

**IMPLEMENTASI ALGORITMA QUEUE UNTUK SISTEM ANTRIAN
BOOKING BARBERSHOP BERBASIS ANDROID**

SKRIPSI



Disusun Oleh :

RANGGA BAGUS BRATTA

1411010031

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG
2018**

**IMPLEMENTASI ALGORITMA QUEUE UNTUK SISTEM ANTRIAN
BOOKING BARBERSHOP BERBASIS ANDROID**

SKRIPSI



Disusun Oleh :

RANGGA BAGUS BRATTA

1411010031

**JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
INSTITUT INFORMATIKA DAN BISNIS DARMAJAYA
BANDAR LAMPUNG
2018**



PERNYATAAN

Saya yang bertanda tangan dibawah ini, menyatakan bahwa skripsi yang saya buat ini adalah hasil karya saya sendiri, tidak terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu perguruan tinggi atau karya yang pernah ditulis atau diterbitkan orang lain kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka. Karya ini adalah milik saya dan pertanggung jawaban sepenuhnya berada di pundak saya.

Bandar Lampung, 28 September 2018



Rangga Bagus Bratta
NPM. 1411010031

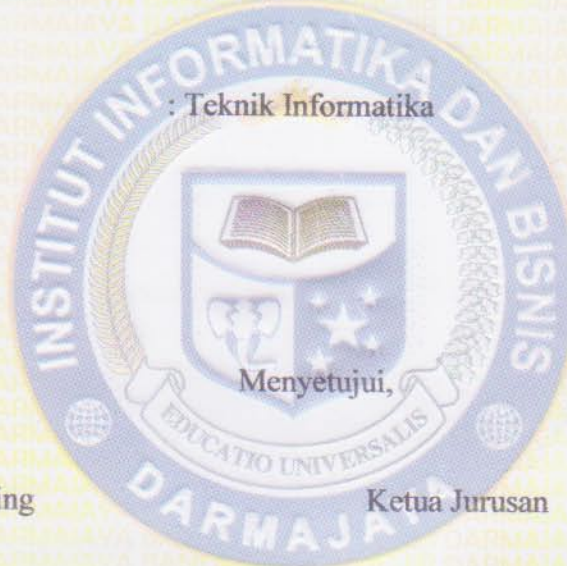
HALAMAN PERSETUJUAN

Judul Skirpsi : **IMPLEMENTASI ALGORITMA QUEUE
UNTUK SISTEM ANTRIAN BOOKING
BARBERSHOP BERBASIS ANDROID**

Nama Mahasiswa : **Rangga Bagus Bratta**

No. Pokok Mahasiswa : **1411010031**

Jurusan : **Teknik Informatika**



Dosen Pembimbing

Ketua Jurusan

Yuni Puspita Sari, S.Kom., M.T.I.
NIK. 12070111

Yuni Arkhiansyah, S.Kom., M.Kom.
NIK. 00480802

HALAMAN PENGESAHAN

Telah Diuji dan Dipertahankan Didepan Tim Penguji Skripsi
Jurusan Teknik Informatika Insitut Informatika dan Bisnis Darmajaya
Bandar Lampung dan Dinyatakan Diterima untuk
Memenuhi Syarat Guna Memperoleh
Gelar Sarjana Komputer

Mengesahkan

1. Tim Penguji

Ketua : Amnah, S.Kom., M.T.I.

Anggota : Yuni Arkhiansyah, S.Kom., M.Kom.

Tanda Tangan



Two handwritten signatures are shown, each with a horizontal dotted line underneath. The first signature is in black ink and the second is in blue ink.

2. Dekan Fakultas Ilmu Komputer



The official stamp of Institut Informatika dan Bisnis Darmajaya is visible. Below it is a handwritten signature in blue ink. Underneath the signature, the name and NIK are printed: **Srivanto, S.Kom., M.M** and **NIK: 00210800**.

Tanggal Lulus Ujian Skripsi : **Kamis, 20 September 2018**

DAFTAR RIWAYAT HIDUP

Penulis dilahirkan di Mekar Jaya, Kec. Bulang Tengah Suku Ulu, Kab. Musi Rawas Sumatera Selatan pada tanggal 07 Juli 1995 putra ke 2 (Dua) dari 2 saudara yang lahir dari pasangan Bapak Sugiyanta, S.Pd. dan Ibu Saminah.

pendidikan formal yang pernah ditempuh oleh penulis antara lain:

1. Sekolah Sekolah Dasar Negeri (SDN) Suka Maju , Bumi Agung Way Kanan Lampung diselesaikan pada tahun 2006.
2. Sekolah Menengah Pertama Negeri (SMPN) 1 Bumi Agung Way Kanan Lampung, diselesaikan pada tahun 2009.
3. Tahun 2012 menyelesaikan Sekolah Menengah Atas Negeri (SMAN) 1 Bumi Agung Way Kanan Lampung.
4. Tahun 2014 terdaftar sebagai mahasiswa aktif pada IIB Darmajaya Bandar Lampung Pada Program Studi Teknik Informatika.

Pada tahun 2017 penulis mengikuti kegiatan Praktek Kerja Pengabdian Masyarakat (PKPM) di Kampung Sri Bawono Kecamatan Way Seputih Kabupaten Lampung Tengah sebagai syarat penulisan Praktek Kerja Pengabdian Masyarakat (PKPM) dan syarat mengambil skripsi sebagai salah satu syarat mencapai gelar sarjana Komputer pada Jurusan Teknik Informatika di perguruan tinggi IIB Darmajaya Bandar Lampung.

PERSEMBAHAN

Semoga hasil karya pikiran ini dapat menjadi persembahan terbaikku untuk :

- ❖ Allah SWT Alhamdulillah, atas segala Nikmat, Rahmat, dan Kekuatan yang senantiasa engkau berikan.
- ❖ Ayahanda Sugiyanta, S.Pd. dan Ibunda Saminah sembah sujud dan hormat ananda, terima kasih atas cinta dan kasih sayangnya, serta do'a dan semangat yang telah ayahanda dan ibunda berikan kepada ananda.
- ❖ Mbaku Aprilia Anggraeni Puspita dan Adikku tercinta satu-satunya Nadira Salsabila, serta seluruh keluarga dan saudara yang telah memberi semangat dan do'a sehingga laporan skripsi ini dapat terselesaikan dengan baik.
- ❖ Teman-teman seperjuangan yang baik, menyenangkan dan selalu membantuku, khususnya angkatan 2014 Teknik Informatika (Sukron Tegar Fauzi, Rama Dwi P, Nesa Nugraha, Iskandar Saputo Bakri, Komang Wijane, Eko prasetyo, Asty, Hafsa Mukaromah, Aditia Rahman, Arif Setiawan, Ulfan, Edi Susanto, Riski Agus S, Pikri Pit Pratama dan Irawanto) dan semua kawan-kawanku yang telah menemaniku dari semester satu sampai sekarang.
- ❖ Untuk Dosen Pembimbing Lapangan PKPM (Ibu Herlina, SE., MM), Pembimbing Skripsi (Ibu Yuni Puspita Sari, S.Kom., M.T.I) dan Ketua Jurusan Teknik Informatika (Bpk. Yuni Arkhiansyah, S.Kom., M.Kom.) dan semua dosen-dosen yang telah memberikanku wawasan, ilmu dan pengetahuan dari semester satu hingga saat ini.
- ❖ Almamaterku tercinta IIB Darmajaya....

MOTTO

“Kunci Mencapai Titik Kesuksesan Itu Adalah Berusaha, Bersabar Dan Terus Berjuang, Karena Tanpa Kesabaran Dan Berusaha Tidak Akan Bisa Menuai Apa Yang Telah Kita Perjuangkan Dengan Sempurna ”

“Bahagia Dan Sukses Itu Ditekad Untuk Tujuan Hidup. Tanpa Tekad Dan Usaha, Kita Tidak Akan Pernah Bisa Mencapai Ke Puncak Kesuksesan”

“Hanya Diri Sendiri Yang Dapat Merubah Dan Menentukan Masa Depan Hidupnya. Oleh Karena Itu Teruslah Berusaha Dan Berdoa, Agar Apa Yang Telah Kita Perjuangkan Dapat Kita Petik Dengan Keberhasilan Dan Kebahagiaan”

ABSTRAK

IMPLEMENTASI ALGORITMA QUEUE UNTUK SISTEM ANTRIAN BOOKING BARBERSHOP BERBASIS ANDROID

Oleh

Rangga Bagus Bratta

1411010031

Kesuksesan suatu bisnis usaha tidak lepas dari peran teknologi informasi sebagai media informasi penunjang layanan dengan salah satu contoh adalah *smartphone* Android. Partner *Babershop* merupakan suatu bisnis usaha pangkas rambut pria yang telah menyediakan layanan jasa pangkas rambut panggilan menggunakan *whatsapp*. Namun, pelayanan melalui *whatsapp* tidak bisa meng-*handle* antrian pelanggan serta memiliki kendala yaitu sulit menentukan pelanggan yang di dahulukan karena pihak *babershop* harus selalu melihat dan mencocokkan jadwal pemesanan para pelanggan. Untuk mengatasi permasalahan tersebut, maka di butuhkanlah suatu sistem yang dapat memproses antrian pemesanan pelanggan berbasis Android. Aplikasi yang dibangun berbasis Android dengan sistem operasi minimum Lollipop. Bahasa pemograman yang digunakan adalah Java dengan *database* MySQL dan editor Android menggunakan Android Studio. Metode pengembangan sistem yang digunakan adalah *prototype*. Jika *client* melakukan pemesanan, maka *client* harus menentukan titik lokasi sehingga memudahkan barber dalam menemukan lokasi *client*. Terdapat kolom *chat* antara barber dengan *client* sehingga memudahkan komunikasi antara *barber* dengan *client*.

Kata Kunci : Babershop, Prototype, Android, Java.

ABSTRACT

IMPLEMENTING ANDROID-BASED APPLICATION USING ALGORITHM QUEUE FOR A QUEUE-BOOKING SYSTEM IN A BARBERSHOP

By

**Rangga Bagus Bratta
1411010031**

The success of a business is inherently caused by the role of information technology as information supporting medium. One of the information supporting mediums is Android Smartphone. Pioneer Barbershop is the haircut business for man using Whatsapp. The problem statements of this research were that the Whatsapp could not handle the customer queue and had the difficulty in determining the first customer. These problems were due to the barbers who must always look and match the customers' ordering schedule. To overcome these problems, an application was needed to design. An application was designed for processing the customer order queues through the Android system. This application was based on Android using Lollipop as a minimum operating system. The programming languages in this system were Java with MySQL database. The Android editor used in this research was the Android Studio. The method to develop this system was Prototype method. This application was able to facilitate the barbers to find out the client location when the clients made an order in the certain point. Moreover, there was also a chat column between the barbers and the clients so that it was able to facilitate the communication between the barbers and the clients.

Keywords: Barbershop, Prototype, Android, Java

ABSTRACT

IMPLEMENTING ANDROID-BASED APPLICATION USING ALGORITHM QUEUE FOR A QUEUE-BOOKING SYSTEM IN A BARBERSHOP

By

Rangga Bagus Bratta
1411010031

The success of a business is inherently caused by the role of information technology as information supporting medium. One of the information supporting mediums is Android Smartphone. Pioneer Barbershop is the haircut business for man using Whatsapp. The problem statements of this research were that the Whatsapp could not handle the customer queue and had the difficulty in determining the first customer. These problems were due to the barbers who must always look and match the customers' ordering schedule. To overcome these problems, an application was needed to design. An application was designed for processing the customer order queues through the Android system. This application was based on Android using Lollipop as a minimum operating system. The programming languages in this system were Java with MySQL database. The Android editor used in this research was the Android Studio. The method to develop this system was Prototype method. This application was able to facilitate the barbers to find out the client location when the clients made an order in the certain point. Moreover, there was also a chat column between the barbers and the clients so that it was able to facilitate the communication between the barbers and the clients.

Keywords: Barbershop, Prototype, Android, Java



PRAKATA

Segala puji syukur saya panjatkan kepada Allah SWT berkat-Nya, sehingga penulis dapat menyelesaikan penyusunan skripsi ini, yang berjudul **“Implementasi Algoritma Queue Untuk Sistem Antrian Booking Barbershop Berbasis Android”**.

Penulis menyadari bahwa dalam penyusunan skripsi ini tidak lepas dari bantuan semua pihak. Oleh karena itu, dalam kesempatan ini penulis menyampaikan terima kasih yang sebesar-besarnya kepada :

1. Bapak DR. Andi Desfiandi, S.E., M.A., selaku Ketua Yayasan Alfian Husin.
2. Bapak Ir. Firmansyah Y. Alfian, MBA., MSc., selaku Rektor IIB Darmajaya.
3. Bapak Dr. RZ. Abdul Aziz, ST., M.T., selaku Wakil Rektor I Bidang Akademik dan Riset Institut Informatika dan Bisnis (IIB) Darmajaya.
4. Bapak Roni Nazar, S.E., M.T., selaku Wakil Rektor II Bidang Administrasi dan Keuangan Institut Informatika dan Bisnis (IIB) Darmajaya.
5. Bapak Muprihan Thaib, S.Sos., M.M., selaku Wakil Rektor III Bidang Akademik dan Sumber Daya Riset Institut Informatika dan Bisnis (IIB) Darmajaya.
6. Bapak Sriyanto, S.Kom., M.M., selaku Dekan Fakultas Ilmu Komputer Institut Informatika dan Bisnis (IIB) Darmajaya.
7. Bapak Yuni Arkhiansyah, S.Kom., M.Kom., selaku Ketua Jurusan Teknik Informatika.

8. Ibu Yuni Puspita Sari, S.Kom., M.T.I., selaku Dosen Pembimbing yang telah membantu membimbing dan mengarahkan serta memberikan petunjuk sehingga skripsi ini dapat diselesaikan dengan baik.
9. Para dosen-dosen jurusan teknik informatika, serta seluruh Dosen, Staf dan Karyawan Institut Informatika dan Bisnis Darmajaya Bandar Lampung yang telah memberi bantuan baik langsung maupun tidak langsung selama saya menjadi mahasiswa.
10. Semua Pihak yang telah memberikan bantuan dan petunjuk sehingga saya dapat lebih mudah dalam menyusun skripsi ini.
11. Almamaterku tercinta.

Demikian banyaknya bantuan berbagai pihak kepada penulis, tentunya tidak menutup kemungkinan bahwa hasil dari laporan ini masih ada kekurangan dan masih jauh dari taraf sempurna. Oleh karena itu, kritik dan saran guna perbaikan di masa depan adalah mutlak sangat penulis perlukan. Semoga Laporan Skripsi ini bermanfaat bagi setiap pembacanya.

