

**RANCANG BANGUN APLIKASI DOORSMEER ONLINE
MENGUNAKAN ALGORITMA A-STAR BERBASIS ANDROID**

SKRIPSI

DISUSUN OLEH

FACHRIZA HABIBI

2009020054



UMSU

Unggul | Cerdas | Terpercaya

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2024

**RANCANG BANGUN APLIKASI DOORSMEER ONLINE
MENGUNAKAN ALGORITMA A-STAR BERBASIS ANDROID**

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar
Sarjana Komputer (S.Kom) dalam Program Studi Teknologi
Informasi pada Fakultas Ilmu Komputer dan Teknologi
Informasi, Universitas Muhammadiyah Sumatera Utara**

FACHRIZA HABIBI

NPM.2009020054

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS MUHAMMADIYAH SUMATERA UTARA**

MEDAN

2024

LEMBAR PENGESAHAN

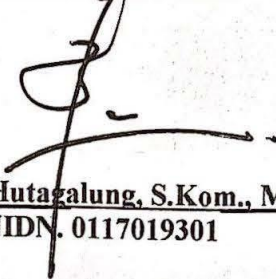
Judul Skripsi : RANCANG BANGUN APLIKASI DOORSMEER
ONLINE MENGGUNAKAN ALGORITMA A-STAR
BERBASIS ANDROID
Nama Mahasiswa : FACHRIZA HABIBI
NPM : 2009020054
Program Studi : TEKNOLOGI INFORMASI

Menyetujui
Komisi Pembimbing



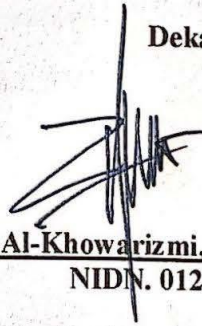
(Ferdy Riza ST, M.Kom)
NIDN.0103068901

Ketua Program Studi



(Fatma Sari Hutagalung, S.Kom., M.Kom.)
NIDN. 0117019301

Dekan



(Dr. Al-Khowarizmi, S.Kom., M.Kom.)
NIDN. 0127099201

PERNYATAAN ORISINALITAS

RANCANG BANGUN APLIKASI DOORSMEER ONLINE MENGUNAKAN ALGORITMA A-STAR BERBASIS ANDROID

SKRIPSI

Saya menyatakan bahwa karya tulis ini adalah hasil karya sendiri, kecuali beberapa kutipan dan ringkasan yang masing – masing disebutkan sumbernya.

Medan, 27 Agustus 2024

Yang membuat pernyataan



Fachriza Habibi

NPM. 2009020054

PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN
AKADEMIS

Sebagai sivitas akademika Universitas Muhammadiyah Sumatera Utara, saya bertanda tangan dibawah ini:

Nama : Fachriza Habibi
NPM : 2009020054
Program Studi : Teknologi Informasi
Karya Ilmiah : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Muhammadiyah Sumatera Utara Hak Bedas Royalti Non-Eksekutif (*Non-Exclusive Royalty free Right*) atas penelitian skripsi saya yang berjudul:

RANCANG BANGUN APLIKASI DOORSMEER ONLINE
MENGUNAKAN ALGORITMA A-STAR BERBASIS ANDROID

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Non-Eksekutif ini, Universitas Muhammadiyah Sumatera Utara berhak menyimpan, mengalih media, memformat, mengelola dalam bentuk database, merawat dan mempublikasikan Skripsi saya ini tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis dan sebagai pemegang dan atau sebagai pemilik hak cipta.

Demikian pernyataan ini dibuat dengan sebenarnya.

Medan, 27 Agustus 2024

Yang membuat pernyataan

A handwritten signature in black ink, appearing to read 'Fachriza Habibi', written on a light-colored, textured background.

Fachriza Habibi

NPM. 2009020054

:

RIWAYAT HIDUP

DATA PRIBADI

Nama Lengkap : Fachriza Habibi
Tempat dan Tanggal Lahir : Semarang, 18 September 2002
Alamat Rumah : Jl. Brigjend. Katamso No.66
Telepon/Faks/HP : 089608323577
E-Mail : Fachrizah8@gmail.com
Instansi Tempat Kerja : -
Alamat Kantor : -

DATA PENDIDIKAN

SD : SD NEGERI 01 NGALIYAN TAMAT : 2014
SMP : SMP NEGERI 36 MEDAN TAMAT : 2017
SMA : SMA SWASTA AL-ULUM TAMAT : 2020

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji dan syukur bagi Allah Subhanahu Wata'ala yang telah memberikan rahmat serta hidayahnya sehingga penulis dapat menyelesaikan skripsi yang berjudul **“RANCANG BANGUN APLIKASI DOORSMEER ONLINE MENGGUNAKAN ALGORITMA A-STAR BERBASIS ANDROID”**. Skripsi ini disusun untuk memenuhi salah satu syarat untuk memperoleh gelar strata satu (S1) pada program studi teknologi informasi.

Dalam penyusunan skripsi ini, penulis banyak mendapatkan arahan, nasihat dan bimbingan sehingga skripsi ini dapat terselesaikan dengan baik. Penulis ingin menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Agussani, M.AP selaku Rektor Universitas Muhammadiyah Sumatera Utara.
2. Bapak Prof. Dr. Muhammad Arifin, S.H., M.Hum selaku Wakil Rektor I Universitas Muhammadiyah Sumatera Utara.
3. Bapak Prof. Dr. Akrim, M.Pd selaku Wakil Rektor II Universitas Muhammadiyah Sumatera Utara.
4. Bapak Assoc. Prof. Dr. Rudianto, S.Sos., M.Si selaku Wakil Rektor III Universitas Muhammadiyah Sumatera Utara.

5. Bapak Dr. Al-Khowarizmi, S.Kom., M.Kom selaku Dekan Fakultas Ilmu Komputer dan Teknologi Informasi.
6. Bapak Halim Maulana., ST., M.Kom selaku Wakil Dekan I Fakultas Ilmu Komputer dan Teknologi Informasi.
7. Bapak Lutfi Basit, S.Sos., M.I.Kom selaku Wakil Dekan III Fakultas Ilmu Komputer dan Teknologi Informasi.
8. Ibu Fatma Sari Hutagalung, S.Kom., M.Kom selaku Ketua Jurusan Program Studi Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi.
9. Bapak Ferdy Riza ST, M.Kom selaku Dosen Pembimbing penulis yang sudah membantu dan membimbing penulis untuk menyelesaikan Skripsinya
10. Almarhum Ayah saya yang tercinta, Bapak Odhing Ralim Putra yang sudah pasti memotivasi penulis sejak lama, dan sudah pasti juga membantu penulis melalui doa ditempat yang sudah pasti lebih baik. Rest In Peace Papa.
11. Yang Tercinta, yaitu ibu penulis sendiri, Tengku Nilda Mufiza yang tiada henti mendorong dan pastinya mendoakan anak-anaknya untuk mendapatkan hal yang terbaik dan dapat mencapai puncaknya. Makasi banyak Mamica.
12. Tidak lupa juga Abang-abang dan Adik penulis, Muhammad Haviz, Nicky El Imran, Audy Akbar, dan adik penulis, Maimoon Al-Razzaq telah mendorong penulis untuk menyelesaikan skripsi melalui motivasi-

motivasi dan juga doa, serta bantuan-bantuan lainnya yang mampu menyemangati penulis untuk menyelesaikan skripsinya. Love You All

13. Dan tentu saja, Teman-teman penulis, tentunya teman-teman kkn penulis yang telah merasakan naik dan pasti turunnya penulis dalam menyelesaikan skripsi, yang turut merasakan perasaan penulis dan mampu membantu penulis dalam menyelesaikan skripsinya. Untuk sekarang hanya terima kasih yang bisa penulis sampaikan kepada seluruh teman-teman penulis. Thanks Guys

Penulis menyadari bahwa masih banyak kekurangan dalam penyusunan skripsi ini. Oleh karena itu, penulis akan sangat menghargai dan menerima kritikan dan saran untuk membuat skripsi ini menjadi lebih baik lagi. Semoga skripsi ini dapat memberikan manfaat bagi kita semua dan semoga bantuan serta partisipasi yang diberikan oleh semua pihak bernilai ibadah disisi Allah Subhanahu Wata'ala.

Wassalamualaikum Warahmatullahi Wabarakatuh

Medan, 27 Agustus 2024

Penulis



Fachriza Habibi

RANCANG BANGUN APLIKASI DOORSMEER ONLINE MENGUNAKAN ALGORITMA A-STAR BERBASIS ANDROID

ABSTRAK

Permintaan akan layanan pencucian kendaraan bermotor kini meningkat di kawasan perkotaan karena menawarkan kenyamanan dan biaya yang lebih terjangkau. Namun, antrian dan kurangnya informasi mengenai waktu layanan sering kali mengakibatkan ketidakpuasan pelanggan. Untuk mengatasi masalah ini dan meningkatkan efisiensi, aplikasi Doorsmeer Online dirancang sebagai platform berbasis Android yang menghubungkan pemilik usaha doorsmeer dan pengguna layanan. Aplikasi ini menawarkan dua peran utama: Owner dan User. User dapat dengan mudah mencari tempat pencucian terdekat, memesan layanan, dan berkomunikasi dengan penyedia layanan. Sementara itu, Owner dapat mengelola profil, harga, dan tugas terkait pemesanan. Dengan memanfaatkan Algoritma A* untuk rute terdekat dan Algoritma Priority Scheduling untuk penjadwalan, aplikasi ini bertujuan untuk mengurangi waktu antrian, meningkatkan kenyamanan pengguna, dan efektivitas kerja pemilik usaha. Pengujian dilakukan menggunakan metode Black Box untuk memastikan fungsionalitas aplikasi sesuai dengan kebutuhan pengguna. Hasilnya menunjukkan bahwa aplikasi ini berhasil meningkatkan efisiensi dalam mencari dan memesan layanan pencucian kendaraan, memberikan solusi praktis bagi kedua peran yang ada.

Kata Kunci: Android; Algoritma A-Star; Doorsmeer; React Native

DESIGN OF ONLINE DOORSMEER APPLICATION USING ANDROID-BASED A-STAR ALGORITHM

ABSTRACT

Demand for car wash services is increasing in urban areas as they offer convenience and affordability. However, queues and lack of information regarding service times often result in customer dissatisfaction. To address these issues and improve efficiency, the Doorsmeer Online app was designed as an Android-based platform that connects doorsmeer business owners and service users. The app offers two main roles: Owner and User. Users can easily search for the nearest washroom, book services, and communicate with service providers. Meanwhile, the Owner can manage profiles, prices, and tasks related to bookings. By utilizing A* Algorithm for the closest route and Priority Scheduling Algorithm for scheduling, this application aims to reduce queue time, increase user convenience, and work effectiveness of business owners. Testing is done using the Black Box method to ensure the application's functionality is in accordance with user requirements. The results show that this application successfully improves efficiency in finding and booking vehicle wash services, providing a practical solution for both existing roles.

Keywords: Android; Algoritma A-Star; Doorsmeer React Native

DAFTAR ISI

LEMBAR PENGESAHAN	i
PERNYATAAN ORISINALITAS.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
RIWAYAT HIDUP	v
KATA PENGANTAR.....	vi
ABSTRAK	ix
ABSTRACT	x
DAFTAR ISI.....	xi
DAFTAR TABEL.....	xiv
DAFTAR GAMBAR.....	xv
DAFTAR LAMPIRAN	xv
BAB I. PENDAHULUAN	1
1.1. Latar Belakang Masalah	1
1.2. Rumusan Masalah	4
1.3. Batasan Masalah	5
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	5
BAB II. LANDASAN TEORI	7
2.1. Doorsmeer	7
2.2. Antrian	9
2.3. Android.....	10
2.4. GPS.....	13
2.5. Algoritma A-Star	13
2.6. Algoritma Priority Scheduling	15
2.7. React Native	17
2.8. Database	17
2.9. React Native	18
2.10. Figma	19
2.11. Visual Studio Code.....	20
2.12. Rapid Application Development (RAD)	21

2.13. Black Box Testing	22
2.14. Flowchart.....	23
2.15. UML (Unified Modelling Language)	29
2.16. Kerangka Berpikir Konseptual	30
BAB III. LANDASAN TEORI.....	32
3.1. Desain Penelitian	32
3.2. Metode Pengembangan Aplikasi	33
3.2.1 Rapid Application Development (RAD)	33
3.3. Teknik Pengumpulan Data	35
3.4. Pengguna Sistem	35
3.5. Flowchart.....	36
3.6. UML (Unified Modelling Language).....	39
3.6.1 Use Case Diagram.....	39
3.6.2 Class Diagram.....	41
3.6.3 Sequence Diagram	41
3.6.4 Activity Diagram	42
3.7. Wireframe.....	42
BAB IV. HASIL DAN PEMBAHASAN.....	47
4.1. Implementasi Interface Aplikasi Berbasis Android	47
4.1.1. Implementasi Halaman Login User	47
4.1.2. Implementasi Halaman Registrasi User	48
4.1.3. Implementasi Halaman Home	49
4.1.4. Implementasi Halaman Order	50
4.1.5. Implementasi Halaman Profile	50
4.1.6. Implementasi Halaman Login <i>Owner</i>	51
4.1.7. Implementasi Halaman Registrasi <i>Owner</i>	52
4.1.8. Implementasi Halaman Home <i>Owner</i>	52
4.1.9. Implementasi Halaman Profil <i>Owner</i>	53
4.1.10. Implementasi Halaman <i>Task Screen</i>	54
4.1.11. Implementasi Halaman <i>Order Room</i>	54
4.2. Pengujian Aplikasi.....	55
BAB V. PENUTUP.....	58

5.1. Kesimpulan.....	58
5.2. Saran.....	59
DAFTAR PUSTAKA.....	60
LAMPIRAN.....	63

DAFTAR TABEL

Tabel 2.1. Versi – Versi Android	12
Tabel 2.2. Perhitungan Algoritma Priority Scheduling	16
Tabel 2.3. Contoh Isi Data Pengguna	19
Tabel 2.4. Simbol Flowchart	24
Tabel 2.5. Simbol Penghubung Alur	25
Tabel 2.6. Simbol Input – Output	26
Tabel 2.7. Simbol Proses	27
Tabel 3.1. Defenisi Aktor	40
Tabel 4.1. Tabel Pengujian Role User	55
Tabel 4.2. Tabel Pengujian Role Owner	56

DAFTAR GAMBAR

Tabel 2.1 Versi – Versi Android.....	12
Tabel 2.2 Perhitungan Algoritma Priority Scheduling.....	16
Tabel 2.3 Contoh Isi Data Pengguna	19
Tabel 2.4 Simbol Flowchart	24
Tabel 2.5 Simbol Penghubung Alur.....	25
Tabel 2.6 Simbol Input – Output	26
Tabel 2.7 Simbol Proses.....	27
Tabel 3.1. Defenisi Aktor.....	40
Tabel 4. 1. Tabel Pengujian Role User	55
Tabel 4.2. Tabel Pengujian Role Owner	56
Lampiran 1. Source Code User	63
Lampiran 2. Source Code Owner	72

DAFTAR LAMPIRAN

Lampiran 1. Source Code User	63
Lampiran 2. Source Code Owner	72

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Layanan mencuci kendaraan bermotor saat ini semakin populer di banyak wilayah perkotaan dan bahkan telah menjangkau pinggiran daerah. Permintaan akan layanan ini meningkat karena tarifnya terjangkau dan lebih praktis daripada mencuci sendiri. Seiring dengan pertumbuhan jumlah kendaraan bermotor yang pesat, permintaan akan layanan pencucian kendaraan juga meningkat. Di berbagai lokasi, terdapat beragam jenis jasa ini, mulai dari yang sederhana hingga yang dilengkapi dengan fasilitas lengkap seperti peralatan mekanis untuk mencuci, mesin semprot air, area tunggu yang nyaman, dan bahkan beberapa tempat menyediakan layanan bengkel, mencuci, dan *café* dalam satu tempat dengan konsep one-stop service. (Bambang & Arhami, 2021)

Salah satu masalah yang sering muncul di tempat umum adalah antrian. Antrian merupakan hal yang biasa kita jumpai dan lakukan dalam kegiatan sehari-hari. Misalnya, menunggu di loket kereta api, pembelian tiket bioskop, antrian di jalan tol, di bank, rumah sakit, atau bahkan di kasir minimarket terdekat (Hidayat et al., 2020). Selain itu, tidak adanya informasi kapan pelanggan akan mendapatkan layanan. Hal ini tentu saja berdampak pada kekecewaan pelanggan yang ujungnya pelanggan pergi dan mencari tempat lain (Setyawatu & Bachtiar Maulachela, 2020).

Perkembangan teknologi informasi telah menjadi landasan utama yang mengubah cara perusahaan mengelola bisnis mereka. Di era di mana informasi menjadi lebih mudah diakses dan tersebar, teknologi informasi memegang peran

kunci dalam membentuk strategi, operasional, serta interaksi perusahaan dengan pelanggan dan mitra bisnis (Prabowo et al., 2023).

Menurut (Purnamasari et al., 2021), Untuk mencapai laba yang layak, salah satu upaya adalah dengan meningkatkan kualitas produk yang diproduksi serta menekan biaya yang dikeluarkan. Bagi para pelaku ekonomi dalam menghadapi persaingan tersebut dapat menggunakan seluruh potensi yang ada secara efektif dan efisien. Salah satu strategi yang ada saat ini dalam perkembangan teknologi manufaktur saat ini dengan sistem *Just In Time* (JIT).

JIT dikenal sebagai suatu filosofi untuk menyelesaikan produk pada setiap tahap seketika (*Just In Time*) dan dengan biaya yang minimum. Setiap perusahaan umumnya bertujuan untuk memaksimalkan laba. Oleh karena itu, untuk mencapai laba yang maksimum tersebut diperlukan suatu sistem agar kemampuan yang dimiliki suatu perusahaan dapat mencapai tujuan tersebut. Dengan menerapkan sistem *Just In Time* ini maka diharapkan perusahaan dalam proses produksinya akan memiliki biaya yang rendah, harga jual yang murah, kualitas yang baik, dan kemampuan ketepatan waktu pengiriman kepada pelanggan. (Purnamasari et al., 2021)

Namun dalam proses penyusunan penjadwalan sering kali mengalami kendala. Dengan pesatnya perkembangan teknologi saat ini, tentunya menjadi hal yang menguntungkan bagi semua pihak termasuk dalam pembuatan jadwal ini. (Sumantri & Laluma, 2022)

Dengan kemajuan teknologi mobile terutama di platform Android, banyak startup yang mulai mengembangkan layanan jasa online. Ini mencakup layanan antar penumpang, pengiriman barang, pengiriman makanan, dan bahkan pembelian

obat-obatan. Sebagian besar startup ini menghasilkan aplikasi berbasis Android dan memanfaatkan Google Maps sebagai peta online untuk mendukung layanan yang mereka tawarkan (Susanto & Handayani, 2022). Perkembangan ini menciptakan kebiasaan baru di masyarakat, di mana orang-orang cenderung mencari produk atau layanan melalui aplikasi smartphone Android sebelum membuat pemesanan secara online atau mengunjungi lokasi fisik (Susanto & Handayani, 2022).

