

**PROYEK III**

**PROGRAM STUDI DIV TEKNIK INFORMATIKA**



**PENGEMBANGAN APLIKASI SISTEM INVENTORI BERBASIS WEB DENGAN  
FRAMEWORK CODE IGNITER**

**ANGGOTA :**

JOSUANSEF PARDEDE      1.18.4.091 D4TI3A

OKKY YUDISTIRA          1.18.4.087 D4TI3A

**PEMBIMBING :**

NISA HANUM HARANI, S.KOM., M.T.

NIK : 117.89.223

**PROGRAM DIPLOMA IV TEKNIK INFORMATIKA**

**POLITEKNIK POS INDONESIA**

**BANDUNG**

**2020**

## **ABSTRAK**

Sistem pergudangan/ inventori menjadi aspek yang paling penting dan prioritas dalam suatu online *marketplace* (*e- marketplace*). Sistem pergudangan/ inventori berguna untuk mengelola persediaan barang/ stok baik itu dari barang masuk dan barang keluar, perjalanan barang serta pengadaan stok barang.

Sebelumnya, perusahaan-perusahaan menggunakan sistem inventori secara manual melalui tenaga kerja sdm yang dibutuhkan secara gesit dan akurat. Sekarang, pengelolaan sistem inventori inipun dapat dilakukan secara otomatis dan tentunya banyak nilai tinggi dan keunggulan jika suatu pengelolaan stok dilakukan secara otomatis. Sistem inventori ini sangat dibutuhkan oleh perusahaan-perusahaan (supplier) dari produk/ barang yang mereka punya guna memakai jasa penyimpanan gudangnya.

Sistem yang digunakan untuk membuat dan melakukan pengembangan aplikasi sistem inventori pada suatu perusahaan/ *e- marketplace* harus efektif. Dengan sistem *database* yang baik, sistem CRUD yang fungsional dan kompleks, fungsi-fungsi sistem yang mendukung dan dengan *Framework Code Igniter* serta dengan *friendly user interface* akan berguna untuk mewujudkan efisiensi dan keefektifan pengelolaan stok pada suatu *e- marketplace*.

Sistem inventori ini juga bisa menjadi media platform untuk menghubungkan pihak satu (supplier/ penjual di olshop maupun toko offline) dengan pihak satunya lagi yaitu gudang/ tempat penyimpanan barang produksi/ tempat kerja. Jadi sistem inventori ini yang menjembatani hubungan antar supplier/ penjual dengan gudang.

**Kata Kunci :** *Gudang, Pengembangan, Inventori, Stok, Framework Code Igniter*

## DAFTAR ISI

ABSTRAK.....	2
DAFTAR ISI.....	3
DAFTAR GAMBAR .....	6
DAFTAR TABEL.....	9
BAB I.....	10
PENDAHULUAN .....	10
1.1    Latar Belakang .....	10
1.2    Idenifikasi Masalah .....	11
1.3    Tujuan.....	11
1.4    Ruang Lingkup.....	12
1.5    Sistematika Penulisan.....	12
BAB II.....	14
STATE OF THE ART .....	14
2.1    Sistem Informasi .....	14
2.2    Unified Modelling Language (UML).....	14
2.3    Flowmap.....	24
2.4    Teori Umum .....	25
BAB III.....	34
ANALISIS DAN PERANCANGAN SISTEM.....	34
3.1    Analisis Sistem .....	34
3.1.1    Analisis Sistem yang Sedang Berjalan .....	34
3.1.1.1    Analisis Prosedur/ Flow Map Berjalan .....	34
3.1.1.2    Analisis Dokumen yang Digunakan .....	36
3.1.1.3    Analisis User.....	38