Bandar Lampung, 28 September 2018
Penulis,

Rangga Bagus Bratta
NPM. 1411010031

DAFTAR ISI

JUDUL LAPORAN	i
SURAT PERNYATAAN	ii
HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
RIWAYAT HIDUP	v
PERSEMBAHAN	vi
MOTTO	vii
ABSTRAK	viii
KATA PENGANTAR	x
DAFTAR ISI	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi

BAB 1 PENDAHULUAN

1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah.....	2
1.3 Ruang Lingkup Penelitian	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
1.6 Sistematika Penelitian	3

BAB 11 LANDASAN TEORI

2.1 Algoritma <i>Queue</i>	5
2.2 Pemesanan (<i>Booking</i>)	6
2.3 Barbershop.....	6
2.4 Android	7
2.5 Google Maps API.....	8
2.6 Metode Pengembangan Sistem.....	9
2.7 Alat Pengembangan Sistem.....	10
2.7.1 <i>Use Case Diagram</i>	10
2.7.2 <i>Activity Diagram</i>	12
2.7.1 <i>Class Diagram</i>	13
2.8 Android Studio	14
2.9 MySQL	14
2.10 Pengujian Sistem	15
2.11 Penelitian Terkait.....	15

BAB 11I METODE PENELITIAN

3.1 Waktu dan Tempat Penelitian	17
3.2 Alat Pendukung Pengembangan Sistem.....	17
3.3 Metode Pengumpulan Data	18
3.4 Metode Pengembangan Sistem/ Perangkat Lunak	18
3.5 Tahap Pemodelan	19
3.5.1 <i>Use Case Diagram</i>	20
3.5.2 <i>Activity Diagram</i>	20
3.5.3 <i>Class Diagram</i>	25

3.5.4 Perancangan <i>User Interface</i>	28
3.5.4.1 <i>User Interface Client</i>	28
3.5.4.2 <i>User Interface Barber</i>	31

BAB 1V HASIL DAN PEMBAHASAN

4.1 Hasil.....	34
4.2 Pembahasan	34
4.2.1 Implementasi Sistem	34
4.2.1.1 Implementasi Barber	35
4.2.1.2 Implementasi <i>Client</i>	40
4.2.2 Pengujian Sistem.....	44
4.2 Kelebihan dan Kekurangan	47
4.2.1 Kelebihan.....	47
4.2.2 Kekurangan.....	47

BAB V SIMPULAN DAN SARAN

4.1 Simpulan.....	48
4.2 Saran	48

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

Tabel 2. 1 Versi Android	7
Tabel 2. 2 Simbol <i>Use Case Diagram</i>	11
Tabel 2. 3 Simbol Diagram Aktivitas.....	12
Tabel 2. 4 Simbol <i>Class Diagram</i>	13
Tabel 2. 5 Penelitian Terkait	15
Tabel 3. 1 Kamus Data Tabel <i>Client</i>	26
Tabel 3. 2 Kamus Data Tabel Pesan.....	26
Tabel 3. 3 Kamus Data Tabel <i>Chat</i>	27
Tabel 3. 4 Kamus Data Tabel <i>Barber</i>	27
Tabel 3. 5 Kamus Data Tabel Model	27
Tabel 4 1 Hasil Pengujian Versi Android Beserta <i>Interface</i>	45

DAFTAR GAMBAR

Gambar 2. 1	Pengertian depan dan belakang dalam antrian.....	5
Gambar 2. 2	Ilustrasi model prototipe	9
Gambar 3. 1	<i>Use case diagram</i> sistem diusulkan.....	20
Gambar 3. 2	Perancangan <i>activity diagram login</i>	21
Gambar 3. 3	Perancangan <i>activity diagram</i> registrasi <i>client</i>	22
Gambar 3. 4	Perancangan <i>activity diagram</i> posting data model rambut.....	22
Gambar 3. 5	Perancangan <i>activity diagram</i> informasi model rambut.....	23
Gambar 3. 6	Perancangan <i>activity diagram</i> pesan layanan pangkas rambut.....	24
Gambar 3. 7	Perancangan <i>activity diagram chat</i>	24
Gambar 3. 8	Perancangan <i>class diagram</i> sistem diusulkan.....	25
Gambar 3. 9	Perancangan <i>user interface</i> halaman registrasi <i>client</i>	28
Gambar 3. 10	Perancangan <i>user interface</i> halaman <i>login</i>	29
Gambar 3. 11	Perancangan <i>user interface</i> halaman utama <i>client</i>	29
Gambar 3. 12	Perancangan <i>user interface</i> pesan layanan pangkas rambut.....	30
Gambar 3. 13	Perancangan <i>user interface</i> halaman pesanan	30
Gambar 3. 14	Perancangan <i>user interface</i> halaman instruksi	31
Gambar 3. 15	Perancangan <i>user interface</i> halaman tentang	31
Gambar 3. 16	Perancangan <i>user interface login</i> barber	31
Gambar 3. 17	Perancangan <i>user interface</i> halaman utama barber	32
Gambar 3. 18	Perancangan <i>user interface</i> halaman daftar pelanggan barber	32
Gambar 3. 19	Perancangan <i>user interface</i> halaman <i>input</i> model rambut.....	33
Gambar 3. 20	Perancangan <i>user interface</i> halaman tentang barber	33
Gambar 4. 1	Implementasi halaman <i>login</i> barber.....	35
Gambar 4. 2	Implementasi halaman utama barber	36
Gambar 4. 3	Implementasi halaman pesan barber.....	37
Gambar 4. 4	Implementasi halaman informasi detail <i>client</i>	37
Gambar 4. 5	Implementasi halaman <i>chat</i> barber dengan <i>client</i>	38
Gambar 4. 6	Implementasi halaman model	38
Gambar 4. 7	Implementasi halaman <i>input</i> data model	39
Gambar 4. 8	Implementasi halaman tentang yang tersedia hak akses barber	39

Gambar 4. 9 Implementasi halaman <i>registrasi client</i>	40
Gambar 4. 10 Implementasi halaman <i>login client</i>	41
Gambar 4. 11 Implementasi halaman utama <i>client</i>	41
Gambar 4. 12 Implementasi halaman pemesanan pangkas rambut.....	42
Gambar 4. 13 Implementasi halaman informasi pemesanan pangkas rambut	43
Gambar 4. 14 Implementasi halaman instruksi penggunaan aplikasi oleh <i>client</i>	43
Gambar 4. 15 Implementasi halaman tentang aplikasi oleh <i>client</i>	44

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Para pelaku bisnis harus mampu menciptakan ide-ide baru agar dapat memberikan nilai lebih kepada konsumen. Selain itu, pelaku bisnis juga harus mampu untuk dapat melihat peluang bisnis yang sedang berkembang. Salah satu bisnis yang sedang berkembang saat ini adalah bisnis jasa potong rambut pada pria (*babershop*). Kesuksesan suatu bisnis usaha tidak lepas dari peran teknologi informasi sebagai media informasi penunjang layanan yang tersedia dalam proses bisnis. Salah satu contoh teknologi informasi yang banyak digunakan dalam bidang bisnis atau usaha adalah *smartphone* Android. Menurut Oik Yusuf dalam artikel yang tertulis di <https://tekno.kompas.com> dengan judul "Populasi Android Lampau Angka 2 Miliar" populasi Android tercatat baru menyentuh kisaran 1,4 miliar perangkat. Pangsa pasar Android pada November 2016 tercatat mencapai 88 persen, iOS besutan Apple untuk gadget iPhone dan iPad, hanya mendapat porsi 12,1 persen.

Partner *Babershop* merupakan suatu bisnis usaha pangkas rambut pria yang terletak di Tanjung Senang, Way Kandis Bandar Lampung. *Babershop* ini memiliki banyak pelanggan tetap dari segala jenis umur, mulai dari anak-anak hingga orang tua. Biasanya pelanggan yang datang harus mengantri untuk dapat memangkas rambutnya di *babershop* ini. Untuk meningkatkan pelayanan, Partner *Babershop* saat ini telah menyediakan layanan jasa pangkas rambut ditempat (pelanggan tidak harus datang ke *babershop* untuk memangkas rambutnya) menggunakan fasilitas media sosial *whatsapp*. Namun, pelayanan melalui media sosial *whatsapp* tidak bisa meng-*handle* antrian pelanggan Partner *Babershop* yang menginginkan jasa pelayanan ini. Layanan ini memiliki beberapa kendala, diantaranya adalah susahya menentukan pelanggan yang didahulukan karena pihak *babershop* harus selalu melihat dan mencocokkan jadwal pemesanan para pelanggan.

Terkadang pihak *babershop* menemukan jadwal yang sama di hari yang sama dengan lokasi pelanggan yang berbeda. Untuk mengatasi permasalahan tersebut, maka dibutuhkanlah suatu sistem yang dapat memproses antrian pemesanan pelanggan berbasis Android.

Dengan adanya pelayanan pangkas rambut Partner *Babershop* berbasis *smartphone* Android, para pelanggan dapat melihat jenis-jenis model rambut yang tersedia. Pelanggan juga dapat melakukan *order/booking* potong rambut berdasarkan model yang diinginkan dan memasukkan *map* lokasi pelanggan sehingga memudahkan barber menuju lokasi pelanggan. Di dalam aplikasi ini juga sudah terdapat sistem antrian menggunakan algoritma *queue* yang memudahkan barber dalam mengolah antrian pelanggan yang menggunakan jasa layanan ini.

1.2 Perumusan Masalah

Dari latar belakang masalah yang telah diuraikan sebelumnya, maka dapat dirumuskan masalah sebagai berikut :

- a. Bagaimana mengatasi pemesanan pelanggan dengan antrian booking pada *babershop* di hari yang sama?
- b. Bagaimana menentukan pelanggan yang didahulukan dalam proses antrian dengan algoritma *queue*?
- c. Sistem ini khusus untuk area Bandar Lampung.

1.3 Ruang Lingkup Penelitian

Penelitian dilakukan di Partner *Babershop* yang berlokasi di Tanjung Senang, Way Kandis Bandar Lampung dengan batasan penelitian sebagai berikut :

- a. Sistem antrian dibangun menggunakan algoritma *queue* berbasis Android.
- b. Pelayanan hanya dibatasi pada 7 orang pelanggan dalam satu hari.
- c. Sistem yang dibangun berjalan di Android versi 6.0 (Lollipop) ke atas.

1.4 Tujuan

Tujuan dari penelitian ini adalah membangun perangkat lunak sistem antrian pemesanan yang dapat membantu pelanggan untuk memesan maupun mendapatkan informasi antrian secara *realtime* dengan menggunakan algoritma *queue* berbasis Android.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut :

1. Pengguna dapat mendapatkan informasi mengenai detail informasi *barbershop* pada *smartphone* Android berupa pemesanan pangkas rambut, jam operasional *barbershop*, dan jumlah antrian *barbershop*.
2. Pengguna tidak perlu datang langsung ke *barbershop*, karena adanya sistem pelayanan panggilan *babershop*.
3. Meningkatkan fasilitas pelayanan pangkas rambut Partner *Babershop*.

1.6 Sistematika Penulisan

Agar memudahkan dalam memberikan gambaran secara utuh penulisan skripsi ini, maka sistematika penulisan ini dibagi menjadi 5 (lima) bab sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan latar belakang dibuatnya aplikasi sistem antrian booking *barbershop*, rumusan masalah yang didapat, batasan masalah yang dibuat, tujuan, manfaat dan sistematika penulisan yang diterapkan.

BAB II LANDASAN TEORI

Bab ini menjelaskan mengenai teori-teori terkait aplikasi sistem antrian booking *barbershop*, untuk mendukung penelitian yang dilaksanakan.

BAB III METODOLOGI PENELITIAN

Bab ini berisi metode yang digunakan dalam penyelesaian permasalahan yang ditanyakan dalam perumusan masalah dan analisa yang dilakukan dalam membangun aplikasi sistem antrian pemesanan *barbershop*. Metode penelitian

terdiri dari metode pengumpulan data dan metode pengembangan perangkat lunak/sistem.

Selain itu, bab ini membahas prosedur sistem baru yang diajukan menggunakan pemodelan *use case diagram*, *activity diagram*, class diagram dan perancangan *user interface*.

BAB IV HASIL PENELITIAN DAN PEMBAHASAN

Bab ini menjelaskan tentang hasil dari pengkodean yang dilakukan, sehingga yang dibahas pada bab ini adalah bagaimana tampilan sistem saat dijalankan. Selanjutnya dipaparkan tentang instalasi perangkat lunak dan bagaimana sistem ini diuji.

BAB V SIMPULAN DAN SARAN

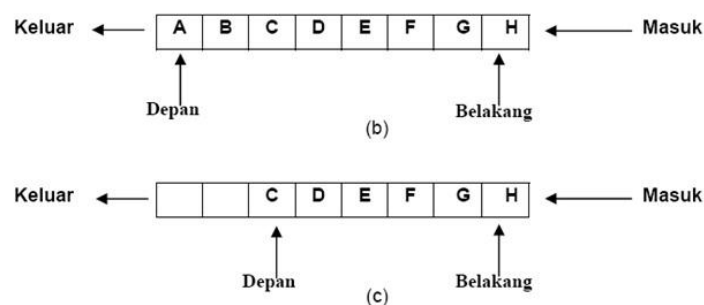
Bab ini berisi tentang kesimpulan dan saran yang diharapkan dapat bermanfaat untuk pengembangan aplikasi sistem antrian *booking barbershop* berbasis Android selanjutnya.

BAB II

LANDASAN TEORI

2.1 Algoritma *Queue*

Antrian (*queue*) adalah struktur data yang mempunyai sifat yakni data yang masuk dulu akan diproses lebih dulu, Dengan kata lain, struktur data ini mempunyai sifat yang dinamakan FIFO (*First-in first out*). Jadi, struktur data ini mencerminkan konsep antrian orang dalam dunia nyata. Antrian (*queue*) adalah struktur data yang mempunyai sifat yakni data yang masuk dulu akan diproses lebih dulu. Jadi, struktur data ini mencerminkan konsep antrian orang dalam dunia nyata (Kadir, 2013).



Gambar 2.1 Pengertian depan dan belakang dalam antrian (Sumber :

<http://darkzone7.blogspot.co.id/2013/04/struktur-organisasi-data-2-queue-antrean.html>)

Gambar 2.1(b) memperlihatkan keadaan ketika terdapat delapan item yang sedang antri. Depan menunjukan pada A yang menyatakan data yang akan dilayani duluan dan Belakang menunjukan ke H yang menyatakan item yang terakhir kali masuk dalam antrian. Apabila kemudian A dilayani maka A tidak lagi berada pada antrian. Hal ini digambarkan pada 2.1(c). Dalam hal ini, depan sekarang menunjuk ke C. Sedangkan ketika C belum dilayani ternyata ada dua item yaitu D dan E yang ikut antri. Operasi dasar pada antrian berupa *insert* dan *remove*.