Android merupakan sistem operasi yang secara luas digunakan dan difokuskan pada pengguna akhir. Dengan meningkatnya berbagai kasus penggunaan, termasuk komunikasi, navigasi, konsumsi media, hiburan, keuangan, kesehatan, serta akses terhadap sensor, aktuator, kamera, atau mikrofon, model keamanannya harus mampu mengatasi sejumlah ancaman praktis dalam berbagai skenario, sekaligus bermanfaat bagi para pakar di luar bidang keamanan. (Mayrhofer et al., 2021)

Algoritma A – Star adalah metode yang digunakan untuk menemukan rute terpendek dari suatu objek menuju ke tujuan dengan memperhitungkan biaya yang harus dikeluarkan untuk mencapai harga terkecil yang perlu dibayarkan (Fallo & Bulu, 2022). Dengan menggunakan Algoritma A – Star, kita dapat mengurangi waktu eksekusi, dan meningkatkan efisiensi algoritma tanpa mengubah proses pencarian titik tujuan (Hong et al., 2021). Dengan memanfaatkan algoritma ini, para pekerja dapat menentukan jarak antara mereka dengan yang menggunakan layanan Doorsmeer Online, untuk mencari rute terdekat dan menghemat biaya yang perlu dikeluarkan untuk mencapai tujuan.

Algoritma *Priority Scheduling* merupakan algoritma penjadwalan berprioritas. Pengerjaan mendahulukan proses yang memiliki prioritas terbesar. Dalam penelitian ini, algoritma *Priority Scheduling - First In First Out (FIFO)* digunakan

untuk menentukan prioritas pekerjaan berdasarkan status dari masing-masing *task*. *Task* yang memiliki nilai prioritas lebih besar akan berada pada urutan lebih awal dibandingkan *task* yang memiliki nilai prioritas lebih rendah (Setyawatu & Bachtiar Maulachela, 2020). Algoritma penjadwalan prioritas *preemptive* adalah salah satu algoritma yang umum digunakan di antara berbagai algoritma lain untuk penjadwalan CPU. Namun, algoritma ini dapat menyebabkan masalah kelaparan, yang terjadi ketika proses dengan prioritas lebih rendah tidak diberi kesempatan untuk menggunakan CPU karena proses dengan prioritas lebih tinggi terus memperpanjang penggunaan CPU (Chandiramani et al., 2019). Dengan adanya algoritma ini, kita dapat menentukan pengguna yang memesan layanan ini terlebih dahulu dan dapat mengimplementasikan fitur penjadwalan, dimana sang pengguna dapat mengetahui kurang lebihnya waktu yang diperlukan untuk sang pekerja mencapai tujuan sang pengguna, memulai pekerjaannya, sampai selesai.

Dengan adanya aplikasi ini dan dengan memanfaatkan algoritma A – Star dan algoritma *Priority Scheduling*, dapat mempermudah urusan yang bersangkutan dan mengurangi masalah yang harus mereka hadapi, dan juga dapat membujuk masyarakat untuk mulai menggunakan teknologi untuk membuat keseharian mereka menjadi lebih mudah.

1.2. Rumusan Masalah

Berdasarkan dari latar belakang diatas, maka permasalahan yang terdapat dalam penelitian ini adalah:

- 1 Bagaimana merancang aplikasi “Doorsmeer Online” berbasis android?
- 2 Bagaimana mengimplementasikan algoritma A-Star dan *Priority Scheduling* kedalam aplikasi tersebut?

1.3. Batasan Masalah

Adapun batasan masalah dalam penelitian ini sebagai berikut :

- 1 Rancangan ini hanya dapat digunakan di kota Medan dan sekitarnya.
- 2 Aplikasi ini dibangun menggunakan React Native
- 3 Aplikasi ini dibuat menggunakan Algoritma A-Star dan Priority Scheduling.

1.4. Tujuan Penelitian

Tujuan Penelitian ini adalah untuk menghilangkan waktu mengantri ditempat umum dan mengubahnya menjadi mengantri di rumah sendiri. Adapula peningkatan efektivitas pekerja pada sebuah tempat Doorsmeer. Penggunaan aplikasi ini bisa digunakan untuk melihat berapa lama waktu yang perlu ditunggu untuk pekerja dapat memulai.

1.5. Manfaat Penelitian

Manfaat yang dapat diambil dari perancangan aplikasi “Doorsmeer Online” berbasis android adalah :

- 1 Bagi Penulis

Dengan mengadakan pembuatan aplikasi ini, maka penulis dapat memajukan masyarakat untuk memanfaatkan teknologi dalam kesehariannya,

- 2 Bagi Almamater :

Hasil penelitian ini dapat digunakan sebagai referensi untuk penelitian yang akan datang untuk bidang yang sama dan diharapkan dapat menambah pengetahuan serta dapat dijadikan studi perbandingan dimasa depan.

3. Bagi Masyarakat :

Untuk memberikan kemudahan kepada orang – orang yang ingin membersihkan kendaraannya, mengurangi waktu mengantri, memberi kenyamanan pada pengguna, dan meningkatkan efektivitas pekerja doorsmeer.

BAB II

LANDASAN TEORI

2.1. Doorsmeer

Doorsmeer merupakan usaha yang bergerak dalam usaha *service* atau perbaikan alat transportasi yang sering digunakan masyarakat seperti kendaraan bermotor. Kegiatan yang paling sering dilakukan dan yang paling banyak diminati konsumen yaitu perbaikan dan pencucian (doorsmeer) mobil. (Fristy, 2021)

Secara umumnya doorsmeer memiliki arti sebagai pencucian atau pembersihan bercak atau noda yang menempel di bagian tubuh mobil. Doorsmeer mobil sama halnya seperti mandi pada manusia yang membersihkan kotoran atau bercak yang terlihat, sehingga mobil tetap bersih dan terawat juga nyaman untuk digunakan setiap hari. (Tampubolon, 2022)

Dewasa ini, sepeda motor dan mobil termasuk bagian primer dalam kehidupan sehari-hari baik bagi para pria dan wanita. Saat ini tidak hanya pria saja yang dapat menggunakan sepeda motor dan mobil tetapi juga para wanita, akan memanfaatkan sepeda motor demi alasan ekonomis, praktis dan efisien. (Fristy, 2021)



Gambar 2.1 Proses Doorsmeer / Cuci Kendaraan

(Sumber : Quora <https://id.quora.com/Berapa-modal-yang-dibutuhkan-untuk->

usaha-doorsmeer-mobil)

2.2. Antrian

Mengantri merupakan sikap mental yang erat kaitannya dengan disiplin individu dan disiplin. Aktivitas mengantri bukanlah hal baru; *lineup* terbentuk ketika permintaan untuk layanan melebihi kapasitas layanan dan fasilitas layanan, mencegah pengguna fasilitas menerima layanan segera setelah mereka tiba karena layanan yang sibuk. Sekelompok orang yang memiliki minat, dan keinginan yang sama untuk melihat kebutuhan dan minat itu terpenuhi terlibat dalam aktivitas sosial mengantri; tetapi, karena keterbatasan waktu dan sumber daya yang langka, setiap orang bergiliran mengikuti norma layanan. (Ayunita et al., 2023)

Menurut uraian tersebut, “budaya antri” mencakup semua pola perilaku yang ditunjukkan oleh seseorang atau sekelompok orang dalam kehidupan bermasyarakat. Pola tersebut diturunkan dari proses mental kelompok yang mematuhi perintah menunggu giliran untuk mendapatkan kesempatan atau barang tertentu. Budaya antri mengisyaratkan kedisiplinan atau kedisiplinan. Dengan kaitan lain, saat mengantri, setiap individu atau kelompok individu perlu disiplin, gigih, dan didukung oleh aspek tanggung jawab. Artinya, orang atau sekelompok orang yang mengantri harus mampu mempertanggungjawabkan posisinya, menjaga posisinya, dan berusaha menjauh dari pengaruh negatif yang mungkin menghamprinya karena memerlukan kesadaran diri dan pertimbangin pengaruh lingkungan/kebiasaan adat, budaya antri dalam kaitannya dengan ajaran Islam merupakan kebiasaan yang perlu diperhatikan. (Ayunita et al., 2023)

Berdasarkan hasil wawancara yang dilakukan oleh pemilik perusahaan menghasilkan suatu permasalahan berupa terjadinya antrian yang mengakibatkan

kerugian waktu dan tenaga, permasalahan tersebut mempengaruhi pendapatan perusahaan dan kualitas kinerja perusahaan ketika pelanggan enggan menunggu dan memilih tempat cuci mobil yang lain, sehingga penting bagi perusahaan untuk menerapkan model antrian yang dapat memberikan informasi waktu tunggu. Antrian terjadi juga diakibatkan belum adanya proses booking secara *online* untuk memberikan tambahan layanan bagi pelanggan sehingga pelanggan tidak harus menunggu pada tempat pencucian. (Saputra et al., 2020)



Gambar 2.2 Antrian Doorsmeer

(Sumber : <https://karosatuklik.com/usaha-doorsmeer-di-kota-kabanjahe-dan-berastagi-tidak-terpengaruh-pandemi/>)

2.3. Android

Android merupakan sistem operasi berbasis linux yang digunakan untuk perangkat mobile atau perangkat tablet (PDA). Android menyediakan *platform* terbuka bagi para pengembang untuk mengembangkan aplikasi buatan sendiri yang

membuat android menjadi sistem operasi mobile yang populer hingga saat ini. Android didirikan pada tahun 2003 oleh Andy Rubin, Nick Sears, Rich Miner dan Chris White yang kemudian dibeli oleh Google pada bulan Juli 2005. (Windane & Lathifah, 2021)

Sistem operasi android saat ini dengan kecepatan yang sangat spektakuler telah tumbuh dengan pesat. Kejayaan android tidak lepas dari sifatnya yang terbuka, yang menyodorkan kode sumber perangkat lunak secara percuma sehingga para pengembang dapat mengembangkan, mengedarkan dan mengadakannya tanpa membayar lisensi. Selain itu android menyediakan aplikasi berbayar maupun yang gratis dari pengembang android sehingga mempermudah pemakai android. (Widiastika et al., 2020)



Gambar 2.3 Logo Android

(Sumber : <https://www.pngwing.com/id/search?q=android>)

Tabel 2.1 Versi – Versi Android

Versi	Rilis
<i>Android 1.0 Alpha.</i>	23 September 2008
<i>Android 1.1 Beta.</i>	9 Februari 2009
<i>Android 1.5 Cupcake.</i>	30 April 2009
<i>Android 1.6 Donut.</i>	15 September 2009
<i>Android 2.0 Eclair.</i>	26 Oktober 2009
<i>Android 2.2 Froyo.</i>	20 Mei 2010
<i>Android 2.3 Gingerbread.</i>	6 Desember 2010
<i>Android 3.0 Honeycomb.</i>	22 Februari 2011
<i>Android 4.0 Ice cream sandwich.</i>	19 Oktober 2011
<i>Android 4.1 Jelly bean.</i>	27 Juli 2012
<i>Android 4.4 Kitkat.</i>	31 Oktober 2013
<i>Android 5.0 Lollipop.</i>	25 Juni 2014
<i>Android 6.0 Marshmallow.</i>	5 Mei 2015
<i>Android 7.0 Nougat.</i>	19 Oktober 2016
<i>Android 8.0 Oreo.</i>	7 November 2017
<i>Android 9.0 Pie.</i>	6 Agustus 2018
<i>Android 10 Quince Tart</i>	3 September 2019
<i>Android 11 Red Velvet Cake</i>	8 September 2020
<i>Android 12 Snow Cone</i>	14 Oktober 2021
<i>Android 13 Tiramisu</i>	15 Agustus 2022
<i>Android 14 Upside Down Cake</i>	4 Oktober 2023

2.4. GPS

Saat ini perkembangan teknologi semakin canggih dimana pada *mobile phone* sudah dipasang modul GPS. Hal ini tentunya akan memudahkan pekerjaan dalam berbagai bidang dan mendukung penggunaan aplikasi pemetaan. Penggunaan smartphone berbasis Android dalam pelaksanaan survey lapangan merupakan hal yang sudah sering dilakukan. Beberapa aplikasi pemetaan seperti *CarryMap*, *Map Plus*, *GoogleMap*, *Waze* dan *GPS Tracker* adalah sebagian dari banyak aplikasi yang umum digunakan pada berbagai platform Android maupun iOS Apple (Farida & Mutiono, 2023).

2.5. Algoritma A-Star

Salah satu jenis algoritma yang digunakan untuk menyelesaikan kasus yang berkaitan dengan pencarian jalan (pencarian jalan) adalah algoritma A - Star. Dengan skor A - Star, hasil pencariannya dianggap lengkap dan optimal. Algoritma A - Star menggunakan metode pencarian awal *Breadth*, yang melakukan pencarian dengan melebar ke setiap *node* pada level yang sama. Metode ini akan menemukan rute terbaik dari titik awal hingga tujuan. Algoritma A - Star juga memiliki fungsi heuristik. Fungsi ini digunakan untuk mengoptimasi dalam menentukan node tujuan yang akan dipilih. Algoritma pencarian pertama *breadth* A - Star selalu memperluas jalan menuju *node* yang akan dilewati, tetapi tidak memperhitungkan biaya ke simpul tersebut. Algoritma ini menggunakan fungsi heuristik yang kompleks untuk memilih jalan. (Hutabalian et al., 2022).

Algoritma ini memeriksa node dengan menggabungkan $g(n)$, yaitu *cost* yang dibutuhkan untuk mencapai sebuah *node* dan $h(n)$ yaitu *cost* yang didapat dari *node* ke tujuan. Sehingga dapat di rumuskan sebagai:

$$\text{Rumus : } f(n) = g(n) + h'(n) \dots (2.1)$$

Keterangan :

$f(n)$: adalah estimasi total biaya (*cost*) sebuah jalur (*path*) dari *node* awal ke *node* tujuan (*goal*) melalui *node* n .

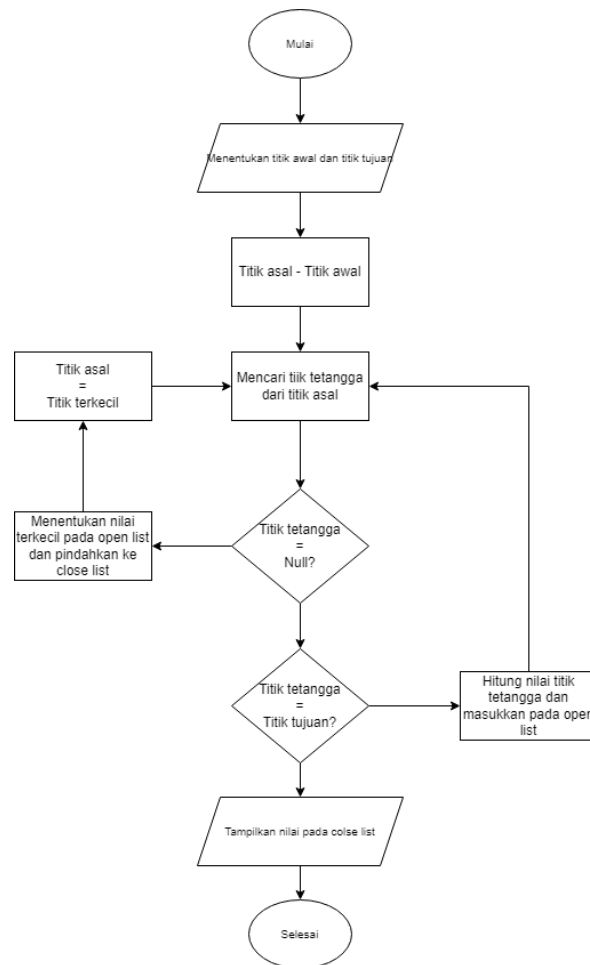
$g(n)$: adalah biaya (*cost*) yang dibutuhkan oleh sebuah jalur (*path*) untuk mencapai *node* n dari *node* awal.

$h'(n)$: adalah estimasi biaya (*cost*) sebuah jalur (*path*). (Hutabalian et al., 2022)

$h'(n)$ adalah nilai Heuristik yang ditentukan dengan melakukan perhitungan dengan rumus *Heuristic Euclidean Distance*. Berikut adalah rumus dari HED:

$$d(x + y) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \dots (2.2) \text{(Mutsaqov, Ativ, Muhammad}$$

Fernando & Megawaty, 2020)



Gambar 2.4 Flowchart Algoritma A-Star

(Mutsaqov, Ativ, Muhammad Fernando & Megawaty, 2020)

2.6. Algoritma Priority Scheduling

Salah satu algoritma penjadwalan berprioritas adalah algoritma penjadwalan prioritas. Proses yang paling penting adalah pengerjaan. Studi ini menggunakan algoritma pengaturan prioritas - *First In First Out* (FIFO) untuk menentukan prioritas pekerjaan berdasarkan status masing-masing tugas. *Task* dengan nilai prioritas rendah akan diprioritaskan lebih awal daripada yang dengan nilai prioritas tinggi (Setyawatu & Bachtiar Maulachela, 2020).

Priority scheduling adalah algoritma penjadwalan *preemptive* yang memberikan nilai prioritas kepada setiap proses. Nilai prioritas ini dapat bersifat statis atau dinamis, yang berubah seiring waktu. Proses dengan prioritas tertinggi akan dijadwalkan lebih dahulu. Jika proses terbaru masuk dengan prioritas yang lebih tinggi dari proses sebelumnya yang sedang berjalan, maka akan terjadi *preemption*, yang mengarahkan prosesor ke proses yang terbaru tersebut. (Sundari & Yogie Syahputra, 2023)

Berikut adalah contoh dari perhitungan algoritma *Priority Scheduling* (Sundari & Yogie Syahputra, 2023):

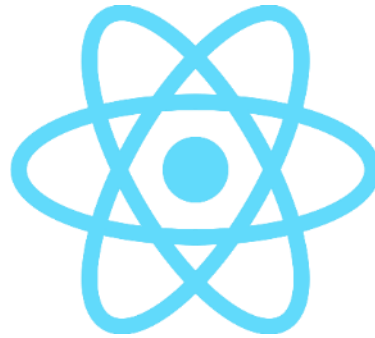
Tabel 2.2 Perhitungan Algoritma Priority Scheduling

Nama Proses	Waktu Tiba	Lama Eksekusi
Proses 5	0	30
Proses 4	10	25
Proses 1	20	30
Proses 8	40	35
Proses 6	50	40
Proses 2	55	60
Proses 3	60	60
Proses 9	65	65
Proses 7	70	40
Proses 10	80	45

2.7. React Native

Dikembangkan oleh pengembang *Facebook* pada tahun 2015, *React Native* cenderung memiliki satu tempat untuk menulis kode dalam *Javascript* ES6 sebagai Bahasa pemrograman dan membangun aplikasi mobile untuk Android dan iOS secara bersamaan untuk kedua platform mobile tersebut. (Gupta et al., 2018)

React Native adalah platform gratis untuk mengembangkan aplikasi *native* untuk ponsel, sebagian besar dikembangkan oleh *Facebook*. Hal yang menarik dari menggunakan *React Native* adalah aplikasi yang dibangun sepenuhnya *native* (Karim & Adriansyah, 2022).



