

E-COMMERCE GOP-BER (GO PABRIK BERAS)
BERBASIS WEB (STUDI KASUS TOKO RAMA 35 MAKASSAR)

TUGAS AKHIR

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

Oleh :

EMMANUEL DORA WYCLIFFE TOLOLIU

1520221053



**PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS FAJAR
MAKASSAR
2021**

HALAMAN PENGESAHAN

**E-Commerce GOpBer (Go Pabrik Beras)
Berbasis Website**

Disusun Oleh :

**Emmanuel D. W. Tololiu
1520221053**

Telah diperiksa dan disetujui oleh Dosen Pembimbing

Makassar, 24 Januari 2023

Pembimbing I



**Kurniawan Harun Rashi, S.T., M.T
NIDN: 0903116901**

Pembimbing II



**Indah Purwitasari Ihsan, S.T., M.T
NIDN: 1221089001**

Mengetahui,

Dekan Fakultas Teknik



**Prof. Dr. Ir Erniati, S.T., M.T
NIDN: 0906107701**

Ketua Program Studi



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

PERNYATAAN ORISINAL

Penulis dengan ini menyatakan bahwa tugas akhir :

"E-Commerce GOPBER (Go - Pabrik Beras) Berbasis Web (Studi Kasus Toko Rama 35 Makassar)" adalah karya orisinal saya dan setiap serta seluruh sumber acuan telah ditulis sesuai dengan panduan penulisan ilmiah yang berlaku di fakultas teknik Universitas Fajar.

Makassar, 24 Januari 2023

Yang Menyatakan



Emmanuel D.W. Tololiu

KATA PENGANTAR

Segala Syukur dan Puji hanya bagi Tuhan Yesus Kristus karena hanya dengan kemuarahan dan kasih setia yang besar sehingga saya dapat menyelesaikan laporan skripsi ini yang berjudul “E-Commerce GOPBER (Go - Pabrik Beras) Berbasis Web (Studi Kasus Toko Rama 35 Makassar”. Laporan skripsi ini disusun sebagai salah satu syarat bagi mahasiswa Program Studi Teknik Elektro untuk dapat menyelesaikan studi Program Strata Satu (S1) pada Universitas Fajar Makassar. Pada saat penyusunan tugas akhir ini sangat banyak mendapat bantuan dari berbagai pihak, oleh karna itu dalam kesempatan ini pula ijinakan saya untuk mengucapkan terima kasih serta penghargaan setinggi-tingginya kepada:

1. Tuhan Yesus Kristus atas kasih dan berkat yang tidak terputus kepada kami.
2. Ayah dan ibu serta keluarga besar yang senantiasa mendoakan agar kami selalu diberi kekuatan dan kesabaran.
3. Ibu Prof. Dr. Erniati ST.,MT selaku Dekan Fakultas Teknik Universitas Fajar Makassar.
4. Safaruddin, S.Si., MT. selaku Ketua Prodi Teknik Elektro Universitas Fajar Makassar
5. Ibu Kurniawan Harun Rasyid, S.T., M.T. selaku Dosen Pembimbing I.
6. Ibu Indah Purwitasari Ihsan, S.T., M.T. selaku Dosen Pembimbing 2.
7. Dosen-dosen Prodi Teknik Elektro Universitas Fajar Makassar
8. Teman-teman Program Studi Teknik Elektro 2015. Terima Kasih untuk kebersamaan berbagi cerita selama ini dan memberi masukan serta arahan dalam penulisan tugass akhir.
9. KBMFT-UNIFA yang telah menerima kami menjadi keluarga.
10. Senior dan adik-adik yang telah melengkapi perjalanan dalam penulisan tugas akhir yang kami buat.
11. Seluruh orang-orang yang telah membantu kami selama proses pengerjaan tugas akhir ini.

Akhirnya, kepada Tuhan Yesus Kristus jualah penulis kembalikan semua permasalahan yang berada diluar kemampuan. Semoga Allah SWT senantiasa

melimpahkan rahmat dan hidayahnya kepada kami atas usaha, perjuangan dan pengorbanan yang dilakukan.

Makassar, 24 Januari 2023

Penulis

ABSTRAK

E-Commerce GOPBER (Go Pabrik Beras) Berbasis Web (Studi Kasus Toko Rama 35 Makassar), Emmanuel Dora Wycliffe Tololiu. Pada era globalisasi saat ini kemajuan teknologi sangat berkembang. Dalam perkembangan yang canggih dan sangat cepat ini sudah mempengaruhi berbagai bidang kehidupan dan profesi, tak terkecuali di dunia bisnis yang juga telah memanfaatkan kemajuan teknologi informasi yang sudah berkembang. Toko rama35 merupakan salah satu kategori toko usaha menengah yang menawarkan berbagai kategori produk beras,serta berbagai macam jenis beras lainnya.cara yang digunakanya saat ini masi semi manual dimana konsumen harus dating ke toko atau memesan via telfon. Pemesanan yang digunakan saat ini dengan cara semi manual masih sangat memiliki banyak kelemahan. Dengan memanfaatkan ilmu pengetahuan serta tekhnologi yang berkembang sangat pesat, maka penelitian ini memanfaatkan teknologi berupa *Smartphone* dan perangkat elektronik lainnya. Maka dari itu dbuatlah suatu aplikasi sistem informasi yang berbasi website untuk lebih memudahkan lagi bagi para konsumen untuk memesan dan memudahkan pemilik toko dalam menerima pesanan. Aplikasi ini dibuat dengan menggunakan bahasa pemrograman PHP dengan menggunakan bantuan *framework codeigniter* sehingga aplikasi ini dapat dijalankan di perangkat elektronik. Hasil yang diperoleh dari e-commerce gopber sesuai dengan hasil pengujian fungsional menggunakan metode blackbox menyatakan sistem 100% berfungsi dengan baik.

Kata Kunci : *E-Commerce, Website, Codeigniter*

ABSTRACT

GOPBER Commerce (Go-Rice Factory) Web-Based(Case Study toko Rama 35 Makassar), Emmanuel Dora Wycliffe Tololiu. In the current era of globalization, technological progress is very developed. In this sophisticated and very fast development has affected various areas of life and profession, not least in the business world which has also taken advantage of the advancement of information technology that has developed. Rama35 store is one of the categories of medium-sized business stores that offer various categories of rice products, as well as various other types of rice. The way it is used today is semi-manual where consumers must come to the store or order via phone. Bookings used today in a semi-manual way still have many disadvantages. By utilizing science and technology that is growing very rapidly, this research utilizes technology in the form of *smartphones* and other electronic devices. Therefore, an information system application that is based on the website to make it easier for consumers to order and make it easier for store owners to receive orders. This application is created using the PHP programming language using the help of *codeigniter framework* so that this application can be run on electronic devices. Results obtained from gopber e-commerce in accordance with the results of functional testing using the blackbox method stated the system 100% works properly.

Keywords: *E-Commerce, Website, Codeigniter*

DAFTAR ISI

HALAMAN JUDUL	
HALAMAN PENGESAHAN	i
HALAMAN ORISINAL	ii
KATA PENGANTAR	ii
ABSTRAK	iv
DAFTAR ISI	vi
DAFTAR GAMBAR	x
DAFTAR TABEL	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah.....	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	2
1.4 Batasan Masalah	2
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI.....	4
2.1 Tinjauan Pustaka	4
2.1.1 Sistem dan Informasi	4
2.1.1.1 Sistem.....	4
2.1.1.2 Informasi	4
2.1.1.4 Sistem Informasi	4
2.1.2 Pengertian E-Commerce	4
2.1.3 Pengertian Program.....	5
2.1.3.1 Website.....	5
2.1.3.2 Pengertian HTML	5
2.1.3.3 Pengertian XAMP	6
2.1.3.4 Pengertian PHP	6
2.1.3.5 Pengertian CSS	6
2.1.3.6 JavaScript	6

2.1.3.7 Codeigniter.....	7
2.1.3.8 Framework	7
2.1.3.9 MySQL.....	7
2.1.3.10 Database	7
2.1.3.11 Domain.....	8
2.1.3.12 Hosting	9
2.1.3.13 MVC.....	9
2.1.4 Flowchart	11
2.1.5 <i>Unifed Modeling Language (UML)</i>	13
2.1.6 Metode Pengembangan Sistem	18
2.1.7 Pengujian Sistem.....	19
2.2 Penelitian Terdahulu (State of The Art).....	20
2.3 Kerangka pikir.....	24
BAB 3 METODE PENELITIAN	25
3.1 Prosedur Penelitian	25
3.2 Desain/Rancangan Sistem.....	26
3.2.1 Usecase Diagram.....	26
3.2.2 Skenario UseCase	27
3.3 Activity Diagram.....	32
3.4 Sequence Diagram	37
3.4.1 Sequence Diagram Login.....	37
3.4.2 Sequence Membuat Akun	38
3.4.3 Sequence Mengelola Data Produk	38
3.4.4 Sequence Mengelola Profil	39
3.4.5 Sequence Memesan Produk	39
3.4.6 Sequence Melihat Produk	40
3.4.7 Sequence Mengecek Pesanan.....	40
3.4.8 Sequence Memproses Pesanan.....	41
3.4.9 Sequence Validasi Pembayaran	41
3.5 Class Diagram	42

3.6 Perancangan Database.....	43
3.7 Waktu dan tempat penelitian.....	44
3.8 Alat dan bahan penelitian.....	44
3.9 Perangkat keras dan lunak.....	45
3.10 Perancangan Antarmuka	45
3.11 Metode Pengumpulan Data	56
3.12 Tahap Analisis Data/Pengujian Sistem	56
BAB IV HASIL DAN PEMBAHASAN.....	57
4.1 Hasil Penelitian	57
4.1.1 Tampilan Halaman Utama	57
4.1.2 Tampilan Admin Penjual	59
4.1.2.1 Tampilan Login.....	59
4.1.2.2 Tampilan Dashboard.....	60
4.1.2.3 Tampilan Kelola Produk	61
4.1.2.4 Tampilan Order	62
4.1.2.5 Tampilan Kelola Pembayaran	63
4.1.2.6 Tampilan Kelola Pelanggan	64
4.1.3 Tampilan Pelanggan.....	65
4.1.3.1 Tampilan Daftar Pelanggan	65
4.1.3.2 Tampilan Dashboard Pelanggan	66
4.1.3.3 Tampilan Keranjang.....	67
4.1.3.4 Tampilan Status Order Pelanggan.....	68
4.1.3.5 Tampilan Pembayaran Oleh Pelanggan	69
4.1.3.6 Tampilan Edit Profil Pelanggan	70
4.1.3.7 Tampilan Review Pelanggan	71
4.1.4 Pembahasan.....	72
4.1.4.1 Pengujian Fungsional	72
4.1.4.2 Pengujian White Box	75
4.1.5 Rekapitulasi.....	97

BAB V PENUTUP	98
5.1 Kesimpulan	98
5.2 Saran.....	98
DAFTAR PUSTAKA	99

DAFTAR GAMBAR

Gambar 2.1 Sistem Pengujian	19
Gambar 3.1 Prosedur Penelitian	25
Gambar 3.2 Use Case Diagram Aplikasi	26
Gambar 3.3 Actifity Diagram Login	32
Gambar 3.4 Actifity Diagram Daftar	32
Gambar 3.5 Actifity Diagram Mengola Data Produk	33
Gambar 3.6 Actifity Mengola Profil	33
Gambar 3.7 Actifity Memesan Produk	34
Gambar 3.8 Actifity Melihat Produk	34
Gambar 3.9 Actifity Mengecek Pesanan	35
Gambar 3.10 Actifity Memproses Pesanan	35
Gambar 3.11 Actifity Validasi Pembayaran	36
Gambar 3.12 Actifity Konfirmasi Pembayaran	36
Gambar 3.13 Actifity Mengecek Status Pesanan	37
Gambar 3.14 Sequence Diagram Login	37
Gambar 3.15 Sequence Membuat Akun	38
Gambar 3.16 Sequence Mengelola Data Produk	38
Gambar 3.17 Sequence Mengola Profil	39
Gambar 3.18 Sequence Memesan Produk	39
Gambar 3.19 Sequence Melihat Produk	40
Gambar 3.20 Sequence Mengecek Pesanan	40
Gambar 3.21 Sequence Proses Pesanan	41
Gambar 3.22 Sequence Validasi Pembayaran	41
Gambar 3.23 Class Diagram	42
Gambar 3.24 Halaman Login	45
Gambar 3.25 Halaman Daftar	46

Gambar 3.26 Mengola Profil	47
Gambar 3.27 Kelola Data Poduk	48
Gambar 3.28 Proses Pesanan	49
Gambar 3.29 Penjual Cek Pesanan	50
Gambar 3.30 Validasi Pembayaran	51
Gambar 3.31 Melihat Produk	52
Gambar 3.32 Konsumen memesan produk	53
Gambar 3.33 Mengecek status pesanan	54
Gambar 3.34 Konsumen konfirmasi pembayaran	55
Gambar 4.1 Halaman Utama Website	57
Gambar 4.2 Script Halaman Utama Website	58
Gambar 4.3 Tampilan Login	59
Gambar 4.4 Script Tampilan Login	59
Gambar 4.5 Tampilan Dashboar	60
Gambar 4.6 Script Tampilan Dashboard	60
Gambar 4.7 Kelola Produk	61
Gambar 4.8 Script Kelola Produk	61
Gambar 4.9 Tampilan Order	62
Gambar 4.10 Script Tampilan Order	62
Gambar 4.11 Tampilan Kelola Pembayaran	63
Gambar 4.12 Script Kelola Pembayaran	63
Gambar 4.13 Tampilan List Pelanggan	64
Gambar 4.14 Script List Pelanggan	64
Gambar 4.15 Tampilan Halaman Buat Akun Untuk Pelanggan	65
Gambar 4.16 Script Halaman Buat Akun Untuk Pelanggan	65
Gambar 4.17 Tampilan Dashboard Pelanggan	66
Gambar 4.18 Script Tampilan Dashboard Pelanggan	66
Gambar 4.19 Tampilan Keranjang	67
Gambar 4.20 Script Tampilan Keranjang	67
Gambar 4.21 Tampilan Status Order Pelanggan	68
Gambar 4.22 Script Status Order Pelanggan	68

Gambar 4.23 Tampilan Pembayaran Oleh Pelanggan.....	69
Gambar 4.24 Script Tampilan Pembayaran Oleh Pelanggan	69
Gambar 4.25 Tampilan Edit Profil Pelanggan.....	70
Gambar 4.26 Script Tampilan Edit Profil Pelanggan	70
Gambar 4.27Tampilan Riview Pelanggan.....	71
Gambar 4.28 Script Tampilan Review Pelanggan.....	71
Gambar 4.29 <i>Flowchart Form Login</i>	75
Gambar 4.30 <i>Flowgraph Form Login</i>	76
Gambar 4.31 <i>Flowchart Buat Akun</i>	77
Gambar 4.32 <i>Flowgraph Form Buat Akun</i>	78
Gambar 4.33 <i>Flowchart Form Mengola Data Produk</i>	79
Gambar 4.34 <i>Flowgraph Form Mengola Data Produk</i>	80
Gambar 4.35 <i>Flowchart Form Mengola Profil</i>	81
Gambar 4.36 <i>Flowgraph Form Mengola Profil</i>	82
Gambar 4.37 <i>Flowchart Form Memesan Produk</i>	83
Gambar 4.38 <i>Flowgraph Form Memesan Produk</i>	84
Gambar 4.39 <i>Flowchart Form Melihat Produk</i>	85
Gambar 4.40 <i>Flowgraph Form Melihat Produk</i>	86
Gambar 4.41 <i>Flowchart Form Mengecek Pesanan</i>	87
Gambar 4.42 <i>Flowgraph Form Mengecek Pesanan</i>	88
Gambar 4.43 <i>Flowchart Form Memproses Pesanan</i>	89
Gambar 4.44 <i>Flowgraph Form Memproses Pesanan</i>	90
Gambar 4.45 <i>Flowchart Form Validasi Pembayaran</i>	91
Gambar 4.46 <i>Flowgraph Form Validasi Pembayaran</i>	92
Gambar 4.47 <i>Flowchart Form Konfirmasi Pembayaran</i>	93
Gambar 4.48 <i>Flowgraph Form Konfirmasi Pembayaran</i>	94
Gambar 4.49 <i>Flowchart Mengecek Status Pesanan</i>	95
Gambar 4.50 <i>Flowgraph Mengecek Status Pesanan</i>	96

DAFTAR TABEL

Tabel 2.1 Simbol – simbol <i>Flowchart</i>	12
Tabel 2.2 Simbol – simbol <i>Use Case Diagram</i>	14
Tabel 2.3 Simbol – simbol <i>Activity Diagram</i>	15
Tabel 2.4 Simbol – simbol <i>Sequence Diagram</i>	16
Tabel 2.5 Simbol – simbol <i>Class Diagram</i>	17
Tabel 2.6 Penelitian Terdahulu.....	20
Tabel 3.1 Skenario Usecase Login	27
Tabel 3.2 Skenario Usecase Penjual.....	27
Tabel 3.3 Skenario Usecase Mengola Profil	27
Tabel 3.4 Skenario Usecase Mengedit data produk	28
Tabel 3.5 Skenario Usecase mengecek pesanan.....	28
Tabel 3.6 Skenario Usecase Validasi Pembayaran.....	29
Tabel 3.7 Skenario Usecase Memproses Pesanan	29
Tabel 3.8 Skenario Usecase Informasi daftar toko.....	29
Tabel 3.9 Skenario Usecase Melihat Produk.....	30
Tabel 3.10 Skenario Usecase Memesan Produk.....	30
Tabel 3.11 Skenario Usecase Konfirmasi Pembayaran.....	31
Tabel 3.12 Skenario Usecase Mengecek Status Pesanan	31
Tabel 4.1 Pengujian BlackBox	72
Tabel 4.2 Rekapitulasi	97

BAB I

PENDAHULUAN

1.1. Latar Belakang

Pada era teknologi modern saat ini perkembangan teknologi sangatlah cepat dan berdampak besar untuk kehidupan setiap masyarakat, teknologi sangatlah berkembang pesat sehingga mendorong manusia untuk berinovasi lebih untuk menciptakan aplikasi informasi agar lebih mempermudah setiap masyarakat untuk menggunakannya, dalam kehidupan sehari-hari teknologi informasi sangatlah di butuhkan dalam setiap kalangan masyarakat untuk menunjang keseharian aktivitas. Teknologi informasi pada era modern sekarang sangatlah berkembang sehingga untuk melakukan pekerjaan sangatlah mudah.

Pada kasus toko rama35 ketika melakukan penjualan maupun transaksi masi menggunakan cara manual, ketika ingin berbelanja sebagian dari pelanggan tidak mengetahui bahwa toko rama35 itu sendiri mempunyai keterbatasan produk sehingga produk yang ingin di beli tidak ada, serta terkadang toko rama35 sendiri ketika melakukan pemasaran masi sangat menggunakan cara manual yaitu dengan berkomunikasi dengan orang yang nantinya akan di sampaikan melalui mulut ke mulut. Serta jarak yang kadang jadi penghalang untuk pelanggan yang lain untuk datang ke toko rama35.

Di PB Saigon, sebuah perusahaan penanaman padi didirikan di Mayalengka. Dalam semua operasi akuntansi, perusahaan selalu menggunakan sistem manual, yang dicatat dalam tagihan besar, misalnya untuk mencatat transaksi pembelian beras dari petani, penjualan produk, dan data persediaan produk. Oleh karena itu, data yang dihasilkan rentan terhadap masalah misalkan data yang tidak realistis, sehingga dapat menyebabkan hilangnya file atau korupsi dikarenakan hanya disimpan di dalam catatan besar serta sulitnya pemasaran barang karena barang tidak dapat menjangkau calon pelanggan (Rifaldi). fahruly dan Fenny syafariani s.si., m.stat, 2018).

Di Suatu tempat beras tiredi kota semarang suatu usaha bisnis dalam bidang sembako yang melakukan aktivitas distribusi secara manual atau secara langsung sehingga teknologi pada aktivitas distribusi belum diterapkan. Pada saat ini aktivitas

pemasaran memasuki tahap yang sudah lebih modern, pada pendistribusian dan pemasaran menggunakan teknologi sebagai sistem informasi . Sehingga diharapkan teknologi mampu memberi manfaat luas terhadap persaingan yang cenderung lebih kompetitif(Hendri dan Indra,2013).

Berdasarkan masalah yang ada di atas maka perlunya di adakan aplikasi untuk pemilik toko yang berjudul “***E-Commerce GOP-BER(Go-Pabrik Beras) Berbasis Website***”yang berstudi kasus di toko Rama 35 agar mempermudah para konsumen serta pelanggan yang ingin membeli beras tanpa perlu keluar rumah dan ingin diantarkan pada lokasi sesuai keinginan pelanggan yang melakukan pemesanan serta lebih mudah diakses melalui teknologi yang ada.

Dengan adanya aplikasi ini sekiranya para pelanggan lebih dimudahkan dalam proses pesan memesan beras dari toko rama35 serta pemilik toko lebih bisa memasarkan lebih besar lagi produk-produk beras yang ada di toko rama35 sehingga baik dari pemilik toko maupun pelanggan dapat berinteraksi satu sama lain tanpa perlu ketemu untuk memesan produk beras.

1.2. Rumusan Masalah

Sesuai dengan latar belakang masalah yang telah di kemukakan, maka dapat di uraikan rumusan masalah, sebagai berikut :

1. Bagaimana cara mempermudah proses penjualan produk beras berbasis sistem informasi ?
2. Bagaimana hasil pengujian sistem e-commerce berbasis sistem informasi?

1.3. Tujuan Penelitian

Tujuan dari pembuatan aplikasi ini adalah :

1. Membangun *E-Commerce* Gopber(Go-Pabrik Beras) berbasis website.
2. Mengetahui hasil pengujian *E-Commerce*(Go-Pabrik Beras) berbasis sistem informasi.

1.4. Batasan Masalah

Agar menghindari meluasnya permasalahan, maka :

1. Berstudi kasus di toko rama35.
2. Belum mencakup laporan penjualan.

3. Proses Validasi/Konfirmasi pembayaran belum terkoneksi dengan sistem Bank.

I.5. Manfaat Penelitian

Dari hasil penelitian terdapat beberapa manfaat, antara lain :

1. Dengan adanya aplikasi ini diharapkan toko beras dapat mampu dikenali oleh masyarakat agar usaha yang dibangun dapat berkembang serta menjadi penghasil beras yang berkualitas.
2. Mempermudah agar masyarakat tak perlu lagi keluar rumah untuk membeli beras, mereka hanya perlu memesan melalui aplikasi *website*, serta bermanfaat untuk banyak orang.

BAB II

LANDASAN TEORI

2.1. Tinjauan Pustaka

2.1.1 Sistem dan Informasi

Sistem merupakan sebuah sistem di mana sebuah kelompok yang merupakan gabungan dari beberapa orang, media, teknologi, fasilitas, dan prosedur – prosedur pengendalian yang di tujukan untuk mengambil suatu keputusan tertentu (Dani Tri Indraswara, 2014).

2.1.1.1 Sistem

Sistem merupakan suatu proses dari tahapan pendefinisian kebutuhan-kebutuhan dari siklus perkembangan sistem baru atau sistem yang akan dibentuk (Sukisno dan Winda Frantika, 2017).

2.1.1.2 Informasi

Informasi adalah data yang telah di klarifikasi atau diolah atau diinterpretasi untuk digunakan dalam proses pengambilan keputusan. Sistem pengolahan informasi mengolah data menjadi informasi atau tepatnya pengolah data dari bentuk tak berguna menjadi berguna bagi penerimanya. Nilai informasi berhubungan dengan keputusan. Bila tidak ada pilihan atau keputusan, maka informasi menjadi tidak diperlukan (Sutabri 2016).

2.1.1.3 Sitem Informasi

Informasi yaitu sebuah sistem yang saling terhubung dan dijalankan untuk mendukung segala kegiatan didalam organisasi (Desi Tri Hernandhi, dkk, 2018).

2.1.2 Pengertian *e-commerce*

E-Commerce adalah cara bagi suatu pelanggan buat bisa belanja produk yg diinginkan secara online, E-Commerce yaitu suatu pemasaran produk yg eksklusif & dipasarkan memakai internet, baik buat bisni (bisnis to bisnis) juga buat pelanggan (*Bisnis to Consumen*) (Fifit Alfiah, dkk, 2014).

2.1.3 Pengertian Program

Secara umum program merupakan sekumpulan instruksi atau statement yang dimana didalam bahasa yang dimengerti komputer. Instruksi tersebut berguna supaya mengatur kegiatan apa saja yang akan dilakukan oleh komputer agar dapat dan menghasilkan suatu hasil atau keluaran yang diharapkan. Dapat juga dikatakan bahwa suatu program merupakan himpunan atau gabungan instruksi tertulis yang dibuat oleh programmer atau suatu bagian *executable* dari sebuah perangkat lunak(Sidik dan Surryoprayogo,2015)

2.1.3.1 Website

Website yaitu suatu kegiatan untuk tampilkan diri atau sekelompok di internet. *Website* anda yaitu suatu lokasi di *internet*, siapa pun yang di dunia ini dapat mengunjunginya, kapan saja mereka dapat mengetahui tentang seseorang atau organisasi, mengajukan pertanyaan pada mereka, mengajukan masukan atau bahkan mengetahui dan membeli barang suatu perusahaan. Internet bagaikan sebuah pusat dagang terbesar di dunia dan *website* anda adalah salah satu toko / kios / kantor di pusat perdagangan tersebut(Rudika Harminingtyas,2014).

