanguna n Sistem Informasi Pelayanan Jasa <i>Laundry</i> Berbasis <i>Web</i> Dengan Fitur <i>Mobile</i> Pada <i>21 Laundry</i> Padang	2016	Metode <i>waterfall</i>	Aplikasi ini mampu menampilkan informasi yang dibutuhkan pemilik atau karyawan melalui aplikasi <i>web</i> maupun pelanggan melalui aplikasi <i>mobile</i> .
2	Simargolan g. Y. M, Nasution. N	Aplikasi Pelayana Jasa <i>Laundry</i> Berbasis <i>WEB</i> (Studi Kasus: Pelangi <i>Laundry</i> Kisaran)	2018	Metode Analisis Deskriptif	Aplikasi dapat memberikan solusi alternatif untuk memudahkan pelanggan dalam memperoleh informasi yang <i>up</i> <i>to date</i> dari Pelangi <i>Laundry</i> tentang jasa yang ditawarkan.
3	Susanto. E,	Sistem	2019	Metode	Sistem ini dapat

No	Nama Peneliti	Judul Penelitian	Tahun Penelitian	Metode Penelitian	Hasil Penelitian
	Utami. H. T, Hermanto. D	Informasi Pemesanan <i>Laundry</i> Berbasis Android di Kota Palembang		<i>Rational Unified Process (RUP)</i>	mempermudah pelanggan dalam mencari tempat <i>laundry</i> terdekat dengan kualitas terbaik berdasarkan dari hasil penilaian.
4	Sylviana. F, Christina. S, Sunartie	Aplikasi Pemesanan Laundry Berbasis SMS <i>Gateway</i>	2019	Metode <i>Waterfall</i>	aplikasi pemesanan laundry yang berbasis sms gateway, aplikasi ini bertujuan untuk menyediakan pemesanan antar jemput laundry dan fasilitas dalam pemesanan laundry.
5	Iqbal. M, B. L. L. P, Kurniati. N	Penerapan Metode <i>Haversine</i> Pada Sistem Informasi	2021	Metode <i>Haversine Formula</i>	Hasil dari Penelitian ini adalah terbentuknya aplikasi pencarian

No	Nama Peneliti	Judul Penelitian	Tahun Penelitian	Metode Penelitian	Hasil Penelitian
		Geografis Pencarian Lokasi <i>Laundry</i> Terdekat di Kota Makassar			yang dapat membantu masyarakat sekitar dalam mencari <i>laundry</i> terdekat dari lokasi mereka berada.

II.3 Kerangka Pemikiran

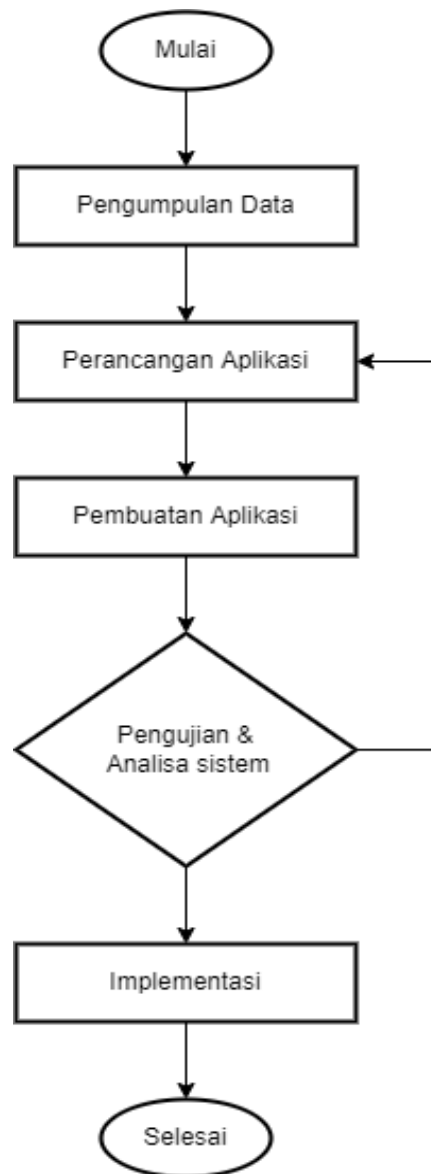


Gambar II. 10 Kerangka Pikir

BAB III METODE PENELITIAN

III.1 Tahapan Penelitian

Tahapan penelitian yang akan dilakukan digambarkan pada Gambar III.1 sebagai berikut:



Gambar III. 1 Tahapan Penelitian

Adapun penjelasan pada tahapan penelitian di atas adalah:

1. Pengumpulan Data

Agar penelitian memperoleh informasi/data yang diperlukan untuk tujuan penelitian maka dilakukan pengumpulan data. Di mana data yang dikumpulkan valid, akurat, terpercaya, dan dapat dipertanggung jawabkan. Data dikumpulkan bersumber dari usaha *laundry* yang terdaftar di internet atau Google Maps.

2. Perancangan Aplikasi

Perancangan aplikasi dilakukan setelah semua kebutuhan dalam tahap sebelumnya terpenuhi. Perancangan dilakukan agar pelaksanaan penelitian tepat dan terorganisir dengan baik dalam setiap unit sehingga dapat mencapai tujuan yang saling terhubung dan menghindari kesalahan yang terjadi serta memberikan gambaran pada peneliti baik dari sisi perancangan sistem maupun perancangan *database*, dalam perancangan aplikasi *laundry* digunakan metode *haversine formula* dalam pencarian *laundry* terdekat berdasarkan lokasi.

3. Pembuatan Aplikasi

Pembuatan aplikasi dilakukan berdasarkan dari perancangan aplikasi yang telah dilakukan sebelumnya. Pada tahap ini akan dilakukan pengolahan data dari data yang telah dikumpulkan sehingga menjadi salah satu komponen pada aplikasi, selanjutnya dilakukan pengkodean untuk membangun sistem pada aplikasi sehingga berjalan sesuai dengan tujuan penelitian.

4. Pengujian dan Analisa Sistem

Tahap selanjutnya akan dilakukan pengujian dan analisa sistem pada aplikasi. Pengujian dilakukan dengan menguji apakah sistem berjalan dengan baik tanpa adanya kesalahan/*error*, pengecekan dilakukan juga untuk fitur-fitur, *database*, dan kenyamanan saat digunakan sesuai dengan apa yang diharapkan. Setelah dilakukan pengujian maka dilakukan penganalisaan terhadap sistem sehingga jika ada yang tidak sesuai dapat direvisi kembali sehingga dapat berjalan sesuai kebutuhan.

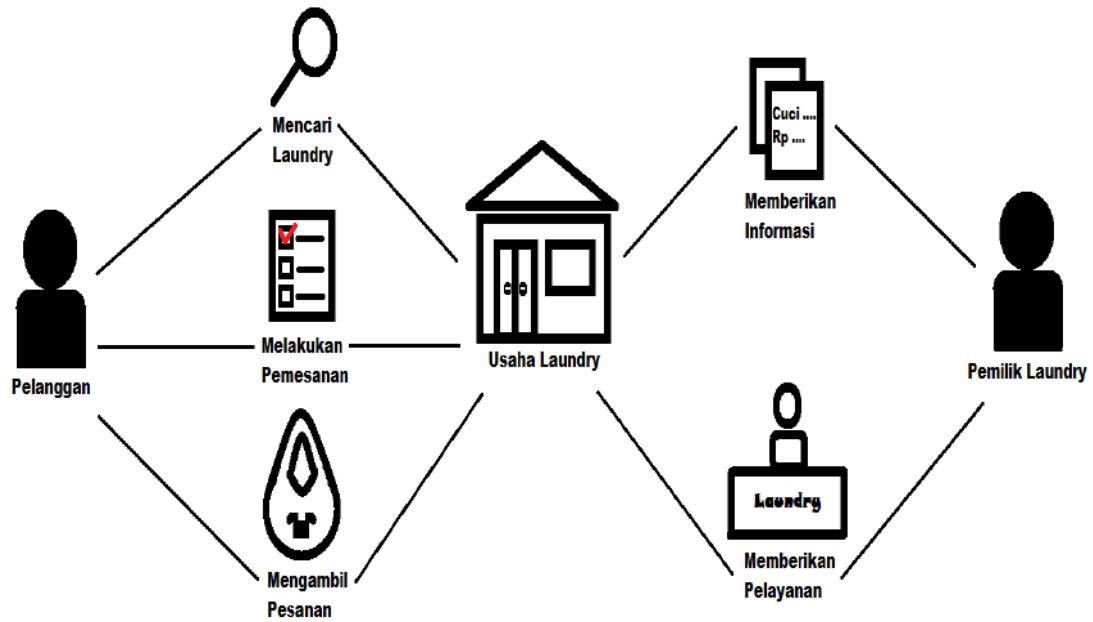
5. Implementasi

Jika semua tahap telah dilaksanakan dan sistem berjalan tanpa adanya masalah maka aplikasi telah siap untuk diimplementasikan.

III.2 Perancangan Penelitian

III.2.1 Sistem yang Sedang Berjalan

Sistem yang berjalan saat ini digambarkan pada gambar III.2 berikut:



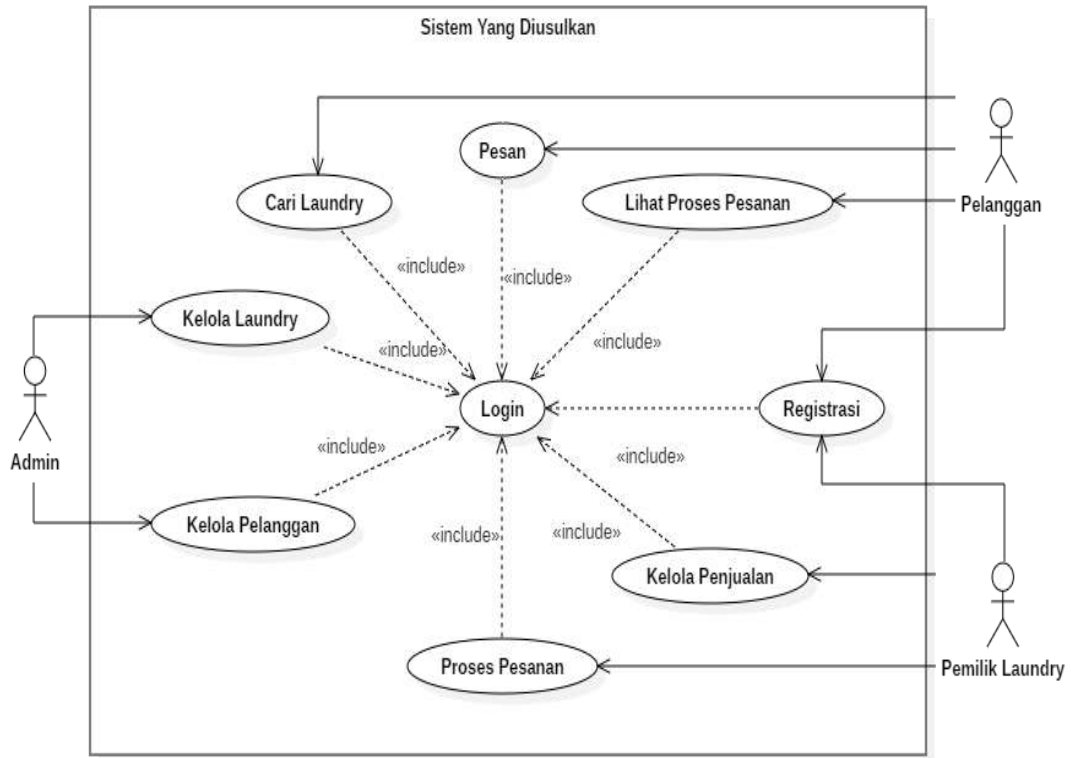
Gambar III. 2 Sistem Yang Sedang Berjalan

Sistem yang berjalan saat ini informasi dan pelayanan tempat jasa *laundry* masih memakai sistem manual yang di mana untuk mendapatkan informasi dan pelayanan, pelanggan harus datang ke tempat jasa *laundry*.

III.2.2 Sistem yang Diusulkan

III.2.3 Use Case Diagram

Sistem yang diusulkan digambarkan seperti terlihat pada gambar III.3 berikut:



Gambar III. 3 Sistem yang Diusulkan

Sistem yang diusulkan adalah aplikasi jasa *laundry* berbasis Android. Di mana admin dapat mengelola data usaha *laundry* dan data pelanggan yang telah terdaftar, untuk pemilik *laundry* dapat mengelola penjualan jasa *laundry*, melakukan proses pesanan dari pelanggan, sedangkan pelanggan dapat melakukan pencarian jasa *laundry*, melihat informasi dari usaha *laundry* dan melakukan pemesanan serta melihat proses/status pesanan. Untuk proses pencarian *laundry* terdekat digunakan metode *haversine formula* dan dibutuhkan data masukan berupa titik awal dan titik akhir untuk menentukan jarak antara kedua titik tersebut.

III.2.3.1 Definisi Aktor

Sistem yang dibangun menggunakan 3 aktor masing-masing yaitu admin, pemilik *laundry* dan pengguna/pelanggan. Penjelasan dari masing-masing aktor pada sistem yang akan dibuat sebagai berikut:

Tabel III. 1 Definisi Aktor

Aktor	Deskripsi
Admin	Pengguna sistem yang memiliki akses terhadap semua pengolahan data dari pengguna atau <i>user</i> .
Pemilik <i>laundry</i>	Pengguna sistem yang memiliki akses untuk registrasi, menginput harga, dan mengolah pesanan.
Pelanggan	Pengguna sistem yang memiliki akses untuk melakukan pencarian, pemesanan, dan melihat proses pesanan <i>laundry</i> .

III.2.3.2 Definisi Use Case

Pada penjelasan *use case* dari sistem yang akan dibuat berdasarkan aktor. Definisi *use case* dari setiap aktor diuraikan sebagai berikut:

Tabel III. 2 Definisi Use Case Pada Aktor Admin

Use Case	Deskripsi
Kelola <i>laundry</i>	Admin dapat mengolah data usaha <i>laundry</i> seperti mengubah dan menghapus yang terdaftar.
Kelola pengguna/pelanggan	Admin dapat menambah, mengubah, dan menghapus data dari pengguna/pelanggan.

Tabel III. 3 Definisi *Use Case* Pada Pemilik *Laundry*

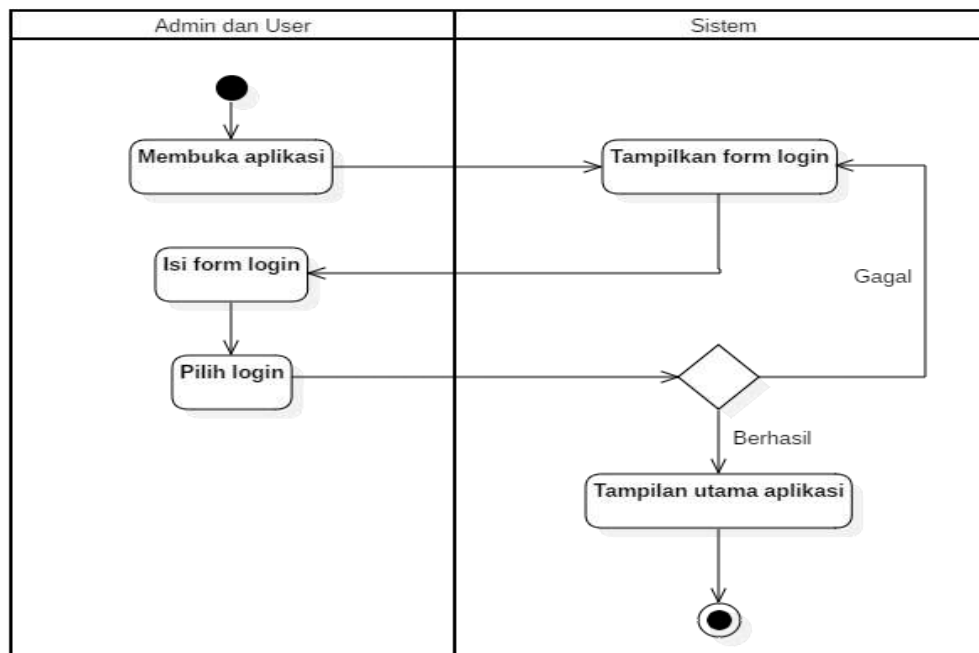
<i>Use Case</i>	Deskripsi
Registrasi	Pemilik dapat melakukan registrasi akun sebagai pemilik <i>laundry</i> untuk mendaftarkan usaha <i>laundry</i> di aplikasi.
Kelola penjualan	Pemilik dapat melakukan penginputan harga sesuai dengan ketentuan harga dari pemilik <i>laundry</i> .
Proses pemesanan	Pemilik dapat memproses pesanan dari pengguna/pelanggan.

Tabel III. 4 Definisi *Use Case* Pada Pengguna/Pelanggan

<i>Use Case</i>	Deskripsi
Cari <i>laundry</i>	Pengguna/pelanggan dapat mencari jasa <i>laundry</i> terdekat berdasarkan lokasi
Pesan	Pengguna/pelanggan dapat melakukan pemesanan sesuai dengan usaha <i>laundry</i> yang telah dipilih.
Melihat proses pemesanan	Pengguna/pelanggan dapat melihat status dari pesanan berdasarkan pesanan yang diproses.
Registrasi	Pengguna/pelanggan dapat melakukan registrasi akun sebagai pengguna/pelanggan untuk menggunakan aplikasi.

III.2.4 Activity Diagram

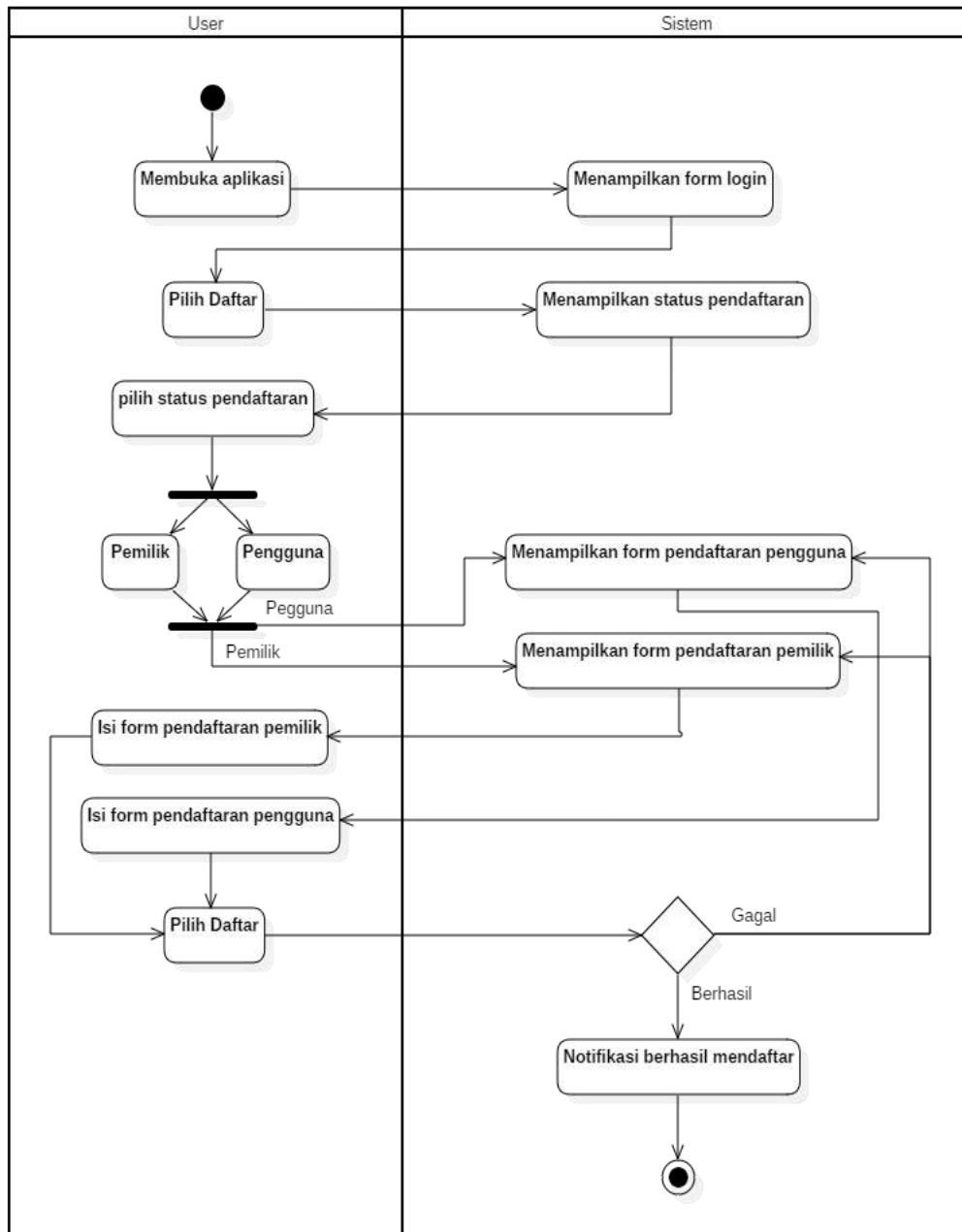
1. Activity Login Admin, Pemilik Laundry dan Pengguna/Pelanggan



Gambar III. 4 Activity Login Admin, Pemilik Laundry dan Pengguna/Pelanggan

Activity diagram gambar III.4 menjelaskan bahwa admin, pemilik laundry dan pengguna/pelanggan akan melakukan login dengan mengisi form menggunakan username dan password, setelah login admin, laundry dan pelanggan diarahkan ke tampilan utama.