1. *Insert* menyatakan operasi untuk memasukkan data ke dalam antrian. Operasi ini biasa dinyatakan *insert* (A, d) atau kadang disingkat *insert* (d). Dalam hal ini, A menyatakan antrian dan d menyatakan item data yang dimasukkan ke dalam antrian A. Istilah enqueueer terkadang digunakan untuk menyatakan operasi pemasukan data ke dalam antrian.
2. *Remove* menyatakan operasi untuk pengambilan data dari antrian. Operasi ini biasanya dinyatakan dengan *remove* (A) atau *remove* () saja. Dalam hal ini, data yang berada di depan dalam antrian A akan dikeluarkan dan menjadi nilai balik *remove* (). Itulah sebabnya, penggunaan *remove* dituangkan dalam bentuk pernyataan.

2.2 Pemesanan (*Booking*)

Booking dapat diartikan juga sebagai reservasi. *Booking* adalah sebuah proses perjanjian berupa pemesanan sebuah produk baik barang maupun jasa dimana pada saat itu telah terdapat kesepakatan antara konsumen dengan produsen mengenai produk tersebut namun belum ditutup oleh sebuah transaksi jual-beli (Zia'ulhaq, 2014). Pada saat *booking* berlangsung biasanya ditandai dengan adanya proses tukar menukar informasi antara konsumen dan produsen agar kesepakatan mengenai produk dapat terwujud.

2.3 *Barbershop*

Pelayanan jasa perawatan pria sudah ada sejak 2000 tahun yang lalu. Praktek pemotongan rambut pada pria (*barbershop*) berawal dari Wilayah Macedonia sekitar 400 tahun sebelum masehi lalu menyebar ke Mesir dan daerah-daerah lainnya. Kata "barber" berasal dari bahasa latin "barba" yang artinya janggut. Bangsa pertama yang mengklaim dirinya paling ahli dalam jasa pelayanan pemotongan rambut adalah bangsa Roma sekitar 296 tahun sebelum masehi. Akan tetapi baik pada bangsa Roma maupun Mesir, *Barbershop* memiliki reputasi yang kurang baik karena orang-orang elit kelas atas pada waktu itu memiliki tukang cukur pribadi. Pada masa itu, janggut pada lelaki menjadi

symbol kekuatan dan intelegensi sehingga harus dirawat dengan baik dan teratur (Hendra Cipta).

Potong rambut mulai mendapatkan pengaturannya di Indonesia sejak berakhirnya masa kerajaan karena di jaman kerajaan dahulu kehormatan seseorang bisa dilihat jika ia berambut panjang, dan hal itu mulai bergeser setelah masuknya pengaruh barat, seperti agama Islam, Kristen dan Budha. Bila sebelumnya rambut panjang dikaitkan dengan kedewasaan serta kekuatan spiritual seseorang, masuknya pengaruh tersebut, menjadikan rambut sebagai penanda seksualitas seseorang. Artinya, terjadi pergeseran pandangan pada persoalan seksualitas yang menekankan pada pengekangan seksual dan perbedaan antara perempuan dan laki-laki (IGA Yudi Artha K).

2.4 Android

Menurut Yuni Puspita Sari (2016) android adalah sebuah sistem operasi untuk *smartphone* dan tablet. Sistem operasi dapat diilustrasikan sebagai 'jembatan' antara piranti (*device*) dan penggunanya, sehingga pengguna bisa berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*.

Menurut Septilia Arfida, dkk (2017) android merupakan sistem operasi *smartphone* yang sangat populer karena bersifat open source yang menjadi magnet bagi para developer untuk mengembangkan aplikasi-aplikasinya.

Google menyediakan peningkatan versi bertahap utama untuk sistem operasi Android setiap enam hingga sembilan bulan, menggunakan nama bertema makanan. Adapun versi-versi sistem operasi Android adalah seperti pada Tabel 2.1.

Tabel 2.1 Versi Android

Nama Kode	Nomor Versi	Tanggal Rilis Awal	API Level
N/A	1.0	23 September 2008	1
N/A	1.1	9 Febuari 2009	2
Cupkace	1.5	27 April 2009	3
Donut	1.6	15 September 2009	4
Eclair	2.0-2.1	26 Oktober 2009	5-7
Froyo	2.2-2.2.3	20 Mei 2010	8
Gingerbread	2.3-2.3.7	6 Desember 2010	9-10

Tabel 2.1 Versi Android (Lanjutan)

Honeycomb	3.0-3.2.6	22 Febuari 2011	11-13
Ice Cream Sandwitch	4.0-4.0.4	18 Oktober 2011	14-15
Jelly Bean	4.1-4.3.1	9 Juli 2012	16-18
KitKat	4.4-4.4.4	31 Oktober 2013	19-20
Lolipop	6.0-6.0.1	5 Oktober 2015	23
Nougat	7.0	22 Agustus 2016	24
Oreo	8.0	21 Agustus 2017	26

Untuk membantu mengembangkan aplikasi secara efisien, Google menawarkan Lingkungan *Development* Terintegrasi (IDE) Java Lengkap yang disebut Android Studio, dengan fitur lanjutan untuk pengembangan, *debug*, dan pemaketan aplikasi Android. Pendistribusian aplikasi Android dapat dilakukan dengan berbagai cara, yaitu melalui email, situs web, atau pasar aplikasi Google Play. Google Play adalah layanan distribusi digital yang dioperasikan dan dikembangkan oleh Google dan berfungsi sebagai toko aplikasi resmi untuk Android yang memungkinkan konsumen mengunduh dan menjelajah aplikasi yang dikembangkan dengan Android SDK dan dipublikasikan melalui Google.

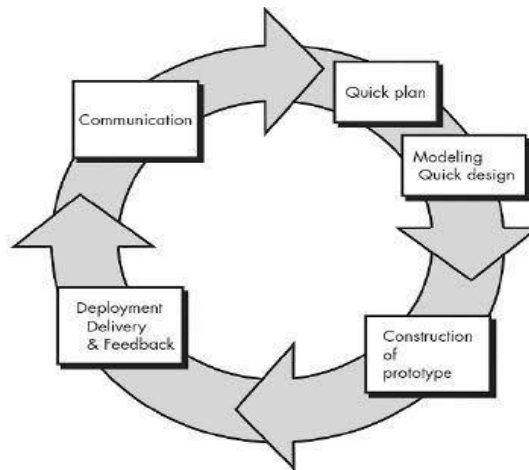
2.5 Google Maps API

Ardhy (2018) menguraikan bahwa Google Maps adalah sebuah jasa peta globe virtual gratis dan online disediakan oleh Google dan dapat ditemukan di <http://maps.google.com/>. Google Maps menawarkan peta yang dapat diseret dan gambar satelit untuk seluruh dunia dan baru-baru ini juga menawarkan perencanaan rute dan pencari letak bisnis di U.S., Kanada, Jepang, Hong Kong,

Cina, UK, Irlandia (hanya pusat kota) dan beberapa bagian Eropa. Google Maps API merupakan aplikasi interface yang dapat diakses lewat javascript agar Google Maps dapat ditampilkan pada halaman web yang sedang kita bangun. Untuk dapat mengakses Google Maps, Kita harus melakukan pendaftaran API Key terlebih dahulu dengan data pendaftaran berupa nama domain web yang kita bangun.

2.6 Metode Pengembangan Sistem Prototipe

Metode prototipe suatu proses pembuatan *software* yang bersifat berulang dan dengan perencanaan yang cepat yang dimana terdapat umpan balik yang memungkinkan terjadinya perulangan dan perbaikan *software* sampai dengan *software* tersebut memenuhi kebutuhan dari sisi pengguna.



Gambar 2.2 Ilustrasi model prototipe (Sumber : Pressman, 2012)

Pembuatan prototipe dimulai dengan dilakukannya komunikasi antara tim pengembang perangkat lunak dengan para pelanggan. Tim pengembang perangkat lunak akan melakukan pertemuan-pertemuan dengan para *stakeholder* untuk mendefinisikan sasaran keseluruhan untuk perangkat lunak yang akan dikembangkan, mengidentifikasi spesifikasi kebutuhan apa pun

yang saat ini diketahui, dan menggambarkan area-area dimana definisi lebih jauh pada iterasi selanjutnya merupakan keharusan. Iterasi pembuatan prototipe direncanakan dengan cepat dan pemodelan (dalam bentuk "rancangan cepat") dilakukan. Suatu rancangan cepat berfokus pada representasi semua aspek perangkat lunak yang akan dilihat oleh para pengguna akhir (misalnya rancangan antar muka pengguna (*user interface*) atau format tampilan). Rancangan cepat (*quick design*) akan memulai konstruksi pembuatan prototipe.

Prototipe kemudian akan diserahkan kepada para stakholder dan kemudian mereka akan melakukan evaluasi-evaluasi tertentu terhadap prototipe yang telah dilakukan sebelumnya, kemudian akhirnya akan memberikan umpan-balik yang akan digunakan untuk memperhalus spesifikasi kebutuhan. Iterasi akan terjadi saat prototipe diperbaiki untuk memenuhi kebutuhan dari para *stakeholder*, sementara pada saat yang sama memungkinkan kita untuk lebih memahami kebutuhan apa yang akan dikerjakan pada saat iterasi selanjutnya.

Idealnya, prototipe bertindak sebagai mekanisme untuk mengidentifikasi spesifikasi-spesifikasi kebutuhan perangkat lunak. Jika suatu prototipe yang dapat digunakan akan dikembangkan, kita bisa menggunakan program yang sudah ada sebelumnya atau dengan menerapkan penggunaan perkakas yang sudah ada misalnya perkakas pembentuk laporan (*report generator*) atau aplikasi untuk melakukan perancangan antarmuka (*window manager*) yang memungkinkan program yang dapat digunakan dapat dibuat dengan mudah dan cepat.




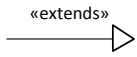

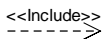
2.7 Alat Pengembangan Sistem

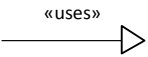
Alat pengembangan sistem yang digunakan menggunakan pemodelan UML (*Unified Modeling Language*) *use case diagram*, *activity diagram* dan *class diagram*.

2.7.1 Use Case Diagram

Use case diagram adalah gambar dari beberapa atau seluruh aktor dan *use case* dengan tujuan yang mengenali interaksi mereka dalam suatu sistem. *Use case* diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem, yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use case* mempresentasikan sebuah interaksi antara *actor* dan sistem. Dalam *use case* diagram terdapat istilah seperti aktor, *use case* dan *case relationship* (Rosa, 2016). Penjelasan simbol *use case* diagram ditunjukkan pada tabel 2.2.

Tabel 2.2 Simbol *Use Case Diagram*

Keterangan	Simbol	Deskripsi
<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal-awal frase nama <i>use case</i>
Aktor		Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar itu sendiri. Jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor.
Asosiasi		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
Ekstensi		Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> , dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek.
Generalisasi		Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
Menggunakan/include/use		Ada dua sudut pandang yang cukup besar mengenai <i>include</i> di <i>use case</i> : a. <i>Include</i> berarti <i>use case</i> yang ditambahkan akan selalu dipanggil saat <i>use case</i>


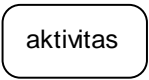


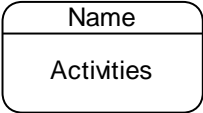

		<p>tambahan dijalankan</p> <p>b. Include berarti use case yang tambahan akan selalu melakukan pengecekan apakah use case yang ditambahkan telah dijalankan sebelum use case tambahan dijalankan.</p>
--	---	--

2.7.2 Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. Diagram aktivitas juga banyak digunakan untuk mendefinisikan hal-hal berikut (Rosa, 2016):

- a. Rancangan proses bisnis dimana setiap urutan aktivitas yang digambarkan merupakan proses bisnis sistem yang didefinisikan.
- b. Urutan atau pengelompokan tampilan dari sistem/*user interface* dimana setiap aktivitas dianggap memiliki sebuah rancangan antarmuka tampilan.
- c. Rancangan pengujian dimana setiap aktivitas dianggap memerlukan sebuah pengujian yang perlu didefinisikan kasus ujinya.

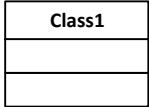
Tabel 2.3 Simbol Diagram Aktivitas

Keterangan	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
Swimlane		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.7.3 Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. kelas memiliki apa yang disebut atribut dan metode atau operasi. Atribut merupakan variabel-variabel yang dimiliki suatu kelas, sedangkan operasi atau metode adalah fungsi-fungsi yang dimiliki oleh suatu kelas (Rosa, 2016).

Tabel 2.4 Simbol Class Diagram

Simbol	Deskripsi
Kelas 	Kelas pada struktur sistem.

Natarmuka/ <i>interface</i> Interface2 ○—	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
Asosiasi —————	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah —————>	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Generalisasi —————>	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
Kebergantungan ----->	Relasi antar kelas dengan makna kebergantungan antar kelas.
Agregasi —————◇	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.8 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) untuk *platform* Android. Android Studio ini diumumkan pada tanggal 16 Mei 2013 pada Konferensi Google I/O oleh Produk Manajer Google, Ellie Powers. Android Studio bersifat *free* dibawah *Apache License 2.0*. Android studio awalnya dimulai dengan versi 0.1 pada bulan mei 2013, Kemudian dibuat versi beta 0.8 yang dirilis pada bulan juni 2014. Yang paling stabil dirilis pada bulan Desember 2014, dimulai dari versi 1.0. Berbasiskan JetBrains' IntelliJ IDEA, Studio didesain khusus untuk Android Development yang kini sudah bisa di download untuk Windows, Mac OS X, dan Linux (Eric, 2016).

2.9 MySQL

Menurut Solichin (2016), MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di

bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL. Tidak seperti Apache yang merupakan *software* yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia yaitu MySQL AB. MySQL AB memegang penuh hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius. Beberapa kelebihan MySQL antara lain : *free* (bebas di *download*), stabil dan tangguh, fleksibel dengan berbagai pemrograman, security yang baik, dukungan dari banyak komunitas, kemudahan management *database*, mendukung transaksi dan perkembangan *software* yang cukup cepat.

2.10 Pengujian Sistem

Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. *Black Box Testing* bukanlah solusi alternatif dari *White Box Testing* tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh *White Box Testing* (Sidi, M Mustaqbal, 2015). *Black Box Testing* cenderung untuk menemukan hal-hal berikut :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.11 Penelitian Terkait

Penelitian yang terkait dengan penelitian yang sudah dilakukan sebelumnya adalah seperti yang tertera pada Tabel 2.5.

Tabel 2.5 Penelitian Terkait

No.	Nama	Judul	Terbit	Uraian
1.	Reni Nursyanti dan Vani Maharani Nasution.	Aplikasi Barbershop Berbasis Android Studi Kasus Malebox Barbershop.	Universitas Informatika dan Bisnis Indonesia, September 2017	Aplikasi smartphone ini dibangun untuk mempermudah transaksi dan mempermudah customer dalam melakukan pemesanan barbershop.

