

SKRIPSI

**RANCANG BANGUN APLIKASI PEMESANAN DAN
INFORMASI JASA MAKEUP BERBASIS ANDROID
MENGUNAKAN METODE *RAPID APPLICATION
DEVELOPMENT***

*Sebagai salah satu syarat untuk menyelesaikan Program Studi Sarjana Terapan
Rekayasa Perangkat Lunak Jurusan Teknik Informatika*



Oleh:

**MIRA FAUZIA
6304171044**

**PROGRAM STUDI SARJANA TERAPAN
REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS
2022**

SKRIPSI

**RANCANG BANGUN APLIKASI PEMESANAN DAN
INFORMASI JASA MAKEUP BERBASIS ANDROID
MENGUNAKAN METODE *RAPID APPLICATION
DEVELOPMENT***

*Sebagai salah satu syarat menyelesaikan Program Studi Sarjana Terapan
Rekayasa Perangkat Lunak*



Oleh :

MIRA FAUZIA
6304171044

**PROGRAM STUDI SARJANA TERAPAN
REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
POLITEKNIK NEGERI BENGKALIS
2022**

LEMBAR PENGESAHAN






RANCANG BANGUN APLIKASI PEMESANAN DAN INFORMASI JASA *MAKEUP* BERBASIS ANDROID MENGUNAKAN METODE *RAPID APPLICATION DEVELOPMENT*

*Sebagai salah satu syarat menyelesaikan Program Studi Sarjana Terapan
Rekayasa Perangkat Lunak*

Oleh :

MIRA FAUZIA
6304171044

Disetujui oleh tim penguji skripsi : Tanggal Ujian : 07 Januari 2022
Periode Wisuda : VI

- 
1. Agus Tedyyana, M.Kom (Pembimbing 1)
- 
2. Fajar Ratrawati, M.Cs (Pembimbing 2)
- 
3. M. Asep Subandri, M.Kom (Penguji 1)
- 
4. Lidya Wati, M.Kom (Penguji 2)
- 
5. Fajri Prasanto Putra, M.Cs (Penguji 3)


Bengkalis, 20 Januari 2022
Ketua Program Studi Rekayasa Perangkat Lunak





Rezki Kurniati, M.Kom
NIP. 198306162018032001

LEMBAR PENGESAHAN

Kami dengan sebenarnya menyatakan bahwa, kami telah membaca dari keseluruhan dari Skripsi ini, dan kami berpendapat bahwa Skripsi ini layak dan memenuhi syarat untuk memperoleh gelar sarjana.

Tanda Tangan : 
Nama Penguji 1 : M. Asep Subandri, M.Kom
Tanggal Pengujian : 07 Januari 2022

Tanda Tangan : 
Nama Penguji 2 : Lidya Wati, M.Kom
Tanggal Pengujian : 07 Januari 2022

Tanda Tangan : 
Nama Penguji 3 : Fajri Profesio Putra, M.Cs
Tanggal Pengujian : 07 Januari 2022

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa Skripsi ini adalah asli hasil karya saya dan tidak terdapat karya yang pernah dilakukan untuk memperoleh gelar kesarjanaan di perguruan tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau dipublikasikan oleh orang lain, kecuali yang secara tertulis disebutkan sumbernya dalam naskah dan dalam daftar pustaka.

Bengkalis, 20 September 2021



Mira Fauzia

RANCANG BANGUN APLIKASI PEMESANAN DAN INFORMASI JASA MAKEUP BERBASIS ANDROID MENGGUNAKAN METODE RAPID APPLICATION DEVELOPMENT

Nama Mahasiswa : Mira Fauzia
NIM : 6304171044
Pembimbing 1 : Agus Tedyyana, M.Kom
Pembimbing 2 : Fajar Ratnawati, M.Cs

ABSTRAK

Aplikasi pemesanan jasa *makeup* ini dapat mempermudah pelanggan dalam melakukan pemesanan jasa *makeup* dan mempermudah perias wajah dalam mempromosikan jasa *makeup*. Biasanya pemesanan jasa *makeup* dilakukan secara manual dengan cara datang langsung ketempat perias wajah, bagi pelanggan yang sibuk, tentu akan membutuhkan waktu untuk datang langsung ketempat perias wajah untuk memesan jasanya, pengumpulan data dengan cara menyebarkan kuisisioner kepada pelanggan dan perias wajah. Tujuan dari penelitian ini adalah membuat aplikasi pemesanan dan informasi jasa *makeup* berbasis android menggunakan metode *Rapid Application Development* (RAD). Metode RAD merupakan metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat dan memiliki empat tahapan yaitu fase perencanaan syarat-syarat, fase perancangan, fase kontruksi dan fase pelaksanaan. Pada saat RAD diimplementasikan, maka para pemakai bisa menjadi bagian dari keseluruhan proses pengembangan sistem dengan bertindak sebagai pengambil keputusan pada setiap tahapan pengembangan. Hasil penelitian ini adalah aplikasi pemesanan jasa *makeup* dan informasi jasa *makeup* dengan menerapkan metode *Rapid Application Developmet* yang dapat mempermudah pelanggan dalam melakukan pemesanan jasa *makeup* dan mempermudah perias wajah dalam mempromosikan jasa *makeup*. Aplikasi pemesanan jasa *makeup* memiliki menu beranda, pesanan, dan profil, diaplikasi pengguna bisa melakukan pemesanan, pembayaran, *favorite* perias, *login* dan *input* kategori *makeup*. Pada penelitian ini melakukan pengujian *blackbox testing* dan *compatibility testing*, pengujian dilakukan kepada pelanggan dan perias wajah. Hasil pengujian aplikasi dapat berjalan sesuai yang diharapkan, aplikasi *responsive* pada berbagai tampilan android dengan berbagai versi.

Kata Kunci : Pemesanan, *makeup*, *Rapid Application Development*, Android.

DESIGN AN ANDROID-BASED MAKEUP ORDERING AND INFORMATION APPLICATION USING THE RAPID APPLICATION DEVELOPMENT METHOD

Name : Mira Fauzia
NIM : 6304171044
Advisor 1 : Agus Tedyyana, M.Kom
Advisor 2 : Fajar Ratnawati, M.Cs

ABSTRACT

This application for ordering makeup services can make it easier for customers to order makeup services and make it easier for makeup to promote makeup services. Usually ordering makeup services is done manually by coming directly to the make-up place, for busy customers, of course it will take time to come directly to the make-up place to order their services, data collection by distributing questionnaires to customers and make-up. The purpose of this study is to create an application for ordering and information on android-based makeup services using the Rapid Application Development (RAD) method. The RAD method is a method of developing an information system with a relatively short time and has four stages, namely the requirements planning phase, the design phase, the construction phase and the implementation phase. When RAD is implemented, users can be part of the overall system development process by acting as decision makers at each stage of development. The results of this study are applications for ordering makeup services and information on makeup services by applying the Rapid Application Development method which can make it easier for customers to order makeup services and make it easier for makeup to promote makeup services. The application for ordering makeup services has a homepage, orders, and profile menus, in the application users can make orders, payments, make-up favorites, login and input makeup categories. In this study, blackbox testing and compatibility testing were carried out, testing was carried out on customers and make-up. The results of testing the application can run as expected, the application is responsive on various Android displays with various versions.

Keywords: Ordering, Makeup, Rapid Application Development, Android.

KATA PENGANTAR

Allhamdulillah, Puji syukur kehadiran Allah SWT yang telah memberikan berkat dan rahmat, sehingga penulis dapat menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Pemesanan dan Informasi Jasa *Makeup* Berbasis Android Menggunakan Metode *Rapid Application Development*” tepat waktu. Skripsi ini ditulis guna memenuhi persyaratan untuk dapat menyelesaikan pendidikan di perguruan tinggi.

Penulis menyadari kekurangan serta keterbatasan yang ada. Sehingga dalam menyelesaikan skripsi ini, penulis memperoleh bantuan dari berbagai pihak, dalam kesempatan ini penulis menyampaikan ucapan terimakasih kepada:

1. Bapak Johny Custer, S.T., M.T selaku Direktur Politeknik Negeri Bengkalis.
2. Bapak Armada, S.T., M.T selaku Wakil Direktur I Bagian Akademik Politeknik Negeri Bengkalis.
3. Bapak Guswandi, S.T., M.T selaku Wakil Direktur II Bagian Keuangan, Umum Dan Kepegawaian Politeknik Negeri Bengkalis.
4. Bapak Akmal Indra, S.Pd., M.T selaku Wakil Direktur III bagian Kemahasiswaan Politeknik Negeri Bengkalis.
5. Bapak Danuri, M.Cs selaku Ketua Jurusan Teknik Informatika Politeknik Negeri Bengkalis.
6. Ibu Rezki Kurniati, M.Kom selaku Ketua Program Studi Diploma IV Rekayasa Perangkat Lunak Politeknik Negeri Bengkalis.
7. Bapak Agus Tedyyana, M.Kom selaku dosen Pembimbing I yang selalu memberikan waktu bimbingan dan arahan selama penyusunan skripsi ini.
8. Ibu Fajar Ratnawati, M.Cs selaku dosen Pembimbing II yang selalu memberikan waktu bimbingan dan arahan selama penyusunan skripsi ini.

9. Bapak M.Asep Subandri, M.Kom selaku dosen Penguji I yang selalu memberikan waktu bimbingan dan arahan selama penyusunan skripsi ini.
10. Bapak Lidya Wati, M.Kom selaku dosen Penguji II yang selalu memberikan waktu bimbingan dan arahan selama penyusunan skripsi ini.
11. Bapak Fajri Profesio Putra, M.Cs selaku dosen Penguji III yang selalu memberikan waktu bimbingan dan arahan selama penyusunan skripsi ini.
12. Seluruh Dosen Program Studi Sarjana Terapan Rekayasa Perangkat Lunak.
13. Bapak Ibu Orang Tua dan keluarga yang telah memberikan semangat dan dorongan berupa materi ataupun nasehat.
14. Teman-teman seperjuangan yang telah memberikan motivasi dan semangat.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat banyak kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan. Semoga skripsi ini dapat bermanfaat tidak hanya bagi penulis tetapi juga bagi pembaca.

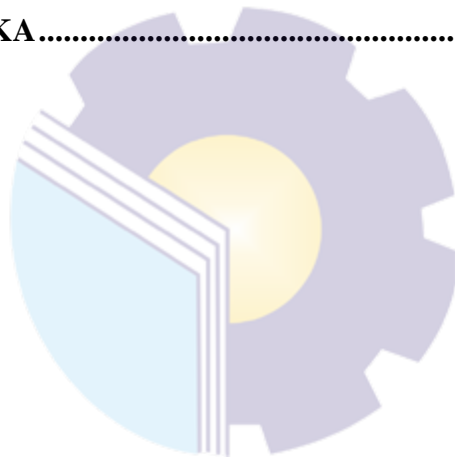
Bengkalis, 24 Desember 2021

Mira Fauzia

DAFTAR ISI

| | |
|--|-------------|
| HALAMAN SAMBUT | i |
| LEMBAR PENGESAHAN | iii |
| LEMBAR PERNYATAAN KEASLIAN | v |
| ABSTRAK | vi |
| ABSTRACT | vii |
| KATA PENGANTAR | viii |
| DAFTAR ISI | x |
| DAFTAR TABEL | xii |
| DAFTAR GAMBAR | xiii |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Rumusan Masalah | 2 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan..... | 3 |
| 1.5 Manfaat..... | 3 |
| BAB II TINJAUAN PUSTAKA | 4 |
| 2.1 Kajian Terdahulu..... | 4 |
| 2.2 Landasan Teori..... | 6 |
| 2.2.1 Aplikasi..... | 6 |
| 2.2.2 Pemesanan | 6 |
| 2.2.3 Informasi..... | 6 |
| 2.2.4 Makeup..... | 7 |
| 2.2.5 Android..... | 7 |
| 2.2.7 Metode Rapid Application Development | 8 |
| BAB III METODOLOGI PENELITIAN | 10 |
| 3.1. Deskripsi Sistem | 10 |
| 3.1.1 Data dan Alat Penelitian | 10 |
| 3.1.2 Rancangan Sistem yang Sedang Berjalan | 11 |
| 3.1.3 Rancangan Sistem yang Diusulkan | 11 |
| 3.2 Alur Metode Rapid Application Development..... | 12 |
| 3.2.1. Fase Perencanaan syarat-syarat..... | 12 |

| | |
|---|-----------|
| 3.2.2. Fase Perancangan | 26 |
| 3.2.3. Fase Konstruksi | 32 |
| 3.2.4. Fase Pelaksanaan | 39 |
| BAB IV HASIL DAN PEMBAHASAN..... | 40 |
| 4.1 Hasil | 40 |
| 4.2 Pembahasan | 47 |
| 4.2.1. Prosedur Pengujian..... | 48 |
| 4.2.2. Hasil Pengujian | 48 |
| 4.2.3 Analisa Pengujian..... | 54 |
| BAB V PENUTUP | 55 |
| 5.1 Kesimpulan | 55 |
| 5.2 Saran..... | 55 |
| DAFTAR PUSTAKA..... | 56 |



DAFTAR TABEL

| | |
|--|----|
| Tabel 3. 1 Kebutuhan Fungsional..... | 18 |
| Tabel 3. 2 Kebutuhan non fungsional..... | 18 |
| Tabel 4. 1 Pengujian form <i>login</i> | 48 |
| Tabel 4. 2 Pengujian form registrasi..... | 50 |
| Tabel 4. 3 Pengujian form kategori <i>makeup</i> | 52 |
| Tabel 4. 4 <i>Compatibility testing</i> | 53 |



DAFTAR GAMBAR

| | |
|---|----|
| Gambar 2.1 Metode rapid application development..... | 9 |
| Gambar 3.2 Rancangan Sistem yang Sedang Berjalan | 11 |
| Gambar 3.3 Rancangan Sistem yang Diusulkan | 11 |
| Gambar 3.4 Rapid Application Development | 12 |
| Gambar 3.5 Hasil Kuisisioner 1 | 12 |
| Gambar 3.5 Hasil Kuisisioner 2 | 13 |
| Gambar 3.7 Hasil Kuisisioner 3 | 13 |
| Gambar 3.8 Hasil Kuisisioner 4 | 13 |
| Gambar 3.9 Hasil Kuisisioner 5 | 14 |
| Gambar 3.10 Hasil Kuisisioner 6 | 14 |
| Gambar 3.11 Hasil Kuisisioner 7 | 15 |
| Gambar 3.12 Hasil kuisisioner 8 | 16 |
| Gambar 3.13 Hasil kuisisioner 9 | 16 |
| Gambar 3.14 Hasil kuisisioner 10 | 16 |
| Gambar 3.15 Hasil kuisisioner 11 | 17 |
| Gambar 3.16 Hasil kuisisioner 12 | 17 |
| Gambar 3.17 Usecase Diagram | 19 |
| Gambar 3.18 Activity diagram pelanggan wajah login | 20 |
| Gambar 3.19 Activity diagram pelanggan wajah registrasi | 21 |
| Gambar 3.20 Activity diagram pelanggan melakukan pemesanan | 22 |
| Gambar 3.21 Activity diagram pelanggan mengelola pesanan | 22 |
| Gambar 3.22 Activity diagram pelanggan melakukan pembayaran..... | 23 |
| Gambar 3.23 Activity diagram pelanggan melakukan pengelolaan profil | 23 |
| Gambar 3.24 Activity diagram perias wajah melihat pesanan | 24 |
| Gambar 3.25 Activity diagram perias wajah mengelola profil | 24 |
| Gambar 3.26 Activity diagram perias wajah mengelola kategori makeup | 25 |
| Gambar 3.27 Perancangan antar muka login pelanggan..... | 26 |
| Gambar 3.28 Perancangan antar muka registrasi pelanggan..... | 27 |
| Gambar 3.29 Perancangan antar muka beranda pelanggan..... | 27 |
| Gambar 3.30 Perancangan antar muka pesanan pelanggan | 28 |
| Gambar 3.31 Perancangan antar muka profil pelanggan | 28 |
| Gambar 3.32 Perancangan antar muka login perias wajah | 29 |
| Gambar 3.33 Perancangan antar muka registrasi perias wajah | 29 |
| Gambar 3.34 Perancangan antar muka beranda perias wajah | 30 |
| Gambar 3.35 Perancangan antar muka pesanan perias wajah..... | 30 |
| Gambar 3.36 Tampilan beranda sebelum melakukan iterasi | 31 |
| Gambar 3.37 Tampilan beranda sesudah melakukan iterasi | 31 |
| Gambar 3.38 Tampilan menu pesanan setelah melakukan iterasi..... | 31 |
| Gambar 3.39 Tampilan menu pesanan sebelum melakukan iterasi..... | 31 |
| Gambar 3.40 Potongan coding login | 32 |
| Gambar 3.41 Tampilan login..... | 32 |

| | |
|--|----|
| Gambar 3.42 Coding registrasi..... | 33 |
| Gambar 3.43 Tampilan registrasi dan verifikasi email..... | 33 |
| Gambar 3.44 Potongan coding halaman beranda..... | 34 |
| Gambar 3.45 Tampilan beranda..... | 34 |
| Gambar 3.46 Coding melihat kategori makeup..... | 35 |
| Gambar 3.47 Tampilan melihat kategori makeup..... | 35 |
| Gambar 3.48 Coding melakukan pemesanan..... | 36 |
| Gambar 3.49 Tampilan melakukan pemesanan..... | 36 |
| Gambar 3.50 Coding tampilan profil..... | 37 |
| Gambar 3.51 Tampilan profil..... | 37 |
| Gambar 3.52 Coding mengelola kategori makeup..... | 38 |
| Gambar 3.53 Tampilan mengelola kategori makeup..... | 38 |
| Gambar 4.1 Tampilan login pelanggan..... | 40 |
| Gambar 4.2 Tampilan registrasi pelanggan dan verifikasi email..... | 41 |
| Gambar 4.3 Tampilan beranda pelanggan..... | 41 |
| Gambar 4.4 Tampilan detail perias wajah..... | 42 |
| Gambar 4.5 Tampilan halaman pemesanan..... | 42 |
| Gambar 4.6 Tampilan daftar pesanan..... | 43 |
| Gambar 4.7 Halaman pembayaran..... | 43 |
| Gambar 4.8 Tampilan profil..... | 44 |
| Gambar 4.9 Tampilan login perias wajah..... | 44 |
| Gambar 4.10 Tampilan registrasi perias wajah..... | 45 |
| Gambar 4.11 Tampilan halaman beranda perias wajah..... | 45 |
| Gambar 4.12 Tampilan daftar pesanan perias wajah..... | 46 |
| Gambar 4.13 Halaman profil perias wajah..... | 46 |
| Gambar 4.14 Halaman validasi pembayaran..... | 47 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Di masa ini berpenampilan menarik dan cantik menjadi kebutuhan sehari-hari terutama pada wajah saat menghadiri acara seperti acara wisuda, pernikahan, maupun acara lainnya. Hal yang dilakukan untuk membuat penampilan menjadi menarik, dan terkesan lebih segar saat menghadiri suatu acara salah satunya dengan cara menggunakan *makeup*. *Makeup* merupakan seni merias wajah atau mengubah bentuk asli dengan bantuan alat dan kosmetik yang bertujuan untuk memperindah serta menutupi kekurangan sehingga wajah terlihat ideal (Elianti dan Pinasti, 2017). Penggunaan *makeup* akan membuat seseorang terlihat lebih segar dan tidak terlihat pucat saat menghadiri suatu acara baik itu wisuda, pernikahan, wisuda, maupun acara lainnya. Kurangnya pengalaman dan pengetahuan tentang cara menggunakan *makeup* sehingga pelanggan menggunakan jasa perias wajah.