3.1.2	Analisis Sistem yang akan Dibangun .....	38
3.1.2.1	Analisis Kebutuhan Aplikasi .....	38
3.1.2.2	Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras.....	39
3.2	Perancangan.....	40
3.2.1	Usecase Diagram .....	41
3.2.2	Class Diagram .....	49
3.2.3	Activity Diagram .....	51
3.2.4	Sequence Diagram .....	63
3.2.5	Statechart Diagram.....	66
3.2.6	Deployment Diagram.....	73
BAB IV .....		74
IMPLEMENTASI DAN PENGUJIAN .....		74
4.1	Implementasi .....	74
4.1.1	Implementasi Perangkat Keras .....	74
4.1.2	Implementasi Perangkat Lunak.....	75
4.2	Perancangan Database.....	76
4.3	Tampilan Antar Muka Web.....	79
4.3.1	Tampilan Halaman Login .....	80
4.3.2	Tampilan Registrasi .....	81
4.3.3	Tampilan Utama (User Admin Login).....	81
4.3.4	Tampilan Utama (User Gudang Login) .....	82
4.3.5	Menu Supplier.....	82
4.3.6	Barang / Produk .....	83
4.3.7	Cetak Laporan / Laporan Transaksi .....	87
4.3.8	Manajemen User .....	87

4.3.9	User Profile .....	88
4.3.10	Pengaturan.....	88
4.3.11	Ubah Password.....	89
BAB V .....		90
KESIMPULAN DAN SARAN.....		90
5.1	Kesimpulan.....	90
5.2	Saran.....	90
DAFTAR PUSTAKA .....		91

## DAFTAR GAMBAR

Gambar 1 Simbol Usecase Diagram .....	17
Gambar 2 Simbol Class Diagram .....	18
Gambar 3 Simbol Activity Diagram .....	18
Gambar 4 Simbol Sequence Diagram .....	19
Gambar 5 Simbol Statechart Diagram .....	20
Gambar 6 Simbol Component Diagram .....	21
Gambar 7 Simbol Deployment Diagram .....	22
Gambar 8 Simbol Collaboration Diagram .....	23
Gambar 9 Simbol Flowmap .....	24
Gambar 10 Ilustrasi Pergudangan .....	26
Gambar 11 Ilustrasi Website .....	26
Gambar 12 Ilustrasi SQL .....	27
Gambar 13 Ilustrasi Kode Javascript .....	28
Gambar 14 Logo XAMPP .....	29
Gambar 15 Ilustrasi CSS Tata letak Horizontal dan Vertical .....	30
Gambar 16 Logo PHP .....	30
Gambar 17 Ilustrasi Penggambaran Sebuah Sistem .....	31
Gambar 18 Flowmap Pendataan Stok Barang .....	35
Gambar 19 Flowmap Transaksi Barang Masuk .....	35
Gambar 20 Flowmap Transaksi Barang Keluar .....	35
Gambar 21 Usecase Diagram Aplikasi Sistem Inventori .....	42
Gambar 22 Class Diagram .....	49
Gambar 23 Activity Diagram Log in .....	52
Gambar 24 Activity Diagram Log out .....	52
Gambar 25 Activity Diagram Tambah Data Barang .....	53
Gambar 26 Activity Diagram Edit Data Barang .....	54
Gambar 27 Activity Diagram Hapus Data Barang .....	55
Gambar 28 Activity Diagram Barang Masuk .....	55
Gambar 29 Activity Diagram Barang Keluar .....	56