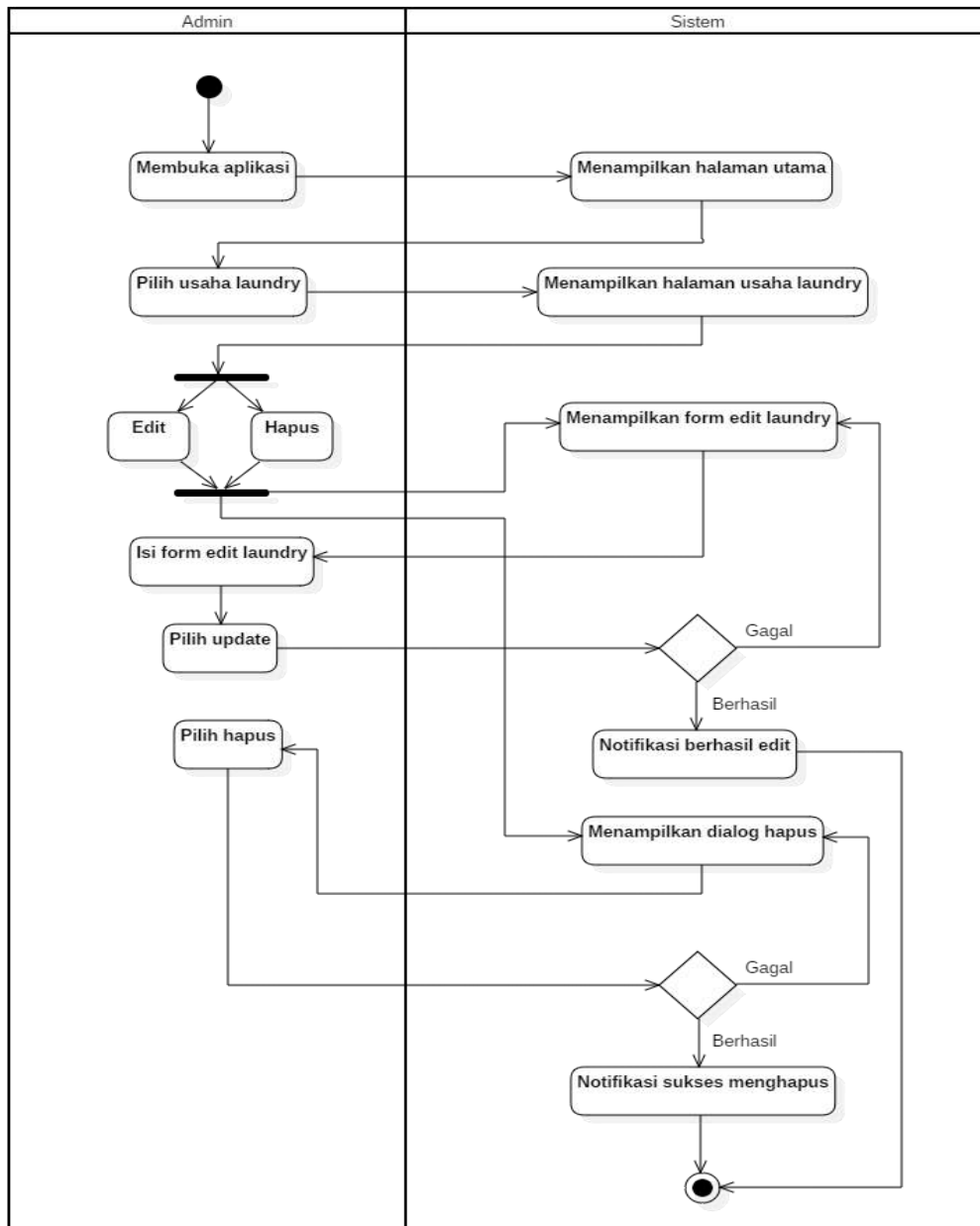
2. Activity Registrasi Pemilik Laundry dan Pengguna/Pelanggan



Gambar III. 5 Activity Registrasi Pemilik Laundry dan Pengguna/Pelanggan

Activity diagram gambar III.5 menjelaskan pemilik laundry mendaftarkan usahanya dan pengguna/pelanggan melakukan registrasi akun.

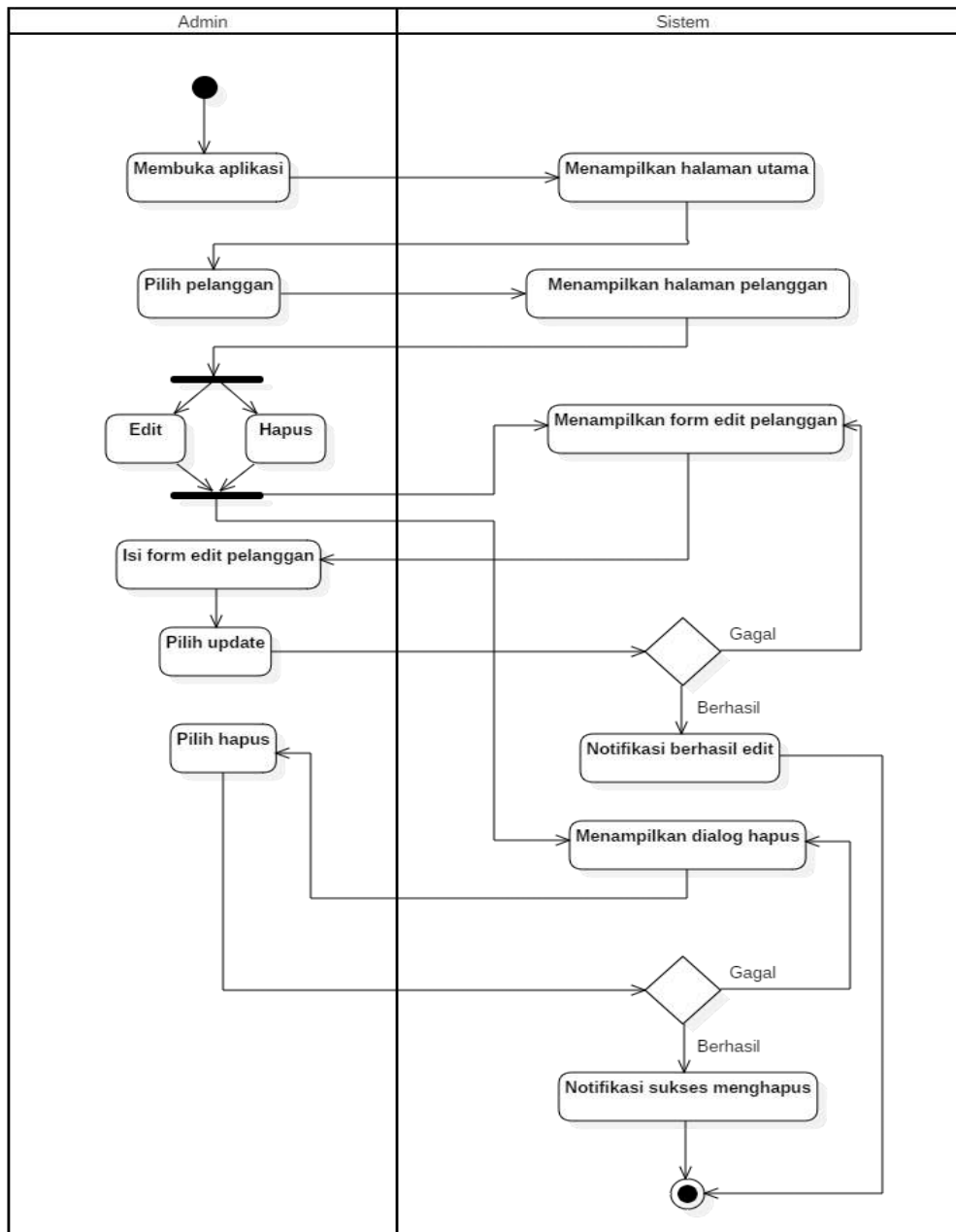
3. Activity Admin Mengelola Laundry



Gambar III. 6 Activity Admin Mengelola Laundry

Activity diagram gambar III.6 menjelaskan admin mengolah data usaha laundry seperti mengedit atau menghapus data usaha laundry yang telah terdaftar.

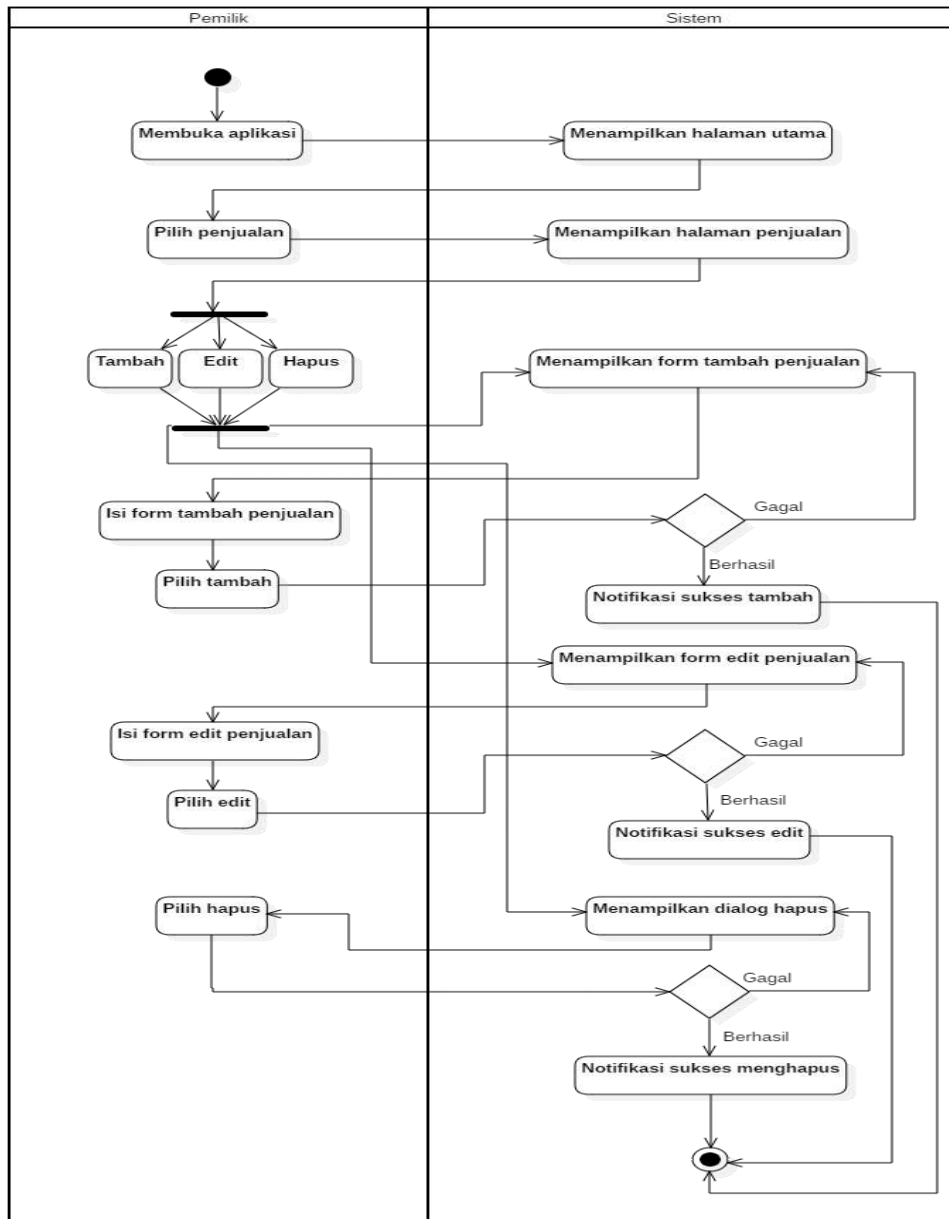
4. Activity Admin Mengelola Pengguna/Pelanggan



Gambar III. 7 Activity Admin Mengelola Pengguna/Pelanggan

Activity diagram gambar III.7 menjelaskan admin mengolah data pengguna/pelanggan seperti mengedit atau menghapus data pengguna/pelanggan.

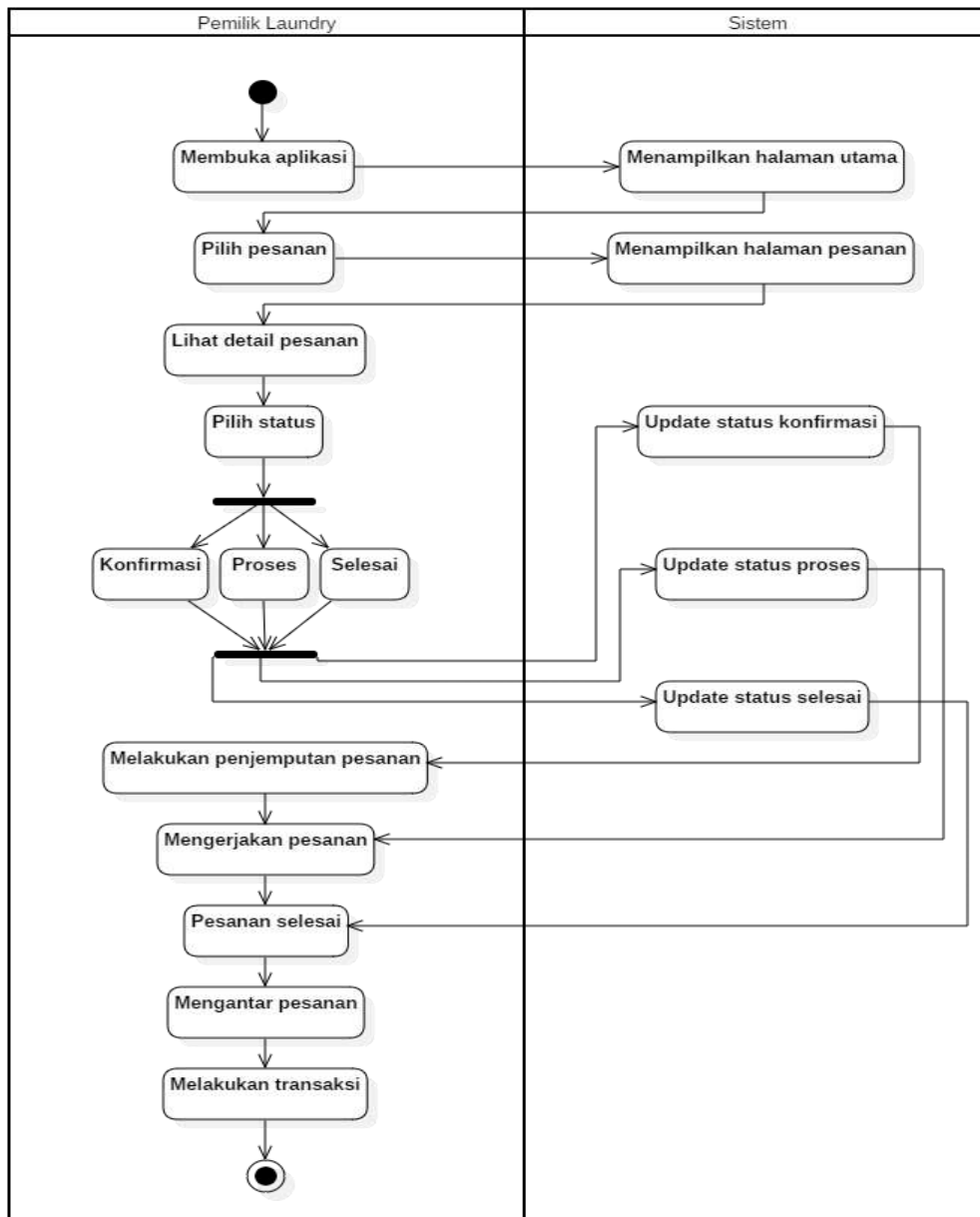
5. Activity Pemilik Laundry Mengelola Penjualan



Gambar III. 8 Activity Pemilik Laundry Mengelola Penjualan

Activity diagram gambar III.8 menjelaskan pemilik laundry mengolah data penjualan seperti menambah, mengedit atau menghapus data penjualan.

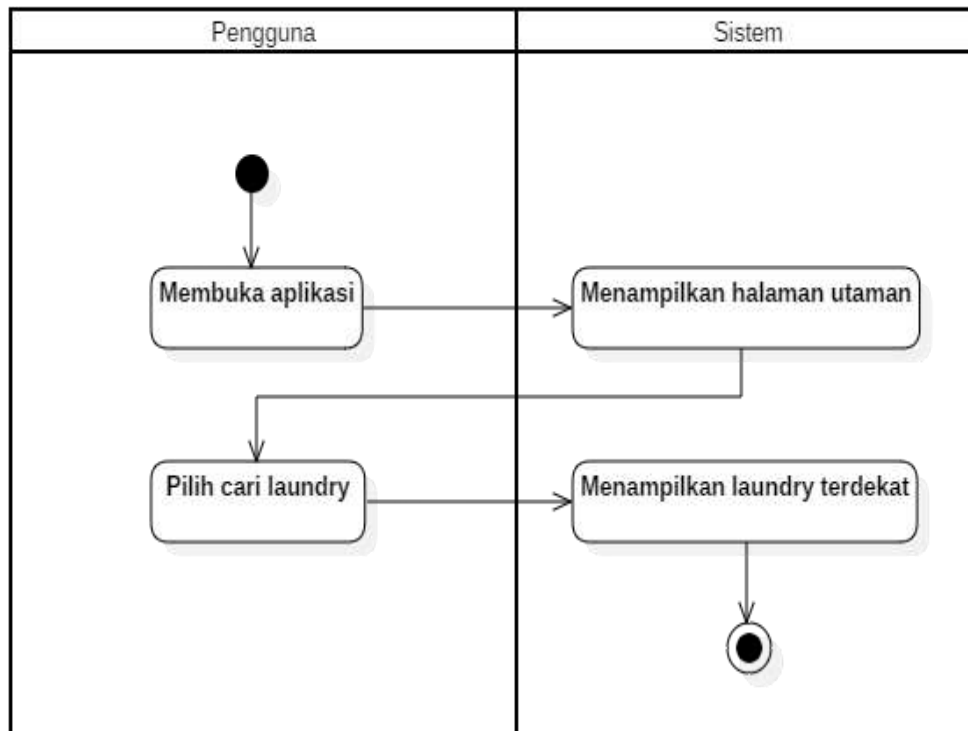
6. Activity Pemilik Laundry Memproses Pesanan



Gambar III. 9 Activity Pemilik Laundry Memproses Pesanan

Activity diagram gambar III.9 menjelaskan pemilik laundry melakukan proses pesanan, mengubah, dan mengkonfirmasi pesanan dari pengguna/pelanggan dan melakukan antar jemput pesanan.

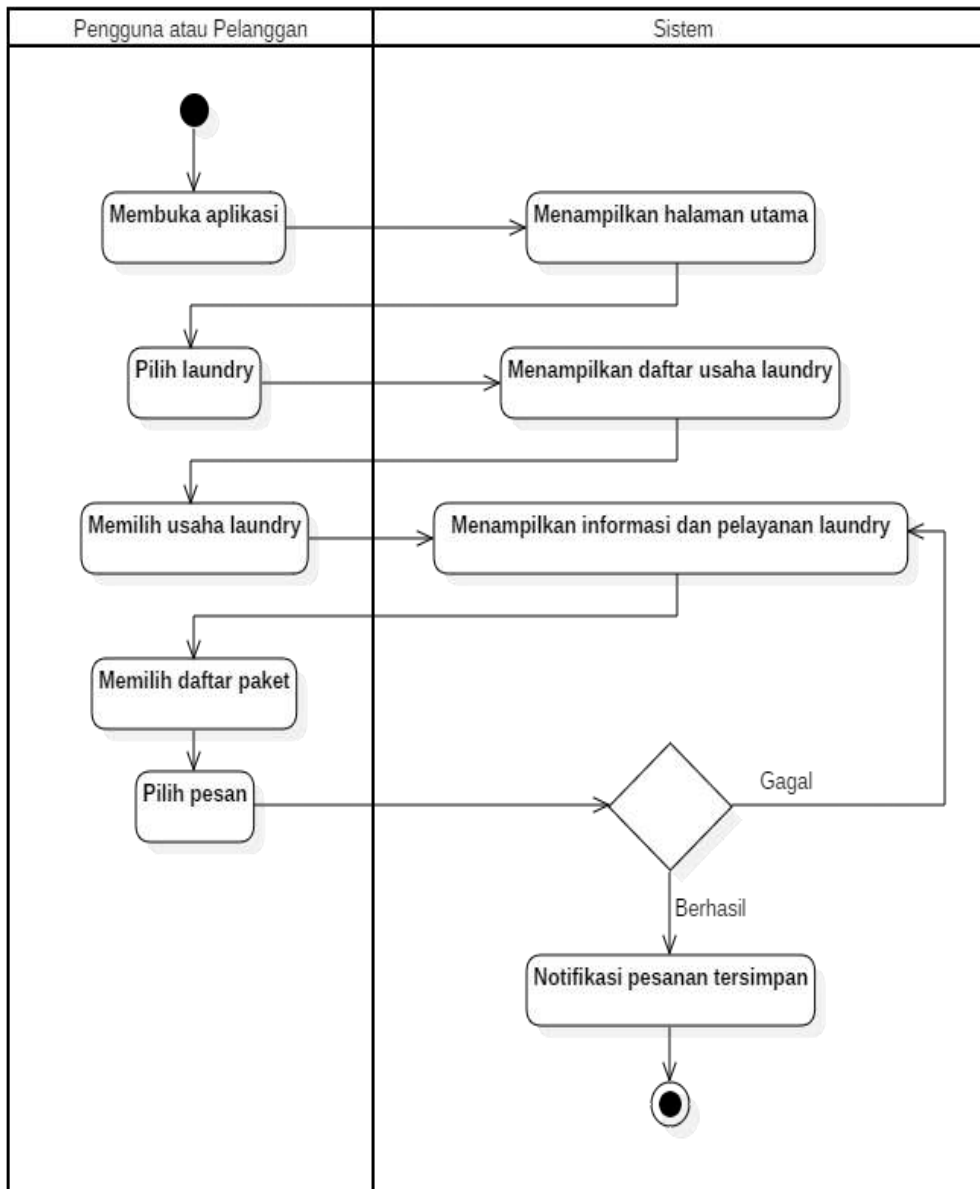
7. Activity Pengguna/Pelanggan Mencari Laundry



Gambar III. 10 Activity Pengguna/Pelanggan Mencari Laundry

Activity diagram gambar III.10 menjelaskan pengguna/pelanggan melakukan pencarian laundry berdasarkan lokasi terdekat.

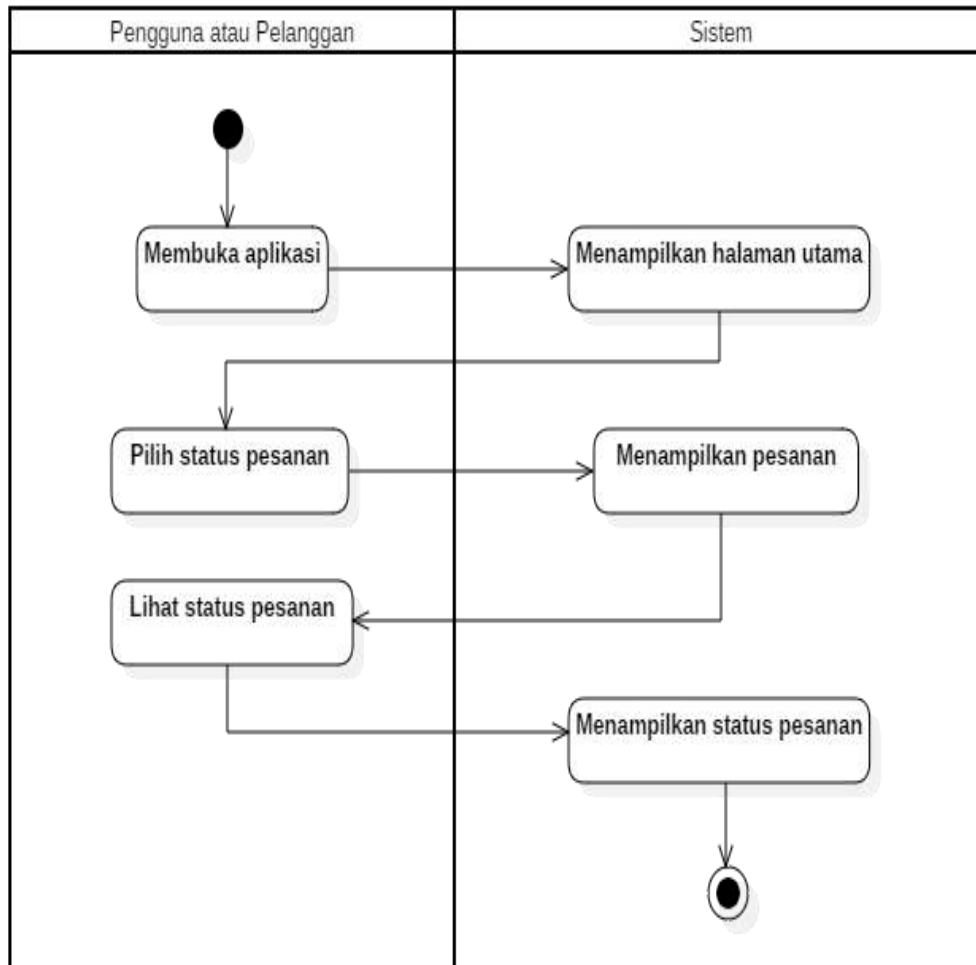
8. Activity Pengguna/Pelanggan Memesan Jasa



Gambar III. 11 Activity Pelanggan Memesan Jasa

Activity diagram gambar III.11 menjelaskan pengguna/pelanggan memilih usaha *laundry*, melihat informasi layanan jasa dan melakukan pemesanan jasa *laundry*.