2.1.3.2 Pengertian HTML

HTML yang berupa singkatan dari *Hyper Text Markup Language* yaitu serangkaian kode program yang merupakan landasan dari representasi visual sebuah halaman *Web*. Didalamnya mempunyai isi kumpulan informasi yang disimpan dalam tag-tag tertentu, dimana tag-tag tersebut dipakai untuk melakukan format terhadap informasi yang dimaksud. Berbagai perkembangan telah dilakukan terhadap kode *HTML* dan telah menghasilkan teknologi-teknologi baru di dalam dunia pemrograman *web*. Sebagai mana demikian, sampai sekarang *HTML* tetap berdiri kokoh sebagai dasar dari bahasa *web* seperti *PHP*, *ASP*, *JSP* dan lainnya. Bahkan secara umum, mayoritas situs *web* yang ada di *Internet* pun masih tetap menggunakan *HTML* sebagai teknologi utama mereka(Frederick Constantianus, 2005).

2.1.3.3 Pengertian XAMP

XAMP merupakan suatu *Apache server software* web yang di mana dalamnya sudah menyediakan *MySQL database server* serta mendukung *programming PHP*. *XAMP* juga merupakan singkatan dari kata X(Untuk empat sistem operasi) yaitu *Apache,MySQL,PHP,Pearl*(Yusi,Eko Adi,dkk,2012).

2.1.3.4 Pengertian PHP

PHP (Hypertext Preprocessor) adalah bahasa scripting server side yang dipadukan dengan HTML dimana semua perintah yang diberikan dijalankan sepenuhnya di server sedangkan yang dikirimkan ke client (browser) hanyalah hasilnya. Sintaks dan semantik PHP memiliki kesamaan dengan C, JAVA, PERL dan <? Seperti pembukaan dan akhir PHP tambahan?>, Pasangan kode kedua ini berfungsi sebagai tag kode PHP. Berdasarkan tag ini, server dapat memahami dan kemudian memproses kode PHP, hasilnya dikirim ke browser.(Yusi Ardi,Eko Adi dan Nurdin Bahtiar,2012).

2.1.3.5 Pengertian CSS

Cascading Style Sheet (CSS) yaitu *language stylesheet* yang digunakan untuk menggambarkan presentasi dokumen yang dibuat dalam bahasa markup. *CSS* adalah dokumen yang berguna untuk mengkonfigurasi komponen halaman web, inti dari dokumen ini adalah untuk memformat halaman web standar menjadi bentuk web dengan kualitas yang lebih indah dan menarik.(Yusi Ardi,Eko Adi dan Nurdin Bahtiar,2012).

2.1.3.6 JavaScript

Javascript yaitu bahasa yang terbentuk kumpulan skrip yang kegunaanya untuk menambahkan interaksi antara halaman *web* dengan pengunjung tampilan *web*. *Javascript* dijalankan pada sisi klien yang akan memberikan kemampuan fitur-fitur tambahan halaman *web* yang lebih baik dibandingkan fitur-fitur yang terdapat pada *HTML*(Yusi Ardi Binarso,dkk ,2012)

2.1.3.7 CodeIgniter

CodeIgniter merupakan aplikasi *open source* yang berupa *framework* dengan model MVC (*Model, View, Controller*) untuk membangun *website* dinamis dengan menggunakan PHP. *Framework* dapat diartikan sebagai kumpulan dari fungsi-fungsi/prosedur-prosedur dan *class-class* untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang pemrograman, tanpa harus membuat fungsi atau *class* dari awal (Yusi Ardi Binarso, dkk, 2012).

2.1.3.8 Framework

Framework yaitu sebuah kerangka kerja atau juga dapat diartikan sebagai kelompok script yang dapat membantu pengembangan aplikasi dalam mengatasi berbagai masalah pemrograman seperti koneksi ke *database*, pemanggilan *variable*, dan file. sehingga *developer* lebih cepat membangun aplikasi. Berbagai macam *framework* yaitu *codeigniter, cake php, yii, spring, zend, panda, dan hibernate* (Dewi Rosmala, dkk, 2011).

2.1.3.9 MySQL

MySQL merupakan suatu sistem manajemen sistem (DBMS) yang populer mempunyai fungsi untuk *Relation Database Manajemen Sistem* (RDBMS). Untuk itu *MySQL* merupakan software sebuah aplikasi sifatnya yang *open source* serta server *MySQL basis data* mempunyai pekerjaan yang *reliable, cepat, serta mudah* ketika digunakan serta arsitektur bekerja dengan *embedded systems* atau *user server*. Karena *open source factor* dan populer tersebut maka *proaes replikasi basis data* cocok untuk didemonstrasikan (Herman Yuliansyah, 2014).

MySQL adalah sistem yang untuk digunakan mengolah basis data yang banyak pakai sebagai untuk merancang sistem yang memakai *database* (Daniel Dido Jantce TJ Sitinjak, dkk, 2020)

2.1.3.10 Database

Basis Data atau *databse* yaitu didalam computer kelompok informasi di simpan secara sistematis sehingga informasi dari basis data itu dapat menggunakan sebuah program computer untuk diperiksa. Sistem basis data kegunaan utama datanya yaitu

agar pengguna dapat sebuah pandangan untuk menyusun (*view*) *abstraksi* data. Bertujuan supaya hal ini sistemnya supaya menyederhanakan interaksi antar basis data dan pengguna dengan sistemnya supaya dapat mempersentasikan kepada para administrator,programer,dan pengguna tentang pandangan.(Andy Andaru,2015).

Database merupakan kelompok field-field antara satu file yang mempunyai kaitan dengan field yang lain membentuk bangunan sehingga data menginformasikan kondisi lalu lintas untuk bahasa tertentu. *Database* juga merupakan sebagian folder yang saling terhubung, dari tiap file yang ada relasi tersebut biasa ditunjukkan dengan kunci. Suatu database menunjukkan kumpulan informasi dalam satu lingkup data yang dipakai. Di dalam suatu folder terdapat record-record yang sama bentuk, sama besar, sejenis, yang seragam merupakan suatu kumpulan entity. Suatu record terdiri dari field-field yang saling terhubung untuk menunjukkan bahwa field tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. Suatu sistem manajemen database berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi sistem aplikasi database serta set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data (Mhd Bustanur Rahmad, Tedy Setiady,2014).

2.1.3.11 Domain

Domain merupakan alamat website untuk mengganti dari alamat *ip* menjadi kata-kata yang mudah diingat dalam suatu identifikasi alamat dalam sebuah jaringan khususnya alamat *web* . Suatu domain mempunyai keunikan yang terletak di setiap eksistensi saat digunakan. Pada Indonesia sendiri ada dua kumpulan ekstensi yang digunakan yaitu *Country Code Top Level Domain (CCLTD)* dan *Top Level Domain(TLD)*. Pada indonesia pertumbuhan pemakai internet berdasarkan data kementerian komunikasi dan informatika (Kemkominfo) yaitu sejumlah 80 juta pemakai. (kominfo.go.id.2014). Dalam kesempatan ini bisnis dalam bidang kreatif industri memberi kesempatan untuk menciptakan peluang penggunaan website sebagai media dalam bidang bisnis . Salah satu industri kreatif yang diciptakan pada tahun 2007 adalah PT. Berkomitmen untuk menyediakan layanan pendaftaran domain dan server hosting untuk website dimana setiap hari permintaan pendaftaran domain mencapai 100 domain dan 70 permintaan pendaftaran hosting baru, Beon Intermedia dapat menjadi

tolak ukur pelayanan prima, *hosting* situs *web* ini memungkinkan pengguna untuk membandingkan manfaat dan kemudahan yang diberikan oleh penyedia layanan yang masih mereka gunakan dengan penawaran pesaing di lini bisnis yang sama. Oleh karena itu, pengguna akan aktif dalam mentransfer data dan nama domain yang mereka daftarkan di layanan sebelumnya ke penyedia layanan baru. Pada pembahasan kali ini hanya disarankan pada proses transfer domain dari service provider lama ke service provider baru. Definisi Domain Menurut nama domain Premysl Raban.eu (domain eu), Domain Name System (DNS) sebenarnya adalah sistem global (Global Addressing System) yang mengubah alat Protokol Internet (IP) yang biasa dibuat angka dalam nama (domain) dan banyak lagi(Samsul Arifin, Yolanda Krisnadita,2017).

DNS (Domain name system) merupakan sebuah fasilitas dalam suatu sistem operasi untuk memetakan alamat/*domain* ke alamat *IP*. Ketika sedang membuka *website*, yang diketikkan adalah merupakan alamat dari *website* tersebut yang sebenarnya yaitu merupakan alamat *IP*. Sebagai contoh yaitu ketika sedang diketik www.google.com maka yang sebenarnya dituju adalah alamat *IP* dari *google*, yaitu *173.194.38.146*. *IP* ini dapat diketahui apabila menjalankan ping dari *CMD* dari komputer dan mengetikkan ping *google.com*, maka ketika komputer telah terkoneksi dengan internet akan terlihat *IP* dari *google*(Taufik Hidayat, dkk,2018).

2.1.3.12 Hosting

Hosting juga sering disebut sebagai tempat penyimpanan data di internet. Data yang sama ini dapat berasal dari dokumen, video, gambar, musik, dll. dapat berupa data. Dengan *hosting* ini, pengguna dapat menyimpan file di Internet. Data yang disimpan di Internet dapat diakses oleh pengguna lain, pribadi (untuk Anda) dan dipublikasikan, tergantung pada konfigurasi yang kami buat.

Ada banyak jenis *hosting* itu sendiri diantaranya *shared hosting* yang merupakan layanan *hosting* dimana sistemnya beroperasi pada satu akun *hosting* yang digabungkan dengan beberapa akun *hosting* lain pada server yang sama, maka layanan atau layanan yang diterima akan dibagikan secara otomatis. Keuntungan dari *shared hosting* adalah murah. Namun kelemahannya adalah jika pengguna lain terlalu buta saat menggunakan *hosting*, server mudah crash dan tidak ada lagi ruang atau ruang yang tersisa.

Dedicated hosting yaitu layanan hosting di mana server digunakan oleh satu pengguna. Pengguna *dedicated hosting* bisa mendapatkan banyak keuntungan, salah satu keuntungannya adalah kinerja website dan keamanan privasi lebih terlindungi karena server hosting yang kita gunakan tidak digunakan oleh pihak ketiga, akan lebih aman dan nyaman di simpan.

Reseller hosting yaitu penjualan kembali hosting oleh pengguna, yaitu siapa saja yang menggunakan *reseller hosting* dapat mengelola kembali hosting yang mereka buat dan kemudian menjualnya kembali dengan merek mereka sendiri dengan harga dan fitur yang ditentukan, siapa saja yang ingin menjadi *reseller* tanpa *reseller* tuan rumah. situasi sulit apa pun. Dan privasi penjual dilindungi dengan baik oleh penyedia web terkemuka.

VPS (Virtual Private Server) yaitu layanan hosting yang menyewakan virtual server, kelebihan yang bisa didapatkan dari pengguna saat menggunakan *VPS (Virtual Private Server)* adalah pengguna bebas menentukan sistem operasi dan platform yang diinginkan. Gunakan untuk memberikan kebebasan dan fleksibilitas lebih kepada pengguna dalam mengoptimalkan layanan bisnis online yang mereka kelola. Tentu saja harga sewa *VPS* jauh lebih mahal daripada server konvensional karena banyak keuntungan yang bisa didapat darinya. Namun, *VPS* itu sendiri memiliki kelemahan yaitu tidak dapat memenuhi kebutuhan konektivitas khusus antar server. Pada dasarnya, ketika pengguna membutuhkan pengaturan yang lebih optimal untuk keamanan dan keterbatasan yang dihadapi oleh web online, pengguna *VPS* tidak dapat lagi membelinya. Pada kenyataannya, dari segi kualitas *VPS* tidak jauh berbeda dengan web hosting, kelebihanannya hanya pada kebebasan pengguna *VPS* untuk menentukan sistem operasi dan platform yang akan mereka gunakan. (Ipan Ripai, M.Kom,2017).

2.1.3.13 MVC

Model-View-Controller (MVC) yaitu suatu konsep yang dipasarkan oleh penemu *smalltalk (trygve reenskaug)* yaitu guna meng_enkapsulasi file bersama dengan proses(model), mengisolasi dari proses manipulasi (controller) dan tampilan (view) untuk di presentasikan di suatu pengguna definisi teknis antarmuka arsitektur *MVC* dibagi menjadi tiga level.

a. Model

Digunakan buat mengelola berita & memberi memahami pengamat mengenai perubahan berita. Hanya contoh yg berisi data & fungsi yg terkait menggunakan pemrosesan data. Model merangkum nir hanya data & fungsi yg beroperasi pada dalamnya. Pendekatan contoh dipakai buat memodelkan atau menghitung abstraksi menurut proses global nyata. Ini nir hanya mencatat status suatu proses atau sistem, namun jua bagaimana sistem itu bekerja. Misalnya, programmer bisa mendefinisikan template yg mengaitkan pemrosesan internal menggunakan antarmuka pengguna grafis (*graphical user interface*).

b. View

Bertanggung jawab buat menata grafis ke suatu perangkat. View umumnya mempunyai interaksi 1-1 menggunakan suatu bagian atas layar & mengetahui bagaimana buat membuatnya. View inheren dalam contoh & merender isinya ke bagian atas layar. Selain itu, waktu contoh berubah, view secara otomatis menggambar ulang bagian layar yg terkena perubahan buat menerangkan perubahan tadi. Terdapat kemungkinan beberapa view dalam contoh yg sama & masing-masing view tadi bisa merender isi contoh buat bagian atas tampilan yg berbeda.

c. Controller




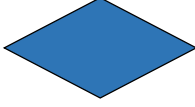




Menerima input menurut pengguna & mengintruksikan contoh & view buat melakukan aksi sinkron masukan tersebut. Sehingga, controller bertanggung jawab buat menata aksi pengguna akhir terhadap respon pelaksanaan. Sebagai contoh, waktu pengguna melakukan klik tombol atau menentukan item menu, controller bertanggung jawab buat memilih bagaiman pelaksanaan seharusnya merespon (Pastima Simanjuntak, Arwin Kusnady, 2016)


2.1.4. Flowchart

Flowchart merupakan urutan-urutan langkah kerja suatu proses yang digambarkan dengan menggunakan simbol-simbol yang disusun secara sistematis. (Iswandy, 2015).

Secara formal, flowchart didefinisikan sebagai skema penggambaran dari algoritma atau proses. Tabel berikut menampilkan simbol-simbol yang digunakan dalam menyusun flowchart. (Harumy, dkk, 2016).

Tabel 2.1 Simbol-simbol Dalam *Flowchart*

	<p>Terminator Sebagai simbol ‘<i>START</i>’ atau ‘<i>END</i>’ untuk memulai atau mengakhiri <i>flowchart</i>.</p>
	<p>Input/Output Digunakan untuk menuliskan proses menerima data atau mengeluarkan data.</p>
	<p>Proses Digunakan untuk menuliskan proses yang diperlukan, misalnya operasi aritmatika.</p>
	<p>Conditional/Decision Digunakan untuk menyatakan proses yang membutuhkan keputusan.</p>
	<p>Preparation Digunakan untuk memberikan nilai awal.</p>
	<p>Arrow Sebagai penunjuk arah dan alur proses.</p>
	<p>Connector (<i>On-page</i>) Digunakan untuk menyatukan arrow.</p>
	<p>Connector (<i>Off-page</i>) Digunakan untuk menghubungkan <i>flowchart</i> yang harus digambarkan pada halaman yang berbeda. Biasanya pada simbol ini diberi nomor sebagai</p>

	penanda, misalnya angka 1.
	<i>Display</i> Digunakan untuk menampilkan data ke <i>monitor</i> .

2.1.5. UML (*Unified Modelling Language*)

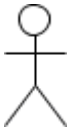



UML (Unified Modelling Language) yaitu suatu standar bahasa yang banyak digunakan di dunia industri untuk mendefinisikan *requirement*, membuat analisis dan desain, serta menggambarkan arsitektur dalam pemrograman berorientasi objek

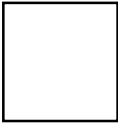

- a) *Use Case Diagram*, *Use case diagram* merupakan pemodelan untuk kelakuan *sistem informasi* yang akan dibuat. *Use case* bekerja dengan mendeskripsikan tipikal interaksi antara user sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sistem itu dipakai.
- b) *Class diagram*, *Class diagram* merupakan gambaran struktur sistem dari segi pendefinisian kelas yang akan dibuat untuk membangun sistem. *Class diagram* terdiri dari atribut dan operasi dengan tujuan pembuat program dapat membuat hubungan antara dokumentasi perancangan dan perangkat lunak sesuai.
- c) *Sequence Diagram*, *Sequence diagram* menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Gambaran sequence diagram dibuat minimal sebanyak pendefinisian use case yang memiliki proses sendiri atau yang penting semua use case yang telah didefinisikan interaksi jalannya pesan sudah dicakup pada sequence diagram sehingga semakin banyak use case yang didefinisikan, maka sequence diagram yang harus dibuat juga semakin banyak.
- d) *Activity Diagram*, *Activity diagram* merupakan diagram yang menggambarkan *workflow* atau aktivitas dari sebuah sistem yang ada pada perangkat lunak.
- e) *Component diagram* mendeskripsikan struktur & interaksi antar komponen piranti lunak, termasuk ketergantungan (*dependency*) pada antaranya. Komponen piranti lunak merupakan modul berisi code, baik berisi source code

juga binary code, baik library juga executable, baik yg ada dalam compile time, link time, juga run time. Deployment/physical diagram merupakan penggambaran lebih jelasnya bagaimana komponen dideploy pada infrastruktur sistem, dimana komponen akan terletak (dalam mesin, server atau piranti keras apa), bagaimana kemampuan jaringan dalam lokasi tersebut, spesifikasi server, & hal-hal lain yg bersifat fisikal Sebuah node merupakan server, workstation, atau piranti keras lain yg dipakai buat mendeploy komponen pada lingkungan sebenarnya. (Putra,2019).

Berikut daftar simbol-simbol dari *Use Case Diagram*, *Actifity Diagram*, *Sequence Diagram*, *Class Diagram*.

Tabel 2.2 Simbol-simbol use case diagram


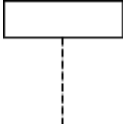




No.	Gambar	Nama	Keterangan
1.		<i>Actor</i>	menspesifikasikan himpunan kiprah yang dimainkan sang pengguna waktu berinteraksi menggunakan use case
2.		<i>generalization</i>	interaksi antara objek anak (descendent) menyebarkan tingkahlaku & struktur data berdasarkan objek yang terdapat pada atasnya objek induk (ancestor)
3.		<i>include</i>	menspesifikasikan bahwa use case asal secara eksplisit
4.		<i>association</i>	apa yang menghubungkan antara objek satu menggunakan objek lainnya

5.		<i>system</i>	Menspesifikasikan paket yg menampilkan sistem secara terbatas
6.		<i>use case</i>	Deskripsi menurut urutan aksi-aksi yg ditampilkan sistem yg membuat suatu output yg terukur bagi seluruh aktor

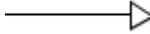
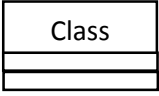


Tabel 2.3 simbol-simbol *Activity diagram*

No.	Gambar	Nama	Keterangan
1.		<i>initial Node</i>	bagaimana objek dibuat atau diawali
2.		<i>partition (vertical)</i>	pengelompokkan aktifitas menurut aktor & sistem
3.		<i>action</i>	State berdasarkan sistem yg mencerminkan hukuman berdasarkan suatu aksi
4.		<i>fork Node</i>	satu genre yg dalam termin eksklusif berubah sebagai beberapa genre
5.		<i>actifity final node</i>	bagaimana objek dibuat & diakhiri
6.		<i>decision node dan merge node</i>	upaya buat menghindari asosiasi menggunakan lebih berdasarkan dua objek

Tabel 2.4 simbol-simbol *sequence diagram*

NO.	GAMBAR	NAMA	KETERANGAN
1.		<i>lifeLine (actor)</i>	Objek actor, antarmuka yg saling berinteraksi
2.		<i>lifeLine (entity)</i>	objek entity, antarmuka yg saling berinteraksi
3.		<i>message</i>	Spesifikasi menurut komunikasi antar objek yg memuat kabar-kabar mengenai aktifitas yg terjadi
4.		<i>Message</i>	spesifikasi menurut komunikasi antar objek yg memuat kabar-kabar mengenai aktifitas yg terjadi
5.		<i>message</i>	pesan yg dikirim buat objek itu sendiri
6.		<i>execution Occurance</i>	memperlihatkan penekanan kontrol objek dalam suatu waktu

Tabel 2.5 Simbol-simbol *Class Diagram*

No.	Gambar	Nama	Keterangan
1.		<i>generalization</i>	interaksi dimana objek anak (descendent) mengembangkan konduite & struktur data menurut objek yg terdapat diatasnya objek induk (ancestor)
2.		<i>class</i>	Himpunan berdasarkan objek-objek yg menyebarkan atribut dan operasi yg sama
3.		<i>include</i>	menspesifikasikan bahwa use case asal secara eksplisit
4.		<i>association</i>	apa yg menghubungkan antara objek satu menggunakan objek lainnya
5.	1 , 1..*	<i>multiplicity</i>	jumlah banyaknya objek sebuah group yg berelasi menggunakan sebuah objek lain dalam group lainyang berasosiasi menggunakan group tersebut

2.1.6. Metode Pengembangan Sistem

Perancangan *Sistem Informasi* ini dibangun dengan menggunakan Model *Waterfall*. Menurut (Roger S. Pressman (2002:37)) *waterfall* yaitu suatu model perkembangan perangkat lunak dilakukan secara sekuensial, dimana satu tahap dilakukan setelah tahap sebelumnya selesai dilaksanakan. Roger S. Pressman membagi model *waterfall* ini menjadi 6 fase, namun secara umum fasenya sama dengan model *waterfall* pada umumnya. Di bawah ini adalah deskripsi langkah-langkah yang dilakukan dalam model ini, menurut Pressman.

:

1. *System / Information Engineering and Modeling*

Simulasi ini dimulai dengan mendefinisikan kebutuhan seluruh sistem yang akan diimplementasikan dalam perangkat lunak. Hal ini sangat penting karena perangkat lunak harus dapat berinteraksi dengan elemen lain seperti perangkat keras, database, dll. Fase ini sering disebut dengan definisi proyek.

2. *Software Requirements Analysis*.

Proses kebutuhan meneliti diintensifkan dan difokuskan pada perangkat lunak. Untuk mengetahui sifat program yang dijalankan, perancang perangkat lunak harus memahami domain informasi perangkat lunak, seperti fungsi yang diperlukan, antarmuka pengguna, dll. Dari dua aktivitas ini (penelitian perangkat lunak dan persyaratan sistem) harus didokumentasikan dan disajikan kepada pelanggan.

3. *Design*.

Proses ini digunakan untuk mengubah kebutuhan-kebutuhan di atas menjadi representasi ke dalam bentuk "*blueprint*" *software* sebelum *coding* dimulai. Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Seperti 2 aktivitas sebelumnya, maka proses ini juga harus didokumentasikan sebagai konfigurasi dari *software*.

4. *Coding*.

Untuk dapat dimengerti oleh mesin, dalam hal ini adalah komputer, maka desain tadi harus diubah bentuknya menjadi bentuk yang dapat dimengerti oleh mesin,

yaitu ke dalam bahasa pemrograman melalui proses *coding*. Tahap ini merupakan implementasi dari tahap design yang secara teknis nantinya dikerjakan oleh *programmer*.

5. *Testing / Verification.*

Sesuatu yang dibuat haruslah diujicobakan. Demikian juga dengan software. Semua fungsi-fungsi software harus diujicobakan, agar *software* bebas dari *error*, dan hasilnya harus benar-benar sesuai dengan kebutuhan yang sudah didefinisikan sebelumnya.

6. *Maintenance.*

Pemeliharaan perangkat lunak, termasuk pengembangan, diperlukan karena perangkat lunak yang dihasilkan tidak selalu demikian. Selama operasi, mungkin masih ada bug kecil yang tidak terdeteksi sebelumnya atau ada fitur tambahan yang tidak ada dalam perangkat lunak. Pengembangan diperlukan ketika perubahan dilakukan di luar perusahaan, seperti mengubah sistem operasi atau perangkat lain.

2.1.7. Pengujian Sistem

a. *Black Box Testing*

Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau Program. Dalam pengujian ini, *tester* menyadari apa yang harus dilakukan oleh program tetapi tidak memiliki pengetahuan tentang bagaimana melakukannya.



Gambar 2.1 Sistem Pengujian

b. *White Box Testing*

White Box Testing untuk menguji suatu aplikasi atau software dengan melihat modul untuk memeriksa dan menganalisis kode *program* ada yang salah atau tidak. Jika modul ini dan telah diproduksi dalam output yang tidak memenuhi persyaratan, kode akan dikompilasi ulang dan diperiksa lagi sampai mencapai apa yang diharapkan.