Gambar 30 Activity Diagram Tambah User .....	57
Gambar 31 Activity Diagram Edit User .....	58
Gambar 32 Activity Diagram Hapus User.....	59
Gambar 33 Activity Diagram Tambah Supplier .....	59
Gambar 34 Activity Diagram Edit Supplier .....	60
Gambar 35 Activity Diagram Hapus Supplier.....	61
Gambar 36 Activity Diagram Aktifkan User.....	61
Gambar 37 Activity Diagram Nonaktifkan User .....	62
Gambar 38 Activity Diagram Cetak Laporan .....	62
Gambar 39 Sequence Diagram Login.....	63
Gambar 40 Sequence Diagram Data Barang (Stok) .....	64
Gambar 41 Sequence Diagram Barang Masuk .....	64
Gambar 42 Sequence Diagram Barang Keluar .....	65
Gambar 43 Sequence Diagram Supplier.....	65
Gambar 44 Sequence Diagram Laporan .....	66
Gambar 45 Statechart Diagram Login .....	67
Gambar 46 Statechart Diagram Sign in .....	68
Gambar 47 Statechart Diagram User .....	68
Gambar 48 Statechart Diagram Data Barang (Stok).....	69
Gambar 49 Statechart Diagram Barang Masuk .....	70
Gambar 50 Statechart Diagram Barang Keluar .....	71
Gambar 51 Statechart Diagram Supplier .....	71
Gambar 52 Statechart Diagram Laporan .....	72
Gambar 53 Deployment Diagram .....	73
Gambar 54 Tampilan XAMP saat mulai .....	76
Gambar 55 Database Aplikasi Sistem Inventori dan Tabel-Tabelnya.....	76
Gambar 56 Desain Database Aplikasi Sistem Inventori.....	77
Gambar 57 Fungsi Trigger pada Aplikasi Sistem Inventori .....	77
Gambar 58 Fungsi Trigger update_stok_keluar .....	78
Gambar 59 Fungsi Trigger update_stok_masuk .....	79
Gambar 60 Tampilan Login.....	80

Gambar 61 Tampilan Menu Sign In (Registrasi) pada Aplikasi.....	81
Gambar 62 Tampilan Utama Aplikasi Sebagai User Admin.....	81
Gambar 63 Tampilan Utama Aplikasi Sebagai User Gudang .....	82
Gambar 64 Tampilan Menu Data Supplier .....	82
Gambar 65 Menu Satuan Barang.....	83
Gambar 66 Menu Jenis Barang.....	83
Gambar 67 Menu Merek Barang .....	84
Gambar 68 Menu Harga Barang .....	84
Gambar 69 Menu Status Barang .....	85
Gambar 70 Tampilan Data barang.....	85
Gambar 71 Tampilan dari barang masuk.....	86
Gambar 72 Tampilan Barang keluar.....	86
Gambar 73 Menu Cetak Laporan Transaksi .....	87
Gambar 74 Tampilan Manajemen user.....	87
Gambar 75 Menu Profile .....	88
Gambar 76 Tampilan menu Setting atau pengaturan pada profile .....	88
Gambar 77 Tampilan Fitur Password pada menu profile .....	89



## DAFTAR TABEL

Tabel 1 Analisis Dokumen Sistem yang Sedang Berjalan.....	36
Tabel 2 Kebutuhan Perangkat Keras.....	39
Tabel 3 Kebutuhan Perangkat Lunak.....	40
Tabel 4 Definisi Aktor .....	42
Tabel 5 Penjelasan Usecase .....	43
Tabel 6 Skenario Usecase Login.....	45
Tabel 7 Skenario Usecase Stok Barang .....	46
Tabel 8 Skenario Usecase Kelola Barang Masuk .....	46
Tabel 9 Skenario Usecase Kelola Barang Keluar .....	47
Tabel 10 Skenario Usecase Kelola Supplier .....	47
Tabel 11 Skenario Usecase Kelola User .....	48
Tabel 12 Skenario Usecase Kelola Laporan .....	48
Tabel 13 Tabel Implementasi Perangkat Keras .....	74
Tabel 14 Tabel Implementasi Perangkat Lunak .....	75

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Sistem inventori atau disebut juga aktivitas barang masuk dan barang keluar merupakan suatu kegiatan dalam proses pengolahan data barang baik barang masuk, barang keluar, stok dan perjalanan barang yang ada dalam suatu gudang sistem inventori. Sistem pergudangan ini juga memiliki pengaruh besar terhadap suatu perusahaan/ instansi khususnya *e- marketplace* karena dapat mewujudkan keefesiensian dan keefektian dalam menyelesaikan masalah pengolahan data barang dan memudahkan pelaporan data barang tersebut. [1]