9. Activity Pengguna/Pelanggan Melihat Proses Pemesanan

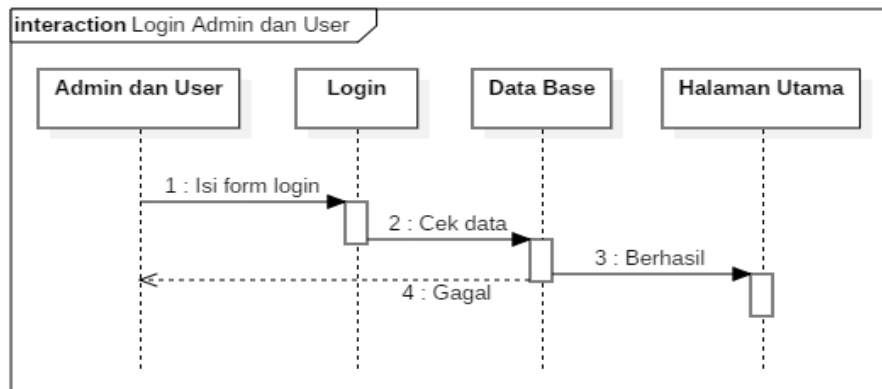


Gambar III. 12 Activity Pengguna/Pelanggan Melihat Proses Pemesanan

Activity diagram gambar III.12 menjelaskan pengguna/pelanggan melihat proses pemesanan atau status pesanan dari pemilik jasa *laundry*.

III.2.5 Sequence Diagram

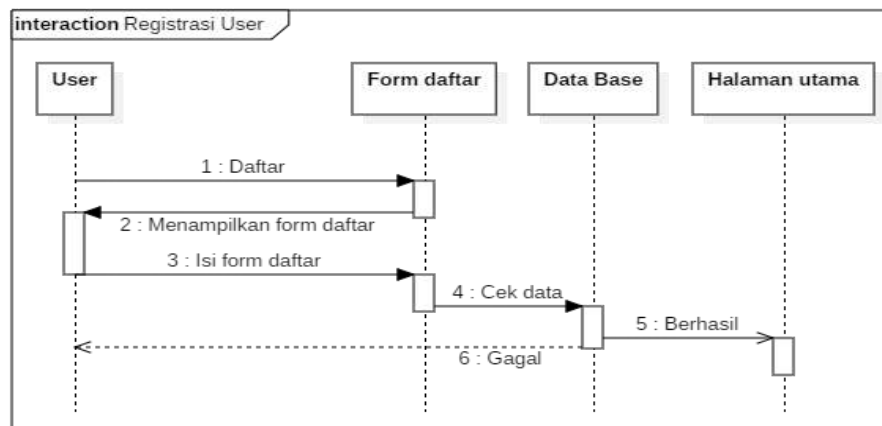
1. Sequence Login Admin, Pemilik Laundry dan Pengguna/Pelanggan



Gambar III. 13 Sequence Login Admin, Pemilik Laundry dan Pengguna/Pelanggan

Sequence diagram gambar III.13 dimulai dengan admin, pemilik laundry, dan pengguna/pelanggan mengakses aplikasi yang diawal sistem menampilkan form login, admin dan user diminta untuk memasukkan username dan password untuk melakukan login, data login akan dicek dari database jika sesuai maka akan diarahkan ke tampilan utama dan jika data tidak sesuai maka akan diarahkan kembali ke form login.

2. Sequence Registrasi Pemilik Laundry dan Pengguna/Pelanggan

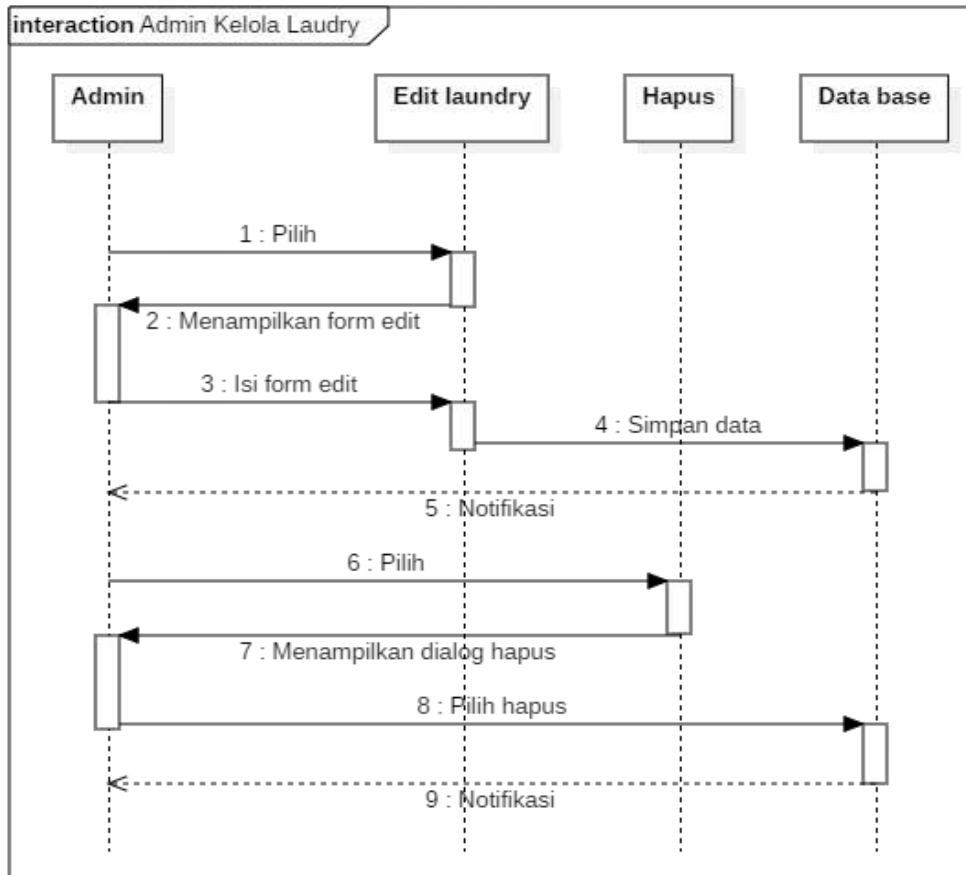


Gambar III. 14 Sequence Registrasi Pemilik Laundry dan Pengguna/Pelanggan

Sequence diagram gambar III.14 dimulai dari pemilik laundry atau pengguna/pelanggan ingin mendaftar akun mengisi form setelah mengisi form dan

menekan tombol daftar akan dilakukan pengecekan data, jika data sudah ada maka akan muncul notifikasi kesalahan dan jika data belum ada maka berhasil daftar dan akan menampilkan halaman utama.

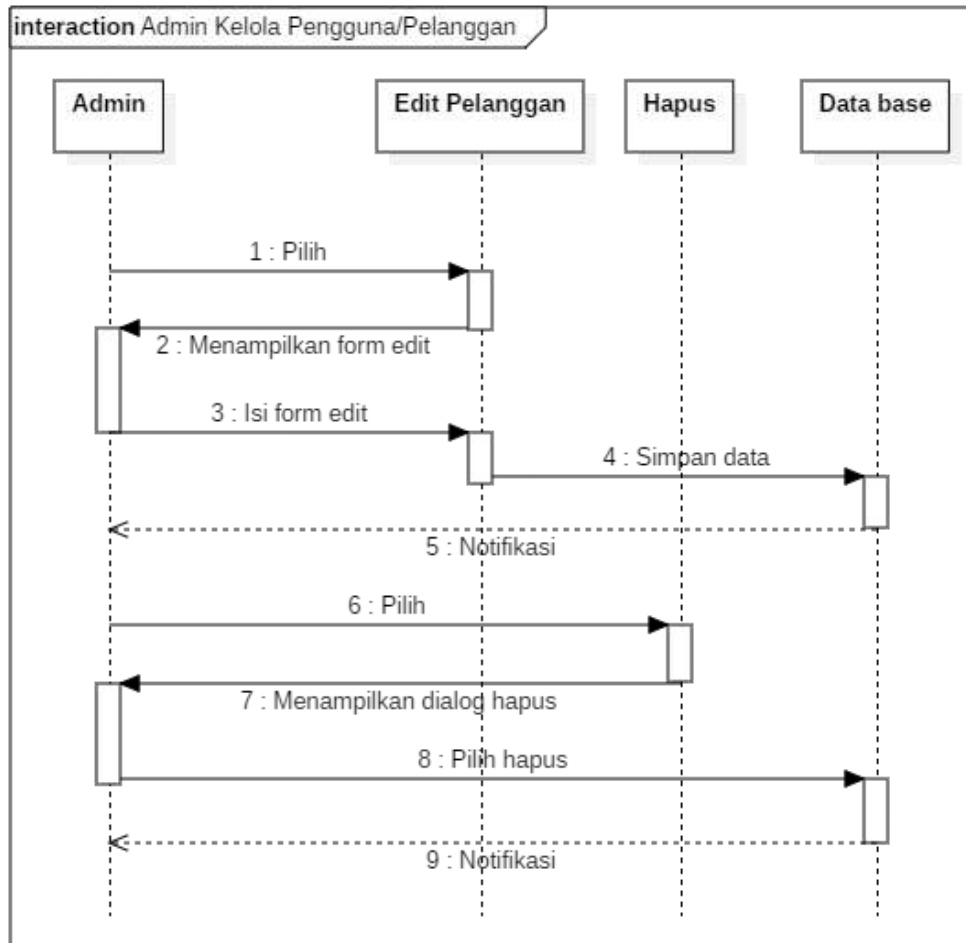
3. *Sequence Admin Mengelola Laundry*



Gambar III. 15 *Sequence Admin Mengelola Laundry*

Sequence diagram gambar III.15 menjelaskan bahwa admin akan melakukan kelola usaha *laundry* baik itu mengubah dan menghapus. Jika salah satu data berhasil dikelola maka akan menampilkan notifikasi sukses kepada admin, jika gagal maka akan menampilkan notifikasi gagal.

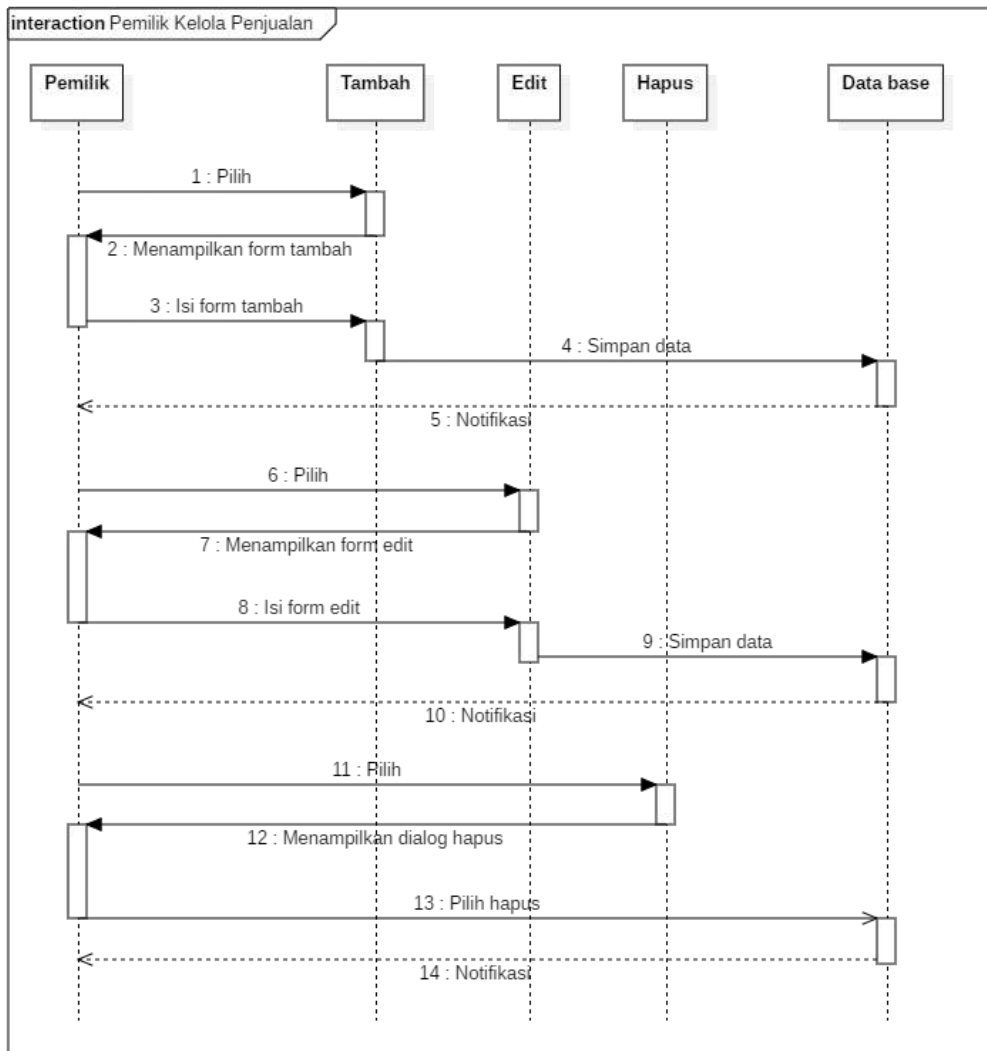
4. Sequence Admin Mengelola Pengguna/Pelanggan



Gambar III. 16 Sequence Admin Mengelola Pengguna/Pelanggan

Sequence diagram gambar III.16 menjelaskan bahwa admin akan melakukan kelola data pengguna/pelanggan baik itu mengubah dan menghapus. Jika salah satu data berhasil dikelola maka akan menampilkan notifikasi sukses kepada admin dan jika gagal maka akan menampilkan notifikasi gagal.

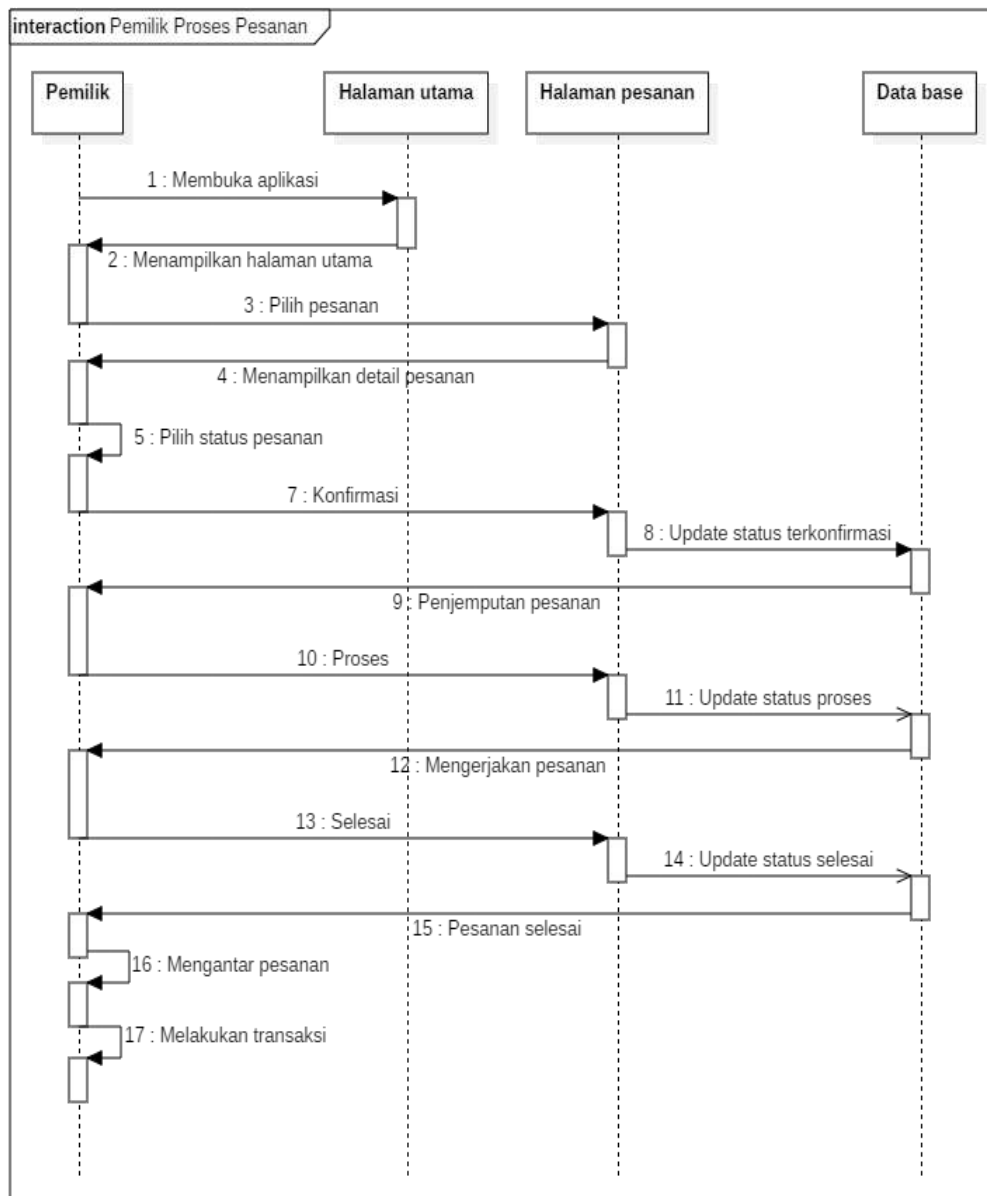
5. Sequence Pemilik Laundry Mengelola Penjualan



Gambar III. 17 Sequence Pemilik Laundry Mengelola Penjualan

Sequence diagram gambar III.17 menjelaskan bahwa pemilik laundry akan melakukan kelola penjualan jasa baik itu menambah, mengubah dan menghapus jika salah satu data berhasil di kelola maka akan menampilkan notifikasi sukses kepada admin, jika gagal maka akan menampilkan notifikasi gagal.

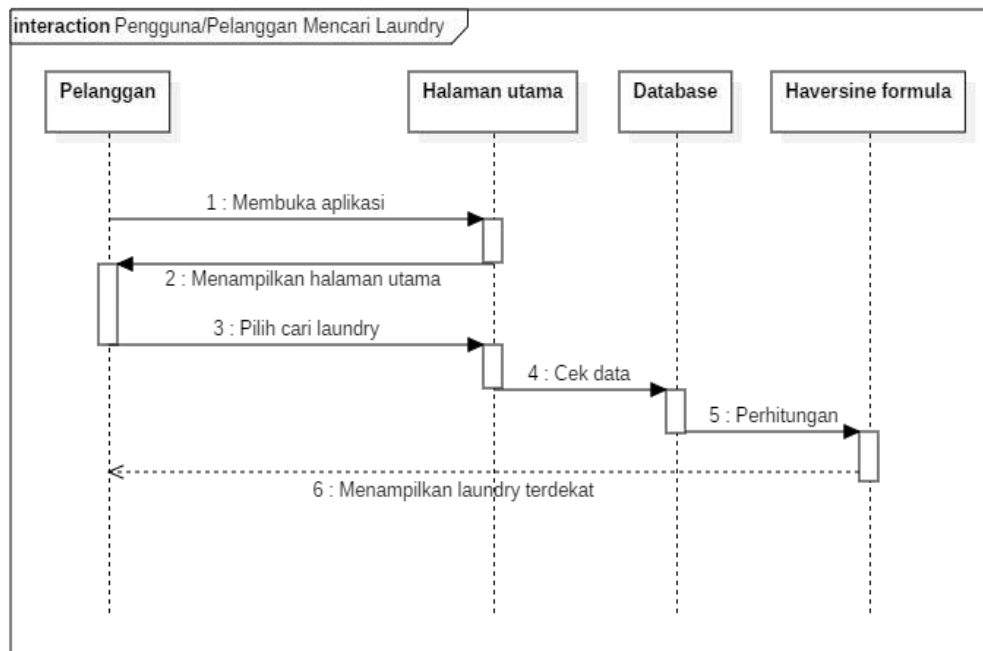
6. Sequence Pemilik Laundry Memproses Pesanan



Gambar III. 18 Sequence Pemilik Laundry Memproses Pesanan

Sequence diagram gambar III.18 menjelaskan bahwa pemilik *laundry* akan melakukan proses pesanan seperti melihat detail pesanan, status pesanan, penjemputan, dan pengantaran pesanan dan pengerjaan pesanan dari pengguna/pelanggan.

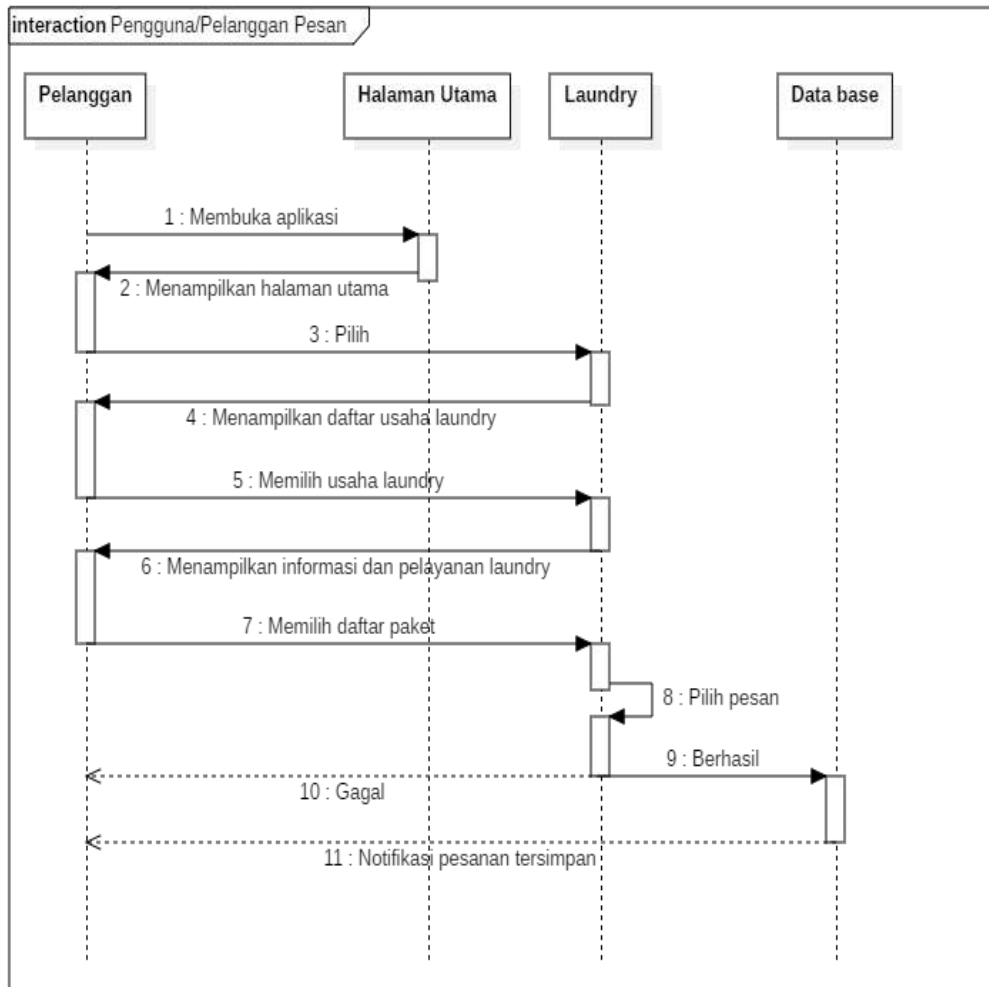
7. Sequence Pengguna/Pelanggan Mencari Laundry



Gambar III. 19 *Sequence* Pengguna/Pelanggan Mencari Laundry

Sequence diagram gambar III.19 menjelaskan bahwa pengguna/pelanggan laundry akan melakukan pencarian laundry terdekat menggunakan *haversine formula* untuk menentukan jarak antara kordinat pengguna/pelanggan dengan kordinat laundry.