2.2 Penelitian Terdahulu (*State of The Art*)

Tabel 2.6 Penelitian Terdahulu

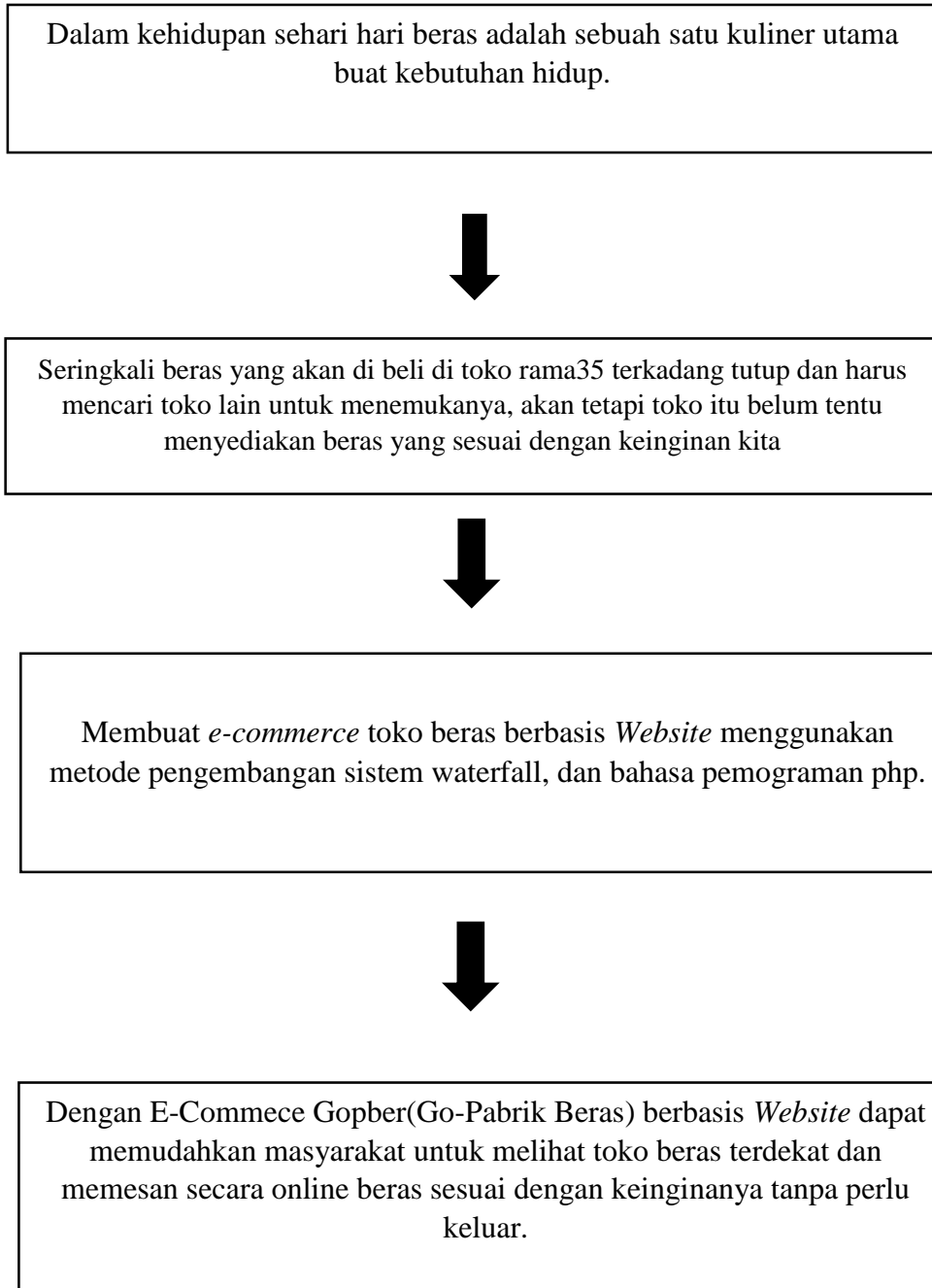
No	Nama Peneliti	Judul Penelitian	Metode	Hasil/Output
1.	Bambang Eko Sumbono·Deni Erlansya dan Rasmila(2020)	Aplikasi <i>E-Commerce</i> Sebagai Media Penjualan Berbasis Web Pada Toko Mau Pempek.	<i>Rational</i> <i>Unified</i> <i>Process(RUP)</i>	Pengembangan sistem informasi ini menggunakan metode <i>Rational Unified Process (RUP)</i> dinilai sangat tepat dalam pembangunan sistem <i>e-commerce</i> ini karena menghasilkan suatu sistem yang sesuai dengan kebutuhan dari user (mempercepat pemesanan).

2.	Putri Sekarudya Puspita Aruningtyas,Putut Haribowo dan Jumi(2019)	Rancang Bangun <i>E-Commerce</i> Berbasis <i>Web Responsive</i> Dan <i>Mobile Android</i> Pada Umkm Aikori Natural Leather Bag Semarang Sebagai Media Promosi Dan <i>Informasi</i> .	<i>Waterfall</i>	Melalui media Promosi berbasis <i>E-Commerce</i> sistem memberikan informasi-informasi produk yang di jual serta transaksi lebih efektif.
3.	Masitah,Abdullah dan Ilyas(2018)	E-Commerce Penjualan Pakaian Pada Lapak Mariati Berbasis Web.	<i>Waterfall</i>	Pada Lapak Mariati ini, konsumen dapat melihat produk sertas info produk,Serta <i>E-commerce</i> penjualan ini dapat menjadi alternatif pemilik agar dapat melihat data hasil laporan penjualan yang akurat.

4.	Andi Ridho Rachman, Beny dan Erick Fernando(2017)	Perancangan <i>E-Commerce</i> Berbasis Website Pada Toko Mirabella Batik Jambi.	<i>Waterfall</i>	sebuah aplikasi <i>e-commerce</i> untuk toko Mirabella Batik Jambi yang dihasilkan dalam penelitian ini dapat memudahkan pelanggan untuk melihat produk secara detail dan melakukan pemesanan tanpa harus datang ke toko hanya dengan mengakses situs web <i>e-commerce</i> toko Mirabella Batik Jambi
5.	Astria Firman, Hans F. Wowor dan Xaverius Najoan(2016)	Sistem Informasi Perpustakaan Online Berbasis Web(E-journal Teknik Elektro dan Komputer.	<i>Waterfall</i>	mencoba membuat sistem informasi yang mempermudah mahasiswa mencari data buku, informasi peminjaman buku dan

				melakukan pengajuan pendaftaran anggota perpustakaan yang bisa dilakukan dengan menggunakan jaringan internet
--	--	--	--	--

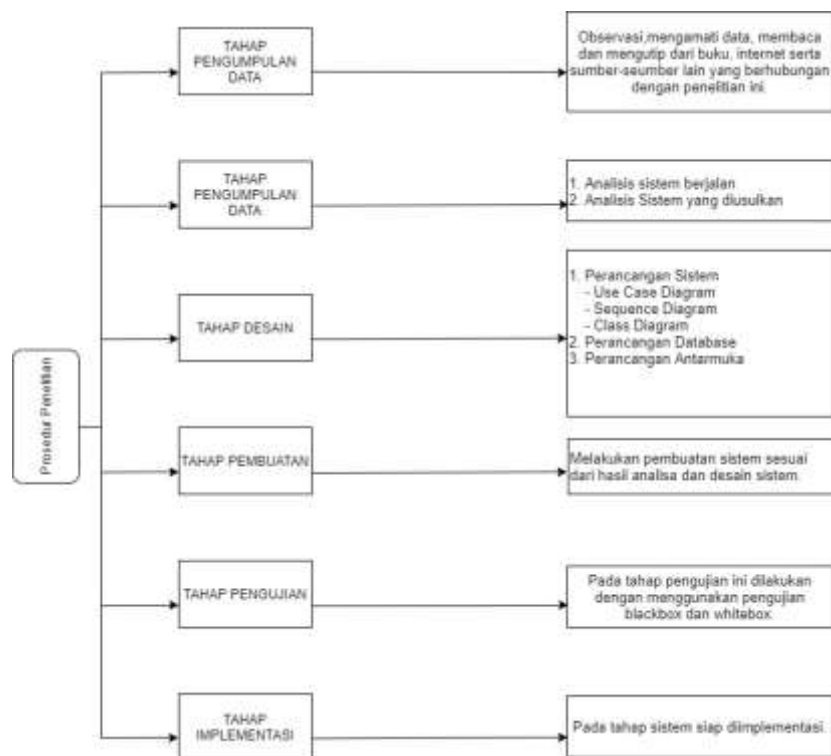
2.1.9 Kerangka Pikir



BAB III METODE PENELITIAN

3.1. Prosedur Penelitian

Tahapan yg dilakukan pada penelitian ini merupakan menjadi berikut :

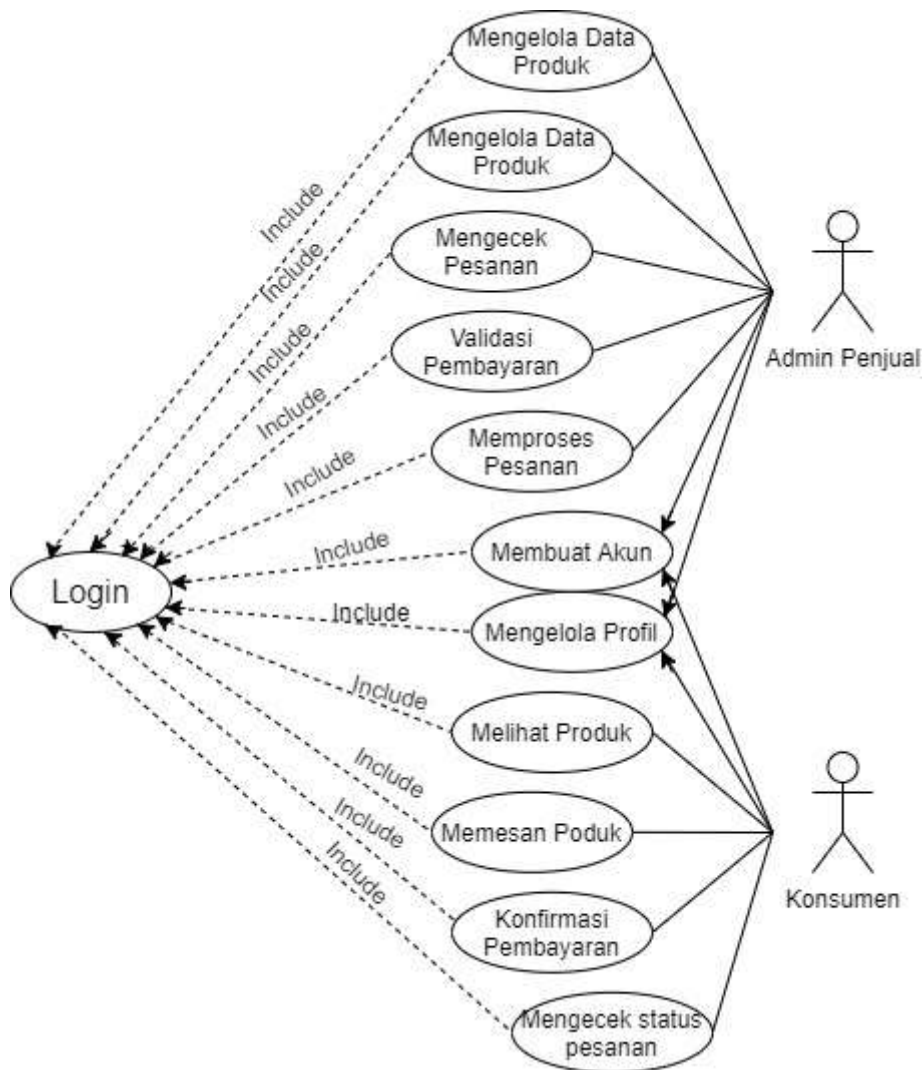


Gambar 3.1 Prosedur Penelitian

3.2. Rancangan Sistem

Perancangan sistem adalah strategi yang digunakan untuk merancang atau membuat perangkat lunak atau program.

3.2.1 Use Case Diagram



Gambar 3. 2 Use Case Diagram Aplikasi

3.2.2 Skenario UseCase

a) Skenario UseCase Login

Use case login

Tabel 3.1 Skenario *Usecase* Login

Tujuan	Melakukan <i>login</i>
Deskripsi	Sistem memungkinkan <i>actor</i> untuk bisa mengakses form aplikasi yang terdapat didalam <i>program</i>
Aktor	Penjual, Konsumen
Skenario utama	
Kondisi awal	aktor mengakses aplikasi
Aksi actor	Reaksi system
1.aktor memilih menu <i>login</i>	sistem merespon dengan menampilkan form <i>login</i> untuk admin penjual, atau konsumen.
Kondisi akhir	Aktor dapat mengakses form yang telah ditetapkan oleh system

b) Skenario Usecase Penjual

Usecase Penjual

Tabel 3.2 Skenario *Usecase* Penjual

Tujuan	Membuat Akun untuk Penjualan Beras
Deskripsi	Sistem ini memungkinkan Penjual untuk melakukan pengelolaan terhadap data-data akun yang ada di dalam program
Aktor	Penjual
Skenario utama	
Kondisi awal	Actor mengakses page menu login dan buat akun
Aksi actor	Reaksi system
aktor memilih menu dashboard	Menampilkan menu dashboard
Kondisi akhir	Tampil form laman utama aplikasi

c) Skenario Usecase Mengelola Profil

Usecase Mengola Profil

Tabel 3.3 Skenario *Usecase* Mengola Profil

Tujuan	Memberikan Informasi Seputar Profil Penjual dan Konsumen
Deskripsi	Sistem ini memungkinkan Penjual dan konsumen untuk melakukan penginputan informasi terkait kegiatan

Aktor	Penjual dan Konsumen
Skenario utama	
Kondisi awal	Aktor mengakses page Laman profil
Aksi actor	Reaksi system
aktor memilih menu Laman Profil	Menu Kegiatan ditampilkan oleh system dan dapat di akses oleh Penjual
Kondisi akhir	Aktor dapat menambahkan informasi kegiatan di laman Aplikasi

d) Skenario Usecase Mengedit Data Produk

Usecase Mengedit Data Produk

Tabel 3.4 Skenario *Usecase Mengedit Data Produk*

Tujuan	Memberikan data terkait Produk yang akan di pasarkan
Deskripsi	Sistem ini memungkinkan admin Penjual untuk melakukan edit data produk terkait produk yang akan dipasarkan
Aktor	Penjual
Skenario utama	
Kondisi awal	Aktor mengakses page edit produk
Aksi actor	Reaksi system
aktor memilih menu edit Toko dan Produk	Menu edit ditampilkan oleh system dan dapat di akses oleh Penjual
Kondisi akhir	Aktor dapat menambahkan Berita di laman Aplikasi

e) Skenario Usecase Mengecek Pesanan

Usecase Mengecek Pesanan

Tabel 3.5 Skenario Usecase Mengecek Pesanan

Tujuan	Agar penjual bisa mengecek pesanan yang di inginkan konsumen
Deskripsi	Sistem ini menampilkan Pesanan yang telah di pilih oleh konsumen
Aktor	Penjual
Skenario utama	
Kondisi awal	Aktor mengakses page cek pesanan
Aksi actor	Reaksi system
aktor memilih menu mengecek pesanan	sistem merespon dengan menampilkan Data konsumen dan produk yang ingin di pesan
Kondisi akhir	Aktor dapat memvalidasi pesanan konsumen

f) Skenario Usecase Validasi Pembayaran

Usecase Validasi Pembayaran

Tabel 3.6 Skenario Usecase Validasi Pembayaran

Tujuan	Penjual mengelola Pembayaran atas pesanan konsumen
Deskripsi	Sistem ini menampilkan sukses pembayaran setelah konsumen membayar pesanan yang telah sampai
Aktor	Penjual
Skenario utama	
Kondisi awal	Aktor mengakses page validasi pembayaran
Aksi actor	Reaksi system
aktor memilih menu Validasi Pembayaran	sistem merespon dengan menampilkan validasi pembayaran
Kondisi akhir	Aktor dapat melihat validasi pembayaran

g) Skenario Usecase Memproses Pesanan

Usecase Memproses Pesanan

Tabel 3.7 Usecase Memproses Pesanan

Tujuan	Penjual memproses pesanan produk yang di pilih oleh konsumen
Deskripsi	Sistem ini menampilkan proses pesanan yang di pilih oleh konsumen
Aktor	Penjual
Skenario utama	
Kondisi awal	Aktor memilih laman pesanan
Aksi actor	Reaksi system
aktor memilih menu Proses	sistem merespon dengan menampilkan Proses Pesanan
Kondisi akhir	Aktor dapat melihat Proses Pesanan

h) Skenario Usecase Informasi Daftar Toko

Usecase Informasi Daftar Toko

Tabel 3.8 Usecase Informasi Daftar Toko

Tujuan	Menampilkan informasi toko yang dibutuhkan oleh konsumen
Deskripsi	Sistem ini memungkinkan konsumen untuk mengakses informasi toko yang telah dipublikasikan oleh Penjual ke

	program
Aktor	Konsumen
Skenario utama	
Kondisi awal	Aktor mengakses page informasi toko
Aksi actor	Reaksi system
aktor memilih menu Informasi Toko	sistem merespon dengan menampilkan informasi toko.
Kondisi akhir	Aktor dapat melihat informasi toko yang dibutuhkan konsumen

i) Skenario Usecase Melihat Produk

Usecase Melihat Produk

Tabel 3.9 Usecase Melihat Produk

Tujuan	Agar konsumen dapat melihat dan memilih produk yang diinginkan
Deskripsi	Sistem ini memungkinkan konsumen untuk melihat produk yang akan dipilih konsumen
Aktor	Konsumen
Skenario utama	
Kondisi awal	Aktor mengakses page Melihat Produk
Aksi actor	Reaksi system
aktor memilih menu Informasi Produk	sistem merespon dengan menampilkan dashboard menu pilihan produk
Kondisi akhir	Aktor dapat melihat produk yang dibutuhkan konsumen

j) Skenario Usecase Memesan Produk

Usecase Memesan Produk

Tabel 3.10 Usecase Melihat Produk

Tujuan	Memesan produk yang diinginkan konsumen
Deskripsi	Sistem ini memungkinkan konsumen untuk memesan produk yang diinginkan
Aktor	Konsumen
Skenario utama	
Kondisi awal	Aktor mengakses page dashboard
Aksi actor	Reaksi system
aktor memilih menu Informasi Produk	sistem merespon dengan menampilkan produk yang ingin di pesan.
Kondisi akhir	Aktor dapat memilih produk yang diinginkan

k) Skenario Usecase Konfirmasi Pembayaran

Usecase Konfirmasi Pembayaran

Tabel

3.11

Tujuan	Konsumen melakukan pembayaran
Deskripsi	Konsumen melakukan pembayaran setelah barang yang dipesan telah tiba
Aktor	Konsumen
Skenario utama	
Kondisi awal	Aktor mengakses konfirmasi pesanan
Aksi actor	Reaksi system
aktor memilih menu Informasi Pembayaan	Sistem merespon dengan menampilkan pembayaran
Kondisi akhir	Aktor melihat status konfirmasi pembayaran

Usecase Konfirmasi Pembayaran

l) Skenario Usecase Mengecek Status Pesanan

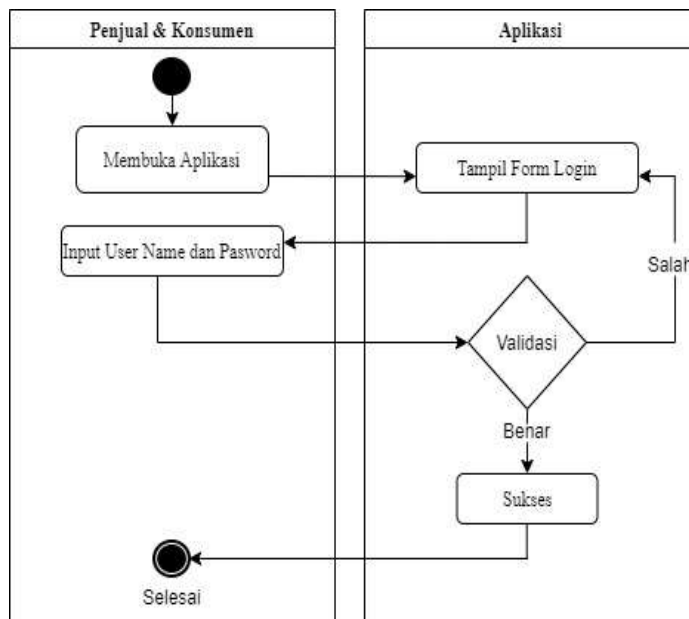
Usecase Mengecek Status Pesanan

Tabel 3.12 Usecase Status Pesanan

Tujuan	Melihat pesanan yang telah di pesan
Deskripsi	Konsumen dapat melihat apakah pesanan yang telah di pesan telah d proses
Aktor	Konsumen
Skenario utama	
Kondisi awal	Aktor mengakses status pesanan
Aksi actor	Reaksi system
aktor memilih menu status Pesanan	Sistem merespon dengan menampilkan status pesanan
Kondisi akhir	Aktor melihat status pesanan