Di masa ini kebanyakan masyarakat masih menggunakan *search engine* untuk mencari penyedia jasa yang cocok untuk mereka dan bahkan hanya menggunakan penyedia jasa *makeup* yang diceritakan oleh orang-orang sekitarnya (Egadhana dkk, 2019). Seorang perias wajah di Bengkalis biasanya menawarkan atau mempromosikan jasanya melalui media sosial dan dari mulut ke mulut, kekurangan dari mulut ke mulut ialah pelanggan tidak bisa melihat hasil *makeup* dari perias, di Bengkalis belum ada aplikasi khusus untuk menawarkan jasa *makeup*. Informasi yang ditampilkan di media sosial masih belum lengkap seperti harga dan gambar hasil *makeup* dan melalui media sosial perias wajah sulit mencatat data pesanan karena harus melihat isi pesan dari pelanggan satu persatu. Melalui media sosial pelanggan sulit mendapatkan perias wajah sesuai yang diinginkan dan melalui media sosial terbilang rumit karena harus mengingat satu-satu nama akun sosial perias wajah, untuk mendapatkan informasi, memilih dan memesan. Biasanya pelanggan memesan jasa *makeup* untuk menghadiri acara seperti acara wisuda, pernikahan, maupun acara lainnya. Biasanya pemesanan jasa

makeup dilakukan dengan cara datang langsung ketempat perias wajah dan pelanggan membayar secara tunai. Kurangnya pengalaman pelanggan dan pengetahuan tentang cara menggunakan *makeup* sehingga pelanggan menggunakan jasa perias wajah, sedikitnya informasi mengenai jasa *makeup*, harga yang ditawarkan, serta bagaimana pemesanannya sehingga diperlukan pemecahan masalahnya.

Rapid Application Development (RAD) adalah salah satu metode pengembangan suatu sistem informasi dengan waktu yang relatif singkat. Untuk pengembangan suatu sistem informasi yang normal membutuhkan waktu minimal 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya dalam waktu 30 sampai 90 hari. Untuk pengembangan suatu sistem informasi yang normal membutuhkan waktu minimal 180 hari, akan tetapi dengan menggunakan metode RAD suatu sistem dapat diselesaikan hanya lebih singkat. Pada saat RAD diimplementasikan, maka para pemakai bisa menjadi bagian dari keseluruhan proses pengembangan sistem dengan bertindak sebagai pengambil keputusan pada setiap tahapan pengembangan. RAD bisa menghasilkan suatu sistem dengan cepat karena sistem yang dikembangkan dapat memenuhi keinginan dari para pemakai sehingga dapat mengurangi waktu untuk pengembangan ulang setelah tahap implementasi (Purwanto dan Hendrit, 2018).

Dari masalah di atas peneliti tertarik untuk membuat sebuah aplikasi pemesanan dan informasi jasa *makeup* berbasis android menggunakan metode *Rapid Application Development*.

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah, maka rumusan masalah dalam penelitian ini adalah bagaimana menerapkan metode *Rapid Application Development* dalam pembuatan aplikasi pemesanan jasa *makeup* berbasis android di wilayah Bengkalis.

1.3 Batasan Masalah

Adapun batasan masalah dalam penelitian ini yaitu:

1. Aplikasi pemesanan dan informasi jasa *makeup* ini hanya memiliki dua aktor yaitu perias wajah dan pelanggan yang ada di Bengkalis.

2. Aplikasi yang dibuat berbasis android.

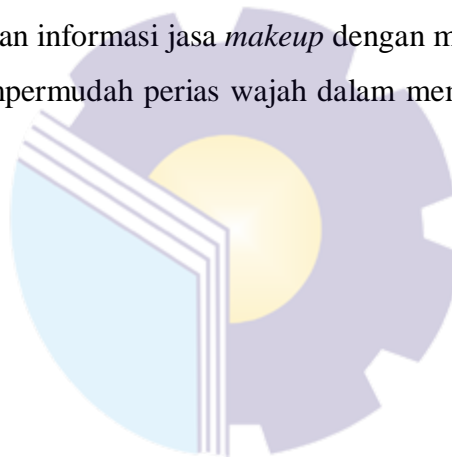
1.4 Tujuan

Tujuan dari penelitian ini adalah membuat aplikasi pemesanan dan informasi jasa *makeup* berbasis android menggunakan metode *Rapid Application Development*.

1.5 Manfaat

Dalam penelitian ini dapat diambil beberapa manfaat yang mencakup 4 hal pokok berikut:

1. Dapat mempermudah pelanggan dalam melakukan pemesanan jasa *makeup*
2. Dapat memilih jenis *makeup* sesuai yang di inginkan
3. Mendapatkan informasi jasa *makeup* dengan mudah dan cepat
4. Dapat mempermudah perias wajah dalam mempromosikan jasa *makeup*.



BAB II TINJAUAN PUSTAKA

2.1 Kajian Terdahulu

Dalam bab ini, peneliti akan memaparkan beberapa kajian terdahulu atau penelitian terdahulu yang berkaitan dengan penelitian ini. Adapun tujuan dari pemaparan kajian terdahulu ini adalah untuk menjelaskan perbedaan dan sangat berguna untuk perbandingan. Adapun penelitian terdahulu yang di maksud adalah:

Penelitian yang pertama dengan judul Perancangan Antar Muka Aplikasi Pemesanan Fotografi dan *Makeup* Berbasis *Online* Pada *Queen Photo Studio Padang*. Penelitian yang dilakukan berfokus pada pemesanan fotografi dan *makeup*, sistem yang dibuat adalah berbasis *website* bertujuan agar mempermudah pihak *queen* dan pelanggan. Metode yang digunakan dalam pengembangan sistem adalah *web engineering*, metode ini merupakan rekayasa web menggunakan konsep rekayasa perangkat lunak secara teknis pada aktifitasnya. Tujuan *web engineering* ini adalah menyusun sistem yang baru untuk memecahkan masalah pada sistem yang lama dan menggunakan metode OOP dengan model air terjun (*waterfall*) dan menggunakan *Unified Modeling Language* sebagai alat bantu pengembangan *system* (Arman dkk, 2019).

Penelitian yang kedua dengan judul Rancang Bangun Aplikasi Penentuan Vendor Pernikahan Pelanggan Untuk Usaha *Wedding Organizer* Menggunakan Metode *Smarter* Berbasis Web. Penelitian ini membantu calon pengantin untuk merencanakan dan melaksanakan pesta pernikahan sesuai dengan yang diimpikan dan telah menjalin kerja sama dengan Ninda *Makeup Artist* dan bisa melakukan pemesanan, sistem ini berbasis *website*. Untuk melakukan pembuatan aplikasi penentuan vendor pernikahan pelanggan untuk usaha *wedding organizer* menggunakan metode *smarter* berbasis web, metode yang digunakan adalah *waterfall* (Cahyo dkk, 2018).

Penelitian yang ketiga dengan judul Implementasi Metode *Rapid Application Development* Dalam Membangun *E-Commerce* Di Bidang UKM. Penelitian ini membantu dalam pengambilan keputusan dan menetapkan langkah-

langkah selanjutnya yang harus dilakukan untuk mempertahankan dan mengembangkan organisasi dan usahanya, selain itu informasi juga mendukung kegiatan operasional dan manajerial suatu organisasi. Menggunakan *Metode Rapid Application Development* (Riyanto, 2017).

Penelitian yang keempat dengan judul Peningkatan Pengetahuan *Makeup* Dan *Mehndi* (*Go Salon Muslimah Berbasis Android*) Untuk Menciptakan Kewirausahaan Mahasiswa Dan Alumni Jurusan Tata Rias Dan Kecantikan UNP. Penelitian ini membantu meningkatkan dan mengembangkan pengetahuan dan keterampilan mahasiswa dan alumni mengenai *makeup* pengantin, wisuda dan *mehndi*, serta meningkatkan keterampilan dalam mengelola usaha salon muslimah, menerapkan *hygiene* dan sanitasi ruang salon muslimah, serta etika sebagai penata rias yang profesional, berwirausaha secara mandiri dengan keterampilan *makeup* pengantin, wisuda dan *mehndi*, serta manajemen usaha secara profesional, dapat menjalankankan usaha dengan berbasis android melalui instagram dan *go-message*(Rahmiati dkk, 2019).

Penelitian yang kelima dengan judul Rancang Bangun Sistem Informasi Pemesanan Ceker Mercon Ibu Elly Menggunakan Metode *Rapid Application Development* Berbasis Android. Penelitian ini dilakukan untuk dapat membuat segala informasi tentang daftar makanan, daftar minuman, perhitungan pembayaran oleh pelanggan pada kasir. metode yang di gunakan yaitu RAD (*rapid application develovment*), RAD adalah metode model proses pengembangan perangkat lunak yang bersifat *incremental* terutama untuk waktu pengerjaan jangka pendek. Pemakaian sistem pemesanan makanan untuk memudahkan proses kerja dalam lingkungan restoran ceker mercon ibu elly, dengan adanya sistem pemesanan makanan di restoran ceker mercon ibu elly berbasis web *service* menggunakann *mobile android* sebagai peningkatan dan penyempurnaan sarana dan prasarana yang telah ada sehingga dapat meningkatkan kinerja karyawan dalam melayani pelanggan dan secara tidak langsung dapat meningkatkan pemasukan di restoran ceker mercon ibu elly (Purwanto dan Hendrit, 2018).

Berdasarkan kajian terdahulu peneliti membuat sebuah aplikasi pemesanan jasa *makeup* yang mempermudah pengguna dalam melakukan pemesanan dan mempermudah perias wajah dalam mempromosi jasa *makeup*.

2.2 Landasan Teori

2.2.1 Aplikasi

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi tersebut, aplikasi mempunyai arti yaitu pemecahan masalah yang menggunakan salah satu teknik pemrosesan data aplikasi yang biasanya berpaku pada sebuah komputansi yang diinginkan atau diharapkan maupun pemrosesan data yang diharapkan. Pengertian aplikasi secara umum adalah alat terapan yang difungsikan secara khusus dan terpadu sesuai kemampuan yang dimilikinya, aplikasi merupakan suatu perangkat komputer yang siap pakai bagi *user* (Novendri dkk, 2019).

2.2.2 Pemesanan

Pemesanan adalah proses perbuatan, cara memesan atau memesankan. Pemesanan merupakan salah satu aktifitas dalam sebuah perusahaan rental buku, pemesanan dilakukan oleh konsumen, adanya pemesanan membantu penjual menentukan jumlah persediaan barang. Kemajuan teknologi sekarang ini memungkinkan pemesanan tidak hanya dapat dilakukan jika konsumen dan penjual bertatap langsung, tetapi dapat dilakukan dari jarak jauh sekalipun, misalnya dengan menggunakan *E-commerce*, yaitu sistem pemesanan barang atau jasa lewat internet (Janah dan Syafitri, 2019).

2.2.3 Informasi

Informasi merupakan hasil pemrosesan data (fakta) menjadi sesuatu yang bermakna dan bernilai untuk pengambilan keputusan. Informasi tidak dapat terlepas dari aspek kehidupan manusia. Siapa, kapan dan dimanapun seseorang akan membutuhkan informasi (Asinjery, 2020).

2.2.4 Makeup

Makeup adalah seni merias wajah atau mengubah bentuk asli dengan bantuan alat dan kosmetik yang bertujuan untuk memperindah serta menutupi kekurangan sehingga wajah terlihat ideal (Elianti dkk, 2017). Istilah *makeup* lebih sering ditujukan kepada pengubahan bentuk wajah, meskipun sebenarnya seluruh tubuh bisa dihias (*makeup*). Untuk merias dengan kosmetik, sebelumnya dibutuhkan banyak pengetahuan tentang:

1. Anatomi fisiologi kulit dan bentuk wajah
2. Karakterisasi Warna dan Garis untuk memberikan karakterisasi personal
3. Gradasi Warna untuk memperhalus hasil akhir tata rias
4. Komposisi Warna guna menciptakan hasil riasan yang tampak natural dan Sempurna.

Makeup sekarang ini telah menjadi bagian dari rutinitas kehidupan masyarakat modern khususnya bagi kaum wanita. Penggunaan tata rias wajah sendiri sudah berkembang menjadi suatu kebutuhan untuk mempercantik diri, menunjukkan jati diri dan kepribadian, serta untuk mengikuti perkembangan mode terutama di dunia *entertainment* di mana berkumpulnya para professional *makeup artist*. Perlu diketahui, bahwa penggunaan tata rias wajah bukan hanya dapat dilakukan oleh seorang professional *makeup artist* (MUA) saja. Wanita biasa pun dapat menggunakan tata rias wajah guna menunjukkan kepedulian terhadap penampilannya, dalam memenuhi aktifitas kehidupannya sehari-hari layaknya seorang professional *makeup artist* (MUA). Namun penggunaan tata rias wajah ini pun memiliki tahapan tertentu yang tidak boleh disepelekan. Pemesanan jasa *makeup* dilakukan dengan cara datang langsung ke perias wajah atau melalui sosial media (Puspa, 2013).

2.2.5 Android

Android adalah sebuah sistem operasi perangkat mobile berbasis *linux* yang mencakup sistem operasi, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google inc.* Membeli *Android In Vang* merupakan pendatang baru yang membuat perangkat lunak untuk ponsel atau smartphone kemudian untuk dalam

mengembangkan android, dibentuk lah *open handset alliance*, konsorsium dari 34 perusahaan peranti keras perangkat lunak dan telekomunikasi, termasuk *Google, HTC, Intel, Motorola, Qualcomm, T-Mobile, dan Nvidia* (Karman dkk, 2019).

2.2.6 Java

Java adalah sebuah teknologi yang diperkenalkan *Microsystem* pada pertengahan tahun 1990. Menurut definisi sun, *Java* adalah nama yang digunakan sekumpulan teknologi untuk menjalankan perangkat lunak pada komputer *standalone* ataupun pada lingkungan jaringan (Bay Haqi, 2019). Teknologi *java* memiliki tiga komponen penting , yaitu :

1. *Java Development Kit*

Java Development Kit merupakan perangkat lunak yang digunakan untuk melakukan proses kompilasi dari kode *Java* menjadi *bytecode* yang dapat mengerti dan dapat dijalankan oleh *java runtime environtment*. *Java development kit* wajib terinstall pada komputer yang akan melakukan proses pembuatan aplikasi berbasis *java*. Namun *java development kit* tidak wajib terinstall di komputer yang akan menjalankan aplikasi yang dibangun menggunakan *java*.

2. *Java Runtime Environtment*

Java Runtime Environtment merupakan perangkat lunak yang digunakan untuk menjalankan aplikasi yang dibangun menggunakan *java*. Versi JRE harus sama atau lebih tinggi dari JDK yang digunakan untuk membangun aplikasi agar aplikasi dapat berjalan sesuai dengan yang diharapkan.

3. *NetBeans IDE*

NetBeans IDE merupakan perangkat lunak yang digunakan membangun perangkat lunak yang lain. *NetBeans IDE* digunakan untuk membangun perangkat lunak berbasis *java standard edition, java enterprise edition, java micro editon, C++,PHP, ruby, groovy dan python*.

2.2.7 Metode Rapid Application Development

Metode *Rapid Application Development* adalah sebuah strategi pengembangan sistem yang menekankan kecepatan pengembangan melalui keterlibatan pengguna yang ekstensif dalam konstruksi, cepat, berulang dan bertambah serangkaian *prototype* bekerja sebuah sistem yang pada akhirnya

berkembang kedalam sistem final model pengembangan RAD memiliki empat fase, yaitu fase perencanaan syarat-syarat, fase perancangan, fase konstruksi, dan fase pelaksanaan (Sintawati, 2018). Adapun tahapan-tahapan dalam metode *rapid application development* yaitu:



Gambar 2.1 Metode *rapid application development*
(Sumber: Data Olahan)

1. Fase Perencanaan syarat-syarat

Pada tahap ini dilakukan pengidentifikasian tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. pada tahap ini peneliti melakukan analisis kebutuhan.

2. Fase Perancangan

Pada tahap ini adalah melakukan proses desain dan melakukan perbaikan-perbaikan apabila masih terdapat ketidaksesuaian desain antara *user* dan *analyst*. Untuk tahap ini keaktifan *user* yang terlibat sangat menentukan untuk mencapai tujuan, karena *user* bisa langsung memberikan komentar apabila terdapat ketidaksesuaian desain.

3. Fase Konstruksi

Setelah desain dari sistem yang akan dibuat sudah disetujui baik itu oleh *user* dan *analyst*, maka tahap ini *programmer* mengembangkan desain menjadi suatu program. Hal terpenting adalah keterlibatan *user* sangat diperlukan supaya sistem yang dikembangkan dapat sesuai dengan permintaan *user*.