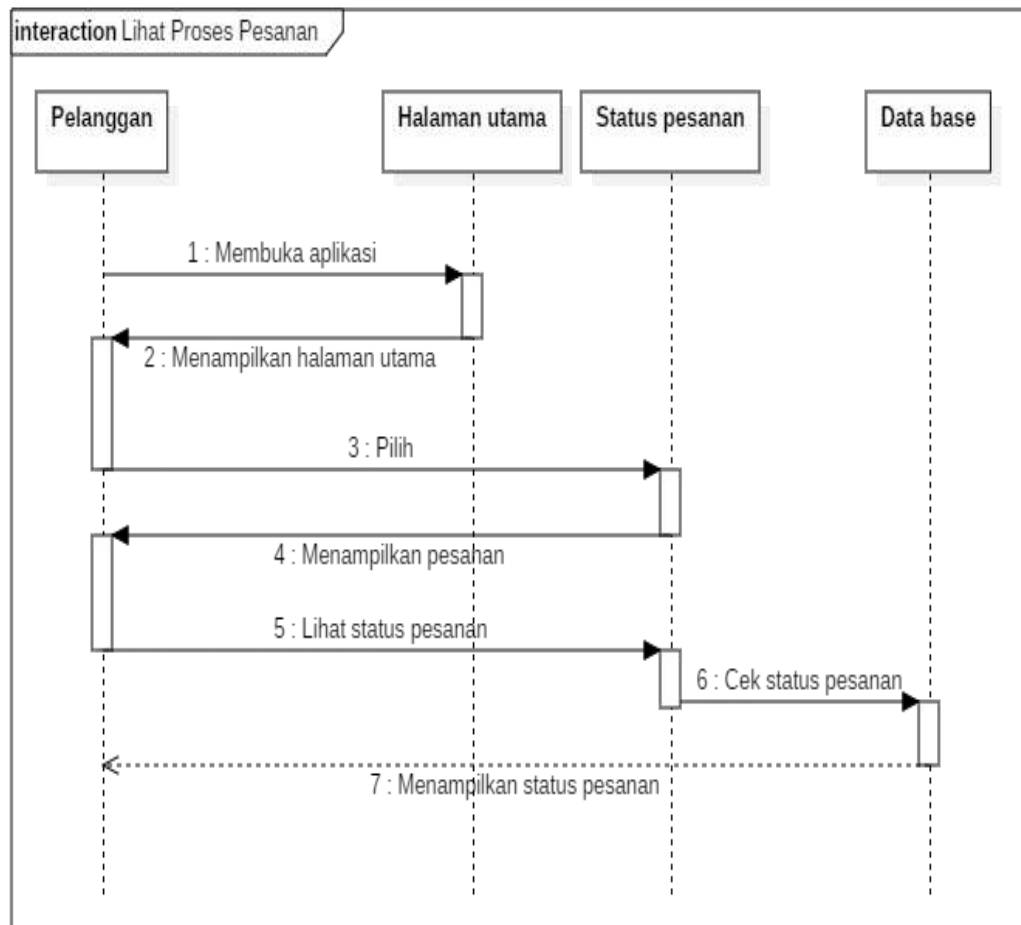
8. Sequence Pengguna/Pelanggan Memesan Jasa



Gambar III. 20 Sequence Pengguna/Pelanggan Memesan Jasa

Sequence diagram gambar III.20 menjelaskan bahwa pengguna/pelanggan akan memilih usaha *laundry* untuk melihat informasi atau melakukan pemesanan jasa pada usaha *laundry*.

9. Sequence Pengguna/Pelanggan Melihat Proses Pesanan

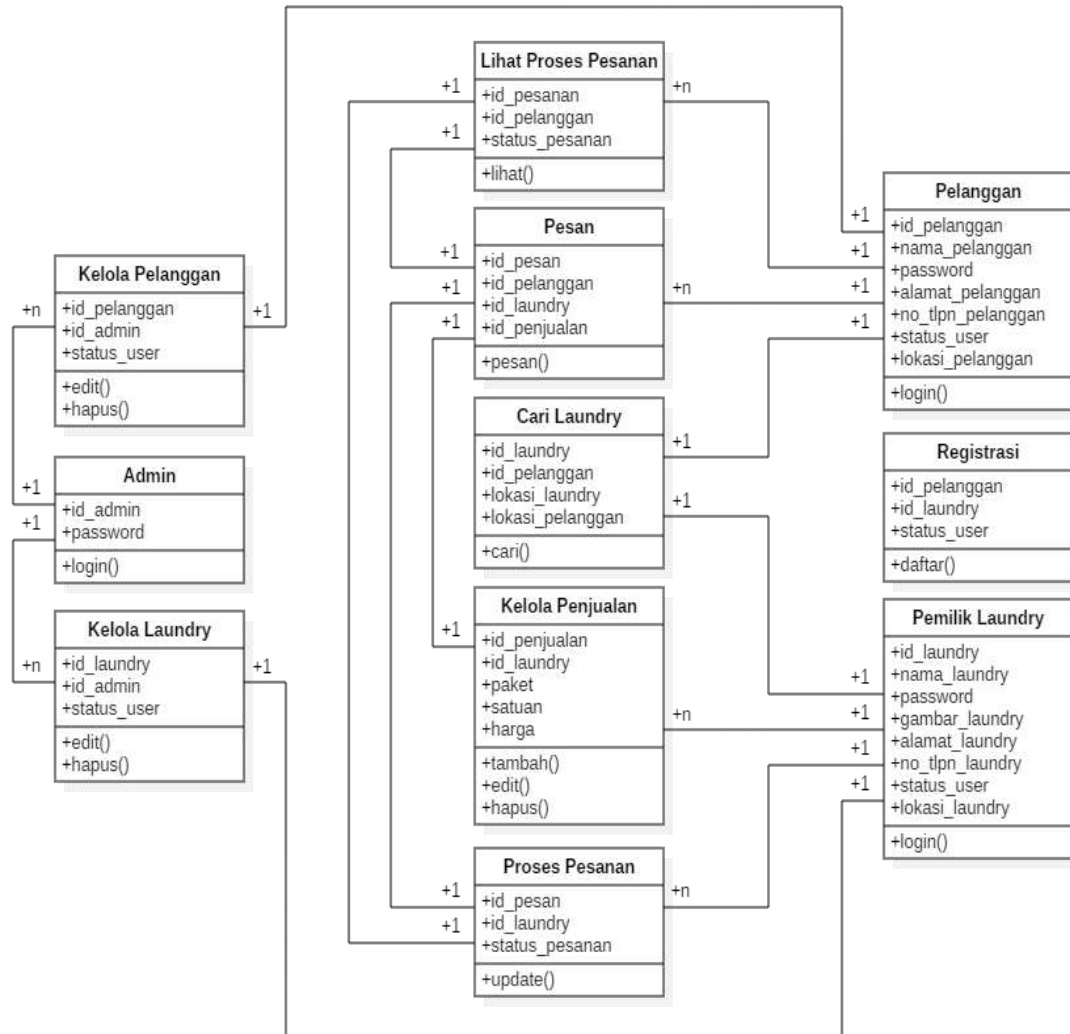


Gambar III. 21 *Sequence* Pengguna/Pelanggan Melihat Proses Pesanan

Sequence diagram gambar III.21 menjelaskan bahwa pengguna/pelanggan akan mencari atau memilih usaha *laundry* untuk melihat informasi atau melakukan pemesanan jasa pada usaha *laundry*.

III.2.6 Class Diagram

Class diagram merupakan diagram yang digunakan untuk menampilkan kelas-kelas yang ada dalam sistem berupa objek yang sedang dikembangkan dari kelas yang satu ke kelas yang lain yang mempunyai relasi.



Gambar III. 22 Class Diagram

IV.2.6.1 Kamus Data

1. Admin

Tabel III. 5 Admin

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_admin	<i>Text</i>	-	Id admin
2	<i>Password</i>	<i>Text</i>	-	<i>Password</i> admin

2. Kelola Laundry

Tabel III. 6 Kelola Laundry

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_laundry	<i>Text</i>	-	Id laundry
2	Id_admin	<i>Text</i>	-	Id admin
3	Status_user	<i>Enum</i>	'Admin','Laundry'	Status pengguna

3. Kelola Pengguna

Tabel III. 7 Kelola Pengguna

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_pelanggan	<i>Text</i>	-	Id pelanggan
2	Id_admin	<i>Text</i>	-	Id admin
3	Status_user	<i>Enum</i>	'Admin','Pengguna'	Status pengguna

4. Registrasi

Tabel III. 8 Registrasi

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_pelanggan	<i>Text</i>	-	Id pelanggan
2	Id_laundry	<i>Text</i>	-	Id laundry
3	Status_user	<i>Enum</i>	'Pelanggan','Laundry'	Status pengguna

5. Pemilik *Laundry*

Tabel III. 9 Pemilik *Laundry*

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	<i>Id_laundry</i>	<i>Text</i>	-	<i>Id laundry</i>
2	<i>Nama_laundry</i>	<i>Varchar</i>	-	<i>Nama laundry</i>
3	<i>Password</i>	<i>Text</i>	-	<i>Password laundry</i>
4	<i>Gambar_laundry</i>	<i>Varchar</i>	-	<i>Gambar laundry</i>
5	<i>Alamat_laundry</i>	<i>Varchar</i>	-	<i>Alamat laundry</i>
6	<i>No_tlpn_laundry</i>	<i>Integer</i>	12	Nomor telpon <i>laundry</i>
7	<i>Status_user</i>	<i>Enum</i>	' <i>Laundry</i> '	Status pengguna
8	<i>Lokasi_laundry</i>	<i>Integer</i>	-	Kordinat <i>laundry</i>

6. Pelanggan

Tabel III. 10 Pelanggan

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	<i>Id_pelanggan</i>	<i>Text</i>	-	<i>Id pelanggan</i>
2	<i>Nama_pelanggan</i>	<i>Varchar</i>	-	<i>Nama pelanggan</i>
3	<i>Password</i>	<i>Text</i>	-	<i>Password pelanggan</i>
4	<i>Alamat_pelanggan</i>	<i>Varchar</i>	-	<i>Alamat pelanggan</i>
5	<i>No_tlpn_pelanggan</i>	<i>Integer</i>	12	Nomor telpon pelanggan
6	<i>Status_user</i>	<i>Enum</i>	'Pelanggan'	Status pengguna
7	<i>Lokasi_pelanggan</i>	<i>Integer</i>	-	Kordinat pelanggan

7. Kelola Penjualan

Tabel III. 11 Kelola Penjualan

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	<i>Id_penjualan</i>	<i>Text</i>	-	<i>Id penjualan</i>
2	<i>Id_laundry</i>	<i>Text</i>	-	<i>Id laundry</i>

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
3	Paket	<i>Varchar</i>	-	Paket <i>laundry</i>
4	Satuan	<i>Varchar</i>	-	Satuan kg/pcs
5	Harga	<i>Integer</i>	-	Harga dari paket cuci

8. Proses Pesanan

Tabel III. 12 Proses Pesanan

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_pesanan	<i>Text</i>	-	Id pesanan
2	Id_laundry	<i>Text</i>	-	Id <i>laundry</i>
3	Status_pesanan	<i>Enum</i>	'Pending', 'Penjemputan', 'Proses', 'Pengantaran', 'Selesai'	Status pesanan

9. Cari Laundry

Tabel III. 13 Cari Laundry

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_laundry	<i>Text</i>	-	Id <i>laundry</i>
2	Id_pelanggan	<i>Text</i>	-	Id pelanggan
3	Lokasi_laundry	<i>Integer</i>	-	Kordinat <i>laundry</i>
4	Lokasi_pelanggan	<i>Integer</i>	-	Kordinat pelanggan

10. Pesan

Tabel III. 14 Pesan

No	<i>Field</i>	<i>Type</i>	Panjang	Keterangan
1	Id_pesanan	<i>Text</i>	-	Id pesanan
2	Id_pelanggan	<i>Text</i>	-	Id pelanggan
3	Id_laundry	<i>Text</i>	-	Id <i>laundry</i>
4	Id_penjualan	<i>Text</i>	-	Id penjualan

11. Lihat Proses Pesanan

Tabel III. 15 Lihat Proses Pesanan

No	Fiel	Type	Panjang	Keterangan
1	Id_pesanan	Text	-	Id pesanan
2	Id_pelanggan	Text	-	Id pelanggan
3	Status_pesanan	Enum	'Pending', 'Penjemputan', 'Proses', 'Pengantaran', 'Selesai'	Status pesanan

III.2.7 Pencarian Laundry Terdekat Menggunakan Haversine Formula



Gambar III. 23 Flowchart Tahapan Haversine Formula

Flowchart di atas menjelaskan tahapan *haversine formula*, dimulai dengan mengaktifkan GPS untuk menentukan kordinat *user* setelah itu sistem mengecek/mengambil kordinat *laundry* yang berada di *database* kemudian jika dua kordinat sudah terpenuhi maka selanjutnya sistem akan menghitung jarak dengan *haversine formula* menggunakan rumus:

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2}) / 2);$$

$$y = (\text{lat2} - \text{lat1});$$

$$d = \text{sqrt}(x * x + y * y) * r$$

keterangan:

x = *longitude* (latitude)

y = *latitude* (longitude)

d = jarak

r = radius bumi (6371 km)

1 derajat = 0.0174532925 radians

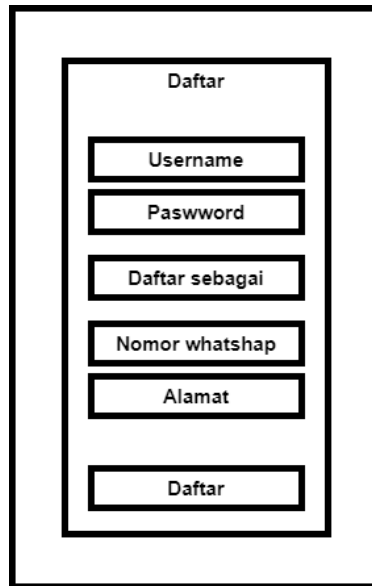
III.2.8 Perancangan *Interface*

1. Tampilan *Form Login*

```
graph TD
    subgraph Form_Login [Form Login]
        direction TB
        Title[Login]
        Username[Username]
        Password[Paswword]
        Submit[Login]
    end
```

Gambar III. 24 Tampilan *Form Login*

2. Tampilan Registrasi



A vertical registration form with a title 'Daftar' at the top. Below the title are six input fields: 'Username', 'Paswword', 'Daftar sebagai', 'Nomor whatshap', 'Alamat', and 'Daftar'.

Gambar III. 25 Tampilan Registrasi

3. Tampilan Admin Mengelola *Laundry*



An admin management form titled 'Data Laundry'. It contains four main sections: a text area for 'Nama Laundry :', 'No Wa :', and 'Alamat :'; a section for 'Lokasi di Peta'; a section for 'Data Paket Laundry'; and a 'Hapus' button at the bottom.

Gambar III. 26 Tampilan Admin Mengelola *Laundry*

4. Tampilan Admin Mengelola Pengguna/Pelanggan

Data Pelanggan	
Nama Pengguna : No Wa : Alamat :	Hapus
Nama Pengguna : No Wa : Alamat :	Hapus
Nama Pengguna : No Wa : Alamat :	Hapus
Nama Pengguna : No Wa : Alamat :	Hapus
Nama Pengguna : No Wa : Alamat :	Hapus

Gambar III. 27 Tampilan Admin Mengelola Pengguna/Pelanggan

5. Tampilan Pemilik *Laundry* Megelola Penjualan

Paket Laundry
Tambah
• Paket : • Satuan : • Harga :
• Paket : • Satuan : • Harga :
• Paket : • Satuan : • Harga :

Gambar III. 28 Tampilan Pemilik *Laundry* Mengelola Penjualan

6. Tampilan Pemilik *Laundry* Memproses Pesanan

Detail Transaksi

Nama Pengguna:
No Wa :
Alamat :

Status Pesanan
.....

Paket :
Harga :
Jumlah :
Total :

Proses

Gambar III. 29 Tampilan Pemilik *Laundry* Proses Pesanan

7. Tampilan Pengguna/Pelanggan Mencari *Laundry*

Hasil Pencarian

Nama Laundry :
Jarak :

Detail

Nama Laundry :
Jarak :

Detail

Nama Laundry :
Jarak :

Detail

Gambar III. 30 Tampilan Pengguna/Pelanggan Mencari *Laundry*

8. Tampilan Pengguna/Pelanggan Memesan Jasa

The screenshot shows a mobile application interface for ordering laundry services. It is titled "Detail Laundry" and contains several sections: a top section with fields for "Nama Laundry :", "No Wa :", and "Alamat : .."; a middle section labeled "Lokasi di Peta"; a section titled "Data Paket" containing fields for "Paket :", "Satuan : Kg/Pcs", and "Harga : .."; and a bottom button labeled "Pesan".

Gambar III. 31 Tampilan Pengguna/Pelanggan Memesan Jasa

9. Tampilan Pengguna/Pelanggan Melihat Proses Pesanan

The screenshot shows a mobile application interface for viewing the order process. It is titled "Detail Transaksi" and contains several sections: a top section with fields for "Nama Laundry :", "No Wa :", and "Alamat : .."; a middle section labeled "Lokasi di Peta"; and a section titled "Status Pesanan" with a sub-label "....." and a box containing fields for "Paket :", "Harga :", "Jumlah :", and "Total : ..".

Gambar III. 32 Tampilan Pengguna/Pelanggan Melihat Proses Pesanan

III.3 Waktu dan Tempat Penelitian

Penelitian akan dilaksanakan dalam waktu 8 bulan terhitung dari bulan Januari 2022 sampai dengan Agustus 2022. Penelitian dilaksanakan di Kota Makassar, Provinsi Sulawesi Selatan.

III.4 Alat dan Bahan

Kebutuhan dalam perancangan sistem terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*), yaitu:

1. Perangkat keras (*hardware*)
 - a. Laptop
 - b. *Smartphone*
 - c. *Mouse*
 - d. *Printer*
2. Perangkat lunak (*software*) yaitu Android Studio

III.5 Metode Pengumpulan Data

Metode pengumpulan data yang akan dilakukan pada penelitian ini adalah sebagai berikut:

1. Studi Literatur

Metode studi literatur adalah metode yang dilakukan dengan cara mempelajari hal-hal yang berkaitan dengan penelitian yang akan dilakukan seperti membaca jurnal atau buku.

2. Studi Dokumentasi

Metode studi dokumentasi adalah metode yang dilakukan dengan cara mempelajari dokumentasi mengenai penelitian berupa dokumentasi pemrograman, *source code* yang digunakan untuk membuat sistem.

3. Observasi

Metode observasi adalah metode yang dilakukan dengan terjun langsung ke lapangan untuk mengamati proses berjalannya sistem secara langsung.

III.6 Metode Pengujian Sistem

Metode pengujian yang dilakukan adalah pengujian dengan menggunakan metode *white box* dan *black box*. Pengujian *white box* berdasarkan pada pengecekan terhadap detail perancangan aplikasi seperti *button* yang memiliki validasi pada sistem sedangkan pengujian dengan metode *black box* dilakukan untuk pengujian fungsi-fungsi yang ada pada sistem seperti menginput, menambahkan data, tampilan aplikasi, dan alur-alur fungsi dari aplikasi.

BAB IV

HASIL DAN PEMBAHASAN

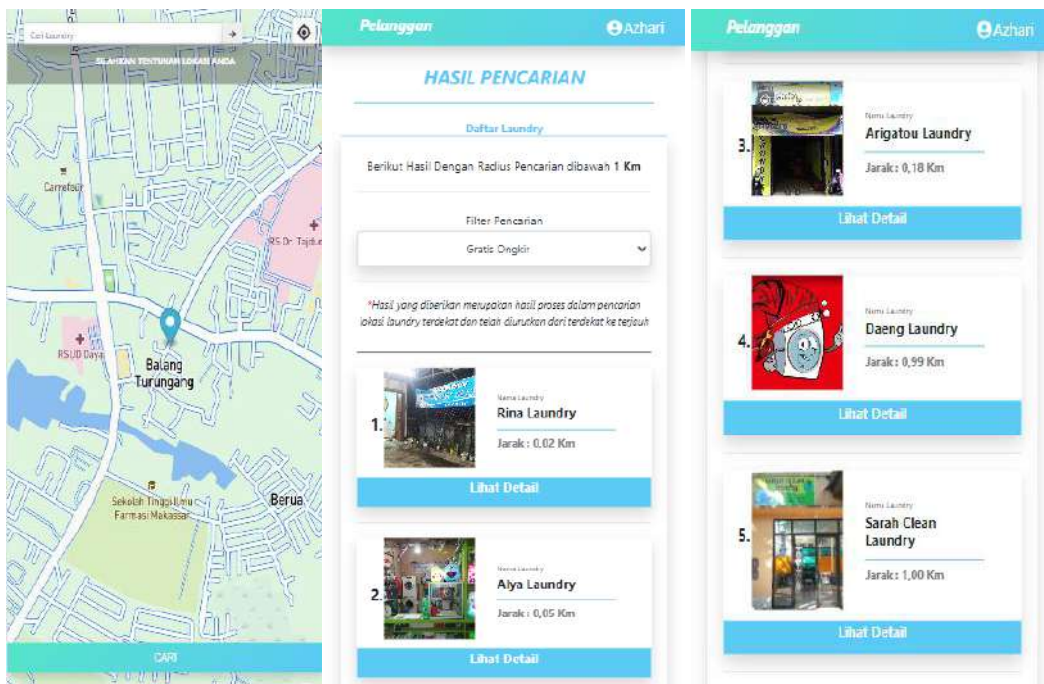
IV.1 Hasil

IV.1.1 Implementasi *Haversine Formula*

Untuk mengetahui keakuratan dari *haversine formula* pada aplikasi, maka dilakukan pengujian dengan membandingkan hasil jarak dari perhitungan aplikasi dan perhitungan manual. Pada penelitian diujikan di Kecamatan Biringkanaya dan Kecamatan Tamalanrea.