Gambar 2.5 Logo React Native

(Sumber :

https://www.iconfinder.com/icons/1174949/js_react_js_logo_react_react_native_icon)

2.8. Database

Database adalah kumpulan tabel yang memuat data fakta yang disimpan secara digital dan dapat diakses menggunakan program komputer. Fungsinya adalah untuk memudahkan pengguna dalam mendapatkan informasi yang mereka perlukan (Jantce TJ Sitinjak et al., 2020).

2.9. React Native

Firebase adalah *database* Bukan Hanya SQL (*NoSQL*). Menggunakan *Firebase* dapat menyimpan dan menyinkronkan data antar pengguna secara *real-time* sehingga *Firebase* dapat mempermudah pengembangan aplikasi dengan latar belakang waktu nyata. Data di *database firebase* disimpan sebagai objek pohon JSON. Kelebihan *Firebase* adalah dapat menyimpan data secara lokal saat perangkat berada tidak tersedia atau tidak terhubung ke akses internet, dapat segera disinkronkan setelah perangkatnya telah mendapat akses internet. dengan *Firebase*, kami dapat menyediakan layanan dalam mengembangkan aplikasi, salah satunya adalah aplikasi berbasis android. *Firebase* juga demikian sebuah layanan prima untuk membantu pengguna dalam mengembangkan aplikasi khususnya yang berbasis Android aplikasi. (Mohammad Daffa Ananda et al., 2023)

Firebase memiliki 2 layanan *database* yaitu *Cloud firestore* dan *Realtime database*. Sistem yang dirancang pada penelitian ini menggunakan layanan *Cloud firestore* oleh karena layanan tersebut merupakan layanan terbaru yang melanjutkan keberhasilan dari layanan *database Realtime database*. *Cloud firestore* juga memiliki fitur kueri yang lebih lengkap dan lebih cepat, serta penskalaan yang lebih mendalam dibandingkan dengan *Realtime database*. (Panjaitan & Pakpahan, 2021)

Cloud firestore menyimpan data pada sebuah dokumen yang berisi pemetaan *field* dengan nilai. Kemudian dokumen tersebut akan disimpan pada sebuah *collection*. Melalui *collection* ini kita dapat mengatur data dan membangun kueri. Sistem yang dibangun memiliki 4 *collection* diantaranya, *users* untuk menyimpan data *user*, *monthlyexpense* untuk menyimpan data laporan bulanan,

medicalexpense untuk menyimpan data laporan medis, dan *notifications* untuk menyimpan data notifikasi. Setiap tabel yang disampaikan pada penjelasan berikut ini merupakan field yang ada pada setiap collection beserta tipe datanya. (Panjaitan & Pakpahan, 2021)

Tabel 2.3 Contoh Isi Data Pengguna

Nama Field	Tipe Field
Id	String
Address	String
Email	String
Name	String
Phone	String
photoUrl	String
Role	String
totalMedicalBudget	Number
totalMonthlyBudget	Number

2.10. Figma

Figma adalah salah satu alat desain yang sering digunakan untuk membuat tampilan aplikasi mobile, desktop, website, dan lainnya. Figma dapat diakses melalui sistem operasi Windows, Linux, atau Mac dengan koneksi internet. Alat ini biasanya populer di kalangan profesional yang bekerja di bidang UI/UX, desain web, dan bidang terkait lainnya. UI dan UX merupakan singkatan dari *User Interface* dan *User Experience*, yang merujuk pada tampilan *visual* dalam aplikasi atau situs *web* yang bertujuan untuk meningkatkan pengalaman pengguna dan

memperkuat merek bisnis atau perusahaan. Di era 5.0, penggunaan aplikasi seperti Figma dalam pembuatan konten interaktif sangat penting karena sangat relevan dengan perkembangan dunia digital saat ini (Suparman et al., 2023).



Gambar 2.6 Logo Figma

(Sumber : <https://www.pngwing.com/id/search?q=logo+figma>)

2.11. Visual Studio Code

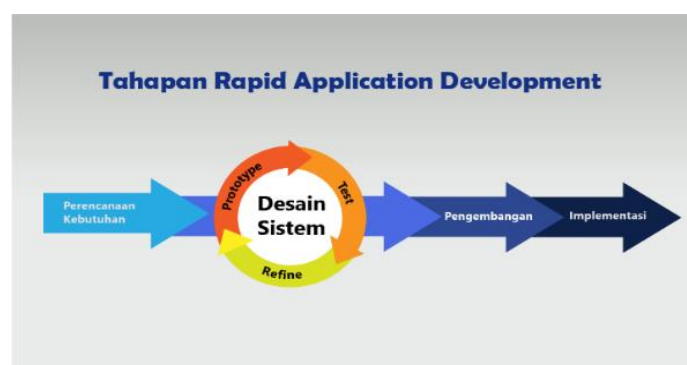
Visual Studio Code merupakan sebuah aplikasi *editor code open source* yang dikembangkan oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOS. *Visual Code* memudahkan dalam penulisan *code* yang mendukung beberapa jenis pemrograman, seperti C++, C#, Java, Python, PHP, GO. *Visual Code* memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkaian *code* tersebut. *Visual Studio Code* juga telah terintegrasi ke Github. Selain itu fitur lainnya adalah kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah ekstensi untuk menambah fitur yang tidak ada di *Visual Studio Code* (Ramdhan & Nufriana, 2019).

Banyak sekali fitur-fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense*, *Git Integration*, *Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur-fitur tersebut akan terus bertambah

seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *Visual Studio Code* dengan teks editor-teks editor yang lain. Teks editor *Visual Studio Code* juga bersifat *open source*, yang mana kode sumbernya dapat kalian lihat dan kalian dapat berkontribusi untuk pengembangannya. Kode sumber dari *Visual Studio Code* ini pun dapat dilihat di link Github. Hal ini juga yang membuat *Visual Studio Code* menjadi favorit para pengembang aplikasi, karena para pengembang aplikasi bisa ikut serta dalam proses pengembangan *Visual Studio Code* ke depannya (Pinaría et al., 2021).

2.12. Rapid Application Development (RAD)

Rapid Application Development (RAD) merupakan salah satu pendekatan dalam pengembangan perangkat lunak yang memiliki kemampuan untuk mempercepat waktu pengembangan. RAD terdiri dari serangkaian proses yang meliputi perancangan, desain sistem, dan implementasi, yang dirancang untuk meningkatkan efisiensi dalam pengembangan aplikasi (Wijaya, 2021).



Gambar 2.7 Tahapan RAD

(Sumber : <https://agus-hermanto.com/blog/detail/metode-pengembangan-rad-rapid-application-development>)

Planning merupakan tahap perencanaan paling mendasar yang dilakukan untuk memahami mengapa sistem informasi itu harus dibangun dan bagaimana cara untuk menentukan tim dalam mengerjakan proyek yang akan di tuntaskan

Analysis merupakan tahap melakukan pengumpulan data yang akan digunakan pada perancangan aplikasi. Data yang dikumpulkan berupa sumber data primer dan sekunder dilakukan dengan cara melakukan studi pustaka, wawancara dan kuesioner. Selanjutnya melakukan Analisa terhadap data yang telah didapat, mengidentifikasi perancangan aplikasi, serta melakukan pengembangan konsep untuk sistem yang baru.

Penelitian yang dilakukan untuk merancang sistem diperoleh dari pengamatan data-data yang telah didapat. Perancangan sistem di sesuaikan dengan data yang ada dan mengimplementasikan model yang diinginkan oleh pengguna. Pemodelan sistem ini berupa pembuatan *use case diagram*, *activity diagram*, dan *sequence diagram* guna mempermudah dalam proses-proses selanjutnya

Tahap ini merupakan pembuatan *prototype system*, yaitu mendefinisikan sistem secara tertulis apa yang bisa dilakukan oleh sistem dan seluruh prosesnya. Tahap ini, sangat melibatkan peran dari pengguna dalam memberikan umpan balik untuk prototipe sistem yang telah dibuat. Jika prototipe dinilai sudah mampu mencukupi kebutuhan dan sesuai dengan keinginan dari pengguna, maka proses dapat dilanjutkan ke tahap *Implementation*. Namun jika tidak, maka proses dapat diulang kembali ke tahap *Analysis* dan *Design*. (Widiarta et al., 2021)

2.13. Black Box Testing

Black box testing adalah metode pengujian kualitas perangkat lunak yang difokuskan pada fungsionalitasnya tanpa memerhatikan struktur internalnya.

Tujuan dari *black box testing* adalah untuk mengidentifikasi kesalahan dalam fungsi perangkat lunak, antarmuka yang tidak benar, struktur data yang salah, performa yang buruk, kesalahan pada inisialisasi dan terminasi, serta memastikan bahwa setiap proses berjalan sesuai dengan kebutuhan yang diharapkan (Wijaya & Astuti, 2021).

Pengujian blackbox yang dilakukan pada penelitian ini mengacu pada aktivitas untuk memastikan bahwa unit program memenuhi kebutuhan yang disebutkan dalam spesifikasi dalam kata lain mengamati hasil eksekusi berdasarkan skenario test case. Pada penelitian ini menggunakan metode requirement test untuk melakukan pengujian blackbox (Putra et al., 2020).






Tahapan penelitian dimulai dengan melakukan studi pendahuluan untuk memperoleh informasi terkait penelitian yang akan dilakukan. Tahap kedua melakukan identifikasi sistem serta rencana pengujian sistem. Kemudian melakukan pengujian dengan menggunakan metode black box testing. Tahap ketiga yaitu pembuatan dokumentasi pengujian pada SIMDOS. Tahap selanjutnya yaitu analisis sistem dari pengujian yang sudah dilakukan sebelumnya. Tahap terakhir yaitu memberikan rekomendasi pengujian (Made Dwi Febriyanti et al., 2021).


2.14. Flowchart

Flowchart adalah representasi grafis dari langkah-langkah dan urutan prosedur dari suatu program. Ini membantu analis dan programmer dalam memecahkan masalah menjadi segmen-segmen yang lebih kecil dan mempertimbangkan alternatif-alternatif dalam operasinya. *Flowchart* seringkali memudahkan penyelesaian masalah, terutama masalah yang memerlukan studi dan evaluasi lebih lanjut. *Flowchart* merupakan gambar atau diagram dengan aliran

sekuensial satu atau dua arah. Digunakan untuk merepresentasikan dan merancang program, sehingga harus mampu mencakup komponen-komponen dalam bahasa pemrograman (Sutanti et al., 2020).

Tabel 2.4 Simbol Flowchart

No	Simbol	Nama Simbol	Keterangan
1		<i>Process</i>	Simbol ini untuk Menunjukkan suatu aktivitas yang akan divisualisasikan dalam alir.
2		<i>Terminal</i>	Simbol ini untuk menunjukkan suatu awalan atau akhiran dalam suatu proses.
3		<i>Flowline</i>	Simbol ini untuk menunjukkan arus dari suatu proses.
4		<i>Decision</i>	Simbol ini berfungsi sebagai suatu penentuan dalam langkah selanjutnya suatu proses. Di proses ini terdapat dua keluaran dalam meneruskan alur pada situasi yang berbeda.
5		<i>Input/Output</i>	Simbol ini berfungsi sebagai perwakilan dalam data yang akan dimasukkan atau data yang akan dikeluarkan.
6		<i>Control/</i>	Simbol ini untuk menentukan suatu langkah atau proses yang akan di

No	Simbol	Nama Simbol	Keterangan
		<i>Inspection</i>	inspeksi atau di kontrol.

Flowchart terdapat 5 (lima) jenis bagian, yakni :

1. *Process Flowchart*

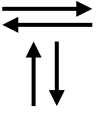
Process Flowchart adalah teknik *visual* dalam rekayasa industri yang digunakan untuk memecah dan menganalisis langkah-langkah berikutnya dalam suatu prosedur atau sistem. Ini adalah alat yang berguna bagi insinyur industri dalam mempelajari dan mengembangkan proses manufaktur, serta untuk menjelajahi alur laporan atau formulir dalam menganalisis sistem.




Di bawah ini adalah notasi atau simbol yang akan digunakan, terbagi menjadi 3 (tiga) kelompok :

a. Simbol Penghubung alur (*Flow Direction Symbols*)

Simbol yang dipakai dalam mengkoneksikan hubungan satu simbol dengan simbol yang lain atau bisa disebut dengan *connecting line*, Berikut adalah simbolnya :

Tabel 2.5 Simbol Penghubung Alur


No	Simbol	Nama Simbol	Keterangan
1		<i>Flow</i>	Simbol ini untuk menggambarkan bagaimana arus suatu proses berjalan
2		<i>Connector</i>	Simbol ini untuk menunjukkan bagaimana proses satu ke proses lainnya terhubung dalam



No	Simbol	Nama Simbol	Keterangan
			halaman atau lembaran yang sama
3		<i>Comunnication Link</i>	Simbol ini untuk menunjukkan bahwa data / informasi bergerak dari satu tempat ke tempat lain.
4		<i>Offline Connector</i>	Simbol ini ntuk menunjukkan bagaimana proses satusama lain terhubung dalam halaman atau lembaran yang berbeda.

b. Simbol Masukkan – Keluaran (*Input – Output Symbols*)

Simbol yang digunakan untuk memperlihatkan jenis komponen apa saja yang dipakai sebagai media saat melakukan masukkan (*input*) atau keluaran (*output*). Berikut adalah simbol – simbol *input/output* :

Tabel 2.6 Simbol Input – Output



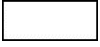
No	Simbol	Nama Simbol	Keterangan
1		<i>Disk Storage</i>	Untuk menunjukkan input dari disk atau output yang disimpan ke disk.
2		<i>Input / Output</i>	Untuk menunjukkan Proses input dan output tanpa



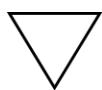
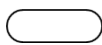
			memperhatikan jenis peralatan.
3		Document	Untuk mencetak pada berkas atau dokumen yang ada.

c. Simbol Proses (*Processing Symbols*)

Simbol ini digunakan untuk memperlihatkan apa saja jenis operasi pengolahan yang terdapat pada suatu prosedur atau pada suatu proses. Berikut adalah simbol – simbol proses :

Tabel 2.7 Simbol Proses

No	Simbol	Nama Simbol	Keterangan
1		<i>Decision</i>	Simbol ini untuk menggambarkan suatu kondisi tertentu dengan pilihan opsi YA atau TIDAK.
2		<i>Predefined Process</i>	Simbol ini untuk menyatakan lokasi penyimpanan suatu pengolahan untuk harga awal.
3		<i>Process</i>	Simbol ini untuk melakukan proses yang telah dilakukan komputer yang pada umumnya telah berhasil merubah data atau informasi.
			Simbol ini untuk

No	Simbol	Nama Simbol	Keterangan
4		<i>Manual Input</i>	menambahkan data secara manual dengan memakai <i>keyword online</i> .
5		<i>Manual Symbol</i>	Simbol ini untuk menunjukkan suatu aksi atau proses yang tidak dilakukan oleh komputer secara manual.
6		<i>Offline Storage</i>	Simbol ini untuk menunjukkan bahwa data akan disimpan ke media tertentu.
7		<i>Terminal</i>	Simbol ini digunakan untuk menunjukkan awal atau akhir suatu program

1. *System Flowchart*

Flowchart sistem adalah representasi grafis dari serangkaian prosedur yang saling terhubung yang membentuk suatu sistem. Ini mengilustrasikan secara menyeluruh alur kerja sistem dan menjelaskan urutan langkah-langkah yang terjadi di dalamnya. *Flowchart* sistem menggambarkan bagaimana data mengalir melalui sistem dan menggambarkan proses yang terjadi pada data tersebut. *Flowchart* sistem dapat dibuat secara digital (menggunakan komputer) atau secara manual (menggunakan mesin tik, kertas, atau kalkulator).

2. *Schematic Flowchart*

Bagan alir skematik, juga disebut bagan alir sistem, menggambarkan prosedur

sistem. Namun, bagan alir skematik menggunakan gambar komputer dan peralatan lainnya, selain simbol bagan alir sistem.

3. *Document Flowchart*

Bagan alir dokumen (*document flowchart*), juga dikenal sebagai bagan alir formulir (*form flowchart*), dan bagan alir kertas (*paperwork flowchart*) adalah bagan alir yang menunjukkan arus dari laporan dan formulir termasuk tembusan-tembusan. Simbol yang digunakan dalam bagan alir sistem juga digunakan dalam bagan alir dokumen ini.

4. *Program Flowchart*

Bagan alir program terbagi menjadi dua jenis, yaitu bagan alir logika program (*program logic flowchart*) dan bagan alir program komputer terinci (*computer program flowchart*). Bagan alir logika program digunakan untuk merinci setiap langkah dari proses program. Sedangkan, bagan alir program komputer terinci dibuat berdasarkan bagan alir sistem. Tugas analisis sistem adalah menyusun bagan alir logika program ini.