4. Fase Pelaksanaan

Pada tahap ini dilakukan pengujian aplikasi, dengan menggunakan pengujian *black box testing* yaitu sebuah pengujian dengan menjalankan setiap fungsi dan aplikasi.

BAB III

METODOLOGI PENELITIAN

3.1. Deskripsi Sistem

3.1.1 Data dan Alat Penelitian

Pada pembangunan aplikasi ini dibutuhkan beberapa data untuk mendukung proses pembuatan aplikasi. Adapun data yang dibutuhkan yaitu data perias wajah, Data jenis *makeup* (kategori *makeup*) dan data pelanggan.

Adapun alat yang digunakan dalam melakukan penelitian ini yaitu:

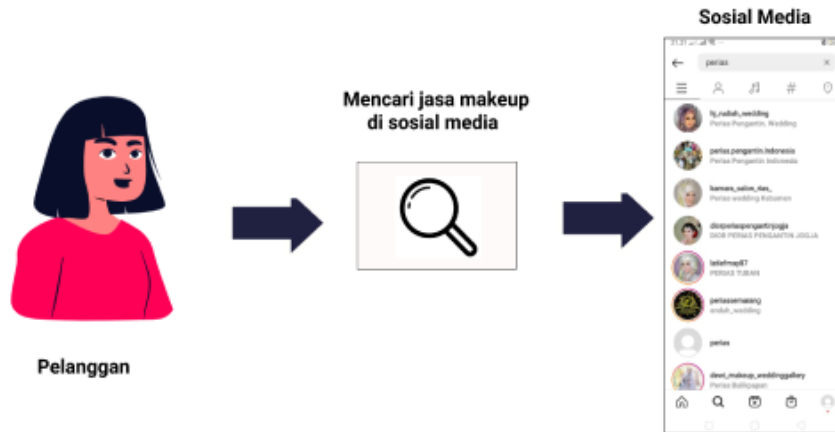
1. *Hardware*

- b. Laptop RAM 8 GB *Processor Intel(R) Core i5*
- c. *Flasdisk* 16 GB
- d. *Hardisk* 120 GB Rpm 7200 Serial ATA
- e. *Keyboard*
- f. *Mouse*
- g. *Smartphone* versi 9 RAM 2 *Processor Eight Core*.

2. *Software*

1. *Microsoft Windows* 10 sebagai sistem operasinya.
2. *Star uml* sebagai *software* pemodelan yang mendukung UML (*Unified Modelling Language*).
3. *Figma* sebagai *design tool* yang biasanya digunakan untuk membuat tampilan aplikasi *mobile, desktop, website*.
4. *Android studio* sebagai *tool* dalam pembuatan aplikasi.
5. *Java* sebagai bahasa pemrograman dalam pembuatan aplikasi.

3.1.2 Rancangan Sistem yang Sedang Berjalan

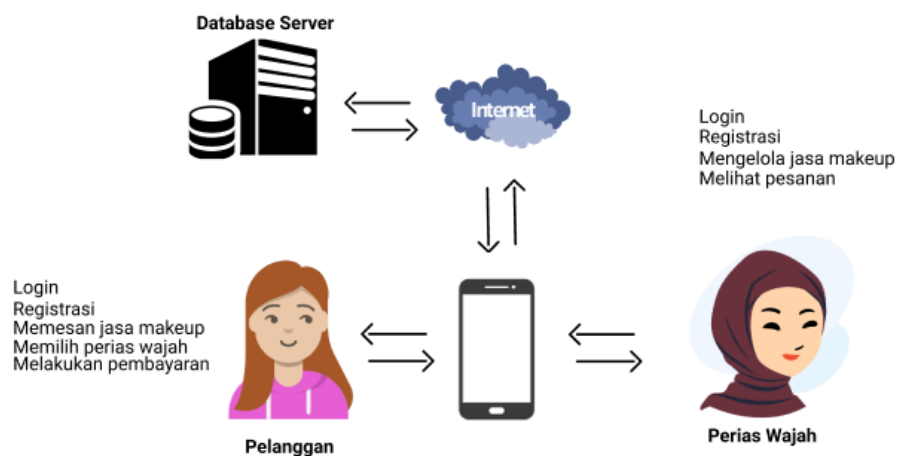


Gambar 3.2 Rancangan Sistem yang Sedang Berjalan
(Sumber: Data Olahan)

Adapun sistem yang berjalan saat ini yaitu pelanggan mencari informasi dan memesan jasa *makeup* melalui media sosial, di mana melalui sosial media Informasi yang ditampilkan di media sosial masih belum lengkap seperti harga dan gambar hasil *makeup* dan melalui media sosial perias wajah sulit mencatat data pesanan karena harus melihat isi pesan dari pelanggan satu persatu.

3.1.3 Rancangan Sistem yang Diusulkan

Dari hasil analisa maka diperoleh rancangan dari aplikasi pemesan dan informasi jasa *makeup* yang diusulkan, maka dapat dilihat pada gambar di bawah ini:



Gambar 3.3 Rancangan Sistem yang Diusulkan
(Sumber: Data Olahan)

Sistem yang diusulkan memiliki dua aktor yaitu pelanggan dan perias wajah. Untuk aktor pelanggan bisa melakukan login, jika belum punya akun pelanggan bisa melakukan daftar untuk bisa masuk kedalam aplikasi, pelanggan bisa memesan jasa make up dengan cara memilih perias wajah yang diinginkan lalu mencari jenis *makeup* atau kategori sesuai kebutuhan yang diinginkan dan pelanggan bisa melakukan pembayaran. Untuk aktor perias wajah bisa melakukan login, daftar, mengelola jenis *makeup* yaitu input jenis *makeup*, edit dan hapus dan perias bisa melihat pesanan yang dirinya di pesan.

3.2 Alur Metode *Rapid Application Development*

Adapun tahapan dari Metode *Rapid Application Development* dapat dilihat pada gambar dibawah ini:



Gambar 3.4 *Rapid Application Development*
(Sumber: Data olahan)

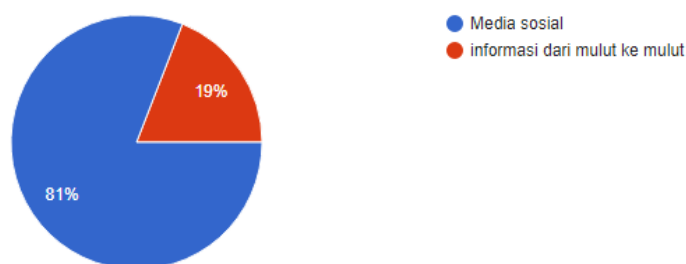
3.2.1. Fase Perencanaan syarat-syarat

Pada tahapan melakukan pengumpulan data, adapun metode pengumpulan data yang digunakan dalam menyelesaikan penelitian ini, yaitu dengan menyebarkan kuisisioner kepada pelanggan dan perias wajah untuk mengetahui keinginan pengguna aplikasi informasi dan pemesanan jasa *makeup*.

1. Kuisisioner Pelanggan

Pada saat ini dimana anda mendapatkan informasi mengenai jasa *makeup*?

21 jawaban

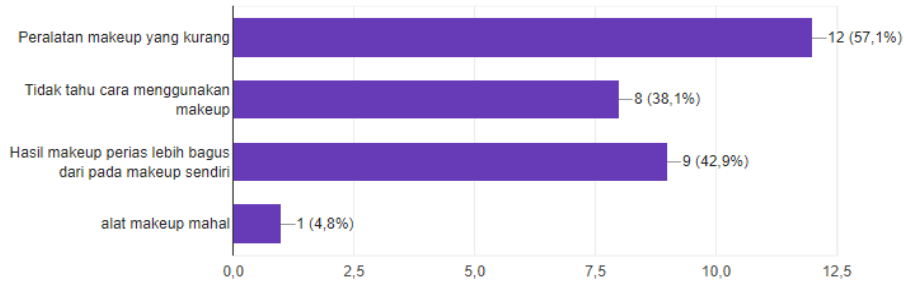


Gambar 3.5 Hasil Kuisisioner 1
(Sumber: Data olahan)

Dari data diatas dapat disimpulkan bahwa 81 % pelanggan mendapat informasi jasa *makeup* melalui media sosial.

Alasan menggunakan jasa *makeup*?

21 jawaban

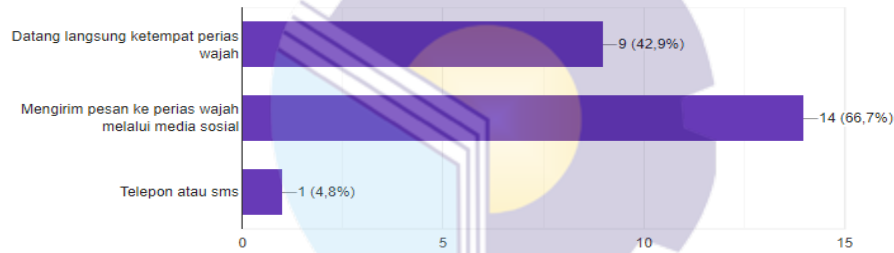


Gambar 3.5 Hasil Kuisisioner 2
(Sumber: Data olahan)

Dari data diatas alasan pengguna memesan jasa *makeup* paling banyak ialah peralatan *makeup* yang kurang sehingga membutuhkan jasa perias wajah.

Bagaimana cara anda melakukan pemesanan jasa *makeup*?

21 jawaban

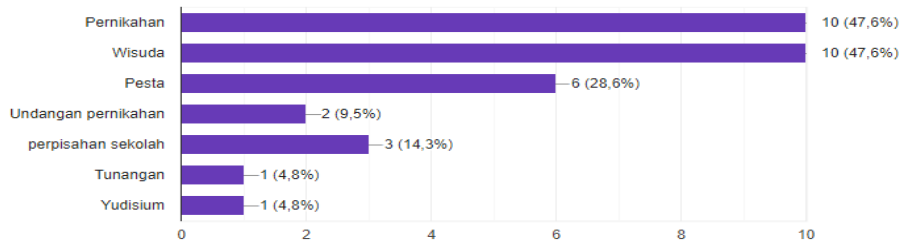


Gambar 3.7 Hasil Kuisisioner 3
(Sumber: Data olahan)

Dari data diatas cara pelanggan melakukan pemesanan jasa *makeup*, paling banyak dengan cara mengirim pesan kepada perias wajah melalui media sosial perias wajah.

Biasanya memesan jasa *makeup* untuk acara apa?

21 jawaban

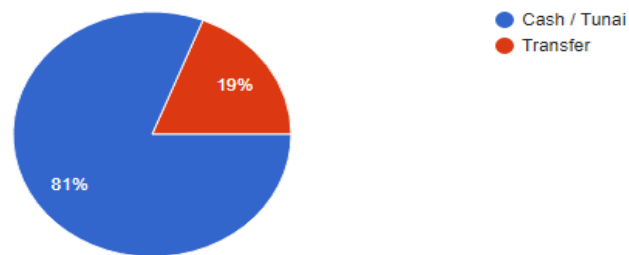


Gambar 3.8 Hasil Kuisisioner 4
(Sumber: Data olahan)

Dari data diatas pelanggan memesan jasa *makeup* untuk menghadiri acara seperti pernikahan, pertunangan, hantaran, undangan pernikahan, perpisahan sekolah, wisuda, yudisium dan pesta, dari data diatas pengguna paling banyak menggunakan jasa *makeup* untuk acara pernikahan dan wisuda.

Bagaimana cara anda membayar perias wajah?

21 jawaban

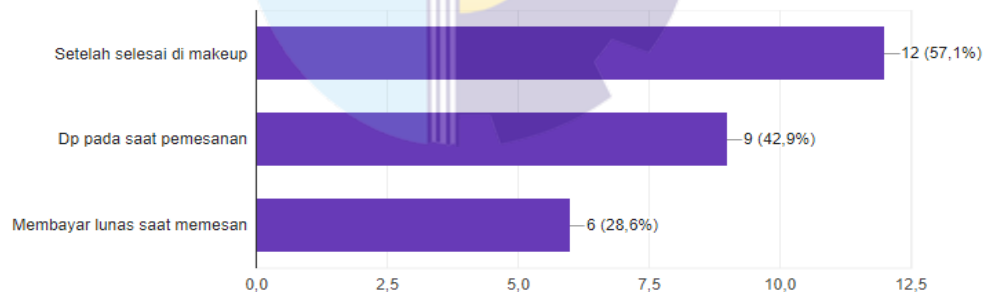


Gambar 3.9 Hasil Kuisiner 5
(Sumber: Data Olahan)

Dari data diatas 85 % pelanggan membayar perias wajah dengan cara *cash* atau tunai, dan 19 % melalui transfer antar *bank*.

kapan anda melakukan pembayaran?

21 jawaban



Dari data di atas pelanggan melakukan pembayaran setelah selesai di *makeup*, bayar setengah pada saat memesan dan membayar lunas saat memesan, paling banyak membayar setelah selesai di *makeup*.

bagaimana menurut pendapat anda jika ada aplikasi pemesanan jasa *makeup* dalam bentuk satu aplikasi?

21 jawaban

- Oke
- Android
- Sangat membantu untuk masyarakat

| |
|--|
| bagus, karna lebih mudah dan tidak perlu keliling lagi untuk mencari tempat pemesanan jasa makeup |
| Sangat membantu |
| Menurut saya bagus, itu sangat membantu karena lebih memudahkan pengguna untuk mengetahui informasi² jasa makeup yang sesuai dengan keinginan kita |
| Bagoss |
| sangat membantu |
| Sangat mempermudah saya Lebih baik lagi, sehingga pelanggan tidak susah datang ke tempat makeupmya lagi |
| Bagus, membantu dlam mencari jasa make up |
| sangat bagusss dan menarik |
| sangat membantu dalam melakukan jadi tidak harus mencari satu-satu jika ada aplikasi nya |
| sangat bagus memudah kan melihat jasa makeup apalagi jika bisa membayar melalui aplikasi |
| menurut pendapat saya itu bagus karena bisa membantu perias untuk memperkenalkan jasa nya |
| tentu nya akan membantu karena banyak pilihan makeup nya |
| ide yang baik tambahkan pembayarann |
| perbanyak informasi tukang makeup |
| menhemat waktu dalam mencari tukangg makeup dan juga bagus krna pilihan tukang makeup nya tidak hanya satu |

Gambar 3.11 Hasil Kuisisioner 6
(Sumber: Data Olahan)

Dari jawaban pelanggan jika ada aplikasi pemesanan jasa *makeup* dibuat dalam bentuk satu aplikasi, beberapa menjawab dengan adanya aplikasi tentunya sangat membantu pelanggan dalam mencari jasa *makeup* dan membantu perias dalam mempromosikan jasanya dan beberapa menyarankan fitur pembayaran di aplikasi.

Fitur apa yang diinginkan pada aplikasi pemesanan jasa makeup?

21 jawaban

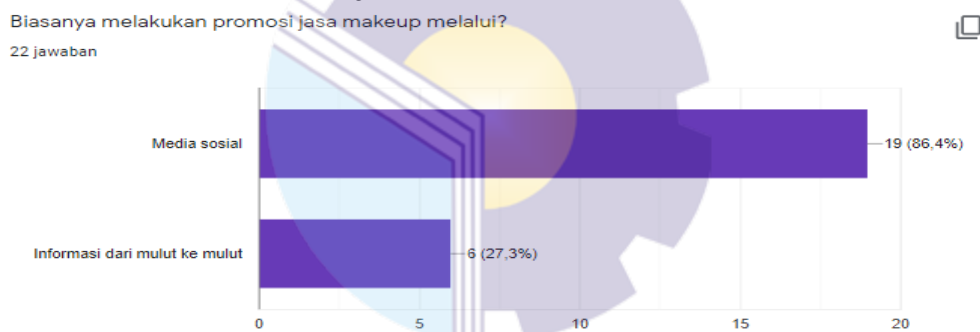
| |
|--|
| pesan |
| bayar makeup |
| bayar |
| pembayaran makeup |
| Ada informasi harga |
| fitur untuk pemesanan |
| Memilih perias wajah yg kita ingin kan |
| Gambar makeup |
| diskon |

| |
|--|
| fitur yg mempermudah mesan makeup |
| memudahkan pelanggan |
| pemesanann |
| androidd |
| pesanan |
| Tampilan menarik dan ada banyak pilihan perias |
| bisa memesan |
| gambar |
| warna aplikasi yang menarik |

Gambar 3.12 Hasil kuisiонер 7
(Sumber: Data Olahan)

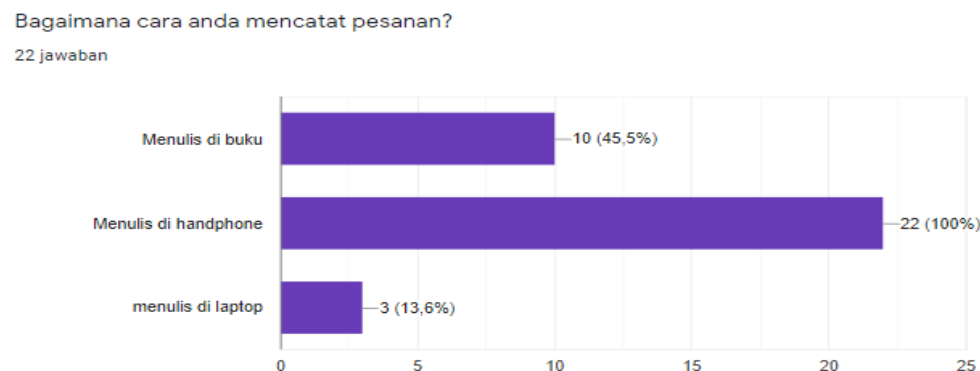
Dari jawaban pelanggan diatas ada beberapa fitur yang diinginkan, antara lain yaitu pemesanan, pembayaran dan menampilkan gambar *makeup*.

2. Kuisiонер Perias Wajah



Gambar 3.13 Hasil kuisiонер 8
(Sumber: Data Olahan)

Dari data diatas dapat disimpulkan bahwa perias melakukan promosi jasa *makeup* paling banyak melalui media sosial.