IV.1.1.1 Hasil Perhitungan Jarak Terdekat Pada Aplikasi

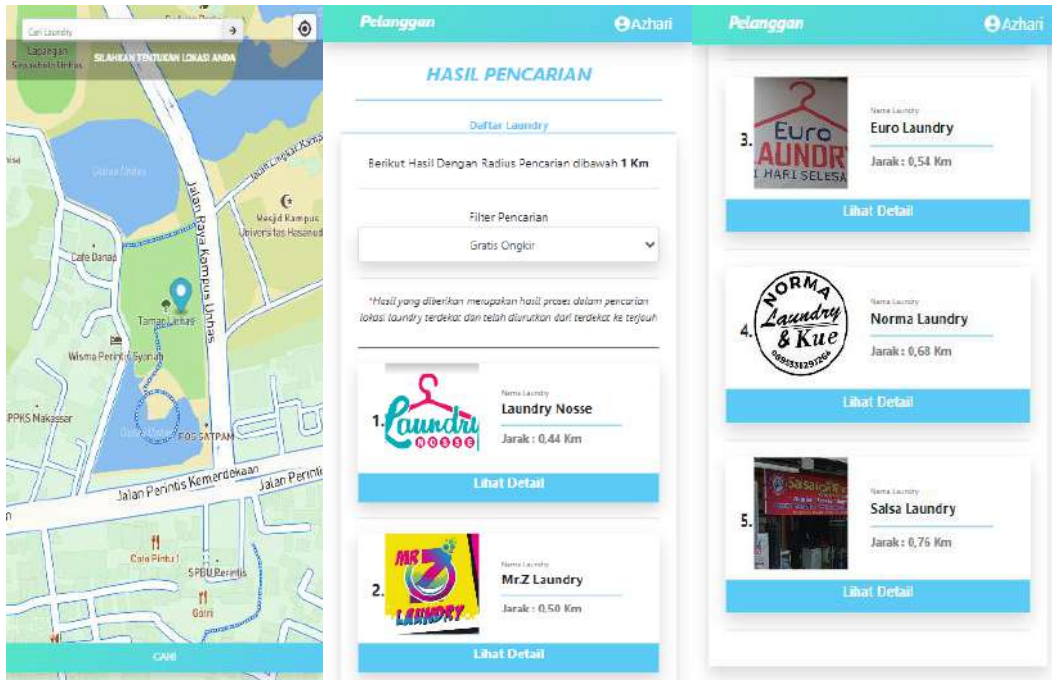
1. Hasil perhitungan jarak terdekat pada aplikasi di Kecamatan Biringkanaya



Gambar IV. 1 Hasil perhitungan Jarak Terdekat pada aplikasi di Kecamatan Biringkanaya

Gambar IV.1 menampilkan hasil pengujian dari perhitungan aplikasi menggunakan *haversine formula* di Kecamatan Biringkanaya dan didapatkan hasil jarak antara kordinat/titik awal dan kordinat/titik akhir.

2. Hasil perhitungan jarak terdekat pada aplikasi di Kecamatan Tamalanrea



Gambar IV. 2 Hasil Perhitungan Jarak Terdekat Pada Aplikasi di Kecamatan Tamalanrea

Gambar IV.2 menampilkan hasil pengujian dari perhitungan aplikasi menggunakan *haversine formula* di Kecamatan Tamalanrea dan di dapatkan hasil jarak antara kordinat/titik awal dan kordinat/titik akhir.

IV.1.1.2 Hasil Perhitungan Jarak Terdekat Secara Manual

1. Hasil perhitungan jarak terdekat secara manual di Kecamatan Biringkanaya

a. Jl. Perintis Kemerdekaan Km.14, Kec. Biringkanaya ke Rina Laundry

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1138857$$

$$\text{Lon1} = 119,513553$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,114048911$$

$$\text{Lon2} = 119,5135442$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1138857 * 0,0174532925 \text{ radian} = -0,089254143 \text{ radian}$$

$$\text{Lon1} = 119,513553 * 0,0174532925 \text{ radian} = 2,085905001 \text{ radian}$$

$$\text{Lat2} = -5,114048911 * 0,0174532925 \text{ radian} = -0,089256992 \text{ radian}$$

$$\text{Lon2} = 119,5135442 * 0,0174532925 \text{ radian} = 2,085904847 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$= (2,085904847 - 2,085905001) * \cos ((-0,089254143 + -0,089256992)/2)$$

$$= -0,000000153887$$

$$y = (\text{lat2} - \text{lat1})$$

$$= -0,089256992 - (-0,089254143)$$

$$= -0,000002849$$

5) Mencari nilai d (jarak)

$$d = \text{sqrt} (x * x + y * y) * r$$

$$= \text{sqrt} (-0,000000153887 * -0,000000153887 + -0,000002849 * -0,000002849) * 6371$$

$$= 0,02 \text{ Km}$$

b. Jl. Perintis Kemerdekaan Km.14, Kec. Biringkanaya ke *Alya Laundry*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1138857$$

$$\text{Lon1} = 119,513553$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,113617185$$

$$\text{Lon2} = 119,513968$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1138857 * 0,0174532925 \text{ radian} = -0,089254143 \text{ radian}$$

$$\text{Lon1} = 119,513553 * 0,0174532925 \text{ radian} = 2,085905001 \text{ radian}$$

$$\text{Lat2} = -5,113617185 * 0,0174532925 \text{ radian} = -0,089249457 \text{ radian}$$

$$\text{Lon2} = 119,513968 * 0,0174532925 \text{ radian} = 2,085912244 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$= (2,085912244 - 2,085905001) * \cos ((-0,089254143 + -0,089249457)/2)$$

$$= -0,00000721417$$

$$\begin{aligned}
y &= (\text{lat2} - \text{lat1}) \\
&= -0,089249457 - (-0,089254143) \\
&= -0,0000004686
\end{aligned}$$

5) Mencari nilai d (jarak)

$$\begin{aligned}
d &= \sqrt{(x * x + y * y)} * r \\
&= \sqrt{(-0,00000721417 * -0,00000721417 + -0,0000004686 * -0,0000004686)} * 6371 \\
&= 0,05 \text{ Km}
\end{aligned}$$

c. Jl. Perintis Kemerdekaan Km.14, Kec. Biringkanaya ke Arigatou *Laundry*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1138857$$

$$\text{Lon1} = 119,513553$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,11366972$$

$$\text{Lon2} = 119,5119151$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1138857 * 0,0174532925 \text{ radian} = -0,089254143 \text{ radian}$$

$$\text{Lon1} = 119,513553 * 0,0174532925 \text{ radian} = 2,085905001 \text{ radian}$$

$$\text{Lat2} = -5,11366972 * 0,0174532925 \text{ radian} = -0,089250373 \text{ radian}$$

$$\text{Lon2} = 119,5119151 * 0,0174532925 \text{ radian} = 2,085876414 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos((\text{lat1} + \text{lat2})/2)$$

$$= (2,085876414 - 2,085905001) * \cos((-0,089254143 + -0,089250373)/2)$$

$$= -0,0000284732$$

$$y = (\text{lat2} - \text{lat1})$$

$$= -0,089250373 - (-0,089254143)$$

$$= -0,00000377$$

5) Mencari nilai d (jarak)

$$d = \sqrt{(x * x + y * y)} * r$$

$$= \sqrt{(-0,0000284732 * -0,0000284732 + -0,00000377 * -0,00000377)} * 6371$$

$$6371$$

$$= 0,18 \text{ Km}$$

d. Jl. Perintis Kemerdekaan Km.14, Kec. Biringkanaya ke Daeng *Laundry*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1138857$$

$$\text{Lon1} = 119,513553$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,118542409$$

$$\text{Lon2} = 119,5211513$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1138857 * 0,0174532925 \text{ radian} = -0,089254143 \text{ radian}$$

$$\text{Lon1} = 119,513553 * 0,0174532925 \text{ radian} = 2,085905001 \text{ radian}$$

$$\text{Lat2} = -5,118542409 * 0,0174532925 \text{ radian} = -0,089335418 \text{ radian}$$

$$\text{Lon2} = 119,5211513 * 0,0174532925 \text{ radian} = 2,086037616 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$= (2,086037616 - 2,085905001) * \cos ((-0,089254143 + -0,089335418)/2)$$

$$= 0,000132087$$

$$y = (\text{lat2} - \text{lat1})$$

$$= -0,089335418 - (-0,089254143)$$

$$= -0,000081275$$

5) Mencari nilai d (jarak)

$$d = \text{sqrt} (x * x + y * y) * r$$

$$= \text{sqrt} (0,000132087 * 0,000132087 + -0,000081275 * -0,000081275) *$$

$$6371$$

$$= 0,99 \text{ Km}$$

e. Jl. Perintis Kemerdekaan Km.14, Kec. Biringkanaya ke Sarah *Clean Laundry*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1138857$$

$$\text{Lon1} = 119,513553$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,105376872$$

$$\text{Lon2} = 119,516379$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1138857 * 0,0174532925 \text{ radian} = -0,089254143 \text{ radian}$$

$$\text{Lon1} = 119,513553 * 0,0174532925 \text{ radian} = 2,085905001 \text{ radian}$$

$$\text{Lat2} = -5,105376872 * 0,0174532925 \text{ radian} = -0,089105636 \text{ radian}$$

$$\text{Lon2} = 119,516379 * 0,0174532925 \text{ radian} = 2,085954324 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$= (2,085954324 - 2,085905001) * \cos ((-0,089254143 + -0,089105636)/2)$$

$$= -0,000049127$$

$$y = (\text{lat2} - \text{lat1})$$

$$= -0,089105636 - (-0,089254143)$$

$$= 0,000148507$$

5) Mencari nilai d (jarak)

$$d = \text{sqrt} (x * x + y * y) * r$$

$$= \text{sqrt} (-0,000049127 * -0,000049127 + 0,000148507 * 0,000148507) * 6371$$

$$6371$$

$$= 0,99 \text{ Km}$$

2. Hasil perhitungan jarak terdekat secara manual di Kecamatan Tamalanrea

a. Jl. Perintis Kemerdekaan Km.10, Kec. Tamalanrea ke *Laundry Nosse*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1393098$$

$$\text{Lon1} = 119,4886062$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,142780189$$

$$\text{Lon2} = 119,4904875$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1393098 * 0,0174532925 \text{ radian} = -0,089697877 \text{ radian}$$

$$\text{Lon1} = 119,4886062 * 0,0174532925 \text{ radian} = 2,085469597 \text{ radian}$$

$$\text{Lat2} = -5,142780189 * 0,0174532925 \text{ radian} = -0,089758447 \text{ radian}$$

$$\text{Lon2} = 119,4904875 * 0,0174532925 \text{ radian} = 2,085502432 \text{ radian}$$

4) Mencari nilai x dan y

$$\begin{aligned}x &= (\text{lon}2 - \text{lon}1) * \cos ((\text{lat}1 + \text{lat}2)/2) \\&= (2,085502432 - 2,085469597) * \cos ((-0,089697877 + -0,089758447)/2) \\&= 0,0000327029 \\y &= (\text{lat}2 - \text{lat}1) \\&= -0,089758447 - (-0,089697877) \\&= -0,00006057\end{aligned}$$

5) Mencari nilai d (jarak)

$$\begin{aligned}d &= \text{sqrt} (x * x + y * y) * r \\&= \text{sqrt} (0,0000327029 * 0,0000327029 + -0,00006057 * -0,00006057) * \\&6371 \\&= 0,44 \text{ Km}\end{aligned}$$

b. Jl. Perintis Kemerdekaan Km.10, Kel. Kec. Tamalanrea ke Mr.Z Laundry

1) Titik Kordinat Awal

$$\begin{aligned}\text{Lat}1 &= -5,1393098 \\ \text{Lon}1 &= 119,4886062\end{aligned}$$

2) Titik Kordinat Akhir

$$\begin{aligned}\text{Lat}2 &= -5,136892718 \\ \text{Lon}2 &= 119,4848089\end{aligned}$$

3) Mengubah derajat ke radian

$$\begin{aligned}\text{Lat}1 &= -5,1393098 * 0,0174532925 \text{ radian} = -0,089697877 \text{ radian} \\ \text{Lon}1 &= 119,4886062 * 0,0174532925 \text{ radian} = 2,085469597 \text{ radian} \\ \text{Lat}2 &= -5,136892718 * 0,0174532925 \text{ radian} = -0,089655691 \text{ radian} \\ \text{Lon}2 &= 119,4848089 * 0,0174532925 \text{ radian} = 2,085403321 \text{ radian}\end{aligned}$$

4) Mencari nilai x dan y

$$\begin{aligned}x &= (\text{lon}2 - \text{lon}1) * \cos ((\text{lat}1 + \text{lat}2)/2) \\&= (2,085403321 - 2,085469597) * \cos ((-0,089697877 + -0,089655691)/2) \\&= -0,0000660097 \\y &= (\text{lat}2 - \text{lat}1) \\&= -0,089655691 - (-0,089697877) \\&= -0,000042186\end{aligned}$$

5) Mencari nilai d (jarak)

$$\begin{aligned}d &= \text{sqrt}(x * x + y * y) * r \\&= \text{sqrt}(-0,000060097 * -0,000060097 + -0,000042186 * -0,000042186) * \\&6371 \\&= 0,50 \text{ Km}\end{aligned}$$

c. Jl. Perintis Kemerdekaan Km.10, Kec. Tamalanrea ke Euro *Laundry*

1) Titik Kordinat Awal

$$\begin{aligned}\text{Lat1} &= -5,1393098 \\ \text{Lon1} &= 119,4886062\end{aligned}$$

2) Titik Kordinat Akhir

$$\begin{aligned}\text{Lat2} &= -5,139953007 \\ \text{Lon2} &= 119,4934289\end{aligned}$$

3) Mengubah derajat ke radian

$$\begin{aligned}\text{Lat1} &= -5,1393098 * 0,0174532925 \text{ radian} = -0,089697877 \text{ radian} \\ \text{Lon1} &= 119,4886062 * 0,0174532925 \text{ radian} = 2,085469597 \text{ radian} \\ \text{Lat2} &= -5,139953007 * 0,0174532925 \text{ radian} = -0,089709103 \text{ radian} \\ \text{Lon2} &= 119,4934289 * 0,0174532925 \text{ radian} = 2,085553769 \text{ radian}\end{aligned}$$

4) Mencari nilai x dan y

$$\begin{aligned}x &= (\text{lon2} - \text{lon1}) * \cos((\text{lat1} + \text{lat2})/2) \\&= (2,085553769 - 2,085469597) * \cos((-0,089697877 + -0,089709103)/2) \\&= -0,0000838336 \\ y &= (\text{lat2} - \text{lat1}) \\&= -0,089709103 - (-0,089697877) \\&= -0,000011226\end{aligned}$$

5) Mencari nilai d (jarak)

$$\begin{aligned}d &= \text{sqrt}(x * x + y * y) * r \\&= \text{sqrt}(-0,0000838336 * -0,0000838336 + -0,000011226 * -0,000011226) \\&* 6371 \\&= 0,54 \text{ Km}\end{aligned}$$

d. Jl. Perintis Kemerdekaan Km.10, Kec. Tamalanrea ke Norma *Laundry*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1393098$$

$$\text{Lon1} = 119,4886062$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,134711168$$

$$\text{Lon2} = 119,4845702$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1393098 * 0,0174532925 \text{ radian} = -0,089697877 \text{ radian}$$

$$\text{Lon1} = 119,4886062 * 0,0174532925 \text{ radian} = 2,085469597 \text{ radian}$$

$$\text{Lat2} = -5,134711168 * 0,0174532925 \text{ radian} = -0,089617616 \text{ radian}$$

$$\text{Lon2} = 119,4845702 * 0,0174532925 \text{ radian} = 2,085399155 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$= (2,085399155 - 2,085469597) * \cos ((-0,089697877 + -0,089617616)/2)$$

$$= -0,0000701591$$

$$y = (\text{lat2} - \text{lat1})$$

$$= -0,089617616 - (-0,089697877)$$

$$= -0,000080261$$

5) Mencari nilai d (jarak)

$$d = \text{sqrt} (x * x + y * y) * r$$

$$= \text{sqrt} (-0,0000701591 * -0,0000701591 + -0,000080261 * -0,000080261)$$

$$* 6371$$

$$= 0,68 \text{ Km}$$

e. Jl. Perintis Kemerdekaan Km.10, Kec. Tamalanrea ke Salsa *Laundry*

1) Titik Kordinat Awal

$$\text{Lat1} = -5,1393098$$

$$\text{Lon1} = 119,4886062$$

2) Titik Kordinat Akhir

$$\text{Lat2} = -5,146126754$$

$$\text{Lon2} = 119,4890329$$

3) Mengubah derajat ke radian

$$\text{Lat1} = -5,1393098 * 0,0174532925 \text{ radian} = -0,089697877 \text{ radian}$$

$$\text{Lon1} = 119,4886062 * 0,0174532925 \text{ radian} = 2,085469597 \text{ radian}$$

$$\text{Lat2} = -5,146126754 * 0,0174532925 \text{ radian} = -0,089816856 \text{ radian}$$

$$\text{Lon2} = 119,4890329 * 0,0174532925 \text{ radian} = 2,085477044 \text{ radian}$$

4) Mencari nilai x dan y

$$x = (\text{lon2} - \text{lon1}) * \cos ((\text{lat1} + \text{lat2})/2)$$

$$= (2,085477044 - 2,085469597) * \cos ((-0,089697877 + -0,089816856)/2)$$

$$= -0,00000741702$$

$$y = (\text{lat2} - \text{lat1})$$

$$= -0,089816856 - (-0,089697877)$$

$$= -0,000118979$$

5) Mencari nilai d (jarak)

$$d = \text{sqrt} (x * x + y * y) * r$$

$$= \text{sqrt} (-0,00000741702 * -0,00000741702 + -0,000118979 * -0,000118979) * 6371$$

$$= 0,76 \text{ Km}$$

IV.1.1.3 Hasil Perhitungan Akurasi *Haversine Formula*

Tabel IV. 1 Perhitungan Jarak

No	Titik/Kordinat Pengguna	Titik/Kordinat <i>Laundry</i>	Perhitungan Jarak Pada Aplikasi	Perhitungan Jarak Secara Manual	Keterangan
1	Jl. Perintis Kemerdekaan Km. 14, Kec. Biringkanaya	Rina <i>Laundry</i> Latitude = -5,114048911 Longitude = 119,5135442	0,02 Km	0,02 Km	Sesuai/Benar
2	-5,1138857,	Alya <i>Laundry</i>	0,05 Km	0,05 Km	Sesuai/Benar

No	Titik/Kordinat Pengguna	Titik/Kordinat Laundry	Perhitungan Jarak Pada Aplikasi	Perhitungan Jarak Secara Manual	Keterangan
	Longitude = 119,513553	Latitude = -5,113617185 Longitude = 119,513968			
3		Arigatou Laundry Latitude = -5,11366972 Longitude = 119,5119151	0,18 Km	0,18 Km	Sesuai/Benar
4		Daeng Laundry Latitude = -5,118542409 Longitude = 119,5211513	0,99 Km	0,99 Km	Sesuai/Benar
5		Sarah Clean Laundry Latitude = -5,105376872 Longitude = 119,516379	1,00 Km	0,99 Km	Tidak Sesuai/Salah
6	Jl. Perintis Kemerdekaan Km. 10, Kec Tamalanrea	Laundry Nosse Latitude = -5,142780189 Longitude =	0,44 Km	0,44 Km	Sesuai/Benar

No	Titik/Kordinat Pengguna	Titik/Kordinat Laundry	Perhitungan Jarak Pada Aplikasi	Perhitungan Jarak Secara Manual	Keterangan
	Latitude =	119,4904875			
7	-5,1393098, Longitude = 119,4886062	Mr. Z Laundry Latitude = -5,136982718 Longitude = 119,4848089	0,50 Km	0,50 Km	Sesuai/Benar
8		Euro Laundry Latitude = -5,139953007 Longitude = 119,4934289	0,54 Km	0,54 Km	Sesuai/Benar
9		Norma Laundry Latitude = -5,134711168 Longitude = 119,4845702	0,68 Km	0,68 Km	Sesuai/Benar
10		Salsa Laundry Latitude = -5,146126754 Longitude = 119,4890329	0,76 Km	0,76 Km	Sesuai/Benar
Jumlah data benar					9

Setelah itu dapat dicari akurasi dengan menggunakan rumus:

$$Akurasi = \frac{Jumlah\ data\ benar}{Jumlah\ data} \times 100\%$$

Jumlah data = 10

Jumlah data benar = 9

$$Akurasi = \frac{9}{10} \times 100\%$$

Akurasi = 90%

Dari perhitungan akurasi didapatkan hasil 90% dari hasil pengujian perhitungan pada aplikasi dengan perhitungan manual, maka dapat disimpulkan penerapan *haversine formula* dalam menghitung jarak antara dua titik berhasil dilakukan dan didapatkan hasil yang akurat dalam pencarian *laundry* berdasarkan lokasi.

IV.1.2 Implementasi *Interface*

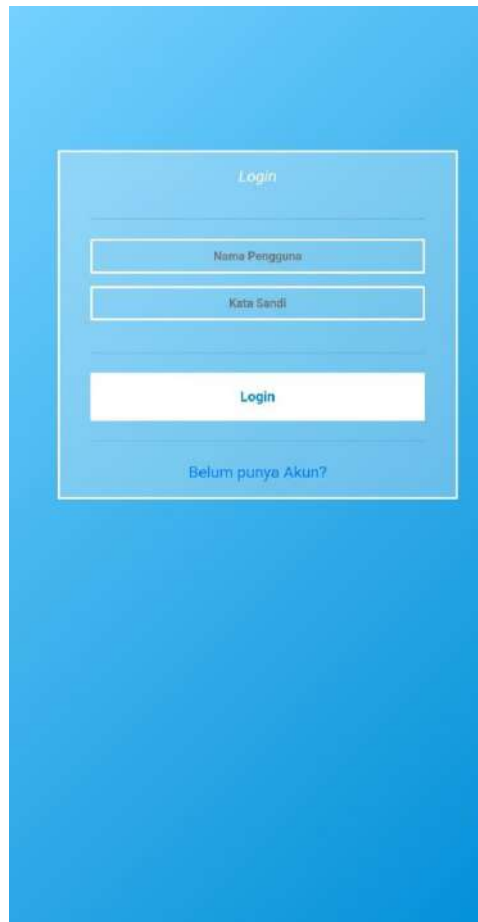
1. Hasil Tampilan Aplikasi



Gambar IV. 3 Hasil Tampilan Aplikasi

Pada gambar IV.3 menunjukkan tampilan awal pada saat mengakses aplikasi. Pada tampilan awal ini terdapat dua *button*, yaitu *button login* untuk pengguna yang telah memiliki akun dan *button* daftar untuk pengguna yang belum memiliki akun.

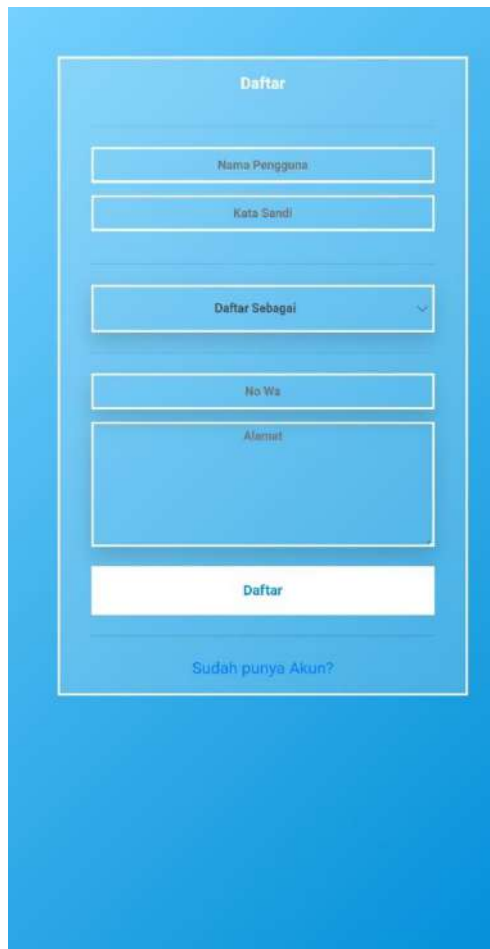
2. Hasil Tampilan *Login*

The image shows a login form centered on a blue gradient background. The form is enclosed in a light blue border and has the title "Login" at the top. It features two input fields: "Nama Pengguna" (User Name) and "Kata Sandi" (Password). Below these fields is a white "Login" button. At the bottom of the form, there is a link that says "Belum punya Akun?" (Don't have an account?).

