**PEMBANGUNAN *E-COMMERCE* PENJUALAN**

**PERANGKAT KOMPUTER DI GEMILANG**

**JAYA *COMPUTER***

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat Kelulusan Program Sarjana Strata-1 (S1)

Program Studi Teknik Informatika Fakultas Teknik Universitas Suryakancana

Cianjur

**OLEH**

**HARIS ANWAR**

**5520118042**



**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS SURYAKANCANA**

**CIANJUR**

**2021**

**LEMBAR PENGESAHAN**

**LAPORAN TUGAS AKHIR**

Nama : HARIS ANWAR

NPM : 5520118042

Program Studi : TEKNIK INFORMATIKA

Judul : PEMBANGUNAN *E-COMMERCE* PENJUALAN

PERANGKAT KOMPUTER DI GEMILANG

JAYA *COMPUTER*

Telah disidangkan dan dinyatakan Lulus Sidang Tugas Akhir pada Program Sarjana Strata-1 (S1), Program Studi Teknik Informatika Fakultas Teknik Universitas Suryakancana Cianjur pada tanggal ……..

Tim Penguji

Nama Dosen Sidang Tugas Akhir Tanda Tangan

1. ………..
2. ………..
3. ………..

Mengetahui :

Ketua Program Studi

Teknik Informatika

(Tarmin Abdulghani MT)

# LEMBAR PERNYATAAN KEASLIAN

**LAPORAN TUGAS AKHIR**

Nama : HARIS ANWAR

NPM : 5520118042

Program Studi : TEKNIK INFORMATIKA

Judul : PEMBANGUNAN *E-COMMERCE* PENJUALAN

PERANGKAT KOMPUTER DI GEMILANG

JAYA *COMPUTER*

Dengan ini saya menyatakan bahwa dalam laporan penelitian ini tidak terdapat karya yang pernah diajukan untuk memperoleh kelulusan gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Cianjur, 29 Oktober 2021

Materai

Haris Anwar

NPM:5520118042

**ABSTRAK**

Penawaran barang, proses pemesanan barang, serta transaksi penjualan merupakan bagian yang dilakukan dalam operasional suatu toko, terutama toko yang bergerak di bidang perdagangan. TOKO GEMILAN JAYA *COMPUTER* merupakan suatu toko yang bergerak di bidang perdagangan, penawaran barang, proses pemesanan barang, serta transaksi penjualan barang dalam operasional toko.

Pada saat ini informasi barang yang tersedia, serta transaksi penjualan pada Gemilang Jaya *Computer* dapat dilakukan dengan datang ke toko*,* transaksi tersebut tidak dapat di jangkau secara luas, ketidakmudahan *customer* dalam mendapatkan informasi secara cepat, mengenai informasi produk yang tersedia, karena promosi produk masih menggunakan brosur atau media cetak. Proses pemesanan membutuhkan waktu lama, karena tidak ada sistem atau aplikasi, mengacu pada pentingnya proses bisnis yang berjalan, maka dibutuhkan suatu sistem atau aplikasi *ecommerce* yang dapat menangani permasalahan tersebut, sehingga proses bisnis dapat berjalan dengan efektif.

Dalam pembuatan perangkat lunak ini, digunakan metode pengembangan sistem berdasarkan paradigma *prototype* yang terdiri komunikasi, perencanaan, pemodelan, kontruksi, dan penyerahan. Bahasa pemrograman yang digunakan untuk membangun sistem aplikasi *ecommerce* yaitu menggunakan bahasa pemrograman *php, jquery, javascript,* dengan menggunakan framework *laravel, nux.js* dan *bootsrap* dan *DBMS MYSQL* dengan perangkat lunak *Laragon*,. Untuk metoda perancangan sistem menggunakan konsep perancangan beriorentasi *object* dimana menggunakan *tools* perancangan yaitu UML.

Hasil dari perancangan pada penelitian ini, di implementasikan menjadi perangkat lunak atau sistem aplikasi *ecommerce* berbasis web, yang berfungsi untuk memberikan informasi produk, proses pemesanan barang, serta transaksi penjualan barang, dengan pembangunan sistem atau perangkat lunak ini diharapkan proses bisnis dapat berjalan dengan efektif.

Kata Kunci : aplikasi, penawaran, pemesanan, transaksi, *php, MyQL, Prototype*

***ABSTRACT***

*The supply of goods, the process of ordering goods, and sales transactions are part of the operations of a store, especially stores engaged in trading. TOKO GEMILANG JAYA COMPUTER is a store that is engaged in trading, offering goods, ordering goods, and selling goods in store operations..*

*At this time information on available goods, as well as sales transactions at Gemilang Jaya Computer can be done by coming to the store, these transactions cannot be widely reached, it is not easy for customers to get information quickly, regarding available product information, because product promotions are still using brochures or print media. The ordering process takes a long time, because there is no system or application, referring to the importance of running business processes, it takes a system or e-commerce application that can handle these problems, so that business processes can run effectively.*

*In making this software, a system development method is used based on the prototype paradigm consisting of communication, planning, modeling, construction, delivery. The programming language used to build the e-commerce application system is using the php, jquery, javascript programming languages, using laravel and bootsrap frameworks and MYSQL DBMS with Laragon software. The system design method uses an object-oriented design concept which uses design tools, namely UML.*

*The results of the design in this study, are implemented into software or web-based e-commerce application systems, which function to provide product information, process goods ordering, and sales transactions of goods, with the development of this system or software, it is expected that business processes can run effectively..*

*Keywords: application, offer, order, transaction, php, MyQL, Prototype*

# KATA PENGANTAR

Dengan memanjatkan puji syukur kehadiran Tuhan Yang Maha Esa yang telah melimpahkan segala rahmat dan hidayahnya kepada penulis, sehingga tersusunlah laporan penelitian tugas akhir ini yang berjudul **“PEMBANGUNAN *E-COMMERCE* PENJUALAN PERANGKAT KOMPUTER DI GEMILANG JAYA *COMPUTER*”**.

Tugas Akhir merupakan pelengkap salah satu persyaratan yang diajukan dalam rangka menempuh ujian akhir untuk memperoleh gelar Sarjana Teknik ( S.T. ) pada Program Sarjana Starta-1 (S1), Program Studi Teknik Informatika di Fakultas Teknik Universitas Suryakancana Cianjur.

Penulis sungguh sangat menyadari, bahwa penulisan laporan Tugas Akhir ini tidak akan terwujud tanpa adanya dukungan dan bantuan dari perbagai pihak. Maka, dalam kesempatan ini penulis memberikan ucapan terima kasih yang sebesar - besarnya kepada yang terhormat :

1. Kedua Orangtua yang selalu memberi dukungan tanpa henti dan semangat kepada penulis
2. Bapak. Tarmin Abdulghani, M.T, selaku ketua jurusan Fakultas Teknik Informatika
3. Bapak. Finsa Nurpandi, M.T, selaku koordinator tugas akhir
4. Ibu. Siti Nazilah, M.Kom, selaku dosen pembimbing tugas akhir
5. Bapak. Ambar S.Kom, selaku pemilik Gemilang Jaya *Computer*

Penulis menyadari bahwa laporan tugas akhir ini masih jauh dari sempurna, oleh karena itu penulis mengharapkan masukan berupa saran atau kritik yang bersifat membangun dari semua pihak, untuk perbaikan dan pengembangan penelitian ini. Akhir kata penulis berharap agar laporan tugas akhir ini, dapat memberikan manfaat bagi perkembangan teknologi informasi.

Cianjur, 29 Oktober 2021

Penulis

# DAFTAR ISI

[LEMBAR PERNYATAAN KEASLIAN ii](#_Toc87910771)

[KATA PENGANTAR v](#_Toc87910772)

[DAFTAR ISI vi](#_Toc87910773)

[DAFTAR TABEL viii](#_Toc87910774)

[DAFTAR GAMBAR ix](#_Toc87910775)

[DAFTAR SIMBOL x](#_Toc87910776)

[A. Simbol *Use Case Diagram* x](#_Toc87910777)

[B. Simbol *Swimlane Diagram* xi](#_Toc87910778)

[C. Simbol *Class Diagram* xii](#_Toc87910779)

[D. Simbol *Sequence Diagram* xiii](#_Toc87910780)

[E. Simbol *Entity Relationship Diagram* xiv](#_Toc87910781)

[F. Simbol *Deployment Diagram* xv](#_Toc87910782)

[DAFTAR LAMPIRAN xvii](#_Toc87910783)

[BAB I 1](#_Toc87910784)

[PENDAHULUAN 1](#_Toc87910785)

[1.1 Latar Belakang Masalah 1](#_Toc87910786)

[1.2 Identifikasi Masalah 2](#_Toc87910787)

[1.3 Maksud dan Tujuan 2](#_Toc87910788)

[1.4 Batasan Masalah 2](#_Toc87910789)

[1.5 Tempat dan Waktu 3](#_Toc87910790)

[1.6 Metode Penelitian 9](#_Toc87910791)

[1.7 Sistematika Penulisan 11](#_Toc87910792)

[BAB II 13](#_Toc87910793)

[TINJAUAN PUSTAKA 13](#_Toc87910794)

[2.1 Tinjauan Pustaka 13](#_Toc87910795)

[2.1.1 Sejarah Toko Gemilang Jaya *Computer* 13](#_Toc87910796)

[2.1.2 Struktur Organisasi Toko Gemilang Jaya *Computer* 14](#_Toc87910797)

[2.1.3 Job Deskripsi 14](#_Toc87910798)

[2.2 Landasan Teori 15](#_Toc87910799)

[2.2.1 Definisi Pembangunan 15](#_Toc87910800)

[2.2.2 Definisi *E-Commerce* 15](#_Toc87910801)

[2.2.3 Definisi Penjualan 16](#_Toc87910802)

[2.2.4 Definisi Perangkat 16](#_Toc87910803)

[2.2.5 Definisi Komputer 16](#_Toc87910804)

[2.2.6 Definisi PHP (*Hypertext Preprocessor*) 16](#_Toc87910805)

[2.2.7 Definisi CSS (*Cascading Style Sheet*) 17](#_Toc87910806)

[2.2.8 Definisi HTML (*Hyper Text Markup Language*) 17](#_Toc87910807)

[2.2.9 Definisi Basis Data (*Database*) 18](#_Toc87910808)

[2.2.10 Definisi DBMS (*Database Management System*) 18](#_Toc87910809)

[2.2.11 Definisi UML (*Unified Modeling Language*) 18](#_Toc87910810)

[2.2.12 Definisi Motode Pengujian (*Blackbox*) 20](#_Toc87910811)

[2.3 *Tools* yang digunakan 21](#_Toc87910812)

[2.3.1 *Bootstrap* 21](#_Toc87910813)

[2.3.2 *Balsamiq Mockup* 22](#_Toc87910814)

[2.3.3 *Laravel* 22](#_Toc87910815)

[2.3.4 *Google Chrome* 23](#_Toc87910816)

[2.3.5 *Visual Studio Code* 24](#_Toc87910817)

[2.3.6 *MySQL* 25](#_Toc87910818)

[2.3.7 *Laragon* 26](#_Toc87910819)

[2.3.8 *Visual Paradigm* 27](#_Toc87910820)

[2.3.9 Raja Ongkir 28](#_Toc87910821)

[2.3.10 *Payment Gateway* 28](#_Toc87910822)

[2.3.11 *Corel Draw X8* 29](#_Toc87910823)

[DAFTAR PUSTAKA 31](#_Toc87910824)

# DAFTAR TABEL

[Tabel 1. 1 Jadwal Rencana Kegiatan 4](#_Toc89520872)

# DAFTAR GAMBAR

[Gambar 1. 1 *Prototype* Pressman 10](#_Toc89520847)

[Gambar 2.1 Struktur Organisasi Toko Gemilang Jaya *Computer* 14](#_Toc89520853)

[Gambar 2.2 *Use Case* *Diagram* 20](#_Toc89520854)

[Gambar 2. 3 *Activity Diagram* 20](#_Toc89520855)

[Gambar 2. 4 *Class Diagram* 21](#_Toc89520856)

[Gambar 2. 5 *Sequance Diagram* 22](#_Toc89520857)

[Gambar 2. 6 *Bootstrap* 23](#_Toc89520858)

[Gambar 2. 7 *Balsamiq Mockup* 24](#_Toc89520859)

[Gambar 2. 8 *Laravel* 25](#_Toc89520860)

[Gambar 2. 9 *Google Chrome* 26](#_Toc89520861)

[Gambar 2. 10 *Visual Studio Code* 27](#_Toc89520862)

[Gambar 2. 11 *Laragon* 28](#_Toc89520863)

[Gambar 2. 12 *Visual Paradigm* 28](#_Toc89520864)

[Gambar 2. 13 Raja Ongkir 29](#_Toc89520865)

[Gambar 2. 14 *Payment Gateway* 30](#_Toc89520866)

[Gambar 2. 15 *Corel Draw X8* 31](#_Toc89520867)

# DAFTAR SIMBOL

# Simbol *Use Case Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Usecase*** : menggambarkan apa yang dilakukan aktor melalui transaksi atau fungsi dari awal sampai akhir. |
|  | ***Actor*** : menggambarkan peran aktor baik itu manusia, media ataupun sistem lain. Disebut juga alat untuk komunikasi dengan bagian *usecase* dalam sistem. |
|  | ***Dependency*** : mengambarkan relasi antar dua model elemen jika satu berubah akan menyebabkan perubahan pada elemen lain. |
|  | ***Include*** : menggambarkan suatu *use case* yang tidak dapat dilakukan apabila *use case* lain belum dilakukan. |
|  | ***Extends*** : menggambarkan suatu *use case* yang dapat dilakukan apabila *use case* lain telah dilakukan, penambahan *use case* bersifat *fleksibel*  Daan  dapat dilakukan tanpa ada persyaratan tertentu. |
|  | ***Generalization*** : menggambarkan turunan dari suatu aktor, contohnya pelanggan ada pelanggan *member* dan *non member*. |
|  | ***System*** : menggambarkan lingkup suatu system |

# Simbol *Swimlane Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Activity*** : koordinasi eksekusi dari subordinat menggunakan kontrol atau *data flow model.* |
|  | ***Activity parameter node*** : aksi yang diterima dari aktifitas sebelumnya. |
|  | ***Action*** : merepresentasikan tahapan tunggal ketika memulai suatu aktifitas. |
|  | ***Decision node*** : menggambarkan sebuah pengkondisian , dari satu aliran *token* yang datang menjadi beberapa aliran yang keluar |
|  | ***Marge node* :** Mengabungkan dari beberap aliran yang telah di pecah oleh decision *node* |
|  | ***Fork node :*** menerima satu aliran untuk di bagi ke beberapa aliran *activity* |
|  | ***Join node* :** Mengabungkan bebearap *action* yang telah di pecah *fork node* |
|  | ***Initial node*** : mengawali suatu aktivitas yang akan dilakukan. |
|  | ***Activity final node*** : mengakhiri semua aktivitas yang dilakukan. |
|  | ***Expansion node*** : suatu aktifitas yang melewati batas. |
|  | ***Horizontal swimlane*** : digunakan untuk partisi *children.* |
|  | ***Control flow*** : aktivitas *node* setelah *node*  yang lain selesai. |

# Simbol *Class Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Class*** : merepresentasikan *entity* dalam *system.* |
|  | ***Generalization*** : ada diantara dua kelas dimana satu kelas *terspesifikasi*. |
|  | ***Aggregation*:**Hubungan yang menggambarkan suatu merupakan bagian dari kelas lain tapi bisa berdidri sendiri |
|  | ***Composition :***bentuk khusu dari agregasi jadi kelas yang menjadi part baru dapat di ciptakan setalah kelas lain yang menjadi *whole* dibuat. jika kelas yang menjadi *whole* dimusnakan kelas *part* akan hilang. |
|  | ***Usage*** : relasi antara dua elemen untuk implementasi atau operasi. |
|  | ***Association*** : relasi antar kelas. |
|  | ***N-ary association*** : model relasi beberapa kelas. |
|  | ***Association class*** : relasi antar kelas yang mempunyai atribut. |
|  | ***Dependency*** : relasi antar dua kelas dimana jika ada perubahan dalam satu kelas, maka akan mempengaruhi kelas lain. |
|  | ***Abstraction*** : konsep sama pada *level* berbeda. |

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Collaboration*** : grup elemen yang saling memiliki fungsi sama. |
|  | ***Model*** : kontainer untuk model elemen yang disimpan. |
|  | ***Note*** : area teks untuk menandai. |
|  | ***Anchor*** : konektor yang mengkoneksikan tambahan *note.* |
|  | ***Constraint*** : kondisi dibawah perilaku yang dieksekusi. |
|  | ***Containment*** : relasi yang menandakan ‘*part of relationship’.* |

# Simbol *Sequence Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Lifeline*** : suatu reprentasi *role* atau objek yang dalam model interaksi. |
|  | ***Message*** : definisi dari komunikasi antar *lifeline*. |
|  | ***Duration message*** : dapat menampilkan jarak antara dua *instance* waktu dalam pesan. |
|  | ***Create message*** : pesan yang merepresentasikan instansiasi *lifeline*. |
|  | ***Self message*** : pesan yang menampilkan operasi dalam objek *lifeline* yang sama. |
| **Gambar** | **Penjelasan** |
|  | ***Recursive message*** : merupakan tipe pesan yang dieksekusi secara rekursif. |
|  | ***Found message*** : merupakan pesan yang menerima *event* yang diketahui tapi tidak tahu pengirimannya. |
|  | ***Lost message*** : merupakan pesan yang diketahui *event*- nya tapi tidak tahu pengirimnya. |
|  | ***Reentrant message*** : merupakan *point* untuk aktivasi di atasnya. |