Tabel 2.5 Penelitian Terkait (Lanjutan)

2.	Muhammad Zia'ulhaq	Pembangunan Perangkat Lunak Online Booking Barbershop Di Bandung Menggunakan Teknologi Mobile Global Positioning System Dan Web Services Pada Platform Android	Universitas Dian Nuswantoro, Semarang 2014	Membangun perangkat lunak yang dapat memberikan informasi lebih mengenai layanan barbershop yang berada di kota Bandung dan dapat melakukan online booking barbershop yang di aplikasikan di perangkat mobile.
3.	IGA Yudi Artha K. , I Wayan Swandi, dan Alvin Raditya	Perancangan Media Komunikasi Visual Gangs Barbershop di Bali	Universitas Kristen Petra Surabaya, September 2014	Membuat promosi Gangs Barbershop melalui Perancangan Media Komunikasi Visual sebagai sarana promosi yang efektif dan menarik.

BAB III

METODOLOGI PENELITIAN

Metode penelitian menggambarkan alur atau tahapan-tahapan pembuatan perangkat lunak antrian *booking babershop* berbasis Android menggunakan algoritma queue. Adapun metode pelaksanaan pembuatan sistem tersebut, dijelaskan pada sub-sub penjelasan dibawah ini.

3.1 Waktu dan Tempat Penelitian

Penelitian dilakukan di Partner *Babershop* yang berlokasi di Tanjung Senang, Way Kandis Bandar Lampung, dengan waktu penelitian dimulai dari 26 Juli 2018 sampai dengan 31 Agustus 2018.

3.2 Alat Pendukung Pengembangan Sistem

Dalam pembuatan perangkat lunak antrian *booking babershop* berbasis Android menggunakan algoritma queue terdapat beberapa perangkat lunak dan perangkat keras yang digunakan untuk mendukung proses sistem informasi, yaitu:

a. Perangkat Keras

Perangkat keras yang digunakan dalam pembuatan perangkat lunak antrian *booking babershop* berbasis Android menggunakan algoritma queue adalah satu unit komputer atau laptop dengan spesifikasi minimal sebagai berikut :

1. *Processor Core i3*
2. *Harddisk 500 GB*
3. *RAM 4 GB*

b. Perangkat Lunak

Perangkat lunak yang digunakan dalam pembuatan perangkat lunak antrian *booking babershop* berbasis Android menggunakan algoritma queue adalah sebagai berikut :

1. Sistem Operasi *Windows* 10 64bit
2. *Database MySQL*
3. *Tools* yang digunakan adalah *Android Studio*, *Edraw Max* dan *Ms. Word*.

3.3 Metode Pengumpulan Data

Pengumpulan data yang digunakan dalam menyusun serta melengkapi data adalah dengan cara observasi, wawancara dan studi pustaka.

a. Observasi

Pengamatan langsung diadakan untuk memperoleh data yang dilakukan pada instansi terkait dengan penelitian yang dilakukan di Partner *Babershop*. Pengamatan dilakukan guna mendapatkan informasi mengenai fasilitas dan harga layanan serta pelayanan yang tersedia di Partner *Babershop*.

b. Wawancara

Wawancara dilakukan dengan cara berkomunikasi langsung dengan Bapak Trio Wanardi selaku pemilik Partner *Babershop*.

c. Studi Pustaka

Studi pustaka dilakukan untuk memperoleh data dan informasi dengan membaca berbagai bahan penulisan, karangan ilmiah serta sumber-sumber lain mengenai permasalahan yang berhubungan dengan penulisan. Data yang di peroleh berupa jurnal penelitian yang terkait dengan peelitian yang dilakukan.

3.4 Metode Pengembangan Sistem/Perangkat Lunak

Metode pengembangan sistem yang digunakan dalam pembuatan perangkat lunak antrian *booking babershop* berbasis *Android* menggunakan algoritma *queue* adalah model prototipe. Adapun tahapan pengembangan sistem menggunakan model prototipe adalah sebagai berikut :

a. *Deployment*

Pada tahap ini, *client* yang ingin membuat sebuah sistem datang kepada pengembang sistem untuk meminta dibuatkan sebuah sistem dengan menginformasikan beberapa kebutuhan sistem. Sistem yang dibutuhkan adalah membuat perangkat lunak sistem antrian pemesanan pangkas rambut pria di Partner *Babershop* berbasis Android demi meningkatkan pelayanan Partner *Babershop*.

b. *Communication*

Selanjutnya, pada tahap ini pengembang *software* mengkomunikasikan kepada pembuat *software* bahwa *client* ingin membuat sebuah *software* yang memiliki kemampuan sesuai dengan yang diinginkan *client*.

c. *Planning*

Pada tahap *planning*, *client* merencanakan atau membuat perencanaan mengenai apa yang ingin dibuat dan apa saja kebutuhan sistem yang diinginkan tersebut.

d. *Modeling*

Setelah melakukan *planning*, maka pembuat *software*/sistem melakukan perancangan model seperti model *form* awal, maupun perancangan model *form-form* selanjutnya. Tahapan ini meliputi perancangan *use case diagram*, *activity diagram*, *class diagram* dan *user interface*.

e. *Construction*

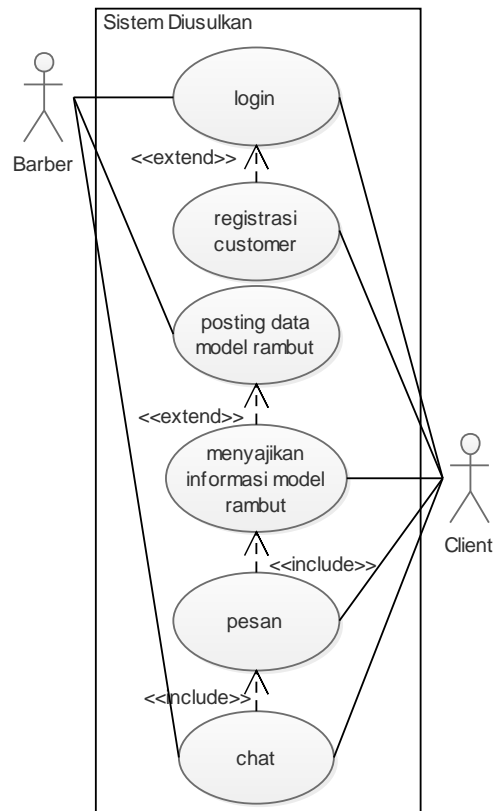
Tahap ini merupakan tahap akhir dalam model prototipe. Disini pembuat *software* melakukan pengodingan sesuai dengan perancangan model dari suatu sistem yang ingin dibuat.

3.5 Tahapan Pemodelan

Tahapan ini lebih mengarah kepada pemodelan/perancangan perangkat lunak yang terdiri dari *use case diagram*, *activity diagram*, *clas diagram* dan perancangan tatap muka.

3.5.1 Use Case Diagram

Use case diagram mempresentasikan sebuah interaksi antara *actor* dan sistem. Perancangan *use case diagram* sistem yang diusulkan adalah seperti pada Gambar 3.1.



Gambar 3.1 *Use case diagram* sistem diusulkan

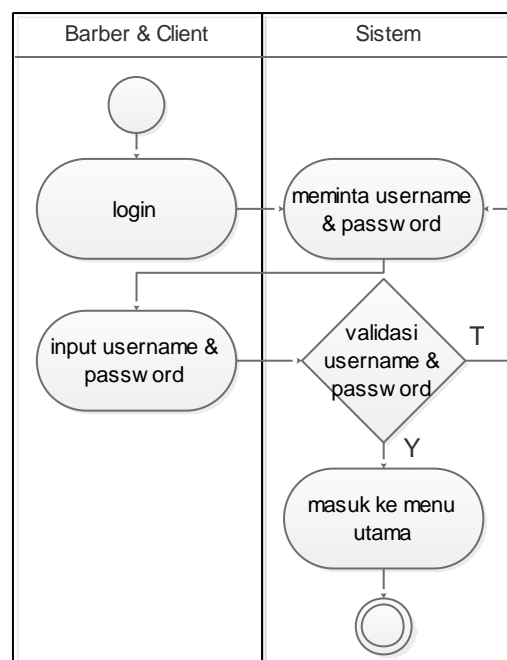
3.5.2 Activity Diagram

Activity diagram menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Perancangan *activity diagram* perangkat lunak antrian *booking babershop* berbasis Android menggunakan algoritma queue adalah sebagai berikut :

a. *Activity Diagram Login*

Activity diagram login menggambarkan aktifitas barber dan *client* dalam melakukan *login* untuk dapat masuk ke dalam sistem dengan cara memasukkan data *username* dan *password*. Perancangan *activity diagram login* adalah seperti pada Gambar 3.2.

Nama *Use Case* : *Login*
Aktor : Barber dan *Client*
Tujuan : Mengakses sistem

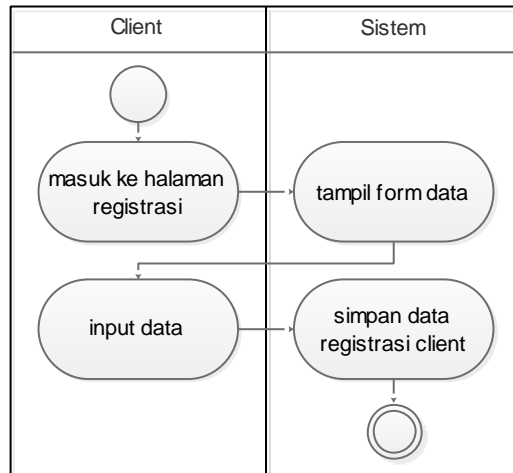


Gambar 3.2 Perancangan *activity diagram login*

b. *Activity Diagram Registrasi Client*

Activity diagram registrasi client menggambarkan aktifitas *client* dalam melakukan registrasi pendaftaran *client*. Perancangan *activity diagram registrasi client* adalah seperti pada Gambar 3.3.

Nama *Use Case* : *Registrasi Client*
Aktor : *Client*
Tujuan : Mendapatkan *login*



Gambar 3.3 Perancangan *activity diagram* registrasi *client*

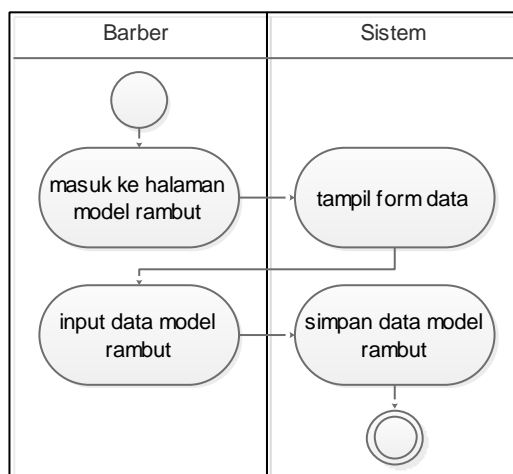
c. *Activity Diagram Posting Data Model Rambut*

Activity diagram posting data model rambut menggambarkan aktifitas *barber* dalam memasukkan data informasi mengenai model pemotongan rambut yang tersedia. Perancangan *activity diagram posting data model rambut* adalah seperti pada Gambar 3.4.

Nama *Use Case* : *Posting Data Model Rambut*

Aktor : *Barber*

Tujuan : Memasukkan data model rambut ke dalam sistem



Gambar 3.4 Perancangan *activity diagram posting data model rambut*

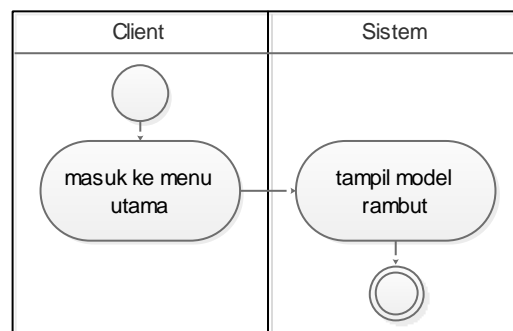
d. *Activity Diagram* Menyajikan Informasi Model Rambut

Activity diagram menyajikan informasi model rambut menggambarkan aktifitas *client* dalam mendapatkan informasi mengenai model rambut yang tersedia. Perancangan *activity diagram* menyajikan informasi model rambut adalah seperti pada Gambar 3.5.

Nama *Use Case* : Menyajikan Informasi Model Rambut

Aktor : *Client*

Tujuan : Melihat informasi data model rambut



Gambar 3.5 Perancangan *activity diagram* menyajikan informasi model rambut

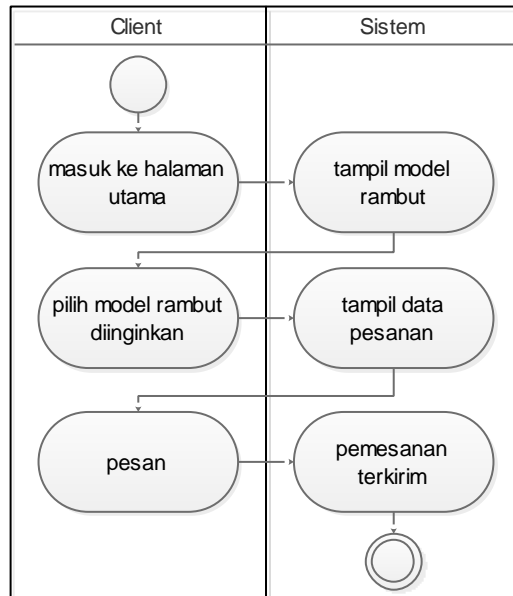
e. *Activity Diagram* Pesan

Activity diagram pesan menggambarkan aktifitas *client* dalam memesan jasa pangkas rambut. Perancangan *activity diagram* pesan adalah seperti pada Gambar 3.6.

Nama *Use Case* : Pesan

Aktor : *Client*

Tujuan : Memesan layanan pangkas rambut



Gambar 3.6 Perancangan *activity diagram* pesan layanan pangkas rambut

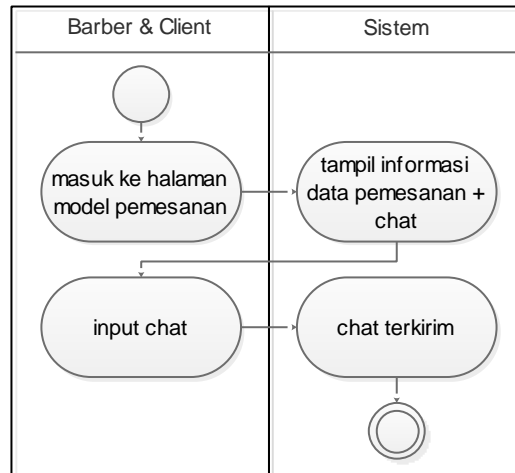
f. *Activity Diagram Chat*

Activity diagram chat menggambarkan aktifitas *client* dan *barber* dalam melakukan chat (berkomunikasi berupa pesan). Perancangan *activity diagram* cha adalah seperti pada Gambar 3.7.