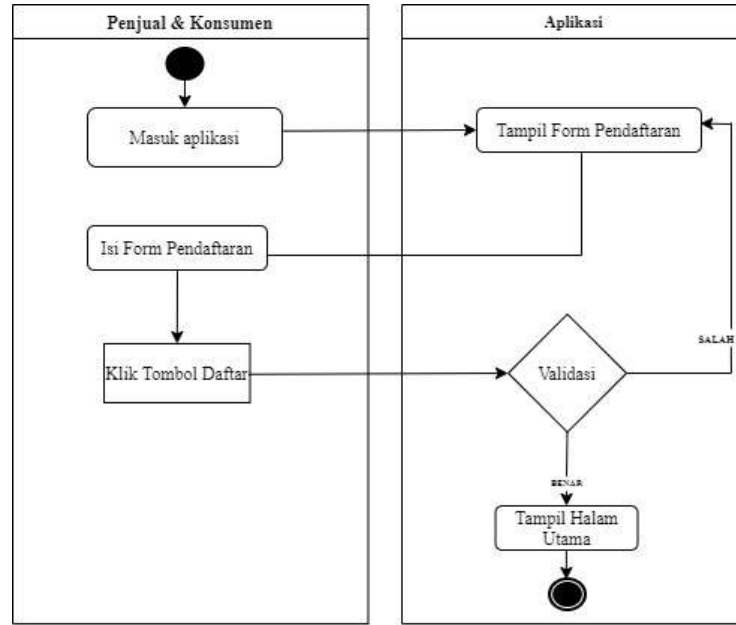
3.3 ActivityDiagram

1. Activity Diagram Login



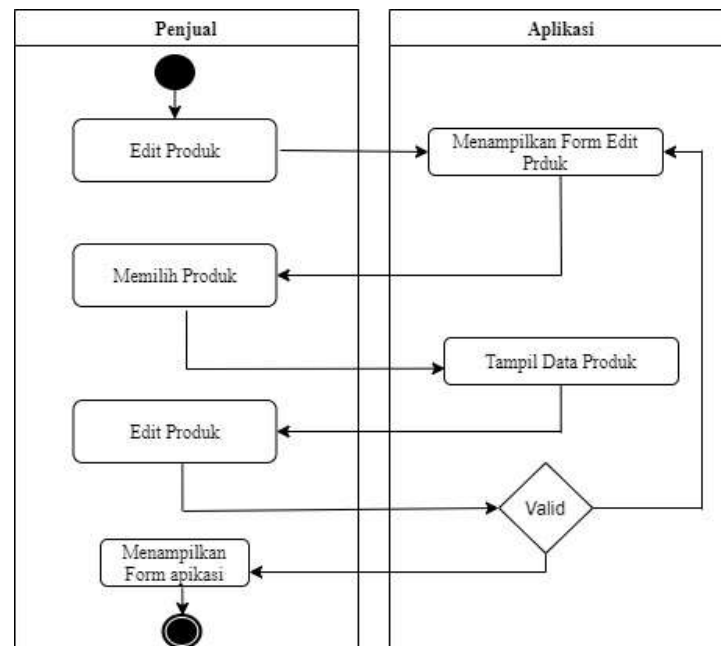
Gambar 3.3 Activity Diagram Login

2. Activity Diagram Daftar



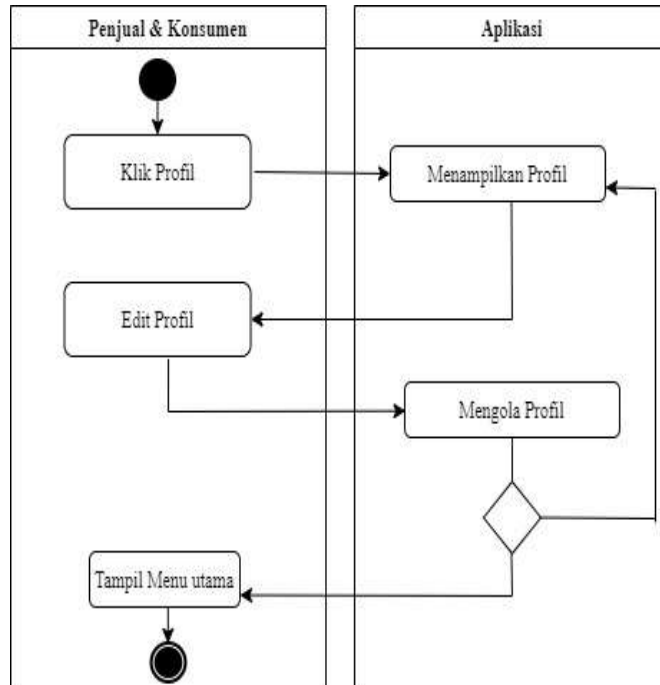
Gambar 3.4 Activity Diagram Daftar

3. Activity Mengola Data Produk



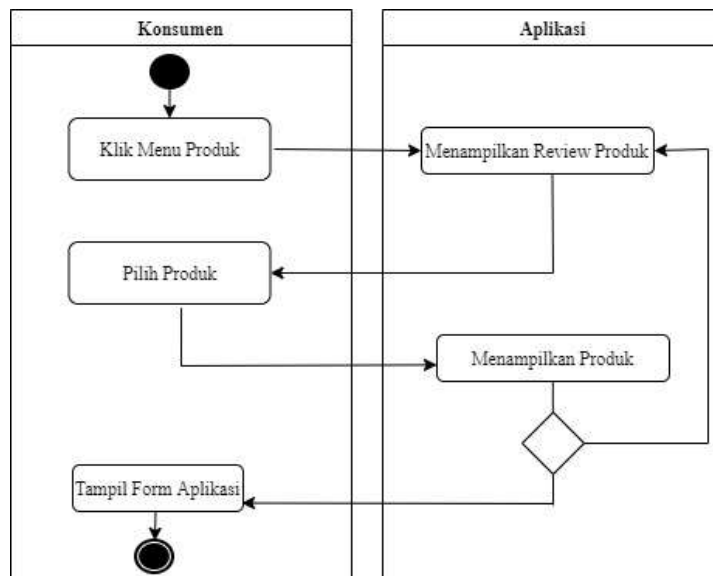
Gambar 3.5 Activity Mengola Data Produk

4. Activity Mengola Profil



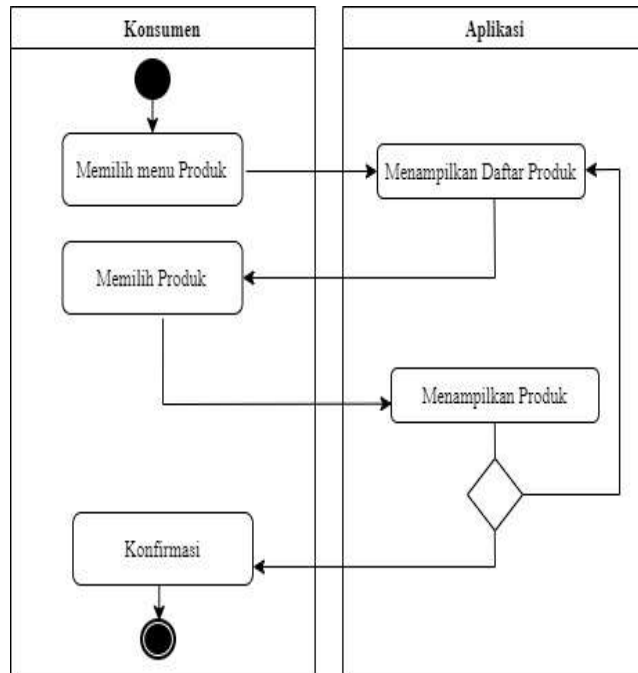
Gambar 3.6 Activity Mengola Profil

5. Activity Diagram Memesan Produk



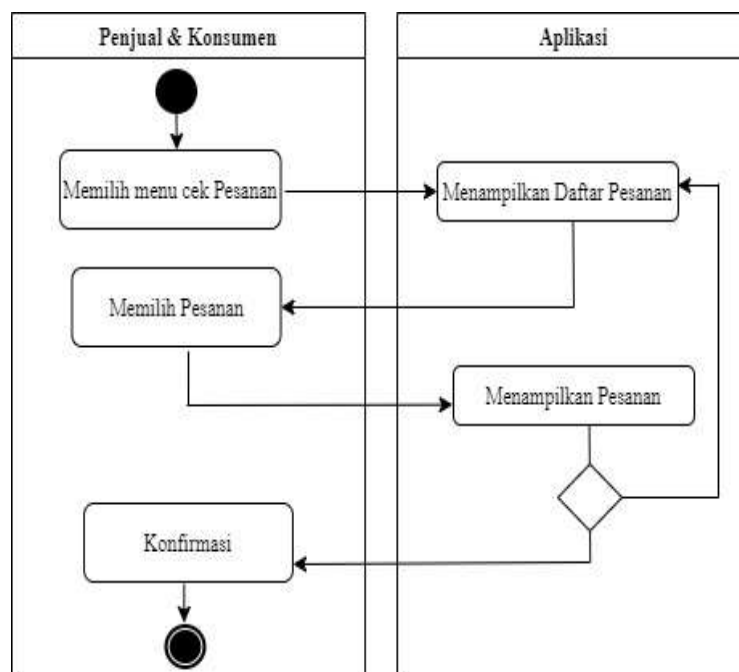
Gambar 3.7 Activity Memesan Produk

6. Activity Melihat Produk



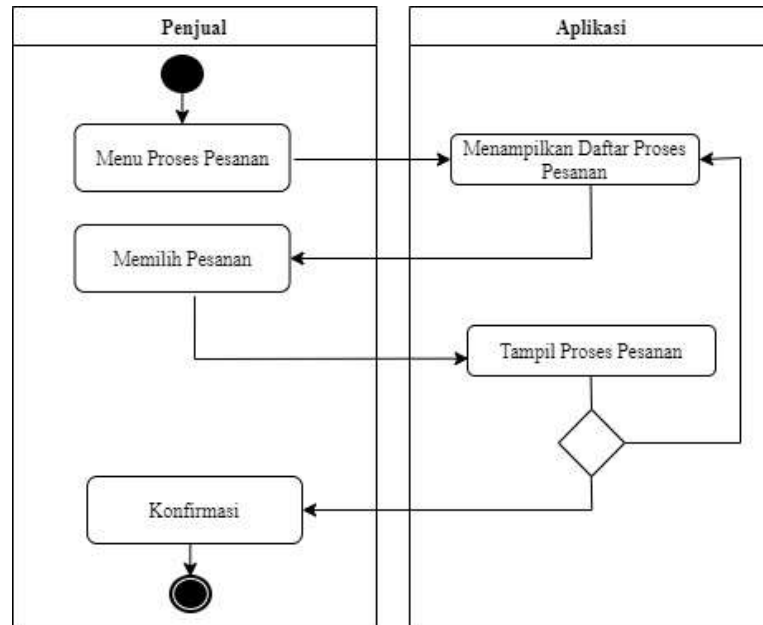
Gambar 3.8 Activity Melihat Produk

7. Activity Mengecek Pesanan



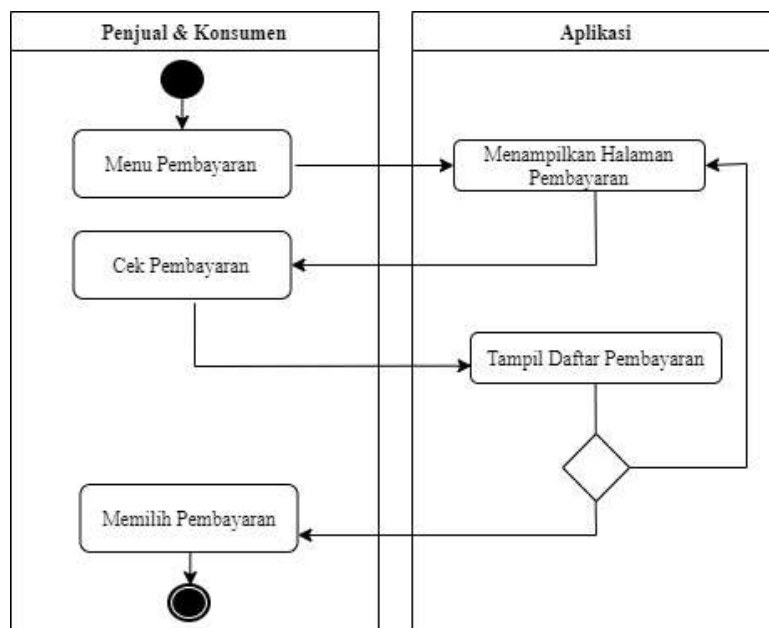
Gambar 3.9 Activity Mengecek Pesanan

8. Activity Memproses Pesanan



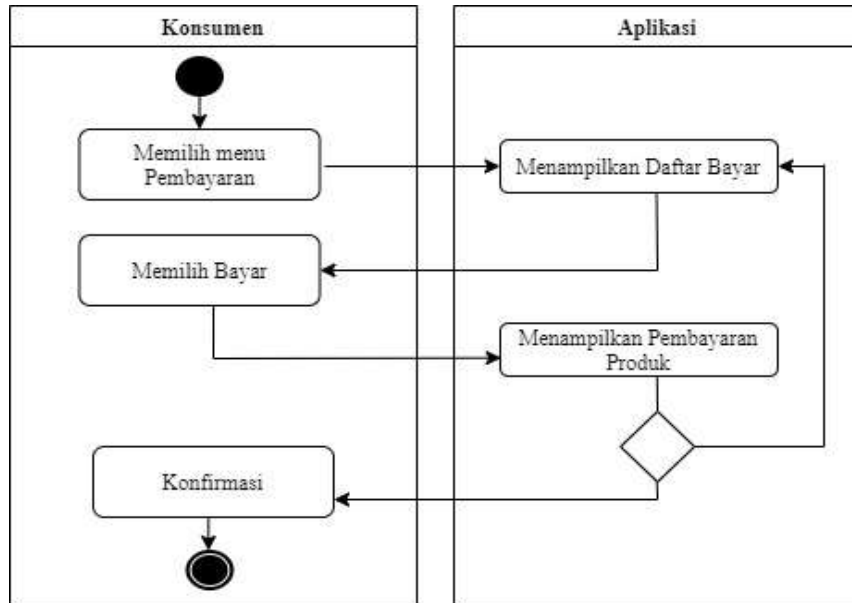
Gambar 3.10 Activity Memproses Pesanan

9. Activity Validasi Pembayaran



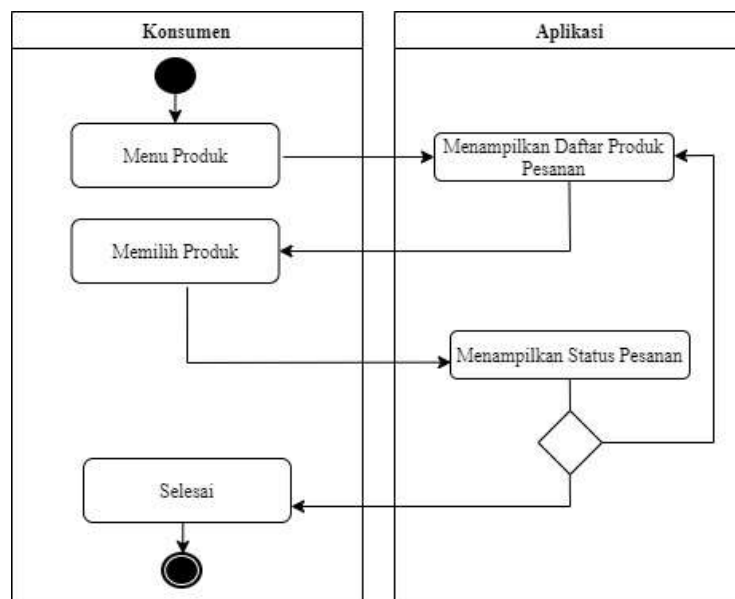
Gambar 3.11 Activity Validasi Pembayaran

10. Activity Konfirmasi Pembayaran



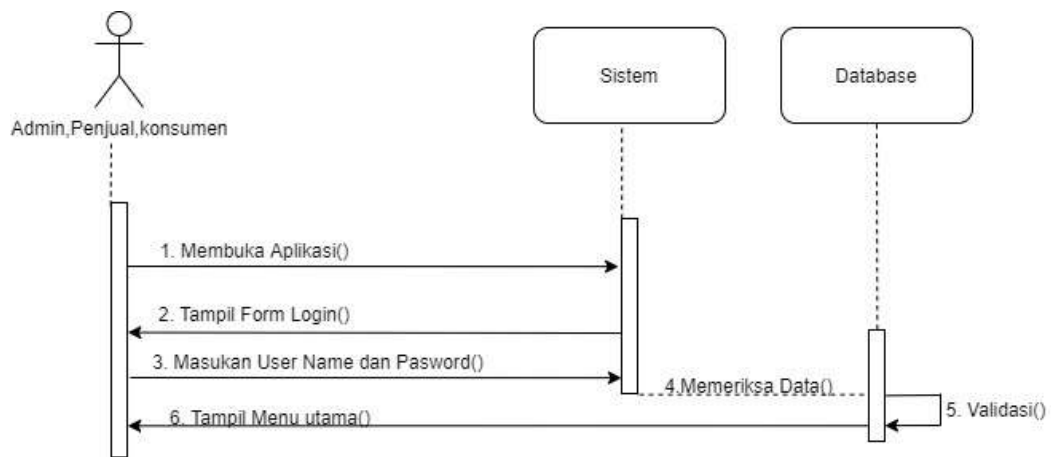
Gambar 3.12 Activity Konfirmasi Pembayaran

11. Activity Mengecek Status Pesanan



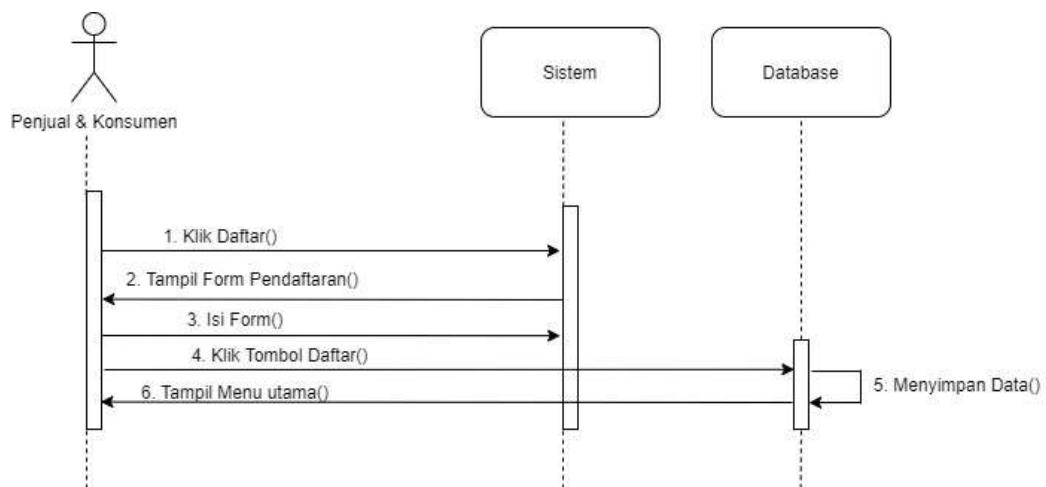
Gambar 3.13 Activity Status Pesanan

3.4 Sequence Diagram



3.4.1 Sequence Diagram Login

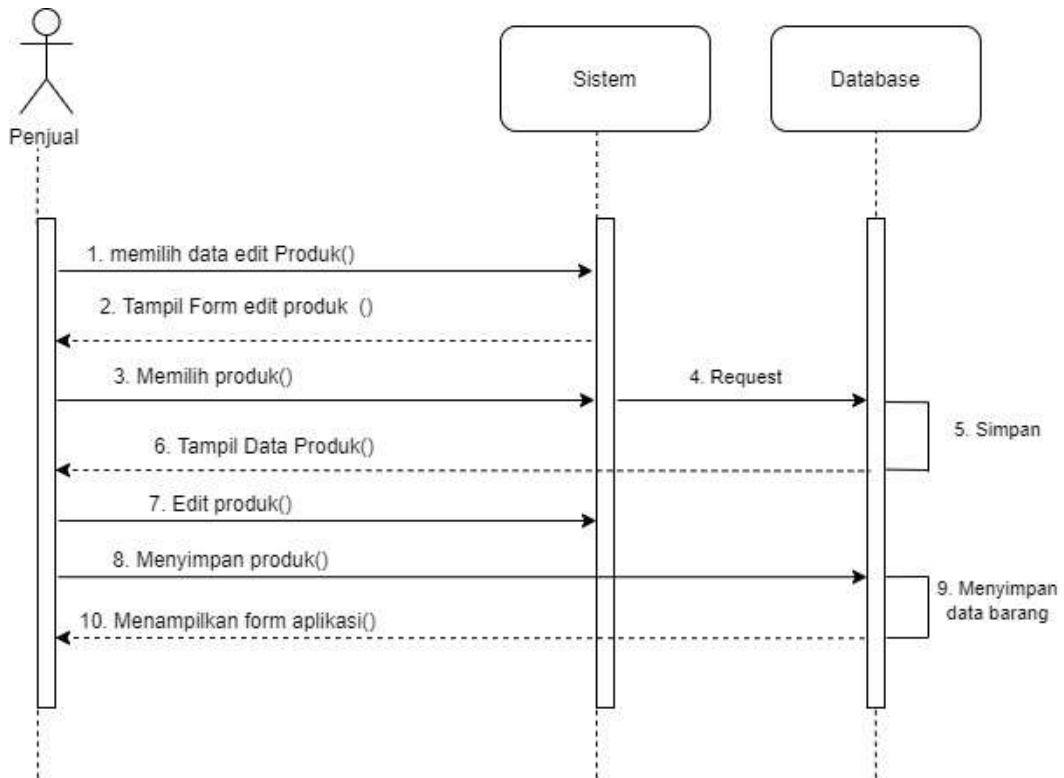
Gambar 3. 14 Sequence Diagram Login



3.4.2 Sequence Membuat Akun

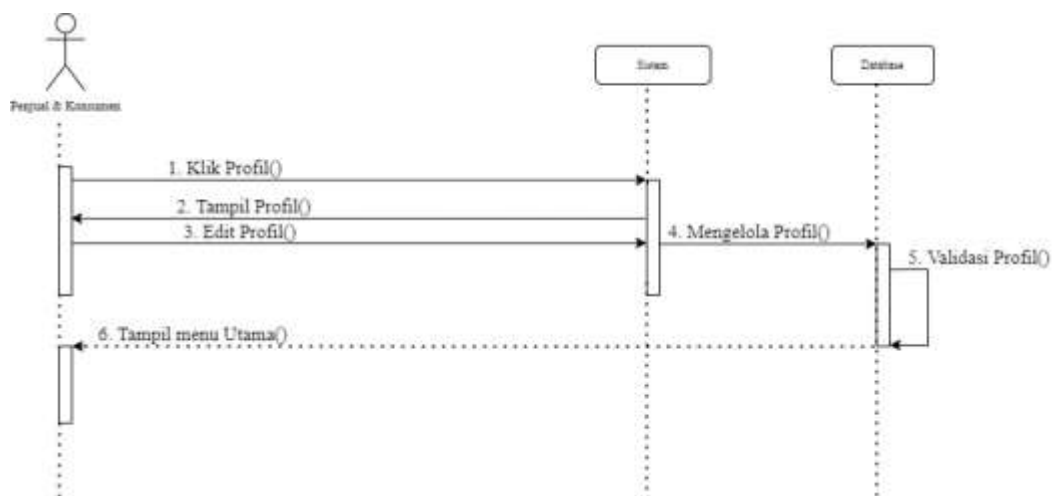
Gambar 3. 15 Sequence Digram Membuat Akun

3.4.3 Sequence Mengelola data produk



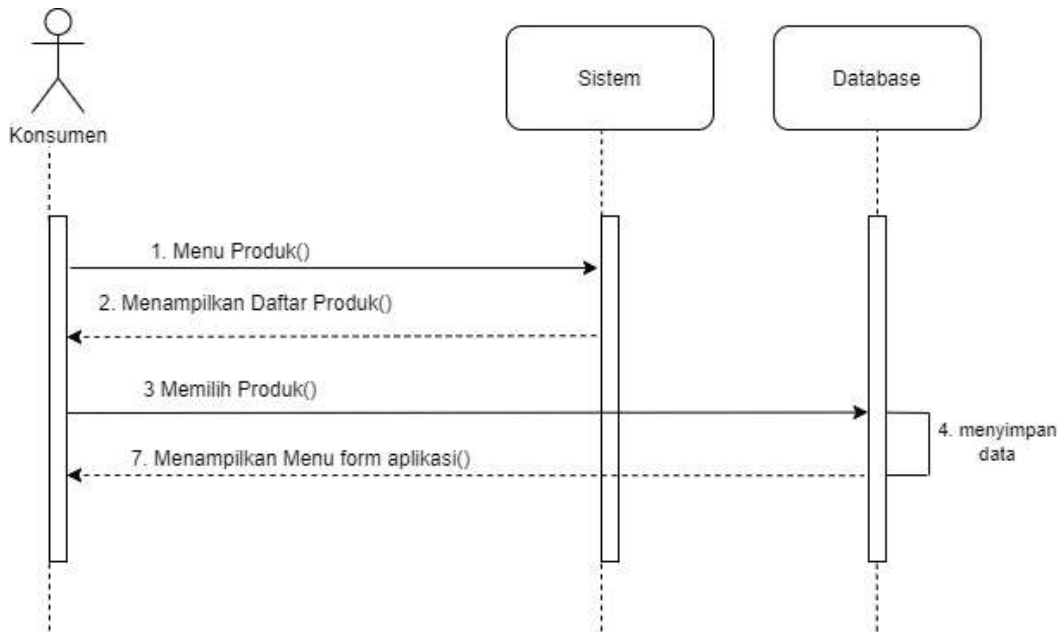
Gambar 3. 16 Sequence Digram Mengola Data Produk

3.4.4 Sequence Mengola Profil



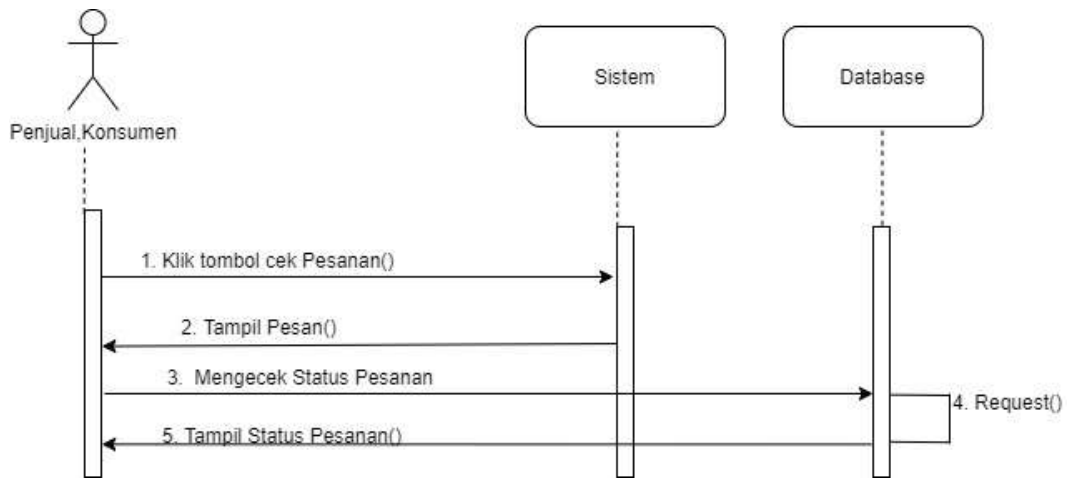
Gambar 3. 17 Sequence Digram Mengola Profil

3.4.5 Sequence Memesan produk



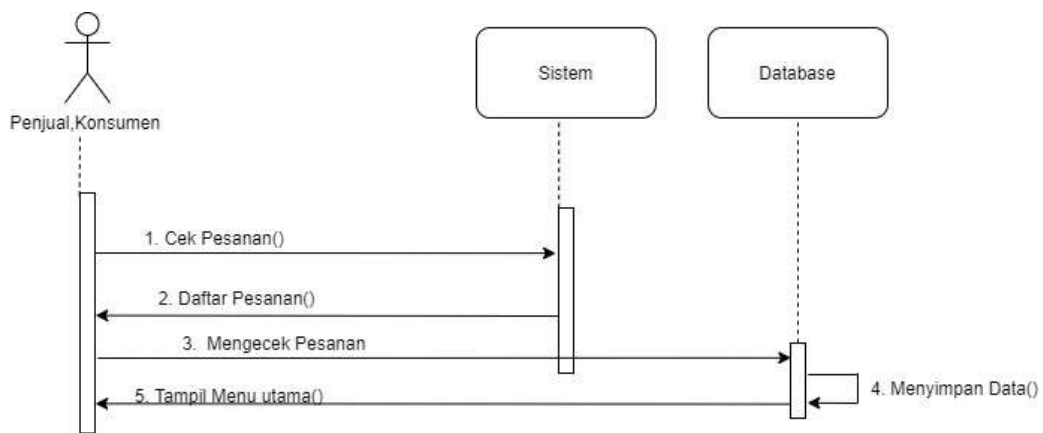
Gambar 3.18 Sequence Digram Memesan Produk