# Simbol *Entity Relationship Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Relationship line*** : Garis Penghubung yang digunakan pada diagram |
|  | ***Entitas*** : Digunakan untuk menggambarkan obyek yang dapat di identifikasikan dalam lingkungan pemakai |
|  | ***Atribut*** : Menggambarkan elemen-elemen dari suatu entitas yang menggambarkan entitas. |
|  | **Satu ke satu** : Satu entitas di dalam himpunan entitas A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B dihubungkan dengan paling banyak satu entitas dalam himpunan entitas A. |

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | **Satu ke banyak** : Satu entitas di dalam himpunan entitas A dihubungkan dengan lebih dari satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B hanya dapat dihubungkan dengan paling banyak satu entitas dalam himpunan entitas A. |
|  | **Banyak ke satu** : Satu entitas didalam himpunan entitas A dihubungkan dengan paling banyak satu entitas di dalam himpunan entitas B di dalam himpunan entitas B dapat dihubungkan dengan lebih dari satu entitas dalam himpunan entitas A. |
|  | **Banyak ke banyak** : Satu entitas di dalam himpunan entitas A dapat dihubungkan dengan lebih dari satu entitas di dalam himpunan entitas B, dan entitas di dalam himpunan entitas B dapat dihubungkan dengan lebih dari satu entitas dalam himpunan entitas A. |

# Simbol *Deployment Diagram*

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Node*** : merepresentasikan sumber daya komputasi tempat *artefak* dapat digunakan untuk eksekusi. |
|  | ***Artifact*** : Spesifikasi dari informasi fisik yang digunakan atau diproduksi oleh proses pengembangan perangkat lunak. |
|  | ***Deployment Specification*** : Sekumpulan *property* yang menentukan parameter |

|  |  |
| --- | --- |
| **Gambar** | **Penjelasan** |
|  | ***Component*** : Mewakili bagian modular dari sistem yang merangkum kontenya dan yang manifestasinya dapat diterima kembali di dalam lingkunganya. |
|  | ***Interfaces*** : mewakili deklarasi sekumpulan fitur dan kewajiban public yang koheren. |
|  | ***Port*** : Menentukan titik interaksi yang berbeda antara pengklasifikasi itu dan lingkunganya. |
|  | ***Found message*** : Diperpanjang dengan kemampuan menjadi target penerapan dalam hubungan penerapan. |
|  | ***Association*** : Mebentukan hubungan semantik yang dapat terjadi di antara *instance type*. |
|  | ***Port*** : Menentukan titik interaksi yang berbeda antara pengklasifikasi itu dan lingkunganya. |
|  | ***Link*** : Sebuah asosiasi menyatakan bahwa mungin ada tautan antara *instance* dari tipe terkait. |
|  | ***Dependency*** : relasi antar dua node dimana jika ada perubahan dalam satu node, maka akan mempengaruhi node lain. |
|  | ***Interaction use*** : ada diantara dua node dimana satu node terspesifikasi. |
|  | ***Note*** : area teks untuk menandai.. |

# DAFTAR LAMPIRAN

LAMPIRAN 1 : *SOURCE CODE* APLIKASI

# BAB I

# PENDAHULUAN

# Latar Belakang Masalah

Internet semakin pesat dalam perkembangannya, semua orang dimudahkan dalam mendapatkan informasi secara mudah dan cepat, dengan perkembangan tersebut tentunya perkembangan aplikasi *web* pun mengikuti perkembangan pada masa kini, dalam penjualan barang, intenet dibutuhkan untuk mempermudah orang dalam transaksi,serta *web* *browser* semakin mudah di akses oleh semua orang melalui jaringan internet yang bersifat *public.*

Gemilang Jaya *Computer* merupakan toko yang bergerak di bidang perdagangan besar peralatan komputer seperti laptop, *monitor, keyboard, mouse, flashdisk, vga, speker* dan masih banyak produk lainnya,. Transaksi penjualan pada Gemilang Jaya *Computer* dilakukan dengan datang ke toko, transaksi tersebut tidak dapat di jangkau secara luas. Ketidakmudahan *customer* dalam mendapatkan informasi secara cepat, mengenai informasi produk yang tersedia, karena promosi produk masih menggunakan brosur atau media cetak. Proses pemesanan membutuhkan waktu lama, karena tidak ada sistem atau aplikasi.

Dengan masalah diatas, tentunya pemanfaatan internet dan aplikasi *web* sangat dibutuhkan dalam proses penjualan barang pada **Gemilang Jaya *Computer***, diharapkan dengan proses tersebut, kepuasan dan kenyamanan *customer* dalam membeli barang meningkat, transaksi dapat dijangkau atau diakses oleh semua orang dimanapun dan kapanpun, dengan perkembangan aplikasi berbasis *web* saat ini, akan mempermudah proses bisnis yang berjalan tentunya mempermudah proses bisnis yang ada.

Untuk mempermudah informasi, proses promosi, pemesanan, serta transaksi penjualan barang yang dibutuhkan *customer*, maka penulis berencana untuk membuat **PEMBANGUNAN *E-COMMERCE* PENJUALAN PERANGKAT KOMPUTER DI GEMILANG JAYA *COMPUTER*,** dengan aplikasi tersebut, diharapkan transaksi *customer* akan lebih mudah, serta akan mempermudah

operasional kerja dari Gemilang Jaya *Computer* dalam penjualan dan penawaran produk, dengan tujuan untuk meningkatkan pelayanan kepada *customer*.

# Identifikasi Masalah

Berdasarkan latar belakang yang penulis sampaikan diatas, maka dapat di identififkasi permasalahan yaitu sebagai berikut:

1. Bagaimana membangun sistem atau aplikasi, yang dapat memberikan kemudahan mengenai informasi produk.
2. Bagaimana membangun sistem atau aplikasi, yang dapat memberikan kemudahan dalam proses pemesanan barang.
3. Bagaimana membangun sistem atau aplikasi, yang dapat memberikan kemudahan dalam proses transaksi penjualan barang.

# Maksud dan Tujuan

1. Maksud

Maksud dari penelitian ini adalah untuk membangun aplikasi *e-commerce* di **Toko** **Gemilang Jaya *Computer***.

1. Tujuan

Adapun tujuan dari pembuatan aplikasi *e-commerce* ini, adalah sebagai berikut:

1. Membuat aplikasi yang memberikan kemudahan mendapatkan informasi produk di Gemilang Jaya *Computer.*
2. Membuat aplikasi yang memberikan kemudahan proses pemesanan barang di Gemilang Jaya *Computer*.
3. Membuat aplikasi yang memberikan kemudahan proses transaksi penjualan barang di Gemilang Jaya *Computer.*

# Batasan Masalah

Adapun batasan masalah dari aplikasi *e-commerce* ini adalah sebagai berikut:

* 1. Aplikasi atau sistem yang akan dibangun, tidak akan membahas mengenai keamanan jaringan.
  2. Apalikasi atau sistem yang akan dibangun, berbasis *web*
  3. Aplikasi atau sistem yang akan dibangun, penjualan hanya dapat dilakukan oleh Gemilang Jaya *Computer* sedangkan pembeli bersifat *public* atau

umum.

* 1. Aplikasi atau sistem yang akan dibangun, tidak akan membahas *return* atau pengembalian barang, karena barang yang dikirim sudah dalam tahap proses pengecekan.
  2. Aplikasi atau sistem yang akan dibangun, tidak akan membahas *tracking* barang.
  3. Aplikasi atau sistem yang akan dibangun terintegrasi dengan raja ongkir
  4. Aplikasi atau sistem yang akan dibangun, tidak akan membahas atau menggunakan *Google API/ Google Map API*.
  5. Aplikasi atau sistem yang akan terintegrasi dengan *payment gateway*
  6. Aplikasi atau sistem yang akan dibangun, tidak akan membahas *voucher*,.
  7. Aplikasi atau sistem yang akan dibangun, menggunakan diskon produk.
  8. Aplikasi atau sistem yang akan dibangun, tidak akan membahas atau menggunakan sistem *COD (Cash On Delivery)*.
  9. Aplikasi atau sistem yang akan dibangun tidak terintegrasi dengan kasir, integrasi hanya dengan *payment gateway*.
  10. Aplikasi atau sistem yang akan dibangun, tidak akan membahas mengenai penglolaan barang secara *detail* serta laporan *stock* barang, aplikasi berfokus pada proses transaksi bisnis dengan *virtual account.*

# Tempat dan Waktu

Tempat penelitian Tugas Akhir dilaksanakan sebagai berikut :

Nama Instansi : Gemilang Jaya *Computer*

Alamat : Jl.Balakang, Sindanglaya, Kec.Cipanas

Kabupaten.Cianjur, Jawa Barat 43253

Waktu : November 2021 – Januari 2021

Tabel 1. 1 Jadwal Rencana Kegiatan

|  |  |  |  |
| --- | --- | --- | --- |
| No | Uraian Langkah Kegiatan | Deskripsi dan Waktu Kegiatan | |
| Deskripsi | November |
|
| 1 | Identifikasi | Mengidentifikasi permasalah  yang terdapat, di tempat  penelitian dengan  beberapa tahapan yang  dilakukan yaitu  1. Observasi  2. Wawancara | Tahapan Identifikasi yaitu   1. Tahapan observasi:   Datang ke tempat  penelitian serta mengambil  data yang dibutuhkan   1. Tahapan wawancara:    1. Proses bisnis yang   berjalan meliputi :   1. Proses bisnis   menyeluruh   1. Fokus pada   proses bisnis yang  akan dianalisis :   1. Proses informasi   barang   1. Proses pemesanan   barang   1. Proses transaksi   barang   1. Masalah pada proses   bisnis tersebut |
|  |
| 2 | Pengumpulan data | Mengumpulkan data atau  informasi dari berbagai sumber baik tempat penelitian, buku  perpustakaan, jurnal, *internet*  dan sumber lain, yang  menyatakan informasi atau  data *valid,* sebagai bahan  referensi untuk penelitian,  adapun tahapan nya  yaitu :   1. Tahahan pengumpulan data | Tahapan Pengumpulan data yaitu:   1. Tahapan pengumpulan data   Mengumpulkan data dari berbagai sumber:   1. Data yang di dapat dari tempat penelitian 2. Sumber *Internet* yang diambil dari jurnal dan karya ilmiah. 3. Buku dari perpustakaan |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| No | Uraian Langkah Kegiatan | Deskripsi dan Waktu Kegiatan | |
| Deskripsi | November |
|
| 3 | Analisis | Penguraian dari suatu data atau informasi yang utuh ke dalam bagian-bagian komponennya  dengan maksud untuk  mengidentifikasikan dan  mengevaluasi permasalahan-  permasalahan, kesempatan,  hambatan yang terjadi dan  kebutuhan yang diharapkan  sehingga dapat diusulkan  perbaikan- perbaikannya,  tahapan analisi, tahapan  analisis pada penelitian  pembangunan  *e-commerce* yaitu :   1. Analisis data atau   Informasi yang  didapatkan   1. Analisis proses bisnis   yang berjalan   1. Analisis sitem yang   akan dibangun | Tahapan Analisis yaitu :   1. Tahapan mengelola data atau informasi yang didapatkan kemudian dianalisa berdasarkan data atau informasi yang didapatkan 2. Tahapan mengelola data, proses bisnis yang berjalan kemudian dianalisa berdasarkan data yang didapatkan 3. Tahapan analisa perencanaan dari data yang di dapat,mengelola data menjadi rancangan sistem yang akan dibangun |
|  |
| 4 | *Design* | Tampilan *visual* sebuah produk yang menjembatani sistem  dengan pengguna *(user).*  tampilan *ui* dapat berupa  bentuk, warna, dan tulisan  yang didesain semenarik  mungkin. Secara sederhana, *ui* adalah bagaimana tampilan  sebuah produk dilihat oleh  pengguna, tahapan *ui* yang  akan dibangun *e-commerce* yaitu:   1. Perancangan *ui* 2. *Tools* yang digunakan | Tahapan *Design* yaitu :   1. Perancangan UML meliputi : 2. *Membuat actor*   *list,usecase, secenario,*  *usecase list.*   1. *Membuat usecase*   *diagram*   1. *Membuat activity*   *diagram*   1. *Membuat sequence*   *Diagram*   1. *Membuat class diagram* |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| No | Uraian Langkah Kegiatan | Deskripsi dan Waktu Kegiatan | |
| November | Desember |
|
| 4 | *Design* |  | 1. *Membuat entity relation diagram* 2. Membuat *struktur table* 3. Membuat *struktur menu* 4. Membuat perancangan *deployment diagram* 5. *Tools* yang digunakan yaitu *ballsamiq mockup* dan *corel draw x8* 6. Pembuatan *ui*   berdasarkan fungsi pada  sistem atau aplikasi yang  akan dibangun   1. Membuat *user*   *interface* dengan  *ballsamiq mockup*  yaitu :   1. Membuat *ui*   sesuai perencanaan   1. Konfirmasi kepada   *client,* terhadap *ui*  yang dibuat.   1. Jika *ui* sesuai tidak sesuai dengan   perancangan dan  harapan *client,*  maka dilakukan  *revisi* atau perbaikan pada tahap  perencanaan,jika  sesuai maka  dilakukan ke  tahap proses  selanjutnya. |
|  |

|  |  |  |  |
| --- | --- | --- | --- |
| No | Uraian Langkah Kegiatan | Deskripsi dan Waktu Kegiatan | |
| Deskripsi | Januari |
|
| 5 | Implementasi | Penerapan aplikasi dari hasil perancangan sistem yang ada untuk mencapai suatu tujuan  yang diinginkan. Implementasi  melaksanakan perintah  yang secara terstruktur dari  awal sampai akhir. Pada  implementasi ini juga akan  mengimplementasikan sebuah  perangkat lunak atau aplikasi  yang akan dibuat, yaitu   1. Implementasi Aplikasi 2. Implementasi *Design* | Tahapan Implementasi yaitu :   1. Implementasi Aplikasi    * + 1. Pembuatan aplikasi   berdasarkan analisis  dan *ui* yang telah  disetujui oleh *client*,  *tools* yang digunakan   * + 1. Bahasa   pemrograman *php,*  *javascript, Jquery*   * + 1. Menggunakan *framework laravel,*   *bootsrap*   * + 1. *Integrasi* raja   ongkir dan  *payment gateway*   1. Implementasi *Design* 2. Membuat *design*   dengan *corel draw x8*  yaitu :   1. Membuat *banner*   serta *design* yang  dibutuhkan untuk  tampilan *web* |
|  |
| 6 | Pengujian | Menjalankan dan mengevaluasi sebuah perangkat lunak secara *manual* maupun *otomatis*  untuk menguji apakah  perangkat lunak sudah  memenuhi persyaratan atau  belum, atau untuk menentukan  perbedaan antara hasil yang  diharapkan dengan hasil  sebenarnya. Peksanaan  pengujian perangkat lunak  disesuaikan dengan metodologi pembangunan perangkat lunak yang digunakan, tahapan  pengujian yaitu :   1. Pengujian aplikasi | Tahapan Pengujian   1. Pengujian aplikasi   berdasarkan keahlian dan  hasil dari  penelitian terhadap  produk yang  dihasilkan  yaitu :   1. Pengujian *ui* 2. Pengujian *blackbox* 3. Pengujian berdasarkan   analis, fungsi dan  kesesuaian terhadap hasil  dari pemodelan |
|  |
| No | Uraian Langkah Kegiatan | Deskripsi dan Waktu Kegiatan | |
| Deskripsi | Januari |
|
| 7 | Penyusuna laporan | bagian penting dalam suatu  informasi di setiap rencana  kerja yang akan dibangun  adalah laporan. Dalam  membuat laporan disesuaikan  dengan proses pekerjaan yang  telah dilakukan, tahapan  penyusunan laporan tugas  akhir ini diharapkan  bermanfaat bagi pembaca  sehingga dapat menilai  laporan yang disajikan apakah  sudah baik dan dapat  dimengerti atau belum | Tahapan Penyusunan  Laporan Tugas Akhir  yaitu:   1. Komunikasi 2. Perencanaan 3. Pemodelan 4. Kontruksi 5. Penyerahan |
|  |
| 8 | Seminar |  | Seminar Tugas Akhir dilaksanakan, menguji hasil analisis yang dihasilkan, diuji oleh 4 orang dosen penguji, pengujian berdasarkan fungsi, kesesuaian dan analisis yang telah dirancang. |

# Metode Penelitian

* 1. Adapun metode pengumpulan data yang penulis gunakan sebagai berikut:

1. Wawancara, merupakan suatu proses komunikasi, dimana kegiatan tanya jawab yang dilakukan oleh seseorang dengan tujuan mendapatkan data atau informasi yang akurat serta menghindari kesalahan informasi, antara pihak ke satu dengan pihak kedua atau pihak pembuat sistem dan *client*.
2. Observasi, merupakan suatu proses pengumpulan data langsung, serta pihak pengumpul data berada di tempat, untuk melihat dan mengambil data atau informasi yang dibutuhkan oleh pembuat aplikasi.
3. Tinjauan Pustaka, yaitu suatu proses untuk mencari suatu data atau informasi dari buku atau para ahli,.
   1. Metode Rekayasa Perangkat Lunak

Metode yang digunakan pada aplikasi *ecommerce* ini yaitu dengan menggunakan metode rekayasa perangkat lunak *prototype*, karena merupakan model sederhana cocok digunakan untuk pengembangan perangkat lunak, dengan melakukan pendekatan dengan *client*, mulai dari komunikasi, perenecanan, pemodelan, kontruksi, sampai pada tahap penyerahan dari aplikasi yang akan dibuat, setiap perubahan di deskripsikan secara lengkap kepada *client*, kemudian *client* akan merevisi jika rancangan aplikasi tidak sesuai, jika sesuai maka akan dilakukan ke tahap berikutnya.