Perusahaan/ instansi dan *e- marketplace* yang tidak memiliki manajemen inventori atau gudang biasanya akan mengalami beberapa masalah pengolahan dan pendataan barang. Aktivitas pendataannya meliputi: pendataan barang masuk, barang keluar, stok, stok lama yang masih layak digunakan, bahkan barang yang rusak dan harus diganti. Setelah itu *history* data tersebut dibuat dalam bentuk laporan data inventori/ pergudangan perusahaan atau *e- marketplace* tersebut dari petugas/ manajer gudang kepada pihak instansi dengan menggunakan *Microsoft excel*.

Proses pendataan manual seperti dengan *Microsoft excel* bisa dikatakan kurang efektif karena sangat dibutuhkan ketelitian dan keakuratan dari SDM-nya dalam melakukan kegiatan pendataan dengan jumlah barang yang tidak sedikit. Pada hal ini, sering terjadi masalah yang umum tapi berakibat fatal yaitu redundansi data (data ganda/ data yang duplikat).

Dibutuhkan proses yang tidak sebentar dalam proses pembuatan laporan data inventori/ pergudangan karena petugas gudang yang bersangkutan harus membuat data baru yang akurat untuk memenuhi laporan. Bahkan faktor *human error* yang dapat mengakibatkan masalah pada aktivitas pergudangan dapat membuat pekerjaan menjadi *double* dan tidak efektif secara waktu.

Oleh karena itu, sistem yang unggul dalam membantu manajemen pengolahan inventori/ gudang sangat dibutuhkan dalam suatu perusahaan/ instansi khususnya pada *e- marketplace*. Dengan sistem yang baik, fungsi-fungsi yang mendukung, kinerja yang akurat serta UI yang

*user friendly* dapat membuat aktivitas pengolahan inventori/ pergudangan menjadi efisien dan efektif.

## 1.2 Identifikasi Masalah

Dalam pengembangan aplikasi sistem inventori berbasis web pada marketplace dengan framework code igniter terdapat batasan masalah atau ruang lingkup pengerjaan agar sistem dapat dikembangkan sesuai dengan tujuan utama. Agar pengembangan aplikasi dapat dengan mudah dimengerti tetapi tidak mengurangi tujuan penelitian, maka penulis membatasi perancangan aplikasi tersebut sebagai berikut :

1. Menjadi media platform virtual yang menghubungkan/ menjembatani gudang dan penjual/ supplier yang membutuhkan tambahan tempat/ gudang utama dari barang mereka (olshop dan toko offline).
2. Sistem ini menghasilkan laporan data barang (masuk, keluar, stok) dan laporan perjalanan kendaraan serta transportasi yang sesuai dengan tujuan barang.
3. Bahasa pemrograman yang digunakan sebagai pembangun sistem adalah PHP berbasis web dengan menggunakan *framework Code Igniter*(CI) yang berbasis OOP, selanjutnya pengolahan *database*-nya / DBMS menggunakan MySQL.
4. Aplikasi ini menggunakan *user interface* yang telah dikembangkan dari aplikasi versi sebelumnya (proyek 2).

## 1.3 Tujuan

Berdasarkan latar belakang dan batasan masalah diatas, maka tujuan yang ingin dicapai adalah sebagai berikut :

1. Mewujudkan keefisienan dan keefektifan pengolahan sistem inventori/ gudang.
2. Meminimalisir terjadinya kesalahan dan ketidakakuratan dalam aktivitas pendataan dan pengolahan barang di gudang.
3. Memastikan setiap data perjalanan barang dari barang masuk, barang keluar, stok, perjalanan barang tidak meleset/ dipastikan akurat.
4. Memudahkan penjual/ supplier (olshop dan toko offline) untuk berkoneksi dengan gudang untuk menyimpan/ jualan/ taruh taruh barang/ tempat produksi mereka.