Nama *Use Case* : Chat

Aktor : *Client* dan *Barber* Babershop

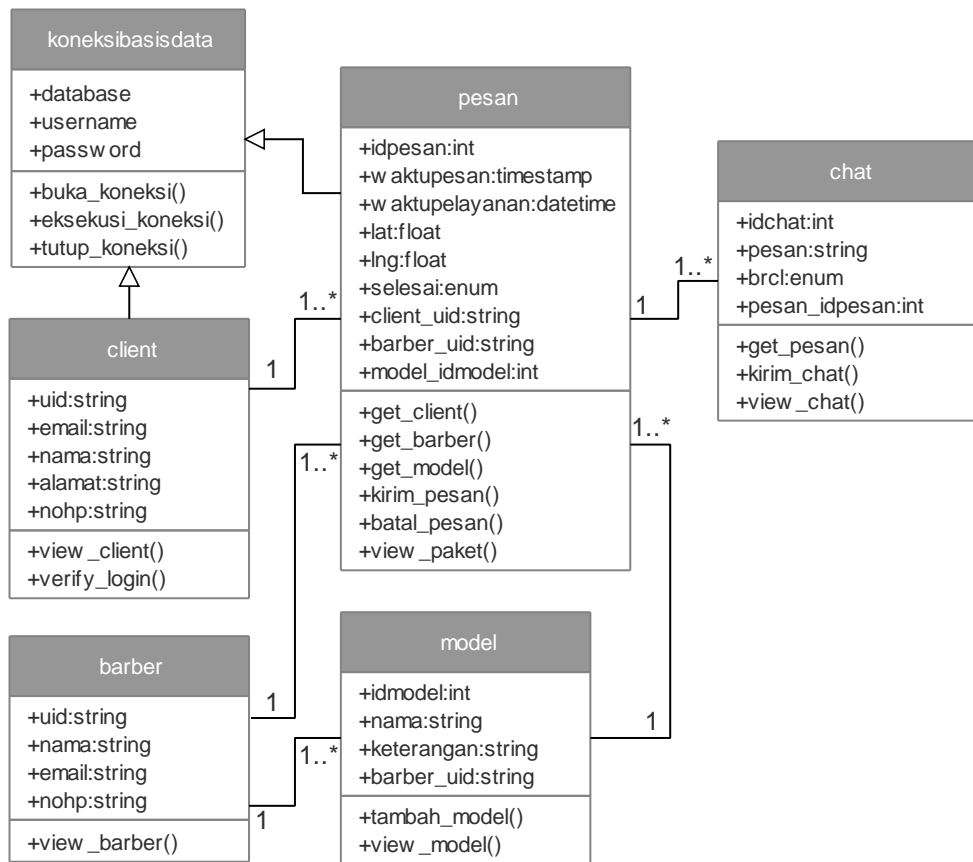
Tujuan : Berkomunikasi melalui pesan



Gambar 3.7 Perancangan *activity diagram chat*

3.5.3 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Perancangan *class diagram booking babershop* berbasis Android menggunakan algoritma queue adalah seperti pada Gambar 3.8.



Gambar 3.8 Perancangan *class diagram* sistem diusulkan

Kamus data dari tabel-tabel yang terdapat pada *database* adalah sebagai berikut :

a. Tabel *Client*

Tabel ini digunakan untuk menyimpan data *client*. Perancangan kamus data tabel *client* adalah seperti pada Tabel 3.1.

Nama tabel : *client*

Primary key : uid

Tabel 3.1 Kamus Data Tabel *Client*

Nama Field	Tipe Data	Size	Deskripsi
uid	varchar	45	<i>user id</i>
email	varchar	100	<i>email client</i>
nama	varchar	45	<i>nama client</i>
alamat	tinytext	-	alamat
nohp	varchar	16	nomor <i>handphone</i>

b. Tabel Pesan

Tabel ini digunakan untuk menyimpan data pesanan. Perancangan kamus data tabel pesan adalah seperti pada Tabel 3.2.

Nama tabel : pesan

Primary key : idpesan

Foreign key : client_uid, barber_uid, model_idmodel

Tabel 32 Kamus Data Tabel Pesan

Nama Field	Tipe Data	Size	Deskripsi
idpesan	int	-	id pesanan
waktupesan	timestamp	-	waktu pemesanan
waktupelayanan	datetime	-	waktu pelayanan
lat	float	12,8	latitude
lng	float	12,8	longitude
selesai	enum	('0','1','2')	status pengerjaan
client_uid	varchar	45	user id client
barber_uid	varchar	45	user id barber
model_idmodel	int	-	id model

c. Tabel Chat

Tabel ini digunakan untuk menyimpan data *chat* atau obrolan antara *client* dengan *barber*. Perancangan kamus data tabel *chat* adalah seperti pada Tabel 3.3.

Nama tabel : chat

Primary key : idchat

Foreign key : pesan_idpesan

Tabel 3.3 Kamus Data Tabel Chat

Nama Field	Tipe Data	Size	Deskripsi
idchat	int	-	id chat
pesan	mediumtext	-	isi pesan
brcl	enum	('br','cl')	barber, client
pesan_idpesan	int	-	id pesan

d. Tabel *Barber*

Tabel ini digunakan untuk menyimpan data *barber*. Perancangan kamus data tabel *barber* adalah seperti pada Tabel 3.4.

Nama tabel : *barber*

Primary key : uid

Tabel 3.4 Kamus Data Tabel *Barber*

Nama Field	Tipe Data	Size	Deskripsi
uid	varchar	45	<i>user id</i>
email	varchar	100	<i>email barber</i>
nama	varchar	45	<i>nama barber</i>
nohp	varchar	16	<i>nomor handphone</i>

e. Tabel Model

Tabel ini digunakan untuk menyimpan data model rambut. Perancangan kamus data tabel model adalah seperti pada Tabel 3.5.

Nama tabel : *client*

Primary key : uid

Foreign key : barber_uid

Tabel 3.5 Kamus Data Tabel Model

Nama Field	Tipe Data	Size	Deskripsi
idmodel	int	-	id model
nama	varchar	45	nama model rambut
keterangan	tinyext	-	keterangan
barber_uid	varchar	45	user id barber

3.5.4 Perancangan *User Interface*

Perancangan *user interface* digunakan untuk menggambarkan tampilan antar muka perangkat lunak yang dibangun. Perancangan ini terdiri dari *user interface barber* dan *user interface client*.

3.5.4.1 User Interface Client

Perancangan *user interface* perangkat lunak antrian *booking babershop* berbasis Android dengan hak akses *client* adalah sebagai berikut :

a. Halaman Registrasi

Halaman registrasi berisikan *form registrasi client* guna menjadi member *babershop*. Perancangan *user interface* halaman registrasi adalah seperti pada Gambar 3.9.

GAMBAR/LOGO	
Form Registrasi	
Email	:
Nama	:
Alamat	:
No. HP	:
Password	:
REGISTRASI	

Gambar 3.9 Perancangan *user interface* halaman registrasi *client*

b. Halaman Login

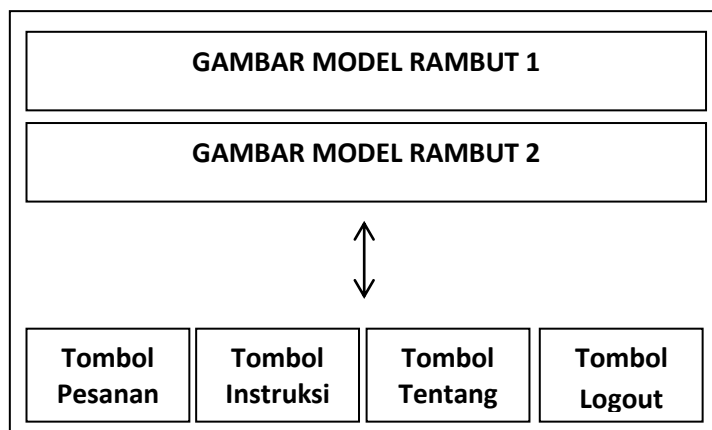
Setelah registrasi, maka *client* dapat masuk ke dalam sistem dengan cara *login*. Perancangan *user interface* halaman *login* adalah seperti pada Gambar 3.10.

GAMBAR/LOGO	
Username	: (email)
Password	:
LOGIN	
<u>REGISTRASI</u>	

Gambar 3.10 Perancangan *user interface* halaman *login*

c. Halaman Utama

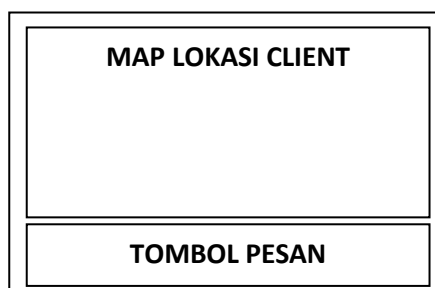
Setelah berhasil login, maka masuk ke halaman utama perangkat lunak. Perancangan *user interface* halaman utama *client* adalah seperti pada Gambar 3.11.



Gambar 3.11 Perancangan *user interface* halaman utama *client*

d. Halaman Pesan Layanan Pangkas

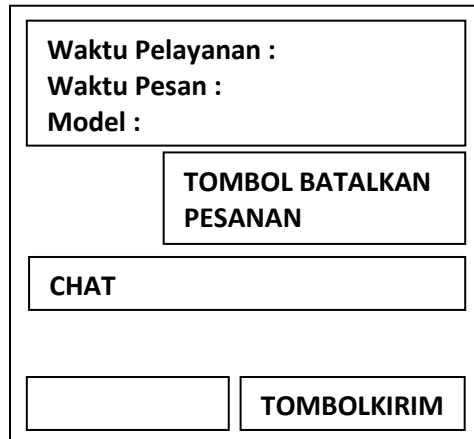
Halaman pesan layanan berisikan map lokasi *client* jenis layanan berisikan berbagai macam jenis pelayanan yang tersedia di *babershop*. Perancangan *user interface* halaman jenis layanan adalah seperti pada Gambar 3.12.



Gambar 3.12 Perancangan *user interface* halaman pesan layanan pangkas rambut

e. Halaman Pesanan

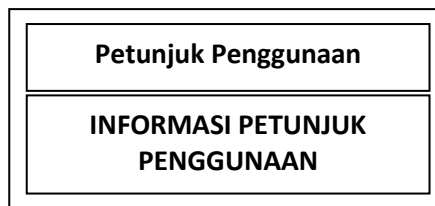
Halaman pesanan berisikan informasi pangkas rambut yang telah dipesan sebelumnya. Perancangan *user interface* halaman pesanan adalah seperti pada Gambar 3.13.



Gambar 3.13 Perancangan *user interface* halaman pesanan

f. Halaman Instruksi

Halaman instruksi berisikan informasi mengenai instruksi penggunaan aplikasi Partner *Babershop*. Perancangan *user interface* halaman instruksi adalah seperti pada Gambar 3.14.



Gambar 3.14 Perancangan *user interface* halaman instruksi

g. Halaman Tentang

Halaman tentang berisikan informasi mengenai Partner *Babershop* (jam operasional, alamat, dan nomor telepon). Perancangan *user interface* halaman tentang adalah seperti pada Gambar 3.15.



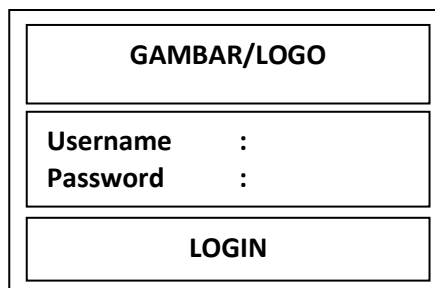
Gambar 3.15 Perancangan *user interface* halaman tentang

3.5.4.2 User Interface Barber

Perancangan *user interface* perangkat lunak antrian *booking babershop* berbasis Android dengan hak akses *barber* adalah sebagai berikut :

a. Halaman *Login*

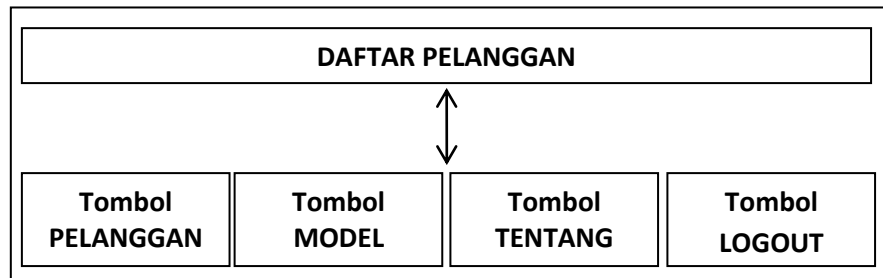
Perancangan *use interface* halaman *login* barber adalah seperti pada Gambar 4.16.



Gambar 3.16 Perancangan *user interface login* barber

b. Halaman Utama *Barber*

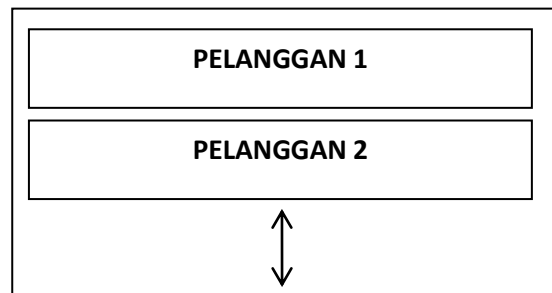
Perancangan *user interface* halaman utama dengan hak akses barber adalah seperti pada Gambar 3.17.



Gambar 3.17 Perancangan *user interface* halaman utama barber

c. Halaman Daftar Pelanggan

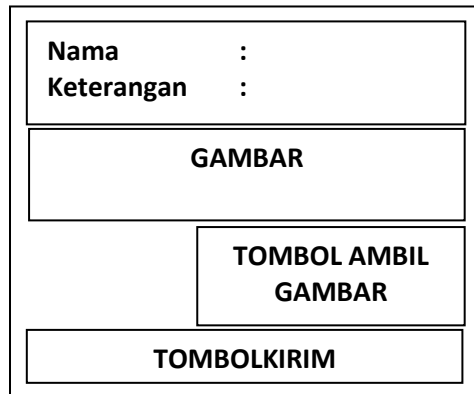
Perancangan *user interface* halaman daftar pelanggan yang melakukan pemesanan pangkas rambut adalah seperti pada Gambar 3.18.



Gambar 3.18 Perancangan *user interface* halaman daftar pelanggan barber

d. Halaman Model Rambut

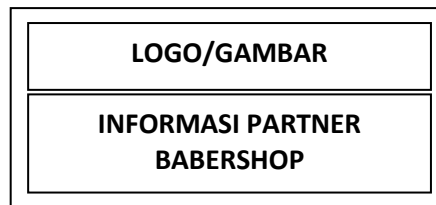
Perancangan *user interface* halaman *input* model pangkas rambut adalah seperti pada Gambar 3.19.