Gambar IV. 4 Hasil Tampilan *Login*

Pada gambar IV.4 menunjukkan tampilan *login* untuk pengguna melakukan *login* agar dapat mengakses aplikasi. Pada tampilan *login* terdapat *form* nama pengguna, kata sandi, dan *button login* untuk melakukan *login*.

3. Hasil Tampilan Registrasi

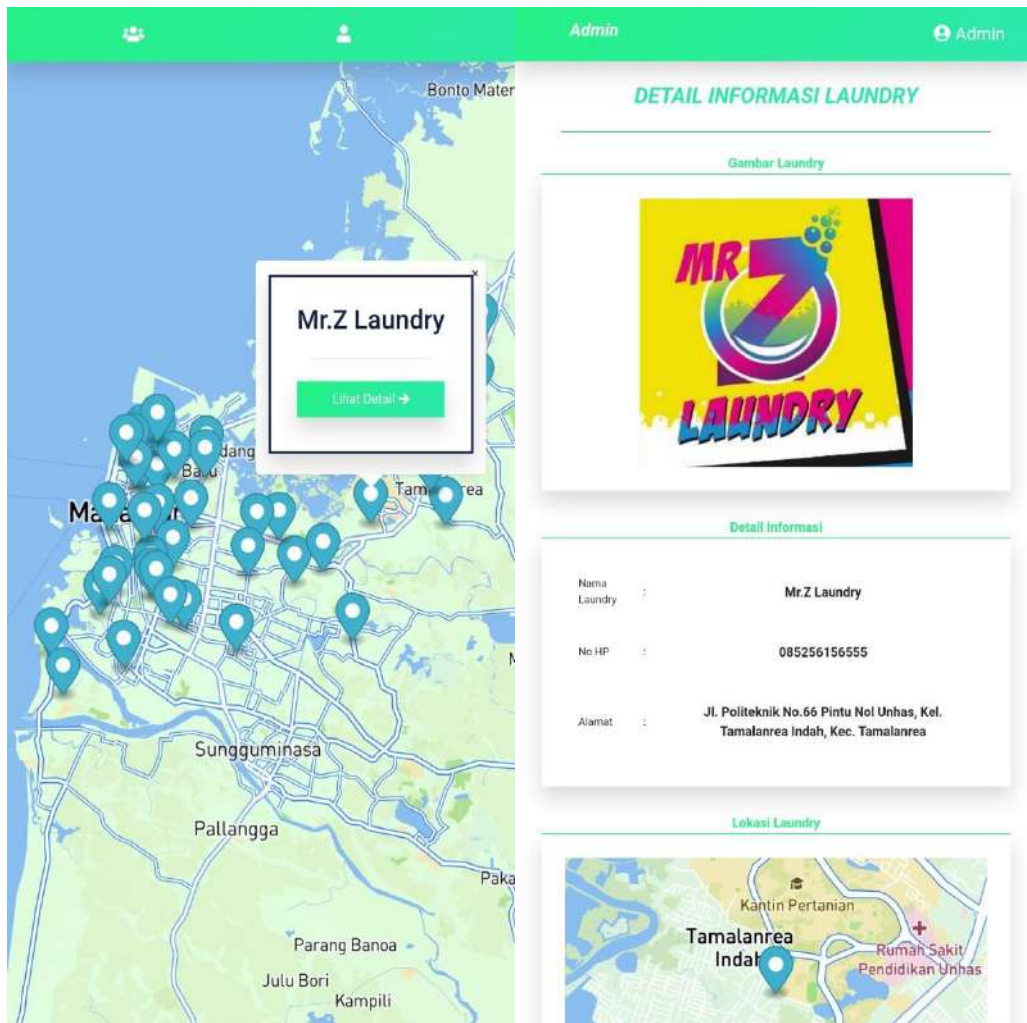


The image shows a registration form titled "Daftar" on a blue background. The form is contained within a white-bordered box. It features the following elements from top to bottom: a text input field for "Nama Pengguna", a text input field for "Kata Sandi", a dropdown menu labeled "Daftar Sebagai" with a downward arrow, a text input field for "No Wa", a larger text input field for "Alamat", a white button with the text "Daftar", and a blue link at the bottom that says "Sudah punya Akun?".

Gambar IV. 5 Hasil Tampilan Registrasi

Pada gambar IV.5 menunjukkan tampilan registrasi untuk pengguna yang belum mempunyai akun. Pada tampilan daftar terdapat *form* nama pengguna, kata sandi, pilihan daftar sebagai pengguna/pelanggan atau pemilik *laundry*, nomor WhatsApp, alamat, dan *button* daftar untuk melakukan pendaftaran.

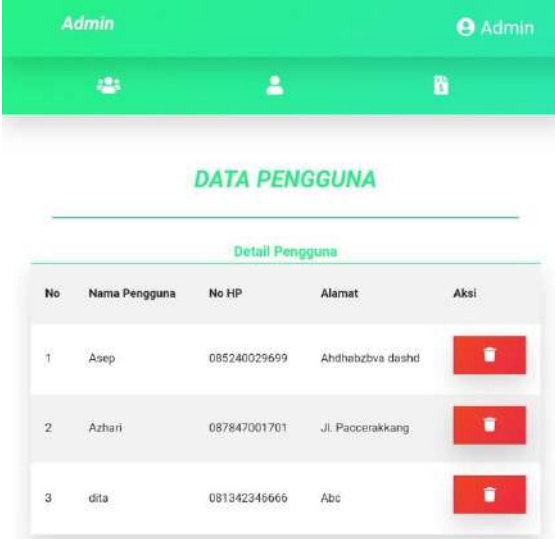
4. Hasil Tampilan Admin Mengelola Laundry






Gambar IV. 6 Hasil Tampilan Admin Mengelola Laundry

Pada gambar IV.6 menunjukkan tampilan data laundry yang dapat dikelola oleh admin dan untuk mengetahui lokasi-lokasi setiap laundry yang terdapat di maps. Pada tampilan data laundry terdapat maps dan mark lokasi laundry yang ketika mark ditekan maka akan muncul nama laundry, button lihat detail yang berisi informasi dari laundry dan button hapus laundry untuk penghapusan data laundry.

5. Hasil Tampilan Admin Mengelola Pengguna/Pelanggan

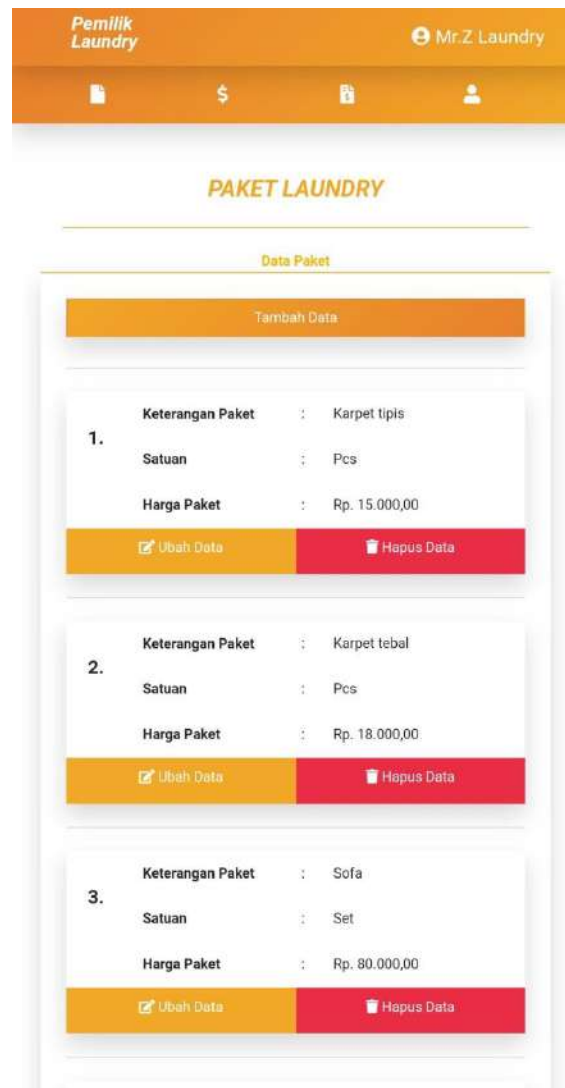


No	Nama Pengguna	No HP	Alamat	Aksi
1	Asep	085240029699	Ahdhabzbva dashd	
2	Azhari	087847001701	Jl. Pacerakkang	
3	dita	081342346666	Abc	

Gambar IV. 7 Hasil Tampilan Admin Mengelola Pengguna/Pelanggan

Pada gambar IV.7 menunjukkan tampilan data pelanggan yang dapat dikelola oleh admin. Pada tampilan data pelanggan terdapat detail pengguna seperti nama, nomor hp/WhatsApp, alamat dan *button* hapus yang digunakan untuk menghapus data pelanggan.

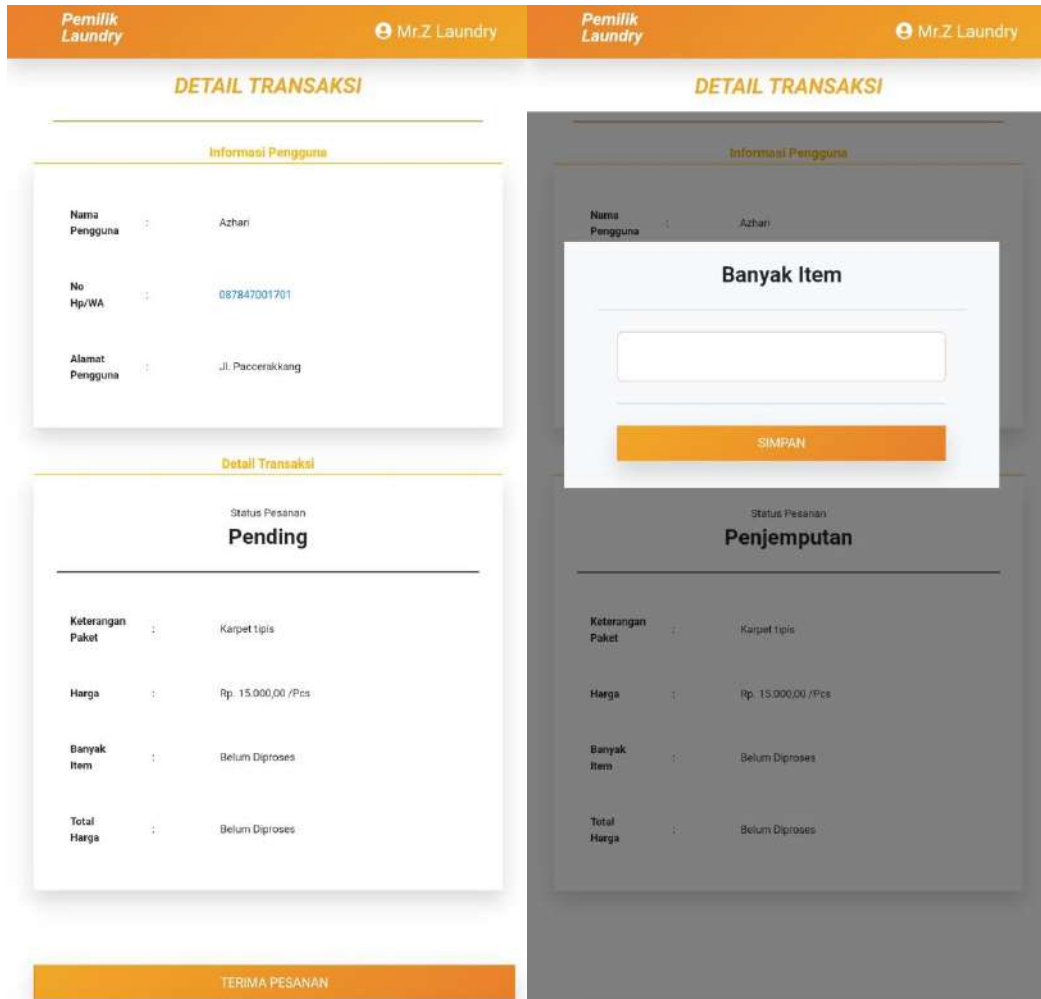
6. Hasil Tampilan Pemilik *Laundry* Mengelola Penjualan



Gambar IV. 8 Hasil Tampilan Pemilik *Laundry* Mengelola Penjualan

Pada gambar IV.8 menunjukkan tampilan data penjualan yang dapat dikelola oleh *laundry* yang nantinya akan ditampilkan untuk pelanggan di detail *laundry*. Pada tampilan data penjualan terdapat *button* tambah untuk menambah data paket, *button* ubah data untuk mengubah data jika terjadi kesalahan penginputan dan *button* hapus data yang digunakan untuk menghapus data.

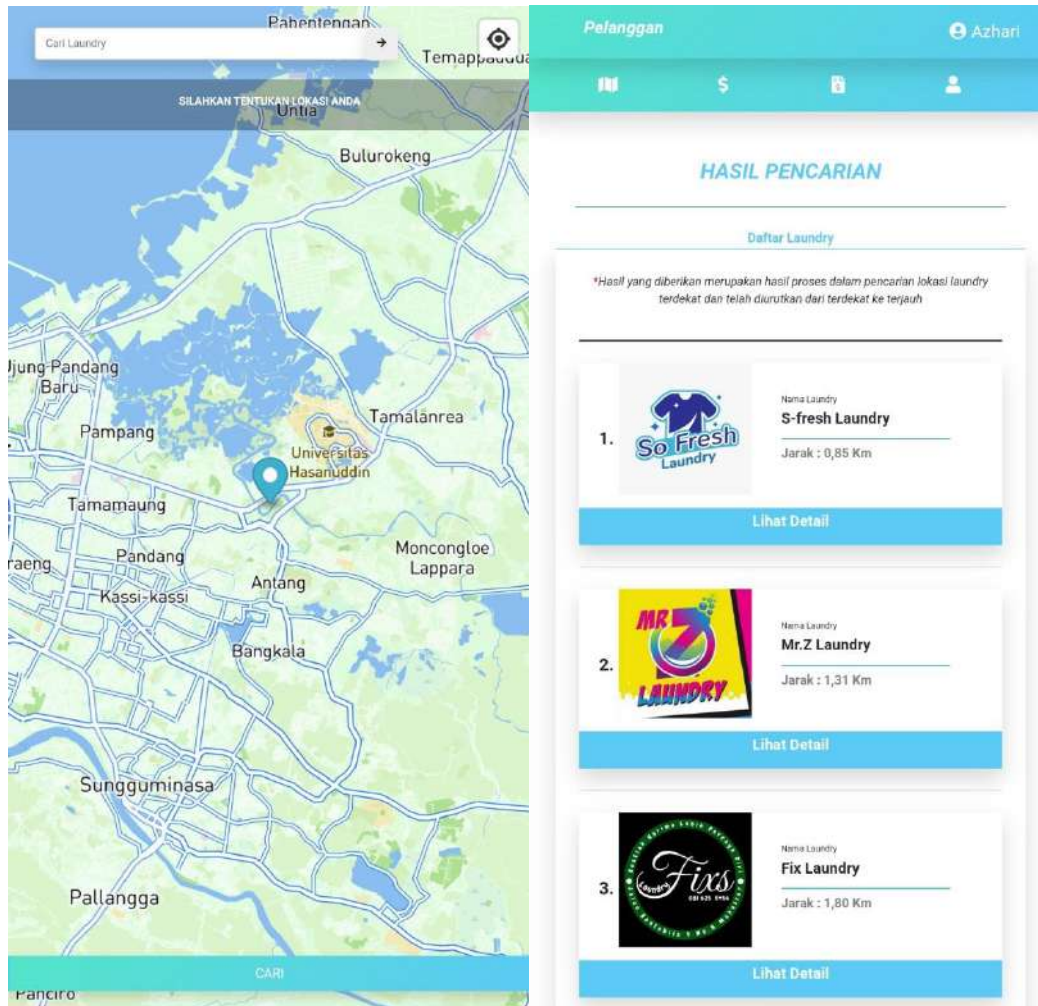
7. Hasil Tampilan Pemilik *Laundry* Memproses Pesanan



Gambar IV. 9 Hasil Tampilan Pemilik *Laundry* Memproses Pesanan

Pada gambar IV.9 menunjukkan tampilan proses pesanan yang dilakukan oleh *laundry*. Pada tampilan proses pesanan terdapat informasi dari pemesan, detail pesanan dari pemesan, dan *button* untuk melakukan aksi seperti terima pesanan dan batalkan pesanan. Pada *button proses* akan muncul *form* penginputan jumlah berdasarkan satuan paket lalu dikalikan dengan harga paket untuk menampilkan total harga dari pesanan yang dapat dilihat oleh pelanggan.

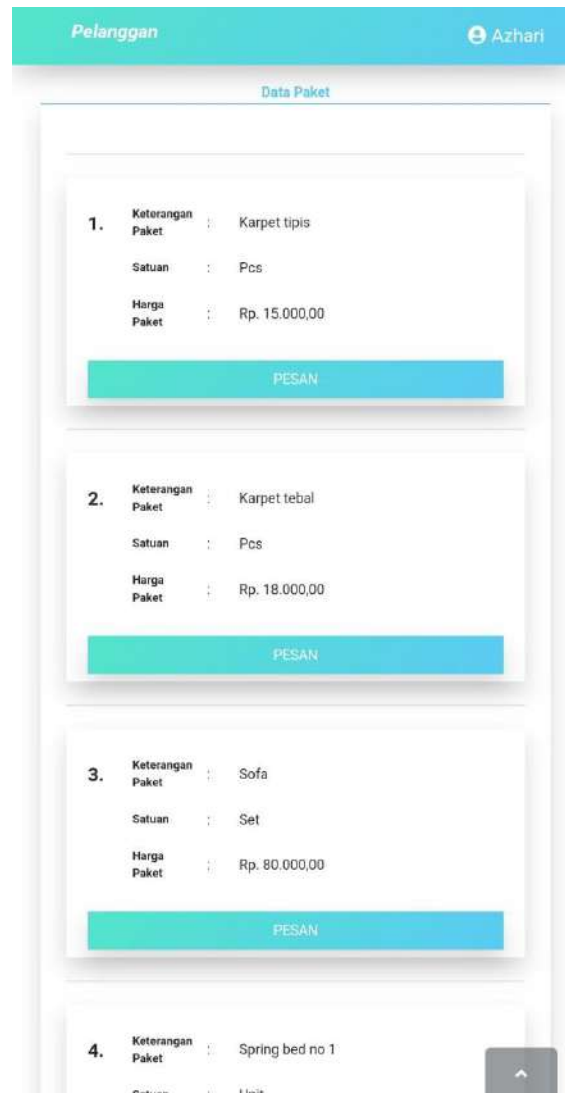
8. Hasil Tampilan Pengguna/Pelanggan Mencari *Laundry*



Gambar IV. 10 Hasil Tampilan Pengguna/Pelanggan Mencari *Laundry*

Pada gambar IV.10 menunjukkan tampilan cari *laundry* terdekat berdasarkan lokasi dari pelanggan. Pada tampilan cari *laundry* terdapat pencarian berdasarkan nama dan lokasi sehingga ketika menekan *button* pencarian maka akan tampil hasil pencarian *laundry*.

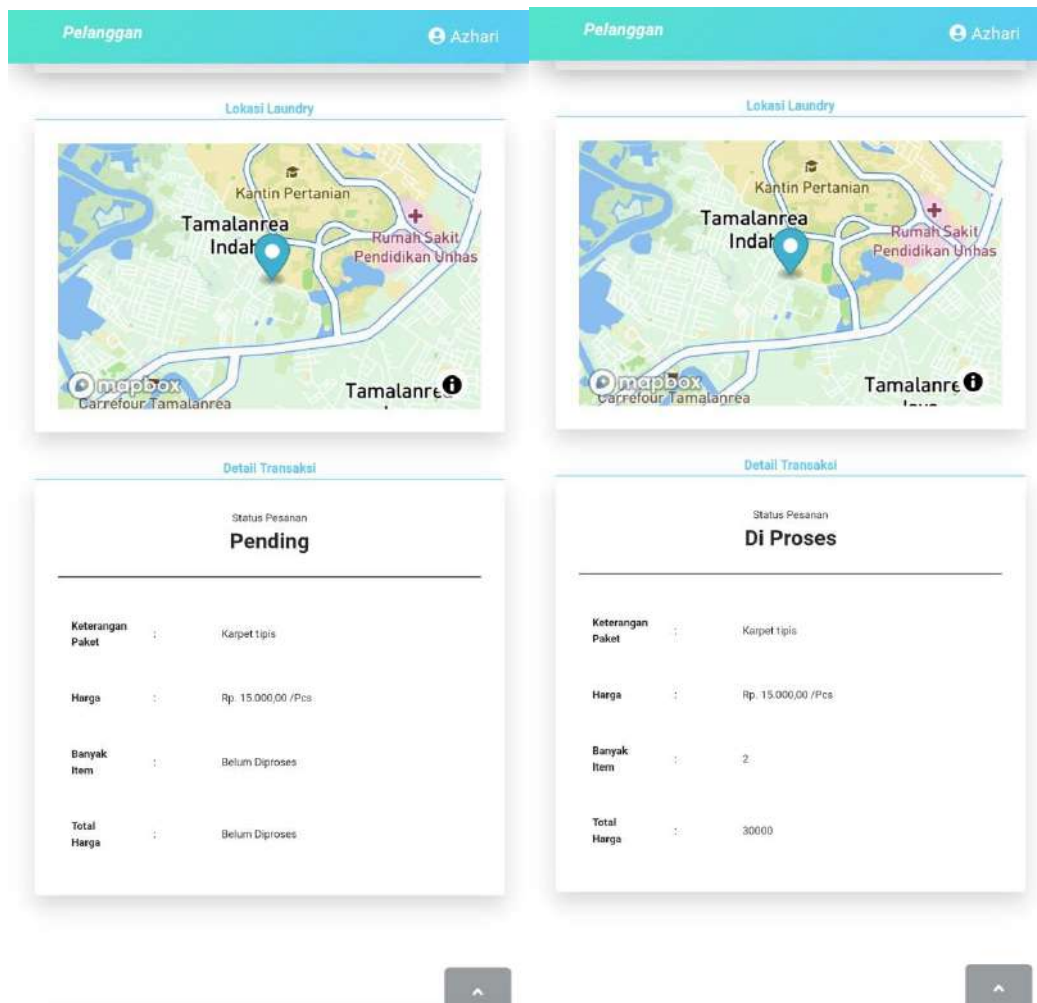
9. Hasil Tampilan Pengguna/Pelanggan Memesan Jasa



Gambar IV. 11 Hasil Tampilan Pengguna/Pelanggan Memesan Jasa

Pada gambar IV.11 menunjukkan tampilan pemesanan jasa yang dilakukan oleh pengguna/pelanggan yang terdapat pada detail *laundry*. Pada tampilan pesan terdapat paket-paket *laundry* yang berisi keterangan, satuan, harga, dan *button* pesan untuk melakukan pemesanan berdasarkan pilihan pelanggan.