Metode *Prototype* merupakan pengembangan perangkat lunak dengan interaksi atau kominikasi antara pembuat sistem dan pengguna sistem, tujuan dari pendekatan tersebut untuk meminimalisir kesalahan data atau informasi antara pembuat sistem dan pengguna sistem (Pressman, 2012: 50). Adapun model pengembangan Prototype digambarkan pada gambar 1.1.

|  |
| --- |
|  |
| Gambar 1. 1 *Prototype* Pressman |

Pembuat sistem akan melakukan pertemuan dengan pengguna sistem atau *client,* dengan utjuan untuk mendeskripsikan secara utuh mengenai sistem atau perangkat lunak yang akan dibangun atau dikembangkan, melakukan *survei* hingga mengidentifikasi kebutuhan yang dibutuhkan untuk membangun sistem, serta menggambarkan *functional area* yang dideskripsikan

Berikut tahap-tahap pada metode pengembangan sistem yang digunakan pada penelitian :

1. Komunikasi, tahapan komunikasi dilakukan dengan melakukan wawancara antara pembuat sistem dengan pemiliki Toko Gemilang Jaya *Computer* yaitu bapak Ambar, observasi datang ke tempat penelitian, meminta data yang dibutuhkan, melakukan wawancara di mulai dari pembahasan proses penawaran barang, proses pemesanan barang, proses transaksi barang, hingga proses bisnis yang berjalan, dari proses komunikasi tersebut munculah suatu permasalahan yang penulis analis serta jadikan topik untuk perencanaan atau langkah selanjutnya yang harus penulis ambil, dalam menangani permasalahan tersebut,.

2. Perencanaan, setelah melalui tahapan komunikasi, dilakukan perencanaan berdasarkan data yang telah didapatkan, membuat *UML* seperti *use case secenario, use case diagram, swimeline diagram, activity diagram, class diagram,*dan *entity relation diagram,* dengan *tools visual paradigm,* perencanan berdasarkan kesesuaian data dan keahlian yang dimiliki, dengan tujuan pengembangan perangkat lunak sesuai kebutuhan sistem, berdasarkan komunikasi dengan harapan sesuai perencanaan yang telah dibuat.

3. Pemodelan, setelah melalui tahapan komunikasi dan perencanaan, tahapan selanjutnya dilakukan pemodelan, berdasarkan perencanaan yang dibuat kemudian digambarkan atau dimodelkan melalui *user interface* menggunakan *tools* *ballsamiq mockup,* setiap proses aktifitas sistem yang akan dibangun dimodelkan, jika pemodelan tidak sesuai dengan rancangan dan harapan *client* maka dilakukan revisi atau perubahan ke tahap perencanaan, ketika pemodelan sudah sesuai harapan *client* maka dilakukan proses selanjutnya yaitu pembuatan aplikasi atau *coding* dengan menggunakan bahasa pemrograman *php, javascript, jquery,* dengan *framework laravel*, *bootsrap*, dengan integrasi pihak ketiga yaitu raja ongkir dan *payment gateway* yang implementasinya akan menjadi sistem atau aplikasi yang telah dimodelkan.

4. Konstruksi, tahapan selanjutnya setelah tahapan pemodelan yaitu kontruksi atau tahapan pengujian sistem, pengujian aplikasi atau sistem dilakukan berdasarkan proses yang telah di rancang atau hasil analisis baik berdasarkan fungsi atau *performace* aplikasi, dengan melakukan serangkaian *testing black box, user interface,* dan fungsi hal tersebut dilakukan agar produk atau aplikasi yang dihasilkan dapat berjalan sesuai dengan perencanaan,.

5. Penyerahan, tahapan ini merupakan tahapan akhir dimana hasil dari analisis berupa sistem atau aplikasi yang telah dibuat diserahkan kepada *client* atau Gemilang Jaya *Coomputer,* dengan harapan terjadinya *feedback* dari pemilik toko kepada pembuat sistem atau aplikasi*,* serta kepuasan dari Gemilang Jaya *Computer* terhadap sistem yang telah dibangun, melalui proses yang telah dilalui dan saling menguntungkan.

# Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini dibagi menjadi 5 (lima) bab. Berikut penjelasan mengenai masing-masing bab :

**BAB I PENDAHULUAN**

Bab ini menjelaskan mengenai latar belakang masalah, indetifikasi masalah, maksud dan tujuan penelitian, batasan masalah yang berguna untuk memfokuskan pada pembangunan sistem atau aplikasi dengan harapan mempermudah proses transaksi, mempermudah informasi atau pomosi yang didapatkan *customer,* serta mempermudah proses pemesanan barang di **Gemilang Jaya *Computer,*** sehingga dapat mencapai hasil yang optimal.

**BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan mengenai teori-teori yang mendukung laporan tugas akhir. Teori berdasarkan para ahli ataupun buku sebagai referensi, pembahasan landasan teori ini meliputi penjelasan tentang definisi-definisi, penjelasan mengenai perangkat lunak yang digunakan dalam perancangan atau pembangunan sistem dan istilah istilah yang terkait didalamnya.

**BAB III ANALISIS DAN PERANCANGAN**

Bab ini menjelaskan uraian tentang analisa sistem yang akan dibuat dan bagaimana merancangnya, sehingga menjadi suatu sistem atau aplilkasi. Serta membuat uraian tentang analisa sistem dengan menggunakan *Unified Modelling Language (UML)* seperti *use case diagram, swimlane diagram, class diagram, sequence diagram, erd* yang akan segera dibuat dan bagaimana merancangnya sehingga menjadi suatu sistem.

**BAB IV IMPLEMENTASI & PENGUJIAN**

Bab ini menjelaskan mengenai pembuatan sistem, yang merupakan pengimplementasian dari hasil analisa dan perancangan yang berisi gambaran struktur menu serta tampilan layar dan langkah pengujian dari sistem yang akan dibangun.

**BAB V PENUTUP**

Bab ini berisi kesimpulan tentang seluruh pembahasan dan pemecahan masalah yang telah dilakukan dalam kegiatan penelitian Tugas Akhir yang telah dilakukan.

# BAB II

# TINJAUAN PUSTAKA

# Tinjauan Pustaka

Tinjauan pustaka ini membahas tentang gambaran toko, tempat *survey* yang didalamnya mencakup sejarah singkat toko, struktur organisasi toko dan deskripsi kerja.

# Sejarah Toko Gemilang Jaya *Computer*

CV. Gemilang Jaya adalah perusahaan yang bergerak di bidang general *suplier* Barang dan Jasa *service*, perdagangan besar komputer, dan perlengkapan komputer seperti laptop, *monitor, keyboard, mouse, flashdisk, vga,* speker dan masih banyak produk lainnya yang melayani perusahaan menengah dan perusahaan besar, baik pemerintahan dan swasta.

Penjualan produk komputer dengan kualitas terbaik dan bergaransi, penjualan berbagai *brand* terkenal seperti *Asus, Acer , Toshiba , Dell* dan lain sebagainya, serta melayani *service,* dengan Standard Operasional Perusahaan atau *SOP*.

Perusahaan kami sudah berjalan dari tahun 2010, Tujuan kami adalah untuk menjadi pilihan utama bagi mitra bisnis kami dengan memberikan kontribusi kepada setiap *client,* melebihi dari yang mereka  harapkan, melalui pelayanan istimewa dari kami secara profesional terus ditingkatkan dan integritas penuh, dalam menjalankan bisnis di dukung oleh sumber daya  yang berkualitas dan berpengalaman untuk menghasilkan kualitas pekerjaan dan pelayanan yang baik sesuai yang di harapkan mitra bisnis kami.

# Visi dan Misi Gemilang Jaya *Computer*

Visi Menjadi perusahaan yang berkembang, bermanfaat dan dipercaya untuk melayani perdagangan barang dan jasa serta,.

Misi membangun bisnis secara *terintegrasi* guna memberikan manfaat dan pelayanan yang luas dan memberikan pelayanan yang baik dan saling menguntungkan bagi pengguna jasa perdagangan yang menciptakan hubungan kekeluargaan

# Struktur Organisasi Toko Gemilang Jaya *Computer*

|  |
| --- |
|  |
| Gambar 2.1 Struktur Organisasi Toko Gemilang Jaya *Computer* |

# Job Deskripsi

Berikut ini merupakan deskripsi pekerjaan dari Struktur Organisasi Gemilang Jaya *Computer* sebagai berikut:

1. Manager
2. Mengotrol semua aktifitas organisasi.
3. Melakukan mediasi dan mengatur proses kerjasama dengan instansi lain.
4. Sebagai pusat koordinasi.
5. Bendahara
6. Melaksanakan pengujian ketersediaan dana yang dibutuhkan dan

pembayaran tagihan

1. Mengelola uang persediaan/pembayaran langsung
2. Melaksanakan tugas wajib pungut pajak
3. Mengerjakan pembukuan dan pertanggung jawaban bendahara

Pengeluaran

1. Melaksanakan penata usahaan dan pengarsipan surat kedinasan, SPJK, SPP, SP2D dan dokumen-dokumen keuangan lainnya
2. Marketing
3. Mengumpulkan informasi mengenai pelanggan, pesaing, serta pelaku dan kekuatan lain yang ada saat ini maupun yang potensial dalam lingkungan pemasaran
4. Mengembangkan dan menyebarkan komunikasi persuasif untuk

membujuk pembelian.

1. Mengatur kesinambungan penyimpanan dan pergerakan produk sampai

ke pelanggan akhir

1. Teknis 1
2. Melakukan pengecekan barang sesuai prosedur perusahaan atau *S.O.P*
3. Bertugas dan bertanggung jawab untuk melaksanakan, mengelola,

mengembangkan, dan memperbaiki perangkat keras dan lunak untuk infrastruktur

1. Memperbaiki layanan di unit kerja lancar dan tersedia dengan baik.
2. Teknis 2
3. Melakukan pengujian barang sesuai prosedur perusahaan atau *S.O.P*
4. Melakukan pengujian produk sebelum diterima *customer*
5. Bertanggung jawab terhadap sarana dan prasarana barang yang berada dilingkungan perusahaan

# Landasan Teori

# Definisi Pembangunan

Pembangunan menurut Rogers (Rochajat,dkk: 2011:3) merupakan perubahan sistem yang berguna untuk sistem sosial atau ekonomi untuk suatu negara. Selanjutnya menurut W.W Rostow (Abdul: 2004:89) pembangunan adalah suatu proses yang bergerak dalam sebuah garis lurus, yakni dari masyarakat terbelakang ke masyarakat negara yang maju.

Dari definisi diatas penulis dapat menyimpulkan pembangunan merupakan suatu proses perubahan sistem yang dapat memberikan reaksi baik dari sosial, ekonomi, politik, teknologi dengan tujuan membangun sistem.

# Definisi *E-Commerce*

Menurut Kotler & Amstrong (2012) *E-commerce* adalah saluran *online* yang dapat dijangkau seseorang melalui komputer, yang digunakan oleh pebisnis dalam melakukan aktifitas bisnisnya dan digunakan konsumen untuk mendapatkan informasi dengan menggunakan bantuan komputer yang dalam prosesnya diawali dengan memberi jasa informasi pada konsumen dalam penentuan pilihan.

Menurut Wong (2010) *e-commerce* adalah proses jual beli dan memasarkan barang serta jasa melalui sistem elektronik, seperti radio, televisi dan jaringan komputer atau internet.

Dari definisi diatas penulis dapat menyimpulkan *e-commerce* merupakan penjualan, pembelian, serta pemasaran barang atau produk dengan bantuan sistem atau aplikasi dengan perantara internet atau jaringan komputer.

# Definisi Penjualan

Menurut Basu Swastha (2001:8) penjualan merupakan ilmu dan seni mempengaruhi pribadi yang dilakukan oleh penjual untuk mengajak orang lain agar bersedia membeli barang yang ditawarkan. Adapun faktor-faktor yang mempengaruhi penjulalan adalah kondisi dan kemampuan penjual, kondisi pasar, modal, dan kondisi organisasi toko.

Dari definisi diatas penulis dapat menyimpulkan bahwa penjualan, merupakan suatu kegiatan menjual produk barang atau jasa dengan tujuan utama yaitu mendatangkan keuntungan.

# Definisi Perangkat

Menurut Kamus Besar Bahasa Indonesia (KBBI), (2002:8) arti kata perangkat adalah alat perlengkapan. Perangkat memiliki arti dalam kelas nomina atau kata benda sehingga perangkat dapat menyatakan nama dari seseorang, tempat, atau semua benda dan segala yang dibendakan.

Dari definisi diatas penulis dapat menyimpulkan, perangkat merupakan suatu benda atau *object* berupa perlengkapan yang dapat digunakan.

# Definisi Komputer

Menurut Sujatmiko (2012:156) komputer adalah mesin yang dapat mengolah data digital dengan mengikuti serangkaian perintah atau program. Menurut Sutanta (2011:01) komputer berasal dari bahasa latin, yaitu komputare yang berarti menghitung *(to compute/to reckon*).

Dari definisi diatas penulis dapat menyimpulkan, komputer merupakan suatu benda atau mesin yang dapat mengelola data digital dan membantu proses pekerjaan.

# Definisi PHP (*Hypertext Preprocessor*)

PHP adalah bahasa pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan. (Sibero, 2011) “*PHP*” (*Hypertext Preprocessor*) PHP adalah sebuah bahasa pemrograman yang berbentuk *scripting*, sistem kerja dari program ini adalah sebagai *interpreter* bukan sebagai *compiler*”. (Nugroho, 2013)

Dari definisi diatas penulis dapat menyimpulkan bahwa, bahasa pemrograman *php* merupakan bahasa pemrograman yang digunakan untuk mengembangkan situs *web* statis atau situs *web* dinamis atau Aplikasi *Web*. *PHP* singkatan dari *Hypertext Pre-processor*, yang sebelumnya disebut *Personal Home Pages*.

# Definisi CSS (*Cascading Style Sheet*)

Berdasarkan pendapat yang telah dikemukakan dapat disimpulkan bahwa CSS (*Cascading Style Sheets*) adalah salah satu jenis bahasa pemrograman untuk mengatur proses pengolahan pada komponen tampilan *web* menjadi bentuk web yang lebih indah dan menarik. (Bekti, 2015)

CSS merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar. (Jayan, 2010).

Dari definisi diatas penulis dapat menyimpulkan CSS, merupakan *script code* yang berfungsi untuk mengatur *layout web*, tulisan, warna, seluruh *style* tampilan halaman *web*.

# Definisi HTML (*Hyper Text Markup Language*)

*Hyper Text Markup Language* atau HTML adalah bahasa yang digunakan pada dokumen *web* sebagai bahasa untuk pertukaran dokumen *web*. (Sibero, 2013), HTML (*Hyper Text Markup Language*) adalah simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang dimaksudkan untuk menampilkan halaman pada *web browser*.

Tag-tag HTML selalu diawali dengan dan diakhiri dengan dimana x tag HTML seperti b, i, u dan sebagainya. Namun ada juga tag yang tidak diakhiri dengan tanda, Astamal (2006:8)

Dari definisi diatas penulis dapat menyimpulkan HTML, merupakan *script code* yang berfungsi untuk mengatur *layout web*, tulisan, seluruh struktur *web* seperti *header, content,* dan *footer* pada tampilan halaman *web*.

# Definisi Basis Data (*Database*)

*Database* adalah koleksi terpadu dari data-data yang saling berkaitan dari suatu *enterprise* (toko, instansi pemerintahan atau swasta). (Ladjamudin, 2013), *Database* (Basis Data) merupakan sekumpulan data yang sangat kompleks, kemudian data tersebut memiliki hubungan antara data yang satu dengan data yang lainya (Nugroho, 2013),.

Dari definisi diatas penulis dapat menyimpulkan *Database*, merupakan sekumpulan data yang tersimpan secara digital yang mempunyai hubungan antara satu dengan yang lain,.

# Definisi DBMS (*Database Management System*)

DBMS adalah “*Database Management System* (DBMS) merupakan kumpulan program aplikasi yang digunakan untuk mengelola basis data. Taufik, 2013). DBMS adalah kumpulan program yang digunakan untuk mendefinisikan, mengatur, dan memproses *database*, sedangkan *database* itu sendiri esensinya adalah sebuah struktur yang dibangun untuk keperluan penyimpanan data. (Budi Raharjo, 2011).

Dari definisi diatas penulis dapat menyimpulkan DBMS , merupakan suatu tempat atau sekumpulan berkas data yang sudah terkomputerisasi yang bertujuan untuk memelihara data atau informasi.

# Definisi UML (*Unified Modeling Language*)

Menurut Hend (2006:5) “*Unified Modeling Language* (UML) adalah bahasa yang telah telah menjadi *standard* untuk *visualisasi*, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”. UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung,

Misanya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. (He UML adalah *metodologi kolaborasi antara metoda-metoda Booch, OMT* (*Object Modeling Technique*), serta *OOSE* (*Object Oriented Software Enggineering*) dan beberapa metoda lainnya, merupakan metodologi yang paling sering digunakan saat ini untuk analisa dan perancangan sistem dengan metodologi berorientasi objek mengadaptasi maraknya penggunaan bahasa ‘pemrograman berorientasi objek’ (OOP). (Nugroho, 2009) rlawati, 2011).