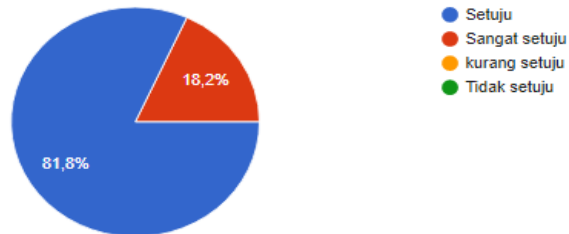


Gambar 3.14 Hasil kuisiонер 9
(Sumber: Data Olahan)

Dari data diatas dapat disimpulkan bahwa perias mencatat pesanan dibuku, handphone, dan laptop, paling banyak menulis menulis pesan di *handphone*.

Bagaimana pendapat anda jika ada aplikasi pemesanan jasa makeup?

22 jawaban



Gambar 3.15 Hasil kuisisioner 10
(Sumber: Data Olahan)

Dari data di atas perias wajah setuju jika ada aplikasi pemesanan jasa makeup, sebanyak 81 % menjawab setuju.

fitur apa yang bisa membantu anda dalam aplikasi ini?

22 jawaban



Gambar 3.16 Hasil kuisisioner 11
(Sumber: Data Olahan)

Dari data di atas perias wajah menginginkan beberapa fitur yaitu fitur pemesanan, promosi, daftar pesanan atau catatan pesanan dan profil.

Dari pengumpulan data yang dilakukan maka didapat kebutuhan fungsional dan non fungsional dari sistem yang akan dibuat antara lain:

3. Kebutuhan fungsional

Kebutuhan fungsional berkaitan dengan fungsi yang bisa dikerjakan di dalam aplikasi, berikut kebutuhan fungsional di aplikasi pemesanan dan informasi jasa makeup:

Tabel 3.1 Kebutuhan Fungsional

| Kode | Keterangan |
|-------|--|
| KF-01 | Pelanggan dan perias wajah melakukan <i>login</i> |
| KF-02 | Pelanggan dan perias wajah melakukan registrasi |
| KF-03 | Pelanggan dapat melakukan pemesanan jasa <i>makeup</i> |
| KF-04 | Pelanggan dapat memilih perias wajah yang di inginkan |
| KF-05 | Pelanggan dan perias dapat melihat pesanan |
| KF-06 | Pelanggan dapat membatalkan pesanan |
| KF-07 | Pelanggan dapat melihat perias wajah |
| KF-08 | Pelanggan dapat melakukan pembayaran |
| KF-09 | Pelanggan dan perias dapat mengisi data diri |
| KF-10 | Pelanggan dan perias dapat mengubah data diri |
| KF-11 | Perias dapat menginput pilihan <i>makeup</i> |
| KF-12 | Perias dapat mengubah pilihan <i>makeup</i> |
| KF-13 | Perias dapat menghapus pilihan <i>makeup</i> |
| KF-14 | Perias dapat melihat pilihan <i>makeup</i> |
| KF-15 | Perias dapat melihat pembayaran |

4. Kebutuhan non fungsional

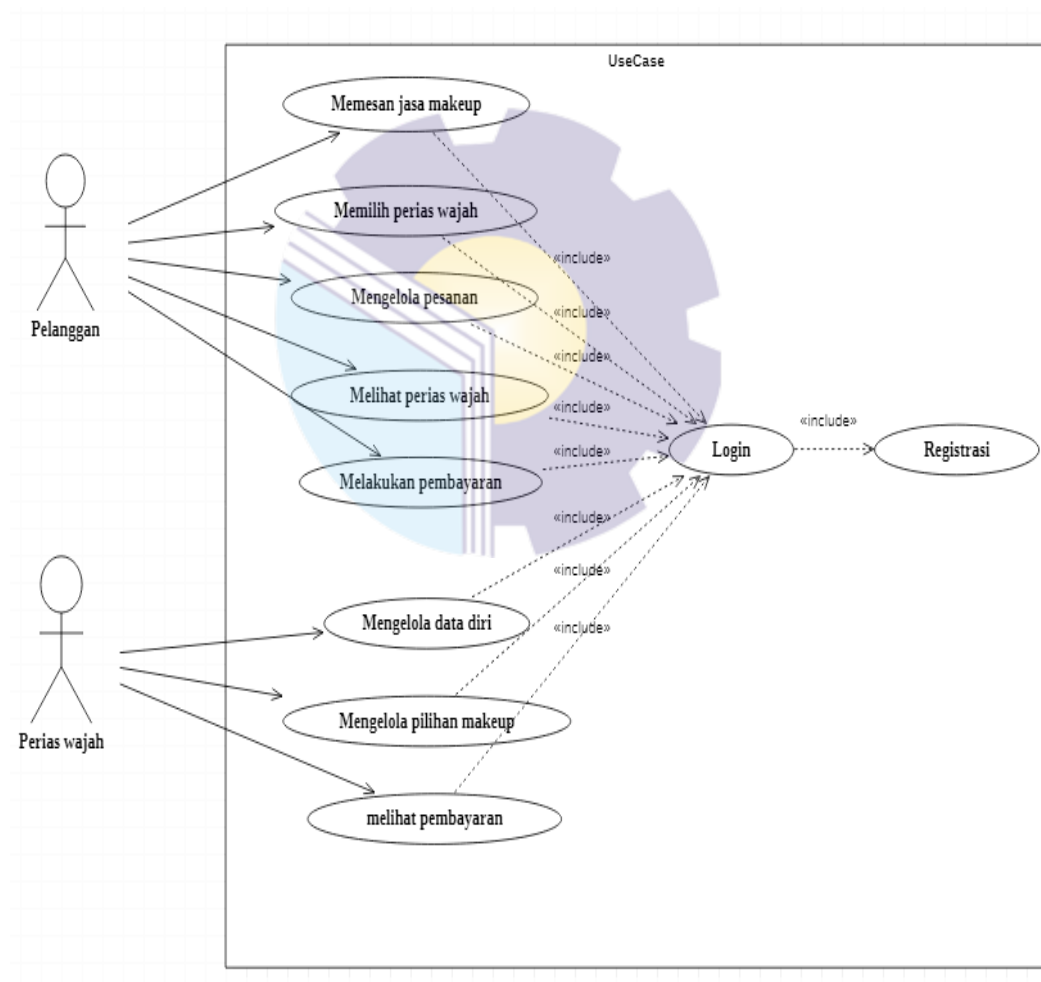
Kebutuhan non fungsional menjelaskan batasan-batasan fungsi yang dikerjakan dan harus dipenuhi di aplikasi yang dikembangkan.

Tabel 3.2 Kebutuhan non fungsional

| Kode | Keterangan |
|--------|---|
| KNF-01 | Perias wajah sebelum <i>login</i> harus registrasi terlebih dahulu. |
| KNF-02 | Sistem dapat menampilkan pilihan <i>makeup</i> |
| KNF-03 | Sistem dapat melihat info pelanggan dan perias wajah |
| KNF-04 | Sistem dapat menampilkan data yang telah di pesan |
| KNF-05 | Sistem dapat melayani <i>request</i> dari <i>user</i> dengan batas waktu kurang lebih 10 detik. |
| KNF-06 | Sistem dapat memastikan bahwa data yang digunakan dalam sistem harus terlindung dari akses yang tidak berwenang |

5. Usecase Diagram

Usecase Diagram menjelaskan interaksi setiap actor pada *system*, aktor merupakan pengguna yang ada di aplikasi yang dapat menggunakan fitur – fitur pada aplikasi yang dibangun. Pada pembuatan aplikasi pemesanan dan informasi jasa *makeup* ini memiliki dua aktor yaitu pelanggan dan perias wajah. Pelanggan dapat melakukan *login*, registrasi, memesan jasa *makeup*, memilih perias wajah, mengelola pesanan, melihat perias wajah, dan melakukan pembayaran sedangkan perias wajah bisa melakukan *login*, registrasi, mengelola data diri, mengelola pilihan *makeup* dan melihat pembayaran.



Gambar 3.17 Usecase Diagram
(Sumber: Data Olahan)

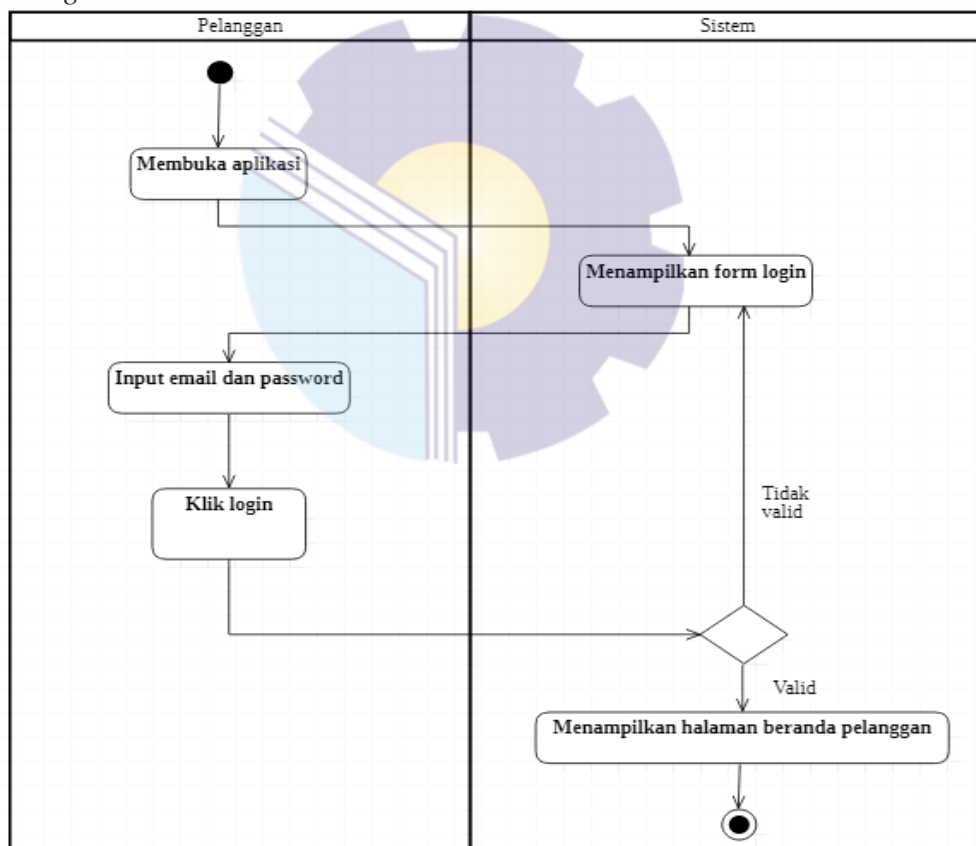
6. Activity Diagram

Activity diagram menggambarkan aliran aktifitas yang dilakukan oleh sistem pada aplikasi yang dibuat. Pada sistem ini memiliki dua aktor yaitu perias wajah dan pelanggan

a. Activity Diagram Pelanggan

1. Login

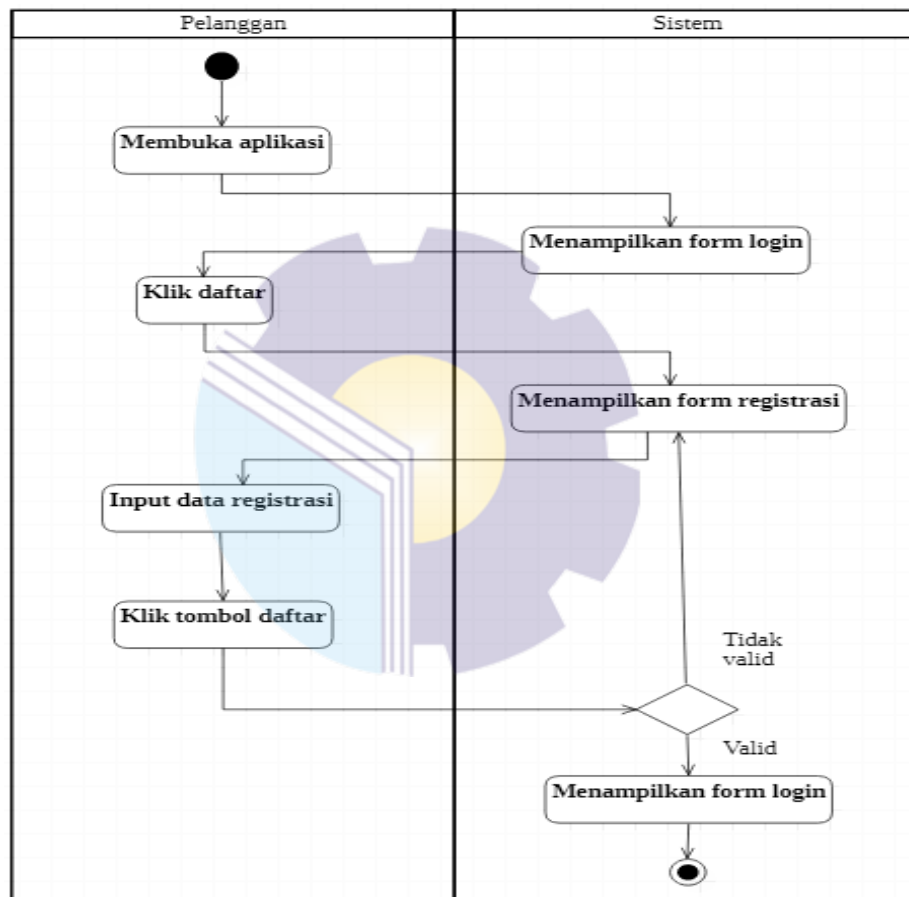
Pada Activity ini pelanggan membuka aplikasi terlebih dahulu, kemudian sistem menampilkan form login setelah form ditampilkan pelanggan memasukan *email* dan *password* jika yang diinput valid maka sistem akan menampilkan halaman beranda pelanggan, jika tidak valid maka sistem akan menampilkan lagi form login.



Gambar 3.18 Activity diagram perias wajah login
(Sumber: Data Olahan)

2. Registrasi

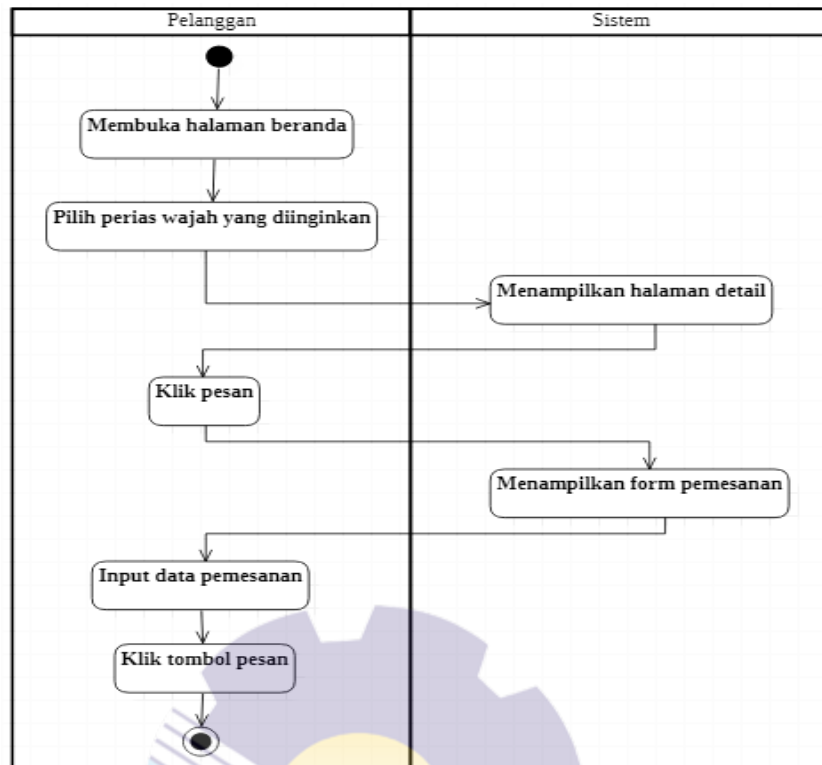
Pada *activity* ini pelanggan membuka aplikasi lalu sistem menampilkan form *login*, kemudian pelanggan klik daftar, sistem menampilkan form *login* selanjutnya pelanggan memasukan data registrasi pada form registrasi, jika data yang diinputkan valid maka sistem menampilkan form *login*, jika tidak valid maka sistem menampilkan form registrasi dan pelanggan mengulangi input data registrasi.



Gambar 3.19 *Activity* diagram perias wajah registrasi
(Sumber: Data Olahan)

3. Memesan jasa *makeup*

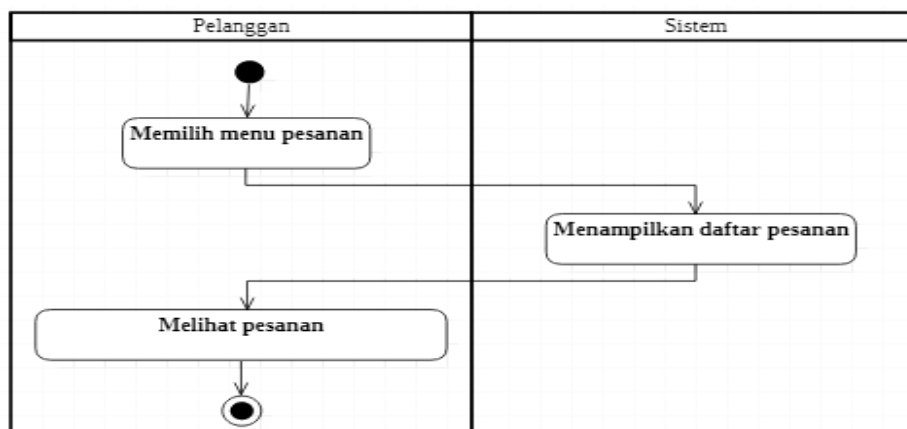
Pada *activity* ini pelanggan membuka halaman beranda lalu memilih perias wajah yang diinginkan, sistem menampilkan halaman detail kemudian pelanggan klik halaman detail, selanjutnya sistem menampilkan form pemesanan lalu pelanggan menginput data pemesan dan pemesanan berhasil dilakukan.