10. Hasil Tampilan Pengguna/Pelanggan Melihat Proses Pesanan



Gambar IV. 12 Hasil Tampilan Pengguna/Pelanggan Melihat Proses Pesanan

Pada gambar IV.12 menunjukkan tampilan lihat proses pesanan yang dilakukan oleh pelanggan setelah melakukan pemesanan. Pada tampilan lihat proses pesanan terdapat informasi dari *laundry*, lokasi, status pesanan, detail pesanan, dan *button* batalkan pesanan untuk membatalkan pesanan pada awal pemesanan. Setelah pihak *laundry* mengkonfirmasi pesanan status dari pesanan akan berubah sesuai dari proses pihak *laundry*. Total harga akan muncul setelah pihak *laundry* mengisi *form* jumlah dari satuan paket dan melakukan proses pesanan.

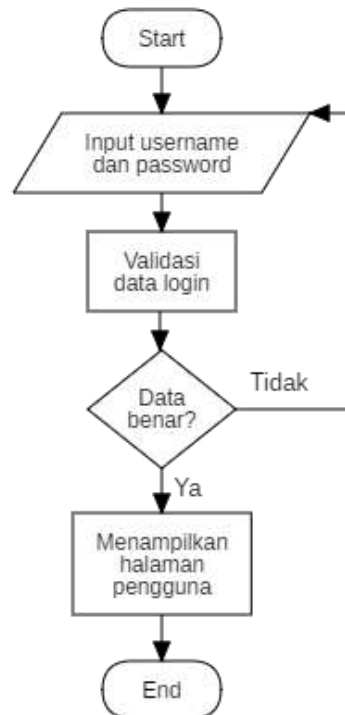
IV.2 Pembahasan

IV.2.1 Pengujian *White Box*

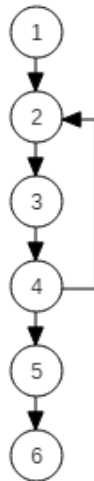
Pengujian *white box* ini menguji dengan cara melihat *pure code* dari suatu aplikasi/*software* yang diuji tanpa memperdulikan tampilan atau UI dari aplikasi tersebut. Tahapan-tahapan pengujian dimulai dari pembuatan *flowchart*, pembuatan *flow graph*, perhitungan kompleksitas siklomatis, perhitungan jalur independen, dan *test case*.

1. Menu *Login*

Menu *login* berfungsi untuk pengguna melakukan *login* sebelum mengakses aplikasi.



Gambar IV. 13 *Flowchart Login*



Gambar IV. 14 *Flow graph Login*

Flow graph login pada gambar IV.14 terdapat 6 *edge* dan 6 *node*. *Flow graph login* standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 6 \text{ edge} - 6 \text{ node} + 2 = 2$
- c. $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.14 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-4-5-6
- b. Jalur 2: 1-2-3-4-2-3-4-5-6

Test case login dijelaskan pada tabel IV.2.

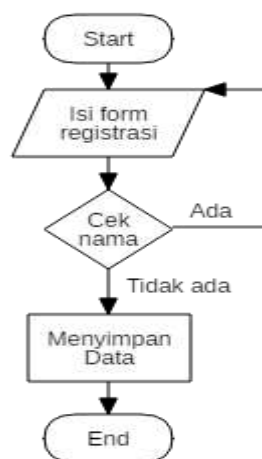
Tabel IV. 2 *Test Case Login*

Path	1
Jalur	1-2-3-4-5-6
Skenario	<i>Start</i> Masukkan <i>username</i> dan <i>password</i> Validasi data <i>login</i> Data benar Sistem menampilkan halaman pengguna <i>End</i>

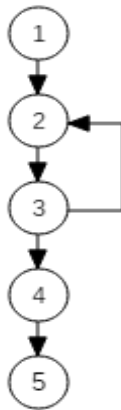
Hasil Pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<i>Start</i> Masukkan <i>username</i> dan <i>password</i> Validasi data <i>login</i> Data salah Masukkan kembali <i>username</i> dan <i>password</i> Valdiasasi data <i>login</i> Data Benar Sistem menampilkan halaman pengguna <i>End</i>
Hasil Pengujian	Berhasil

2. Menu Registrasi

Menu registrasi berfungsi untuk pengguna baru yang belum mempunyai akun sehingga dapat melakukan *login*.



Gambar IV. 15 *Flowchart* Registrasi



Gambar IV. 16 *Flow graph* Registrasi

Flow graph registrasi pada gambar IV.16 terdapat 5 *edge* dan 5 *node*. *Flow graph* registrasi standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 5 \text{ edge} - 5 \text{ node} + 2 = 2$
- c. $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.16 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-4-5
- b. Jalur 1: 1-2-3-2-3-4-5

Test case registrasi dijelaskan pada tabel IV.3.

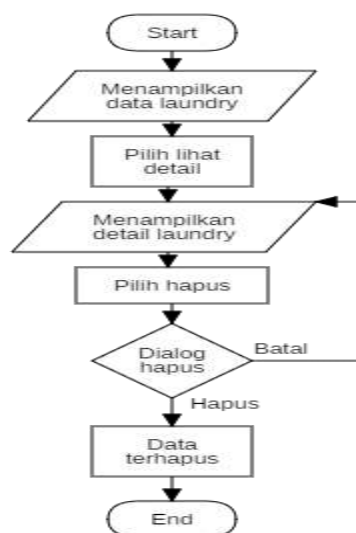
Tabel IV. 3 *Test Case* Registrasi

Path	1
Jalur	1-2-3-4-5
Skenario	<i>Start</i> Isi <i>form</i> Registrasi Cek nama (tidak ada) Menyimpan data <i>End</i>
Hasil pengujian	Berhasil
Path	2

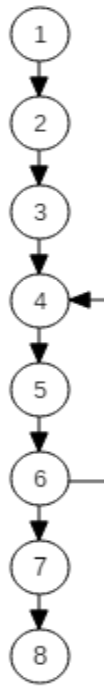
Jalur	1-2-3-2-3-4-5
Skenario	<i>Start</i> Isi <i>form</i> Registrasi Cek nama (ada) Isi <i>form</i> Registrasi Cek nama (tidak ada) Menyimpan data <i>End</i>
Hasil pengujian	Berhasil

3. Menu Kelola *Laundry*

Menu kelola *laundry* berfungsi untuk melihat atau menghapus data *laundry* yang telah terdaftar pada aplikasi.



Gambar IV. 17 Flowchart Kelola *Laundry*



Gambar IV. 18 *Flow graph* Kelola Laundry

Flow graph kelola laundry pada gambar IV.18 terdapat 8 *edge* dan 8 *node*. *Flow graph* kelola laundry standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 8 \text{ edge} - 8 \text{ node} + 2 = 2$
- c. $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.18 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-4-5-6-7-8
- b. Jalur 2: 1-2-3-4-5-6-4-5-6-7-8

Test case kelola laundry dijelaskan pada tabel IV.4.

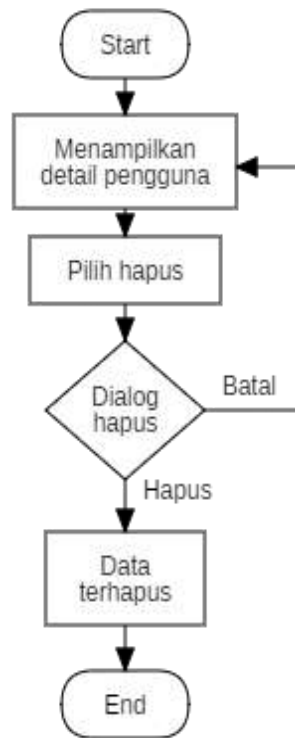
Tabel IV. 4 *Test Case* Kelola Laundry

Path	1
Jalur	1-2-3-4-5-6-7-8
Skenario	<i>Start</i>

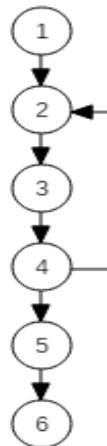
	Menampilkan data <i>laundry</i> Pilih lihat detail Menampilkan detail <i>laundry</i> Pilih hapus Dialog hapus (hapus) Data terhapus <i>End</i>
Hasil pengujian	Berhasil
Path	2
Jalur	1-2-3-4-5-6-4-5-6-7-8
Skenario	<i>Start</i> Menampilkan data <i>laundry</i> Pilih lihat detail Menampilkan detail <i>laundry</i> Pilih hapus Dialog hapus (batal) Menampilkan detail <i>laundry</i> Pilih hapus Dialog hapus (hapus) Data terhapus <i>End</i>
Hasil Pengujian	Berhasil

4. Menu Kelola Pengguna

Menu kelola pengguna berfungsi untuk melihat atau menghapus data pengguna yang telah terdaftar pada aplikasi.



Gambar IV. 19 *Flowchart* Kelola Pengguna



Gambar IV. 20 *Flow graph* Kelola Pengguna

Flow graph kelola pengguna pada gambar IV.20 terdapat 6 *edge* dan 6 *node*. *Flow graph* kelola pengguna standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 6 \text{ edge} - 6 \text{ node} + 2 = 2$

c. $V(G) = 1$ simpul yang diperkirakan $+ 1 = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.20 berjumlah 2. Dengan jalur independennya, yaitu:

1. Jalur 1: 1-2-3-4-5-6
2. Jalur 2: 1-2-3-4-2-3-4-5-6

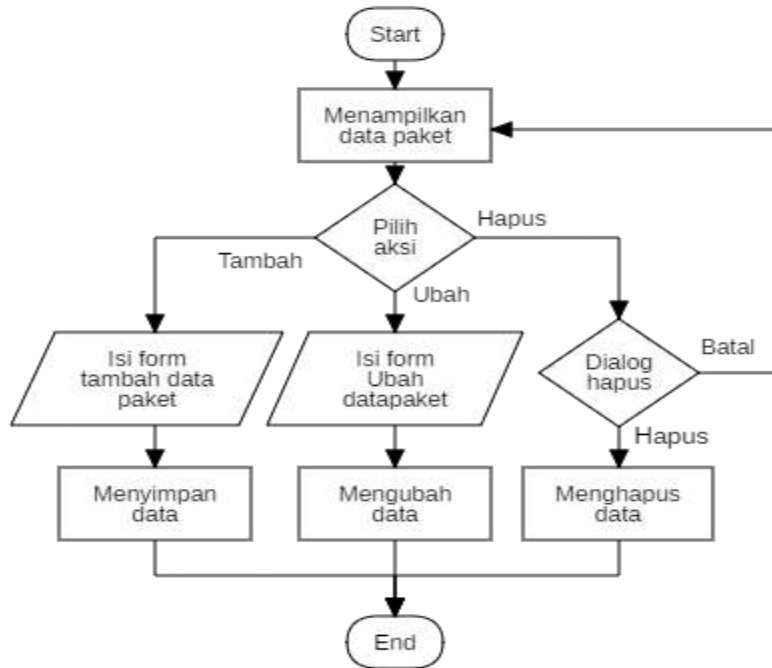
Test case kelola pengguna dijelaskan pada tabel IV.5.

Tabel IV. 5 *Test Case* Kelola Pengguna

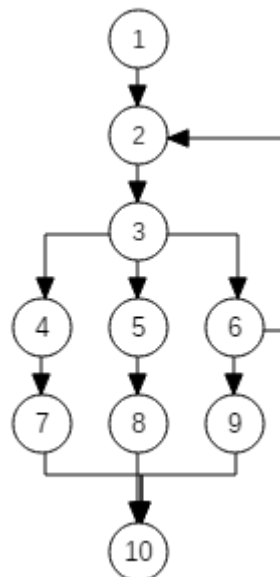
Path	1
Jalur	1-2-3-4-5-6
Skenario	<i>Start</i> Menampilkan detail pengguna Pilih hapus Dialog hapus (hapus) Data terhapus <i>End</i>
Hasil pengujian	Berhasil
Path	2
Jalur	1-2-3-4-2-3-4-5-6
Skenario	<i>Start</i> Menampilkan detail pengguna Pilih hapus Dialog hapus (batal) Menampilkan detail pengguna Pilih hapus Dialog hapus (hapus) Data terhapus <i>End</i>
Hasil pengujian	Berhasil

5. Menu Kelola Penjualan

Menu kelola penjualan berfungsi untuk menambahkan, mengubah, atau menghapus data paket *laundry*.



Gambar IV. 21 *Flowchart* Kelola Penjualan



Gambar IV. 22 *Flow graph* Kelola Penjualan

Flow graph kelola penjualan pada gambar IV.22 terdapat 12 *edge* dan 10 *node*. *Flow graph* kelola penjualan standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 4 region
- b. $V(G) = 12 \text{ edge} - 10 \text{ node} + 2 = 4$
- c. $V(G) = 3 \text{ simpul yang diperkirakan} + 1 = 4$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.22 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-4-7-10
- b. Jalur 2: 1-2-3-5-8-10
- c. Jalur 3: 1-2-3-6-9-10
- d. Jalur 4: 1-2-3-6-2-3-6-9-10

Test case kelola penjualan dijelaskan pada tabel IV.6.

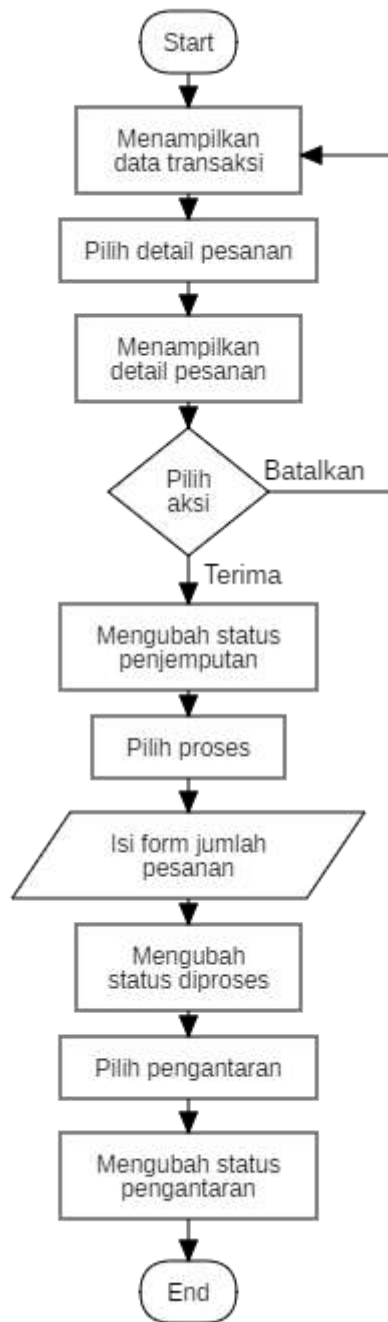
Tabel IV. 6 *Test Case* Kelola Penjualan

Path	1
Jalur	1-2-3-4-7-10
Skenario	<i>Start</i> Menampilkan data paket Pilih aksi (tambah) Isi data paket Menyimpan data <i>End</i>
Hasil pengujian	Berhasil
Path	2
Jalur	1-2-3-5-8-10
Skenario	<i>Start</i> Menampilkan data paket Pilih aksi (ubah) Isi data paket Ubah data paket

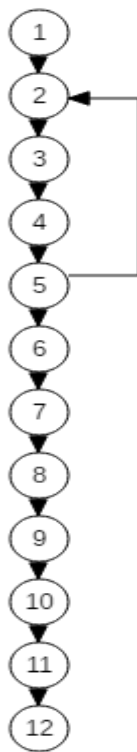
	Mengubah data <i>End</i>
Hasil pengujian	Berhasil
Path	3
Jalur	1-2-3-6-9-10
Skenario	<i>Start</i> Menampilkan data paket Pilih aksi (hapus) Dialog hapus (hapus) Menghapus data <i>End</i>
Hasil pengujian	Berhasil
Path	4
Jalur	1-2-3-6-2-3-6-9-10
Skenario	<i>Start</i> Menampilkan data paket Pilih aksi (hapus) Dialog hapus (batal) Menampilkan data paket Pilih aksi (hapus) Dialog hapus (hapus) Menghapus data <i>End</i>
Hasil pengujian	Berhasil

6. Menu Proses Pesanan

Menu proses pesanan berfungsi untuk memproses pesanan *laundry* yang dilakukan oleh pelanggan/pengguna.



Gambar IV. 23 *Flowchart* Proses Pesanan



Gambar IV. 24 *Flow graph* Proses Pesanan

Flow graph proses pesanan pada gambar IV.24 terdapat 12 *edge* dan 12 *node*. *Flow graph* proses pesanan standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 12 \text{ edge} - 12 \text{ node} + 2 = 2$
- c. $V(G) = 1 \text{ simpul yang diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.24 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-4-5-6-7-8-9-10-11-12
- b. Jalur 2: 1-2-3-4-5-2-3-4-5-6-7-8-9-10-11-12

Test case proses pesanan dijelaskan pada tabel IV.7.

Tabel IV. 7 *Test Case* Proses Pesanan

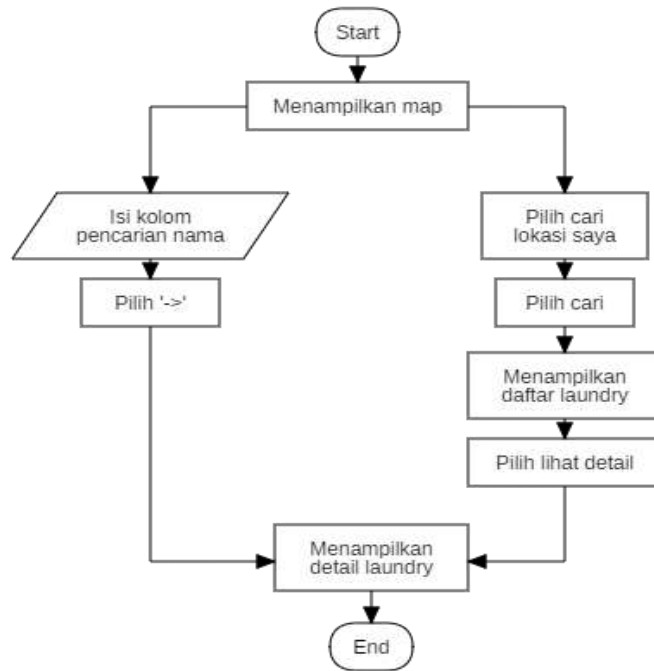
Path	1
Jalur	1-2-3-4-5-6-7-8-9-10-11-12

Skenario	<p><i>Start</i></p> <p>Menampilkan data transaksi</p> <p>Pilih detail pesanan</p> <p>Menampilkan detail pesanan</p> <p>Pilih aksi (terima)</p> <p>Mengubah status penjemputan</p> <p>Pilih proses</p> <p>Isi <i>form</i> jumlah pesanan</p> <p>Mengubah status diproses</p> <p>Pilih pengantaran</p> <p>Mengubah status pengantaran</p> <p><i>End</i></p>
Hasil pengujian	Berhasil
Path	2
Jalur	1-2-3-4-5-2-3-4-5-6-7-8-9-10-11-12
Skenario	<p><i>Start</i></p> <p>Menampilkan data transaksi</p> <p>Pilih detail pesanan</p> <p>Menampilkan detail pesanan</p> <p>Pilih aksi (batalkan)</p> <p>Menampilkan data transaksi</p> <p>Pilih detail pesanan</p> <p>Menampilkan detail pesanan</p> <p>Pilih aksi (terima)</p> <p>Mengubah status penjemputan</p> <p>Pilih proses</p> <p>Isi <i>form</i> jumlah pesanan</p> <p>Mengubah status diproses</p> <p>Pilih pengantaran</p> <p>Mengubah status pengantaran</p>

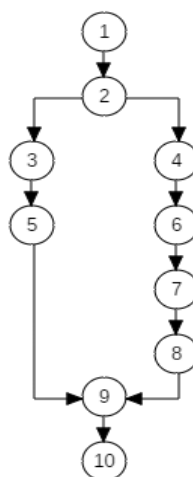
	<i>End</i>
Hasil pengujian	Berhasil

7. Menu Cari Laundry

Menu cari *laundry* berfungsi untuk mencari *laundry* terdekat berdasarkan lokasi pelanggan dan mencari berdasarkan nama.



Gambar IV. 25 Flowchart Cari Laundry



Gambar IV. 26 Flow graph Cari Laundry

Flow graph cari laundry pada gambar IV.26 terdapat 10 *edge* dan 10 *node*. *Flow graph* cari laundry standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 10 \text{ edge} - 10 \text{ node} + 2 = 2$
- c. $V(G) = 2 \text{ simpul diperkirakan} = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.26 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-5-9-10
- b. Jalur 2: 1-2-3-4-6-7-8-9-10

Test case cari laundry dijelaskan pada tabel IV.8.

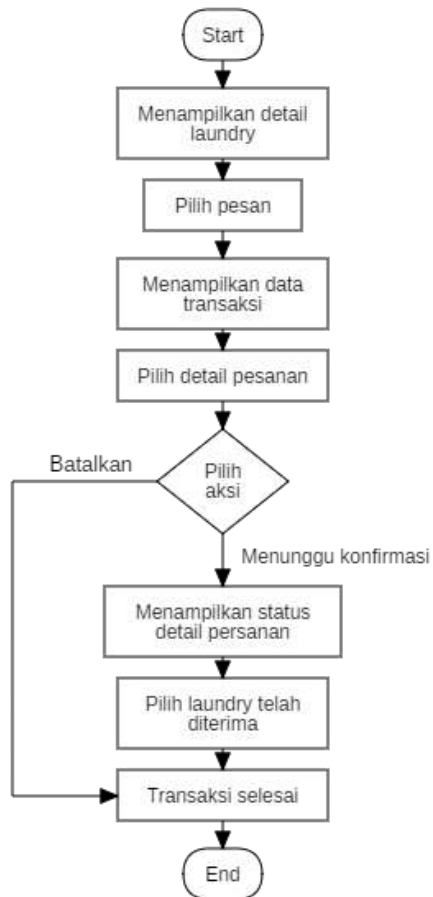
Tabel IV. 8 *Test Case* Cari Laundry

Path	1
Jalur	1-2-3-5-9-10
Skenario	<i>Start</i> Menampilkan map Isi kolom pencarian nama Pilih ‘->’ Menampilkan detail laundry <i>End</i>
Hasil pengujian	Berhasil
Path	2
Jalur	1-2-3-4-6-7-8-9-10
Skenario	<i>Start</i> Menampilkan map Pilih cari lokasi saya Pilih cari Menampilkan daftar laundry Pilih lihat detail Menampilkan detail laundry

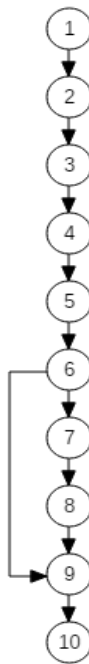
	<i>End</i>
Hasil pengujian	Berhasil

8. Menu Pesan

Menu pesan berfungsi untuk pengguna melakukan pemesanan *laundry*.