3.4.6 Sequence Melihat produk



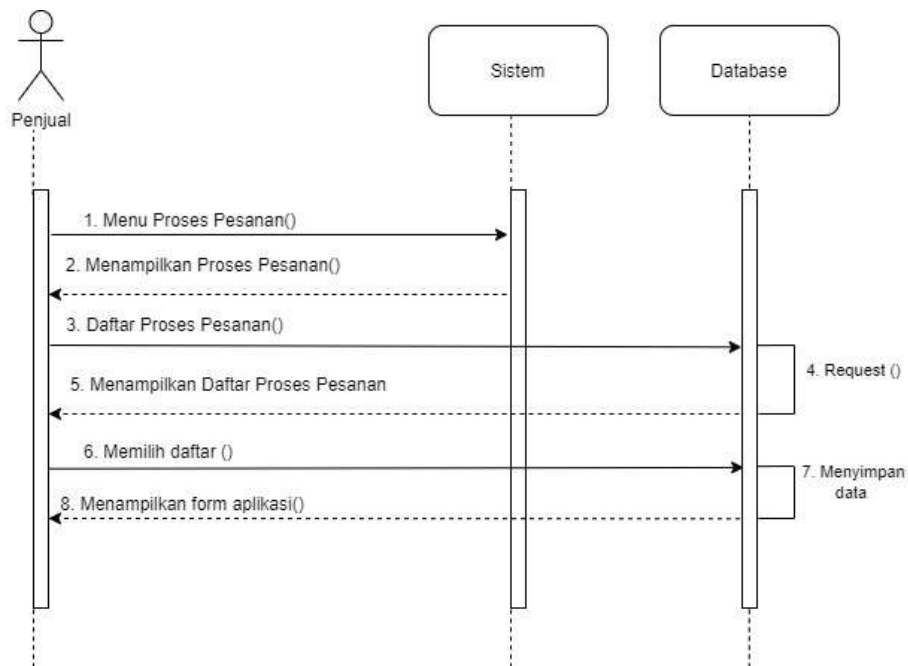
Gambar 3.19 Sequence Mengecek Pesanan

3.4.7 Sequence Mengecek Pesanan



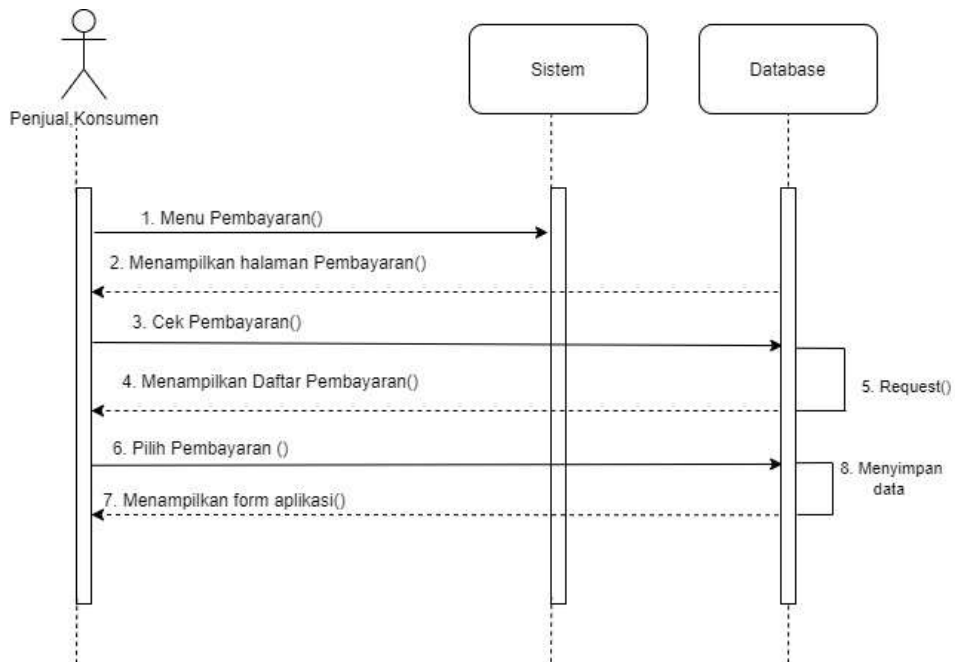
Gambar 3.20 Sequence Mengecek Pesanan

3.4.8 Sequence Memproses Pesanan



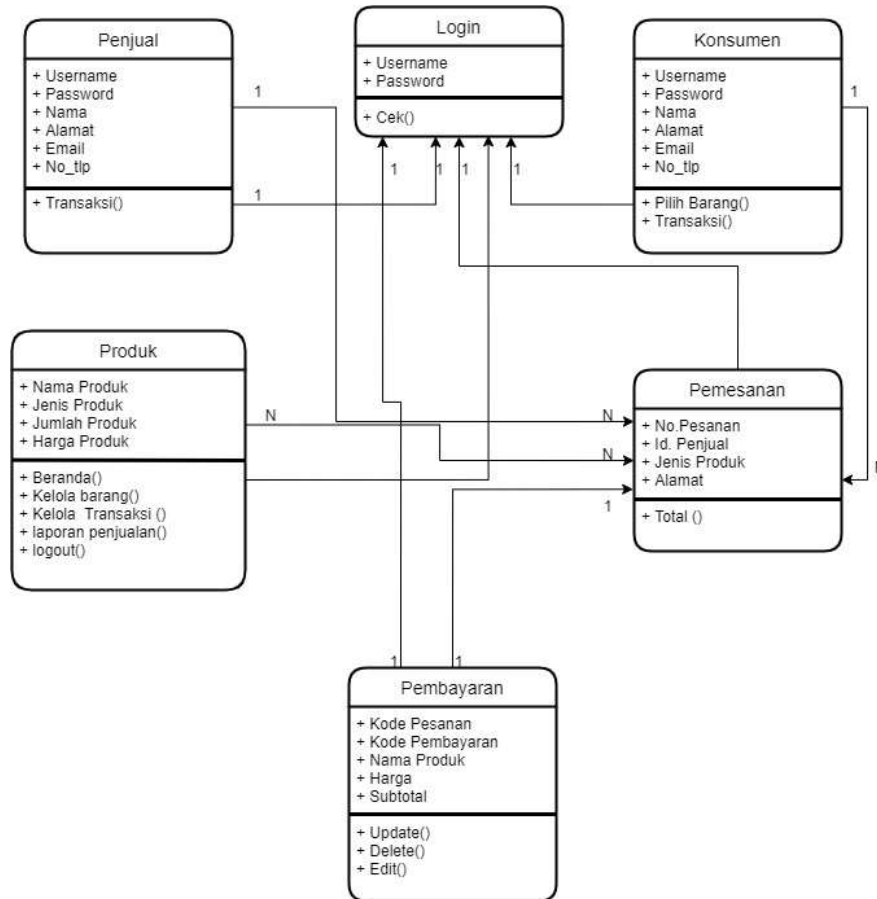
Gambar 3.21 Sequence Memproses Pesanan

3.4.9 Sequence Validasi Pembayaran



Gambar 3. 22 Sequence Mengecek Validasi Pembayaran

3.5 Class Diagram



Gambar 3.23 Class Diagram

3.6 Perancangan Database

a. *Penjual*

NO	Kolom	Tipe	Keterangan
1.	User_id	Int(11)	
2.	Nama	Text	
3.	Email	Text	
4.	Password	Text	
5.	No_Hp	Int(12)	

b. *Konsumen*

NO	Kolom	Tipe	Keterangan
1.	User_id	Int(11)	
2.	Nama	Text	
3.	Email	Text	
4.	Password	Text	
5.	No_Hp	Int(12)	

c. *Produk*

No	Kolom	Tipe	Keterangan
1.	Barang_id	Int(11)	
2.	Nama_Barang	Varchar(40)	
3.	Jumlah_Barang	Int(11)	
4.	Harga_Barang	Int(11)	
5.	Jenis_Barang	Varchar(20)	
6.	Satuan	Text	
7.	Gambar	Text	
8.	Keterangan	Text	

d. Pemesanan

No	Kolom	Tipe	Keterangan
1.	No.Pesanan	Int(11)	
2.	Id_Penjual	Int(40)	
3.	Jenis Produk	Varchar(11)	
4.	Alamat	Text(25)	

e. Pembayaran

No	Kolom	Tipe	Keterangan
1.	Kode Pesanan	Int(11)	
2.	Kode Pembayaran	Int(40)	
3.	Nama Produk	Text(11)	
4.	Harga Barang	Int(11)	
5.	Subtotal	Varchar(20)	

3.7 Waktu Dan Tempat Penelitian

Penelitian dilaksanakan di Toko Rama35, Kota Makassar, Sulawesi Selatan 90231. Waktu penelitian dilaksanakan mulai bulan Juni 2021 – Agustus 2021.

3.8 Alat dan Bahan Penelitian

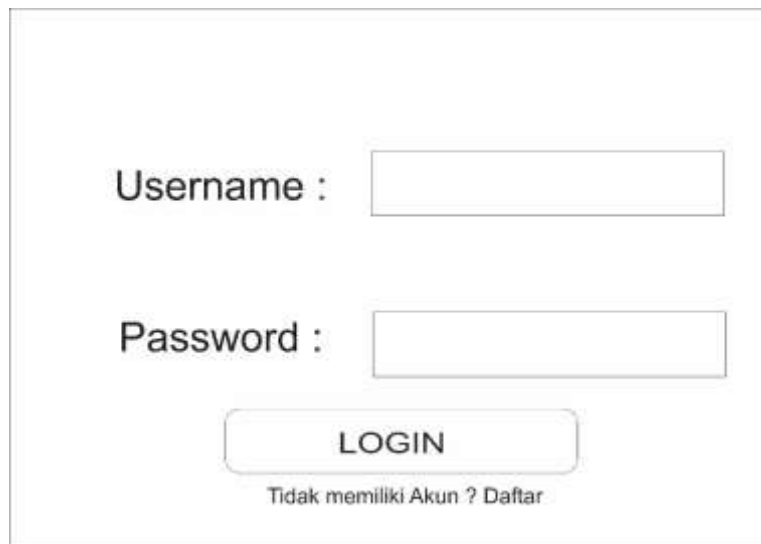
Alat dan bahan yang digunakan untuk merancang dan membuat sistem adalah sebagai berikut:

3,9 Perangkat Keras dan Perangkat Lunak

Perangkat Keras	Perangkat Lunak
1. Laptop Intel(R) Core(TM) i5- 5200U processor 2.20Ghz	1. Sistem Operasi Windows 10
2. RAM 8 GB	2. Sublime Text Editor 3
	3. Web Browser Google Chrome

3.10 Perancangan Antarmuka

a) Halaman Login



The image shows a login interface with the following elements:

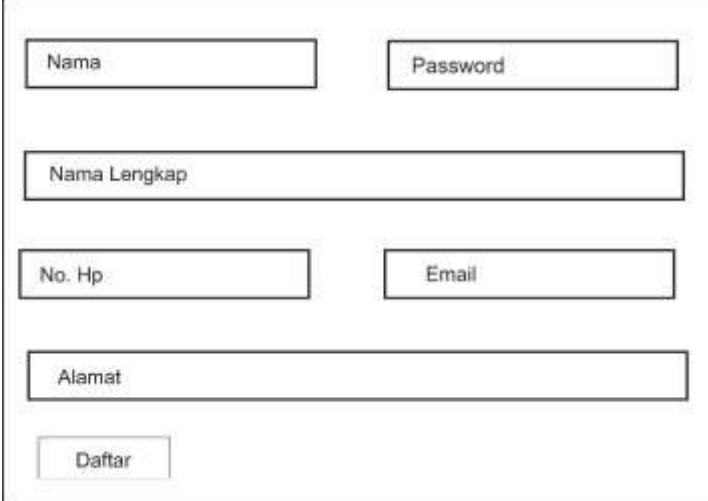
- A label "Username :" followed by a text input field.
- A label "Password :" followed by a text input field.
- A rounded rectangular button labeled "LOGIN".
- A link below the button that reads "Tidak memiliki Akun ? Daftar".

Gambar 3.24 Halaman *Login*

Pada gambar 3.24 menampilkan sebelum masuk ke hidangan primer pelaksanaan Aktor wajib melakukan login terlebih dahulu menggunakan akun yang telah dibuat. Sedangkan jika *Aktor* belum mempunyai akun, *Aktor* harus mendaftar terlebih dahulu..Pada tahap ini, jika *Aktor* telah

mempunyai akun, *Aktor* harus memasukkan *email* dan *password* untuk selanjutnya *login* ke aplikasi.

b) Halaman Daftar



The image shows a registration form with the following fields and a button:

- Nama
- Password
- Nama Lengkap
- No. Hp
- Email
- Alamat
-

Gambar 3.25 Halaman *Daftar*

Pada gambar 3.25 dari halaman *login*, jika *Aktor* yang belum mempunyai akun dan ingin membuat akun, akan diarahkan ke halaman daftar agar dapat *login* ke aplikasi. Pada tahap ini, data yang diperlukan ketika ingin membuat akun adalah nama, telepon (No.Hp), *email*, dan *password*. Setelah data di *input*, terdapat tombol “lanjut” untuk membuat akun dan data yang di *input* akan masuk ke *database* agar *Aktor* dapat *login* ke aplikasi tersebut.

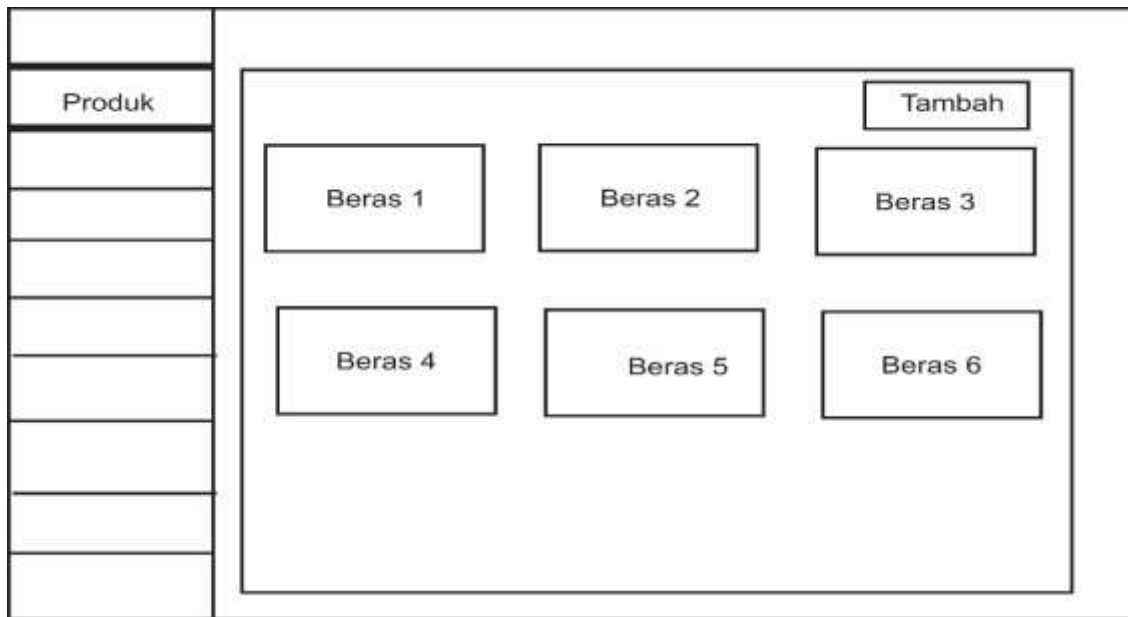
c) Penjual Mengola Profil

Dahboard	<div style="border: 1px solid black; padding: 10px;"><input type="text" value="Nama Toko"/> <input type="text" value="No.Hp"/> <input type="text" value="Email"/> <input type="text" value="Alamat"/> <input type="text" value="Tagline"/> <input type="text" value="Deskripsi"/></div>
Kategori Produk	
Produk	
Pesanan	
Kupon	
Pembayaran	
Pelanggan	
Review Pelanggan	
Kontak	

Gambar 3.26 Mengola Profil

Pada gambar 3.26 jika *Aktor* yang ingin mengedit atau mengola profil data pribadi mereka maka akan diarahkan ke halaman Ubah profil agar dapat mengedit data profil pribadi aktor. Pada tahap ini, data yang ingin diedit adalah nama,username,jenis kelamin,tanggal lahir telepon(No.Hp),*email*,dan ubah *password*. Setelah data di *input*, terdapat tombol “lanjut” untuk membuat akun dan data yang di *input* akan masuk ke *database* tersebut.

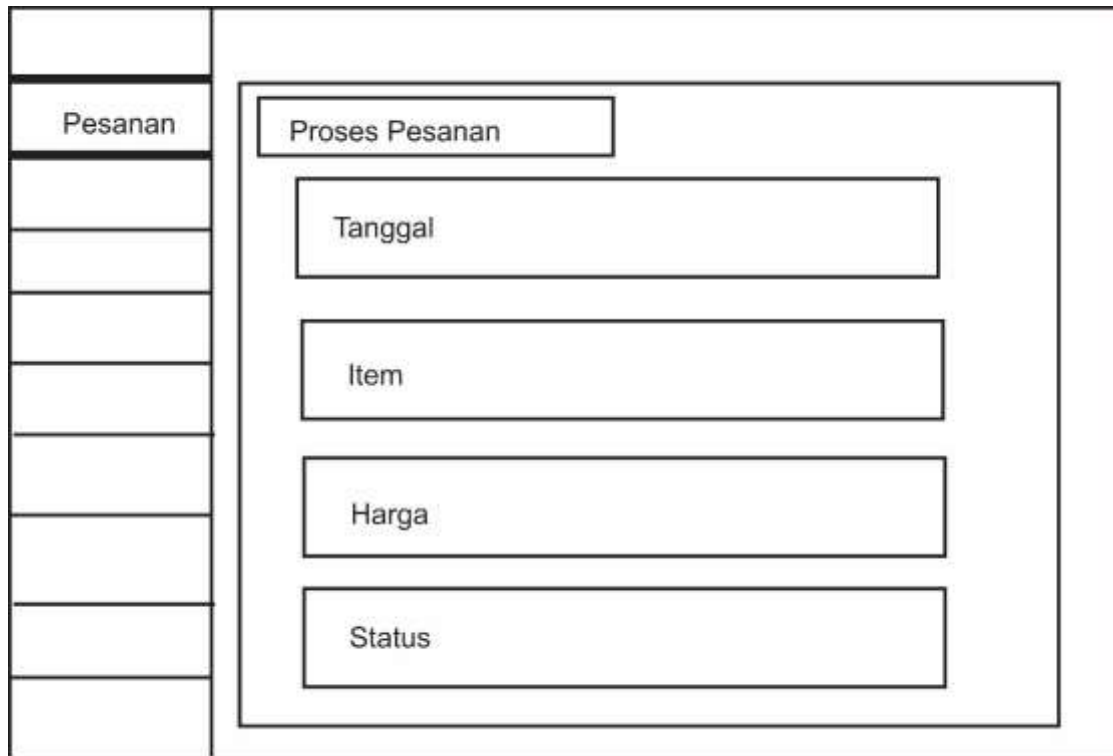
d) Penjual Kelola Data Produk



Gambar 3.27 Kelola Data Produk

Pada gambar 3.27 menunjukkan Penjual masuk ke Sistem aplikasi kelola data produk dimana penjual melakukan *input data produk* yang akan dijual untuk konsumen, tampilan ini pula memberikan penjual untuk memeriksa data produk yang tidak ada di tokonya serta produk-produk yang ingin dipasarkan bisa sesuai dengan kategori dan harga konsumen,

e) Penjual Proses Pesanan



Gambar 3.28 Proses Pesanan

Pada gambar 3.28 menunjukkan Penjual masuk ke aplikasi Proses Pesanan dimana penjual melihat proses pesanan yang akan dijual untuk konsumen, tampilan ini pula memberikan penjual informasi apakah produk sudah dalam proses pengemasan untuk konsumen atau belum.

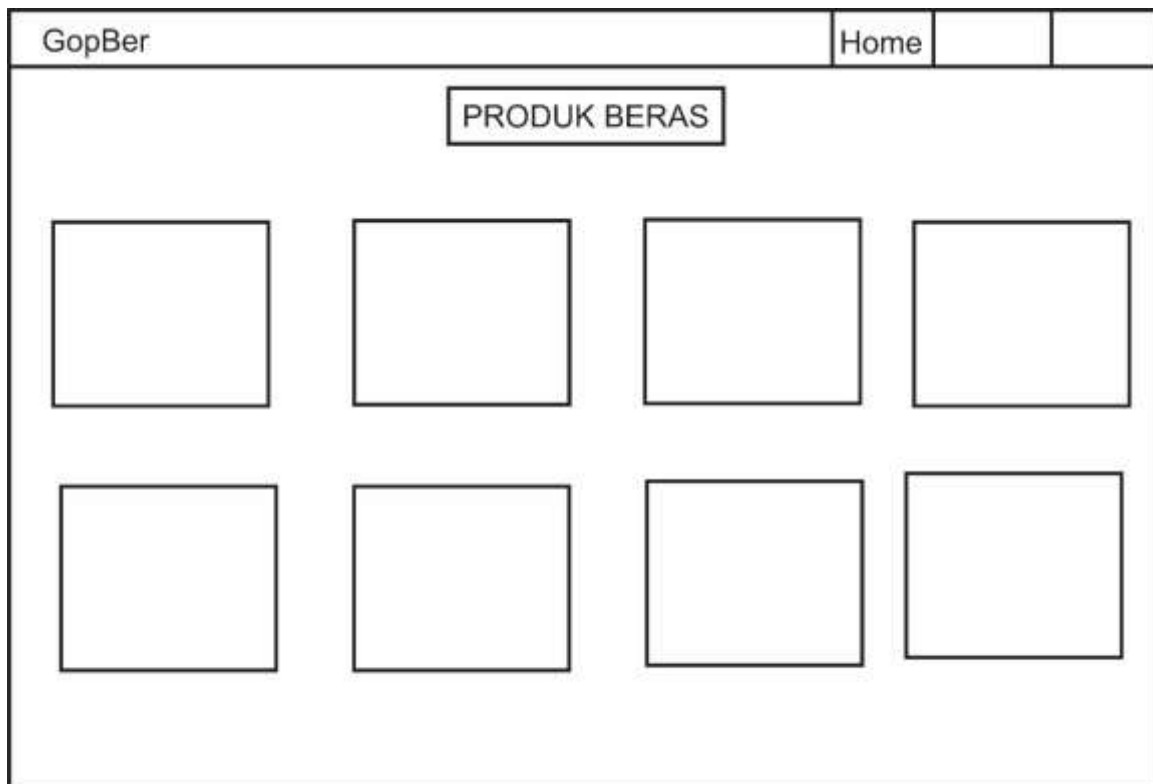
g) Penjual Validasi Pembayaran

Pembayaran	<div style="border: 1px solid black; padding: 5px; margin-bottom: 10px;">Kelola Pembayaran</div> <table border="1"> <thead> <tr> <th>ID</th> <th>Pembayaran Order</th> <th>Pelanggan</th> <th>Tanggal</th> <th>Jumlah</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	ID	Pembayaran Order	Pelanggan	Tanggal	Jumlah	Status																																										
ID	Pembayaran Order	Pelanggan	Tanggal	Jumlah	Status																																												

Gambar 3.30 Validasi Pembayaran

Pada gambar 3.30 menunjukkan Penjual masuk ke Sistem aplikasi validasi pembayaran dimana tampilannya menunjukkan id,pembayaran order,pelanggan tanggal jumlah,dan status yang di pesan oleh konsumen.

h) Konsumen Melihat Produk



Gambar 3.31 Melihat Produk

Pada gambar 3.32 menunjukkan konsumen berada pada tampilan ingin memesan suatu produk, dalam tampilan ini aplikasi menampilkan produk-produk terlaris yang sering di pesan dan produk beras lainnya.

i) Konsumen Memesan Produk

GopBer			Keranjang	
Gambar	Produk	Harga	Kuantitas	Sub Total
				Checkout

Gambar 3.33 Melihat Produk

Pada gambar 3.33 menunjukkan konsumen berada pada tampilan memesan suatu produk,dalam,tampilan ini aplikasi menampilkan produk yang telah di pesan beserta jumlah,biaya krim dan totalnya.

j) **Konsumen Mengecek Status Pesanan**

Order Saya	<table border="1"><tr><td colspan="2">Order</td></tr><tr><td>Data Order</td><td></td></tr><tr><td>Nomor</td><td></td></tr><tr><td>Tanggal</td><td></td></tr><tr><td>Item</td><td></td></tr><tr><td>Harga</td><td></td></tr><tr><td>Metode Pembayaran</td><td></td></tr><tr><td>Status</td><td></td></tr></table>	Order		Data Order		Nomor		Tanggal		Item		Harga		Metode Pembayaran		Status	
Order																	
Data Order																	
Nomor																	
Tanggal																	
Item																	
Harga																	
Metode Pembayaran																	
Status																	

Gambar 3.34 Melihat Produk

Pada gambar 3.34 menunjukkan konsumen berada pada tampilan memesan suatu produk,dalam,tampilan ini aplikasi menampilkan produk yang telah di pesan beserta jumlah,biaya krim dan totalnya.

k) Konsumen Konfirmasi pembayaran

	Order	
	Tanggal	
Pembayaran	Jumlah Transfer	
	Transfer dari	
	Transfer ke	
	Status	

Gambar 3.35 Konfirmasi Pembayaran

Pada gambar 3.35 menunjukkan konsumen berada pada tampilan konfirmasi pembayaran di mana konsumen setelah memesan suatu produk, maka konsumen akan masuk dalam, tampilan ini dimana produk yang telah di pesan akan muncul konfirmasi pembayarannya.