2.15. UML (Unified Modelling Language)

UML adalah bahasa visual yang umum digunakan untuk merancang, memodelkan, dan mendokumentasikan berbagai jenis sistem, baik perangkat lunak maupun non-perangkat lunak. Dikembangkan oleh *Object Management Group* (OMG), UML menyediakan serangkaian diagram terintegrasi yang membantu pengembang dalam menspesifikasikan, memvisualisasikan, dan membangun komponen sistem secara efektif. Tujuannya adalah untuk menjadi notasi standar yang dapat digunakan oleh berbagai metode berorientasi objek, serta memberikan kerangka kerja untuk berbagai aktivitas sistem seperti analisis, desain, penyebaran,

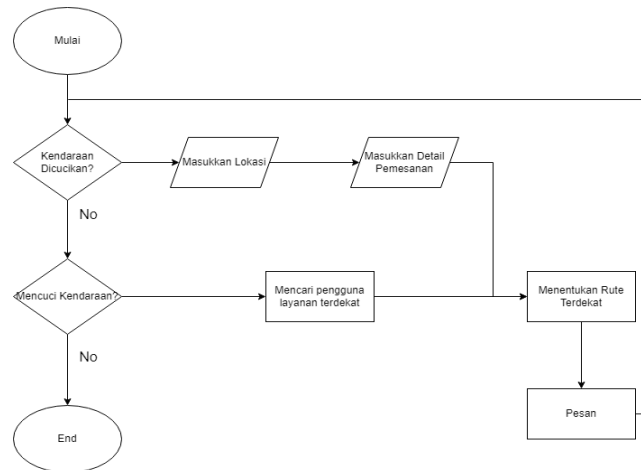
dan distribusi system (Pinayungan, 2019).

2.16. Kerangka Berpikir Konseptual

Kerangka berpikir atau kerangka pemikiran adalah dasar pemikiran dari penelitian yang disintesis dari fakta-fakta, observasi dan kajian kepustakaan. Oleh karena itu, kerangka berpikir memuat teori, dalil atau konsep-konsep yang akan dijadikan dasar dalam penelitian. Di dalam kerangka pemikiran variabel-variabel penelitian dijelaskan secara mendalam dan relevan dengan permasalahan yang diteliti, sehingga dapat dijadikan dasar untuk menjawab permasalahan penelitian.

Kerangka berpikir merupakan perlengkapan peneliti untuk menganalisa perencanaan dan berargumentasi kecenderungan asumsi ke mana akan dilabuhkan, penelitian kuantitatif kecenderungan akhirnya adalah diterima atau ditolak hipotesis penelitian tersebut, sedangkan penelitian yang berbentuk pernyataan atau narasi-narasi peneliti bertolak dari data dan memanfaatkan teori yang digunakan sebagai bahan penjelasan dan berakhir dengan pembaharuan suatu pernyataan atau hipotesa.

Penyusunan kerangka berpikir mesti berdasarkan narasi, dapat menghasilkan kesimpulan yang ditanggung jawabkan. Berdasarkan kesimpulan ini, kita dapat membuat atau menyusun jawaban sementara untuk memecahkan masalah berdasarkan kerangka berpikir yang telah dibuat maka dapat ditarik kesimpulan, jika motivasi kerja karyawan tinggi maka kinerjanya akan meningkat. (Syahputri et al., 2023)



Gambar 2.8 Kerangka Berpikir Konseptual

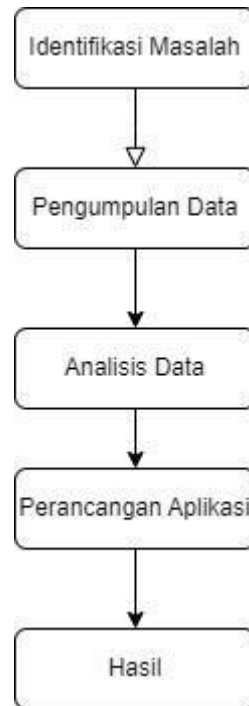
Berdasarkan kerangka berpikir diatas, dapat disimpulkan bahwa apabila para pengguna ingin menggunakan layanan Doorsmeer Online, pengguna harus membuka aplikasi Doorsmeer Online, lalu memasukkan lokasi agar dapat dijumpai oleh pekerja dan memasukkan detail pemesanan, seperti nomor handphone, dan juga detail dari lokasi agar pekerja dapat lebih mudah menemukan pengguna layanan dan dapat memulai pekerjaannya.

Sedangkan untuk pekerja, dibutuhkan untuk bersiaga apabila adanya pesanan yang masuk dari pengguna. Pekerja juga dapat menerima dan menolak datangnya pesanan dari pelanggan apabila sedang tidak ingin bekerja. Apabila pekerja menerima, pekerja akan mendapatkan detail pemesanan dan lokasi dari pengguna, dan sistem akan mencari rute tersingkat agar dapat memangkas waktu yang dibutuhkan untuk mencapai lokasi pengguna.

BAB III

LANDASAN TEORI

3.1. Desain Penelitian



Gambar 3.1 Desain Penelitian

1. Identifikasi Masalah

Masalah utama yang dihadapi dalam penelitian ini adalah kurangnya keinginan para pengguna kendaraan untuk membawa kendaraannya untuk dicuci, dikarenakan adanya kesibukan atau kekurangan waktu. Proses membawa kendaraan ke tempat doorsmeer masih melibatkan waktu sang pemilik untuk membawa kendaraannya ke tempat doorsmeer tersebut. Tidak hanya karena kurangnya waktu, tetapi jalan pulang sesudah mencuci kendaraan tersebut juga masih ada resiko yang dapat menyebabkan kendaraan tersebut kotor lagi.

2. Pengumpulan Data

Peneliti melakukan studi literatur dan observasi terhadap beberapa tempat doorsmeer dengan tujuan untuk memperoleh pemahaman yang dalam mengenai informasi yang diperlukan oleh sistem dalam merancang aplikasi sebagai sarana informasi bagi pengguna doorsmeer online.

3. Analisis Data

Melakukan analisis terhadap data yang diperoleh dari observasi guna menggambarkan desain serta merancang aplikasi doorsmeer online.

4. Perancangan Aplikasi

Proses perancangan aplikasi dimulai dengan mengembangkan pengalaman pengguna (UX) dan antarmuka pengguna (UI) menggunakan Figma, diikutidengan implementasi kode menggunakan React Native.

5. Hasil

Hasil dari penelitian ini adalah pengembangan sebuah aplikasi yang mempermudah proses pencucian kendaraan di Kota Medan. Seluruh tahapan darihasil akhir penelitian ini akan diuraikan secara menyeluruh dalam Bab IV.

3.2. Metode Pengembangan Aplikasi

3.2.1 Rapid Application Development (RAD)

Rapid Application Development (RAD) menekankan pada pengembangan dalam waktu singkat dan dengan fokus pada kualitas yang tinggi. Model RAD cocok digunakan untuk proyek pengembangan yang memiliki cakupan yang terbatas dan dikerjakan oleh tim kecil. Penelitian ini mengadopsi model RAD dan memiliki beberapa tahapan pengembangan yang akan diuraikan sebagai berikut:

1. *Requirement Planning* (Rencana Kebutuhan)

Pada fase ini, peneliti mengidentifikasi masalah yang timbul dan merancang rencana untuk mengatasi masalah tersebut dengan lebih efektif. Perangkat yang diperlukan peneliti untuk membuat Aplikasi antara lain :

a. *Hardware* (perangkat keras) :

1. *Smartphone* :

- *Processor* : *MediaTek Dimensity 1080*
- *Memory* : *8 GB*
- *Internal* : *256 GB*
- *Operating System* : *Android 13 (Tiramisu)*

2. *Laptop* :

- *Processor* : *Intel Core i5 12500H*
- *Memory* : *8 GB*
- *SSD* : *512 GB*

b. *Software* (perangkat lunak) :

- *Platform* : *Microsoft*
- *Operating System* : *Windows 11*
- *Bahasa Pemograman* : *React Native*
- *Tools* : *Visual Studio Code, Figma , Draw.io*
- *DBMS* : *Firebase Realtime Database*

c. *Pembuatan Aplikasi*

Pada fase ini dilakukan pembuatan aplikasi berdasarkan *requirement planning* yang telah dilakukan sebelumnya.

d. Pengujian dan *Maintenance*

Pada fase ini ,dilakukan pengujian dari aplikasi yang telah dibuat, hal tersebut dilakukan untuk mengetahui fungsi – fungsi yang ada pada aplikasi berjalan dengan lancar atau tidak. Kemudian dilakukan maintenance apabila terdapat error atau malfungsi pada aplikasi.

3.3. Teknik Pengumpulan Data

Pengumpulan data merupakan metode untuk mengumpulkan informasi yang menjadi panduan bagi peneliti dalam menjalankan penelitian. Cara untuk memperoleh dan mengumpulkan data tersebut melibatkan proses seperti observasi dan studi literatur. Berikut adalah langkah-langkah yang dilakukan oleh peneliti dalam pengumpulan data :

1. Studi Literatur

Dalam penelitian ini, kegiatan studi literatur mencakup pencarian, pemahaman, dan penguraian berbagai referensi yang terdapat dalam jurnal dan buku yang relevan dengan topik penelitian.

2. Observasi (Pengamatan)

Pada tahap ini, dilakukan pengamatan langsung terhadap objek penelitian dengan mencatat secara terstruktur semua informasi yang diperoleh untuk mengumpulkan data. Proses observasi penelitian ini dilakukan dengan mengamati secara langsung lokasi penelitian, dimulai dengan observasi menyeluruh dan dilanjutkan dengan fokus yang lebih mendalam.

3.4. Pengguna Sistem

1. Pemberi Layanan

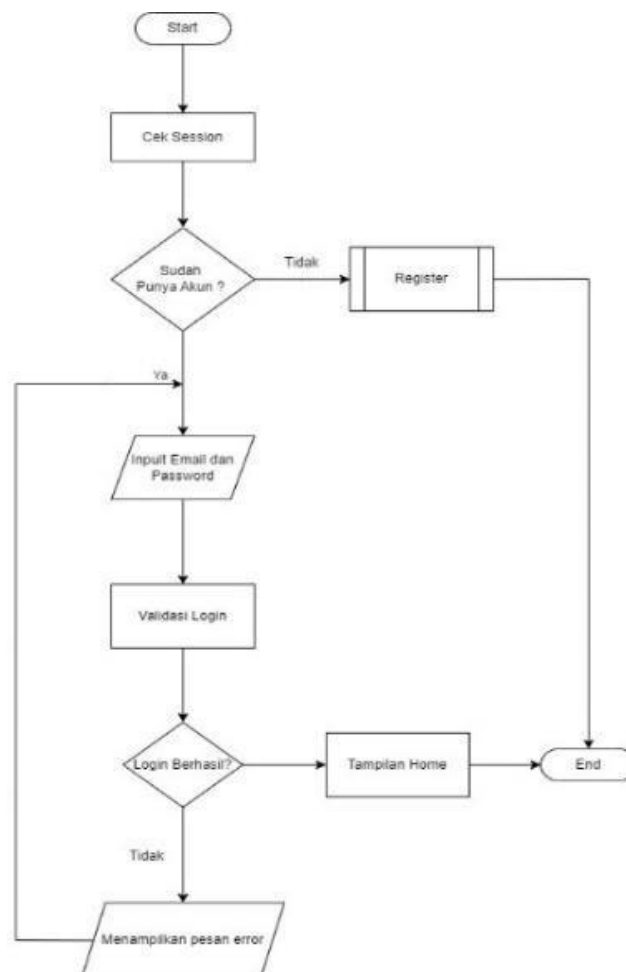
Pemberi Layanan adalah seorang yang memberi pengguna layanan yang diberikan sesuai dengan fitur yang ada di aplikasi.

2. Pengguna Layanan

Pengguna Layanan adalah seorang yang menggunakan layanan yang diberikan sesuai dari fitur aplikasi, yang akan dilayani oleh pemberi layanan.

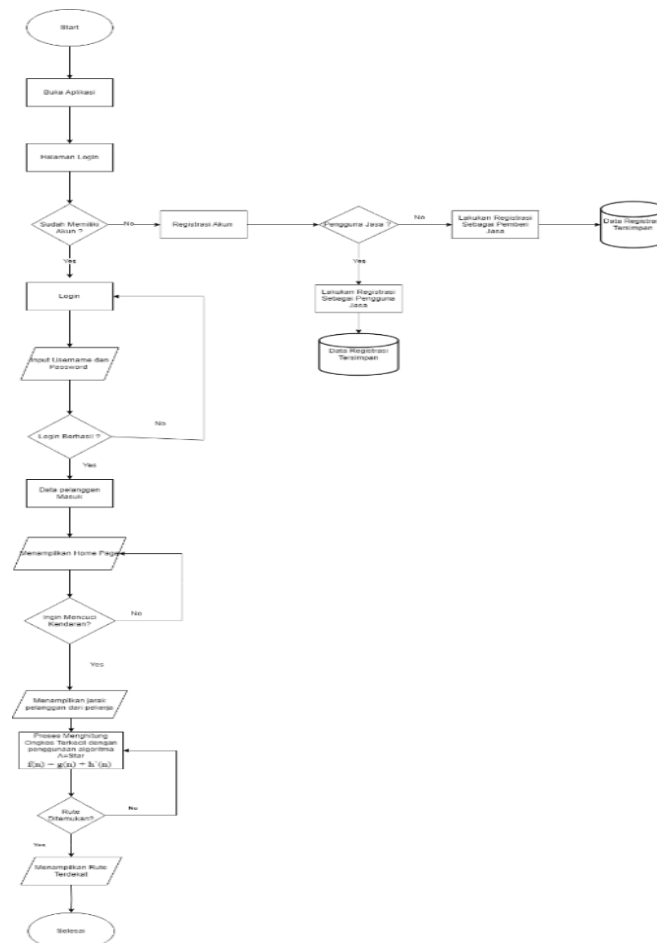
3.5. Flowchart

Flowchart ini menggambarkan pengguna yang akan dihadapkan dengan formulir yang terdiri dari *username* atau email dan *password*.



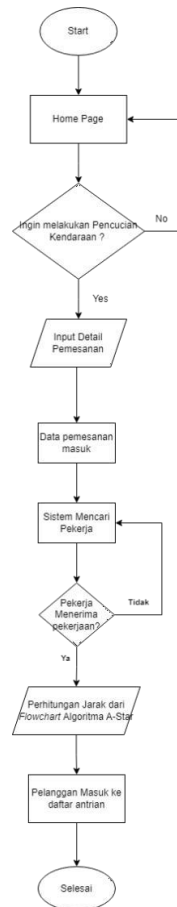
Gambar 3.2 *Flowchart* Aplikasi Doorsmeer Online

Berdasarkan flowchart diatas, dimulai dari membuka aplikasi, sistem akan memunculkan *splash screen* dan meminta user untuk login atau daftar. Apabila user memilih untuk mendaftar, user akan dibawa ke halaman pendaftaran. Jika user sudah mendaftar dengan sukses, user akan dibawa Kembali ke halaman login. Apabila user gagal login, user akan dibawa Kembali ke halaman login dan memasukkan email dan *password* yang benar. Apabila login sukses, user akan dibawa ke halaman home sesuai dengan peran yang diambil, yaitu pengguna layanan dan pekerja dan dapat menggunakan fitur sesuai dengan peran yang diambil.



Gambar 3.3 Flowchart Algoritma A-Star

Berdasarkan flowchart diatas, Sesudah user membuka aplikasi Doorsmeer Online, user akan disambut dengan halaman login, apabila belum memiliki akun, user dapat memilih daftar dan akan dibawa ke halaman daftar. Di halaman daftar tersebut, user dapat memilih antara menjadi pemberi jasa atau pengguna jasa. Sesudah mendaftar, baik menjadi pemberi atau pengguna jasa, kredensial dari user tersebut akan disimpan di database. Sesudah mendaftar, user akan dikembalikan ke halaman login. Apabila kredensial login dari user tidak cocok dengan yang ada di database, user akan diminta untuk memasukkannya Kembali. Apabila login berhasil, user akan dibawa ke halaman utama dan diminta untuk memilih fitur apa yang ingin digunakan sesuai dengan peran yang diambil. Apabila user ingin kendaraannya dicucikan, user harus mengisi detail dari pemesanan, seperti nomor handphone, dan catatan agar pekerja dapat menemukan pengguna dengan mudah. Begitu pekerja mendapatkan pesanan dari pengguna tadi, pekerja akan mendapatkan lokasi yang sudah dihitung menggunakan rumus algoritma A – Star agar dapat mencapai lokasi pengguna dengan cepat, dan tepat dikarenakan adanya catatan yang diberikan oleh pengguna sebelumnya.



Gambar 3.4 Flowchart Algoritma Priority Scheduling

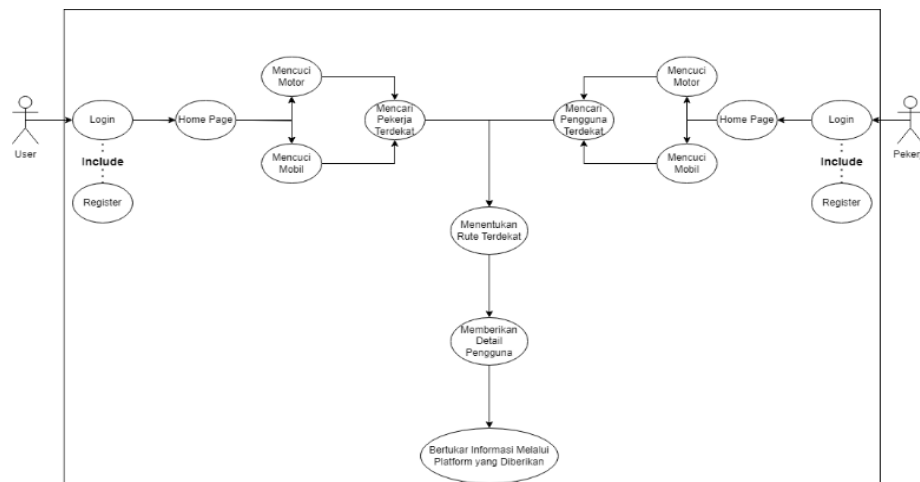
Berdasarkan flowchart diatas, diambil dari narasi flowchart sebelumnya, sesudah pekerja mendapatkan data pemesanan dari pengguna jasa, algoritma A – Star akan mencari rute tersingkat, tetapi apabila pekerja masih memiliki pekerjaan yang belum selesai sebelumnya, pengguna tersebut akan masuk kedalam daftar antrian dan akan dimulai dikerjakan sesudah pekerja menyelesaikan pekerjaan sebelumnya.