Gambar 3.20 Activity digram pelanggan melakukan pemesanan
(Sumber: Data Olahan)

4. Mengelola pesanan

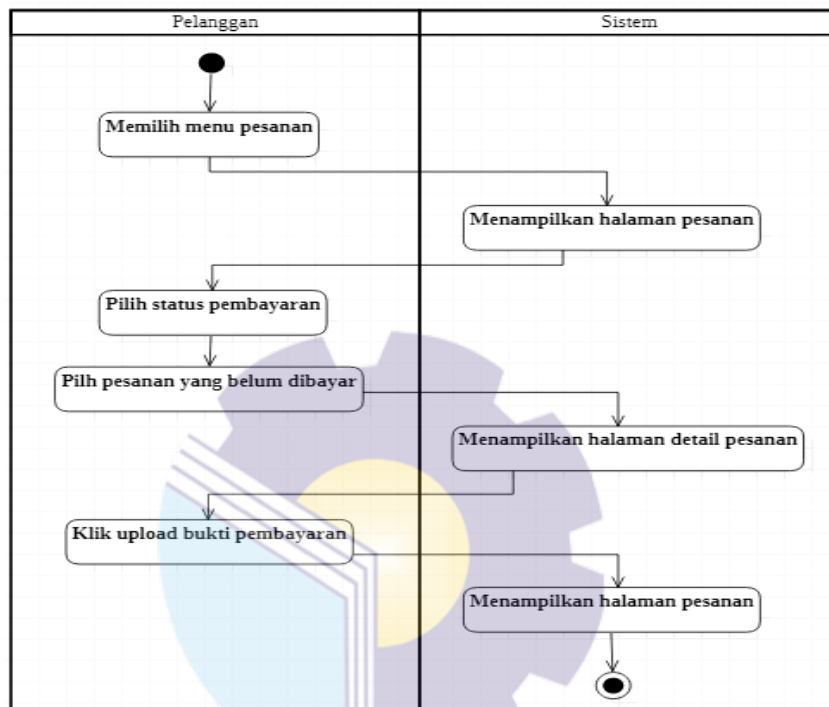
Pada *activity* ini pelanggan memilih menu pesanan pada halaman beranda, sistem menampilkan halaman daftar pesanan kemudian pelanggan bisa melihat dan membatalkan pesanan, pada daftar pesanan pelanggan bisa melihat pesanan yang belum di bayar, sudah dibayar dan pesanan selesai pada menu status pesanan.



Gambar 3.21 Activity diagram pelanggan mengelola pesanan
(Sumber: Data Olahan)

5. Melakukan pembayaran

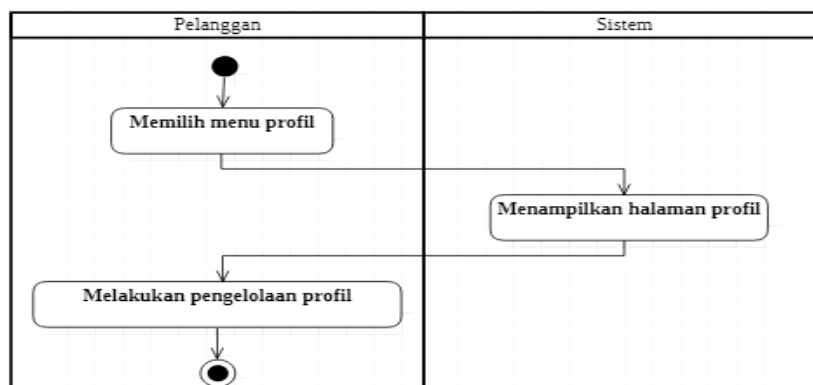
Pada *activity* ini pelanggan memilih menu pesanan pada halaman beranda, sistem menampilkan halaman daftar pesanan kemudian pelanggan memilih menu status pesanan lalu pilih pesanan yang belum dibayar, sistem menampilkan pesanan yang belum dibayar selanjutnya pelanggan upload bukti pembayaran.



Gambar 3.22 *Activity* diagram pelanggan melakukan pembayaran
(Sumber: Data Olahan)

6. Mengelola profil

Pelanggan membuka menu profil lalu sistem menampilkan halaman profil kemudian pelanggan melakukan pengelolaan profil yaitu bisa melihat dan edit profil.

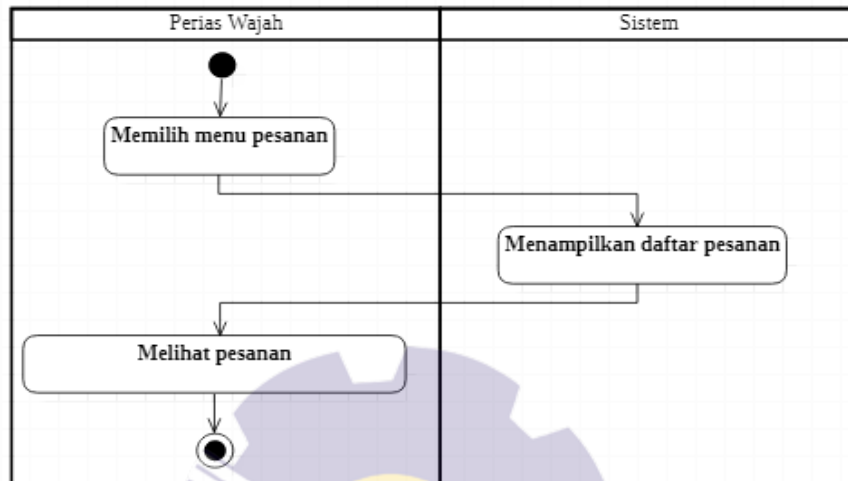


Gambar 3.23 *Activity* diagram pelanggan melakukan pengelolaan profil
(Sumber: Data Olahan)

b. Activity Diagram Perias wajah

1. Melihat pesanan

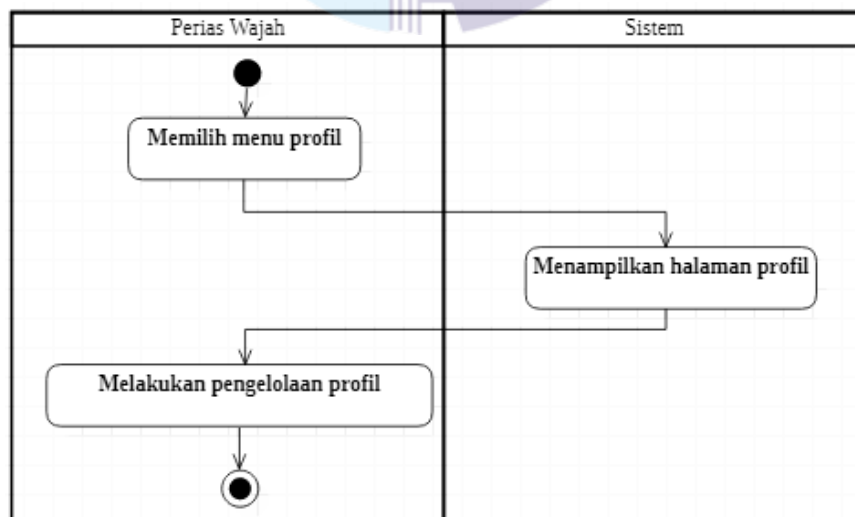
Pada *activity* ini pelanggan memilih menu pesanan pada halaman beranda, sistem menampilkan halaman daftar pesanan kemudian pelanggan melihat pesanan.



Gambar 3.24 Activity diagram perias wajah melihat pesanan
(Sumber: Data Olahan)

2. Mengelola profil

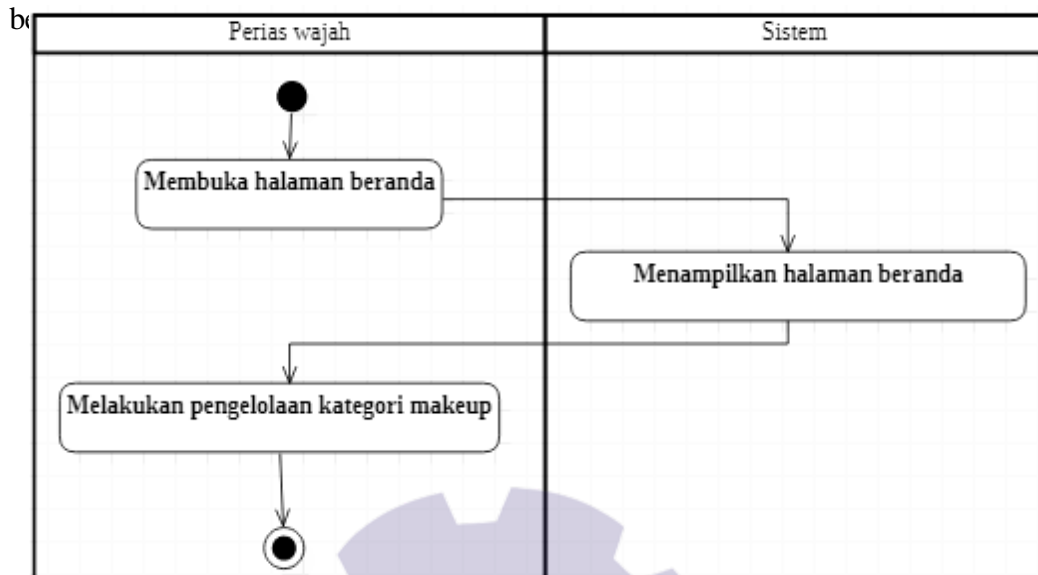
Perias wajah membuka menu profil lalu sistem menampilkan halaman profil kemudian profil melakukan pengelolaan profil yaitu bisa melihat dan edit profil.



Gambar 3.25 Activity diagram perias wajah mengelola profil
(Sumber: Data Olahan)

3. Mengelola jenis *makeup*

Perias wajah membuka halaman beranda, sistem menampilkan halaman



Gambar 3.26 *Activity diagram* perias wajah mengelola kategori *makeup*
(Sumber: Data Olahan)

3.2.2. Fase Perancangan

Pada fase perancangan sistem ini dilakukan berdasarkan dari hasil dari analisa sistem yang telah dilakukan pada tahapan sebelumnya. Dari kebutuhan fungsional yang dihasilkan maka akan dirancang rancangan antar muka. Pada perancangan ini memiliki dua aktor yaitu pelanggan dan perias wajah.

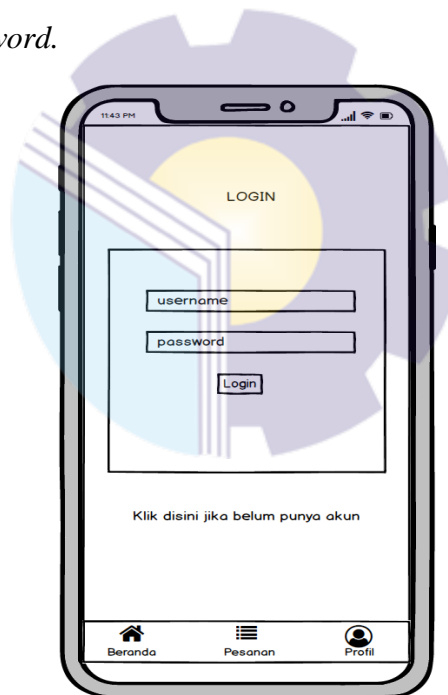
1. Perancangan Antar Muka

Rancangan antar muka pelanggan dan perias wajah memiliki tampilan menu beranda, pesanan, profil, *login* dan registrasi.

a. Pelanggan

1. *Login*

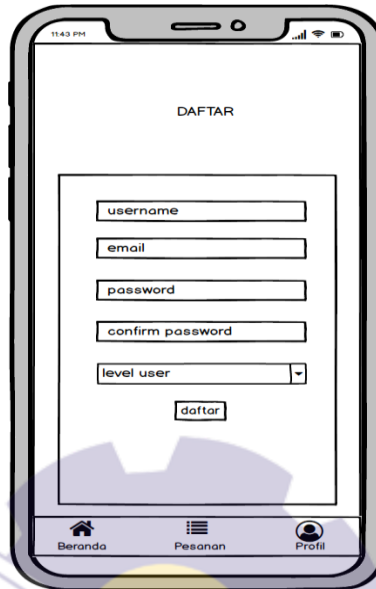
Halaman *login* ini di akses oleh pelanggan, pada *login* diminta memasukan *username* dan *password*.



Gambar 3.27 Perancangan antar muka login pelanggan
(Sumber: Data Olahan)

2. Registrasi

Pada halaman registrasi pelanggan diminta memasukan *username*, *email*, *password*, *confirm password* dan *level user*.

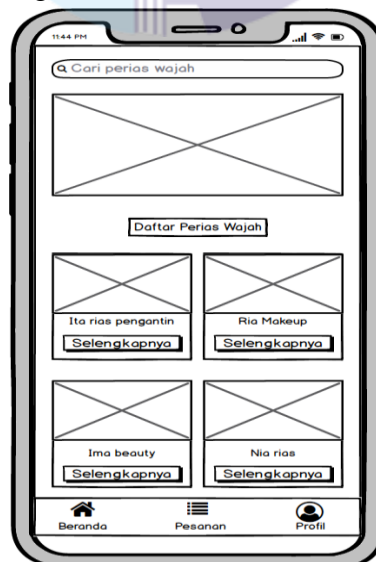


The screenshot shows a mobile application interface for registration. At the top, the title 'DAFTAR' is centered. Below it, there are five input fields: 'username', 'email', 'password', 'confirm password', and 'level user' (which is a dropdown menu). A 'daftar' button is positioned below the input fields. At the bottom of the screen, there is a navigation bar with three icons: a home icon labeled 'Beranda', a list icon labeled 'Pesanan', and a profile icon labeled 'Profil'.

Gambar 3.28 Perancangan antar muka registrasi pelanggan
(Sumber: Data Olahan)

3. Beranda

Pada tampilan beranda pelanggan bisa mencari perias wajah dan memilih perias wajah sesuai yang diinginkan.

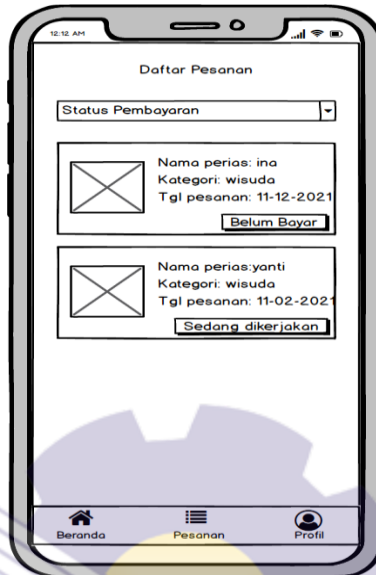


The screenshot shows a mobile application interface for finding makeup artists. At the top, there is a search bar with the placeholder text 'Cari perias wajah'. Below the search bar, there is a large empty rectangular area with a diagonal cross. Underneath this area, there is a button labeled 'Daftar Perias Wajah'. Below the button, there are four cards, each representing a makeup artist. Each card has a placeholder image with a diagonal cross, a name, and a 'Selengkapnya' button. The names of the artists are: 'Ita rias pengantin', 'Ria Makeup', 'Ima beauty', and 'Nia rias'. At the bottom of the screen, there is a navigation bar with three icons: a home icon labeled 'Beranda', a list icon labeled 'Pesanan', and a profile icon labeled 'Profil'.

Gambar 3.29 Perancangan antar muka beranda pelanggan
(Sumber: Data Olahan)

4. Pesanan

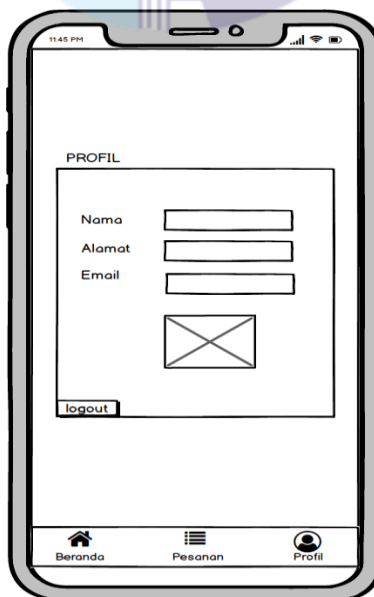
Pada tampilan pesanan pelanggan bisa melihat status pesanan yaitu pesanan belum bayar, sudah bayar, sedang dikerjakan dan pesanan yg selesai dikerjakan.



Gambar 3.29 Perancangan antar muka pesanan pelanggan
(Sumber: Data Olahan)

5. Profil

Pada halaman profil pelanggan bisa melihat profil nya dan bisa melakukan *logout*.

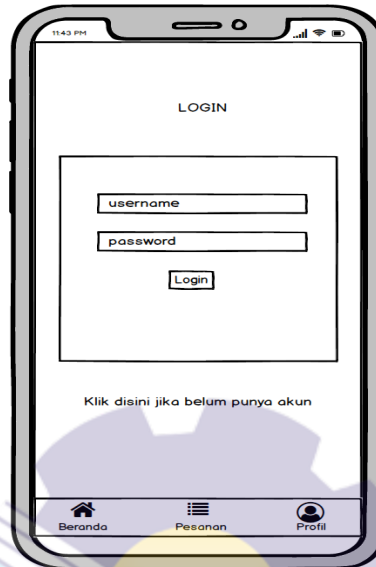


Gambar 3.30 Perancangan antar muka profil pelanggan
(Sumber: Data Olahan)

b. Perias Wajah

1. Login

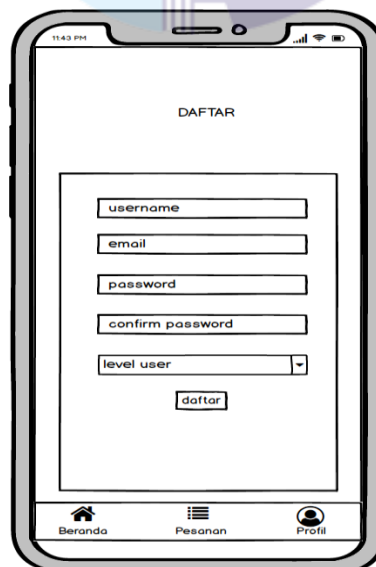
Halaman *login* ini di akses oleh perias wajah, pada *login* diminta memasukan *username* dan *password*.