Diagram berbentuk grafik yang menunjukan simbol elemen model yang

disusun untuk mengilustrasikan bagian atau aspek tertentu dari sistem. Sebuah diagram merupakan bagian dari suatu *view* tertentu dan ketika digambarkan biasanya dialokasikan untuk *view* tertentu. Adapun jenis-jenis diagram antara lain:

1. *Use Case Diagram*

*Use Case Diagram* merepresentasikan sejumlah *external actors* dan hubungannya ke *use case* yang diberikan oleh sistem. *Use case* adalah menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case mempresentasikan sebuah interaksi antara aktor dengan sistem.

Berikut ini merupakan contoh gambar use case:

|  |
| --- |
| Use Case Diagram: Pengertian, Simbol, Komponen &amp; Contohnya |
| Gambar 2.2 *Use Case* *Diagram* |

* 1. *Activity Diagram*

Diagram ini menggambarkan rangkaian aliran dari aktivitas didalam sebuah sistem yang bersifat dinamis, diagram ini digunakan untuk mendeskripsikan aktifitas yang dibentuk dalam suatu operasi sehingga dapat juga digunakan untuk aktifitas lainnya seperti use case atau interaksi.

Berikut ini merupakan contoh gambar *activity* *diagram*:

|  |
| --- |
|  |
| Gambar 2. 3 *Activity Diagram* |

* 1. *Class Diagram*

*Class Diagram* adalah pandangan aplikasi yang bersifat *statis*. *Class Diagram* tidak hanya menggambarkan *visualisasi*, tetapi juga menggambarkan dan mendokumentasikan aspek yang berbeda dalam sistem, tetapi juga untuk kontruksi eksekusi kode dalam software atau aplikasi.

Berikut ini merupakan contoh gambar *class* *diagram*:

|  |
| --- |
|  |
| Gambar 2. 4 *Class Diagram* |

*Class Diagram* digunakan untuk mengelompokan hal-hal inti dari setiap proses yang ingin dilakukan. Semua proses dimasukkan ke dalam tiap-tiap *class* dan saling dihubungkan pada *class-class* lainnya yang saling berhubungan.

* 1. *Sequence Diagram*

Diagram ini menggambarkan kolaborasi dinamis antara sejumlah *object* yang bersifat dinamis. Kegunaanya untuk menunjukkan rangkaian pesan yang dikirim terhadap object juga interaksi antara *object*, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem.

Berikut ini merupakan contoh gambar *sequance* *diagram*:

|  |
| --- |
|  |
| Gambar 2. 5 *Sequance Diagram* |

# Definisi Motode Pengujian (*Blackbox*)

“*Blackbox* *testing* merupakan tipe *testing* yang memperlakukan perangkat lunak yang tidak diketahui kinerja *internal*nya”. Sedangkan menurut Mustaqbal, dkk (2015:34) “*BlackBox Testing* befokus pada spesifikasi fungsional dari perangkat lunak, kumpulan kondisi input dan melakukan pengetesan pada fungsional program”. Menurut Rizky (2011:264) “*Blackbox testing* merupakan menguji perangkat lunak dari segi spesifikasi fungsional tanpa menguji desain dan kode program”. Menurut Rosa dan Salahuddin (2015:275).

Menurut Simarmata Janner (2010 : 316) Klasifikasi *blackbox testing* mencakup beberapa pengujian yaitu :

1. Pengujian fungsional (*functional testing*)
2. Pengujian tegangan (*stress testing*)
3. Pengujian beban (*load testing*)
4. Pengujian khusus (*ad-hoc testing*)

Dari definisi diatas penulis dapat menyimpulkan metode pengujian *Blackbox*, merupakan metode untuk menguji kelancaran program yang telah dibuat sebelumnya. Pengujian dilakukan agar program tidak terjadi kesalahan ketika dijalankan.

# *Tools* yang digunakan

Untuk memudahkan dalam pembuatan sistem, maka diperlukan beberapa alat atau *tools* yang digunakan dalam pembangunan *e-commerce* penjualan perangkat computer di Gemilang Jaya *Computer*, serta untuk proses pelaksanaan pembuatan sistem yang akan dibangun.

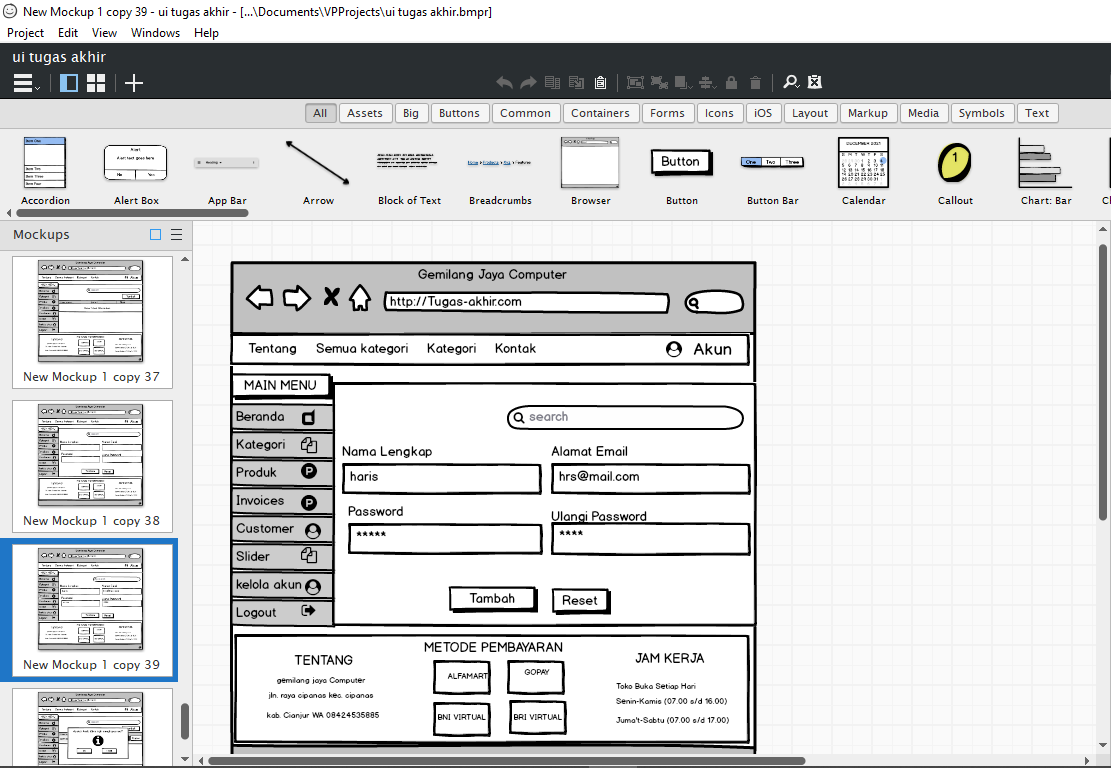
## *Bootstrap*

Menurut para ahli *Bootstrap* merupakan kerangka kerja *front-end* atau *design* untuk pengembangan *web* yang lebih cepat dan mudah tentunya gratis. *Bootstrap* mencakup *template* desain berbasis HTML dan CSS untuk *tipografi*, bentuk, tombol, tabel, *navigasi, modals*, korsel gambar dan banyak lainnya, serta *plugin JavaScript* opsional. *Bootstrap* juga memberikan sobat kemampuan untuk membuat desain *responsif* dengan mudah.

|  |
| --- |
|  |
| Gambar 2. 6 *Bootstrap* |

*Responsive* *Layout,* maka hal tersebut pada aplikasi *web* yang didesain dengan menggunakan *Bootstrap* akan langsung menyesuaikan dengan lebar dari media perambahannya. Sehingga *Framework* ini support untuk semua jenis *device* baik dari *smartphone, tablet*, laptop ataupun *PC* *Desktop*. Di samping itu, *Boostrap* juga sudah support untuk HTML 5 dan C.

## *Balsamiq Mockup*

*Balsamiq Mockup* merupakan suatu program aplikasi yang digunakan, dalam pembuatan tampilan *user interface* sebuah aplikasi. *Software* ini sudah menyediakan *tools* yang dapat memudahkan dalam membuat desain *prototyping* aplikasi yang akan dibuat. Berfokus pada konten yang ingin digambar dan fungsionalitas yang dibutuhkan oleh pengguna.

|  |
| --- |
|  |
| Gambar 2. 7 *Balsamiq Mockup* |

Menggambar sketsa (*wireframe*) atau *prototype* rancangan desain *website* di atas kertas balsamiq *mockups* membantu seorang *web designer* dapat membuat tampilan *web* dalam bentuk gambar di komputer. Tujuannya agar membuat tampilan (*design*) *website* menarik juga dapat menyesuaikan dengan kebutuhan *customer* (pelanggan). Dengan alat pembuat *mockup* maka seorang *web designer* dapat menganalisa tata letak, desain dan fungsi.

## *Laravel*

*Laravel* adalah salah satu *Framework PHP* yang paling populer dan paling banyak digunakan di seluruh dunia dalam membangun aplikasi *web* mulai dari proyek kecil hingga besar. *Framework* ini banyak digunakan oleh *Web Developer* karena kinerja, fitur, dan skalabilitas nya.

|  |
| --- |
|  |
| Gambar 2. 8 *Laravel* |

*Framework* ini mengikuti struktur MVC (*Model View Controller*), MVC adalah sebuah metode aplikasi dengan memisahkan data dari tampilan berdasarkan komponen- komponen aplikasi, seperti : manipulasi data, *controller*, dan *user interface*.

## *Google Chrome*

*Google Chrome* merupakan suatu aplikasi peramban yang digunakan untuk menjelajah dunia maya seperti halnya *Firefox*, Opera ataupun *Microsoft Edge*. Jika *Firefox* dikembangkan oleh *Mozilla, Google Chrome* dibuat dan dirancang oleh Google, toko internet terbesar di dunia yang juga mempunyai *Android*.

|  |
| --- |
|  |
| Gambar 2. 9 *Google Chrome* |

Proyek yang dimiliki yakni *open source* digunakan oleh Google disebut Chromium, menggunakan mesin rendering *Webkit* sampai dengan versi 27 dan dirancang untuk bekerja dengan kecepatan di atas rata-rata namun tetap ringan dijalankan di perangkat desktop dan *mobile*.

## *Visual Studio Code*

*Visual Studio* *Code* (*VS Code*) merupakan suatu teks editor ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac,* dan *Windows*. Teks editor ini secara langsung mendukung bahasa pemrograman *JavaScript, Typescript,* dan *Node.js,* serta bahasa pemrograman lainnya dengan bantuan *plugin* yang dapat dipasang *via marketplace* *Visual Studio Code* (seperti *C++, C#, Python, Go, Java,* dst).

|  |
| --- |
|  |
| Gambar 2. 10 *Visual Studio Code* |

Banyaknya fitur yang disediakan oleh *Visual Studio Code*, diantaranya *Intellisense*, *Git Integration, Debugging*, dan fitur ekstensi yang menambah kemampuan teks editor. Fitur tersebut akan terus bertambah seiring dengan bertambahnya versi *Visual Studio Code*. Pembaruan versi *Visual Studio Code* ini juga dilakukan berkala setiap bulan, dan inilah yang membedakan *VS Code* dengan teks editor-teks editor yang lain.

## *Laragon*

*Laragon* adalah perangkat lunak bebas yang mendukung banyak sistem operasi, berfungsi sebagai server diri sendiri / *localhost*. *Laragon* menyediakan banyak *services, tools*, dan fitur mulai dari *Apache, MySQL, PHP Server, Memchaced, Redis, Composer, Xdebug, PhpMyAdmin, Cmder* dan *Laravel*.

|  |
| --- |
|  |
| Gambar 2. 11 *Laragon* |

## *Visual Paradigm*

*Visual Paradigm* suatu *software* model sistem visualisasi yang memungkinkan model yang telah dibuat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur lengkap. *Visual Paradigm* sebuah *software model* dengan sistem *visualisasi.*

|  |
| --- |
|  |
| Gambar 2. 12 *Visual Paradigm* |

*Visual Paradigm* memungkinkan model yang telah anda buat dapat digunakan sebagai representasi proyek-proyek lain dilengkapi dengan beberapa fitur yang ada didalamnya sampai pada menganalisa sebuah proyek yang akan dikerjakan. Diagram dapat disusun sedemikian rupa sehingga dapat dipustakakan menjadi proyek per proyek yang saling berkaitan.

## Raja Ongkir

RajaOngkir merupakan sebuah situs dan *web service* (API) yang menyediakan informasi ongkos kirim dari berbagai kurir di Indonesia seperti POS Indonesia, JNE, TIKI, PCP, ESL, dan RPX. Secara umum, Raja Ongkir ditujukan kepada pengguna yang ingin mengetahui dan membandingkan ongkos kirim dari berbagai kurir dan secara khusus bagi pemilik *toko online*, maupun bagi orang yang sering berbelanja *online*.

|  |
| --- |
| Tutorial Program Cek Ongkos Kirim Dengan Menggunakan API RajaOngkir (Akun  Pro) - Catatan Ozkadon |
| Gambar 2. 13 Raja Ongkir |

## *Payment Gateway*

Payment gateway adalah sebuah media transaksi yang disediakan oleh layanan aplikasi *e-commerce*. Bisa memberi otorisasi pemrosesan kartu kredit, ataupun pembayaran langsung bagi pelanggan dalam aktivitas bisnis *online*. Indonesia *payment gateway* ini bisa mempermudah para pebisnis sekaligus pelanggan dalam bertransaksi,.

|  |
| --- |
|  |
| Gambar 2. 14 *Payment Gateway* |

Ada banyak manfaat yang bisa didapat dengan menggunakan *payment gateway* yaitu pelaku bisnis tidak perlu membuat banyak rekening, mudahnya rekap pembayaran dan pembelian, keamanan transaksi sangat terjamin, transaksi tidak terbatas.

## *Corel Draw X8*

*Corel Draw* adalah *software* atau aplikasi berbasis *dekstop* komputer yang digunakan untuk membuat atau melakukan *editor grafik vektor*. Aplikasi yang dibuat oleh *Corel* (sebuah perusahaan software yang berbasis di Ottawa, Kanada) dirilis pertama kali dengan versi 1 pada Januari 1989. Dalam perkembangannya, aplikasi ini sudah muncul versi terbarunya yakni *Corel Draw 8* atau *Corel Draw X8* yang dirilis pada tanggal 15 Maret 2016.

|  |
| --- |
| Mengatasi Corel Draw X8 Yang Tidak Bisa di Install - Mr Leong |
| Gambar 2. 15 Corel Draw X8 |

Aplikasi Corel Draw ini fokus pada editor gambar, sehingga banyak dipakai oleh pengguna dalam bidang *advertising, desain visual*, serta percetakan dan bidang lain yang memerlukan format *visualisasi*.

# BAB III

# ANALISIS DAN PERANCANGAN SISTEM

# Analisis Sistem

Aplikasi *e-commerce* yang akan dibangun merupakan suatu sistem yang dapat memberikan informasi mengenai suatu produk atau barang yang tersedia, proses pemesanan barang yang dapat dilakukan secara *realtime* terintegrasi dengan raja ongkir, serta proses transaksi terintegrasi dengan *payment gateway* atau *virtual account*.

Aplikasi *e-commerce* berkembang pesat setiap tahunnya, sesuai dengan perkembangan ilmu pengetahuan dan teknologi, dan banyak digunakan di berbagai perusahaan atau toko yang melakukan penjualan, terutama toko yang bergerak di bidang penjualan untuk memperoleh kemudahan *customer* dalam mendapatkan informasi produk yang tersedia, pemesanan barang yang dapat dilakukan dimana pun dan kapanpun selama terhubung dengan akses internet, proses transaksi dilakukan tanpa harus memiliki rekening *bank*, transaksi dapat dilakukan melalui *alfamart, indomart,* *virtual account* seperti *gopay, akulaku, bri, mandiri, bca* dan lain sebagainnya. Adapun tahapan proses sistem yang akan dibangun dapat dikategorikan sebagai berikut

1. Proses promosi produk

Promosi produk didapatkan melaui aplikasi yang akan dibangun, adapun informasi mengenai ketersediaan produk atau *quantity* produk, berbagai *discount* produk yang ditawarkan, pada produk di Gemilang Jaya *Computer*.

1. Proses pemesanan barang

Proses pemesanan barang dapat dilakukan dimanapun dan kapanpun selama *customer* memiliki *device* dan terhubung dengan akses internet, pada tahap ini *customer* memilih barang yang akan dibeli sesuai dengan *quantity* barang yang dipesan di Gemilang Jaya *Computer*, pada saat melakukan pemesanan barang, *customer* dapat mengetahui total pembelian yang harus di bayarkan, serta dapat memilih jasa kirim karena terintegrasi dengan raja ongkir.

1. Proses transaksi barang

Proses transaksi barangdengan sistem yang akan dibangun terintegrasi dengan *payment gateway,* untuk proses transaksi dilakukan tanpa harus memiliki rekening *bank*, transaksi dapat dilakukan melalui *alfamart, indomart,* *virtual account* seperti *gopay, akulaku, bri, mandiri, bca* dan lain sebagainnya tanpa konfirmasi pembayaran, konfirmasi pembayaran *otomatis* ketika *customer* menyelesaikan pembayaran dengan *virtual account*.