3.11 Metode Pengumpulan Data

Teknik pengumpulan data adalah teknik yang digunakan oleh peneliti untuk mengumpulkan data yang berkaitan dengan masalah penelitian yang sedang mereka hadapi. Pengumpulan data dilakukan untuk mendapatkan informasi yang diperlukan agar mendapat tujuan penelitian. Di dalam penelitian ini, penulis mengumpulkan data melalui: observasi, observasi data, membaca dan kutipan dari internet dan sumber lain yang berhubungan dengan penelitian ini.

3.12. Tahap Analisis Data/Pengujian Sistem

Metode pengujian yang digunakan pada penelitian ini yaitu metode *Black Box Testing* dan *White Box Testing*. Pengujian *Black-Box* berfokus pada persyaratan fungsional dan suatu sistem. *White box testing* adalah pengujian yang didasarkan pada pengecekan terhadap detail perancangan, menggunakan struktur kontrol dari desain program secara procedural untuk membagi pengujian ke dalam beberapa kasus pengujian.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil

Berdasarkan penelitian yang telah dilakukan, berikut adalah hasil yang diperoleh dari sistem yang telah dirancang ditampilkan. Pengguna yang berhak mengakses sistem ini yaitu penjual dan pelanggan.

4.1.1 Tampil halaman utama



Gambar 4.1 Tampilan halaman utama Website

Gambar 4.1 menunjukkan tampilan beranda pada *Website* Gobper. Dimana halaman ini merupakan halaman awal yang tampil ketika *User* mengakses *Website* tersebut. Pada halaman ini terdapat beberapa konten fitur seperti Belanja sekarang home shop tentang kami kontak akun dan keranjang, script dari gambar diatas :

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<section id="home-section" class="hero">
  <div class="home-slider owl-carousel">
    <div class="slider-item" style="background-image: url(<?php echo get_theme_uri('images/beras1.jpg'); ?>);">
      <div class="overlay"></div>
      <div class="container">
        <div class="row slider-text justify-content-center align-items-center" data-scrollax-parent="true">

          <div class="col-md-12 ftco-animate text-center">
            <h1 class="mb-2">Kami Menjual Hanya Beras Yang Terbaik</h1>
            <h2 class="subheading mb-4">Beras Dengan Kualitas Terbaik</h2>
            <p><a href="#products" class="btn btn-primary">Belanja Sekarang</a></p>
          </div>

        </div>
      </div>
    </div>

    <div class="slider-item" style="background-image: url(<?php echo get_theme_uri('images/beras2.jpg'); ?>);">
      <div class="overlay"></div>
      <div class="container">
        <div class="row slider-text justify-content-center align-items-center" data-scrollax-parent="true">

          <div class="col-sm-12 ftco-animate text-center">
            <h1 class="mb-2">100% Beras Murni dan Sehat</h1>
            <h2 class="subheading mb-4">Beras Dengan Kualitas Terbaik</h2>
            <p><a href="#products" class="btn btn-primary">Belanja Sekarang</a></p>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>
<section class="ftco-section" id="products">
  <div class="container">
    <div class="row no-gutters ftco-services">
      <div class="col-md-3 text-center d-flex align-self-stretch ftco-animate">
        <div class="media block-6 services mb-md-0 mb-4">
          <div class="icon bg-color-1 active d-flex justify-content-center align-items-center mb-2">
            <span class="flaticon-shipped"></span>
          </div>
          <div class="media-body">
            <h3 class="heading">Gratis Ongkir</h3>
            <span>Belanja minimal Rp <?php echo format_rupiah(get_settings('min_shop_to_free_shipping_cost')); ?></span>
          </div>
        </div>
      </div>
      <div class="col-md-3 text-center d-flex align-self-stretch ftco-animate">
        <div class="media block-6 services mb-md-0 mb-4">
          <div class="icon bg-color-2 d-flex justify-content-center align-items-center mb-2">
            <span class="flaticon-diet"></span>
          </div>
          <div class="media-body">
            <h3 class="heading">Selalu Baru</h3>
            <span>Dengan Kualitas Terbaik</span>
          </div>
        </div>
      </div>
    </div>
  </div>
</section>

```

Gambar 4.2 Script Tampilan halaman Utama Website

4.1.2 Tampilan Penjual/Penjual

4.1.2.1 Tampilan *Login*



Gambar 4.3 Tampilan *login*

Gambar 4.3 menunjukkan tampilan login untuk penjual dan pelanggan yang menjadi gerbang awal sebelum melakukan pengolahan terhadap *Website*. berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Login_model extends CI_Model {
    protected $username;
    protected $password;

    public function __construct()
    {
        parent::__construct();
    }

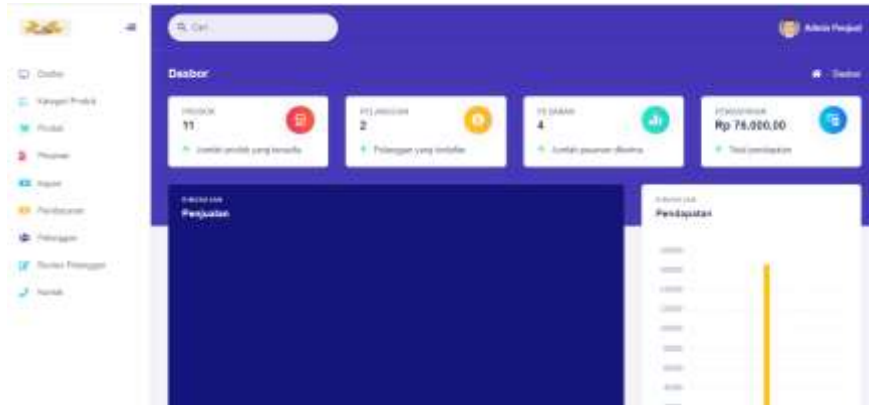
    public function login($username = "", $password = "")
    {
        $this->username = $username;
        $this->password = $password;
    }

    public function is_user_exist()
    {
        $username = $this->username;

        $check = $this->db
```

Gambar 4.4 Script *login*

4.1.2.2 Tampilan Dashboard



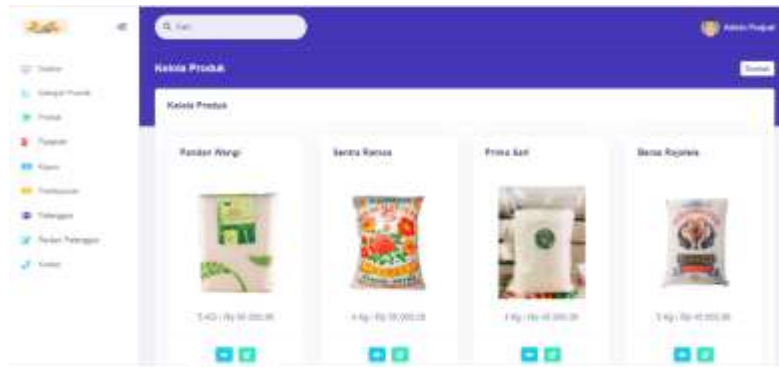
Gambar 4.5 Tampilan Dashboard Penjual

Gambar 4.5 menunjukkan tampilan awal dashboar dari halaman admin ketika admin/penjual melakukan login pada aplikasi. Disini admin/penjual dapat melihat informasi berupa tampilan seperti produk, pelanggan, pesanan serta pendapatan.

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<!-- Header -->
<div class="header bg-primary pb-6">
  <div class="container-fluid">
    <div class="header-body">
      <div class="row align-items-center py-4">
        <div class="col-lg-6 col-7">
          <h2 class="h2 text-white d-inline-block mb-0">Dasbor</h2>
        </div>
        <div class="col-lg-6 col-5 text-right">
          <nav aria-label="breadcrumb" class="d-none d-md-inline-block ml-md-4">
            <ol class="breadcrumb breadcrumb-links breadcrumb-dark">
              <li class="breadcrumb-item"><a href="#"><i class="fas fa-home"></i></a></li>
              <li class="breadcrumb-item active" aria-current="page">Dasbor</li>
            </ol>
          </nav>
        </div>
      </div>
    </div>
  </div>
<!-- Card stats -->
<div class="row">
  <div class="col-xl-3 col-md-6">
```

Gambar 4.6 Dashboar Admin/Penjual

4.1.2.3 Tampilan Kelola Produk



Gambar 4.7 Tampilan *Kelola Produk*

Gambar 4.7 menunjukkan tampilan yang ditampilkan oleh sistem apabila admin/penjual ingin menambah sebuah produk beras ke dalam *Website* aplikasi Gobper. Disini admin/penjual bisa mengelola informasi berupa menambah produk serta mengedit salah satu dari produk beras tersebut sesuai informasi produk itu sendiri. berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Products extends CI_Controller {
    public function __construct()
    {
        parent::__construct();

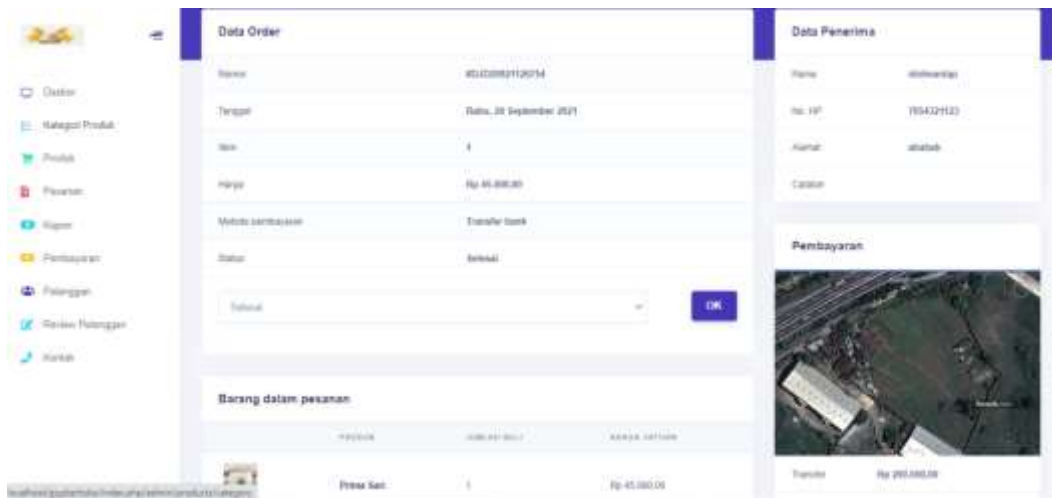
        verify_session('admin');

        $this->load->model(array(
            'product_model' => 'product',
            'order_model' => 'order'
        ));
        $this->load->library('form_validation');
    }

    public function index()
```

Gambar 4.8 Tampilan *Kelola Produk*

4.1.2.4 Tampilan Order



Gambar 4.9 Tampilan Order

Gambar 4.9 menunjukkan tampilan dimana admin/penjual melihat aktivitas dari orderan pelanggan. Disini admin dapat melihat aktivitas berupa tanggal, item, harga, metode pembayaran serta status. Berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Admin extends CI_Controller {
    public function __construct()
    {
        parent::__construct();

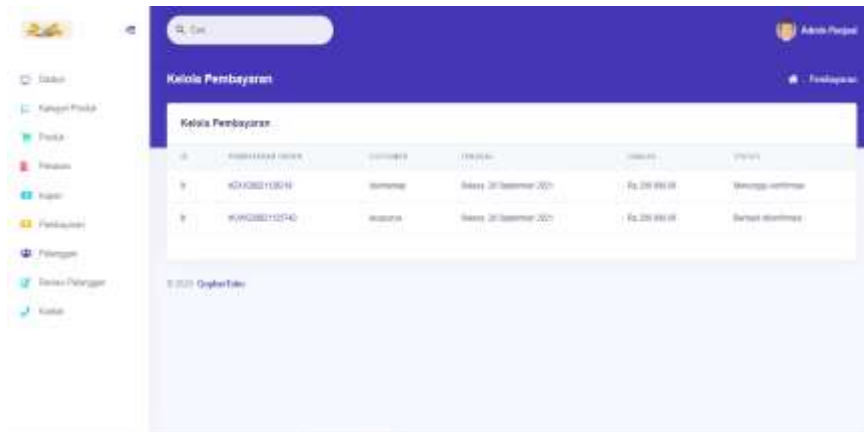
        verify_session('admin');

        $this->load->model(array(
            'product_model' => 'product',
            'customer_model' => 'customer',
            'order_model' => 'order',
            'payment_model' => 'payment'
        ));
    }

    public function index()
    {
        $params['title'] = 'Admin '. get_store_name();
    }
}
```

Gambar 4.10 Script Order

4.1.2.5 Tampil Kelola Pembayaran



Gambar 4.11 Tampilan Kelola Pembayaran

Gambar 4.11 menunjukkan tampilan dimana admin/penjual melihat aktifitas dari pembayaran pelanggan. Disini admin dapat melihat apakah proses pembayaran masi belum di konfirmasi atau sdah selesai. Berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Payments extends CI_Controller {
    public function __construct()
    {
        parent::__construct();

        verify_session('admin');

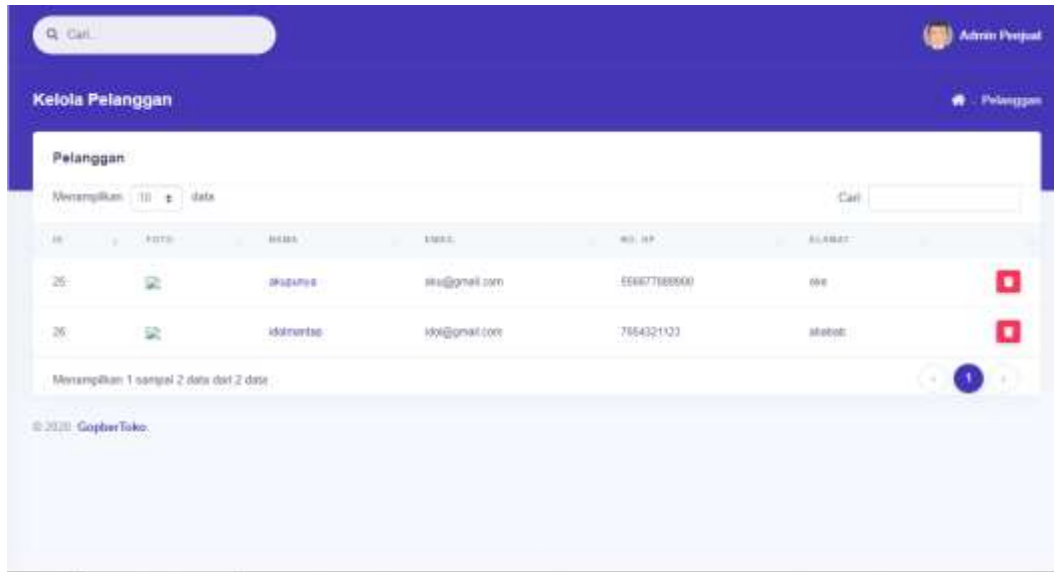
        $this->load->model(array(
            'order_model' => 'order',
            'payment_model' => 'payment'
        ));
    }

    public function index()
    {
        $params['title'] = 'Kelola Pembayaran';

        $config['base_url'] = site_url('admin/payments/index');
        $config['total_rows'] = $this->payment->count_all_payments();
        $config['per_page'] = 10;
    }
}
```

Gambar 4.12 Script edit Kelola Pembayaran

4.1.2.6 Tampilan Data Pelanggan



Gambar 4.13 Tampilan Data Pelanggan

Gambar 4.13 menunjukkan tampilan dimana admin/penjual melihat aktifitas dari pelanggan. Disini admin dapat melakukan validasi hapus. Berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

class Customer_model extends CI_Model {
    public function __construct()
    {
        parent::__construct();
    }

    public function count_all_customers()
    {
        return $this->db->get('customers')->num_rows();
    }

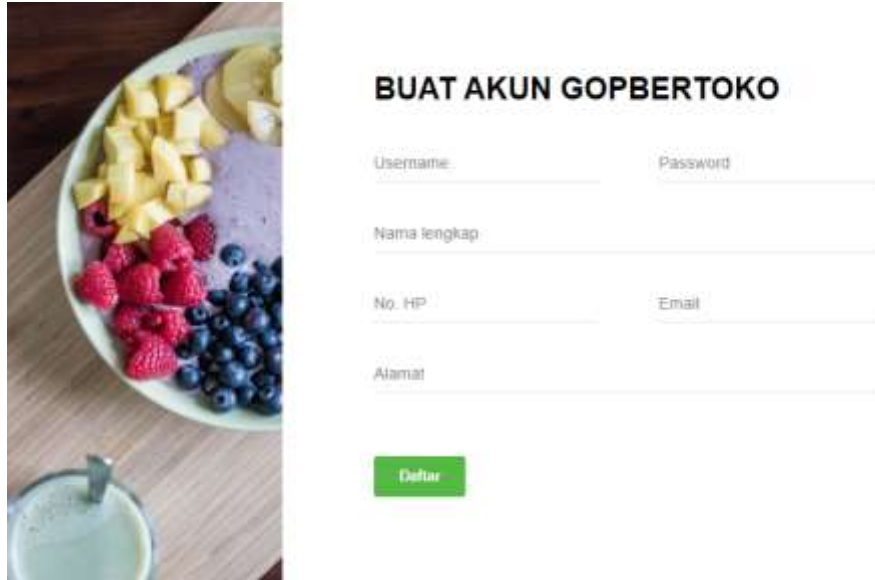
    public function latest_customers()
    {
        return $this->db->order_by('id', 'DESC')->get('customers')->result();
    }

    public function get_all_customers()
    {
        $customers = $this->db->query("
            SELECT c.user_id as id, c.profile_picture, c.name, u.email, c.phone_number,
            c.address
```

Gambar 4.14 Script List Pelanggan

4.1.3 Tampilan Pelanggan

4..1.3.1 Tampilan Halaman Buat Akun Pelanggan



BUAT AKUN GOPBERTOKO

Username: _____ Password: _____

Nama lengkap: _____

No. HP: _____ Email: _____

Alamat: _____

Daftar

Gambar 4.15 Tampilan halaman Buat Akun Pelanggan

Gambar 4.15 menunjukkan tampilan registrasi akun pelanggan yang akan di isi oleh pelanggan apabila ingin melakukan pembelian beras. berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

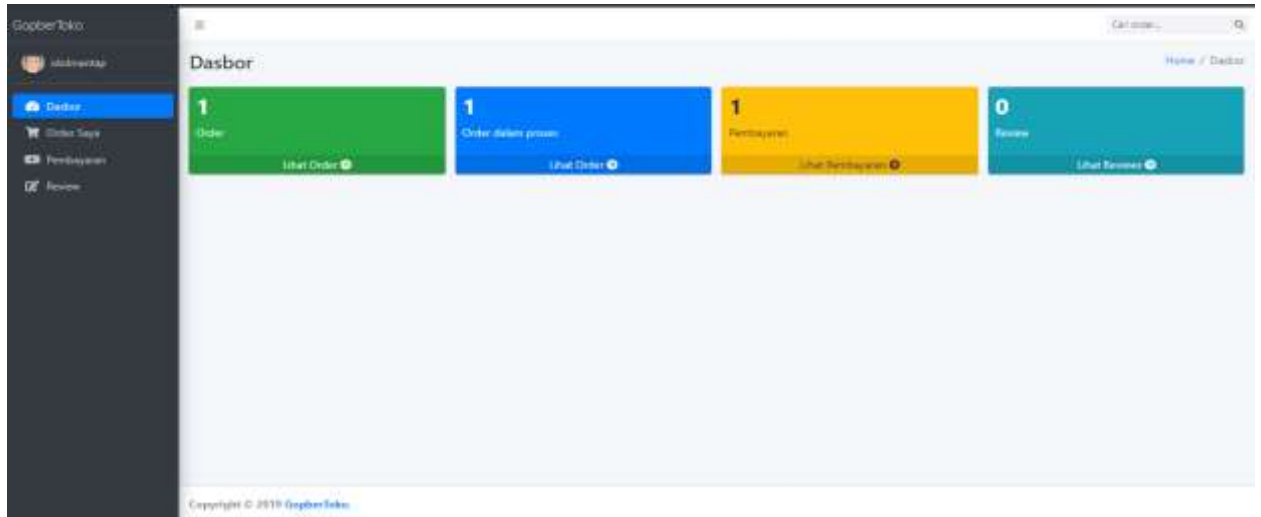
class Register extends CI_Controller {
    public function __construct()
    {
        parent::__construct();

        $this->load->library(['form_validation', 'encryption']);
        $this->load->model('auth/Register_model', 'register');
    }

    public function index()
    {
        $this->load->view('auth/register');
    }
}
```

Gambar 4.16 Script halaman buat akun

4.1.3.2 Tampilan Dashboard Pelanggan



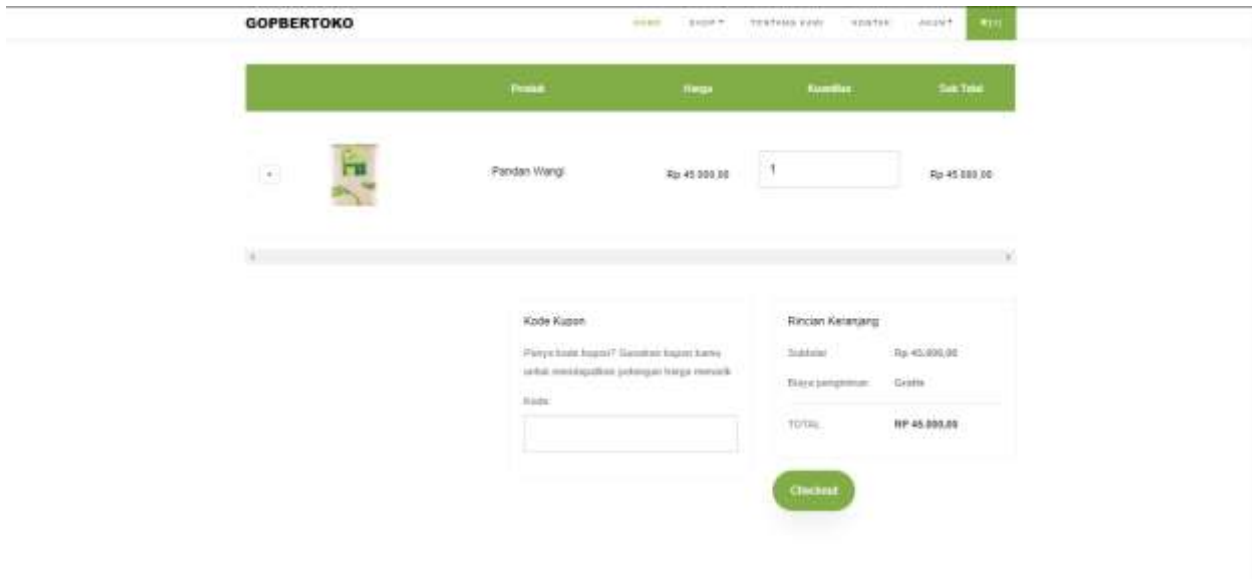
Gambar 4.17 Dashboard Pelanggan

Gambar 4.17 menunjukkan halaman dashboard dari pelanggan apabila telah selesai melakukan pendaftaran, pelanggan akan melihat tampilan berupa order, dalam proses, pembayaran serta review, berikut scrip dari tampilan dashboard :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<div class="content-wrapper">
  <div class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1 class="m-0 text-dark">Dasbor</h1>
        </div><!-- /.col -->
        <div class="col-sm-6">
          <?php if ($flash) : ?>
            <div class="text-success font-weight-bold"><?php echo $flash; ?></div>
          <?php else : ?>
            <ol class="breadcrumb float-sm-right">
              <li class="breadcrumb-item"><a href="#">Home</a></li>
              <li class="breadcrumb-item active">Dasbor</li>
            </ol>
          <?php endif; ?>
        </div>
      </div>
    </div>
  </div>
</div>
```

Gambar 4.18 Script tampilan dashboard Pelanggan

4.1.3.3 Tampilan Keranjang



Gambar 4.19 Tampilan Keranjang

Gambar 4.19 menunjukkan halaman berupa kumpulan beras-beras yang akan telah di pilih oleh pembeli yang nantinya akan diproses oleh penjual untuk di antarkan,berikut scrip dari tampilan keranjang :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');

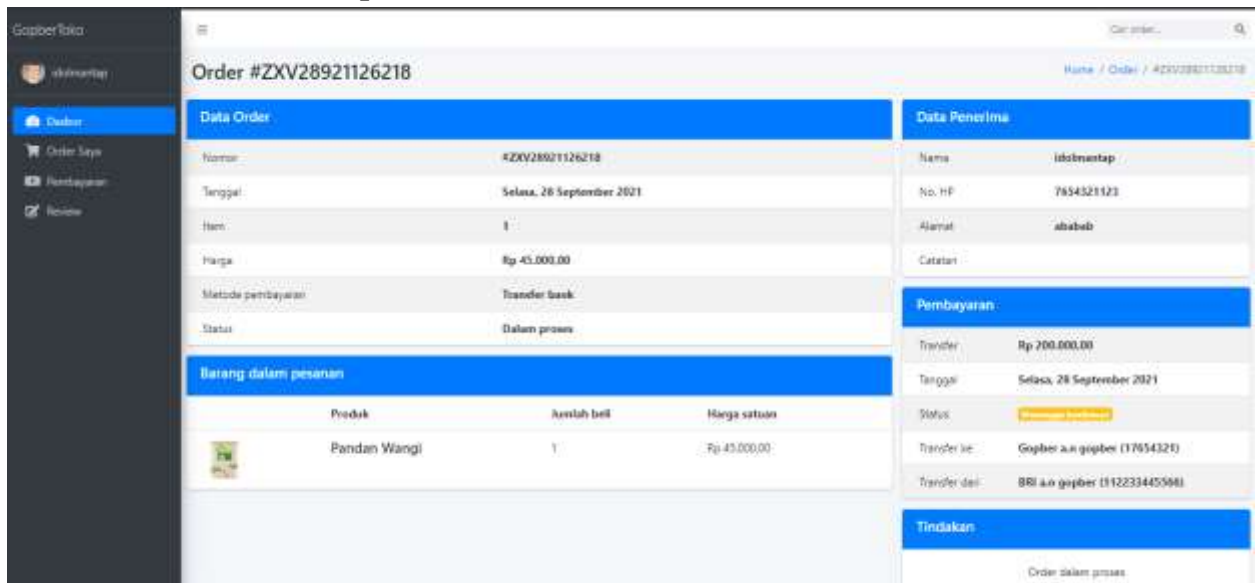
class Shop extends CI_Controller {
    public function __construct() {
        parent::__construct();

        $this->load->library('cart');
        $this->load->model(array(
            'product_model' => 'product',
            'customer_model' => 'customer'
        ));
    }

    public function product($id = 0, $sku = '')
    {
        if ($id == 0 || empty($sku))
        {
            show_error('Akses tidak sah!');
        }
    }
}
```