Gambar 3.19 Perancangan *user interface* halaman *input* model rambut

e. Halaman Tentang

Halaman tentang berisikan informasi mengenai Partner *Babershop* (jam operasional, alamat, dan nomor telepon). Perancangan *user interface* halaman tentang adalah seperti pada Gambar 3.20.



Gambar 3.20 Perancangan *user interface* halaman tentang barber

BAB IV

HASIL DAN PEMBAHASAN

Hasil beserta pembahasan mengenai implementasi dari perangkat lunak antrian *booking babershop* berbasis Android dijelaskan pada sub-sub pokok bahasan di bawah ini.

4.1 Hasil

Hasil dari penelitian ini berupa aplikasi antrian *booking babershop* berbasis Android dengan metode algoritma queue. Aplikasi ini berjalan pada *handphone* yang telah didukung oleh sistem operasi Android yang dapat memudahkan *client* yang memiliki *smartphone* Android dalam melakukan pemesanan pangkas rambut. Di dalam aplikasi tersebut tersedia model-model rambut yang di dalamnya terdapat pemesanan pangkas rambut oleh *client* beserta lokasi *client* sehingga memudahkan *client* dalam melakukan pangkas rambut tanpa datang ke lokasi *barbershop*. Adapun pembahasan mengenai implementasi dari perancangan aplikasi yang telah dirancang sebelumnya adalah dijelaskan pada sub pokok pembahasan.

4.2 Pembahasan

Di dalam pokok pembahasan terdiri dari dua sub bagian, yaitu pembahasan mengenai implementasi sistem dan pembahasan mengenai pengujian sistem.

4.2.1 Implementasi Sistem

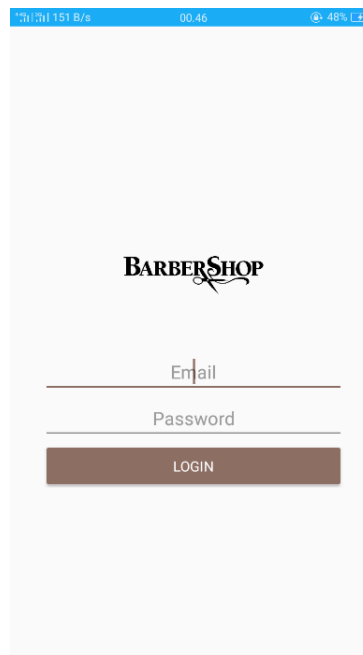
Pembahasan dari implementasi aplikasi antrian *booking babershop* berbasis Android dengan metode algoritma queue adalah terdiri dari implementasi *barber* dan implementasi *client*.

4.2.1.1 Implementasi Barber

Implementasi dari perancangan *user interface* dengan hak akses barber adalah sebagai berikut :

a. Implementasi Halaman *Login* Barber

Barber harus *login* terlebih dahulu untuk dapat masuk ke dalam sistem dengan memasukkan *username* dan *password*. Implementasi halaman *login* barber adalah seperti pada Gambar 4.1.



Gambar 4.1 Implementasi halaman *login* barber

b. Implementasi Halaman Utama Barber

Setelah barber berhasil *login*, maka akan muncul menu utama yang terdiri dari daftar pelanggan dan terdapat 4 tombol operasi yaitu : pesanan, model, tentang aplikasi, serta tombol logout seperti pada Gambar 4.2.



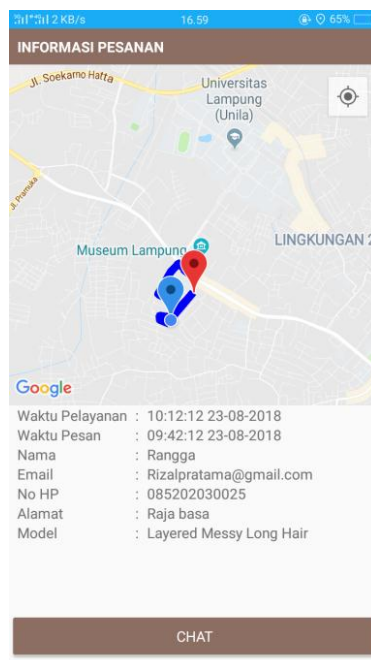
Gambar 4.2 Implementasi halaman utama barber

c. Implementasi Halaman Pesanan Barber

Ketika barber menekan tombol pesanan, maka akan muncul daftar pesanan pangkas rambut oleh *client* seperti pada Gambar 4.3. Ketika barber menekan salah satu daftar pelanggan, maka akan muncul informasi detail yang terdiri dari lokasi pelanggan beserta jati diri pelanggan dan model pangkas rambut yang dipilih oleh pelanggan seperti pada Gambar 4.4. barber juga dapat melakukan komunikasi dengan client dengan cara chat seperti pada gambar 4.5.



Gambar 4.3 Implementasi halaman pesan barber



Gambar 4.4 Implementasi halaman informasi detail *client*



Gambar 4.5 Implementasi halaman *chat* barber dengan *client*

d. Implementasi Halaman Model

Di dalam halaman ini, barber dapat melakukan input data model pangkas rambut seperti pada Gambar 4.6. Ketika barber menekan tombol tambah, maka akan muncul halaman input data model pangkas rambut seperti pada Gambar 4.7.



Gambar 4.6 Implementasi halaman model



Gambar 4.7 Implementasi halaman *input* data model

e. Implementasi Halaman Tentang

Halaman tentang berisikan informasi mengenai tentang aplikasi berbershop. Implementasi halaman tentang adalah seperti pada Gambar 4.8.



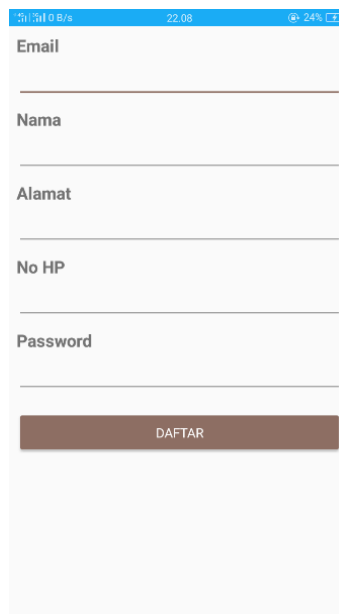
Gambar 4.8 Implementasi halaman tentang yang tersedia di dalam hak akses barber

4.2.1.2 Implementasi *Client*

Implementasi dari perancangan *user interface* dengan hak akses *client* adalah sebagai berikut :

a. Implementasi Halaman Registrasi *Client*

Client harus melakukan pendaftaran atau registrasi terlebih dahulu untuk dapat login dan mengakses sistem. Implementasi halaman

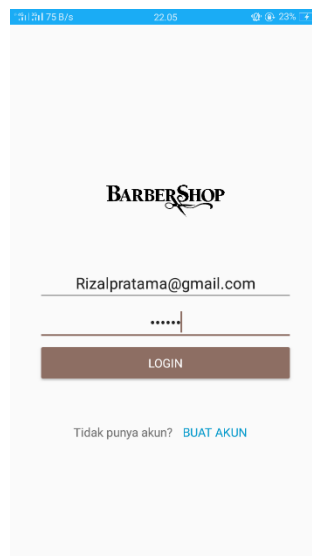
The image shows a mobile application interface for client registration. It features a white background with a blue header bar at the top. The header bar contains the text 'DAFTAR PELANGGAN' on the left, the time '22:08' in the center, and the battery level '24%' on the right. Below the header, there are five input fields with labels: 'Email', 'Nama', 'Alamat', 'No HP', and 'Password'. Each field has a horizontal line indicating the input area. At the bottom of the form, there is a brown button with the text 'DAFTAR' in white.

registrasi client yang telah dirancang sebelumnya adalah seperti pada gambar 4.9.

Gambar 4.9 Implementasi halaman *registrasi client*

b. Implementasi Halaman *Login Client*

Setelah berhasil registrasi, maka client login dengan memasukkan



username berupa email dan *password* seperti pada gambar 4.10.

Gambar 4.10 Implementasi halaman *login client*

c. Implementasi Halaman Utama *Client*

Setelah berhasil *login*, maka client akan masuk ke halaman utama yang berisikan daftar model pangkas rambut beserta tombol operasi pemesanan, instruksi dan tentang aplikasi seperti pada Gambar 4.11.



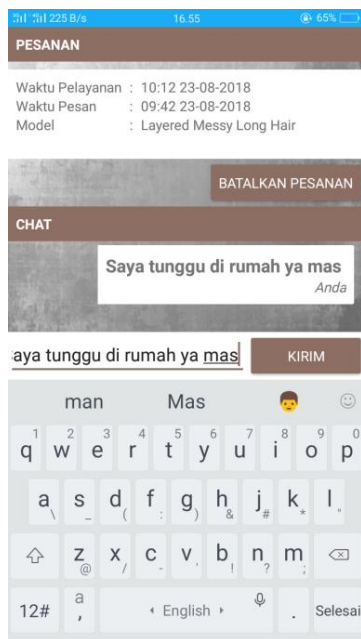
Gambar 4.11 Implementasi halaman utama *client*

d. Implementasi Halaman Pemesanan Pangkas Rambut

Ketika *client* menekan salah satu daftar model yang tertera, maka akan muncul halaman lokasi *client* beserta tombol operasi pesan seperti pada Gambar 4.12. Ketika *client* menekan tombol operasi pesan, maka *client* telah melakukan pemesanan pangkas rambut dengan model tersebut seperti pada Gambar 4.13. *Client* dapat melakukan pembatalan pemesanan dan melakukan chat dengan barber di dalam halaman ini.



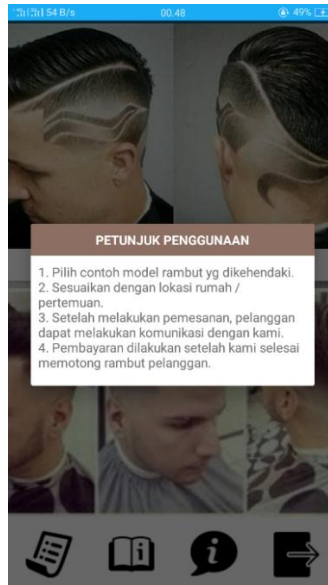
Gambar 4.12 Implementasi halaman pemesanan pangkas rambut



Gambar 4.13 Implementasi halaman informasi detail pemesanan pangkas rambut

e. Implementasi Halaman Instruksi

Halaman instruksi berisikan informasi mengenai instruksi penggunaan aplikasi berbershop. Implementasi halaman instruksi adalah seperti pada Gambar 4.14.



Gambar 4.14 Implementasi halaman instruksi penggunaan aplikasi oleh *client*

f. Implementasi Halaman Tentang Aplikasi

Halaman tentang berisikan informasi mengenai tentang aplikasi berbershop. Implementasi halaman tentang adalah seperti pada Gambar 4.15.






Gambar 4.15 Implementasi halaman tentang aplikasi oleh *client*



4.2.2 Pengujian Sistem


Sistem diuji dengan menggunakan metode *black box testing*. Sistem yang diuji terdiri dari pengujian versi Android beserta *interface* menu utama aplikasi (*client*). Pengujian ini bertujuan untuk mengetahui apakah aplikasi ini dapat berjalan pada beberapa versi Android yang diujikan. Pengujian ini menggunakan beberapa kriteria yang sudah ditetapkan yaitu kelas uji, , skenario uji, hasil gambar *device* dan hasil yang diperoleh. Pengujian terhadap versi Android beserta *interface* menu utama aplikasi (*client*) yang digunakan adalah seperti pada Tabel 4.1

Tabel 4.1 Hasil Pengujian Versi Android Beserta *Interface*

Kelas Uji	Skenario Uji	Hasil Gambar <i>Device</i>	Hasil yang Diperoleh
Pengujian kompatibilitas versi Sistem Operasi Android	Pengujian pada <i>Smartphone</i> Merk Xiaomi Redmi 4C versi Android 5.0 (<i>Lollipop</i>) dengan resolusi layar 5 <i>inch</i> .		Berhasil Kompatibel dengan Android versi 5.0 (<i>Lollipop</i>)
	Pengujian pada <i>Smartphone</i> Merk Oppo A71(2018) Android versi 7.1.1 (<i>Nougat</i>) dengan resolusi layar 5.2 <i>inch</i> .		Berhasil Kompatibel dengan Android versi 7.1.1 (<i>Nougat</i>)
Pengujian pada menu utama aplikasi antrian <i>booking babershop</i>	Pilih <i>icon</i> pesan		Berhasil menampilkan deskripsi data pesanan pangkas rambut yang telah dipesan

Tabel 4.1 Hasil Pengujian Versi Android Beserta *Interface* (Lanjutan)

	<p>Menekan salah satu model pangkas rambut</p>		<p>Berhasil menampilkan map lokasi <i>client</i> yang di dalamnya terdapat tombol pesan</p>
	<p>Pilih <i>icon</i> instruksi</p>		<p>Berhasil menampilkan instruksi penggunaan aplikasi</p>

	Pilih icon tentang aplikasi		Berhasil menampilkan informasi tentang aplikasi
--	-----------------------------	--	---

Dengan hasil pengujian kompatibilitas sistem operasi Android pada Tabel 4.1, maka dapat disimpulkan bahwa aplikasi antrian *booking babershop* berbasis Android dengan metode algoritma queue dapat berjalan dengan baik atau kompatibel dengan versi Android 5.0 (Lollipop) dan versi Android 7.1.1 (Nougat). Dari hasil pengujian *user interface* menu utama (*client*) yang tertera pada pada Tabel 4.1, maka dapat disimpulkan bahwa aplikasi antrian *booking babershop* berbasis Android dengan metode algoritma queue ini berhasil menampilkan tampilan yang baik dan sesuai dengan yang diharapkan.

4.3 Kelebihan dan Kekurangan

4.3.1 Kelebihan

Kelebihan dari aplikasi antrian *booking babershop* berbasis Android dengan metode algoritma *queue* adalah sebagai berikut :

- a. Pelayanan *barbershop* berbasis Android yang dapat memudahkan *client* dalam melakukan *booking* pelayanan potong rambut ditempat berdasarkan model yang diinginkan.
- b. Menambah jasa pelayanan dan media informasi *berbershop*.
- c. Mempermudah barber dalam menemukan lokasi *client* yang meminta jasa pelayanan ini.

- d. Aplikasi ini dilengkapi dengan pengolahan data antrian *client* menggunakan algoritma *queue* yang memudahkan barber dalam menentukan urutan *client* selanjutnya.