1. Proses pengiriman barang

Proses pengiriman barang terintegrasi dengan raja ongkir, menyediakan ongkos kirim terpadu dari setiap jenis perusahaan kurir, *customer* tidak harus menghitung biaya ongkos kirim, sistem akan otomatis akan menangani masalah berikut.

# Analisis Masalah

Tujuan dari analis masalah untuk memperoleh informasi atau data mengenai masalah yang ada di perusahaan atau tempat penelitian, masalah yang timbul terjadi pada operasional toko, umumnya setiap masalah yang timbul bermacam - macam tergantung kondisi dalam toko, dengan harapan setiap masalah yang terjadi dapat ditemukannya solusi atau pemecahan masalah.

Masalah yang timbul di Gemilang Jaya *Computer* yakni transaksi penjualan pada Gemilang Jaya *Computer* dilakukan datang ke toko, transaksi tersebut tidak dapat di jangkau secara luas. ketidakmudahan *customer* dalam mendapatkan informasi secara cepat mengenai *quantity* produk yang tersedia, *discount* produk, karena promosi produk menggunakan brosur atau media cetak..

Proses pemesanan membutuhkan waktu lama, *customer* harus mengantri untuk di layani jika toko dalam keadaanramai dipadati *customer* atau pengunjung, proses pembayaran dimuka atau di toko tidak dapat dilakukan secara *fleksibel* atau pembayaran tidak dapat dilakukan dimanapun dan kapanpun.

# Analisis sitem yang sedang berjalan

Analisis sistem yang sedang berjalan menjelaskan bagaimana proses bisnis kegiatan informasi, pemesanan dan transaksi penjualan di Gemilang Jaya *Computer*.

Berikut merupakan *Swimlane* Diagram dari sistem yang sedang berjalan di Gemilang Jaya *Computer*:

|  |
| --- |
|  |
| Gambar 3. 1  *Swimlane* Diagram proses informasi produk |

1. Proses informasi produk

Keterangan: Berdasarkan Gambar 3.1 *Swimlane* Diagram proses informasi produk yang sedang berjalan menunjukan langkah-langkah aktivitas Gemilang Jaya *Computer* melakukan promosi serta penyebaran informasi kepada *customer* untuk meningkatkan penjualan.

1. Proses pemesanan produk

|  |
| --- |
|  |
| Gambar 3. 2  *Swimlane* Diagram proses pemesanan produk |

Keterangan: Berdasarkan Gambar 3.2 *Swimlane* Diagram proses pemesanan produk yang sedang berjalan menunjukan langkah-langkah Gemilang Jaya *Computer* terhadap *customer,* pemesanan hingga konfirmasi pesanan guna berfungsi untuk memberikan pelayanan terbaik.

1. Proses transaksi produk

|  |
| --- |
|  |
| Gambar 3. 3  *Swimlane* Diagram proses transaksi produk |

Keterangan: Berdasarkan Gambar 3.3 *Swimlane* Diagram proses transaksi produk yang sedang berjalan menunjukan langkah-langkah Gemilang Jaya *Computer* terhadap *customer,* pengecekan data hingga konfirmasi data pembelian guna untuk memastikan data yang diterima *valid*.

1. Proses pengiriman produk

|  |
| --- |
|  |
| Gambar 3. 4  *Swimlane* Diagram proses pengiriman produk |

Keterangan: Berdasarkan Gambar 3.4 *Swimlane* Diagram proses pengiriman produk yang sedang berjalan menunjukan langkah-langkah Gemilang Jaya *Computer* terhadap *customer*, pengecekan produk, pengemasan, pemilihan jasa kurir hingga pengiriman produk, serta produk diterima *customer*.

# Analisis sitem yang akan dibangun

Analisis sistem yang akan dibangun menjelaskan bagaimana proses bisnis yang akan diperbaharui, tentang bagaimana proses pemesanan hingga transaksi penjualan oleh suatu sistem. Berikut merupakan *Swimlane* Diagram dari sistem yang akan dibangun sebagai berikut

|  |
| --- |
|  |
| Gambar 3. 5 *Swimeline Diagram* proses bisnis yang akan dibangun |

Keterangan: Berdasarkan Gambar 3.5 *Swimlane* Diagram proses bisnis yang akan dibangun menunjukan langkah-langkah aktivitas tiap *user* menggunakan sistem atau aplikasi *e-commerce* yang akan dibangun .

# Analisis kebutuhan non fungsional

Kebutuhan *non* Fungsional adalah kebutuhan yang menitikberatkan pada *property* perilaku yang dimiliki oleh sistem, atau suatu fungsi yang menggambarkan kebutuhan yang diperlukan untuk menjalankan aplikasi atau sistem yang akan dibangun.

Untuk menjalankan sistem atau aplikasi *e-commerce* ada beberapa kebutuhan yang harus dipersiapkan diantaranya yaitu kebutuhan perangkat keras, kebutuhan perangkat lunak, dan pengguna yang menggunakan aplikasi. Analisa kebutuhan *non* fungsional dibangun dengan tujuan aplikasi dapat di digunakan oleh pemilik atau karyawan Gemilang Jaya *Computer*.

## Analisa Pengguna

Analisa pengguna merupakan suatu aktivitas yang menunjukan siapa saja yang menggunakan sistem dan apa saja yang bisa dilakukan oleh pengguna di dalam sistem. Berikut ini merupakan pengguna dari aplikasi *e-commerce* yang akan dibangun di Gemilang Jaya *Computer*:

1. Admin

Admin memiliki tugas untuk mengelola dan menggunakan sistem secara keseluruhan, adapun syarat yang harus dimiliki yaitu sebagai berikut:

1. Paham dan dapat mengoperasikan komputer dengan baik dan benar
2. Teliti dan jujur
3. Memahami alur secara keseluruhan dari aplikasi *e-commerce*
4. *Customer*

*Customer* sebagai pengunjung Toko Gemilang Jaya *Computer* serta bersifat umum atau *public* dalam menggunakan sistem, adapun syarat yang harus dimiliki yaitu sebagai berikut:

1. Paham dan dapat mengoperasikan komputer dengan baik dan benar
2. Teliti dan jujur
3. Bisa menggunakan aplikasi penjualan *e-commerce* dengan baik

## Analisa kebutuhan perangkat keras

Analisa kebutuhan perangkat keras atau *hardware* bertujuan agar aplikasi *e-commerce* dapat berjalan dengan baik, spesifikasi yang dibutuhkan untuk menjalankan aplikasi *e-commerce* yaitu sebagai berikut:

Tabel 3. 1 Tabel Kebutuhan Perangkat Keras

|  |  |  |
| --- | --- | --- |
| No | Perangkat | Spesifikasi |
| 1 | *Processor* | *Dual core i3 up to* |
| 2 | *Memory Ram* | *3 GB up to* |
| 3 | *Hardisk* | 500 GB *up to* |
| 4 | *Monitor* | *Standard* |
| 5 | *Keyboard* | *Standard* |
| 6 | *Mouse* | *Standard* |

## Analisa kebutuhan perangkat lunak

Analisa perangkat lunak merupakan suatu perangkat lunak yang berkaitan dengan operasional kerja komputer, dan dapat digunakan oleh komputer untuk menjadi pendukung agar sistem dapat berjalan dengan baik,diantaranya dalah sebagai berikut

1. Sistem Operasi

Sistem operasi digunakan sebagai perantara antara perangkat lunak dan perangkat keras komputer terdapat berbagai macam sistem operasi baik itu *Windows* 7/8/10 atau *linux* dalam aplikasi ini sistem operasi yang di rekomendasikan yaitu *windows* 10 karena mudah dalam operasional maupun sistem yang *up date*. Sistem operasi menjadi pengatur atau bertanggung jawab dalam penggunaan perangkat keras dan bertugas untuk menjalankan perangkat lunak atau aplikasi yang digunakan

1. *Database*

Dengan menggunakan *database*, maka pengguna diberikan fasilitas untuk melakukan penyimpanan, pengolahan, dan transformasi data dalam basis data, dalam aplikasi ini *database* yang di rekomendasikan yakni *MYSQL* karena mudah dalam penggunaan dan lebih familiar.

1. Server lokal

Dengan menggunakan server lokal maka pengguna diberikan fasilitas untuk menggunakan *database mysql*, untuk server lokal yang direkomendasikan yakni *laragon*, karena lebih *fleksibel*, familiar dan mudah untuk digunakan.

# Analisis pemodelan sistem

Analisis pemodelan sistem yanga akan dibangun, konsep perangkat lunak digunakan untuk menetukan tujuan dari aplikasi yang akan dibuat serta capaian dari produk yang akan dihasilkan. Tujuan-tujuan ini harus secara jelas dijabarkan agar bisa terpenuhi dalam menentukan konsep perangkat lunak yang akan dibuat. Dalam tahap ini, dilakukan analisa kebutuhan untuk mengumpulkan dan merincikan persyaratan dari sebuah sistem dan mengidentifikasi fungsi-fungsi dasar yang akan dijalankan software, oleh karena itu analisis kebutuhan ini digambarkan dalam bentuk *use case* atau UML (*Unified Modeling Leanguange)*.

# Analisis kebutuhan fungsional

Aplikasi yang dibangun harus mempunyai fungsi utama yang mendukung pemecahan masalah yang ada pada sebuah instansi. Fungsi yang dimaksud adalah apa saja yang dapat dilakukan dengan menggunakan aplikasi tersebut. Beberapa fungsi utama dalam pembanguanan *e-commerce* penjualan perangkat komputer di Gemilang jaya *Computer* adalah sebagai berikut:

* 1. Kelola akun
  2. Lupa sandi
  3. Pendaftaran
  4. *Login*
  5. Halaman Utama
  6. Beranda *customer*
  7. Kelola akun *customer*
  8. Pemesanan barang
  9. *Logout*
  10. Beranda admin
  11. Kelola kategori
  12. Kelola produk
  13. *Invoice*
  14. Kelola *customer*
  15. Kelola slider
  16. Kelola admin

Analisis aplikasi *e-commerce* yang akan di bangun pada Toko Gemilang Jaya *Computer* dirancang menggunakan UML (*Unifed Modeling Leanguage*), diantaranya *Secenario-Based Modeling* yang meliputi *Use Case Secenario* (Naratif). *Actor List, Use Case List, Use Case Diagram*, dan *Swimlane Diagram*. Data modeling yang meliputi *Class Diagram*, dan *Behavioral Modeling* yaitu *Sequence Diagram*.

## *Actor List*

Aktor merupakan orang yang menggunakan sistem atau perangkat baik *software/hardware*, serta memiliki sifat dan fungsi yang dijelaskan dalam sebuah konteks yang memiliki peran terhadap sistem, berikut adalah pendefinisian aktor pada aplikasi *e-commerce* yang akan dibangun di Toko Gemilang Jaya *Computer*

Tabel 3. 2 Actor List

|  |  |  |
| --- | --- | --- |
| No | Aktor | Deskripsi |
| 1 | Admin | Orang yang mengelola sistem dan akun untuk pengguna atau *user*, serta melaksanakan transaksi pada sistem yang berjalan. |
| 2 | *Customer* | Orang yang mempunyai hak akses untuk masuk ke sistem dan melakukan proses transaksi. |

## *Usecase Scenario (Naratif)*

1. *Usecase Scenario* Kelola Akun

Tabel 3. 3 *Usecase Scenario* Kelola *Login*

|  |  |  |  |
| --- | --- | --- | --- |
| No | *Use Case* | Aktor | *Activity* |
| 1 | U1. Kelola Akun | *Customer* | * 1. Aktor masuk ke sistem.   2. Sistem akan menampilkan   halaman utama   * 1. Aktor memilih produk   2. Aktor mengisi data pembelian   3. Aktor memilih jasa kurir untuk   produk yang dipilih   * 1. Aktor melakukan *checkout*   pesanan   * 1. Sistem akan menampilkan   halaman kelola akun   * 1. Sistem akan menampilkan   halaman akun yang terdiri dari *login* lupa *password,* dan pendaftaran |
| 2 | U2. Lupa  Sandi | *Customer* | 1. Aktor menekan tombol lupa sandi 2. Sistem akan menampilkan form   *email*   1. Aktor mengisi *form* *email* 2. Aktor menekan tombol kirim email 3. Jika *email* tidak terdaftaran maka   email tidak akan muncul, jika *email* terdaftar sistem akan menampilkan pesan *email* terkirim |
| No | *Use Case* | Aktor | *Activity* |
| 3 | U3. Pendaftaran | *Customer* | 1. Aktor menekan tombol pendaftaran maka akan menampilkan halaman   pendaftaran   1. Aktor mengisi data pada *form*   pendaftaran   1. Aktor menekan tombol daftar, jika tidak ada data yang dimasukan maka menampilkan pesan   kesalahan “*form* di wajib di isi", jika data yang di isi benar maka menampilkan pesan “pendaftaran  anda berhasil” dan diarahkan ke  halaman utama |
| 4 | U4. *Login* | Admin, *Customer* | 1. Aktor mengisi alamat *email* dan *password* 2. Aktor menekan tombol *login* 3. Sistem akan mengecek verifikasi   apakah benar atau tidak.   1. Jika benar maka sistem akan   menampilkan halaman utamanya   1. Jika salah maka kana menampilkan pesan kesalahan “*email* dan   *password* salah” |
| 5 | U5.Halaman  Utama | *Customer* | 1. *Customer* masuk ke dalam sistem 2. Memilih produk 3. Menekan tombol tambahkan ke   keranjang |
| No | *Use Case* | Aktor | *Activity* |
| 5 | U5.Halaman  Utama | *Customer* | 1. Jika *customer* tidak terdaftar dalam sistem dan tidak melakukan *login*   sebelumnya maka *customer*  diarahkan ke halaman *login*   1. Jika *customer* telah terdaftar dalam sistem dan telah melakukan *login*   maka *customer* diarahkan ke  halaman pesanan   1. Mengisi *form* pesanan, menekan   tombol *checkout*   1. Menampilkan halaman detail   pesanan |
| 6 | U6. Beranda  *Customer* | *Customer* | 1. Aktor masuk ke halaman *customer* 2. Aktor menekan tombol beranda   *customer*   1. Sistem menampilkan nama   *customer,* serta status pemesanan |
| 7 | U7. Kelola  *Customer* | *Customer* | 1. Aktor masuk ke halaman *customer* 2. Aktor menekan tombol kelola   *customer*   1. Sistem menampilkan kelola   *customer*   1. Aktor memiliki fungsi *update* data 2. Jika aktor ingin mengubah data maka: 3. Aktor menekan tombol ubah 4. Sistem menampilkan   data yang akan diubah. |
| No | *Use Case* | Aktor | *Activity* |
| 7 | U7. Kelola  *Customer* | *Customer* | 1. Aktor mengubah data 2. Aktor menekan tombol ubah   Sistem akan menampilkan data yang telah diubah |
| 8 | U8. Pesanan  Barang | *Customer* | 1. Aktor masuk ke halaman *customer* 2. Aktor memiliki fungsi *search* dan detail pesanan 3. Jika aktor ingin mencari data   maka:   1. Aktor mengisi *form* cari 2. Aktor menekan tombol cari 3. Jika data yang dicari tidak ada maka menampilkan pesan “data tidak ditemukan”, jika data yang dicari ada maka menampilkan data yang dicari dalam *list.* 4. Jika aktor menekan tombol detail maka: 5. Sistem menampilkan detail   pesanan produk   1. *Customer* menekan tombol bayar 2. Sistem menampilkan halaman   pembayaran dari *payment*  *gateway*   1. *Customer* menekan tombol   *continue*   1. Sistem menampilkan   metode pembayaran |
| No | *Use Case* | Aktor | *Activity* |
| 8 | U8. Pesanan  Barang | *Customer* | 1. *Customer* memilih metode   pembayaran   1. *Customer* menekan tombol *see*   *account number*   1. Sistem menampilkan kode *virtual account* 2. Jika *customer* menyelesaikan transaksi maka status transaksi berhasil, jika *customer* tidak menyelesaikan transaksi maka transaksi gagal. 3. Menekan tombol kembali 4. Sistem menampilkan halaman   utama |
| 9 | U9. *Logout* | Admin, *Cusomer* | 1. Aktor masuk kedalam sistem 2. Aktor menekan tombol *logout* 3. Sistem mengecek verifikasi apakah benar atau tidak? 4. Jika tidak maka tidak terjadi   perubahan halaman pada sistem.   1. Jika ya maka sistem akan   menampilkan halaman *login.* |
| 10 | U10. Beranda  Admin | Admin | 1. Aktor masuk ke halaman admin 2. Aktor menekan tombol beranda   admin   1. Sistem menampilkan status   pesanan serta statistik pendapatan |
| No | *Use Case* | Aktor | *Activity* |
| 11 | U11. Kelola  Kategori | Admin | 1. Aktor memilih menu kelola kategori di halaman admin 2. Sistem akan menampilkan halaman kelola kategori 3. Aktor memiliki fungsi tambah,   ubah, hapus, cari,.   1. Jika aktor menekan tombol tambah : 2. Sistem akan menampilkan form   tambah kategori   1. Aktor dapat mengisi *form* tambah kategori 2. Jika aktor menekan tombol   simpan   1. Sistem akan menampilkan pesan data berhasil ditambah 2. Jika aktor menekan tombol   ulangi Sistem aka me-*reset* data  yang telah di isi.   1. Jika aktor menekan tombol ubah : 2. Aktor memilih data kategori yang akan diubah 3. Aktor menekan tombol ubah 4. Sistem menampilkan data   kategori yang akan diubah   1. Aktor mengubah data kategori yang telah dipilih. 2. Jika aktor menekan tombol   simpan |
| No | *Use Case* | Aktor | *Activity* |
| 11 | U11. Kelola  Kategori | Admin | 1. Sistem akan menampilkan pesan data berhasil diubah 2. Jika aktor menekan tombol ulangi 3. Sistem akan me-*reset* data yang telah di isi. 4. Jika aktor menekan tombol hapus : 5. Sistem akan menampilkan pesan   konfirmasi ya atau tidak.   1. Jika tidak maka halaman sistem   tidak akan berubah   1. Jika ya maka sistem akan   menampilkan data berhasil  dihapus   1. Jika aktor akan mencari data : 2. Aktor mengisi *form* pencarian 3. Aktor menekan tombol cari 4. Jika data tidak ditemukan,   sistem akan menampilkan pesan data tidak ditemukan   1. Jika data ditemukan, sistem akan menampilkan data yang dicari. |
| 12 | U12. Kelola  Produk | Admin | 1. Aktor memilih menu kelola produk di halaman admin 2. Sistem akan menampilkan halaman kelola produk 3. Aktor memiliki fungsi tambah,   ubah, hapus, cari. |
| No | *Use Case* | Aktor | *Activity* |
| 12 | U12. Kelola  Produk | Admin | 1. Jika aktor menekan tombol tambah : 2. Sistem akan menampilkan *form*   tambah produk   1. Aktor dapat mengisi *form* tambah produk 2. Jika aktor menekan tombol   simpan   1. Sistem akan menampilkan pesan data berhasil ditambah 2. Jika aktor menekan tombol   ulangi   1. Sistem aka me-*reset* data yang telah di isi. 2. Jika aktor menekan tombol ubah : 3. Aktor memilih data produk yang akan diubah 4. Aktor menekan tombol ubah 5. sistem menampilkan data   produk yang akan diubah   1. Aktor mengubah data produk   yang telah dipilih.   1. Jika aktor menekan tombol   simpan   1. Sistem akan menampilkan pesan data berhasil diubah 2. Jika aktor menekan tombol   ulangi |
| No | *Use Case* | Aktor | *Activity* |
| 12 | U12. Kelola  Produk | Admin | 1. Sistem akan me-*reset* data yang telah di isi. 2. Jika aktor menekan tombol hapus : 3. Sistem akan menampilkan pesan   konfirmasi ya atau tidak.   1. Jika tidak maka halaman di   sistem tidak akan berubah   1. Jika ya maka sistem akan   menampilkan data berhasil  dihapus   1. Jika aktor akan mencari data : 2. Aktor mengisi *form* pencarian 3. Aktor menekan tombol cari 4. Jika data tidak ditemukan,   sistem akan menampilkan pesan data tidak ditemukan   1. Jika data ditemukan, sistem akan   menampilkan data yang dicari. |
| 13 | U13. Kelola  *Invoice* | Admin | 1. Aktor memilih menu kelola *invoice* di halaman admin 2. Sistem akan menampilkan halaman kelola *invoice* 3. Aktor memiliki fungsi cari,   dan detail *invoice*   1. Jika aktor akan mencari data : 2. Aktor mengisi *form* pencarian 3. Aktor menekan tombol cari |
| No | *Use Case* | Aktor | *Activity* |
|  |  |  | 1. Jika data tidak ditemukan,   sistem akan menampilkan pesan data tidak ditemukan   1. Jika data ditemukan, sistem akan   menampilkan data yang dicari.   1. Jika aktor memilih detail *invoice*, sistem akan menampilkan detail   *invoice* serta status pesanan produk |
| 14 | U14. Kelola  *Customer* | Admin | 1. Aktor memilih menu kelola kelola *customer* di halaman admin 2. Sistem akan menampilkan halaman kelola *customer* 3. Aktor memiliki fungsi cari 4. Jika aktor akan mencari data : 5. Aktor mengisi *form* pencarian 6. Aktor menekan tombol cari 7. Jika data tidak ditemukan,   sistem akan menampilkan pesan data tidak ditemukan   1. Jika data ditemukan, sistem akan   menampilkan data yang dicari. |
| 15 | U15. Kelola  *Slider* | Admin | 1. Aktor memilih menu kelola slider di halaman admin 2. Sistem akan menampilkan halaman kelola *slider* 3. Aktor memiliki fungsi tambah dan hapus |
| No | *Use Case* | Aktor | *Activity* |
| 15 | U15. Kelola  *Slider* | Admin | 1. Jika aktor menekan tombol tambah : 2. Sistem akan menampilkan *form*   tambah *slider*   1. Aktor dapat mengisi *form* tambah produk 2. Jika aktor menekan tombol   simpan   1. Sistem akan menampilkan pesan data berhasil ditambah 2. Jika aktor menekan tombol   ulangi   1. Sistem aka me-*reset* data yang telah di isi. 2. Jika aktor menekan tombol hapus : 3. Sistem akan menampilkan pesan   *verifikasi* ya atau tidak.   1. Jika tidak maka halaman di   sistem tidak akan berubah   1. Jika ya maka sistem akan   menampilkan data berhasil  dihapus |
| 16 | U16. Kelola  Admin | Admin | 1. Aktor memilih menu kelola produk di halaman admin 2. Sistem akan menampilkan halaman kelola produk 3. Aktor memiliki fungsi tambah,   ubah, hapus, cari. |
| No | *Use Case* | Aktor | *Activity* |
| 16 | U16. Kelola  Admin | Admin | 1. Jika aktor menekan tombol tambah : 2. Sistem akan menampilkan *form*   tambah produk   1. Aktor dapat mengisi *form* tambah produk 2. Jika aktor menekan tombol   Simpan   1. Sistem akan menampilkan pesan data berhasil ditambah 2. Jika aktor menekan tombol   ulangi   1. Sistem aka me-*reset* data yang telah di isi. |