Gambar 3.31 Perancangan antar muka *login* perias wajah
(Sumber: Data Olahan)

2. Registrasi

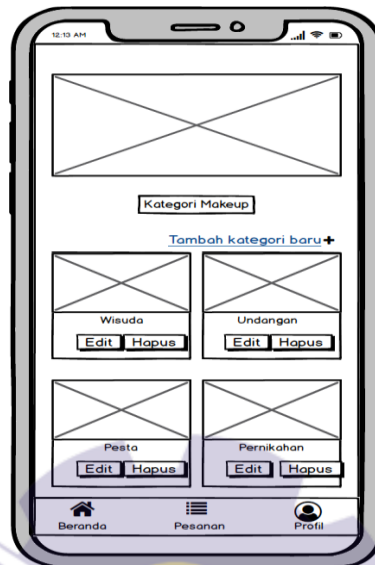
Pada halaman registrasi perias wajah diminta memasukan *username*, *email*, *password*, *confirm password* dan *level user*.



Gambar 3.32 Perancangan antar muka registrasi perias wajah
(Sumber: Data Olahan)

3. Beranda

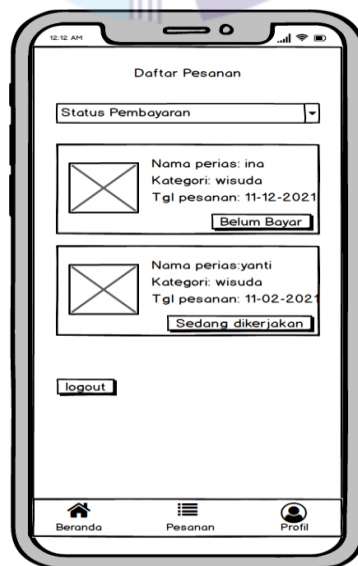
Pada tampilan beranda perias wajah bisa menambah kategori *makeup*, edit dan hapus kategori *makeup*.



Gambar 3.33 Perancangan antar muka beranda perias wajah
(Sumber: Data Olahan)

4. Pesanan

Pada tampilan pesanan perias wajah bisa melihat status pesanan yaitu pesanan belum bayar, sudah bayar, sedang dikerjakan dan pesanan yg selesai dikerjakan.



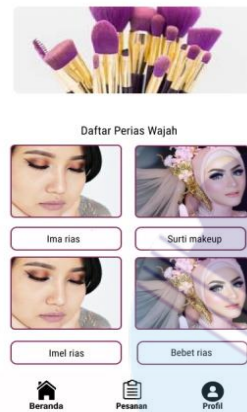
Gambar 3.34 Perancangan antar muka pesanan perias wajah
(Sumber: Data Olahan)

2. Proses iterasi

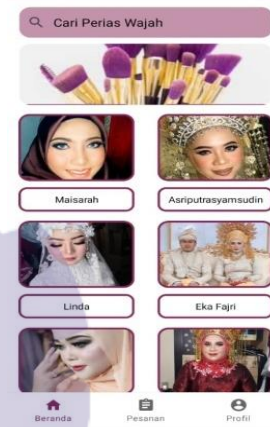
Dalam perancangan aplikasi pemesanan dan informasi *jasamakeup* ini, peneliti melakukan beberapa tahapan iterasi pada perancangan, agar menghasilkan aplikasi pemesanan dan informasi jasa *makeup* kebutuhan pengguna, perbaikan yang dilakukan adalah sebagai berikut :

a. Menambahkan fitur pencarian

Pada iterasi ini pelanggan meminta untuk menambahkan fitur pencarian agar memudahkan pelanggan dalam mencari perias wajah.



Gambar 3.35 Tampilan beranda sebelum melakukan iterasi



Gambar 3.36 Tampilan beranda sesudah melakukan iterasi

b. Menambahkan fitur pembayaran pada menu pesanan

Pada fitur pembayaran bisa melihat status pembayaran yaitu belum dibayar, menunggu, sudah dibayar dan pesanan selesai.



Gambar 3.38 Tampilan menu pesanan sebelum melakukan iterasi



Gambar 3.39 Tampilan menu pesanan setelah melakukan iterasi

3.2.3. Fase Konstruksi

Fase ketiga adalah fase konstruksi, di mana pada fase ini desain yang sudah dibuat maka dikembangkan menjadi sebuah program. Dimana program yang sudah dibuat ini dipresentasikan dalam bentuk menu-menu. Tampilan menu-menu yang penulis buat adalah menu beranda, pesanan, detail dan daftar pesanan.

1. Login

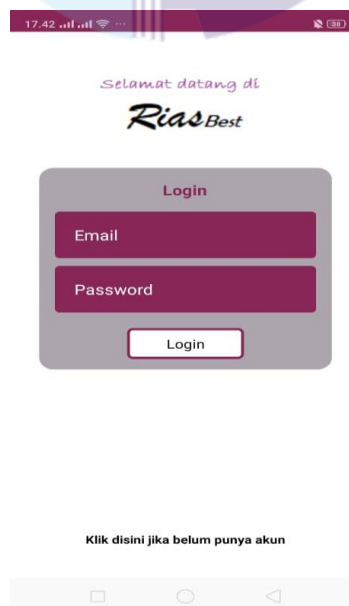
Login di bawah ini merupakan langkah awal untuk masuk ke dalam aplikasi pemesanan dan informasi jasa *makeup* yang bisa diakses oleh pelanggan dan perias wajah, pada form *login* menginputkan *username* dan *password* berikut potongan coding dan tampilan halaman *login*.

a. Coding

```
private void loginUser() {  
    String email = binding.email.getText().toString().trim();  
    String password = binding.password.getText().toString().trim();  
  
    if(email.isEmpty()) {  
        Toast.makeText(context, LoginActivity.this, text: "Email tidak boleh kosong!", Toast.LENGTH_SHORT).show();  
        return;  
    } else if (password.isEmpty()){  
        Toast.makeText(context, LoginActivity.this, text: "Password tidak boleh kosong!", Toast.LENGTH_SHORT).show();  
        return;  
    }  
}
```

Gambar 3.37 Potongan coding login
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.38 Tampilan login
(Sumber: Data Olahan)

2. Registrasi

Melakukan pendaftaran di aplikasi jika belum punya akun untuk login, dilakukan oleh aktor pengguna dan perias wajah, pada form *login* menginputkan *name*, *email*, *username*, *password*, *confirm password*, *level user* dan kemudian melakukan verifikasi melalui *email*. Berikut potongan coding dan tampilan halaman registrasi.

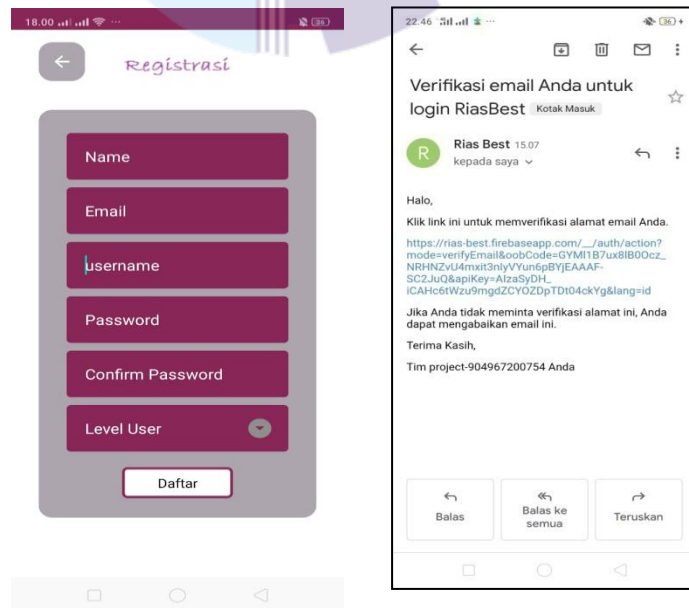
a. Coding

```
private void getUserInput() {
    String name = binding.nameEt.getText().toString().trim();
    String email = binding.emailEt.getText().toString().trim();
    String password = binding.passwordEt.getText().toString().trim();
    String username = binding.usernameEt.getText().toString().trim();
    String cfPassword = binding.confirmPassword.getText().toString().trim();

    if(name.isEmpty()) {
        Toast.makeText(context: RegisterActivity.this, text: "Nama tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if(email.isEmpty()) {
        Toast.makeText(context: RegisterActivity.this, text: "Email tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if(!email.contains("@") || !email.contains(".")) {
        Toast.makeText(context: RegisterActivity.this, text: "Format email tidak sesuai!", Toast.LENGTH_SHORT).show();
        return;
    } else if(password.isEmpty() || cfPassword.isEmpty()) {
        Toast.makeText(context: RegisterActivity.this, text: "Password tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if(!password.equals(cfPassword)) {
        Toast.makeText(context: RegisterActivity.this, text: "Password & Konfirmasi Password tidak sama!", Toast.LENGTH_SHORT).show();
        return;
    } else if(username.isEmpty()) {
        Toast.makeText(context: RegisterActivity.this, text: "Username tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if(levelUser == null) {
        Toast.makeText(context: RegisterActivity.this, text: "level user tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    }
}
```

Gambar 3.39 Coding registrasi
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.40 Tampilan registrasi dan verifikasi *email*
(Sumber: Data Olahan)

3. Beranda

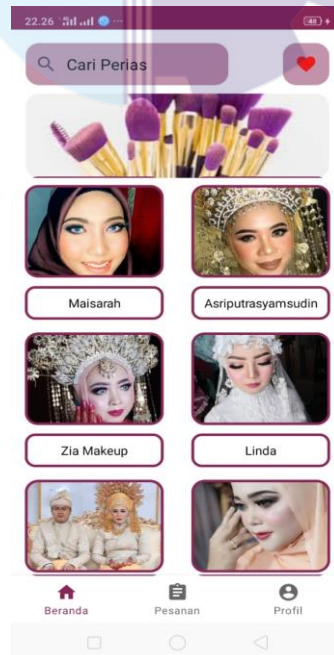
Beranda di bawah ini merupakan tampilan awal dari aplikasi pemesanan dan informasi jasa *makeup*, pada tampilan ini bisa memilih perias wajah yang diinginkan dan bisa menjadikan perias sebagai perias *favorite*. Berikut potongan *coding* dan tampilan halaman beranda.

a. Coding

```
public class HomepageActivity extends AppCompatActivity {  
    @SuppressWarnings("NonConstantResourceId")  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        com.riiasbest.riiasbest.databinding.ActivityHomepageBinding binding = ActivityHomepageBinding.inflate(  
            setContentView(binding.getRoot());  
  
        BottomNavigationView navView = findViewById(R.id.nav_view);  
        // Passing each menu ID as a set of Ids because each  
        // menu should be considered as top level destinations.  
        navView.setOnNavigationItemSelectedListener(item -> {  
            Fragment selectedFragment = new BerandaFragment();  
            switch (item.getItemId()) {  
                case R.id.navigation_beranda: {  
                    navView.getMenu().findItem(R.id.navigation_beranda).setEnabled(false);  
                    navView.getMenu().findItem(R.id.navigation_pesanan).setEnabled(true);  
                    navView.getMenu().findItem(R.id.navigation_profile).setEnabled(true);  
                    selectedFragment = new BerandaFragment();  
                    break;  
            }  
        }  
    }  
}
```

Gambar 3.41 Potongan *coding* halaman beranda
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.42 Tampilan beranda
(Sumber: Data Olahan)

4. Melihat kategori *makeup*

Pada tampilan ini pelanggan bisa melihat kategori *makeup* yang disediakan perias wajah dan bisa melakukan pemesanan, tampilan ini hanya bisa diakses oleh pelanggan. Berikut potongan coding dan tampilan halaman beranda.

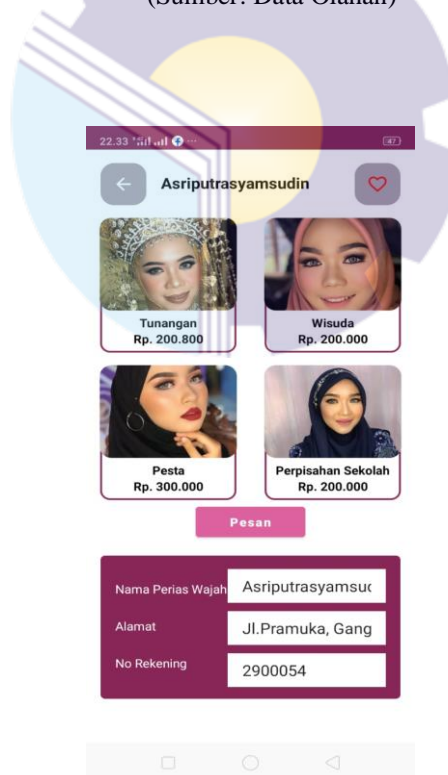
a. Coding

```
private void formValidation() {
    String name = binding.name.getText().toString().trim();
    String price = binding.price.getText().toString().trim();

    if(name.isEmpty()) {
        Toast.makeText(context: PemesananActivity.this, text: "Nama Lengkap tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if (price.isEmpty()) {
        Toast.makeText(context: PemesananActivity.this, text: "Nama Lengkap tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if (dateTime == null) {
        Toast.makeText(context: PemesananActivity.this, text: "Tanggal pemesanan tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if (category == null) {
        Toast.makeText(context: PemesananActivity.this, text: "Kategori tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    }
}
```

Gambar 3.43 Coding melihat kategori *makeup*
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.44 Tampilan melihat kategori *makeup*
(Sumber: Data Olahan)

5. Melakukan pemesanan

Pada tampilan ini pelanggan melakukan pemesanan *makeup*, lalu mengisi form pemesanan, setelah pesanan di buat maka akan muncul pemberitahuan pesanan berhasil dilakukan. Tampilan ini hanya bisa diakses oleh pelanggan. Berikut potongan *coding* dan tampilan halaman melakukan pemesanan

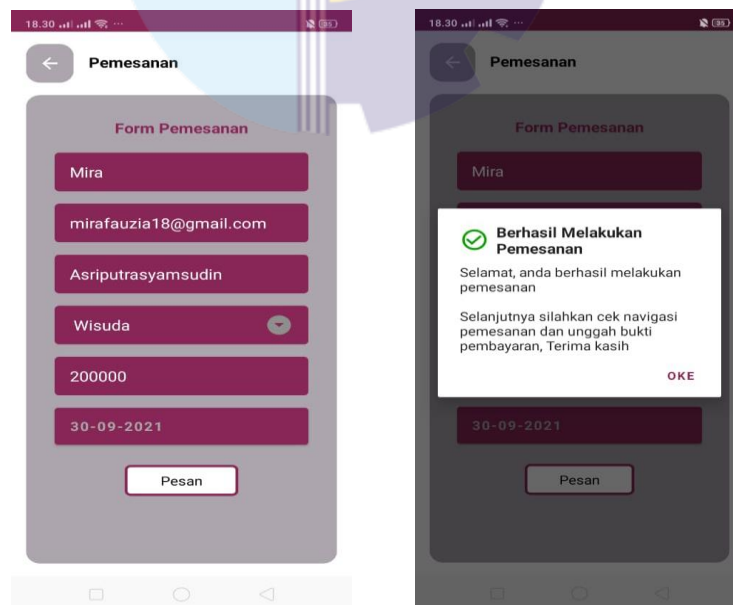
a. Coding

```
private void formValidation() {
    String name = binding.name.getText().toString().trim();
    String price = binding.price.getText().toString().trim();

    if(name.isEmpty()) {
        Toast.makeText(context: PemesananActivity.this, text: "Nama Lengkap tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if (price.isEmpty()) {
        Toast.makeText(context: PemesananActivity.this, text: "Nama Lengkap tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if (dateTime == null) {
        Toast.makeText(context: PemesananActivity.this, text: "Tanggal pemesanan tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    } else if (category == null) {
        Toast.makeText(context: PemesananActivity.this, text: "Kategori tidak boleh kosong!", Toast.LENGTH_SHORT).show();
        return;
    }
}
```

Gambar 3.48 Coding melakukan pemesanan
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.49 Tampilan melakukan pemesanan
(Sumber: Data Olahan)

6. Profil

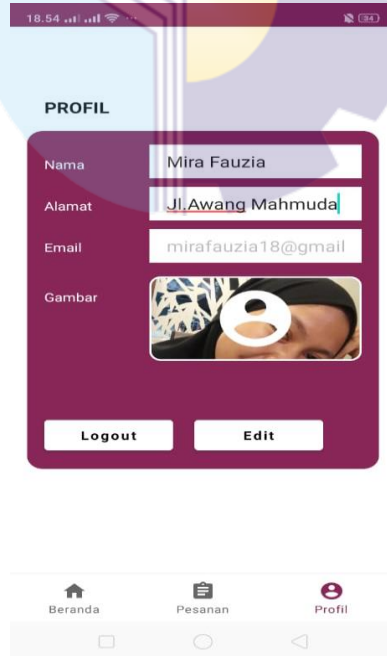
Tampilan ini pelanggan merupakan tampilan profil, profil bisa di edit dan dari halaman ini bisa melakukan *logout*. Berikut potongan coding dan tampilan halaman profil.

a. Coding

```
public class ProfileFragment extends Fragment {  
  
    private FragmentProfileBinding binding;  
    private FirebaseUser user;  
  
    // variabel  
    private static final int REQUEST_FROM_GALLERY_TO_SELF_PHOTO = 1001;  
  
    public View onCreateView(@NonNull LayoutInflater inflater,  
        ViewGroup container, Bundle savedInstanceState) {  
  
        binding = FragmentProfileBinding.inflate(inflater, container, attachToParent: false);  
        user = FirebaseAuth.getInstance().getCurrentUser();  
  
        // check role  
        checkRole();  
  
        // AMBIL SEMUA DATA DARI DATABASE DAN TAMPILKAN  
        populateUI();  
  
        return binding.getRoot();  
    }  
}
```

Gambar 3.45 Coding tampilan profil
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.46 Tampilan profil
(Sumber: Data Olahan)