Gambar IV. 27 *Flowchart* Pesan



Gambar IV. 28 *Flow graph* Pesan

Flow graph pesan pada gambar IV.28 terdapat 10 *edge* dan 10 *node*. *Flow graph* pesan standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 2 region
- b. $V(G) = 10 \text{ edge} - 10 \text{ node} + 2 = 2$
- c. $V(G) = 1 \text{ simpul diperkirakan} + 1 = 2$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.28 berjumlah 2. Dengan jalur independennya, yaitu:

- a. Jalur 1: 1-2-3-4-5-6-7-8-9-10
- b. Jalur 2: 1-2-3-4-5-6-9-10

Test case pesan dijelaskan pada tabel IV.9.

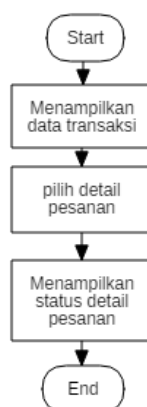
Tabel IV. 9 *Test Case* Pesan

Path	1
Jalur	1-2-3-4-5-6-7-8-9-10
Skenario	<i>Start</i> Menampilkan detail <i>laundry</i> Pilih pesan

	Menampilkan data transaksi Pilih detail pesanan Pilih aksi (dikonfirmasi) Menampilkan status detail pesanan Pilih <i>laundry</i> telah diterima Transaksi selesai <i>End</i>
Hasil pengujian	Berhasil
Path	2
Jalur	1-2-3-4-5-6-9-10
Skenario	<i>Start</i> Menampilkan detail <i>laundry</i> Pilih pesan Menampilkan data transaksi Pilih detail pesanan Pilih aksi (batalkan) Transaksi selesai <i>End</i>
Hasil pengujian	Berhasil

9. Menu Lihat Pesanan

Menu lihat pesanan berfungsi untuk melihat status pesanan dari pengguna.



Gambar IV. 29 *Flowchart* Lihat Pesanan



Gambar IV. 30 *Flow graph* Lihat Pesanan

Flow graph lihat pesanan pada gambar IV.30 terdapat 4 *edge* dan 5 *node*. *Flow graph* lihat pesanan standar kompleksitas siklomatis dihitung menggunakan 3 (tiga) cara, yaitu:

- a. Grafik alir mempunyai 1 region
- b. $V(G) = 4 \text{ edge} - 5 \text{ node} + 2 = 1$
- c. $V(G) = 1 \text{ simpul yang diperkirakan} = 1$

Dengan demikian kompleksitas siklomatis dari *flow graph* yang dijelaskan pada gambar IV.30 berjumlah 1. Dengan jalur independennya, yaitu:

1. Jalur 1: 1-2-3-4-5

Test case lihat pesanan dijelaskan pada tabel IV.10.

Tabel IV. 10 *Test Case* Lihat Pesanan

Path	1
Jalur	1-2-3-4-5
Skenario	<i>Start</i> Menampilkan data transaksi Pilih detail pesanan Menampilkan status detail pesanan <i>End</i>
Hasil pengujian	Berhasil

Rekapitulasi hasil pengujian menggunakan *white box* digambarkan pada tabel IV.11.

Tabel IV. 11 Rekapitulasi Hasil Pengujian

Nama Modul/Menu	CC	Region	Path	Keterangan
<i>Login</i>	2	2	2	Berhasil
Registrasi	2	2	2	Berhasil
Kelola <i>Laundry</i>	2	2	2	Berhasil
Kelola Pengguna	2	2	2	Berhasil
Kelola Penjualan	4	4	4	Berhasil
Proses Pesanan	2	2	2	Berhasil
<i>Cari Laundry</i>	2	2	2	Berhasil
Pesan	2	2	2	Berhasil
Lihat Proses Pesanan	1	1	1	Berhasil

Berdasarkan tabel IV.11 didapatkan hasil rekapitulasi yang dapat menyimpulkan bahwa semua modul/menu yang diujikan berhasil atau bebas dari kesalahan/*error*.

IV.2.2 Pengujian *Black Box*

Pengujian *black box* ini lebih menguji ketampilan luar (*Interface*) dari suatu aplikasi agar mudah digunakan oleh *customer*. Pengujian ini tidak melihat dan menguji *source code* program. *black box* testing bekerja dengan mengabaikan struktur kontrol sehingga hanya terfokus pada informasi domain.

1. Pengujian Fungsional

Tabel IV. 12 Pengujian Fungsional Sistem

No	Komponen Yang Diujikan	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
1	Membuka aplikasi	Menekan ikon aplikasi	Aplikasi akan menampilkan halaman awal	[√] Berfungsi [] Tidak Berfungsi
2	<i>Login</i>	Pilih <i>login, input</i>	Sistem	[√] Berfungsi

No	Komponen Yang Diujikan	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
		<i>username</i> , dan <i>password</i> benar	menerima akses login dan kemudian menampilkan halaman utama tiap pengguna	<input type="checkbox"/> Tidak Berfungsi
		Pilih <i>login</i> , <i>input username</i> dan <i>password</i> salah	Sistem menolak akses login dan tampilan tetap berada di halaman login	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
3	Registrasi	Pilih daftar, isi <i>form</i> registrasi	Sistem menyimpan data <i>user</i>	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
4	Menu data pengguna	Pilih hapus	Sistem menampilkan data pengguna dan dapat dihapus	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
5	Menu data <i>laundry</i>	Pilih lihat detail	Sistem menampilkan detail data <i>laundry</i>	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
		Pilih hapus	Sistem akan melakukan penghapusan	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
6	Menu riwayat transaksi	Pilih lihat detail	Sistem menampilkan	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi

No	Komponen Yang Diujikan	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
			data transaksi dan dapat dilihat detail transaksi	
5	Menu paket <i>laundry</i>	Pilih tambah data lalu pilih simpan	Sistem akan menampilkan <i>form</i> tambah data paket ketika selesai disimpan sistem akan menyimpan data paket	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
		Pilih ubah dan hapus	Sistem akan melakukan pengubahan dan penghapusan	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
6	Menu transaksi	Pilih lihat detail lalu pilih terima pesanan	Sistem akan menampilkan detail pesanan dan mengubah status pesanan	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
		Pilih lihat detail lalu pilih batalkan pesanan	Sistem akan menampilkan detail pesanan dan mengubah status pesanan	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi
7	Menu pencarian <i>laundry</i>	Pilih titik lokasi pengguna lalu pilih cari	Sistem akan menampilkan daftar <i>laundry</i>	<input checked="" type="checkbox"/> Berfungsi <input type="checkbox"/> Tidak Berfungsi

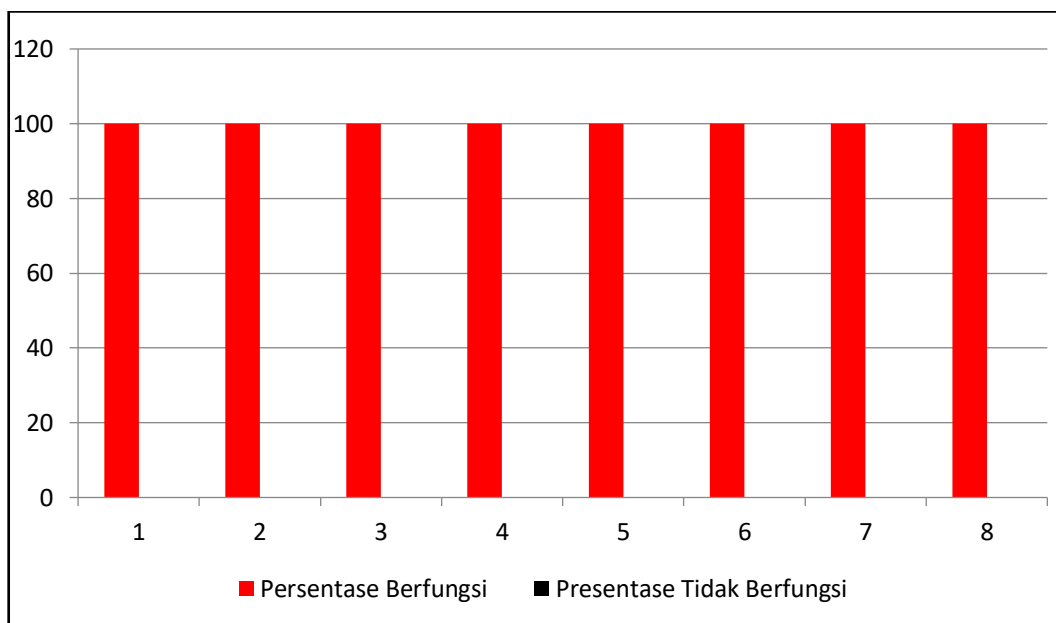
No	Komponen Yang Diujikan	Skenario	Hasil Yang Diharapkan	Hasil Pengujian
			yang telah diurutkan dari terdekat ke terjauh	
		Pilih kolom pencarian nama	Sistem akan menampilkan detail <i>laundry</i> yang telah dicari berdasarkan nama	[√] Berfungsi [] Tidak Berfungsi
8	Menu profil	Pilih ubah	Sistem akan melakukan perubahan data <i>user</i>	[√] Berfungsi [] Tidak Berfungsi

Berdasarkan hasil pengujian fungsional sistem yang dilakukan oleh 10 orang responden dari berbagai kalangan maka didapatkan data pada tabel IV.12 bahwa semua instrument kinerja sistem ini berfungsi sesuai dengan yang diharapkan. Berdasarkan hasil pengujian fungsional yang dilakukan oleh 10 responden dari berbagai kalangan maka didapatkan data sebagai berikut:

Tabel IV. 13 Tabel Pengujian Fungsional

Responden	Pengujian Nomor							
	1	2	3	4	5	6	7	8
1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1

Responden	Pengujian Nomor							
	1	2	3	4	5	6	7	8
6	1	1	1	1	1	1	1	1
7	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1
Total Nilai	10	10	10	10	10	10	10	10
Persentase	100%	100%	100%	100%	100%	100%	100%	100%



Gambar IV. 31 Grafik Hasil Pengujian Fungsional

Grafik hasil pengujian fungsional pada gambar IV.31, dimana persentase dari grafik ini didapatkan dari data hasil inputan 10 orang responden sesuai dengan tabel IV.13.

2. Pengujian Kelayakan Sistem

Pengujian kelayakan perangkat lunak dilakukan dengan pengujian kuesioner yang diuji disini adalah kualitas sistem dan kualitas informasinya, tes berupa angket

terdiri dari 5 pertanyaan yang dibagikan kepada 10 responden. Kuesioner dibuat dengan menggunakan skala *likert* dari skala 1 sampai 7. Berdasarkan data yang dihasilkan dari kuesioner, perhitungan dilakukan dengan menggunakan skala *likert*. Untuk menghitung skor maksimum untuk setiap jawaban, kalikan skor dengan jumlah total responden, yaitu skor dikalikan 10 responden.

Tabel IV. 14 Nilai Skor Maksimum

Jawaban	Skor	Skor maksimum (Skor * Jumlah responden)
Sangat setuju	5	50
Setuju	4	40
Cukup setuju	3	30
Kurang setuju	2	20
Tidak setuju	1	10

Setelah itu dapat dicari persentase masing-masing jawaban dengan menggunakan rumus:

$$Y = \frac{TS}{Skor\ Ideal} \times 100\%$$

Y = Nilai persentase

TS = Total skor responden = Σ skor * responden

Skor ideal = skor * jumlah responden = 5 x 10 = 50

Tabel IV. 15 Kriteria Skor

Kategori	Keterangan
0% - 20%	Tidak setuju
21% - 40%	Kurang setuju
41% - 60%	Cukup setuju
61% - 80%	Setuju
81% - 100%	Sangat Setuju

Kuesioner ini diujikan pada 10 responden (hasil kuisisioner terlampir).

1. Pertanyaan 1: “Apakah anda setuju aplikasi ini cukup mudah digunakan?”. Hasil kuesioner pertanyaan pertama dapat dilihat pada tabel IV.16.

Tabel IV. 16 Hasil Kuesioner Pertanyaan 1

Pertanyaan ke	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
1	Sangat setuju	5	6	30	$(46/50) * 100$
	Setuju	4	4	16	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	46	92%

Berdasarkan nilai penyajian pertanyaan pertama dapat disimpulkan bahwa 92% responden menilai aplikasi cukup mudah digunakan.

2. Pertanyaan 2: “Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi?”. Hasil kuesioner pertanyaan kedua dapat dilihat pada tabel IV.17.

Tabel IV. 17 Hasil Kuesioner Pertanyaan 2

Pertanyaan ke	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
2	Sangat setuju	5	7	35	$(46/50) * 100$
	Setuju	4	2	8	
	Cukup setuju	3	1	3	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	46	92%

Berdasarkan nilai penyajian pertanyaan kedua dapat disimpulkan bahwa 92 % responden menilai setuju mengenai tampilan desain dan warna pada aplikasi.

3. Pertanyaan 3: “Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?”. Hasil kuesioner pertanyaan ketiga dapat dilihat pada tabel IV.18.

Tabel IV. 18 Hasil Kuesioner Pertanyaan 3

Pertanyaan ke	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
3	Sangat setuju	5	3	15	$(41/50) * 100$
	Setuju	4	5	20	
	Cukup setuju	3	2	6	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	41	82%

Berdasarkan nilai penyajian pertanyaan ketiga dapat disimpulkan bahwa 82 % responden menilai setuju mengenai tampilan menu-menu pada aplikasi.

4. Pertanyaan 4: “Apakah anda setuju fitur-fitur dalam aplikasi ini sudah berfungsi dengan baik?”. Hasil kuesioner pertanyaan keempat dapat dilihat pada tabel IV.19.

Tabel IV. 19 Hasil Kuesioner Pertanyaan 4

Pertanyaan ke	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
4	Sangat setuju	5	5	25	$(45/50) * 100$
	Setuju	4	5	20	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	45	90%

Berdasarkan nilai penyajian pertanyaan keempat dapat disimpulkan bahwa 90% responden menilai setuju fitur-fitur dalam aplikasi ini sudah berfungsi dengan baik.

5. Pertanyaan 5: “Apakah anda setuju mengenai tampilan informasi pada aplikasi Ini?”. Hasil kuesioner pertanyaan kelima dapat dilihat pada tabel IV.20.

Tabel IV. 20 Hasil Kuesioner Pertanyaan 5

Pertanyaan ke	Jawaban	Skor	Responden	Jumlah Skor	Nilai Persentase
5	Sangat setuju	5	6	30	$(46/50) * 100$
	Setuju	4	4	16	
	Cukup setuju	3	0	0	
	Kurang setuju	2	0	0	
	Tidak setuju	1	0	0	
Jumlah			10	46	92%

Berdasarkan nilai penyajian pertanyaan kelima dapat disimpulkan bahwa 92% responden menilai setuju mengenai tampilan informasi pada aplikasi.

Tabel IV. 21 Hasil Rata-Rata Persentase

Pertanyaan	Nilai Persentase	Keterangan
1	92%	Sangat setuju
2	92%	Sangat setuju
3	82%	Sangat setuju
4	90%	Sangat setuju
5	92%	Sangat setuju
Total Persentase	$92\% + 92\% + 82\% + 90\% + 92\% = 448$	Sangat setuju
Rata - Rata	$448 / 5 = 89,6\%$	

Pada tabel IV.21 hasil dari setiap pertanyaan dijumlahkan untuk mendapatkan nilai total persentase dan nilai rata-rata. Dari hasil pada tabel IV.21 dapat ditarik kesimpulan bahwa 89,6% responden sangat setuju dengan kualitas yang ada pada aplikasi.

BAB V

PENUTUP

V.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan selama membuat aplikasi pa'*laundry* berbasis Android. Maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Aplikasi pa'*laundry* telah berhasil dibuat dan dapat mewadahi usaha *laundry* dalam penyediaan informasi dan pelayanan jasa *laundry*. Aplikasi ini dibuat menggunakan metode *haversine formula*. Hasil pengujian fungsionalitas aplikasi berfungsi dengan baik berdasarkan hasil pengujian efektifitas responden dengan nilai rata-rata 89,6% sangat setuju pada kualitas aplikasi
2. Pengimplementasian metode *haversine formula* pada aplikasi telah berhasil diterapkan berdasarkan perbandingan antara studi kasus perhitungan akurasi yang dikerjakan oleh sistem dan perhitungan akurasi secara manual yang dilakukan dengan menghitung jarak dari titik pertama dan titik kedua pada lokasi, hasil dari perhitungan akurasi antara kedua kasus tersebut didapatkan nilai 90% akurat dalam pencarian *laundry* terdekat berdasarkan lokasi.

V.2 Saran

Dari kesimpulan yang telah dibahas di atas, sebagai sebuah aplikasi yang baru dibuat dan belum sempurna, masih banyak terdapat kekurangan pada aplikasi. Oleh karena itu, beberapa saran untuk pengembangan aplikasi dikemudian hari, yaitu:

1. Penambahan fitur-fitur pada aplikasi seperti fitur pembayaran online pada aplikasi.
2. Pengembangan aplikasi dengan metode perhitungan jarak terdekat lainnya.

DAFTAR PUSTAKA

- Febrianto, A., & Franita, R. (2020). *Penggunaan Library Fast Android Networking Untuk Membangun Fitur Register Pada Proyek Lotus (Studi Kasus : Pt . Agung Trans Solusindo)*. XX(1), 439–445.
- Felicia Sylviana, Sherly Christina, & Sunartie. (2019). Aplikasi Pemesanan Laundry Berbasis Sms Gateway. *Jurnal Teknologi Informasi Jurnal Keilmuan Dan Aplikasi Bidang Teknik Informatika*, 13(2), 12–17. <https://doi.org/10.47111/jti.v13i2.248>
- Gata, W., & Gata, G. (2013). PENERAPAN BAHASA PEMROGRAMAN JAVA DALAM SISTEM INFORMASI PENJUALAN VERSI DESKTOP. *Budi Luhur Information Technology*, 10(1).
- Iqbal, M., L.B., P., & Kurniati, N. (2021). Penerapan Metode Haversine Pada Sistem Informasi Geografis Pencarian Lokasi Laundry Terdekat di Kota Makassar. *Buletin Sistem Informasi dan Teknologi Islam*, 2(1), 12-16. Retrieved from <http://jurnal.fikom.umi.ac.id/index.php/BUSITI/article/view/710>
- Kusumastuti, A. (2020). Aplikasi Jasa Service Elektronik Berbasis Android Dan Web. *Eprints.Uty.Ac.Id*. <http://eprints.uty.ac.id/6303/>
- Kusuma, S. A. (2019, October). Rancang Bangun Media Pembelajaran Pengenalan Tumbuhan Bagi Penyandang Tuna Rungu Pada SDLB Manisrejo Kota Madiun. In *Prosiding Seminar Nasional Teknologi Informasi dan Komunikasi (SENATIK)* (Vol. 2, No. 1, pp. 281-286).
- Kamil, H., & Duhani, A. (2016). Pembangunan Sistem Informasi Pelayanan Jasa Laundry Berbasis Web Dengan Fitur Mobile Pada 21 Laundry Padang. *Prosiding Semnastek*.
- Mur, M. M., Lia, L., Hafiz, A., Informatika, J. M., Dian, A., Cendikia, C., & Lampung, B. (2019). *Metode Extreme Programming Dalam Membangun Aplikasi Kos-Kosan Di Kota Bandar Lampung Berbasis Web*. XVIII(2013), 377–383.
- Ninik Sri Lestari1, D. S. R. M. (2020). Perancangan Aplikasi Pembuatan Kartu Keluarga Berbasis Web Menggunakan Php Dan Mysql. *Isu Teknologi Stt Mandala*, 15(2), 1–13.
- Nugroho, A., Jumardi, R., & Ramadhania, N. F. (2020). Penerapan Metode Haversine Formula Untuk Penentuan Titik Kumpul pada Aplikasi Tanggap Bencana. *Metik Jurnal*, 4(2), 69-75.

- Purnama Sari, D., & Wijanarko, R. (2020). Implementasi Framework Laravel pada Sistem Informasi Penyewaan Kamera (Studi Kasus di Rumah Kamera Semarang). *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 2(1), 32. <https://doi.org/10.36499/jinrpl.v2i1.3190>
- Rumondor, A. G., Sentinuwo, S. R., & Sambul, A. M. (2019). Perancangan Jalur Terpendek Evakuasi Bencana di Kawasan Boulevard Manado Menggunakan Algoritma Dijkstra. *Jurnal Teknik Informatika*, 14(2), 261-268.
- Simargolang, M. Y., & Nasution, N. (2018). Aplikasi Pelayanan Jasa Laundry Berbasis WEB (Studi Kasus: Pelangi Laundry Kisaran). *Jurnal Teknologi Informasi*, 2(1), 9. <https://doi.org/10.36294/jurti.v2i1.402>
- Susanto, E., Utami, T. H., & Hermanto, D. (2019). Sistem Informasi Pemesanan Laundry Berbasis Android Di Kota Palembang. *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 5(2), 158–168. <https://doi.org/10.35957/jatisi.v5i2.144>
- Suharsana, I. K., & Wirawan, I. W. W. (2016). Implementasi Model View Controller Dengan Framework Codeigniter Pada E-Commerce Penjualan Kerajinan Bali. *Jurnal Sistem dan Informatika (JSI)*, 11(1), 19-28.

**L
A
M
P
I
R
A
N**

KUESIONER

Nama : RINA

Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?	✓				
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?			✓		
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?	✓				
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?	✓				
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?		✓			

Lampiran 1 Kuesioner *Laundry* Responden 1

KUESIONER

Nama : SPY WAHYUNINGSIH

Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?	✓				
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?			✓		
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?	✓				
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?	✓				

Lampiran 2 Kuesioner *Laundry* Responden 2

KUESIONER

Nama : Atsya / ST Laundry
 Status : Belanggan/ Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?	✓				
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?			✓		
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?	✓				
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?		✓			

Lampiran 3 Kuesioner *Laundry* Responden 3

KUESIONER

Nama : Naya
 Status : Belanggan/ Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?	✓				
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?		✓			
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?	✓				
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?	✓				
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?		✓			

Lampiran 4 Kuesioner *Laundry* Responden 4

KUESIONER

Nama : Fajrul Alim Huda
 Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?	✓				
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?		✓			
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?	✓				
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?	✓				
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?		✓			

Lampiran 5 Kuesioner *Laundry* Responden 5

KUESIONER

Nama : Iin Mulyana
 Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?		✓			
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?		✓			
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?		✓			
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?	✓				

Lampiran 6 Kuesioner Pengguna 1

KUESIONER

Nama : Nisa
 Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?		✓			
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?		✓			
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?		✓			
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?	✓				

Lampiran 7 Kuesioner Pengguna 2

KUESIONER

Nama : Norahim
 Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?		✓			
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?		✓			
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?		✓			
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?	✓				

Lampiran 8 Kuesioner Pengguna 3

KUESIONER

Nama : Yudha Arfandi

Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?		✓			
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?		✓			
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?		✓			
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?	✓				

Lampiran 9 Kuesioner Pengguna 4

KUESIONER

Nama : Arisanti Ratih

Status : Pelanggan/Laundry

No	Pertanyaan	Penilaian				
		Sangat Setuju	Setuju	Cukup Setuju	Kurang Setuju	Tidak Setuju
1	Apakah anda setuju aplikasi ini cukup mudah digunakan?	✓				
2	Apakah anda setuju mengenai tampilan desain dan warna pada aplikasi ini?	✓				
3	Apakah anda setuju mengenai tampilan menu-menu pada aplikasi ini?		✓			
4	Apakah anda setuju fitur - fitur dalam aplikasi ini sudah berfungsi dengan baik?		✓			
5	Apakah anda setuju mengenai tampilan informasi pada aplikasi ini?	✓				

Lampiran 10 Kuesioner Pengguna 5