## Daftar *usecase*

Dafar *use case* menjelaskan tentang *use case* yang ada dalam sistem, daftar *usecase* berupa kode *usecase*, nama *usecase* dan keterangan dari tiap *usecase*. Adapun *usecase list* yang terdapat pada *e-commerce* yang akan dibangun diToko Gemilang Jaya *Computer*, sebagai berikut:

|  |  |  |
| --- | --- | --- |
| Kode | Sistem *Usecase* | Keterangan |
| U1 | Kelola Akun | Digunakan untuk memastikan tiap *user* untuk memiliki hak akses masuk kedalam sistem, kelola akun terdiri dari *login,* lupa sandi, dan pendaftarandimana persyaratan tersebut  harus terpenuhi untuk masuk kedalam sistem dan mengakses halaman yang dizinkan |
| U2 | Lupa Sandi | Digunakan oleh aktor sebagai alternatif jika aktor tidak mengingat *username* atau  *password* yang terdaftar |
| U3 | Pendaftaran | Digunakan oleh aktor untuk membuat akun atau *member,* agar aktor dapat mengakses  dan terdaftar dalam sistem |
| U4 | *Login* | Digunakan untuk memastikan setiap *user* masuk kedalam sistem, sesuai hak aksesnya  berdasarkan *username* dan *password* yang disertakan. |
| U5 | Halaman Utama | Digunakan untuk memastikan setiap *user* masuk kedalam sistem, mengakses halaman utama dan melakukan pemilihan produk serta melalukan transksi ketika *user* telah *login* |
| U6 | Beranda *Customer* | Digunakan oleh *customer* untuk mengakses beranda atau halaman *customer* sesuai hak aksesnya |
| Kode | Sistem *Usecase* | Keterangan |
| U7 | Kelola Akun *Customer* | Digunakan oleh *customer* untuk mengelola akun atau pengguna yang terdaftar dalam sistem |
| U8 | Pesanan Barang | Digunakan oleh *customer* untuk melihat detail pesanan dan mencari *invoice* pesanan barang yang dipesan. |
| U9 | *Logout* | Digunakan untuk memastikan setiap *user* baik admin ataupun *customer* keluar dari sistem, sesuai hak aksesnya |
| U10 | Beranda Admin | Digunakan oleh adminuntuk mengakses beranda atau halaman admin serta melihat status pemesanan dan *statistic* penjualan |
| U11 | Kelola Kategori | Digunakan oleh admin untuk mengelola kategori produk yang akan di *publish* pada halaman utama sistem yang akan dibangun. |
| U12 | Kelola Produk | Digunakan oleh admin untuk mengelola produk yang akan di *publish* pada halaman utama sistem yang akan dibangun. |
| U13 | *Invoices* | Digunakan oleh admin untuk melihat detail *invoice* pesanan dari *customer* serta dapat melihat dan mencari *invoice.* |
| U14 | Kelola *Customer* | Digunakan oleh admin untuk melihat *customer* yang telah *join* atau terdaftardalam sistem |
| U15 | Kelola *Slidder* | Digunakan oleh admin untuk menambahkan atau menghapus *slider* atau *design* yang akan tampil pada sistem. |
| Kode | Sistem *Usecase* | Keterangan |
| U16 | Kelola Admin | Digunakan oleh admin untuk mengelola akun atau pengguna yang terdaftar dalam sistem |

## *Usecase Diagram*

|  |
| --- |
|  |
| Gambar 3. 6 *Usecase Diagram* |

Berdasarkan Gambar 3.6 *Use Case Diagram* pembangunan *e-commerce* penjualan perangkat komputer di Gemilang Jaya *Computer* terdapat 16 *use case*. Jika *customer* ingin memasuki halaman utama *customer,* dan semua halaman yang terdapat pada halaman *customer,* serta melakukan proses transaksi dengan Gemilang Jaya *Computer* maka harus memiliki akun yang terdaftar dalam sistem, sedangkan halaman utama admin yakni Gemilang Jaya *Computer* dan semua halaman yang terdapat pada halaman admin memiliki hak akses penuh dalam penjualan produk.

# Perancangan Sistem

Pemodelan perancangan bertujuan untuk merincikan arsitektur perangkat lunak, struktur data, antar muka dan komponen lain yang bertujuan untuk mengimplementasikan sistem. Produk yang dihasilkan terdiri dari arsitektur, antarmuka, level komponen, dan deployment (Pressman, 2012).

## *Swimelane Diagram*

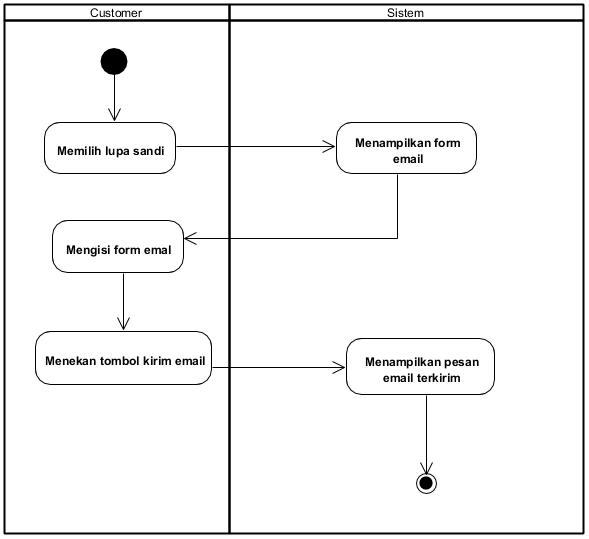
Berikut ini adalah *Swimlane Diagram* dari pembangunan *e-commerce* penjualan perangkat komputer di Gemilang Jaya *Computer*

1. *Swimlane Diagram* Kelola Akun

|  |
| --- |
|  |
| Gambar 3. 7 *Swimlane Diagram* Kelola Akun |

Berdasarkan Gambar 3.7 *Swimlane* *Diagram* Kelola Akun menunjukan langkah-langkah atau aktivitas *user* dalam melakukan pembuatan akun, agar *user*

memiliki akses yang terdaftar dalam sistem.

1. *Swimlane* Diagram Lupa Sandi

|  |
| --- |
|  |
| Gambar 3. 8 *Swimlane Diagram Logout* |

Berdasarkan Gambar 3.8 *Swimlane Diagram* Lupa Sandi menunjukan langkah atau aktivitas *customer* ketika lupa *password,* hal tersebut sebagai alternative ataupun solusi dari permasalahan tersebut.

1. *Swimlane* Diagram Pendaftaran

|  |
| --- |
|  |
| Gambar 3. 9 *Swimlane Diagram* Pendaftaran |

Berdasarkan Gambar 3.9 *Swimlane Diagram* Pendaftaran menunjukan langkah langkah atau aktivitas *customer* dalam melakukan pendaftaran, agar *user* memiliki akun yang terdaftar dalam sistem.

1. *Swimlane Diagram* *Login*

|  |
| --- |
|  |
| Gambar 3. 10 *Swimlane Diagram* *Login* |

Berdasarkan Gambar 3.10 *Swimlane* *Diagram* *Login* menunjukan langkah langkah atau aktivitas *user* masuk kedalam sistem dengan memasukan *email* dan *password* yang telah terdaftar dalam sistem

1. *Swimlane Diagram* Halaman Utama

|  |
| --- |
|  |
| Gambar 3. 11 *Swimlane Diagram* Halaman Utama |

Berdasarkan Gambar 3.11 *Swimlane* *Diagram* Halaman Utama memperlihatkan aktivitas *customer* melakukan pemesanan produk dengan syarat *customer* bisa melakukan pemesanan jika *customer* telah *login* dalam sistem.

1. *Swimlane Diagram* Beranda *Customer*

|  |
| --- |
|  |
| Gambar 3. 12 *Swimlane Diagram* Beranda *Customer* |

Berdasarkan Gambar 3.12 *Swimlane* *Diagram* Beranda *Customer* memperlihatkan halaman *customer* seperti nama *customer* serta status pemesanan produk yang telah atau sedang dilakukan.

1. *Swimlane Diagram* Kelola *Customer*

|  |
| --- |
|  |
| Gambar 3. 12 *Swimlane Diagram* Kelola *Customer* |

Berdasarkan Gambar 3.12 *Swimlane Diagram* Kelola *Customer* menunjukan langkah-langkah atau aktivitas *customer* yang memiliki akses untuk mengubah *password* pada akun yang digunakan.

1. *Swimlane Diagram* Pesanan Barang

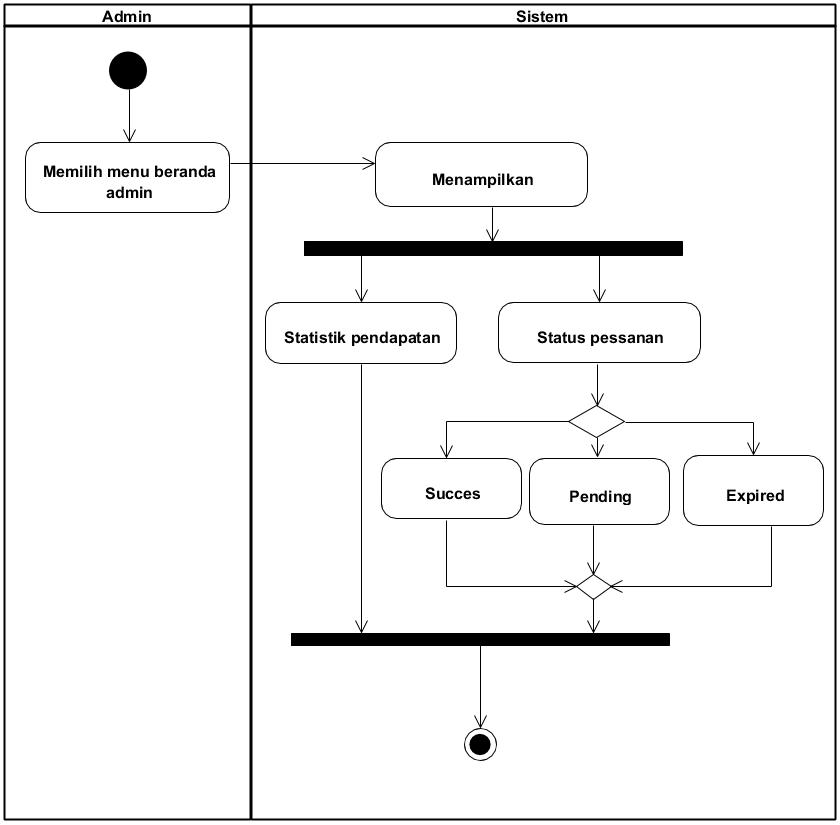
|  |
| --- |
|  |
| Gambar 3. 13 *Swimlane Diagram* Pesanan Barang |

Berdasarkan Gambar 3.13 *Swimlane* *Diagram* Pesanan Barangmenunjukan langkah-langkah atau aktivitas *customer* dalam mencari pesananan produk serta detail dari pembelian produk.

1. *Swimlane Diagram* *Logout*

|  |
| --- |
|  |
| Gambar 3. 14 *Swimlane Diagram* *Logout* |

Berdasarkan Gambar 3.14 *Swimlane Diagram* *Logout* menunjukan langkah atau aktivitas *user* keluar dari sistem atau aplikasi, aktivitas tersebut dilakukan ketika *user* sedang berada dalam sistem.