Gambar 4.20 Script Tampil Keranjang

4.1.3.4 Tampilan Status Order Konsumen



Gambar 4.21 Tampilan Status Order Konsumen

Pada gambar 4.21 menampilkan informasi seputar profil penjual dan konsumen aplikasi gopber, nantinya semua berita tentang profile penjual dan konsumen akan tampil di aplikasi *gopber* ini. berikut merupakan script dari gambar diatas :

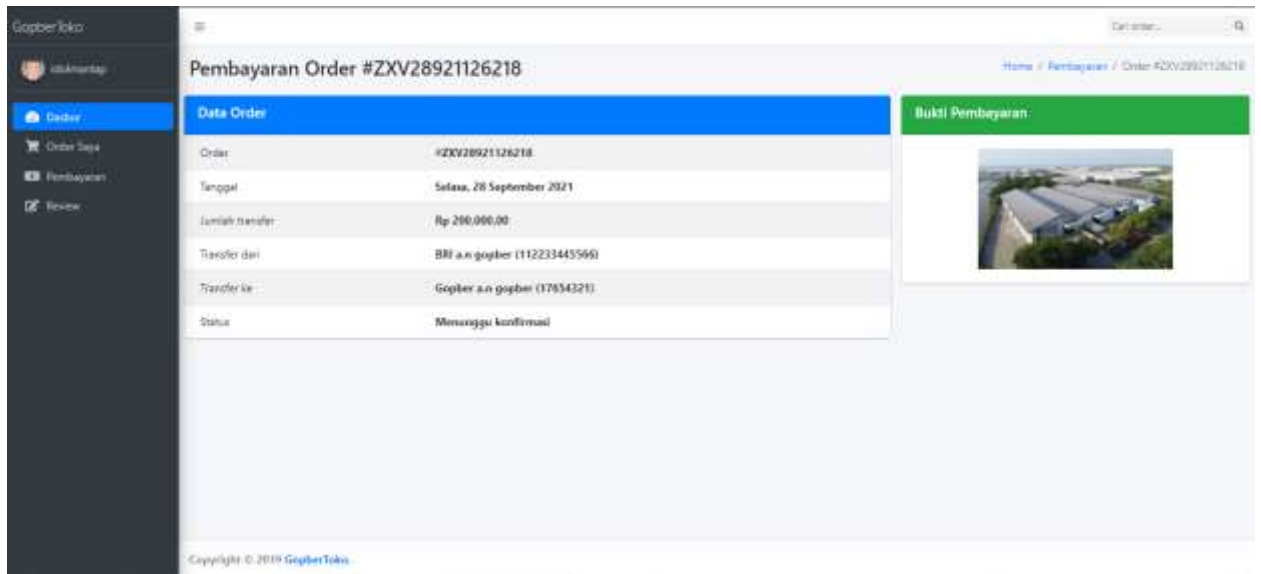
```

<?php
defined('BASEPATH') OR exit('No direct script access
allowed');
?>
<div class="content-wrapper">
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Order Saya</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><?php echo
anchor(base_url(), 'Home'); ?></li>
            <li class="breadcrumb-item
active">Order</li>
          </ol>
        </div>
      </div>
    </div>
  </div>
</section>

```

Gambar 4.22 Script Order Konsumen

4.1.3.5 Tampilan Pembayaran oleh Pelanggan



Gambar 4.23 Tampilan Bukti Pembayaran

Pada gambar 4.23 menampilkan sekumpulan tampilan produk yang akan di pesan oleh sang pembeli atau konsumen tampilan atau kegiatan terbaru yang dilakukan oleh pembeli nantinya akan tampil di *aplikasi* pemesanan ini. berikut merupakan script dari gambar diatas :

```
public function confirm()
{
    $order = $this->input->get('order');

    $params['title'] = 'Konfirmasi Pembayaran';

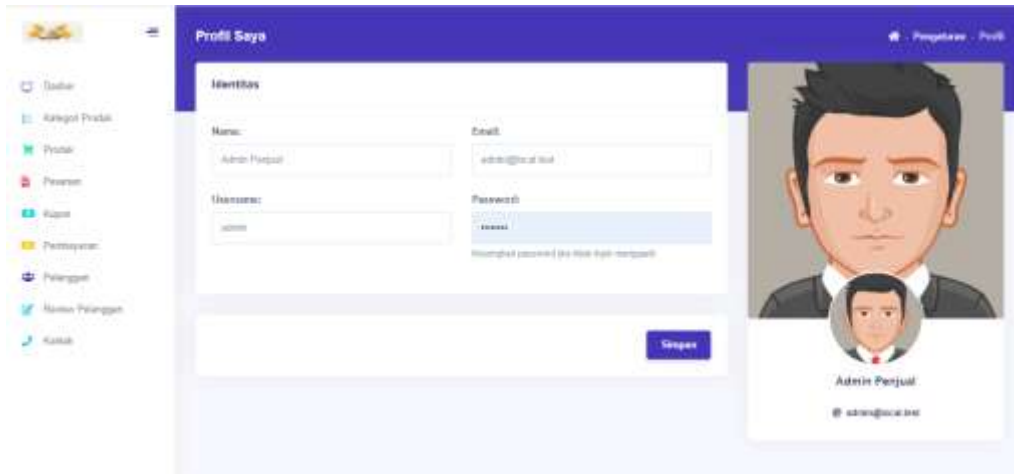
    $payments['orders'] = $this->order->order_with_bank_payments();
    $payments['banks'] = (Array) json_decode(get_settings('payment_banks'));
    $payments['order_id'] = $order;
    $payments['flash'] = $this->session->flashdata('payment_flash');
    $payments['payments'] = $this->payment->payment_list();

    $this->load->view('header', $params);
    $this->load->view('payments/confirm', $payments);
    $this->load->view('footer');
}

public function do_confirm()
{
    $this->form_validation->set_error_delimiters('<div class="font-weight-bold text-danger">', '</div>');
```

Gambar 4.24 Script Tampilan Bukti Pembayaran

4.1.3.6 Tampilan Edit Profil Pelanggan



Gambar 4.25 Tampilan Edit Profil Pelanggan

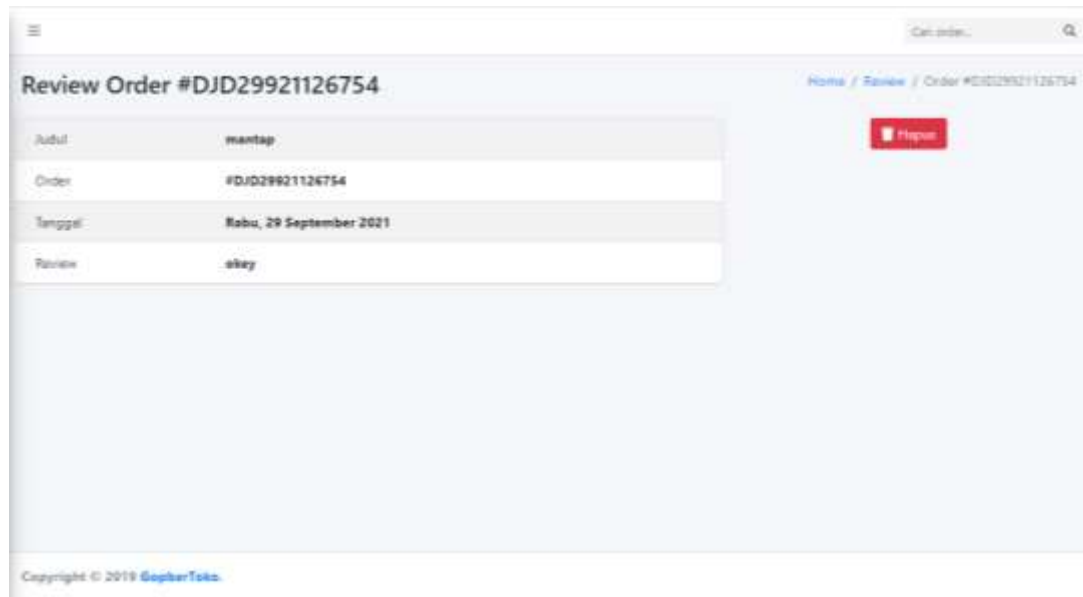
Pada gambar 4.25 menampilkan informasi seputar aktivitas dari profil pelanggan, nantinya pelanggan dapat mengubah profil mereka sesuai dengan apa yang mereka inginkan. berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<div class="content-wrapper">
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Profile</h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><?php echo anchor(base_url(), 'Home'); ?></li>
            <li class="breadcrumb-item active">Profil</li>
          </ol>
        </div>
      </div>
    </div><!-- /.container-fluid -->
  </section>

  <!-- Main content -->
  <section class="content">
    <div class="container-fluid">
      <div class="row">
        <div class="col-md-3">
```

Gambar 4.26 script Profil Pelanggan

4.1.3.7 Tampilan Review Pelanggan



Gambar 4.27 Tampilan Review Pelanggan

Pada gambar 4.27 menampilkan informasi seputar aktivitas yang merupakan review yang telah di upload oleh pelanggan. nantinya semua berita tersebut dapat dilihat oleh penjual maupun pelanggan yang mengakses Aplikasi ini. berikut merupakan script dari gambar diatas :

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
?>
<div class="content-wrapper">
  <section class="content-header">
    <div class="container-fluid">
      <div class="row mb-2">
        <div class="col-sm-6">
          <h1>Review Order #<?php echo $review->order_number; ?></h1>
        </div>
        <div class="col-sm-6">
          <ol class="breadcrumb float-sm-right">
            <li class="breadcrumb-item"><?php echo anchor(base_url(), 'Home'); ?></li>
            <li class="breadcrumb-item"><?php echo anchor('customer/reviews', 'Review'); ?></li>
            <li class="breadcrumb-item active">Order #<?php echo $review->order_number;
?></li>
          </ol>
        </div>
      </div>
    </div>
  </section>

  <section class="content">
```

Gambar 4.28 Script Tampilan Riview Pelanggan

4.1.4 Pembahasan

Berdasarkan hasil yang telah didapatkan dalam perancangan *Website Gopber* sistem ini telah dibuat sesuai dengan perancangan-perancangan yang dan pengujian yang telah dilakukan dan direncanakan sebelumnya.

Pengujian yang dilakukan berupa pengujian yang bertujuan untuk mengetahui hasil dari website tampilan website dan aplikasi android dengan menggunakan Metode *Black Box* dan *White Box*. Tujuan dari pengujian ini adalah untuk memastikan fungsi-fungsi pada perangkat lunak yang dibuat sudah berjalan sebagaimana mestinya. Berikut ini instrument yang digunakan untuk pengujian fungsional :

4.1.4.1 Pengujian Fungsional

Pengujian fungsional dilakukan untuk mengetahui apakah semua fungsi pada sistem sudah berjalan sesuai fungsi pada perancangan sistem dan sesuai dengan E-Commerce Gop-Ber (Go-Pabrik Beras).

Pengujian fungsional dilakukan menggunakan metode *black box* dan *white box*. Berikut table Pengujiannya :

Tabel 4.1 Pengujian blackbox

No	Fungsi	Test Fungsional	Hasil		Keterangan
			B	T	
1	Login Penjual	Menampilkan halaman login	✓		Berhasil menampilkan halaman login ketika web pertama kali di akses
2	Daftar Pelanggan	Menampilkan halaman daftar	✓		Berhasil menampilkan halaman pendaftaran ketika tombol

					daftar di klik
3	Profile Penjual	Menampilkan halaman profile	✓		Berhasil menampilkan halaman profile admin.
4	Dashboard Penjual	Menampilkan dashboard Penjual	✓		Berhasil menampilkan halaman dashboard admin.
5	Kelola Produk	Kelola Produk	✓		Berhasil menampilkan halaman untuk mengelola produk pada akun penjual.
6	Kelola Akun Pelanggan	Menampilkan halaman Kelola akun pelanggan	✓		Berhasil menampilkan halaman yang berisi daftar semua pelanggan.
7	Kelola Order Data	Menampilkan Order Data	✓		Berhasil menampilkan Order Data.
8	Order Pesanan Penjual	Menampilkan halaman pesanan dari Pelanggan.	✓		Berhasil menampilkan pesanan pelanggan.
9	Pembayaran	Menampilkan halaman	✓		Berhasil

	Penjual	aktivitas dari pembayaran pelanggan			Menampilkan halaman pembayaran pelanggan
10	Pelanggan Toko	Menampilkan halaman dari pelanggan yang mendaftar di aplikasi gopber	✓		Berhasil menampilkan halaman Pelanggan.
11	Dashboard Pelanggan	Menampilkan halaman Dasboard Pelanggan	✓		Berhasil menampilkan dashboard pelanggan.
12	Order Pelanggan	Menampilkan halaman dari orderan produk yang diinginkan	✓		Berhasil menampilkan halaman Orderan.
13	Pembayaran	Menampilkan halaman Pembayaran untuk penjual	✓		Berhasil menampilkan halaman Pembayaran
14	Keranjang	Menampilkan halaman keranjang untuk membeli produk.	✓		Berhasil Menampilkan halaman Keranjang.
15	Review	Menampilkan halaman review setelah melakukan pemesanan	✓		Berhasil menampilkan halaman review.
16	Profil Pelanggan	Menampilkan halaman Aktivitas Pelanggan yang ingin di edit atau	✓		Berhasil menampilkan halaman

		logout			Aktivitas
--	--	--------	--	--	-----------

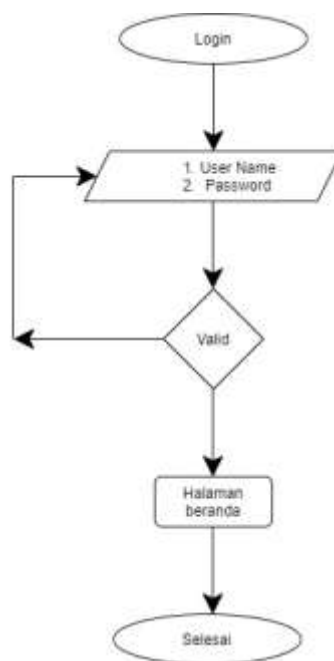
Setelah dilakukan pengujian program aplikasi dari 16 pengetesan yang dilakukan aplikasi berjalan sesuai dengan yang di inginkan.

4.1.4.2 Pengujian *White Box*

A. Form Login

1. *Flowchart* form login

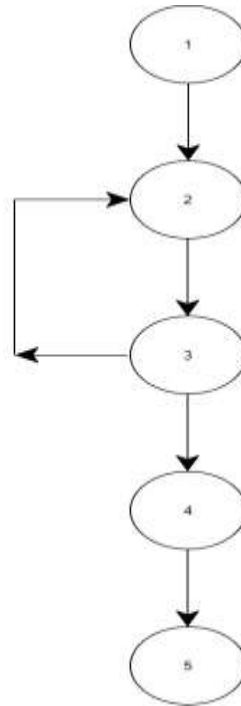
Adapun *flowchart* dari form login dapat dilihat pada gambar 4.29 sebagai berikut



Gambar IV. 29 *Flowchart* form login

2. *Flowgraph* form login

Berdasarkan *flowchart* form login maka dapat ditentukan *flowgraph* sebagai berikut:



Gambar 4. 30 Flowgraph form login

3. Menghitung *Cyclomatic Complexity*

Hasil dari *Cyclomatic complexity* berdasarkan *flowgraph* pada atas memiliki region = 4,

1. Hitungan *cyclomatic complexity* berdasarkan edge & node

rumus : $V(G) = E - N + 2$

dimana : E (Jumlah Edge dalam *flowgraph*) = 5

N (Jumlah Node dalam *flowgraph*) = 5

penyelesaian : $V(G) = 5 - 5 + 2$

$V(G) = 2$

2. *Independent path*

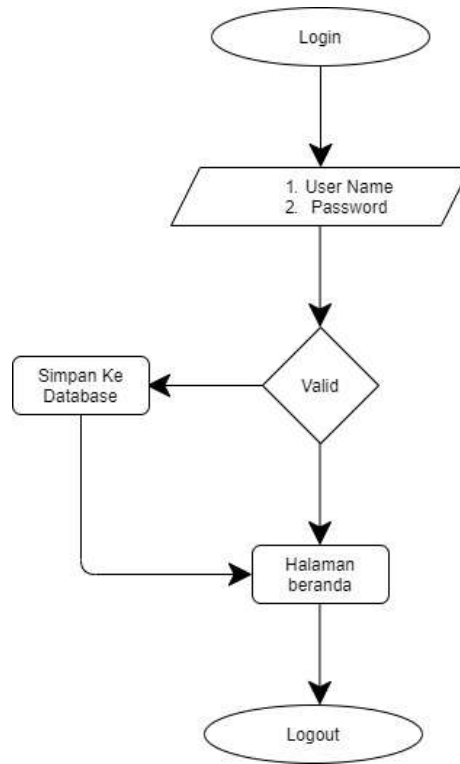
Path 1 = 1 – 2 – 3 – 4 – 5

Path 2 = 1 – 2 – 3 – 2 – 3 – 4 – 5

B. Buat Akun

1. *Flowchart* Buat Akun

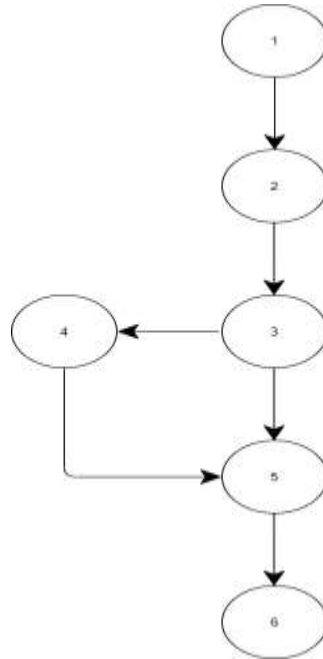
Adapun *flowchart* dari buat Akun dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 31 *Flowchart* buat Akun

2. *Flowgraph* buat Akun

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



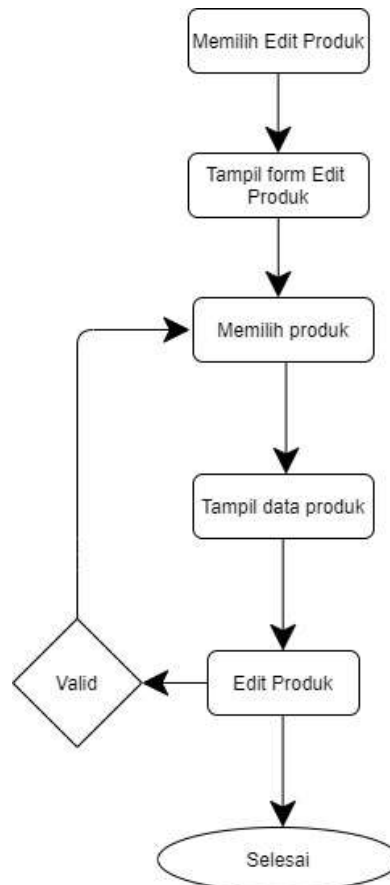
Gambar IV. 32 *Flowgraph* buat Akun

1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 2$
 - dimana : E (Jumlah Edge dalam flowgraph) = 5
 - N (Jumlah Node dalam flowgraph) = 5
 - penyelesaian : $V(G) = 5 - 5 + 2$
 - $V(G) = 2$
2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

C. Mengola Data Produk

1. *Flowchart* Mengola Data Produk

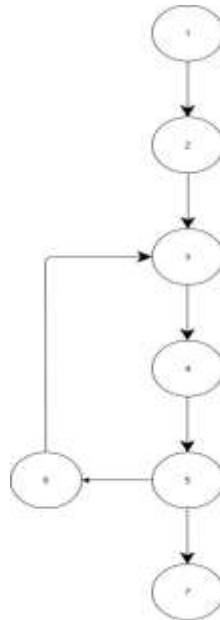
Adapun *flowchart* dari buat Akun dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 33 *Flowchart* Mengola Data Produk

2. *Flowgraph* mengola data produk

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



Gambar IV. 34 *Flowgraph* mengola data produk

3. Hitungan *cyclomatic complexity* dari *edge* dan *node*

rumus : $V(G) = E - N + 2$

dimana : E (jumlah *edge* pada *Flowgraph*) = 7

N (jumlah *node* pada *Flowgraph*) = 7

penyelesaian : $V(G) = 7 - 7 + 2$

$V(G) = 2$

4. Independent Path

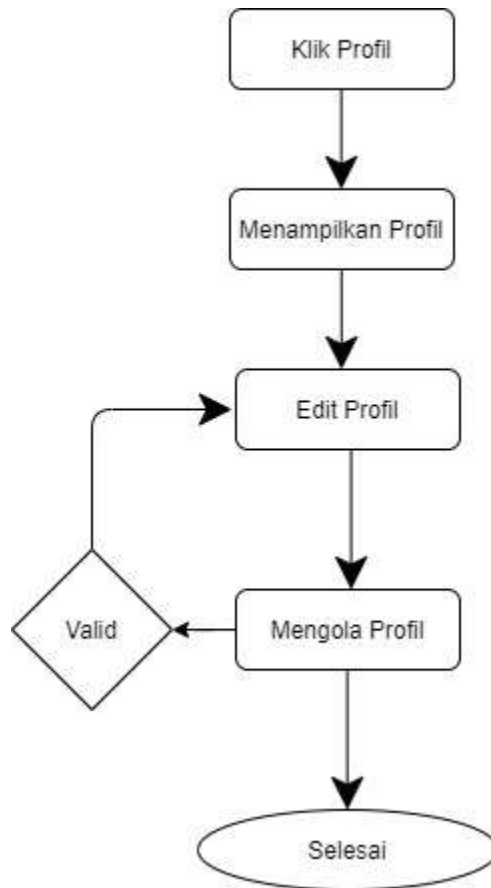
Path 1 = 1 - 2 - 3 - 4 - 5 - 7

Path 2 = 1 - 2 - 3 - 4 - 5 - 6 - 6 - 3 - 4 - 5 - 7

D. Mengola Profil

1. *Flowchart* mengola Profil

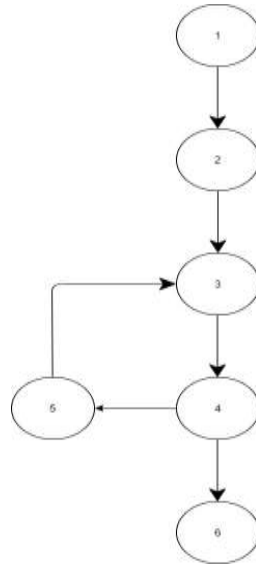
Adapun *flowchart* dari mengola profil dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 35 *Flowchart* buat Akun

2. *Flowgraph* Mengola Profil

Berdasarkan *flowchart* mengola profil maka dapat ditentukan *flowgraph* sebagai berikut :



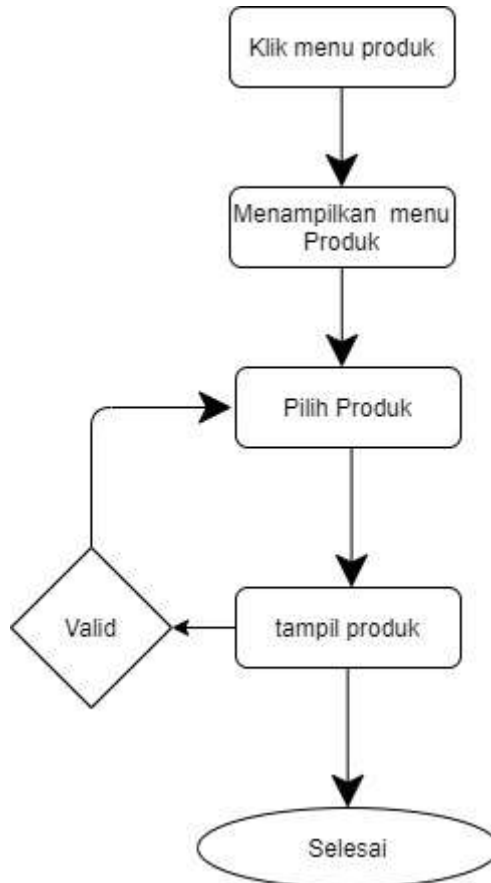
Gambar IV. 36 *Flowgraph* mengola profil