3.6. UML (Unified Modelling Language)

3.6.1 Use Case Diagram

Use Case Diagram adalah diagram yang memvisualisasikan interaksi antara

pengguna (aktor) dan sistem melalui sebuah narasi untuk menjelaskan cara penggunaan sistem tersebut. Diagram ini terdiri dari aktor yang mewakili entitas yang berinteraksi dengan sistem, yang bisa berupa manusia, perangkat keras, atau sistem lainnya. Ini biasanya disebut sebagai diagram *use case*, di mana setiap *use case* menggambarkan sebuah interaksi spesifik antara aktor dan sistem. Berikut adalah *Use case diagram* dari Aplikasi Doorsmeer Online :



Gambar 3.5 Use Case Aplikasi Doorsmeer Online

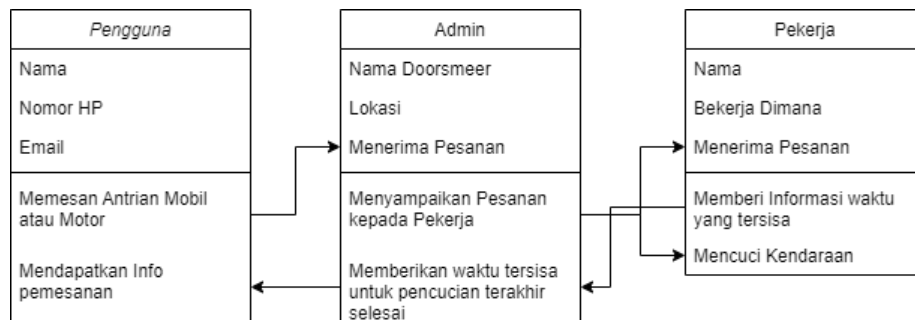
Dibawah ini adalah tabel penjelasan disetiap aktor pada *use case diagram* diatas :

Tabel 3.1. Defenisi Aktor

User (Pengguna Layanan)	User ialah orang yang mempunyai hak akses setelah melakukan registrasi dan melakukan login. Pengguna dapat memesan layanan pencucian Kendaraan dengan Pekerja melalui platform yang diinginkan, seperti Whatsapp, Telegram, Dll.
Pekerja	Pekerja ialah orang yang mempunyai hak akses untuk mendapatkan detail dari pengguna yang ingin menggunakan layanan mencuci kendaraan dan mendapatkan rute terdekat dari sang pengguna untuk mulai mencuci kendaraan.

3.6.2 Class Diagram

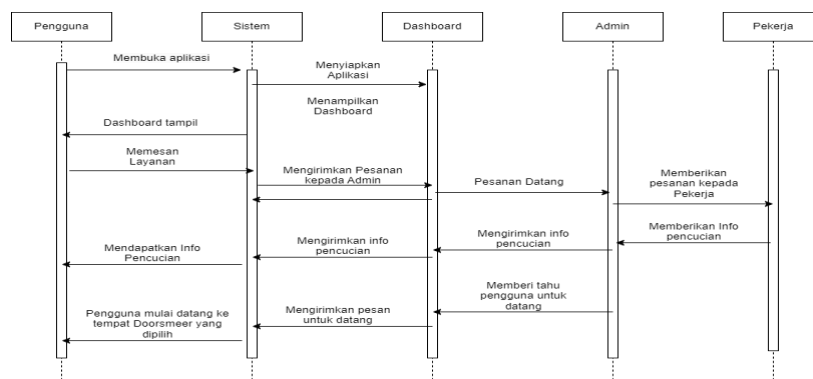
Class diagram atau diagram kelas adalah salah satu jenis diagram struktur pada UML yang menggambarkan dengan jelas struktur serta deskripsi class, atribut, metode, dan hubungan dari setiap objek. Ia bersifat statis, dalam artian diagram kelas bukan menjelaskan apa yang terjadi jika kelas-kelasnya berhubungan, melainkan menjelaskan hubungan apa yang terjadi.



Gambar 3.6 Class Diagram Aplikasi Doorsmeer Online

3.6.3 Sequence Diagram

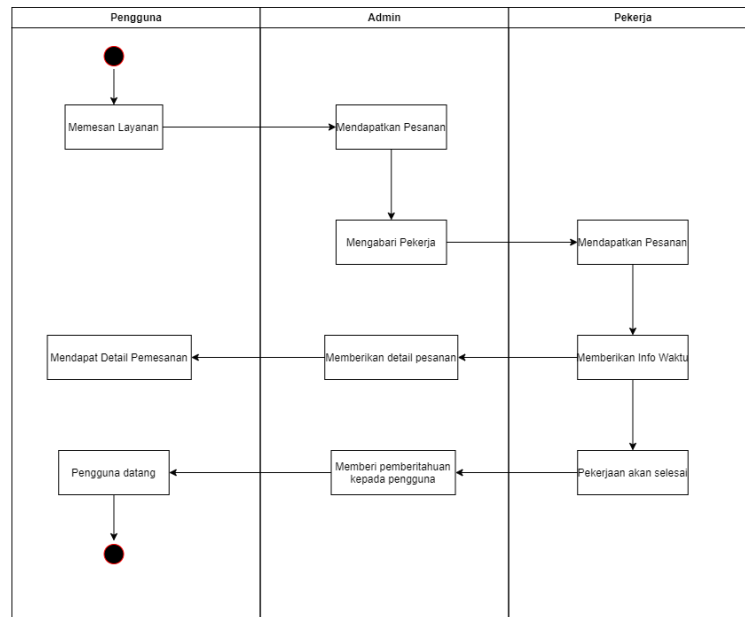
Sequence diagram atau diagram urutan adalah sebuah diagram yang digunakan untuk menjelaskan dan menampilkan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Selain itu sequence diagram juga akan menampilkan pesan atau perintah yang dikirim, beserta waktu pelaksanaannya



Gambar 3.7 Sequence Diagram Aplikasi Doorsmeer Online

3.6.4 Activity Diagram

Diagram aktivitas adalah bentuk visual dari alir kerja yang berisi aktivitas dan tindakan, yang juga dapat berisi pilihan, pengulangan, dan concurrency. Dalam Unified Modeling Language, diagram aktivitas dibuat untuk menjelaskan aktivitas komputer maupun alur aktivitas dalam organisasi.



Gambar 3.8 Activity Diagram Aplikasi Doorsmeer Online

3.7. Wireframe

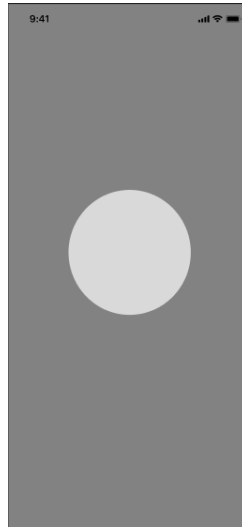
Wireframe merupakan struktur dasar yang digunakan untuk mengatur komponen-komponen dalam suatu halaman aplikasi. Komponen-komponen tersebut meliputi *banner*, *header*, konten, *footer*, tautan, formulir, dan elemen-elemen lainnya..

Dalam Pembuatan *Wireframe* ini peneliti menggunakan Figma sebagai media desain. Berikut adalah *Wireframe* dari Aplikasi Doorsmeer Online :

1. Tampilan Splash Screen

Fungsinya adalah untuk memberikan pengguna suatu indikasi bahwa aplikasi

sedang memuat atau akan segera memulai. Selain itu, splash screen juga akan digunakan untuk menampilkan logo dari aplikasi yang akan dibuat.

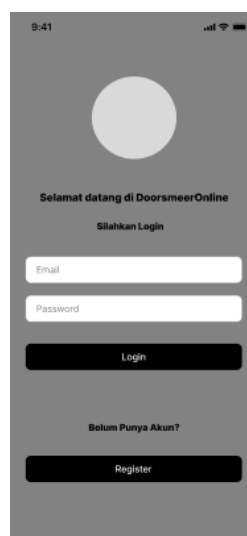


Gambar 3.6 Tampilan Splash Screen

2. Tampilan Login

Tampilan login berfungsi untuk memberikan hak akses untuk menjalankan aplikasi dengan cara memasukkan email dan password.

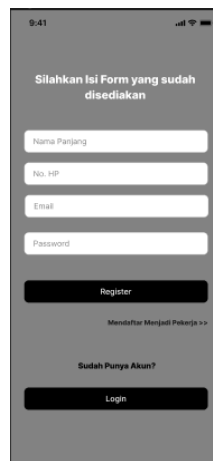
Pada halaman ini juga user dapat memilih untuk melakukan registrasi jika belum memiliki akun yang terdaftar.



Gambar 3.7 Halaman Login

3. Tampilan Registrasi

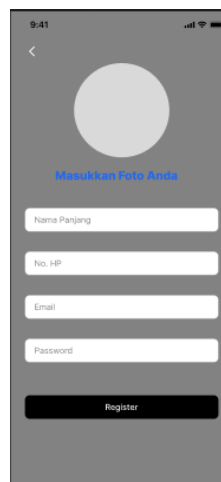
Pada Halaman ini user dapat melakukan pendaftaran akun dengan memasukkan nama, email, nomor telephone, dan password. Setelah melakukan pendaftaran user dapat mencoba untuk login dengan akun yang sudah didaftarkan.

A screenshot of a mobile application registration screen. At the top, the time is 9:41. The screen displays the instruction "Silahkan Isi Form yang sudah disediakan" (Please fill in the provided form). Below this are four input fields: "Nama Panjang" (Full Name), "No. HP" (Phone Number), "Email", and "Password". A black "Register" button is positioned below the fields. Underneath the button is the text "Mendaftar Menjadi Pekerja >>" (Register as Employee >>). At the bottom, there is a section titled "Sudah Punya Akun?" (Already have an account?) with a black "Login" button.

Gambar 3.8 Tampilan Registrasi

4. Tampilan Registrasi Sebagai Pekerja

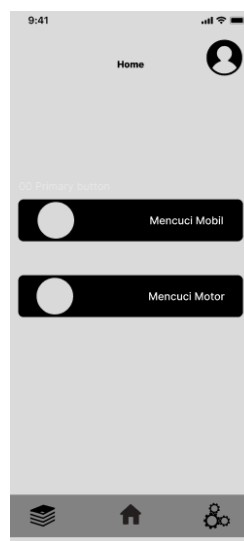
Dihalaman ini, kurang lebih hampir sama dengan registrasi sebagai pengguna layanan, yang membedakan adalah adanya kewajiban untuk mengisi foto profil agar mudah ditandai oleh pengguna layanan nantinya, dan juga fitur yang dapat diakses oleh pengguna layanan dan pekerja sudah pasti berbeda.

A screenshot of a mobile application registration screen for an employee. At the top, the time is 9:41. The screen features a large grey circle placeholder for a profile picture, with the text "Masukkan Foto Anda" (Add your photo) below it. Below the photo placeholder are four input fields: "Nama Panjang" (Full Name), "No. HP" (Phone Number), "Email", and "Password". A black "Register" button is located at the bottom of the form.

Gambar 3.9 Tampilan Registrasi Sebagai Pekerja

5. Tampilan Homepage

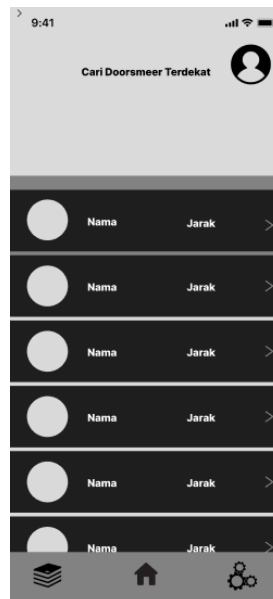
Halaman ini merupakan tampilan awal dari aplikasi ini yang akan ditampilkan saat sudah melakukan login sebagai pengguna layanan. Pada halaman home terdapat lokasi yang sudah ditentukan oleh GPS. Tetapi, apabila tidak akurat, pengguna dapat mereshuffle GPS, atau bisa mengisi lokasi tersebut dengan manual. Pada halaman ini juga terdapat fitur untuk memesan layanan untuk mencuci kendaraan, baik mobil ataupun motor.



Gambar 3.10 Tampilan Homepage

6. Tampilan Mencari Pekerja

Pada halaman ini, kita dapat memilih pekerja terdekat, berbeda dengan yang ada di halaman utama yang mencari pekerja dengan otomatis, disini kita bisa mencari pekerja sesuai dengan keyakinan pengguna, misalkan sudah pernah menggunakan layanannya, dan ingin menggunakan layanannya lagi.



Gambar 3.11 Tampilan Cari Pekerja

BAB IV

HASIL DAN PEMBAHASAN

Aplikasi Doorsmeer Online berbasis android didesain untuk di implementasikan untuk masyarakat umum dengan tujuan untuk mempermudah masyarakat dalam mencari tempat pencucian kendaraan terdekat di Kota Medan. Pengguna yang ditargetkan untuk menggunakan aplikasi ini dibedakan menjadi 2 role yaitu :

1. *Owner* : *Role* ini didesain untuk digunakan oleh pemilik usaha doorsmeer yang ingin mendaftarkan doorsmeer nya pada aplikasi ini. *Owner* selaku pemilik doorsmeer akan dapat menerima orderan yang dilakukan oleh *user*.
2. *User* : *Role* ini didesain untuk digunakan oleh pengguna aplikasi yang memiliki kebutuhan untuk mencuci kendaraan yang dimilikinya. *User* dapat melakukan pencarian doorsmeer terdekat dan melakukan pemesanan untuk pencucian kendaraan sesuai kebutuhan user.

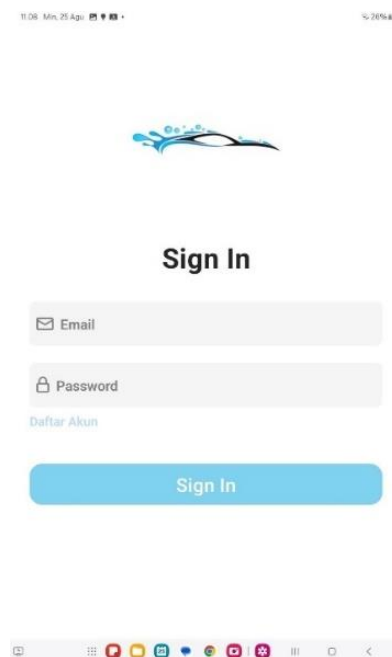
4.1. Implementasi Interface Aplikasi Berbasis Android

Aplikasi berbasis android diimplementasikan untuk user dan owner dalam memudahkan owner dalam mendaftarkan usaha doorsmeernya dan user dapat dengan mudah mencari informasi detail doorsmeer di sekitarnya dan melakukan pencucian kendaraan sesuai dengan kebutuhan dari user . Berikut adalah hasil implementasi aplikasi interface aplikasi berbasis android.

4.1.1. Implementasi Halaman Login User

Halaman login memungkinkan pengguna untuk masuk ke aplikasi dengan

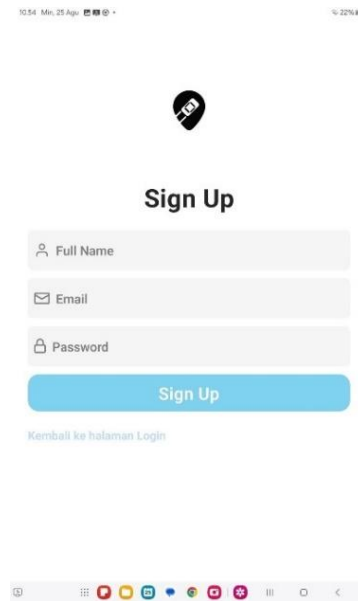
memasukkan email dan kata sandi yang sudah terdaftar. hanya pengguna terdaftar yang dapat mengakses fitur aplikasi. Setelah pengguna memasukkan kredensial mereka dan menekan tombol "Login", aplikasi akan memverifikasi informasi yang dimasukkan dan akan mengarahkan pengguna ke halaman utama jika kredensial tersebut valid. Jika pengguna belum memiliki akun, pengguna dapat memilih untuk melakukan daftar akun untuk mendapatkan kredensial yang valid.



Gambar 4.1. Tampilan Halaman Login untuk User

4.1.2. Implementasi Halaman Registrasi User

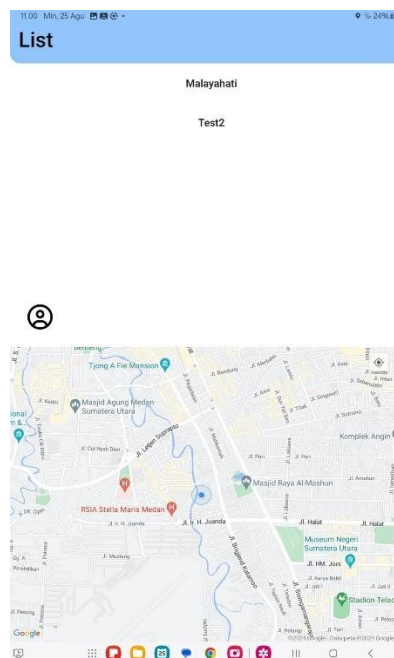
Halaman Registrasi ini adalah langkah awal dari user untuk mendapatkan kredensial yang diperlukan untuk mengakses aplikasi ini. Di halaman ini user diharuskan untuk mengisi kolom *fullname*, *email*, dan *password*. setelah mengisi kolom tersebut user dapat melakukan pendaftaran akun.



Gambar 4.2. Tampilan Halaman Registrasi untuk User

4.1.3. Implementasi Halaman Home

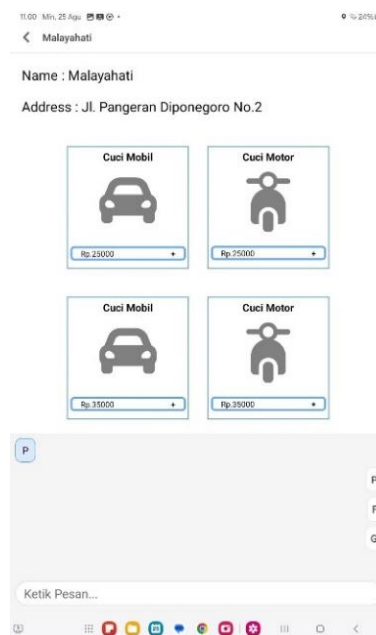
Setelah melakukan login maka halaman home akan tertampil, pada halaman ini user akan dapat masuk ke halaman order, profil , dan juga map.



Gambar 4.3. Tampilan Halaman Home

4.1.4. Implementasi Halaman Order

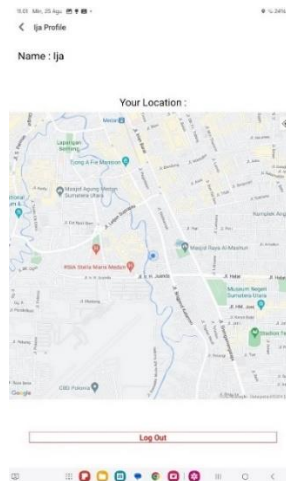
Pada Halaman Order user dapat melakukan order jasa pencucian kendaraan sesuai dengan kebutuhan user , user juga dapat mengetahui letak doorsmeer terdekat yang ada disekitarnya dan juga user dapat melakukan obrolan dengan pemberi layanan doorsmeer.