1. *Swimlane Diagram* Beranda Admin

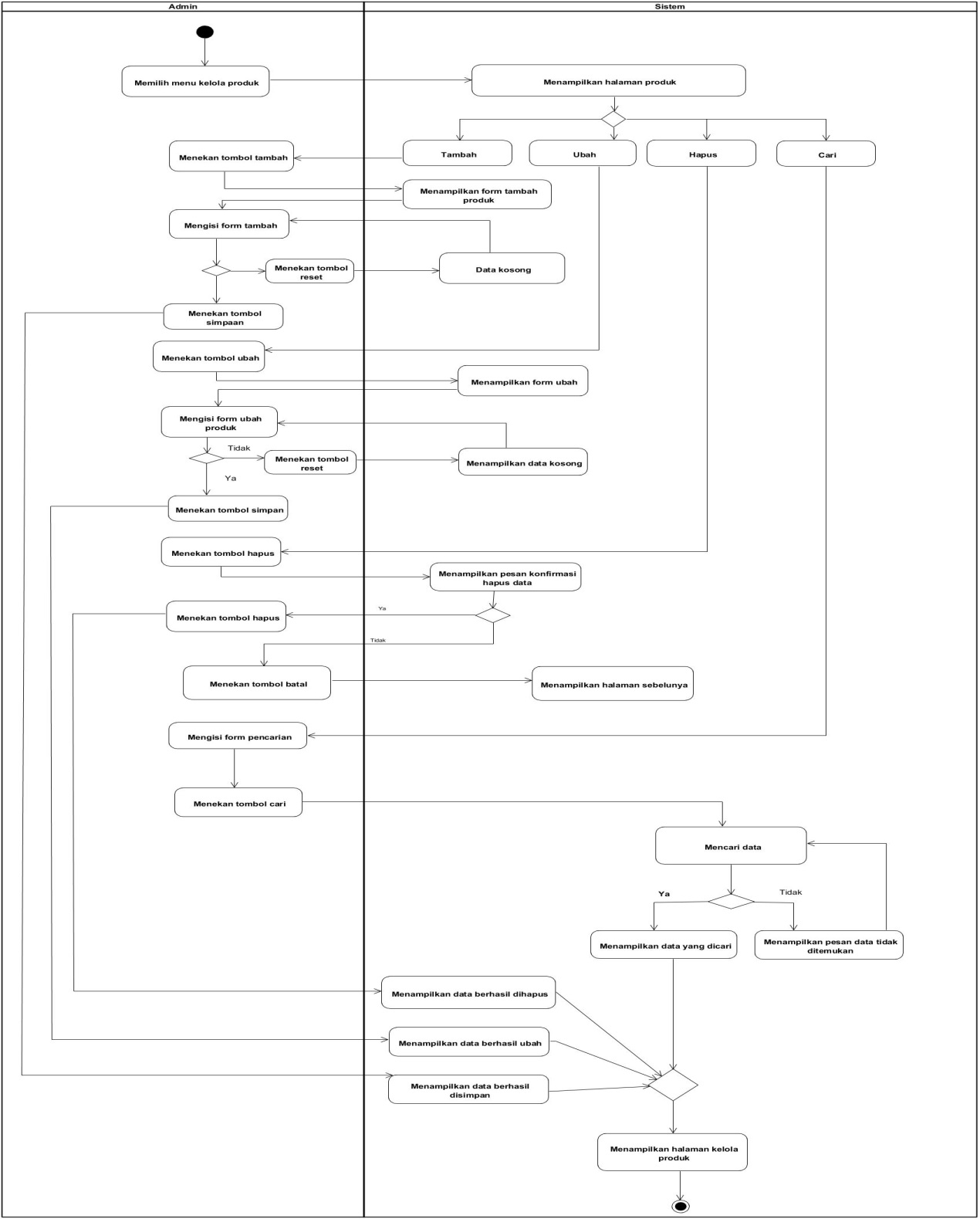
|  |
| --- |
|  |
| Gambar 3. 15 *Swimlane Diagram* Beranda Admin |

Berdasarkan Gambar 3.15 *Swimlane* *Diagram* Beranda Adminmemperlihatkan halaman adminseperti statistic pendapatan serta status pemesanan produk yang telah atau sedang dilakukan.

1. *Swimlane Diagram* Kelola Kategori

|  |
| --- |
|  |
| Gambar 3. 16 *Swimlane Diagram* Kelola Kategori |

Berdasarkan Gambar 3.16 *Swimlane* *Diagram* Kelola Kategorimenunjukan aktifitas admin yang memiliki akses tambah, ubah, hapus, dan cari data produk berdasarkan kategori.

1. *Swimlane Diagram* Kelola Produk

|  |
| --- |
|  |
| Gambar 3. 17 *Swimlane Diagram* Kelola Produk |

Berdasarkan Gambar 3.17 *Swimlane* *Diagram* Kelola Produkmenunjukan aktifitas admin yang memiliki akses tambah, ubah, hapus, dan cari data produk.

1. *Swimlane Diagram* *Invoice*

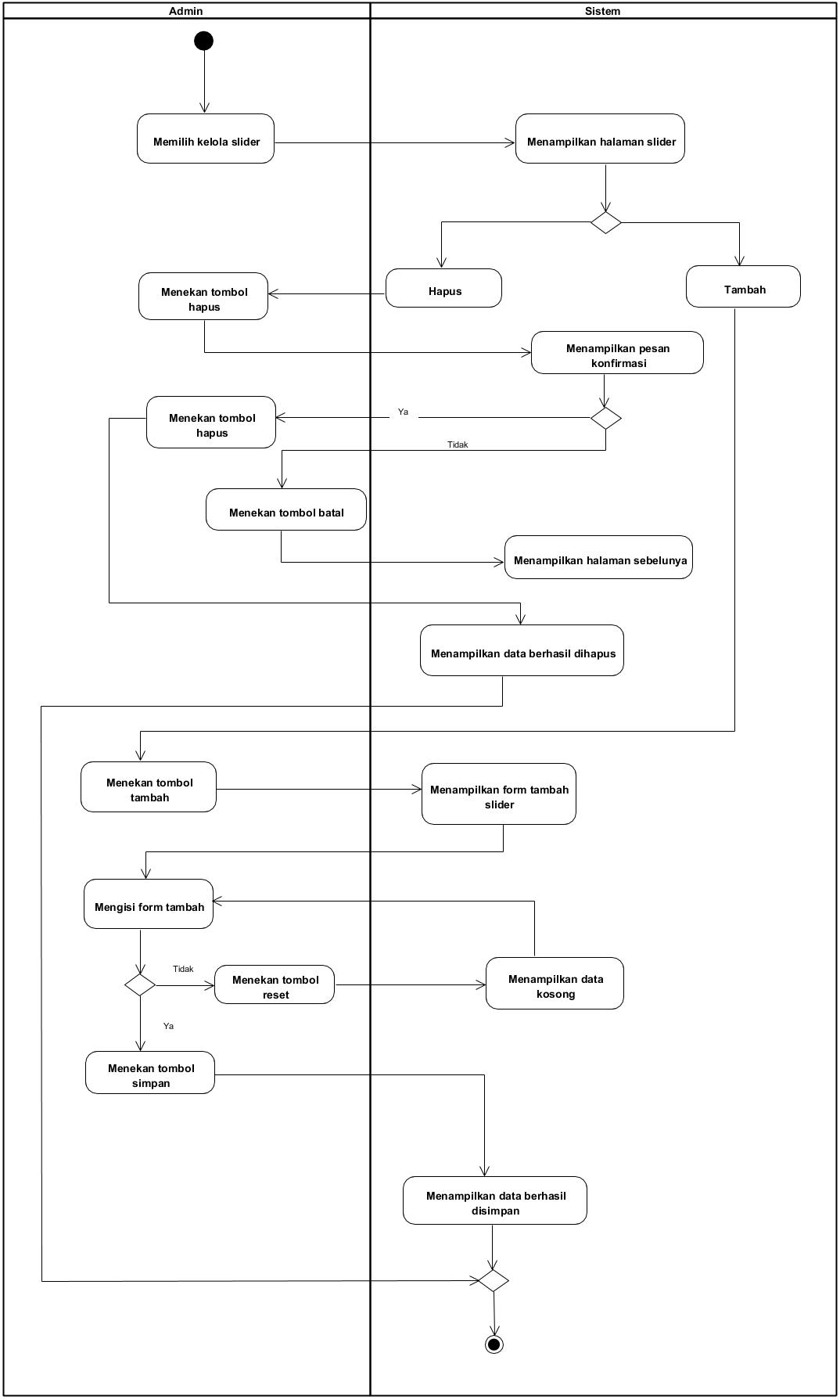
|  |
| --- |
|  |
| Gambar 3. 18 *Swimlane Diagram* *Invoice* |

Berdasarkan Gambar 3.18 *Swimlane* *Diagram* *Invoice* memperlihatkan detail *invoice* serta pencarian data mengenai *invoice* dari penjualan yang telah dilakukan.

1. *Swimlane Diagram* Kelola *Customer*

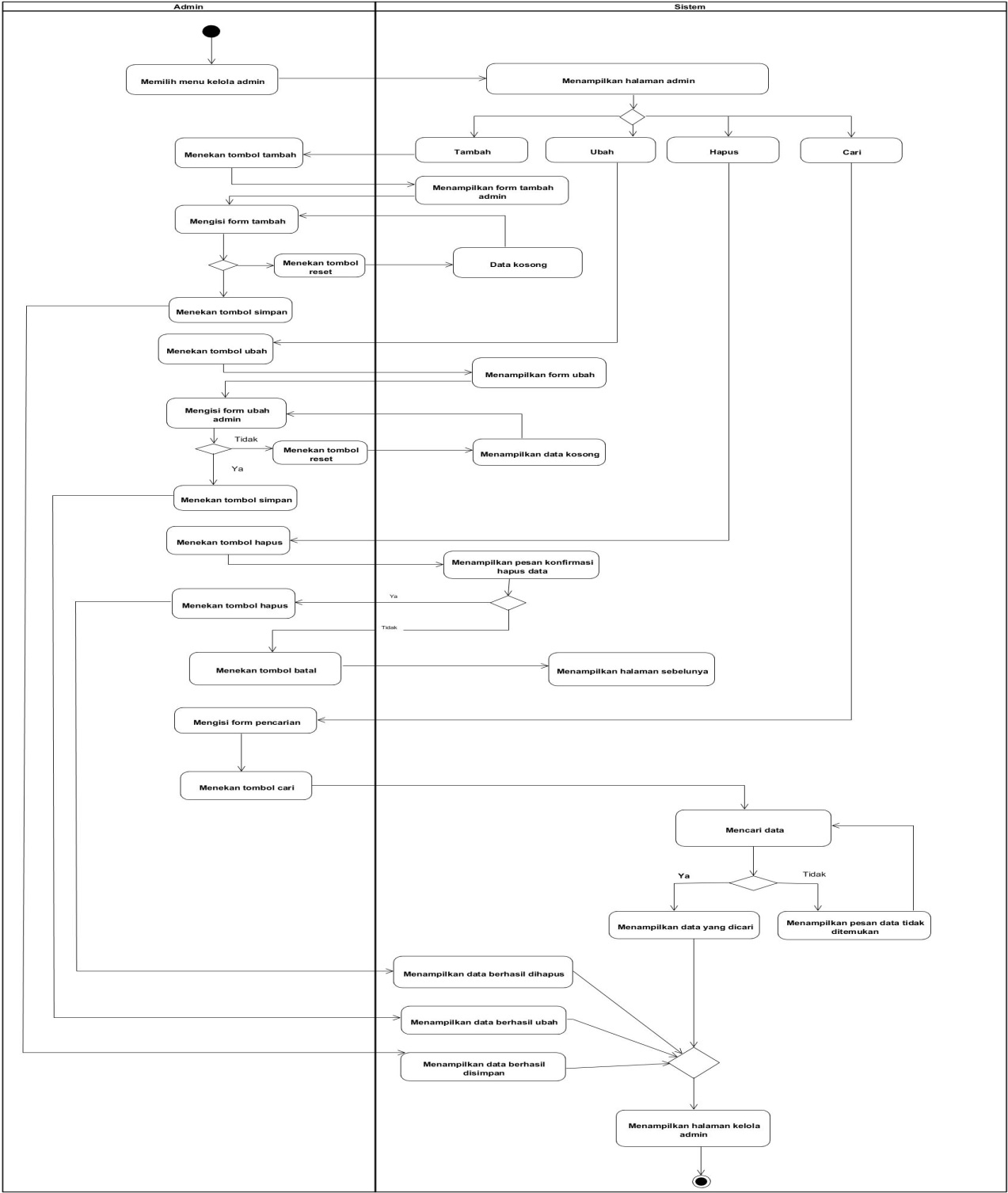
|  |
| --- |
|  |
| Gambar 3. 19 *Swimlane Diagram* Kelola *Customer* |

Berdasarkan Gambar 3.19 *Swimlane Diagram* Kelola *Customer* memperlihatkan *customer* yang telah bergabung pada sistem, admin dapat melakukan pencarian *customer* tersebut.

1. *Swimlane Diagram* Kelola *Slider*

|  |
| --- |
|  |
| Gambar 3. 20 *Swimlane Diagram* Kelola *Slider* |

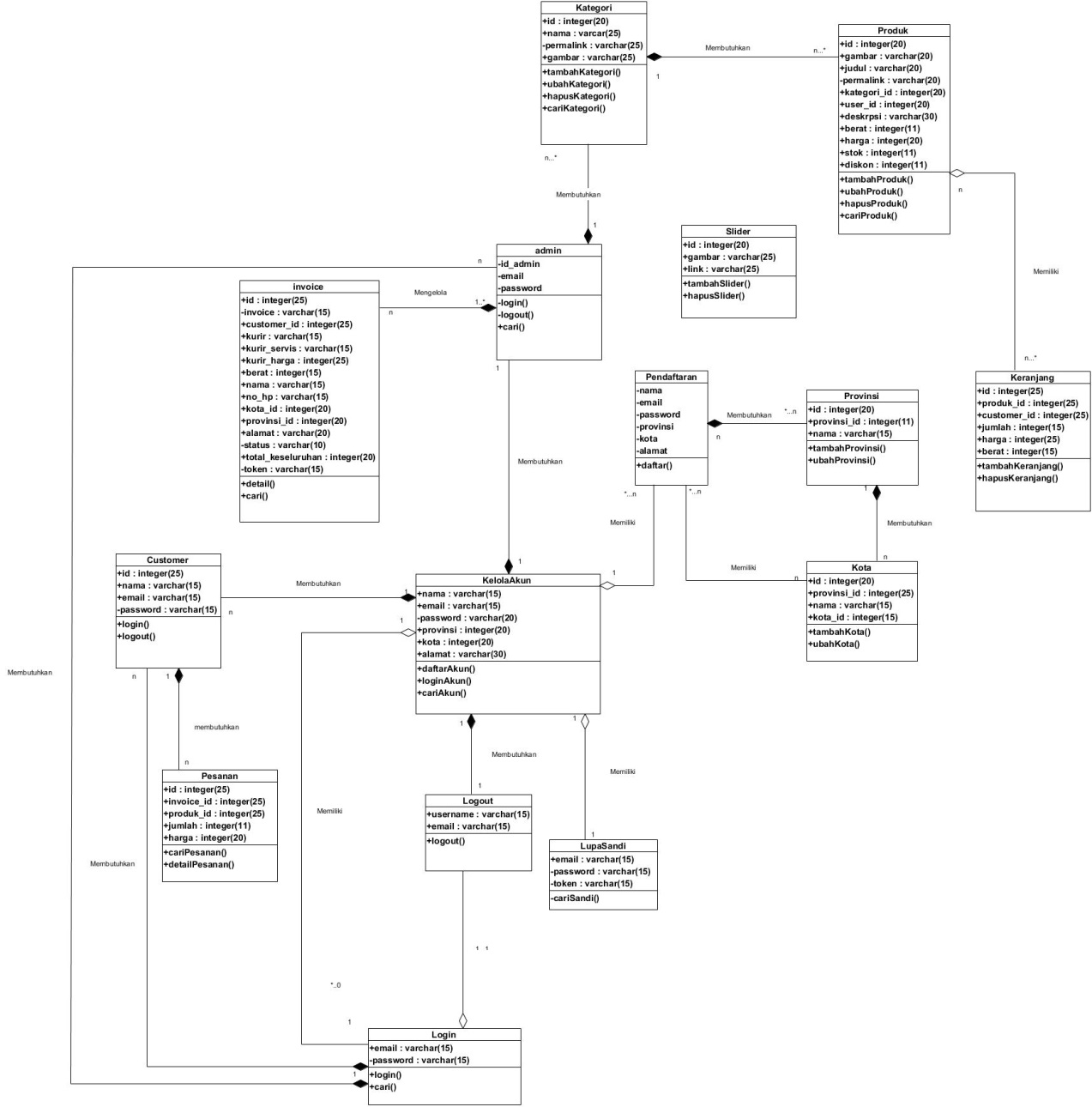
Berdasarkan Gambar 3.20 *Swimlane Diagram* Kelola *Slider* menunjukan aktifitas admin yang memiliki akses untuk menambah dan menghapus gambar atau *design* yang akan tampil pada halaman utama.

1. *Swimlane Diagram* Kelola Admin

|  |
| --- |
|  |
| Gambar 3. 21 *Swimlane Diagram* Kelola Admin |

Berdasarkan Gambar 3.21 *Swimlane Diagram* Kelola Admin menunjukan aktifitas admin yang memiliki akses untuk tambah, ubah, cari, dan hapus data admin,.