7. Mengelola kategori *makeup*

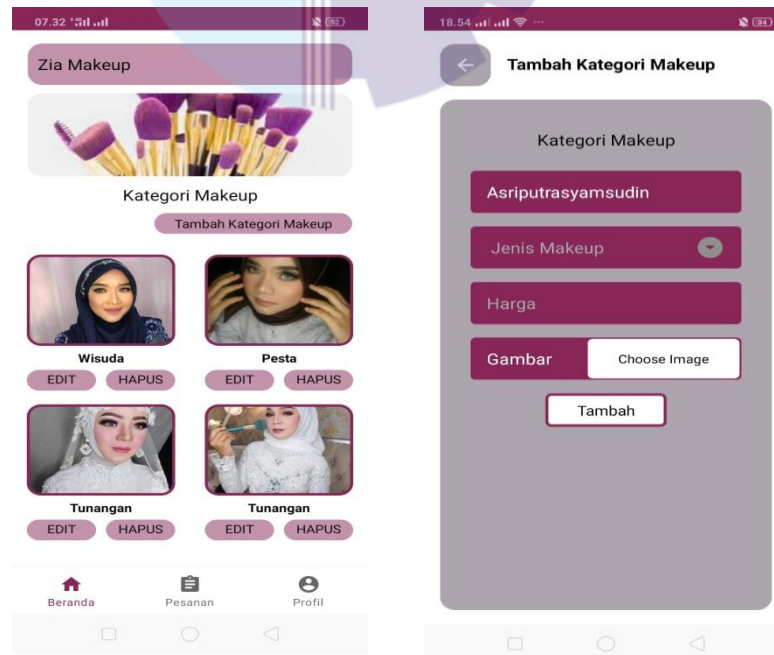
Pada tampilan ini perias wajah pengelolaan kategori *makeup*, yaitu bisa melihat, menambah, edit dan hapus. Tampilan ini hanya bisa diakses oleh perias wajah. Berikut potongan coding dan tampilan mengelola kategori *makeup*.

a. Coding

```
public class PeriasAddCategoryActivity extends AppCompatActivity {  
  
    public static final String EXTRA_NAME = "name";  
    private ActivityPeriasAddCategoryBinding binding;  
    private String category;  
    private String image;  
    private static final int REQUEST_FROM_GALLERY_TO_SELF_PHOTO = 1001;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        binding = ActivityPeriasAddCategoryBinding.inflate(getLayoutInflater());  
        setContentView(binding.getRoot());  
  
        // set nama perias  
        binding.name.setText(getIntent().getStringExtra(EXTRA_NAME));  
  
        // kembali ke halaman sebelumnya  
        binding.backButton.setOnClickListener(view -> onBackPressed());  
  
        // submit data  
        binding.view2.setOnClickListener(view -> addCategory());  
  
        // ambil gambar dari galeri  
        binding.view6.setOnClickListener(view -> pickImage());  
    }  
}
```

Gambar 3.52 Coding mengelola kategori *makeup*
(Sumber: Data Olahan)

b. Tampilan



Gambar 3.473 Tampilan mengelola kategori *makeup*
(Sumber: Data Olahan)

3.2.4. Fase Pelaksanaan

Fase yang terakhir adalah fase pelaksanaan, dimana pada fase ini dilakukan pengujian sistem yang sudah dibuat. Pengujian ini menggunakan *blackbox testing* dan *Compatibility Testing* untuk mengetahui apakah aplikasi yang dibuat dapat berjalan di versi di beberapa versi android. *Blackbox testing* merupakan jenis testing yang memperlakukan perangkat lunak yang tidak mengetahui kinerja internalnya sehingga penguji hanya memandang perangkat lunak seperti sebuah kotak hitam yang tidak penting tanpa melihat isinya, tapi hanya menguji di bagian luar (Rizky, 2011). Pada tahapan pengujian ini menggunakan *handphone* android untuk menjalankan aplikasi yang akan diuji, untuk penjelasan lebih lengkap akan dijelaskan pada bab 4 bagian pengujian.



BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

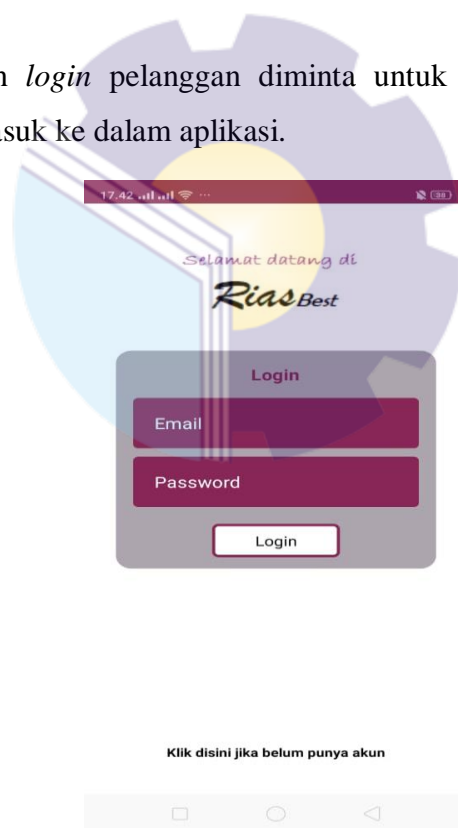
Hasil dari pembuatan aplikasi informasi dan pemesanan jasa *makeup* menggunakan metode *Rapid application development* yang dilakukan sesuai dengan tahapan *Rapid application development* adalah sebagai berikut :

a. Pelanggan

Pelanggan merupakan aktor yang memesan jasa *makeup* dan bisa melakukan aktifitas di aplikasi pemesanan dan informasi jasa *makeup* antara lain:

1. *Login*

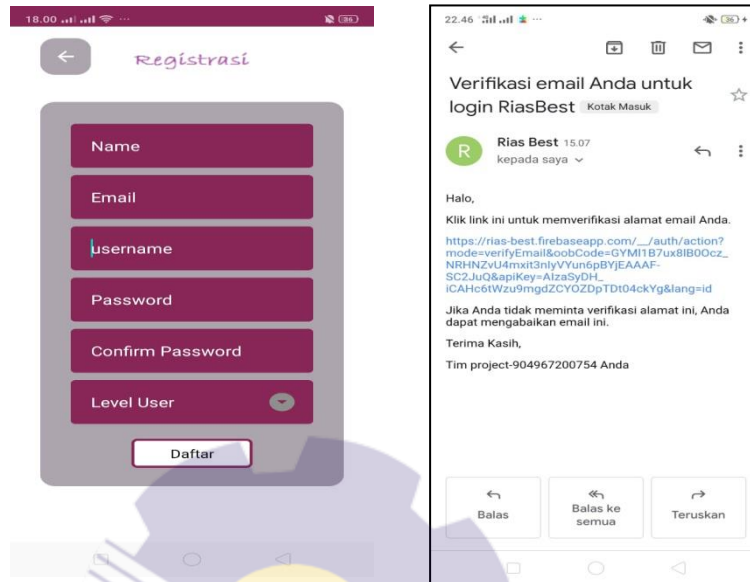
Pada halaman *login* pelanggan diminta untuk menginputkan *email* dan *password* untuk masuk ke dalam aplikasi.



Gambar 4.1 Tampilan login pelanggan
(Sumber: Data Olahan)

2. Registrasi

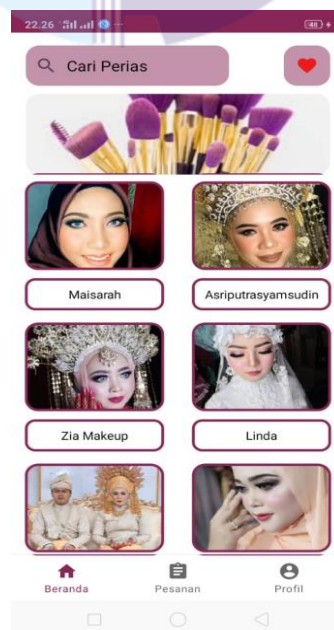
Pada halaman ini diminta menginputkan *name*, *email*, *username*, *password*, *confirm password* dan *level user*, kemudian melakukan verifikasi melalui *email*.



Gambar 4.2 Tampilan registrasi pelanggan dan verifikasi email
(Sumber: Data Olahan)

3. Beranda

Pada halaman beranda pelanggan mencari dan memilih perias wajah yang diinginkan, pada halaman ini memiliki menu pesanan dan profil dan fitur *favorite*.



Gambar 4.3 Tampilan beranda pelanggan
(Sumber: Data Olahan)

4. Halaman detail perias wajah

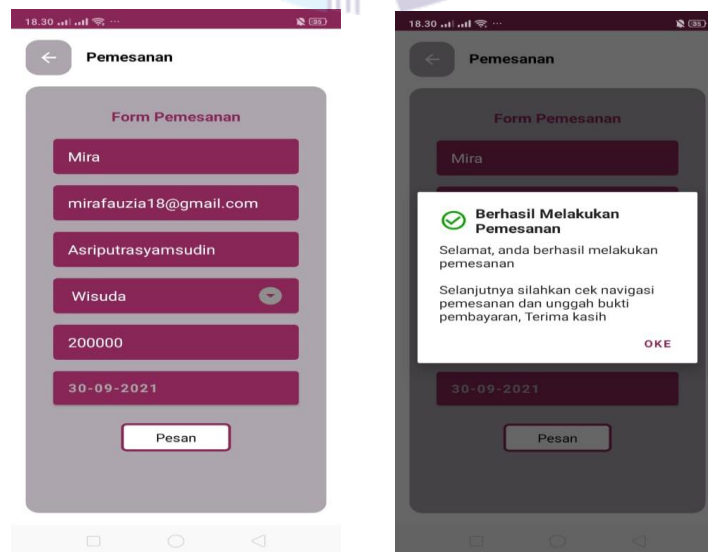
Pada halaman ini menampilkan halaman detail dari perias wajah yang dipilih, pada halaman ini menampilkan beberapa kategori *makeup*, info perias dan bisa melakukan pesanan.



Gambar 4.4 Tampilan detail perias wajah
(Sumber: Data Olahan)

5. Halaman pemesanan

Pada halaman ini menampilkan form pemesanan, setelah pesanan di buat lalu muncul pemberitahuan pemesanan berhasil dilakukan.



Gambar 4.5 Tampilan halaman pemesanan
(Sumber: Data Olahan)

6. Halaman daftar pesanan

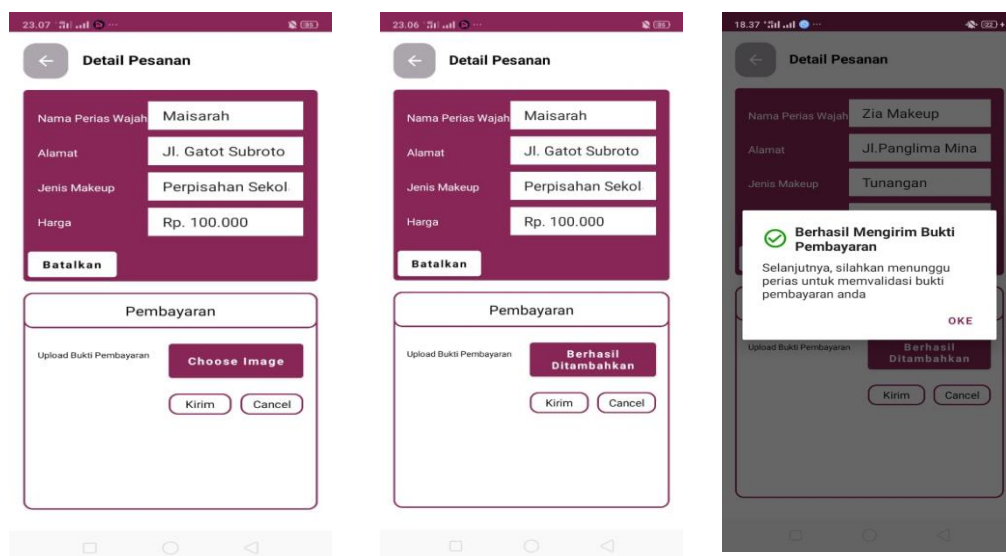
Pada halaman ini menampilkan daftar pesanan yang dibuat, dan bisa menampilkan status pembayaran yaitu daftar pesanan yang belum dibayar, sudah dibayar dan selesai.



Gambar 4.6 Tampilan daftar pesanan
(Sumber: Data Olahan)

7. Halaman pembayaran

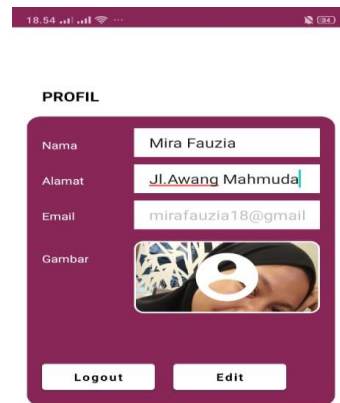
Pada halaman ini diminta mengunggah bukti pembayaran, setelah berhasil diupload maka akan muncul pemberitahuan pembayaran berhasil dilakukan namun dalam tahap menunggu di validasi oleh perias.



Gambar 4.7 Halaman pembayaran
(Sumber: Data Olahan)

8. Halaman profil

Halaman ini menampilkan profil pelanggan, pelanggan bisa edit profil dan *logout*.



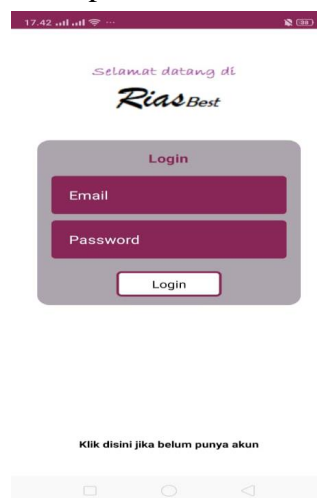
Gambar 4.8 Tampilan profil
(Sumber: Data Olahan)

b. Perias wajah

Perias wajah merupakan aktor yang menyediakan jasa untuk di pesan oleh pelanggan, dan perias wajah yang menginputkan kategori *makeup*.

1. Login

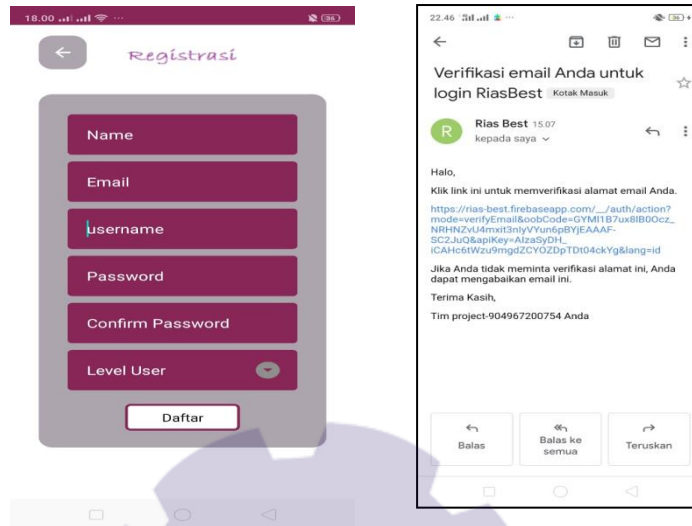
Pada halaman *login* perias wajah diminta untuk menginputkan *email* dan *password* untuk masuk ke dalam aplikasi.



Gambar 4.9 Tampilan login perias wajah
(Sumber: Data Olahan)

2. Registrasi

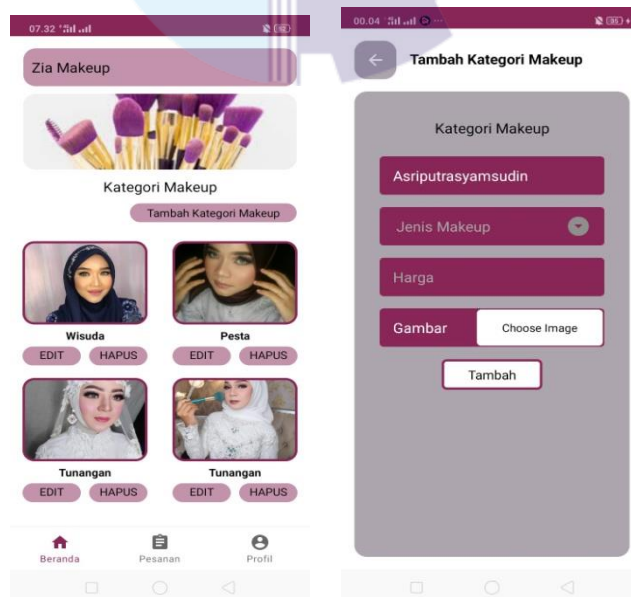
Pada halaman ini diminta menginputkan *name*, *email*, *username*, *password*, *confirm password* dan level *user*.



Gambar 4.10 Tampilan registrasi perias wajah
(Sumber: Data Olahan)

3. Beranda

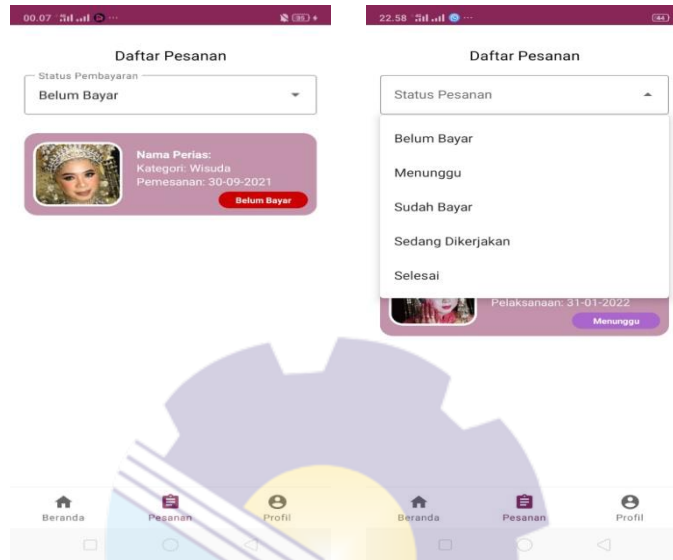
Pada halaman beranda perias wajah bisa mengelola kategori *makeup* yaitu menambah, edit dan hapus, pada halaman ini memiliki menu pesanan dan profil.



Gambar 4.11 Tampilan halaman beranda perias wajah
(Sumber: Data Olahan)

4. Halaman daftar pesanan

Pada halaman ini menampilkan daftar pesanan yang dibuat oleh pelanggan dan bisa dilihat perias wajah dan perias wajah bisa melihat status pembayaran yaitu daftar pesanan yang belum dibayar, sudah dibayar, menunggu, sedang dikerjakan dan selesai.