Gambar 4.4. Tampilan Halaman Order

4.1.5. Implementasi Halaman Profile

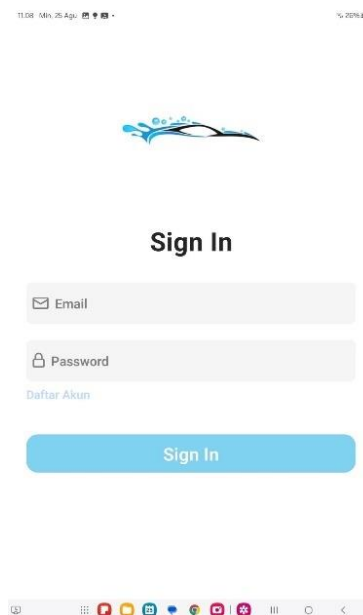
Halaman Profil menampilkan informasi nama akun, lokasi terkini , dan juga pada halaman ini user dapat melakukan logout.



Gambar 4.5. Tampilan Halaman Profil

4.1.6. Implementasi Halaman Login *Owner*

Halaman Login *Owner* dikhususkan untuk pemilik *doorsmeer* untuk masuk kedalam aplikasi dan menggunakan fitur-fitur aplikasi. Hanya *owner* yang terdaftar atau memiliki kredensial yang valid yang dapat menggunakan aplikasi ini. Jika belum memiliki akun *owner* dapat memilih untuk melakukan registrasi terlebih dahulu.



Gambar 4.6. Tampilan Halaman Login untuk Owner

4.1.7. Implementasi Halaman Registrasi *Owner*

Pada Halaman Registrasi ini , Owner diharuskan untuk mengisi kolom nama doormeer, alamat , lokasi doormeer, email dan password, dan harga jasa cuci kendaraan. Setelah mengisi kolom tersebut owner bisa melakukan registrasi dan mendapatkan kredensial untuk login kedalam aplikasi.

The screenshot shows a mobile application interface for 'Owner Sign Up'. At the top, there is an icon of a person standing next to a car. Below the icon, the title 'Owner Sign Up' is displayed. The form consists of several input fields arranged in two columns:

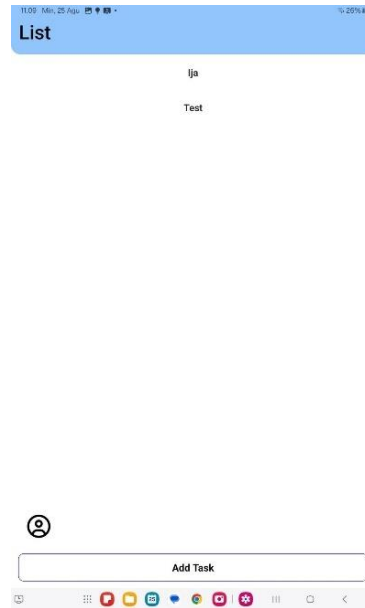
- Company Name (with a document icon)
- Address (with a location pin icon)
- Email (with an envelope icon)
- Password (with a lock icon)
- Harga Cuci Mobil (with a car icon)
- Harga Cuci Motor (with a motorcycle icon)
- Harga Cuci Mobil VIP (with a car icon)
- Harga Cuci Motor VIP (with a motorcycle icon)

Below the input fields, there is a small note: 'Masukkan Angka 0 apabila ada yang tidak digunakan'. A prominent blue button labeled 'Sign Up' is centered below the form. Underneath the button, there is a link that says 'Kembali ke halaman Login'. The bottom of the screenshot shows a mobile browser navigation bar with various icons and a back arrow.

Gambar 4.7. Tampilan Halaman Registrasi untuk Owner

4.1.8. Implementasi Halaman Home *Owner*

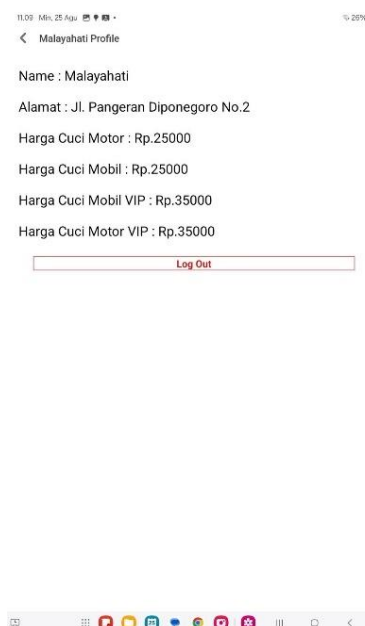
Halaman home ini akan tampil pada saat melakukan login . Pada halaman home owner ini , owner akan bisa beralih ke menu task ,dan profil.



Gambar 4. 8. Tampilan Halaman Home untuk Owner

4.1.9. Implementasi Halaman Profil *Owner*

Pada Halaman Profil Owner ini akan menampilkan detail informasi harga jasa yang telah diinput oleh owner , nama doorsmeer, dan alamat doorsmeer. Pada halaman ini juga owner apat melakukan logout.



Gambar 4. 9. Tampilan Halaman Profil untuk Owner

4.1.10. Implementasi Halaman *Task Screen*

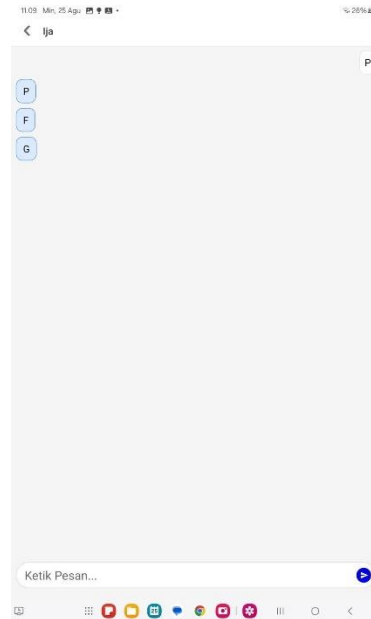
Pada halaman task ini owner dapat melakukan create , update, dan delete terhadap task yang ada. Pengeditan terhadap task tersebut dilakukan berdasarkan kesepakatan yang terjadi di order room.



Gambar 4.10. Tampilan Task Screen

4.1.11. Implementasi Halaman *Order Room*

Halaman order room adalah halaman dimana owner dan user dapat melakukan obrolan untuk menentukan kesepakatan terhadap jasa pencucian kendaraan apa yang diinginkan oleh user.



Gambar 4.11. Tampilan Halaman Order Room

4.2. Pengujian Aplikasi

Pengujian pada aplikasi ini akan dilakukan menggunakan metode Black Box, yang fokus pada pengujian fungsionalitas sistem. Metode ini digunakan untuk memastikan apakah perangkat lunak berfungsi dengan baik atau tidak. Pada pengujian kali ini, akan dibagi menjadi dua bagian, yaitu pengujian untuk *aplikasi user* dan *owner*.

Tabel 4. 1. Tabel Pengujian Role User

Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Mengisi form login dengan email dan password yang sudah terdaftar	User berhasil Melakukan login dan system menampilkan menu utama	Sesuai harapan	Valid
Mengisi form login dengan email dan password yang tidak terdaftar	Sistem akan menolak login	Sesuai harapan	Valid
Mengisi form registrasi	Sistem akan	Sesuai harapan	Valid

Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
	menerima dan menyimpan kredensial akun		
Menekan logo profil	Sistem akan mengalihkan ke menu profil	Sesuai harapan	Valid
Menekan Map	Sistem akan mengalihkan ke Map	Sesuai harapan	Valid
Menekan button order	Sistem akan mengalihkan ke <i>order screen</i>	Sesuai harapan	Valid
Order jasa cuci kendaraan	Sistem akan menerima orderan dan memulai obrolan dengan owner	Sesuai harapan	Valid
Logout	Aplikasi tertutup	Sesuai harapan	Valid

Tabel 4.2. Tabel Pengujian Role Owner

Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
Mengisi form login dengan email dan password yang sudah terdaftar	User berhasil Melakukan login dan system menampilkan menu utama	Sesuai harapan	Valid
Mengisi form login dengan email dan password yang tidak terdaftar	Sistem akan menolak login	Sesuai harapan	Valid
Mengisi form registrasi	Sistem akan menerima dan menyimpan kredensial akun	Sesuai harapan	Valid
Menekan logo profil	Sistem akan mengalihkan ke menu profil	Sesuai harapan	Valid
Menekan button task	Sistem akan mengalihkan ke	Sesuai harapan	Valid

Skenario Pengujian	Hasil yang diharapkan	Hasil Pengujian	Kesimpulan
	halaman <i>order room</i>		
Create Task	Sistem akan menambahkan Task	Sesuai harapan	Valid
Update Task	Sistem akan mengedit task sesuai yang diupdate oleh <i>owner</i>	Sesuai harapan	Valid
Delete Task	Sistem akan menghapus task	Sesuai harapan	Valid
Menerima Orderan	Sistem akan mengalihkan <i>owner</i> ke chatting room dengan <i>user</i>	Sesuai harapan	Valid
Logout	Aplikasi tertutup	Sesuai harapan	Valid

BAB V

PENUTUP

5.1. Kesimpulan

Aplikasi Doorsmeer Online bertujuan untuk mempermudah masyarakat Kota Medan dalam mencari dan memesan layanan pencucian kendaraan dengan menyediakan platform yang terintegrasi bagi pemilik usaha doorsmeer dan pengguna. Aplikasi ini mengidentifikasi dua peran utama: Owner (pemilik doorsmeer) dan User (pengguna layanan).

Role User:

- **Fitur:** Pengguna dapat melakukan login setelah mendaftar, mencari tempat pencucian kendaraan terdekat, memesan layanan, berkomunikasi dengan penyedia layanan, dan mengakses informasi profil serta lokasi mereka.
- **Halaman Utama:** Menampilkan opsi untuk masuk ke halaman Order, Profil, dan Map.
- **Fungsionalitas:** Mempermudah pengguna dalam mencari dan memesan layanan doorsmeer serta berinteraksi dengan penyedia layanan.

Role Owner:

- **Fitur:** Pemilik doorsmeer dapat melakukan login, mengelola profil mereka, mengatur harga layanan, dan mengelola tugas yang terkait dengan pemesanan.
- **Halaman Utama:** Menyediakan akses ke menu Task dan Profil.
- **Fungsionalitas:** Memungkinkan pemilik doorsmeer untuk mengelola pendaftaran usaha mereka, mengatur layanan yang ditawarkan, dan berkomunikasi dengan pengguna melalui halaman Order Room.

Pengujian:

Pengujian aplikasi dilakukan menggunakan metode Black Box, yang berfokus pada pengujian fungsionalitas tanpa melihat struktur internal sistem. Pengujian ini dibagi menjadi dua bagian, masing-masing untuk aplikasi User dan Owner, guna memastikan aplikasi berfungsi dengan baik dan sesuai dengan kebutuhan pengguna.

Secara keseluruhan, aplikasi ini dirancang untuk meningkatkan efisiensi dalam menemukan dan memesan layanan pencucian kendaraan dengan menyediakan antarmuka yang intuitif untuk kedua peran utama: pengguna dan pemilik doorsmeer.

5.2. Saran

Berdasarkan kesimpulan di atas, beberapa saran dapat diajukan untuk pengembangan selanjutnya:

Penambahan fitur pelacakan real-time, sistem ulasan, dan notifikasi untuk meningkatkan pengalaman pengguna.

1. Pertimbangkan ekspansi ke kota lain dan integrasi dengan layanan otomotif lainnya.
2. Lakukan pengujian performa dan tingkatkan keamanan aplikasi.
3. Memperbaiki desain UI/UX dan pertimbangkan dukungan multibahasa untuk menjangkau lebih banyak pengguna.

DAFTAR PUSTAKA

- Ayunita, S., Khadijah, K., Harahap, E. F., & ... (2023). Penerapan Budaya Antri Dalam Meningkatkan Sosial Emosional Anak Usia Dini. *Atthiflah: Journal of Early Childhood Islamic Education*, 10(02), 239–246.
- Bambang, T. B. A. K., & Arhami, L. (2021). Pelatihan Penggunaan Mesin Steam Sc 2, 2 Berpenggerak Motor Bensin 5, 5 Hp Untuk Mencuci Kendaraan Bermotor Di Kp. Cinangka, Banten Provinsi Banten. In *Prosiding Seminar Nasional Pengabdian Kepada Masyarakat* (Vol. 2). <http://journal.unj.ac.id/unj/index.php/snppm>
- Chandiramani, K., Verma, R., & Sivagami, M. (2019). A Modified Priority Preemptive Algorithm for CPU Scheduling. *Procedia Computer Science*, 165(2019), 363–369. <https://doi.org/10.1016/j.procs.2020.01.037>
- Fallo, D. Y. A., & Bulu, V. R. (2022). PENERAPAN ALGORITMA A STAR (A*) PADA GAME LABIRIN. *Jurnal Pendidikan Teknologi Informasi (JUKANTI)*, 5.
- Farida, A., & Mutiono, M. (2023). Pelatihan Pengambilan Data Di Lapangan Menggunakan GPS dan Avenza Maps. *AMMA: Jurnal Pengabdian Masyarakat*, 2(4), 356–361.
- Fristy, P. (2021). Pemilihan Lokasi Strategis Dan Kepuasan Konsumen Terhadap Kualitas Pelayanan Jasa Di Bengkel Dina Doorsmeer. *Juripol*, 4(1), 365–375. <https://doi.org/10.33395/juripol.v4i1.11162>
- Gupta, K., Deepali, G., Kaushik, V., & Gupta, D. (2018). *React Native Application Development 4th International Conference on Cyber Security and Privacy in Communication Networks React Native Application Development*. March. <https://www.researchgate.net/publication/331429981>
- Hidayat, A., Malik, R. F., & Nurmaini, S. (2020). Group Decision Support System (GDSS) dengan Metode Entropy untuk Menentukan Prioritas Antrian Layanan Rumah Sakit Menggunakan Multiple Channel Model (M/M/s). *JATISI (Jurnal Teknik Informatika Dan Sistem Informasi)*, 7(2), 114–125. <https://doi.org/10.35957/jatisi.v7i2.342>
- Hong, Z., Sun, P., Tong, X., Pan, H., Zhou, R., Zhang, Y., Han, Y., Wang, J., Yang, S., & Xu, L. (2021). Improved a-star algorithm for long-distance off-road path planning using terrain data map. *ISPRS International Journal of Geo-Information*, 10(11). <https://doi.org/10.3390/ijgi10110785>
- Hutabalian, M., Sunanto, S., & Januar Al Amien. (2022). Sistem Informasi Geografis Pemetaan Tempat Pembuangan Sampah Sementara di Kota Pekanbaru Dengan Mencari Rute Terdekat Menggunakan Algoritma A Star (A*). *Jurnal CoSciTech (Computer Science and Information Technology)*, 2(2), 33–42. <https://doi.org/10.37859/coscitech.v2i2.2936>
- Jantce TJ Sitinjak, D. D., Maman, ., & Suwita, J. (2020). Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang. *Insan Pembangunan Sistem Informasi Dan Komputer (IPSIKOM)*, 8(1). <https://doi.org/10.58217/ipsikom.v8i1.164>
- Karim, M. A., & Adriansyah, A. R. (2022). Jurnal Informatika Terpadu MENGGUNAKAN METODE HYBRID BERBASIS REACT NATIVE. *Jurnal Informatika Terpadu*, 8(1), 26–34. <https://journal.nurulfikri.ac.id/index.php/JIT>

- Made Dwi Febriyanti, N., KOMPIANG Oka Sudana, A. A., & Nyoman Piarsa, I. (2021). Implementasi Black Box Testing pada Sistem Informasi Manajemen Dosen. *Jitter- Jurnal Ilmiah Teknologi Dan Komputer*, 2(3), 1–10.
- Mayrhofer, R., Stoep, J. Vander, Brubaker, C., & Kravlevich, N. (2021). The Android Platform Security Model. *ACM Transactions on Privacy and Security*, 24(3). <https://doi.org/10.1145/3448609>
- Mohammad Daffa Ananda, M. D. A., Yuliarman Saragih, Y. S., Ridwan Satrio Hadikusuma, R. S. H., & Ahmad Fadhlul Kamal, A. F. K. (2023). Design of Smart Agricultural Systems Using MIT App and Firebase. *Techné: Jurnal Ilmiah Elektroteknika*, 22(2), 183–196. <https://doi.org/10.31358/techne.v22i2.366>
- Mutsaqov, Ativ, Muhammad Fernando, Y., & Megawaty, D. A. (2020). Penerapan Algoritma A-Star Pada Aplikasi Pencarian Lokasi Foto Berbasis Android. *Inovasi Pembangunan: Jurnal ...*, 8(1), 39–52.
- Panjaitan, J., & Pakpahan, A. F. (2021). Perancangan Sistem E-Reporting Menggunakan ReactJS dan Firebase. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1), 20–34. <https://doi.org/10.28932/jutisi.v7i1.3098>
- Pinaria, G. C., Rindengan, Y. D., Najooan, X. B. N., Elektro, T., Sam, U., & Manado, J. K. B. (2021). Web Based E-Commerce Application Buying and Selling Food Ingredients for Manado City. *Jurnal Teknik Informatika*, 1–8.
- Pinayungan, E. M. (2019). Sistem Informasi Kost Berbasis Android di Kota Medan. *Экономика Региона*, 04(479), 133–141.
- Prabowo, O. H., Merthayasa, A., & Saebah, N. (2023). Pemanfaatan Teknologi Informasi Dan Manajemenperubahan Pada Kegiatan Bisnis Di Era Globalisasi. *Syntax Idea*, 5(7), 883–892. <https://www.jurnal.syntax-idea.co.id/index.php/syntax-idea/article/view/2522/1530>
- Purnamasari, M., Nurleli, & Fitriah, E. (2021). Analisis Penerapan Just In Time (JIT) dalam Meningkatkan Efisiensi Biaya Produksi. *Jurnal Riset Akuntansi*, 1(1), 9–14. <https://doi.org/10.29313/jra.v1i1.52>
- Putra, A. P., Andriyanto, F., Karisman, K., Harti, T. D. M., & Sari, W. P. (2020). Pengujian Aplikasi Point of Sale Menggunakan Blackbox Testing. *Jurnal Bina Komputer*, 2(1), 74–78. <https://doi.org/10.33557/binakomputer.v2i1.757>
- Ramadhan, N. A., & Nufriana, D. A. (2019). Rancang Bangun Dan Implementasi Sistem Informasi Skripsi Oline Berbasis WEB. *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, 1(02), 1–12. <https://doi.org/10.46772/intech.v1i02.75>
- Saputra, R. A., Parjito, P., & Wantoro, A. (2020). Implementasi Metode Jackson Network Queue Pada Pemodelan Sistem Antrian Booking Pelayanan Car Wash (Studi Kasus: Autosshine Car Wash Lampung). *Jurnal Teknologi Dan Sistem Informasi*, 1(2), 80–86. <https://doi.org/10.33365/jtsi.v1i2.433>
- Setyawatu, R., & Bachtiar Maulachela, A. (2020). Penerapan Algoritma Dynamic Priority Scheduling Pada Aplikasi Antrian Pencucian Mobil Berbasis Mobile (Implementation of Dynamic Priority Scheduling Algorithm in Mobile Car Queue Washing Applications). *Jurnal Teknologi Informasi Dan Multimedia*, 2(1), 29–35.
- Sumantri, G. W., & Laluma, R. H. (2022). Penerapan Algoritma Genetika Untuk Optimasi Penjadwalan Akademik Di Universitas Sangga Buana. *Prosiding Seminar Sosial*

Politik, Bisnis, Akuntansi Dan Teknik, 4, 158.
<https://doi.org/10.32897/sobat.2022.4.0.1919>

Sundari, S., & Yogie Syahputra, M. (2023). Penerapan algoritma Shortest Job First (SJF) dan Priority Scheduling (PS) Pada Maintenance Mesin ATM. *Jurnal Ilmu Komputer Dan Informatika*, 6341(April), 1.