4.3.2 Kekurangan

Kekurangan dari aplikasi antrian *booking babershop* berbasis Android dengan metode algoritma *queue* adalah sebagai berikut :

- a. Fasilitas pelayanan yang tertera pada aplikasi tidak selengkap dengan yang ada di tempatnya. Pelayanan yang ada di aplikasi hanya berupa potong rambut, tidak ada layanan lainnya seperti mewarnai rambut dan sebagainya.
- b. Tidak ada perhitungan biaya jarak menuju lokasi pelanggan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan proses pembangunan aplikasi antrian *booking babershop* berbasis Android menggunakan algoritma queue yang dimulai dari tahap analisis hingga implementasi, maka dapat ditarik kesimpulan sebagai berikut :

- a. Aplikasi yang dibangun menampilkan model potong rambut beserta pemesanan. Jika *client* melakukan pemesanan, maka *client* harus menentukan titik lokasi client sehingga memudahkan barber dalam menemukan lokasi *client*.
- b. Di dalam aplikasi terdapat kolom *chat* antara barber dengan *client* sehingga memudahkan komunikasi antara barber dengan *client*.
- c. Cara kerja metode queue yang dibangun adalah jika ada antrian baru, maka diperlukan rentang waktu 90 menit dari pelanggan terakhir. Jika terdapat *client* yang selesai kurang dari waktu 90 menit, maka antrian akan dimajukan. Jika ada *client* yang membatalkan pesanan, maka *client* berikutnya akan dimajukan jadwalnya.
- d. Aplikasi hanya dapat di *install* pada sistem operasi Android dengan spesifikasi minimum Lollipop atau versi 5.0.

5.2 Saran

Agar aplikasi antrian *booking babershop* berbasis Android menggunakan algoritma queue lebih sempurna, maka pada penelitian berikutnya disarankan:

- a. Menambahkan fitur layanan barbershop, tidak hanya potong rambut saja tetapi layanan lainnya seperti mewarnai dan sebagainya.
- b. Menambahkan fitur peringkat kesukaan dari model gaya rambut yang tersedia.

DAFTAR PUSTAKA

- Ardhy, Ferly. 2018. Sistem Informasi Geografis Penyedia Jasa Rumah Kos Berbasis *Website* (Studi Kasus: Wilayah Kotabumi Lampung Utara). *Jurnal Informatika*, Vol. 1, No. 1, Maret 2018.
- Arfida, Septilia., dkk. 2017. Informasi Profil Tenaga Pendidik Bersertifikasi Pada Sekolah Dasar Negeri Provinsi Lampung Berbasis Android. Konferensi Nasional Sistem & Informatika, Agustus 2017.
- Artha K, Iga Yudi., dkk. (Tanpa Tahun). Perancangan Media Komunikasi Visual Gangs Babershop di Bali. Universitas Kristen Petra, Surabaya.
- A.S, Rosa dan Shalahuddin, M. 2016. Modul Pembelajaran Rekayasa Perangkat Lunak (Terstruktur dan Berorientasi Objek). Modula, Bandung.
- Cipta, Hendra., & Rangga, Gelar. (Tanpa Tahun). Pembangunan Perangkat Lunak Online Booking Barbershop Di Bandung Menggunakan Teknologi Mobile Global Positioning System Dan Web Services Pada Platform Android. Universitas Komputer, Bandung.
- Cundomanik, Eric Prabowo., dkk. 2015. Aplikasi Survei Berbasis Android. Universitas Kristen Petra, Surabaya.
- Kadir, Abdul. 2013. Teori dan Aplikasi Struktur Data Menggunakan C++. Andi, Yogyakarta.
- Sari, Yuni Puspita. 2016. Membangun Kamus *OBSTETRI* Berbasis Android (Studi Kasus : Kampus Kebidanan Adila Bandar Lampung). *Jurnal Informatika*, Vol. 16, No. 1, Juni 2016.

Sidi, M Mustaqbal., dkk. 2015. Pengujian Aplikasi Menggunakan *Black Box Testing Boundary Value Analysis* (Studi Kasus : Aplikasi Prediksi Kelulusan SNMPTN). Jurnal Ilmiah Teknologi Informasi Terapan, Volume I, No 3, ISSN : 2407 – 3911.

Solichin, Achmad. 2016. Pemrograman Web dengan PHP dan MySQL. *E-Book* diunduh dari https://www.researchgate.net/publication/236885805_Pemrograman_Web_dengan_PHP_dan_MySQL.

Zia'ulhaq, Muhammad. (2014). Penerapan Sistem Informasi Berbasis Web untuk Mendukung Pengelolaan Administrasi di Promusic Recording Studio Jepara. Universitas Dian Nuswantoro, Semarang.

LAMPIRAN

PartnerBarbershopAndBarber (Admin)

ClientAdp :

```
package com.flaniest.partnerbarbershopandbarber.adp;

import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.LinearLayout;
import android.widget.TextView;

import com.flaniest.partnerbarbershopandbarber.R;
import com.flaniest.partnerbarbershopandbarber.mdl.ClientMdl;

import java.util.List;

public class ClientAdp extends
RecyclerView.Adapter<ClientAdp.ViewHolder> {
    private List<ClientMdl> mdls;

    public ClientAdp(List<ClientMdl> mdls) {
        this.mdls = mdls;
    }

    public class ViewHolder extends RecyclerView.ViewHolder
    {
        private TextView tvNama, tvEmail, tvNoHP, tvAlamat;

        public ViewHolder(View itemView) {
            super(itemView);

            tvNama = itemView.findViewById(R.id.tvNama);
            tvEmail = itemView.findViewById(R.id.tvEmail);
            tvNoHP = itemView.findViewById(R.id.tvNoHP);
            tvAlamat = itemView.findViewById(R.id.tvAlamat);
        }
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.cl
ient_item, parent, false);
        return new ViewHolder(view);
    }
}
```

```

        @Override
        public void onBindViewHolder(@NonNull ViewHolder holder,
int position) {
            ClientMdl mdl = mdls.get(position);

            holder.tvNama.setText(mdl.nama);
            holder.tvEmail.setText(mdl.email);
            holder.tvNoHP.setText(mdl.nohp);
            holder.tvAlamat.setText(mdl.alamat);
        }

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

```

PesanAdp :

```

package com.flaniest.partnerbarbershopandbarber.adp;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.graphics.Color;
import android.support.annotation.NonNull;
import android.support.v7.widget.CardView;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.enwif.dtc.RqsDtc;
import com.flaniest.partnerbarbershopandbarber.R;
import com.flaniest.partnerbarbershopandbarber.cnn.Cnn;
import com.flaniest.partnerbarbershopandbarber.cnt.PesanDetail;
import com.flaniest.partnerbarbershopandbarber.mdl.PesanMdl;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GetTokenResult;

import java.text.DateFormat;
import java.text.ParseException;

```

```

import java.text.SimpleDateFormat;
import java.time.Instant;
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;

public class PesanAdp extends
RecyclerView.Adapter<PesanAdp.ViewHolder> {
    private List<PesanMdl> mdls;
    private Context context;

    private FirebaseAuth mAuth;

    public PesanAdp(List<PesanMdl> mdls) {
        this.mdls = mdls;
        mAuth = FirebaseAuth.getInstance();
    }

    public class ViewHolder extends RecyclerView.ViewHolder
    {
        private CardView cv;
        private TextView tvWaktuPelayanan, tvNama, tvAlamat,
tvWaktuPesan, tvSelesai;

        public ViewHolder(View itemView) {
            super(itemView);

            cv = itemView.findViewById(R.id.cv);
            tvWaktuPelayanan =
itemView.findViewById(R.id.tvWaktuPelayanan);
            tvNama = itemView.findViewById(R.id.tvNama);
            tvAlamat = itemView.findViewById(R.id.tvAlamat);
            tvWaktuPesan =
itemView.findViewById(R.id.tvWaktuPesan);
            tvSelesai =
itemView.findViewById(R.id.tvSelesai);

            cv.setOnClickListener(new View.OnClickListener()
            {
                @Override
                public void onClick(View view) {
                    if
(mdls.get(getAdapterPosition()).selesai.equals("0")) {
                        Intent intent = new Intent(context,
PesanDetail.class);
                        intent.putExtra("mdl",
mdls.get(getAdapterPosition()));
                    }
                }
            });
        }
    }
}

```

```

                context.startActivity(intent);
            }
        }
    });

    cv.setOnLongClickListener(new
View.OnLongClickListener() {
        @Override
        public boolean onLongClick(View view) {
            if
(mdls.get(getAdapterPosition()).selesai.equals("0"))
chs(getAdapterPosition());
            return false;
        }
    });
}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
    context = parent.getContext();
    View view =
LayoutInflater.from(parent.getContext()).inflate(R.layout.pe
san_item, parent, false);
    return new ViewHolder(view);
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder,
int position) {
    PesanMdl mdl = mdls.get(position);

    try {
        if (android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O) {
            Instant instant;
            LocalDateTime localDateTime;

            instant = Instant.parse(mdl.waktupelayanan);
            localDateTime =
LocalDateTime.from(instant.atZone(ZoneId.systemDefault()));

            holder.tvWaktuPelayanan.setText(DateTimeFormatter.ofPattern(
"HH:mm dd-MM-yyyy").format(localDateTime));

            instant = Instant.parse(mdl.waktupesana);
            localDateTime =
LocalDateTime.from(instant.atZone(ZoneId.systemDefault()));

            holder.tvWaktuPesana.setText(DateTimeFormatter.ofPattern("HH:
mm dd-MM-yyyy").format(localDateTime));

```

```

        } else {
            DateFormat dateFormat = new
SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'", Locale.US);

dateFormat.setTimeZone(TimeZone.getTimeZone("UTC"));
            DateFormat dateFormatLocal = new
SimpleDateFormat("HH:mm dd-MM-yyyy", Locale.US);

dateFormatLocal.setTimeZone(TimeZone.getDefault());
            String a = TimeZone.getDefault().toString();

            Date date =
dateFormat.parse mdl.waktupelayanan);

System.out.println(dateFormatLocal.format(date));
            String b = dateFormatLocal.format(date);

holder.tvWaktuPelayanan.setText(dateFormatLocal.format(date)
);

holder.tvWaktuPesan.setText(dateFormatLocal.format(dateForma
t.parse mdl.waktupesana));
        }
    } catch (ParseException e) {
        e.printStackTrace();
    }
}
holder.tvNama.setText mdl.clientnama);
holder.tvAlamat.setText mdl.clientalamat);
if (mdl.selesai.equals("0")) {
    holder.tvSelesai.setText(R.string.strSelesai0);
    holder.cv.setCardBackgroundColor(Color.WHITE);
} else if (mdl.selesai.equals("1")) {
    holder.tvSelesai.setText(R.string.strSelesai1);
    holder.cv.setCardBackgroundColor(Color.CYAN);
} else if (mdl.selesai.equals("2")) {
    holder.tvSelesai.setText(R.string.strSelesai2);
    holder.cv.setCardBackgroundColor(Color.GRAY);
}
}

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

private void chs(final int position) {
    AlertDialog.Builder builder = new
AlertDialog.Builder(context);

    builder.setTitle(R.string.strStatusSelesai)
        .setItems(new
CharSequence[]{context.getString(R.string.strSelesai1),

```

```

context.getString(R.string.strSelesai2)},
        new
DialogInterface.OnClickListener() {
            @Override
            public void
onClick(DialogInterface dialogInterface, int i) {
                switch (i) {
                    case 0:
                        cekSend("1",
position);

                        break;
                    case 1:
                        cekSend("2",
position);

                        break;
                }
            }
        });

        builder.create().show();
    }

    private void cekSend(final String value, final int
position) {
        FirebaseUser firebaseUser = mAuth.getCurrentUser();
        if (firebaseUser != null) {
            final String uid = firebaseUser.getUid();
            firebaseUser.getIdToken(true)
                .addOnCompleteListener(new
OnCompleteListener<GetTokenResult>() {
                    @Override
                    public void onComplete(@NonNull
Task<GetTokenResult> task) {
                        if (task.isSuccessful()) {
                            send(value, position,
task.getResult().getToken());
                        }
                    }
                });
        }
    }

    private void send(String value, final int position,
String token) {
        Map<String, String> params = new HashMap<>();
        params.put("idpesan", mdl.get(position).idpesan);
        params.put("selesai", value);
        params.put("token", token);

        RqsDtc.StrRqsPOSTDtc(context, Cnn.url + "pesan/upd",
params, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {

```

```

        if (!response.equals("null") &&
!response.equals("0")) {
            notifyItemChanged(position);
            ntf();
        } else Toast.makeText(context,
R.string.strFail, Toast.LENGTH_LONG).show();
    }
}, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Log.e("Send", error.toString());
    }
});
}

private void ntf() {
    Map<String, String> params = new HashMap<>();
    params.put("t1", "PARTNER BARBERSHOP");
    params.put("bd", "Terdapat perubahan pada antrian
waktu pelayanan, buka aplikasi untuk melihatnya");

    RqsDtc.StrRqsPOSTDtc(context, Cnn.url +
"ntfy/notif", params, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            //
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("Send", error.toString());
        }
    });
}
}
}

```

Home.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/bg"
tools:context=".cnt.Home">

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:background="@color/colorPrimary"

```

```

        android:orientation="horizontal"
        android:padding="8dp"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="@string/strDaftarPelanggan"
            android:textAllCaps="true"
            android:textColor="@android:color/white"
            android:textStyle="bold" />

        <TextView
            android:id="@+id/tvTgl"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:textAllCaps="true"
            android:textColor="@android:color/white"
            android:textStyle="bold" />
    </LinearLayout>

    <view
        android:id="@+id/srl"
        class="android.support.v4.widget.SwipeRefreshLayout"
        id="@+id/view"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginEnd="8dp"
        android:layout_marginStart="8dp"

        app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"

        app:layout_constraintTop_toBottomOf="@+id/linearLayout">

        <android.support.v7.widget.RecyclerView
            android:id="@+id/rv"
            android:layout_width="match_parent"
            android:layout_height="match_parent" />

    </view>

    <LinearLayout
        android:id="@+id/linearLayout2"
        android:layout_width="0dp"
        android:layout_height="75dp"
        android:background="@android:color/white"
        android:orientation="horizontal"
        android:padding="8dp"

```



```

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent">

<ImageButton
    android:id="@+id/btnClient"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="@null"
    android:contentDescription="@string/strBlank"
    app:srcCompat="@drawable/client" />

<ImageButton
    android:id="@+id/btnModel"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="@null"
    android:contentDescription="@string/strBlank"
    app:srcCompat="@drawable/model" />

<ImageButton
    android:id="@+id/btnAbout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="@null"
    app:srcCompat="@drawable/about" />

<ImageButton
    android:id="@+id/btnLogout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="1"
    android:background="@null"
    app:srcCompat="@drawable/logout" />
</LinearLayout>

</android.support.constraint.ConstraintLayout>

```

Login.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".cnt.Login">

```

```

<LinearLayout
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginEnd="32dp"
    android:layout_marginStart="32dp"
    android:layout_marginTop="8dp"
    android:gravity="center"
    android:orientation="vertical"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

    <ImageView
        android:layout_width="150dp"
        android:layout_height="150dp"
        android:contentDescription="@string/strBlank"
        android:scaleType="centerInside"
        app:srcCompat="@drawable/logo" />

    <EditText
        android:id="@+id/etEmail"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/strEmail"
        android:inputType="textEmailAddress"
        android:textAlignment="center" />

    <EditText
        android:id="@+id/etPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="@string/strPassword"
        android:inputType="textPassword"
        android:textAlignment="center" />

    <Button
        android:id="@+id/btn"
        style="@style/Widget.AppCompat.Button.Colored"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/strLogin" />
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

PartnerBarbershopAndClient (Customer)

PesanAdp :