## *Class Diagram*

Berikut ini adalah Class Diagram pembangunan *e-commerce* penjualan perangkat komputer di Gemilang Jaya *Computer*:

|  |
| --- |
|  |
| Gambar 3. 22 *Class Diagram* |

Berdasarkan gambar 3.20 Class Diagram pembangunan *e-commerce* penjualan perangkat komputer di Gemilang Jaya *Computer* terdiri dari 16 class yaitu kelola akun, *login, logout,* lupa sandi, pendaftaran, admin, *invoice,* pesanan, kategori, produk, *slider,* *customer,* provinsi, kota, keranjang, *riview* dapat diketahui hubungan semua class yang ada di system tersebut, class kelola akun memiliki hubungan asosiasi dengan class yang lainnya, dikarenakan class tersebut memiliki ketergantungan terhadap class kelola akun.

Beberapa class memiliki hubungan *strong* *agregation* dengan beberapa class lainnya karena jika class tersebut hilang atau tidak digunakan maka class lain juga hilang atau ketergantungan penuh. Selain itu terdapat beberapa class yang memiliki hubungan *composition* dengan beberapa class lainnya, jika class tersebut tidak ada atau hilang, masih bisa digunakan atau ketergantungan lemah, selain itu setiap class memiliki kardinalitas sesuai dengan atribut class yang bersangkutan.

## *Sequance Diagram*

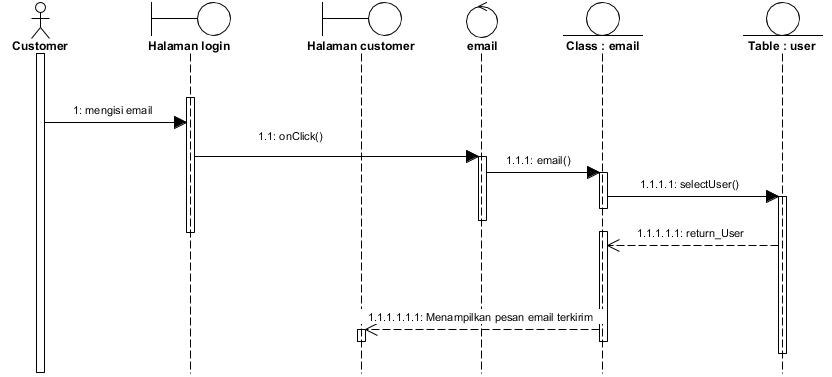
Berikut ini adalah Sequence Diagram pembangunan *e-commerce* penjualan perangkat komputer di Gemilang Jaya *Computer* adalah sebagai berikut:

1. *Sequance Diagram* Kelola Akun

|  |
| --- |
|  |
| Gambar 3. 23 *Sequance Diagram* Kelola Akun |

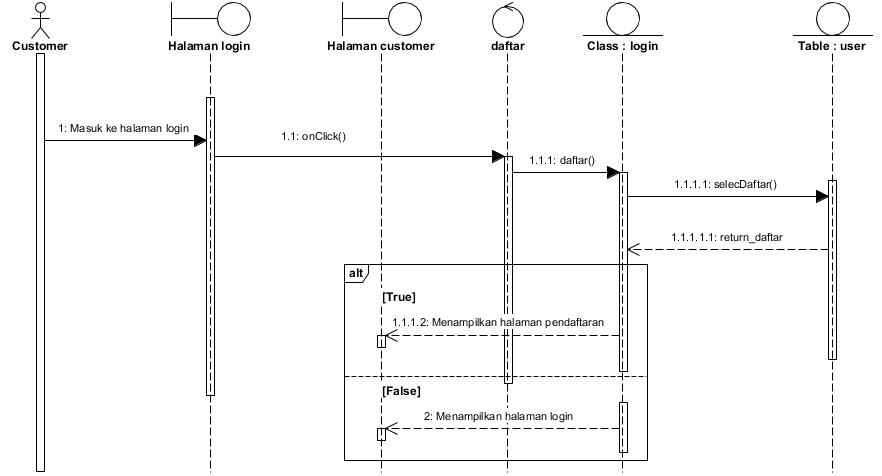
Berdasarkan Gambar 3.23 *Sequance* *Diagram* Kelola Akun menunjukan langkah langkah atau aktivitas *user* dalam melakukan pembuatan akun, agar *user*

memiliki akun yang terdaftar dalam sistem.

1. *Sequance* *Diagram* Lupa Sandi

|  |
| --- |
|  |
| Gambar 3. 24 *Sequance Diagram* Lupa Sandi |

Berdasarkan Gambar 3.24 *Sequance Diagram* Lupa Sandi menunjukan langkah atau aktivitas *customer* ketika lupa *password,* hal tersebut sebagai alternatif dari permaslahan tersebut.

1. *Sequance* *Diagram* Pendaftaran

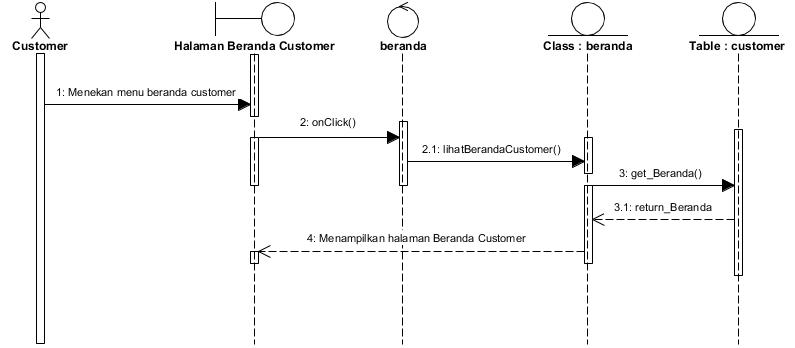
|  |
| --- |
|  |
| Gambar 3. 25 *Sequance Diagram* Pendaftaran |

Berdasarkan Gambar 3.25 *Sequance Diagram* Pendaftaran menunjukan langkah langkah atau aktivitas *customer* dalam melakukan pendaftaran, agar *user* memiliki akun yang terdaftar dalam sistem.

1. *Sequance Diagram* *Login*

|  |
| --- |
|  |
| Gambar 3. 26 *Sequance Diagram* *Login* |

Berdasarkan Gambar 3.26 *Sequance* *Diagram* *Login* menunjukan langkah langkah atau aktivitas *user* masuk kedalam sistem dengan memasukan *email* dan *password* yang telah terdaftar dalam sistem

1. *Sequance Diagram* Beranda *Customer*

|  |
| --- |
|  |
| Gambar 3. 27 *Sequance Diagram* Beranda *Customer* |

Berdasarkan Gambar 3.27 *Swimlane* Diagram Beranda *Customer* memperlihatkan halaman *customer* seperti nama *customer* serta status pemesanan produk yang telah atau sedang dilakukan.

1. *Sequance Diagram* Kelola Akun *Customer*

|  |
| --- |
|  |
| Gambar 3. 28 *Sequance Diagram* Kelola Akun *Customer* |

Berdasarkan Gambar 3.28 *Sequance* Diagram Kelola Akun *Customer* menunjukan langkah-langkah atau aktivitas *customer* yang memiliki akses untuk mencari dan mengubah *password* pada akun yang digunakan.

1. *Sequance Diagram* Pesanan Barang

|  |
| --- |
|  |
| Gambar 3. 29 *Sequance Diagram* Pesanan Barang |

Berdasarkan Gambar 3.29 *Sequance Diagram* Pesanan Barangmenunjukan langkah-langkah atau aktivitas *customer* dalam mencari pesananan, serta detail dari pembelian produk.

1. *Sequance Diagram* *Logout*

|  |
| --- |
|  |
| Gambar 3. 30 *Sequance Diagram* *Logout* |

Berdasarkan Gambar 3.30 *Sequance Diagram* *Logout* menunjukan langkah atau aktivitas *user* keluar dari sistem atau aplikasi, aktivitas tersebut dilakukan ketika *user* sedang berada dalam sistem.

.

1. *Sequance Diagram* Beranda Admin

|  |
| --- |
|  |
| Gambar 3. 31 *Sequance Diagram* Beranda Admin |

Berdasarkan Gambar 3.31 *Sequance* *Diagram* Beranda Adminmemperlihatkan halaman adminseperti *statistic* pendapatan serta status pemesanan produk yang telah atau sedang dilakukan.

1. *Sequance Diagram* Kelola Kategori

|  |
| --- |
|  |
| Gambar 3. 32 *Sequance Diagram* Kelola Kategori |

Berdasarkan Gambar 3.32 *Sequance* *Diagram* Kelola Kategorimenunjukan aktifitas admin yang memiliki akses tambah, ubah, hapus, dan cari data produk berdasarkan kategori.

1. *Sequance Diagram* Kelola Produk

|  |
| --- |
|  |
| Gambar 3. 33 *Sequance Diagram* Kelola Produk |

Berdasarkan Gambar 3.17 *Swimlane* *Diagram* Kelola Produkmenunjukan aktifitas admin yang memiliki akses tambah, ubah, hapus, dan cari data produk.

1. *Sequance Diagram* *Invoice*

|  |
| --- |
|  |
| Gambar 3. 34 *Sequance Diagram* *Invoice* |

Berdasarkan Gambar 3.34 *Sequance* *Diagram* *Invoice* memperlihatkan detail *invoice* serta pencarian data mengenai *invoice* dari penjualan yang telah dilakukan.

1. *Swimlane Diagram* Kelola *Customer*

|  |
| --- |
|  |
| Gambar 3. 35 *Sequance* Kelola *Customer* |

Berdasarkan Gambar 3.35 *Swimlane Diagram* Kelola *Customer* memperlihatkan *customer* yang telah bergabung pada sistem, admin dapat melakukan pencarian *customer* tersebut.

1. *Sequance Diagram* Kelola *Slider*

|  |
| --- |
|  |
| Gambar 3. 36 *Sequance* Kelola *Slider* |

Berdasarkan Gambar 3.36 *Sequance Diagram* Kelola *Slider* menunjukan aktifitas admin yang memiliki akses untuk menambah dan menghapus gambar atau *design* yang akan tampil pada halaman utama.

1. *Sequance Diagram* Kelola Admin

|  |
| --- |
|  |
| Gambar 3. 37 *Sequance Diagram* Kelola Admin |

Berdasarkan Gambar 3.37 *Sequance* *Diagram* Kelola Admin menunjukan aktifitas admin yang memiliki akses untuk tambah, ubah, cari, dan hapus data admin,.

## *Sequance Diagram*

Berikut ini adalah *Entity Relation Diagram (ERD)* pembangunan *e-commerce* penjualan perangkat komputer di Gemilang Jaya *Computer*:

|  |
| --- |
|  |
| Gambar 3. 38 *Entity Relation Diagram* |

Keterangan

Berdasarkan Gambar 3.38 Terdapat 10 tabel yakni *user, customer,* keranjang, produk, kategori, pesanan, *slider, invoice,* provinsi, dan kota setiap tabel saling terhubung satu sama lain dengan *primary key* ataupun *foreign key* yang terdapat dalam setiap tabel yang bersangkutan, dalam satu tabel memungkinkan memilki *foreign key* lebih dari satu, sedangkan kardinalitas terdapat *one to one, one to many*, dan *many to many,* yang implementasinya terdapat pada gambar *ERD* (*Entity Relationship Diagram)* di atas.

## *Struktur Table*

Struktur *tabel* menggambarkan detail tabel yang berisi *field*, tipe data, panjang data, dan keterangan lainnya. Berikut ini merupakan rancangan tabel pada database Gemilang Jaya *Computer* :

1. Tabel Keranjang

Tabel 3. 4 Tabel Keranjang

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Produk\_id | Integer | 25 | *Foreign key* |
| Customer\_id | Integer | 25 | *Foreign key* |
| Quantity | Integer | 15 |  |
| Harga | Integer | 25 |  |
| Berat | Integer | 15 |  |

1. Tabel Kategori

Tabel 3. 13 Tabel Kategori

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Nama | Varchar | 15 |  |
| Slug | Varchar | 30 |  |
| Gambar | Varchar | 25 |  |

1. Tabel Kota

Tabel 3. 14 Tabel Kota

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Provinsi\_id | Integer | 25 | *Foreign key* |
| Kota\_id | Integer | 25 | *Foreign key* |
| Nama | Varchar | 15 |  |

1. Tabel *Customer*

Tabel 3. 15 Tabel *Customer*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Nama | Varchar | 15 |  |
| Email | Varchar | 15 |  |
| Password | Vachar | 20 |  |

1. Tabel *Invoice*

Tabel 3. 16 Tabel *Invoice*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| *Invoice* | Varchar | 25 |  |
| Customer\_id | Varchar | 25 | *Foreign key* |
| Kurir | Integer | 10 |  |
| Service\_kurir | Integer | 10 |  |
| Harga\_kurir | Integer | 10 |  |
| Berat | Integer | 15 |  |
| Nama | Varchar | 15 |  |
| No\_hp | Varchar | 15 |  |
| Kota\_id | Integer | 25 | *Foreign key* |
| Provinsi\_id | Integer | 25 | *Foreign key* |
| Alamat | Varchar | 25 |  |
| Status | Varchar | 10 |  |
| Grand\_total | Integer | 25 |  |
| Token | Varchar | 25 |  |

1. Tabel Pesanan

Tabel 3. 17 Tabel Pesanan

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| *Invoice\_id* | Integer | 25 |  |
| Produk\_id | Integer | 25 | *Foreign key* |
| Quantity | Integer | 10 |  |
| Harga | Integer | 25 |  |

1. Tabel Produk

Tabel 3. 18 Tabel Produk

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Gambar | Varchar | 25 |  |
| Judul | Varchar | 25 |  |
| Slug | Varchar | 25 |  |
| Kategori\_id | Integer | 25 | *Foreign key* |
| User\_id | Integer | 25 | *Foreign key* |
| Deskripsi | Varchar | 30 |  |
| Berat | Integer | 11 |  |
| Harga | Integer | 25 |  |
| Stok | Integer | 11 |  |
| Diskon | Integer | 11 |  |

1. Tabel Provinsi

Tabel 3. 19 Tabel Provinsi

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Provinsi\_id | Integer | 11 | *Foreign key* |
| Nama | Varchar | 15 |  |

1. Tabel *Slider*

Tabel 3. 21 Tabel *Slider*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Gambar | Varchar | 25 |  |
| *Link* | Varchar | 25 |  |

1. Tabel *User*

Tabel 3. 22 Tabel *Slider*

|  |  |  |  |
| --- | --- | --- | --- |
| **Nama Field** | **Tipe Data** | **Panjang** | **Deskripsi** |
| Id | Integer | 25 | *Primary key* |
| Nama | Varchar | 15 |  |
| *Email* | Varchar | 20 |  |
| *Password* | Varchar | 20 |  |

## Perancangan Struktur Menu

Perancangan struktur menu adalah gambaran jalur pemakaian yang akan dibangun. Berikut merupakan struktur menu Aplikasi *E-Commerce* Penjualan Perangkat Komputer Di Gemilang Jaya *Computer*.

|  |
| --- |
|  |
| Gambar 3. 39 Perancangan Struktur Menu Gemilang Jaya *Computer* |

## Perancangan Antarmuka

# DAFTAR PUSTAKA

Wardani, R., & Si, S. (2012). Program Pendidikan dan Pelatihan Profesi Guru Teknologi Informasi dan Komunikasi Rekayasa Perangkat Lunak. *Program Pendidikan Dan Pelatihan Profesi Guru Teknologi Informasi Dan Komunikasi*, 1–43.

García Reyes, L. E. (2013). *Journal of Chemical Information and Modeling*, *53*(9), 1689–1699.

Putra, D. A., Sasmita, G. M. A., & Wiranatha, A. K. A. C. (2020). E-Commerce Marketplace Petshop Menggunakan Integrasi Rajaongkir API dan iPaymu Payment Gateway API. *JITTER-Jurnal Ilmiah Teknologi Dan Komputer*, *1*(1).

Budi, D. S., Siswa, T. A. Y., & Abijono, H. (2017). Analisis Pemilihan Penerapan Proyek Metodologi Pengembangan Rekayasa Perangkat Lunak. *Teknika*, *5*(1), 24–31. https://doi.org/10.34148/teknika.v5i1.48

Raushan Fikri Tunny, & Aji, A. S. (2020). NASKAH PUBLIKASI PERANCANGAN APLIKASI E-COMMERCE PENJUALAN MENGGUNAKAN PAYMENT GATEWAY (Studi kasus: M22 Celluler Yogyakarta). 12.

Ghifary, Muhammad. 2010. Rancangan Protokol Modifikasi Transaksi Belanja Online dengan Kartu Kredit Secara CNP (Card Not Present).

Rini, Dian Palupi. 2010 Metodologi Pengembangan Sistem Informasi.

Mukherjee, Momim, dan Sahadev Roy. 2017. “E-Commerce and Online Payment in the Modern Era.” International Journal of Advanced Research in Computer Science and Software Engineering 7 (5): 1-5.

Marta, Revi Fajar. 2010. Implementasi Kriptografi Pada E-Commerce.

Riyanto, Ema Utami, dan Armadiyah Amborowaty. 2013. “Pemanfaatan Web Service Pada Integrasi Data Farmasi Di RSU Banyumas.” Seminar Nasional Informatika 2013 (SemnasIF 2013).

Akhir, B. T. (n.d.). *Tugas Akhir Teknik Informatika*. 23–34.

Sutrisno, J., Strategi Pengembangan Teknologi E-Commerce Dengan Metode SWOT: Studi Kasus PT. Chingmix Berhan Sejahtera, Jurnal Telematika Mkom, Vol.3 No.2, September, 2011