Suparman, M., Rosada, M., Lutpi, M., Kamaliya, P., Sabaniah, F., Alfian, R. H., Ramadhan, F., Alfaro, I., & Rosdiana, M. (2023). Mengenal Aplikasi Figma Untuk Membuat Content Menjadi Lebih Interaktif di Era Society 5.0. *Abdi Jurnal Publikasi*, 1(6), 55
 2–555.

[https://jurnal.portalpublikasi.id/index.php/AJP/article/download/283/191#:~:text=Fi gma](https://jurnal.portalpublikasi.id/index.php/AJP/article/download/283/191#:~:text=Fi%20gma)
 adalah salah satu tools, pelaksanaan PKM berbagai desain lainnya

Susanto, A., & Handayani, S. F. (2022). Aplikasi Pencarian Jasa Pangkas Rambut Di Bandar Lampung Menggunakan Algoritma a-Star Berbasis Android. *Jurnal Data Mining Dan Sistem Informasi*, 3(1), 39. <https://doi.org/10.33365/jdmsi.v3i1.1944>

Sutanti, A., MZ, M. K., Mustika, M., & Damayanti, P. (2020). Rancang Bangun Aplikasi Perpustakaan Keliling Menggunakan Pendekatan Terstruktur. *Komputa : Jurnal Ilmiah Komputer Da*

Syahputri, A. Z., Fallenia, F. Della, & Syafitri, R. (2023). *Kerangka Berfikir Penelitian Kuantitatif*.

Tampubolon, H. S. (2022). Sistem Informasi Pengolahan Data Doorsmeer pada Astro Autocare Pematang Siantar berbasis WEB. *Jurnal Bisantara Informatika (JBI)*, 6(2), 89–96.
<https://bisantara.amikparbinanusantara.ac.id/index.php/bisantara/article/view/75/62>

Widiarta, I. M., Julkarnain, M., & Imanulloh, J. (2021). Rancang Bangun Aplikasi Uts in Me Berbasis Android Menggunakan Flutter Dengan Metode Rapid Application Development. *Jurnal Informatika Teknologi Dan Sains*, 3(4), 447–452.
<https://doi.org/10.51401/jinteks.v3i4.1323>

Widiastika, M. A., Hendracipta, N., & Syachruroji, A. (2020). Pengembangan Media Pembelajaran Mobile Learning Berbasis Android Pada Konsep Sistem Peredaran Darah di Sekolah dasar. *Jurnal Basicedu*, 5(1), 47–64.
<https://doi.org/10.31004/basicedu.v5i1.602>

Wijaya, Y. D. (2021). Penerapan Metode Rapid Application Development (Rad) Dalam Pengembangan Sistem Informasi Data Toko. *Jurnal SITECH: Sistem Informasi Dan Teknologi*, 3(2), 95–102. <https://doi.org/10.24176/sitech.v3i2.5141>

Wijaya, Y. D., & Astuti, M. W. (2021). Pengujian Blackbox Sistem Informasi Penilaian Kinerja Karyawan Pt Inka (Persero) Berbasis Equivalence Partitions. *Jurnal Digital Teknologi Informasi*, 4(1), 22. <https://doi.org/10.32502/digital.v4i1.3163>

Windane, W. W., & Lathifah, L. (2021). E-Commerce Toko Fisago.Co Berbasis Android. *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, 2(3), 285–303.
<https://doi.org/10.33365/jatika.v2i3.1139>

LAMPIRAN

Lampiran 1. Source Code User

SignIn.js

```
import { Alert, Image, Pressable, Text, TextInput, TouchableOpacity, View } from 'react-native'
import React, { useRef, useState } from 'react'
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-responsive-screen'
import { StatusBar } from 'expo-status-bar'
import { Octicons } from '@expo/vector-icons'
import { useRouter } from 'expo-router'
import Loading from '../components/Loading'
import { useAuth } from '../context/authContext'

export default function SignIn() {
  const router = useRouter();
  const [loading, setLoading] = useState(false);
  const { login } = useAuth();

  const emailRef = useRef("");
  const passwordRef = useRef("");

  const handleLogin = async ()=>{
    if(!emailRef.current || !passwordRef.current){
      Alert.alert('Sign In', 'Please fill all the fields')
      return;
    }

    setLoading(true);
    const response = await login(emailRef.current, passwordRef.current);
    setLoading(false);
    if(!response.success){
      Alert.alert('Sign In :', response.msg);
    }
  }

  return (
    <View className="flex-1">
      <StatusBar style='dark' />
      <View style={{paddingTop:hp(8), paddingHorizontal:wp(5)}} className="flex-1 gap-12">
        <View className="items-center">
```

```

    <Image style = {{height:hp(25)}} resizeMode='contain'
source={{require('../assets/images/carwash.jpg')}} />
  </View>

  <View className="gap-10">
    <Text style={{fontSize: hp(4)}} className="font-bold tracking-wider text-center
text-neutral-800">
      Sign In
    </Text>
    <View className="gap-10 pt-5">
      <View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100
items-center rounded-2xl">
        <Octicons name='mail' size={{hp(2.7)}} color="gray" />
        <TextInput
          onChangeText={value=> emailRef.current=value}
          style = {{fontSize:hp(2)}}
          className="flex-1 font-semibold text-neutral-700"
          placeholder='Email'
          placeholderTextColor={'gray'}
        />
      </View>

      <View className="gap-3 pb-10">
        <View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100
items-center rounded-2xl pb-">
          <Octicons name='lock' size={{hp(2.7)}} color="gray" />
          <TextInput
            onChangeText={value=> passwordRef.current=value}
            style = {{fontSize:hp(2)}}
            className="flex-1 font-semibold text-neutral-700"
            placeholder='Password'
            secureTextEntry
            placeholderTextColor={'gray'}
          />
        </View>
        <Pressable onPress={()=> router.push('signUp')}>
          <Text style={{fontSize:hp(1.8)}} className="font-semibold text-blue-200">
            Daftar Akun
          </Text>
        </Pressable>

      </View>

    <View>
      {
        loading? (
          <View className="flex-row justify-center">

```

```

        <Loading size = {hp(20)} />
      </View>
    ):
      <TouchableOpacity onPress = {handleLogin} style={{backgroundColor:#81d2f0,
alignItems:'center', justifyContent:"center", borderRadius:25, height:hp(6.5)}}>
        <Text style={{fontSize:hp(2.7)}} className="text-white font-semibold tracking-
wider">
          Sign In
        </Text>
      </TouchableOpacity>
    )
  }

</View>

</View>

</View>

</View>
</View>
)
}

```

SignUp.js

```

import { Alert, Image, Pressable, Text, TextInput, TouchableOpacity, View } from 'react-
native'
import React, { useRef, useState } from 'react'
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-
responsive-screen'
import { StatusBar } from 'expo-status-bar'
import { Feather, Octicons } from '@expo/vector-icons'
import { useRouter } from 'expo-router'
import Loading from '../components/Loading'
import CustomKeyboardView from '../components/CustomKeyboardView'
import { useAuth } from '../context/authContext'
import UploadMediaFile from '../components/UploadMediaFile'

export default function SignUp() {
  const router = useRouter();
  const [loading, setLoading] = useState(false);
  const { register } = useAuth();

```

```

const emailRef = useRef("");
const passwordRef = useRef("");
const fullnameRef = useRef("");
const roleRef = useRef("Pengguna");
const locationRef = useRef("");

const handleRegister = async ()=>{
  if(!emailRef.current || !passwordRef.current || !fullnameRef.current || !roleRef.current ){
    Alert.alert('Sign Up', 'Please fill all the fields')
    return;
  }
  setLoading(true);

  let response = await register(emailRef.current, passwordRef.current, fullnameRef.current,
roleRef.current, locationRef.current);
  setLoading(false);

  if(!response.success){
    Alert.alert('Sign Up', response.msg);
  }

}

return (
  <CustomKeyboardView>
    <StatusBar style='dark' />
    <View style={{paddingTop:hp(8), paddingHorizontal:wp(5)}} className="flex-1 gap-8">
      <View className="items-center">
        <Image style = {{height:hp(20)}} resizeMode='contain'
source={require('../assets/images/signup.jpg')} />
      </View>

      <View className="gap-5">
        <Text style={{fontSize: hp(4)}} className="font-bold tracking-wider text-center text-neutral-800">
          Sign Up
        </Text>
        <View className="gap-5 pt-5">
          <View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100 items-center rounded-2xl">
            <Feather name='user' size={hp(2.7)} color="gray" />
            <TextInput
              onChangeText={value=> fullnameRef.current=value }

```

```

    style = { {fontSize:hp(2)} }
    className="flex-1 font-semibold text-neutral-700"
    placeholder='Full Name'
    placeholderTextColor={ 'gray' }
  />
</View>
<View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100
items-center rounded-2xl">
  <Octicons name='mail' size={hp(2.7)} color="gray" />
  <TextInput
    onChangeText={ value=> emailRef.current=value }
    style = { {fontSize:hp(2)} }
    className="flex-1 font-semibold text-neutral-700"
    placeholder='Email'
    placeholderTextColor={ 'gray' }
  />
</View>
  <View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100
items-center rounded-2xl pb-">
    <Octicons name='lock' size={hp(2.7)} color="gray" />
    <TextInput
      onChangeText={ value=> passwordRef.current=value }
      style = { {fontSize:hp(2)} }
      className="flex-1 font-semibold text-neutral-700"
      placeholder='Password'
      secureTextEntry
      placeholderTextColor={ 'gray' }
    />

  </View>

  <View>
    {
      loading? (
        <View className="flex-row justify-center">
          <Loading size = {hp(20)} />
        </View>
      ): (
        <TouchableOpacity onPress = {handleRegister}
style={{ backgroundColor:'#81d2f0', alignItems:'center', justifyContent:"center",
borderRadius:25, height:hp(6.5)}} >
          <Text style={{fontSize:hp(2.7)}} className="text-white font-semibold tracking-
wider">
            Sign Up
          </Text>
        </TouchableOpacity>
      )
    }
  </View>

```



```

    }

    </View>
    <View className="flex-row justify-between mt-5">
      <View>
        <Pressable onPress={() => router.push('signIn')}>
          <Text style={{fontSize:hp(1.8)}} className="font-semibold text-blue-200">
            Kembali ke halaman Login
          </Text>
        </Pressable>
      </View>

    </View>

  </View>

</View>

</View>

</CustomKeyboardView>
)
}

```

Home.js

```

import { View, Text, Button, ActivityIndicator, TouchableOpacity } from 'react-native'
import React, { useEffect, useState } from 'react'
import { useAuth } from '../context/authContext'
import { StatusBar } from 'expo-status-bar';
import { ownersRef, usersRef } from '../firebaseConfig';
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-responsive-screen'
import ChatList from '../components/ChatList';
import Loading from '../components/Loading';
import { getDocs, query, where } from 'firebase/firestore';
import { FontAwesome6 } from '@expo/vector-icons';
import { useRouter } from 'expo-router';
import * as Location from 'expo-location'

export default function Home ({item}) {

  const router = useRouter();

```

```

const handleProfile = () =>{
  router.push({pathname: '/profile', params: item});
}
const [latitude, setLatitude] = useState();
const [longitude, setLongitude] = useState();

const {user, logout} = useAuth();

const [users, setUsers] = useState([]);
useEffect(()=>{
  if(user?.uid)
    getUsers();
},[])

const getUsers = async()=>{
  const q = query(ownersRef, where('userId', '!=', user?.uid));

  const querySnapShot = await getDocs(q);
  let data=[];
  querySnapShot.forEach(doc=>{
    data.push({...doc.data()});
  });

  setUsers(data);
}

const [location, setLocation] = useState();

useEffect(()=>{
  const getPermissions = async()=>{
    let {status} = await Location.requestForegroundPermissionsAsync();
    if (status !== 'granted'){
      console.log("Please Grant Location Permissions")
      return;
    }
    let currentLocation = await Location.getCurrentPositionAsync({});
    setLocation(currentLocation);
    console.log("Location :");
    console.log(currentLocation)
  };
  getPermissions();
}, [])

return (
  <View className = "flex-1 bg-white">

```

```

<StatusBar style='dark'/>

{
  users.length>0? (
    <ChatList users={users} />
  ):(
    <View className="flex-1">
      <View className="flex items-center" style={{top:hp(30)}}>
        <Loading size = {hp(20)}/>
      </View>
      <View className="ml-10">
        <TouchableOpacity onPress={handleProfile}>
          <FontAwesome6 name = "user-circle" size={hp(4)} color="black"/>
        </TouchableOpacity>
      </View>
    </View>
  )
}

</View>
)
}

```

Profile.js

```

import { View, Text, TouchableOpacity, Dimensions, StyleSheet } from 'react-native'
import React, { useState, useEffect } from 'react'
import { useAuth } from '../context/authContext';
import { router, useLocalSearchParams } from 'expo-router';
import ProfileInside from '../components/ProfileInside';
import ProfileHeader from '../components/ProfileHeader';
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-responsive-screen'
import * as Location from 'expo-location'
import MapView from 'react-native-maps'

export default function profile() {

  const {user, logout} = useAuth();
  const [users, setUsers] = useState([]);
  const item = useLocalSearchParams();
  const [location, setLocation] = useState(null);
  const [latitude, setLatitude] = useState();
  const [longitude, setLongitude] = useState();

```

```

const handleLogout = async()=>{
  await logout();
}

useEffect(()=>{
  const getPermissions = async()=>{
    let {status} = await Location.requestForegroundPermissionsAsync();
    if (status !== 'granted'){
      console.log("Please Grant Location Permissions")
      return;
    }
    let currentLocation = await Location.getCurrentPositionAsync({});
    setLocation(currentLocation);
    setLatitude(currentLocation.coords.latitude);
    setLongitude(currentLocation.coords.longitude)
    console.log("Location :");
    console.log(currentLocation)
    console.log("Latitude :", currentLocation.coords.latitude);
    console.log("Longitude :", currentLocation.coords.longitude);

  };
  getPermissions();
}, [])

let text = "waiting"
if(location){
  text = JSON.stringify(location);
}

const styles = StyleSheet.create({
  map:{
    width : Dimensions.get('window').width,
    height : 800
  }
})

return (
  <View className="flex-1 bg-white">
    <ProfileHeader user={user} router={router}/>
    <ProfileInside user={user} router={router}/>
    <View className="flex-1 items-center justify-center">
      <Text style={{fontSize:25, marginBottom:10}}>Your Location :</Text>
      <MapView
        showsMyLocationButton={true}

```

```

      showsUserLocation={true}
      style={styles.map}
    />
  </View>
  <View className="flex justify-center items-center border border-red-500 mx-16 mb-20">
    <TouchableOpacity onPress={handleLogout}>
      <Text style={{fontSize:hp(1.5), color:"red"}} className="font-semibold" >
        Log Out
      </Text>
    </TouchableOpacity>
  </View>
</View>
)
}

```

Lampiran 2. Source Code Owner

SignIn.js

```

import { Alert, Image, Pressable, Text, TextInput, TouchableOpacity, View } from 'react-native'
import React, { useRef, useState } from 'react'
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-responsive-screen'
import { StatusBar } from 'expo-status-bar'
import { Octicons } from '@expo/vector-icons'
import { useRouter } from 'expo-router'
import Loading from '../components/Loading'
import { useAuth } from '../context/authContext'

export default function SignIn() {
  const router = useRouter();
  const [loading, setLoading] = useState(false);
  const { login } = useAuth();

  const emailRef = useRef("");
  const passwordRef = useRef("");

  const handleLogin = async ()=>{
    if(!emailRef.current || !passwordRef.current){
      Alert.alert('Sign In', 'Please fill all the fields')
      return;
    }

    setLoading(true);

```

```

const response = await login(emailRef.current, passwordRef.current);
setLoading(false);
if(!response.success){
  Alert.alert('Sign In :', response.msg);
}
}

return (
  <View className="flex-1">
    <StatusBar style='dark' />
    <View style={{paddingTop:hp(8), paddingHorizontal:wp(5)}} className="flex-1 gap-12">
      <View className="items-center">
        <Image style = {{height:hp(25)}} resizeMode='contain'
source={require('../assets/images/carwash.jpg')} />
      </View>

      <View className="gap-10">
        <Text style={{fontSize: hp(4)}} className="font-bold tracking-wider text-center text-neutral-800">
          Sign In
        </Text>
        <View className="gap-10 pt-5">
          <View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100 items-center rounded-2xl">
            <Octicons name='mail' size={hp(2.7)} color="gray" />
            <TextInput
              onChangeText={value=> emailRef.current=value}
              style = {{fontSize:hp(2)}}
              className="flex-1 font-semibold text-neutral-700"
              placeholder='Email'
              placeholderTextColor={gray}
            />
          </View>

          <View className="gap-3 pb-10">
            <View style={{height:hp(7)}} className="flex-row gap-4 px-4 bg-neutral-100 items-center rounded-2xl pb->
              <Octicons name='lock' size={hp(2.7)} color="gray" />
              <TextInput
                onChangeText={value=> passwordRef.current=value}
                style = {{fontSize:hp(2)}}
                className="flex-1 font-semibold text-neutral-700"
                placeholder='Password'
                secureTextEntry
                placeholderTextColor={gray}
              />
            </View>
          </View>
        </View>
      </View>
    </View>
  </View>
)