```
package com.enwif.partnerbarbershopandclient.cnt;

import android.content.DialogInterface;
import android.content.Intent;
import android.support.annotation.NonNull;
import android.support.v7.app.AlertDialog;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.support.v7.widget.LinearLayoutManager;
import android.support.v7.widget.RecyclerView;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import com.android.volley.Response;
import com.android.volley.VolleyError;
import com.enwif.dtc.RqsDtc;
import com.enwif.partnerbarbershopandclient.R;
import com.enwif.partnerbarbershopandclient.adp.ChatAdp;
import com.enwif.partnerbarbershopandclient.cnn.Cnn;
import com.enwif.partnerbarbershopandclient.mdl.ChatMdl;
import com.enwif.partnerbarbershopandclient.mdl.ClientMdl;
import com.enwif.partnerbarbershopandclient.mdl.PesanMdl;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;
import com.google.firebase.auth.GetTokenResult;

import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;

import java.text.DateFormat;
import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.time.Instant;
import java.time.LocalDateTime;
import java.time.ZoneId;
import java.time.format.DateTimeFormatter;
import java.util.ArrayList;
import java.util.Date;
import java.util.HashMap;
import java.util.List;
import java.util.Locale;
import java.util.Map;
import java.util.TimeZone;
```

```

public class Pesanan extends AppCompatActivity {
    private Button btnCnl;

    private RecyclerView rv;
    private RecyclerView.LayoutManager rvLm;

    private List<ChatMdl> mdls;
    private ChatAdp adp;

    private EditText etChat;
    private Button btnSend;

    private TextView tvWaktuPelayanan, tvWaktuPesan,
tvModel;

    private FirebaseAuth mAuth;
    private ClientMdl clientMdl;

    private PesanMdl mdl;

    Thread thread;
    boolean runThread = true;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.pesanan);

        btnCnl = findViewById(R.id.btnCnl);

        rv = findViewById(R.id.rv);
        rvLm = new LinearLayoutManager(this);

        btnCnl.setOnClickListener(new View.OnClickListener()
{
            @Override
            public void onClick(View view) {
                AlertDialog.Builder builder = new
AlertDialog.Builder(Pesanan.this);
                builder.setTitle(R.string.strDelete)
                    .setPositiveButton(R.string.strOK,
new DialogInterface.OnClickListener() {
                        @Override
                        public void
onClick(DialogInterface dialogInterface, int i) {
                            cekUserDel();
                        }
                    })
                .setNegativeButton(R.string.strCancel, new
DialogInterface.OnClickListener() {

```

```

        @Override
        public void
onClick(DialogInterface dialogInterface, int i) {
        //
        }
    });
    builder.show();
}
});

    etChat = findViewById(R.id.etChat);
    btnSend = findViewById(R.id.btnSend);

    btnSend.setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View view) {
            cekUserChat();
        }
    });

    tvWaktuPelayanan =
findViewById(R.id.tvWaktuPelayanan);
    tvWaktuPesan = findViewById(R.id.tvWaktuPesan);
    tvModel = findViewById(R.id.tvModel);

    mAuth = FirebaseAuth.getInstance();

    cekUserDt();

    thread = new Thread() {
        @Override
        public void run() {
            try {
                while (runThread) {
                    sleep(10000);
                    runOnUiThread(new Runnable() {
                        @Override
                        public void run() {
                            setDtChat();
                        }
                    });
                }
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
    };
}
}

```

```

private void cekUserDt() {
    FirebaseUser firebaseUser = mAuth.getCurrentUser();
    if (firebaseUser != null) {
        clientMdl = new ClientMdl();
        clientMdl.uid = firebaseUser.getUid();
        clientMdl.email = firebaseUser.getEmail();
        clientMdl.nama = firebaseUser.getDisplayName();

        setDt();
    } else {
        Intent intent = new Intent(this, Login.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
}

private void setDt() {
    Map<String, String> params = new HashMap<>();
    params.put("client_uid", clientMdl.uid);
    params.put("selesai", "0");

    RqsDtc.StrRqsPOSTDtc(this, Cnn.url + "pesan/last",
params, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if (!response.equals("null") &&
!response.equals("0")) {
                try {
                    JSONObject jsonObject = new
JSONObject(response);

                    mdl = new PesanMdl();
                    mdl.idpesan =
jsonObject.getString("idpesan");
                    mdl.waktupesan =
jsonObject.getString("waktupesan");
                    mdl.waktupelayanan =
jsonObject.getString("waktupelayanan");
                    mdl.lat =
jsonObject.getString("lat");
                    mdl.lng =
jsonObject.getString("lng");
                    mdl.selesai =
jsonObject.getString("selesai");
                    mdl.client_uid =
jsonObject.getString("client_uid");
                    mdl.barber_uid =
jsonObject.getString("barber_uid");
                    mdl.model_idmodel =
jsonObject.getString("model_idmodel");

```

```

        /*view*/
        mdl.clientuid =
jsonObject.getString("clientuid");
        mdl.clientemail =
jsonObject.getString("clientemail");
        mdl.clientnama =
jsonObject.getString("clientnama");
        mdl.clientalamat =
jsonObject.getString("clientalamat");
        mdl.clientnohp =
jsonObject.getString("clientnohp");
        mdl.barberuid =
jsonObject.getString("barberuid");
        mdl.barberemail =
jsonObject.getString("barberemail");
        mdl.barbernama =
jsonObject.getString("barbernama");
        mdl.barbernohp =
jsonObject.getString("barbernohp");
        mdl.modelidmodel =
jsonObject.getString("modelidmodel");
        mdl.modelnama =
jsonObject.getString("modelnama");
        mdl.modelketerangan =
jsonObject.getString("modelketerangan");
    } catch (JSONException e) {
        e.printStackTrace();
    }

    if (mdl != null) {
        try {
            if
(android.os.Build.VERSION.SDK_INT >=
android.os.Build.VERSION_CODES.O) {
                Instant instant;
                LocalDateTime localDateTime;

                instant =
Instant.parse(mdl.waktupelayanan);
                localDateTime =
LocalDateTime.from(instant.atZone(ZoneId.systemDefault()));

                tvWaktuPelayanan.setText(DateTimeFormatter.ofPattern("HH:mm
dd-MM-yyyy").format(localDateTime));

                instant =
Instant.parse(mdl.waktupesana);
                localDateTime =
LocalDateTime.from(instant.atZone(ZoneId.systemDefault()));

                tvWaktuPesana.setText(DateTimeFormatter.ofPattern("HH:mm dd-
MM-yyyy").format(localDateTime));
            } else {

```

```

        DateFormat dateFormat = new
SimpleDateFormat("yyyy-MM-dd'T'HH:mm:ss.SSS'Z'", Locale.US);

dateFormat.setTimeZone(TimeZone.getTimeZone("UTC"));
        DateFormat dateFormatLocal =
new SimpleDateFormat("HH:mm dd-MM-yyyy", Locale.US);

dateFormatLocal.setTimeZone(TimeZone.getDefault());

        Date date =
dateFormat.parse mdl.waktupelayanan);

System.out.println(dateFormatLocal.format(date));

tvWaktuPelayanan.setText(dateFormatLocal.format(dateFormat.p
arse mdl.waktupelayanan));

tvWaktuPesanan.setText(dateFormatLocal.format(dateFormat.parse
(mdl.waktupesanan));
    }
    } catch (ParseException e) {
        e.printStackTrace();
    }
    tvModel.setText mdl.modelnama);

    setDtChat();
    thread.start();
} else {
    AlertDialog.Builder builder = new
AlertDialog.Builder(Pesanan.this);

builder.setTitle(R.string.strInformasi)

.setMessage(R.string.strTidakAdaPesanan)

.setNegativeButton(R.string.strOK, new
DialogInterface.OnClickListener() {
        @Override
        public void
onClick(DialogInterface dialogInterface, int i) {
            runThread = false;
            finish();
        }
    });
    builder.show();
}
} else Toast.makeText(Pesanan.this,
R.string.strFail, Toast.LENGTH_LONG).show();
}
}, new Response.ErrorListener() {
    @Override
    public void onResponse(VolleyError error) {

```



```

        Log.e("Send", error.toString());
    }
});
}

private void setDtChat() {
    mdls = new ArrayList<>();
    adp = new ChatAdp(mdls);

    rv.setLayoutManager(rvLm);
    rv.setAdapter(adp);

    Map<String, String> params = new HashMap<>();
    params.put("pesan_idpesan", mdl.idpesan);

    RqsDtc.StrRqsPOSTDtc(this, Cnn.url + "chat/some",
params, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            try {
                JSONArray jsonArray = new
JSONArray(response);
                for (int i = 0; i < jsonArray.length();
i++) {
                    JSONObject jsonObject =
jsonArray.getJSONObject(i);

                    ChatMdl mdl = new ChatMdl();
                    mdl.idchat =
jsonObject.getString("idchat");
                    mdl.pesan =
jsonObject.getString("pesan");
                    mdl.brcl =
jsonObject.getString("brcl");
                    mdl.pesan_idpesan =
jsonObject.getString("pesan_idpesan");

                    mdls.add(mdl);
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }
            adp.notifyDataSetChanged();
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("Dt", error.toString());
        }
    });
}
}

```

```

private void cekUserChat() {
    FirebaseUser firebaseUser = mAuth.getCurrentUser();
    if (firebaseUser != null) {
        clientMdl = new ClientMdl();
        clientMdl.uid = firebaseUser.getId();
        clientMdl.email = firebaseUser.getEmail();
        clientMdl.nama = firebaseUser.getDisplayName();

        firebaseUser.getIdToken(true)
            .addOnCompleteListener(new
OnCompleteListener<GetTokenResult>() {
                @Override
                public void onComplete(@NonNull
Task<GetTokenResult> task) {
                    if (task.isSuccessful()) {

addChat(task.getResult().getToken());
                    }
                }
            });
    } else {
        Intent intent = new Intent(this, Login.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_CLEAR_TASK |
Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
}

private void addChat(String token) {
    Map<String, String> params = new HashMap<>();
    params.put("pesan", etChat.getText().toString());
    params.put("brcl", "cl");
    params.put("pesan_idpesan", mdl.idpesan);
    params.put("token", token);

    RqsDtc.StrRqsPOSTDtc(this, Cnn.url + "chat/ins",
params, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            if (!response.equals("null") &&
!response.equals("0")) {
                ChatMdl mdl = new ChatMdl();
                mdl.idchat = response;
                mdl.pesan = etChat.getText().toString();
                mdl.brcl = "cl";
                mdl.pesan_idpesan =
Pesanan.this.mdl.idpesan;

                mdls.add(mdl);

                adp.notifyItemInserted(mdls.size());
            } else Toast.makeText(Pesanan.this,

```

```

R.string.strFail, Toast.LENGTH_LONG).show());
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("Send", error.toString());
        }
    });
}

private void cekUserDel() {
    FirebaseUser firebaseUser = mAuth.getCurrentUser();
    if (firebaseUser != null) {
        clientMdl = new ClientMdl();
        clientMdl.uid = firebaseUser.getId();
        clientMdl.email = firebaseUser.getEmail();
        clientMdl.nama = firebaseUser.getDisplayName();

        firebaseUser.getIdToken(true)
            .addOnCompleteListener(new
                OnCompleteListener<GetTokenResult>() {
                    @Override
                    public void onComplete(@NonNull
                        Task<GetTokenResult> task) {
                        if (task.isSuccessful()) {

                            del(task.getResult().getToken());
                        }
                    }
                });
    } else {
        Intent intent = new Intent(this, Login.class);
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
            Intent.FLAG_ACTIVITY_CLEAR_TASK |
            Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    }
}

private void del(String token) {
    Map<String, String> params = new HashMap<>();
    params.put("idpesan", mdl.idpesan);
    params.put("token", token);

    RqsDtc.StrRqsPOSTDtc(this, Cnn.url + "pesan/del",
        params, new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                if (!response.equals("null") &&
                    !response.equals("0")) {
                    runThread = false;
                    ntf();
                    finish();
                }
            }
        });
}

```

```

        } else Toast.makeText(Pesanan.this,
R.string.strFail, Toast.LENGTH_LONG).show();
    }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("Send", error.toString());
        }
    });
}

private void ntf() {
    Map<String, String> params = new HashMap<>();
    params.put("tl", "PARTNER BARBERSHOP");
    params.put("bd", "Terdapat perubahan pada antrian
waktu pelayanan, buka aplikasi untuk melihatnya");

    RqsDtc.StrRqsPOSTDtc(this, Cnn.url + "ntfy/notif",
params, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            //
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Log.e("Send", error.toString());
        }
    });
}
}
}

```

Home.xml :

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:background="@drawable/bg"
tools:context=".cnt.Home">

    <android.support.v7.widget.RecyclerView
        android:id="@+id/rv"
        android:layout_width="0dp"
        android:layout_height="0dp"

        app:layout_constraintBottom_toTopOf="@+id/linearLayout2"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

```

```
<LinearLayout
    android:id="@+id/linearLayout2"
    android:layout_width="0dp"
    android:layout_height="75dp"
    android:background="@android:color/white"
    android:orientation="horizontal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent">

    <ImageButton
        android:id="@+id/btnPesanan"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@null"
        android:contentDescription="@string/strBlank"
        app:srcCompat="@drawable/pesanan" />

    <ImageButton
        android:id="@+id/btnInstruction"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@null"
        android:contentDescription="@string/strBlank"
        app:srcCompat="@drawable/instruction" />

    <ImageButton
        android:id="@+id/btnAbout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@null"
        android:contentDescription="@string/strBlank"
        app:srcCompat="@drawable/about" />

    <ImageButton
        android:id="@+id/btnLogout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:background="@null"
        android:contentDescription="@string/strBlank"
        app:srcCompat="@drawable/logout" />
</LinearLayout>

</android.support.constraint.ConstraintLayout>
```

Model_Item.xml :

```
<?xml version="1.0" encoding="utf-8" ?>
<android.support.v7.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:id="@+id/cv"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginBottom="8dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">

        <ImageView
            android:id="@+id/iv"
            android:layout_width="match_parent"
            android:layout_height="250dp"
            android:contentDescription="@string/strBlank"
            android:scaleType="centerCrop" />

        <TextView
            android:id="@+id/tvNama"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:background="@color/colorPrimary"
            android:padding="8dp"
            android:textAlignment="center"
            android:textAllCaps="true"
            android:textColor="@android:color/white"
            android:textStyle="bold" />

    </LinearLayout>
</android.support.v7.widget.CardView>
```