Gambar 4.12 Tampilan daftar pesanan perias wajah (Sumber: Data Olahan)

5. Halaman profil

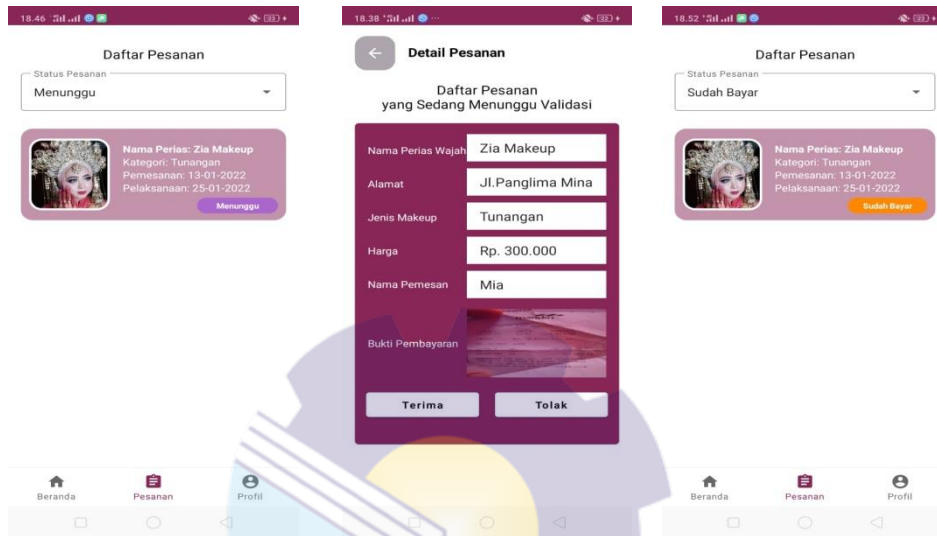
Halaman ini menampilkan profil pelias wajah, perias bisa melakukan edit profil dan *logout*.



Gambar 4.13 Halaman profil perias wajah (Sumber: Data Olahan)

6. Halaman validasi pembayaran

Pada halaman ini perias melakukan validasi terhadap pesanan yang dibuat pelanggan yang sedang di tahap menunggu validasi dari perias, apakah sudah memasukan bukti pembayaran yang benar atau belum, dan setelah perias memvalidasi pembayaran maka pesanan berubah status menjadi sudah dibayar.



Gambar 4.14 Halaman validasi pembayaran
(Sumber: Data Olahan)

4.2 Pembahasan

Pada pembahasan ini melakukan pengujian pada aplikasi pemesanan dan informasi jasa *makeup* menggunakan metode *rapid application development*, pengujian bertujuan agar aplikasi yang dibuat sudah minim dengan *bug*.

4.2.1. Prosedur Pengujian

Pada pengujian ini peneliti melakukan pengujian dengan teknik pengujian *black box testing*, dengan cara melakukan pengujian langsung ke pelanggan dan perias wajah untuk menjalankan fungsi-fungsi di aplikasi.

a. *Black box testing*

Berikut prosedur pengujian *black box testing* yang dilakukan:

1. Menyiapkan *handphone* dengan versi yang berbeda-beda
2. Menginstal aplikasi
3. Menyusun *test case* untuk proses pengujian
4. Melakukan pengujian di setiap form inputan dengan berbagai *test cast* yang dibuat
5. Mengamati output dari hasil inputan dan menulis hasil dari pengamatan.

b. *Compatibility Testing*

Berikut prosedur pengujian *Compatibility Testing* yang dilakukan:

1. Menyiapkan *handphone* dengan versi yang berbeda-beda
2. Menginstal aplikasi
3. Mengamati *output* dari hasil inputan dan menulis hasil dari pengamatan.

4.2.2. Hasil Pengujian

a. *Black box testing*

Blackbox testing merupakan jenis *testing* yang memperlakukan perangkat lunak yang tidak mengetahui kinerja internalnya sehingga penguji hanya memandang perangkat lunak seperti sebuah kotak hitam yang tidak penting tanpa melihat isinya, tapi hanya menguji di bagian luar (Rizky, 2011).

1. *Login*

Melakukan pengujian pada form *login* dengan menggunakan 3 *handphone* yang berbeda.

Tabel 4.1 Pengujian form *login*

| Spesifikasi <i>handphone</i> : <i>Eight core</i> ROM 32GB RAM 2GB | | | |
|---|--|---|------------|
| Sistem operasi : <i>Android 9.0 Pie</i> | | | |
| Aksi | Yang Diharapkan | Hasil yang Didapat | Kesimpulan |
| Memasukkan inputan yang valid (<i>email</i> dan <i>password</i>) | Aplikasi memberikan akses kepada perias atau pelanggan untuk masuk ke dalam aplikasi | Berhasil masuk ke dalam aplikasi, aplikasi menampilkan halaman beranda | Berhasil |
| Mengosongk an <i>email</i> dan <i>password</i> | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “ <i>email</i> dan <i>password</i> tidak boleh kosong” | Berhasil |
| Memasukkan inputan format <i>email</i> yang salah pada <i>filed email</i> | Menampilkan pesan kesalahan | Aplikasi menampilkan pesan “silahkan cek data kembali” | Berhasil |

| Spesifikasi <i>handphone</i> : <i>Eight core</i> ROM 64GB RAM 4GB | | | |
|--|--|--|------------|
| Sistem operasi : <i>Android 8.1.0 Oreo</i> | | | |
| Aksi | Yang Diharapkan | Hasil yang Didapat | Kesimpulan |
| Memasukkan inputan yang valid (<i>email</i> dan <i>password</i>) | Aplikasi memberikan akses kepada perias atau pelanggan untuk masuk ke dalam aplikasi | Berhasil masuk ke dalam aplikasi, aplikasi menampilkan halaman beranda | Berhasil |
| Mengosongk an <i>email</i> dan | Menampilkan pesan kesalahan pada | Aplikasi menampilkan pesan | Berhasil |

| | | | |
|---|-----------------------------|--|----------|
| <i>password</i> | inputan | “ <i>email</i> dan <i>password</i> tidak boleh kosong” | |
| Memasukkan inputan format <i>email</i> yang salah pada <i>field email</i> | Menampilkan pesan kesalahan | Aplikasi menampilkan pesan “silahkan cek data kembali” | Berhasil |

2. Form registrasi

Melakukan pengujian pada form registrasi, dengan menggunakan 3 *handphone* yang berbeda.

Tabel 4.2 Pengujian form registrasi

| Spesifikasi <i>handphone</i> : <i>Eight core</i> ROM 32GB RAM 2GB Sistem operasi : <i>Android 9.0 Pie</i> | | | |
|---|--|--|------------|
| Aksi | Yang Diharapkan | Hasil yang Didapat | Kesimpulan |
| Memasukkan inputan yang valid (<i>name, email, username, password, conrim password</i> dan level <i>user</i>) | Menampilkan pemberitahuan bahwa registrasi berhasil dan menampilkan halaman <i>login</i> | Aplikasi menampilkan “berhasil melakukan registrasi” kemudian menampilkan halaman <i>login</i> | Berhasil |
| Mengosongkan salah satu <i>field</i> yaitu <i>field name</i> | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “ <i>name</i> tidak boleh kosong” | Berhasil |
| Memasukkan inputan format <i>email</i> yang | Menampilkan pesan kesalahan | Aplikasi menampilkan pesan “format <i>email</i> tidak | Berhasil |

| | | | |
|--|-----------------------------|--|----------|
| salah pada <i>filed email</i> | | sesuai” | |
| Memasukkan <i>password</i> dan <i>confirm password</i> tidak menggunakan <i>password</i> yang sama | Menampilkan pesan kesalahan | Aplikasi menampilkan pesan “ <i>password</i> dan <i>confirm password</i> tidak sama” | Berhasil |

| Spesifikasi <i>handphone</i> : <i>Eight core</i> ROM 64GB RAM 4GB | | | |
|---|--|--|------------|
| Sistem operasi : Android 8.1.0 Oreo | | | |
| Aksi | Yang Diharapkan | Hasil yang Didapat | Kesimpulan |
| Memasukkan inputan yang valid (<i>name, email, username, password, conrim password</i> dan level <i>user</i>) | Menampilkan pemberitahuan bahwa registrasi berhasil dan menampilkan halaman <i>login</i> | Aplikasi menampilkan “berhasil melakukan registrasi” kemudian menampilkan halaman <i>login</i> | Berhasil |
| Mengosongkan salah satu <i>field</i> yaitu <i>field name</i> | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “ <i>name</i> tidak boleh kosong” | Berhasil |
| Memasukkan inputan format <i>email</i> yang salah pada <i>filed</i> | Menampilkan pesan kesalahan | Aplikasi menampilkan pesan “format <i>email</i> tidak sesuai” | Berhasil |

| | | | |
|--|-----------------------------|--|----------|
| <i>email</i> | | | |
| Memasukkan <i>password</i> dan <i>confirm password</i> tidak menggunakan <i>password</i> yang sama | Menampilkan pesan kesalahan | Aplikasi menampilkan pesan “ <i>password</i> dan <i>confirm password</i> tidak sama” | Berhasil |

3. Form kategori *makeup*

Melakukan pengujian pada form kategori *makeup* dengan menggunakan 3 *handphone* yang berbeda.

Tabel 4.3 Pengujian form kategori *makeup*

| Spesifikasi <i>handphone</i> : <i>Eight core</i> ROM 32GB RAM 2GB | | | |
|---|--|--|------------|
| Sistem operasi : <i>Android 9.0 Pie</i> | | | |
| Aksi | Yang Diharapkan | Hasil yang Didapat | Kesimpulan |
| Memasukkan inputan yang valid (nama, jenis <i>makeup</i> , harga, gambar) | Menampilkan pemberitahuan bahwa data kategori <i>makeup</i> berhasil ditambahkan | Aplikasi menampilkan “berhasil menambah kategori <i>makeup</i> ” | Berhasil |
| Mengosongkan salah satu <i>field</i> yaitu <i>field</i> harga | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “ <i>harga</i> tidak boleh kosong” | Berhasil |
| Mengosongkan salah satu <i>field</i> upload gambar | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “gambar tidak boleh kosong” | Berhasil |

| Spesifikasi <i>handphone</i> : <i>Eight core</i> ROM 64GB RAM 4GB | | | |
|---|--|--|------------|
| Sistem operasi : Android 8.1.0 Oreo | | | |
| Aksi | Yang Diharapkan | Hasil yang Didapat | Kesimpulan |
| Memasukkan inputan yang valid (nama, jenis <i>makeup</i> , harga, gambar) | Menampilkan pemberitahuan bahwa data kategori <i>makeup</i> berhasil ditambahkan | Aplikasi menampilkan “berhasil menambah kategori <i>makeup</i> ” | Berhasil |
| Mengosongkan salah satu <i>field</i> yaitu <i>field</i> harga | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “ <i>harga</i> tidak boleh kosong” | Berhasil |
| Mengosongkan salah satu <i>field</i> <i>upload</i> gambar | Menampilkan pesan kesalahan pada inputan | Aplikasi menampilkan pesan “gambar tidak boleh kosong” | Berhasil |

b. *Compatibility Testing*

Compatibility Testing untuk mengetahui apakah aplikasi yang dibuat dapat berjalan di versi di beberapa versi *android*.

Tabel 4.4 *Compatibility testing*

| No | Alat uji | Spesifikasi alat | Hasil pengujian |
|----|--|--|---|
| 1. | Menggunakan <i>handphone</i> android Realme C2 | Versi Android 9.0 Pie, RAM 2GB, resolusi layar HD+ 1560 x 720 pixel. | Aplikasi berhasil di <i>install</i> , semua halaman <i>responsive</i> , dan semua fungsi berjalan dengan baik |
| 2. | Menggunakan <i>handphone</i> android OPPO A1k | Versi Android 9.0 Pie, RAM 2GB, resolusi layar HD+ 1560 x 720 pixel. | Aplikasi berhasil di <i>install</i> , semua halaman <i>responsive</i> , dan semua fungsi berjalan dengan baik |
| 3. | Menggunakan | Versi Android 8.1.0 | Aplikasi berhasil di <i>install</i> , |

| | | | |
|----|--|--|--|
| | <i>handphone</i> android OPPO A7 | Oreo, RAM 4GB, resolusi layar 1520 x 720 pixel | semua halaman <i>responsive</i> , dan semua fungsi berjalan dengan baik |
| 4. | Menggunakan <i>handphone</i> android Realme C11 | Versi Android 10, RAM 3 GB, resolusi layar 1560 x 720 pixel | Aplikasi berhasil di <i>install</i> , semua halaman <i>responsive</i> , dan semua fungsi berjalan dengan baik |
| 5. | Menggunakan <i>handphone</i> android VIVO V7 | Versi Android 7.0 nougat, RAM 4 GB, resolusi layar 1440 x 720 pixel | Aplikasi berhasil di <i>install</i> , semua halaman <i>responsive</i> , dan semua fungsi berjalan dengan baik |

4.2.3 Analisa Pengujian

Dari hasil pengujian yang telah dilakukan maka dapat disimpulkan pada pengujian *black box testing* aplikasi dapat berjalan sesuai dengan yang diharapkan dan setiap fungsi dapat berjalan dengan baik. Jika ada kesalahan aplikasi bisa memberikan pemberitahuan bahwa terjadi kesalahan. Pada pengujian *compatibility testing*, aplikasi tetap *responsive* meski di instal di berbagai tampilan android dengan berbagai versi.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan, maka dapat disimpulkan sebagai berikut :

1. Metode *Rapid Application Development* dapat membantu dalam proses pembuatan aplikasi pemesanan dan informasi jasa *makeup*, dengan menerapkan metode *Rapid Application Development* pembuatan aplikasi menjadi singkat dan cepat hanya dalam waktu 90 hari.
2. Proses pemesanan jasa *makeup* sesuai dengan perias wajah dan pelanggan harapkan, sehingga memberikan kemudahan kepada pelanggan memesan jasa *makeup*, memudahkan perias dalam mempromosikan jasa *makeup* dan mencatat pesanan.
3. Aplikasi yang dibuat dapat berjalan sesuai yang diharapkan, aplikasi *responsive* pada tampilan android dengan versi android 7.0 (*nougat*), 8.0 (*oreo*), 9.0 (*pie*), dan android 10.

5.2 Saran

Aplikasi aplikasi pemesanan dan informasi jasa *makeup* masih perlu dilakukan pengembangan selanjutnya agar aplikasi ini menjadi lebih baik. Untuk pengembangan aplikasi ini selanjutnya agar bisa menambahkan *user admin*, galeri foto *makeup* yang lebih banyak, dan menambahkan informasi jenis *makeup* yang sesuai dengan bentuk wajah.

DAFTAR PUSTAKA

- Arman, Rahmelina, L., Sotar, dan Siska, Y. (2019) Perancangan Antar Muka Aplikasi Pemesanan Fotografi Dan Make Up Berbasis Online Pada Queen Photo Studio Padang, *Jurnal Sistem Informasi*.
- Asinjery. (2020) *Pengantar Teknologi Sistem Informasi*, Asinjery
- Bay Haqi, M. (2019). *Aplikasi SPK Pemilihan Dosen Terbaik Metode Simple Additive Weighting (SAW) dengan Java*. Sleman: DEEPUBLISH (Grup Penerbitan CV BUDI UTAMA).
- Cahyo, F. I., Sulistiowati, dan Limantara, J. (2018) Rancang Bangun Aplikasi Penentuan Vendor Pernikahan Pelanggan Untuk Usaha Wedding Organizer Menggunakan Metode Smarter Berbasis Web, *JSIKA*.
- Egadhana, E., Ferry, N., Budiarti, R., Rohmah, S., Leny, N. S., dan Magdalen, M. (2019) Rencana Bisnis Pengembangan Usaha Aplikasi “O-Bridge” Aplikasi Penyedia Vendor Jasa, *Journal of Entrepreneurship, Management, and Industry (JEMI)*.
- Elianti, L. D., dan Pinasti, V. S. (2017) *Makna Penggunaan Make Up Sebagai Identitas Diri, Universitas Negeri Yogyakarta*.
- Janah, R., dan Syafitri, Y. (2019). Membangun Aplikasi Pemesanan Barang Berbasis Web Dengan Menggunakan Framework Ajax Pada Penamart Bandar Lampung. *Jurnal Sistem Informasi Akuntansi (JUSINTA)* .
- Karman, J., Mulyono, H., dan Martadinati, A. T. (2019) *Sistem Informasi Geografis Berbasis Android Studi Kasus Aplikasi SIG Pariwisata*.
- Purwanto, dan hendrit, A. (2018) Rancang Bangun Sistem Informasi Pemesanan Ceker Mercon Ibu Elly, *Jurnal Informatika SIMANTIK* .
- Puspa, C. (2013). *Broadcast Make-up*. Jakarta: PT Gramedia Pustaka.
- Novendri, M. S., Saputra, A., dan Firman, C. E. (2019). Aplikasi Inventaris Barang Pada Mts Nurul Islam. *Jurnal Manajemen Dan Teknologi Informasi*.
- Rahmiati, Astuti, M., dan Dewi, I. P. (2019) Peningkatan Pengetahuan Make Up Dan Mehndi (Go Salon Muslimah Berbasis Android) Untuk Menciptakan

- Kewirausahaan Mahasiswa Dan Alumni Jurusan Tata Rias Dan Kecantikan Unp Improving Make Up And Mehndi), *Journal of Community Service*.
- Riyanto, V. (2017) Implementasi Metode Rapid Appliation Development Dalam Membangun E-commerce di Bidang Ukm, *Jurnal Pilar Nusa Mandiri*.
- Rizky, S. (2011). *Konsep Dasar Rekayasa Perangkat Lunak*. Jakarta: Prestasi Pustaka Publisher.
- Sintawati, I. D. (2018) Pembuatan Sistem Informasi Penjualan Kue Berbasis Web, *jurnal of information system management, applied, accounting and research*.
- Solichin, A. (2016) *Pemrograman web dengan php dan mysql*, Jakarta: Achmad Solichin.
- Syani, M., dan Werstantia, N. (2018) Perancangan Aplikasi Pemesanan Catering, *Jurnal Ilmiah Ilmu dan Teknologi Rekayasa* .