```

```

    </View>
    <Pressable onPress={()=> router.push('ownerSignUp')}>
      <Text style={{fontSize:hp(1.8)}} className="font-semibold text-blue-200">
        Daftar Akun
      </Text>
    </Pressable>

    </View>

    <View>
    {
      loading? (
        <View className="flex-row justify-center">
          <Loading size = {hp(20)} />
        </View>
      ):(
        <TouchableOpacity onPress = {handleLogin} style={{backgroundColor:'#81d2f0',
alignItems:'center', justifyContent:"center", borderRadius:25, height:hp(6.5)}} >
          <Text style={{fontSize:hp(2.7)}} className="text-white font-semibold tracking-
wider">
            Sign In
          </Text>
        </TouchableOpacity>
      )
    }

    </View>

    </View>

    </View>

    <View>
    <View>
    )
  }

```

OwnerSignUp.js

```

import { Alert, Image, Pressable, Text, TextInput, TouchableOpacity, View } from 'react-
native'
import React, { useRef, useState, useEffect } from 'react'
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-
responsive-screen'
import { StatusBar } from 'expo-status-bar'

```

```

import { Entypo, Feather, FontAwesome, FontAwesome5, Fontisto, Octicons } from
 '@expo/vector-icons'
import { useRouter } from 'expo-router'
import Loading from '../components/Loading'
import CustomKeyboardView from '../components/CustomKeyboardView'
import { useAuth } from '../context/authContext'
import * as Location from 'expo-location'

export default function SignUp() {
  const router = useRouter();
  const [loading, setLoading] = useState(false);
  const { register } = useAuth();
  const [latitude, setLatitude] = useState("");
  const [longitude, setLongitude] = useState("");

  const fullnameRef = useRef("");
  const emailRef = useRef("");
  const passwordRef = useRef("");
  const latitudeRef = useRef("");
  const longitudeRef = useRef("");
  const addressRef = useRef("");
  const mobilRef = useRef("");
  const motorRef = useRef("");
  const mobilRef = useRef("");
  const motorRef = useRef("");

  const handleRegisterOwner = async ()=>{
    if(!emailRef.current || !passwordRef.current || !fullnameRef.current || !latitudeRef.current ||
    !longitudeRef.current || !addressRef.current || !motorRef.current || !mobilRef.current ||
    !mobilRef.current || !motorRef.current){
      Alert.alert('Sign Up', 'Please fill all the fields')
      return;
    }
    setLoading(true);

    let response = await register(emailRef.current , passwordRef.current, fullnameRef.current,
    latitudeRef.current, longitudeRef.current, addressRef.current, mobilRef.current,
    motorRef.current, mobilRef.current, motorRef.current);
    setLoading(false);

    if(!response.success){
      Alert.alert('Sign Up', response.msg);
    }
  }
}

```



```

const [location, setLocation] = useState();

useEffect(()=>{
  const getPermissions = async()=>{
    let {status} = await Location.requestForegroundPermissionsAsync();
    if (status !== 'granted'){
      console.log("Please Grant Location Permissions")
      return;
    }
    let currentLocation = await Location.getCurrentPositionAsync({});
    setLocation(currentLocation);
    setLatitude(currentLocation.coords.latitude);
    setLongitude(currentLocation.coords.longitude);
    console.log(currentLocation.coords.latitude)
  };
  getPermissions();
}, [])

const latitudeValue = (JSON.stringify(latitude))
const longitudeValue = (JSON.stringify(longitude))

return (
  <CustomKeyboardView>
    <StatusBar style='dark' />
    <View style={{paddingTop:hp(8), paddingHorizontal:wp(5)}} className="flex-1 gap-8">
      <View className="items-center">
        <Image style = {{height:hp(20)}} resizeMode='contain'
source={require('../assets/images/owner.png')} />
      </View>

      <View className="gap-5">
        <Text style={{fontSize: hp(4)}} className="font-bold tracking-wider text-center text-neutral-800">
          Owner Sign Up
        </Text>
        <View className="gap-5 pt-5">
          <View className="flex-row justify-between">
            <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-neutral-100 items-center rounded-2xl">
              <FontAwesome name='building' size={hp(2.7)} color="gray" />
              <TextInput
                onChangeText={value=> fullnameRef.current=value}
                style = {{fontSize:hp(1.5)}}
                className="flex-1 font-semibold text-neutral-700"
                placeholder='Company Name'

```

```

        placeholderTextColor={'gray'}
      />
    </View>
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl">
      <Entypo name='location-pin' size={hp(2.7)} color="gray" />
      <TextInput
        onChangeText={value=> addressRef.current=value }
        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholder='Address'
        placeholderTextColor={'gray'}
      />
    </View>
  </View>

  <View className="flex-row justify-between">
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl">
      <Entypo name='location-pin' size={hp(2.7)} color="gray" />
      <TextInput
        onChangeText={value=> latitudeRef.current=value }
        value = {latitudeValue}
        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholderTextColor={'gray'}
      />
    </View>
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl">
      <Entypo name='location-pin' size={hp(2.7)} color="gray" />
      <TextInput
        onChangeText={value=> longitudeRef.current=value }
        value = {longitudeValue}
        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholderTextColor={'gray'}
      />
    </View>
  </View>

  <View className="flex-row justify-between">
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl">
      <Octicons name='mail' size={hp(2.7)} color="gray" />
      <TextInput
        onChangeText={value=> emailRef.current=value }

```

```

        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholder='Email'
        placeholderTextColor={'gray'}
    />
</View>
<View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl pb->
    <Octicons name='lock' size={hp(2.7)} color="gray" />
    <TextInput
        onChangeText={value=> passwordRef.current=value }
        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholder='Password'
        secureTextEntry
        placeholderTextColor={'gray'}
    />
</View>
</View>
<View className="flex-row justify-between">
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl">
        <FontAwesome5 name='car-alt' size={hp(2.7)} color="gray" />
        <TextInput
            onChangeText={value=> mobilRef.current=value }
            style = {{fontSize:hp(1.5)}}
            className="flex-1 font-semibold text-neutral-700"
            placeholder='Harga Cuci Mobil'
            placeholderTextColor={'gray'}
            keyboardType='numeric'
        />
    </View>
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl pb->
        <Fontisto name='motorcycle' size={hp(2.7)} color="gray" />
        <TextInput
            onChangeText={value=> motorRef.current=value }
            style = {{fontSize:hp(1.5)}}
            className="flex-1 font-semibold text-neutral-700"
            placeholder='Harga Cuci Motor'
            placeholderTextColor={'gray'}
            keyboardType='numeric'
        />
    </View>
</View>
</View>
<View className="flex-row justify-between">

```

```

    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl">
      <FontAwesome5 name='car-alt' size={hp(2.7)} color="gray" />
      <TextInput
        onChangeText={value=> mobilRef.current=value }
        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholder='Harga Cuci Mobil VIP'
        placeholderTextColor={'gray'}
        keyboardType='numeric'
      />
    </View>
    <View style={{height:hp(7), width:hp(25)}} className="flex-row gap-4 px-4 bg-
neutral-100 items-center rounded-2xl pb-">
      <Fontisto name='motorcycle' size={hp(2.7)} color="gray" />
      <TextInput
        onChangeText={value=> motorRef.current=value }
        style = {{fontSize:hp(1.5)}}
        className="flex-1 font-semibold text-neutral-700"
        placeholder='Harga Cuci Motor VIP'
        placeholderTextColor={'gray'}
        keyboardType='numeric'
      />
    </View>
  </View>
  <View className="flex justify-center items-center">
    <Text>
      Masukkan Angka 0 apabila ada yang tidak digunakan
    </Text>
  </View>

  <View>
    {
      loading? (
        <View className="flex-row justify-center">
          <Loading size = {hp(20)} />
        </View>
      ):
      <TouchableOpacity onPress = {handleRegisterOwner }
style={{backgroundColor:#81d2f0, alignItems:'center', justifyContent:"center",
borderRadius:25, height:hp(6.5)}}>
        <Text style={{fontSize:hp(2.7)}} className="text-white font-semibold tracking-
wider">
          Sign Up
        </Text>
      </TouchableOpacity>

```

```

    )
  }

  </View>
  <View className="flex-row justify-between mt-5">
    <View>
      <Pressable onPress={() => router.push('signIn')}>
        <Text style={{fontSize:hp(1.8)}} className="font-semibold text-blue-200">
          Kembali ke halaman Login
        </Text>
      </Pressable>
    </View>
  </View>

  </View>

  </View>

  </View>
  </CustomKeyboardView>
)
}

```

Home.js

```

import { View, TouchableOpacity, Button, Text } from 'react-native'
import React, { useEffect, useState } from 'react'
import { useAuth } from '../context/authContext'
import { StatusBar } from 'expo-status-bar';
import { usersRef } from '../firebaseConfig';
import { heightPercentageToDP as hp } from 'react-native-responsive-screen'
import ChatList from '../components/ChatList';
import Loading from '../components/Loading';
import { getDocs, query, where } from 'firebase/firestore';
import { AntDesign, FontAwesome6 } from '@expo/vector-icons';
import { useRouter } from 'expo-router';
import * as Location from 'expo-location'

export default function Home ({item}) {

  const router = useRouter();

  const handleProfile = () =>{
    router.push({pathname: '/profile', params: item});
  }

  const handleTask = () => {

```

```

router.push({pathname: '/tasks'});
}

const {user, logout} = useAuth();

const [users, setUsers] = useState([]);
useEffect(()=>{
  if(user?.uid)
    getUsers();
},[])

const getUsers = async()=>{
  const q = query(usersRef, where('userId', '!=', user?.uid));

  const querySnapShot = await getDocs(q);
  let data=[];
  querySnapShot.forEach(doc=>{
    data.push({...doc.data()});
  });

  setUsers(data);
}

const [location, setLocation] = useState();

useEffect(()=>{
  const getPermissions = async()=>{
    let {status} = await Location.requestForegroundPermissionsAsync();
    if (status !== 'granted'){
      console.log("Please Grant Location Permissions")
      return;
    }
    let currentLocation = await Location.getCurrentPositionAsync({});
    setLocation(currentLocation);
    console.log("Location :");
    console.log(currentLocation)
  };
  getPermissions();
}, [])

return (
  <View className = "flex-1 bg-white">
    <StatusBar style='dark'/>

    {
      users.length>0? (
        <ChatList users={users} />

```

```

):(
  <View className="flex-1">
    <View className="flex items-center" style={{top:hp(30)}}>
      <Loading size = {hp(20)}/>
    </View>
    <View className="ml-10">
      <TouchableOpacity onPress={handleProfile}>
        <FontAwesome6 name = "user-circle" size={hp(4)} color="black"/>
      </TouchableOpacity>
    </View>

  </View>
)
}

```

```

</View>
)
}

```

ChatRoom.js

```

import { View, Text, TextInput, TouchableOpacity, Alert } from 'react-native'
import React, { useEffect, useRef, useState } from 'react'
import { useLocalSearchParams, useRouter } from 'expo-router'
import ChatRoomHeader from '../components/ChatRoomHeader';
import MessageList from '../components/MessageList';
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-responsive-screen'
import { MaterialCommunityIcons } from '@expo/vector-icons';
import CustomKeyboardView from '../components/CustomKeyboardView';
import { useAuth } from '../context/authContext';
import { addDoc, collection, doc, onSnapshot, orderBy, query, setDoc, Timestamp } from 'firebase/firestore';
import { getRoomId } from '../utils/common'
import { db } from '../firebaseConfig';

```

```

export default function ChatRoom() {
  const item = useLocalSearchParams();
  const {user} = useAuth();
  const router = useRouter();
  const [messages, setMessages] = useState([]);
  const textRef = useRef("");
  const inputRef = useRef(null);

```

```

useEffect(()=>{
  createRoomIfNotExists();

  let roomId = getRoomId(user?.userId, item?.userId);
  const docRef = doc(db, "rooms", roomId);
  const messagesRef = collection(docRef, "messages");
  const q = query(messagesRef, orderBy('createdAt', 'asc'));

  let unsub = onSnapshot(q, (snapshot)=>{
    let allMessages = snapshot.docs.map(doc=>{
      return doc.data();
    });
    setMessages([...allMessages]);
  });

  return unsub;
}, [])

const createRoomIfNotExists = async ()=>{
  let roomId = getRoomId(user?.userId, item?.userId);
  await setDoc(doc(db, "rooms", roomId), {
    roomId,
    createdAt: Timestamp.fromDate(new Date())
  });
}

const handleSendMessage = async ()=>{
  let message = textRef.current.trim();
  if(!message) return;
  try {
    let roomId = getRoomId(user?.userId, item?.userId);
    const docRef = doc(db, "rooms", roomId);
    const messagesRef = collection(docRef, 'messages')
    textRef.current="";
    if(inputRef) inputRef?.current?.clear();
    const newDoc = await addDoc(messagesRef, {
      userId : user?.userId,
      text : message,
      senderName : user?.fullname,
      createdAt : Timestamp.fromDate(new Date())
    });

    console.log('new Message id:', newDoc.id);
  }catch(err){
    Alert.alert('Message', err.message);
  }
}

```



```

return (
  <CustomKeyboardView>
  <View className="flex-1 bg-white">
    <ChatRoomHeader user={item} router={router}/>
    <View className="flex-1 justify-between bg-neutral-100 overflow-visible">
      <View className="flex-1">
        <MessageList messages={messages} currentUser={user}/>
      </View>
      <View style={{marginBottom:hp(2)}} className="pt-2">
        <View className="flex-row justify-between mx-3 bg-white border p-2 border-
neutral-300 rounded-full pl-5">
          <TextInput
            ref={inputRef}
            onChangeText={value=> textRef.current = value}
            placeholder='Ketik Pesan...'
            style={{fontSize:hp(1.8)}}
            className="flex-1 mr-2"
          />
          <TouchableOpacity onPress={handleSendMessage} className="bg-neutral-200
p-2 mr-[1px] rounded-full">
            <MaterialCommunityIcons name="send-circle" size={hp(3)} color="blue"/>
          </TouchableOpacity>
        </View>
      </View>
    </View>
  </View>
  </View>
  </View>
  </CustomKeyboardView>
)
}

```

Profile.js

```

import { View, Text, TouchableOpacity } from 'react-native'
import React, { useState } from 'react'
import { useAuth } from '../context/authContext';
import { router, useLocalSearchParams } from 'expo-router';
import ProfileInside from '../components/ProfileInside';
import ProfileHeader from '../components/ProfileHeader';
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-
responsive-screen'

```

```

export default function profile() {

  const {user, logout} = useAuth();
  const [users, setUsers] = useState([]);
  const item = useLocalSearchParams();

  const handleLogout = async()=>{
    await logout();
  }

  return (
    <View className="flex-1 bg-white">
      <ProfileHeader user={user} router={router}/>
      <ProfileInside user={user} router={router}/>
      <TouchableOpacity onPress={handleLogout}>
        <View className="flex justify-center items-center mt-10 border border-red-500 mx-16">
          <Text style={{fontSize:hp(1.5), color:"red"}} className="font-semibold" >
            Log Out
          </Text>
        </View>
      </TouchableOpacity>
    </View>
  )
}

```

Tasks.js

```

import React, { useState, useEffect } from 'react';
import { View, TextInput, Button, FlatList, Text, Alert } from 'react-native';
import AsyncStorage from '@react-native-async-storage/async-storage';
import { TouchableOpacity } from 'react-native';
import { EvilIcons, FontAwesome6 } from '@expo/vector-icons';
import { widthPercentageToDP as wp, heightPercentageToDP as hp } from 'react-native-responsive-screen'

const App = () => {
  const [textInput, setTextInput] = React.useState("")
  const [tasks, setTasks] = React.useState([])

  React.useEffect(() => {
    getTasks()
  }, [])
}

```

```

React.useEffect(() => {
  saveTasks(tasks)
}, [tasks])

const addTask = ()=>{
  if (textInput === "") {
    return;
  }else{
    const newTask = {
      id:Math.random(),
      task:textInput
    };
    setTasks([...tasks, newTask]);
    setTextInput("")
  }
}

const deleteTask = taskId => {
  const newTask = tasks.filter(item => item.id !== taskId)
  setTasks(newTask)
}

const clearTask = () => {
  Alert.alert('Confirmation', 'Clear All Task?', [
    {
      text:"No"
    },{
      text:'Yes',
      onPress: () => setTasks([])
    }
  ])
}

const saveTasks = async tasks =>{
  try {
    const stringifyTasks = JSON.stringify(tasks)
    await AsyncStorage.setItem('tasks', stringifyTasks)
  } catch(e){
    console.log(e)
  }
}

const getTasks = async () => {
  try {
    const tasks = await AsyncStorage.getItem('tasks')

```

```

    if (tasks !== null) {
      setTasks(JSON.parse(tasks))
    }
  } catch (error) {
    console.log(error)
  }
}

```

```

const ListItem = ({ task }) => {
  return <View
    style={{
      borderColor:'black',
      borderWidth:1,
      borderRadius:6,
      marginVertical:10,
      marginHorizontal:16,
      paddingHorizontal:12,
      paddingVertical:20,
      flexDirection:'row',
      alignItems:'center'
    }}>
    <Text
      style={{
        color:'black',
        fontSize:20,
        flex:1,
        fontWeight:'bold'
      }}
    >
      {task?.task}
    </Text>

    <TouchableOpacity>
      <EvilIcons
        name='trash'
        size={45}
        color="red"
        style={{marginLeft:15}}
        onPress={() => deleteTask(task?.id)}
      />
    </TouchableOpacity>

  </View>
}

return (

```

```

<View className="flex-1 bg-white">
  <View style={{marginHorizontal:16, marginVertical:16}}>
    <TextInput
      style={{
        borderWidth:1,
        borderColor:'blue',
        borderRadius:6,
        paddingVertical:12,
        paddingHorizontal:12
      }}
      placeholder="Add a task"
      value={textInput}
      onChangeText={(text) => setTextInput(text)}
    />
  </View>
  <View>
    <TouchableOpacity
      style={{
        backgroundColor:'black',
        alignItems:'center',
        borderRadius:6,
        paddingVertical:8,
        marginTop:24,
        marginHorizontal:16
      }}
      onPress={addTask}
    >
    <Text style={{color:'white', fontWeight:'bold', fontSize:20}}>
      Add
    </Text>
  </TouchableOpacity>
</View>
<View style={{marginTop:20}}>
  <FlatList
    data={tasks}
    renderItem={({item}) => <ListItem task={item}/>}
  />
</View>
<View className="ml-10 mb-5" style={{ flex:1,justifyContent:'flex-end',
alignItems:'center'}}>
  <TouchableOpacity
    style={{
      borderWidth:1,
      borderColor:'red',
      borderRadius:20,
      marginHorizontal:5,
      marginVertical:5,

```

```
paddingVertical:10,  
paddingHorizontal:10  
  }  
  onPress={clearTask}  
>  
  <EvilIcons name='trash'  
    size={45}  
    color="red"  
    style={{}}  
  />  
</TouchableOpacity>  
</View>
```

```
</View>  
);  
};
```

```
export default App;
```