1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 6$
 - dimana : E (Jumlah Edge dalam flowgraph) = 6
 - N (Jumlah Node dalam flowgraph) = 6
 - penyelesaian : $V(G) = 6 - 6 + 2$
 - $V(G) = 2$
2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

E. Mengola Memesan Produk

1. Flowchart Memesan Produk

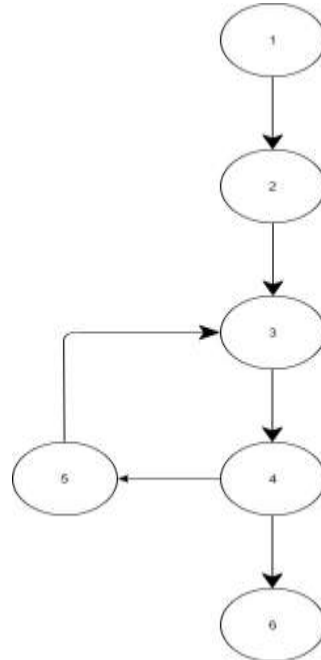
Adapun *flowchart* dari buat Akun dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 37 Flowchart memesan produk

2. Flowgraph Memesan Produk

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



Gambar IV. 38 *Flowgraph* Memesan Produk

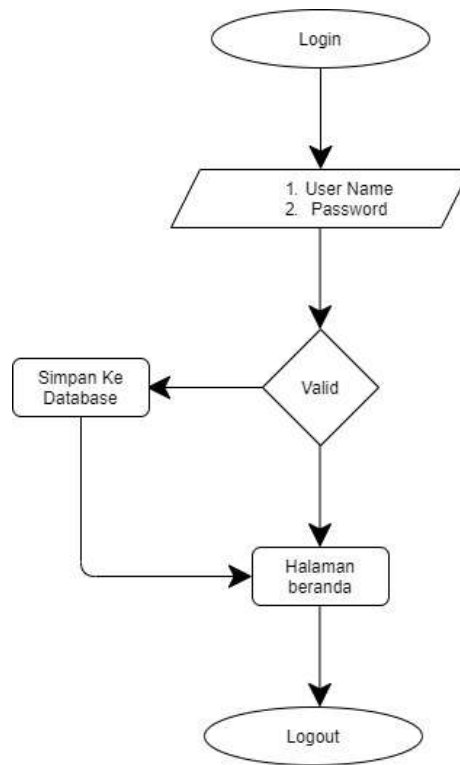
1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 2$
 - dimana : E (Jumlah Edge dalam flowgraph) = 6
 - N (Jumlah Node dalam flowgraph) = 6
 - penyelesaian : $V(G) = 6 - 6 + 2$
 - $V(G) = 2$

2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 5 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

F. Mengola Melihat Produk

1. *Flowchart* Melihat Produk

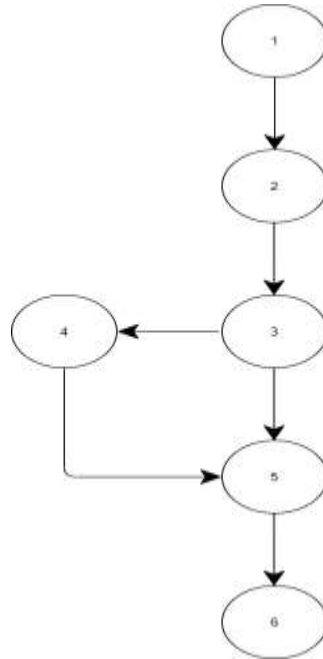
Adapun *flowchart* dari buat Akun dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 39 *Flowchart* buat Akun

2. *Flowgraph* buat Akun

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



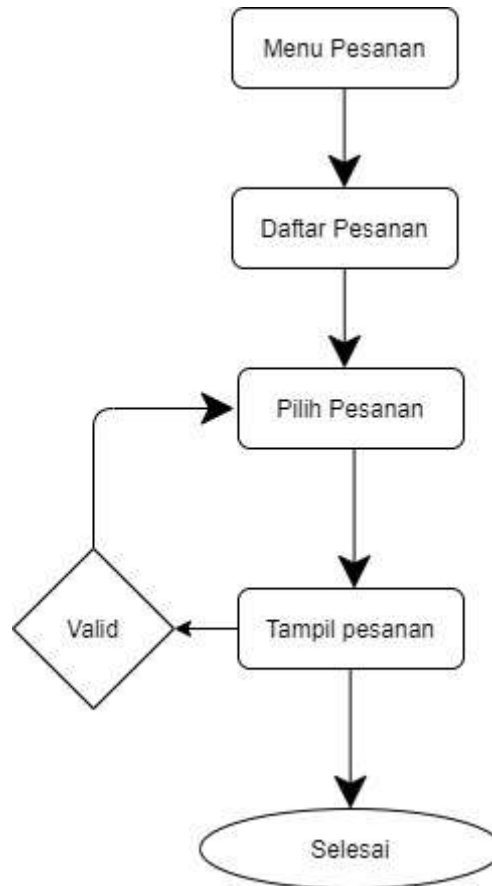
Gambar IV. 40 *Flowgraph* buat Akun

1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 6$
 - dimana : E (Jumlah Edge dalam flowgraph) = 6
 - N (Jumlah Node dalam flowgraph) = 6
 - penyelesaian : $V(G) = 6 - 6 + 2$
 - $V(G) = 2$
2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

G. Mengecek Pesanan

1. *Flowchart* Buat Akun

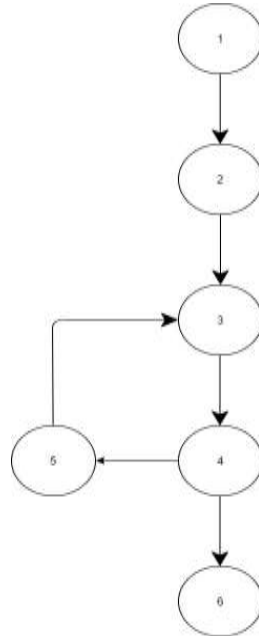
Adapun *flowchart* dari buat Akun dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 41 *Flowchart Mengecek Pesanan*

2. *Flowgraph* Mengecek Pesanan

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



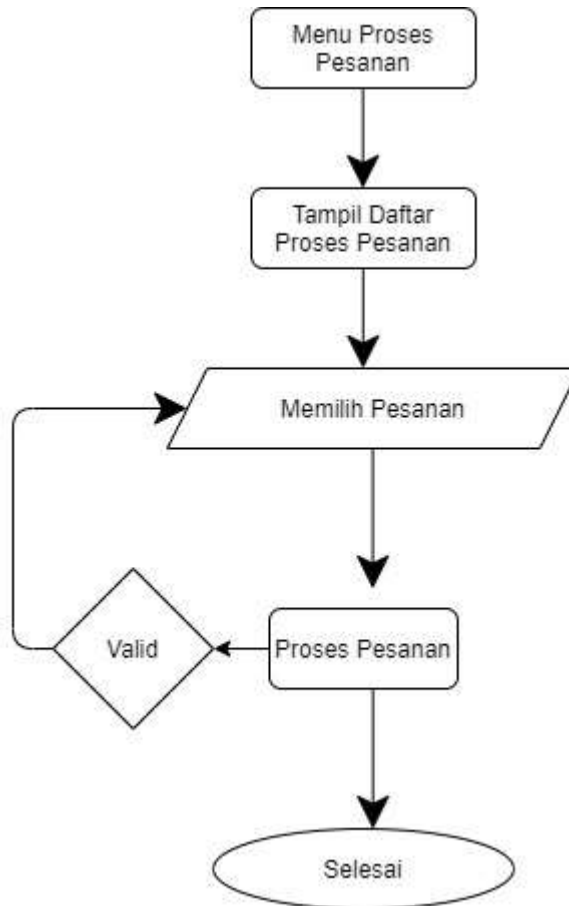
Gambar IV. 42 *Flowgraph* Mengecek Pesanan

1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 6$
 - dimana : E (Jumlah Edge dalam flowgraph) = 6
 - N (Jumlah Node dalam flowgraph) = 6
 - penyelesaian : $V(G) = 6 - 6 + 2$
 - $V(G) = 2$
2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

H. Memproses Pesanan

1. *Flowchart* Memproses Pesanan

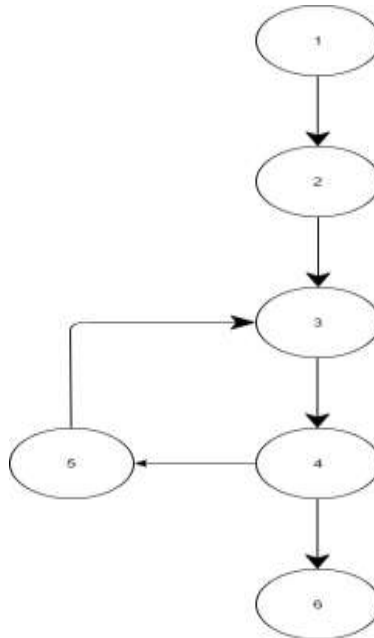
Adapun *flowchart* dari memproses pesanan dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 43 *Flowchart* Memproses Pesanan

2. *Flowgraph* Memproses Pesanan

Berdasarkan *flowchart* Memproses Pesanan maka dapat ditentukan *flowgraph* sebagai berikut :



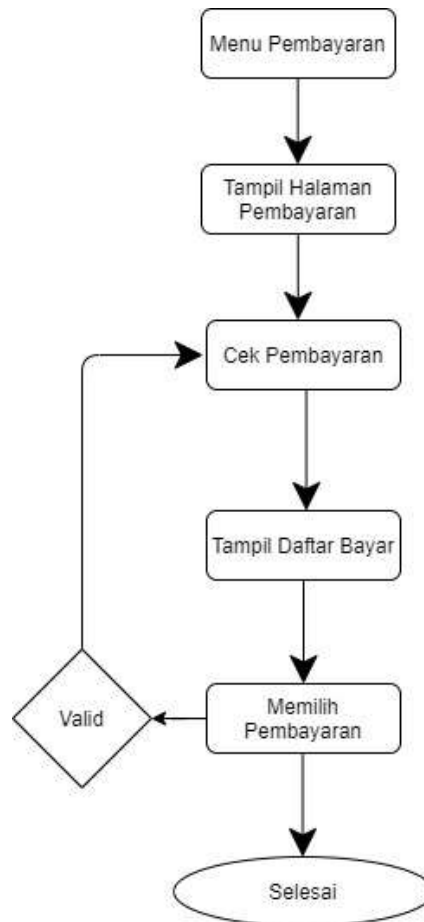
Gambar IV. 44 *Flowgraph* Memproses Pesanan

1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 6$
 - dimana : E (Jumlah Edge dalam flowgraph) = 6
 - N (Jumlah Node dalam flowgraph) = 6
 - penyelesaian : $V(G) = 6 - 6 + 2$
 - $V(G) = 2$
2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

I. Validasi Pembayaran

1. Flowchart Validasi Pembayaran

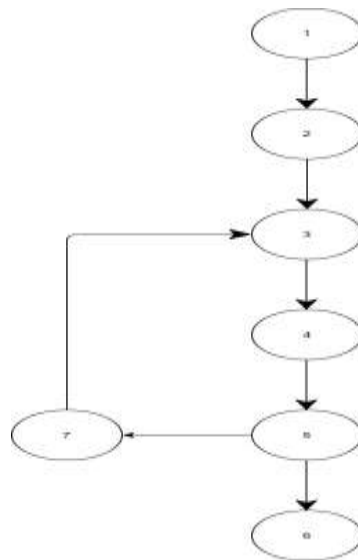
Adapun *flowchart* dari Validasi Pembayaran dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 45 *Flowchart* Validasi Pembayaran

2. Flowgraph Validasi Pembayaran

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



Gambar IV. 46 *Flowgraph* Validasi Pembayaran

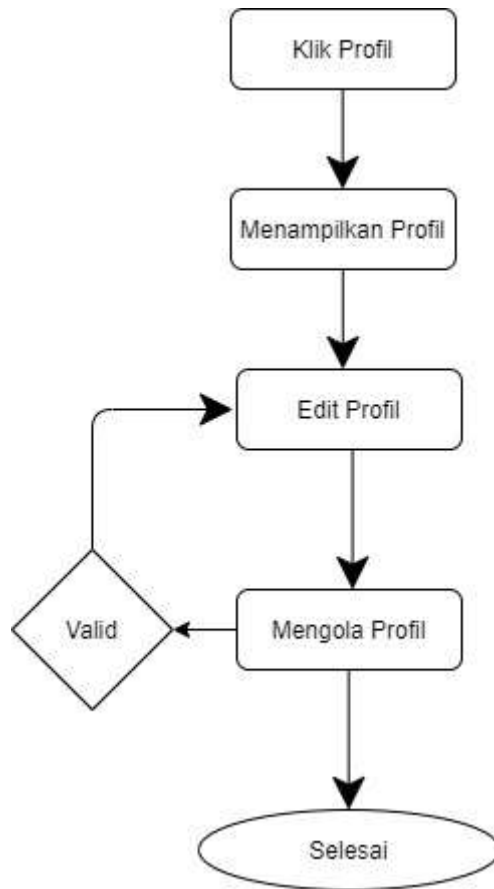
1. Hitungan cyclomatic complexity berdasarkan edge & node
 - rumus : $V(G) = E - N + 6$
 - dimana : E (Jumlah Edge dalam flowgraph) = 6
 - N (Jumlah Node dalam flowgraph) = 6
 - penyelesaian : $V(G) = 6 - 6 + 2$
 - $V(G) = 2$

2. Independent path
 - path 1 = 1 - 2 - 3 - 4 - 6
 - path 2 = 1 - 2 - 3 - 5 - 6

J. Konfirmasi Pembayaran

1. Flowchart Memproses Pesanan

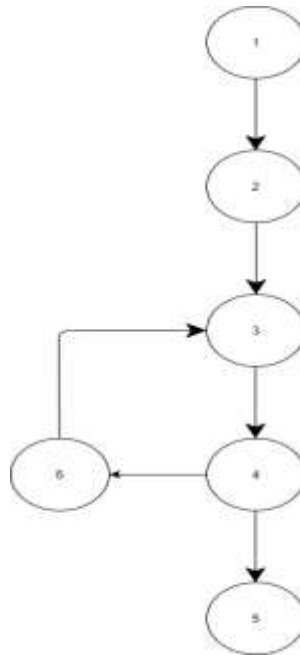
Adapun *flowchart* dari Konfirmasi Pembayaran dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 47 *Flowchart* Konfirmasi Pembayaran

2. *Flowgraph* Konfirmasi Pembayaran

Berdasarkan *flowchart* Konfirmasi Pembayaran maka dapat ditentukan *flowgraph* sebagai berikut :



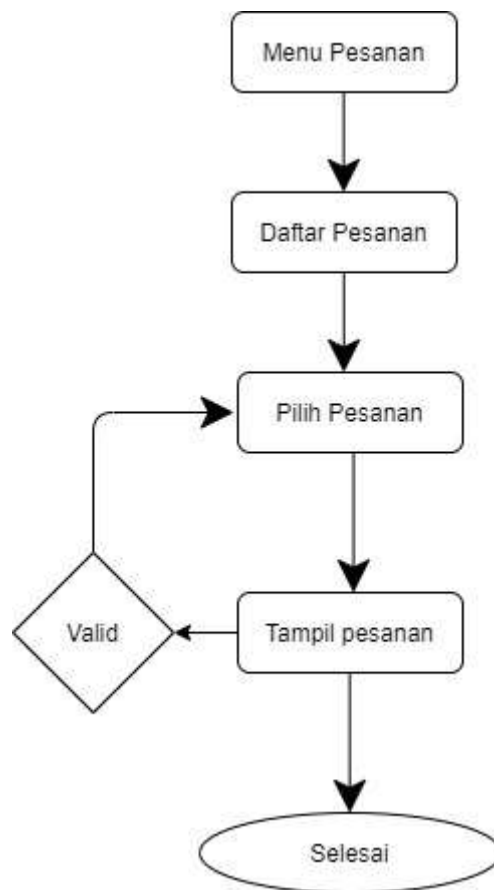
Gambar IV. 48 *Flowgraph* Konfirmasi Pembayaran

1. Hitungan cyclomatic complexity berdasarkan edge & node
rumus : $V(G) = E - N + 6$
dimana : E (Jumlah Edge dalam flowgraph) = 6
N (Jumlah Node dalam flowgraph) = 6
penyelesaian : $V(G) = 6 - 6 + 2$
 $V(G) = 2$
2. Independent path
path 1 = 1 - 2 - 3 - 4 - 6
path 2 = 1 - 2 - 3 - 5 - 6

K. Mengecek Status Pesanan

1. Flowchart Mengecek Status Pesanan

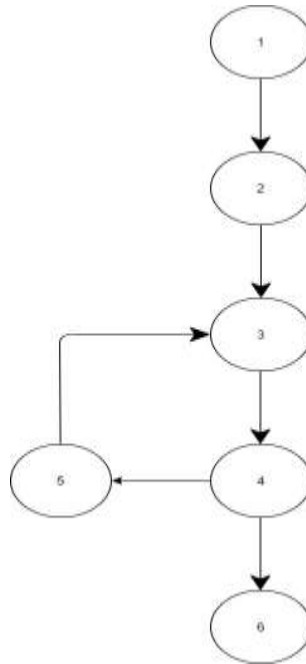
Adapun *flowchart* dari Mengecek Status Pesanan dapat dilihat pada gambar IV.71 sebagai berikut:



Gambar IV. 49 *Flowchart* Validasi Pembayaran

2. Flowgraph Mengecek Status Pesanan

Berdasarkan *flowchart* buat Akun maka dapat ditentukan *flowgraph* sebagai berikut :



Gambar IV. 50 Flowgraph Mengecek Status Pesanan

1. Hitungan cyclomatic complexity berdasarkan edge & node
rumus : $V(G) = E - N + 6$
dimana : E (Jumlah Edge dalam flowgraph) = 6
N (Jumlah Node dalam flowgraph) = 6
penyelesaian : $V(G) = 6 - 6 + 2$
 $V(G) = 2$
2. Independent path
path 1 = 1 - 2 - 3 - 4 - 6
path 2 = 1 - 2 - 3 - 5 - 6

4.1.5 Rekapitulasi

Hasil dari rekapitulasi pengujian white box

Tabel 4.2 Pengujian blackbox

No	<i>Fungsi uji</i>	<i>Jumlah Path</i>	Hasil
1	Diagram Login	2	Berhasil menampilkan halaman login ketika web pertama kali di akses
2	Diagram Daftar	2	Berhasil menampilkan halaman pendaftaran ketika tombol daftar di klik
3	Diagram Mengola Data Produk	2	Berhasil menampilkan halaman Kelola data produk.
4	Diagram Mengola Profil	2	Berhasil menampilkan halaman aktivitas untuk kelola profil.
5	Diagram Memesan Produk	2	Berhasil menampilkan halaman untuk memesan Produk.
6	Diagram Melihat Produk	2	Berhasil menampilkan halaman yang berisi daftar Produk.
7	Diagram Mengecek Pesanan	2	Berhasil menampilkan data pesanan.
8	Diagram Memproses Pesanan	2	Berhasil menampilkan pesanan yang diproses.
9	Diagram Validasi Pembayaran	2	Berhasil Menampilkan halaman untuk validasi Pembayaran
10	Diagram Konfirmasi Pembayaran	2	Berhasil menampilkan halaman untuk konfirmasi pembayaran.
11	Diagram Mengecek Status Pesanan	2	Berhasil menampilkan halaman untuk mengecek status pesanan .

Setelah dilakukan pengujian program aplikasi dari 11 pengetesan yang dilakukan aplikasi berjalan sesuai dengan yang diinginkan. Serta sudah terbebas dari kesalahan logika.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa :

1. *E-Commerce* GOPBER (Go-Pabrik Beras) berhasil di jalankan sesuai dengan sistem perancangan dan tujuan yang ada, serta fungsi dari beberapa fitur yang ada di admin penjual seperti kategori produk, konfirmasi pembayaran, status, dan yang ada di pelanggan kategori bayar pilih produk dan edit semua berhasil dengan baik.
2. Berdasarkan hasil pengujian fungsional menggunakan *Black box* dan *White box*, sistem berjalan dengan baik.

5.2 Saran

Berdasarkan masalah yang ada dengan ini penulis memberikan saran yang kiranya dapat membantu meningkatkan kinerja toko Rama35. Saran tersebut antara lain :

1. Agar kelancaran pengolahan aplikasi yang ada, kiranya komputer dapat di perhatikan perawatannya mulai dari perangkat lunak serta perangkat kerasnya.
2. Sekiranya user yang menjadi pelanggan agar diberikan instruksi tentang cara pakai aplikasi.
3. Untuk keamanan data disarankan untuk menyimpan file yang di aplikasi di perangkat lainya, guna menghindari kerusakan sistem atau kehilangan siste

DAFTAR PUSTAKA

4. Daniel Dido Jantce TJ Sitinjak,dkk, (2020). Analisa Dan Perancangan Sistem Informasi Administrasi Kursus Bahasa Inggris Pada Intensive English Course Di Ciledug Tangerang. Jurnal Ipsikom. Vol. 8 No.1, Juni 2020 ISSN : 2338-4093, E-ISSN : 2686-6382.
5. Dani Tri Indraswara (2014), rekayasa perangkat lunak sistem penjualan jasa konsultasi desain interior,3d animasi berbasis web pada pt. Jaya metro. Jurnal tugas akhir teknik informatika s1 universitas dian nuswantoro semarang 2014.
6. Desi Tri Hernandhi,dkk, (2018). Perancangan sistem informasi inventory spare part elektronik berbasis web php (studi cv. Human global service yogyakarta). Jurnal Sarjana Teknik Informatika e-ISSN: 2338-5197 Volume 2 Nomor 2, Juni 2014.
7. Dewi Rosmala, dkk, (2011). Komparasi *framework mvc(codeigniter, dan cakephp)* pada aplikasi berbasis *web*. Jurnal informatika. No.2, Vol. 2, Mei – Agustus 2011.
8. Frederick Constantianus, (2005). Analisa dan Desain Sistem Bimbingan Tugas Akhir Berbasis Web dengan Studi Kasus Fakultas Teknologi Informasi. Jurnal Informatika UKM, Vol. I, No. 2, Desember 2005: 93 – 106.
9. Fifit Alfiah,dkk, (2014) perancangan sistem *e-commerce* untuk penjualan pakaian pada toko a&s Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer Vol 6 No 1 – Februari 2020.
10. Hendri Krida Prakasa, Indra Gamayanto, ST, MITM2 (2013), Perancangan dan Implementasi *E-Commerce* pada rumah beras tiredi kota Semarang
11. Ipan Ripai, M.Kom, (2017), Mengukur Kinerja Auto Generate Content menggunakan plugin *Wordpress WP AGC EBO* dalam menghasilkan dollar di *Google Adsense*. Jurnal ICT Learning Vol. 3 No. 2 Desember 2017.
12. Mhd Bustanur Rahmad, Tedy Setiady, (2014), Perancangan Sistem Informasi *Inventory Spare Part Elektronik* Berbasis Web Php (Studi CV. Human Global Service Yogyakarta). Jurnal Sarjana Teknik Informatika e-ISSN: 2338-5197 Volume 2 Nomor 2, Juni 2014.

13. Pastima Simanjuntak, Arwin Kusnady, (2016), analisis model *view controller* (mvc) pada bahasa php. Jurnal ISD Vol.2 No.2 Juli - Desember 2016 e-ISSN : 2528-5114.
14. Rudika Harminingtyas, (2014). analisis layanan website sebagai media promosi, media transaksi dan media informasi dan pengaruhnya terhadap brand image perusahaan pada Hotel Ciputra di Kota Semarang Jurnal STIE Semarang, vol 6, no 3, edisi oktober 2014 (issn : 2252-7826).
15. Rifaldy Fahruly Azhar Muhammad, R. Fenny Syafariani S.Si., M.Stat (2018). Sistem Informasi Pengelolaan Beras Di Pb.Saigon Berbasis Web.
16. Samsul Arifin, Yolanda Krisnadita, (2017), Aplikasi Plugin Transfer Domain Di Pt Beon Intermedia. Jurnal Teknologi Informasi| ISSN 2086-2989 Vol. 8 No. 1 Maret 2017.
17. Sidik, Anton Surryoprayogo, (2015), implementasi algoritma brute force pada perancangan aplikasi kamus bahasa sunda berbasis android Jurnal Informatika dan Bisnis.
18. Sukisno S.Kom., M.kom, Winda Frantika Wuni (2017), analisa dan perancangan sistem informasi tracking acuan quality departemen brushing berbasis web di pt. Indotaichen textile industry. Journal of Informatics Engineering Vol.5 No.1 April 2017 Faculty of Engineering Islamic University Syekh-Yusuf Tangerang.
19. Taufik Hidayat, dkk, (2018), Optimalisasi Kinerja Server Menggunakan manajemen Dns Optimizing Server Performance Using Dns Management (Studi Kasus : Ist Akprind Yogyakarta). Jurnal JARKOM Vol. 5 No. 2 Juni 2018 ISSN:2338-6304.
20. Yusi Ardi Binarso, dkk, (2012), Pembangunan sistem informasi alumni berbasis web pada program studi teknik informatika universitas diponegoro. *Journal of Informatics and Technology*, Vol 1, No 1, Tahun 2012, p 72-84.