## 1.4 Ruang Lingkup

### 1. Pengembangan

Menurut Kamus Besar Bahasa Indonesia yang dimaksud pengembangan adalah pembangunan secara bertahap dan teratur yang menjurus ke sasaran yang dikehendaki. [2]

### 2. Sistem

Menurut Kamus Besar Bahasa Indonesia yang dimaksud dari sistem adalah perangkat unsur yang secara teratur saling berkaitan dan membentuk pola sehingga membentuk suatu totalitas. [3]

### 3. *Framework*

*Framework* merupakan sebuah software yang berguna untuk memudahkan para *programmer* dan *developer* untuk membuat sebuah aplikasi *web*. Pada *framework* terdapat berbagai fungsi diantaranya *plugin*, dan konsep untuk membentuk suatu sistem tertentu agar tersusun secara terstruktur dan rapih. [4]

Dengan menggunakan *framework* bukan berarti kita akan terbebas dengan pengkodean/ koding. Sebagai *programmer/ developer* yang menggunakan *framework*, haruslah menggunakan fungsi – fungsi dan *variable* yang ada di dalam sebuah *framework* yang digunakan.

### 4. Gudang

Dalam praktik operasional setiap perusahaan/ instansi cenderung memiliki suatu ketidakpastian akan *demand* atau permintaan, ini menjadi suatu hal mendorong timbulnya kebijakan dari suatu perusahaan tersebut untuk melakukan sistem persediaan (*inventory*) agar permintaan dapat diantisipasi dengan bijak. Dengan kebijakan tersebut, dapat mendorong perusahaan untuk menyediakan fasilitas gudang sebagai wadah untuk menyimpan barang *inventory*. [5]

## 1.5 Sistematika Penulisan

Berikut ini adalah sistem penulisan dari laporan Pengembangan Aplikasi Sistem Inventori Berbasis Web dengan Framework CI (code igniter).

1. Judul Besar dan Halaman Muka
2. Logo Poltekpos

3. Abstrak
4. Daftar Isi
5. Daftar Gambar
6. Daftar Tabel
7. Bab I Pendahuluan
  - a. Latar Belakang
  - b. Identifikasi Masalah
  - c. Tujuan
  - d. Ruang Lingkup
  - e. Sistematika Penulisan
8. Bab II State of the Art
  - a. Sistem Informasi
  - b. UML
  - c. Flowmap
  - d. Teori Umum
9. Bab III Analisis dan Perancangan Sistem
  - a. Analisis Sistem
  - b. Perancangan
10. Bab IV Implementasi dan Pengujian
  - a. Implementasi
  - b. Pembuatan Database
  - c. Tampilan Antar Muka Web
11. Bab V Kesimpulan dan Saran
  - a. Kesimpulan
  - b. Saran
12. Daftar Pustaka

## **BAB II**

### **STATE OF THE ART**

#### **2.1 Sistem Informasi**

Menurut Robert A. Leitch dan K. Roscoe Davis (2005). Sistem informasi adalah suatu sistem yang mengelola kerja suatu organisasi. Sistem informasi sangat diperlukan untuk mengelola kegiatan-kegiatan seperti transaksi harian, mendukung operasi mengelola kegiatan yang bersifat manajerial, kegiatan yang digunakan untuk menentukan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan. Menurut Sutabri (2005).

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian. Fungsi sistem informasi digunakan oleh organisasi untuk mengelola kegiatan yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk dapat menyediakan kepada pihak luar tertentu dengan laporan-laporan yang diperlukan. Berdasarkan beberapa pendapat diatas, sistem informasi dapat didefinisikan sebagai prosedur yang dilakukan oleh suatu organisasi untuk mencapai tujuan yang ingin dicapai organisasi dan untuk mengelola kegiatan-kegiatan yang bersifat manajerial.

Saat pembuatan aplikasi Pergudangan berbasis website, terdapat beberapa fitur yang akan dibangun. Sebelum membuat aplikasi ini, ada beberapa perancangan terlebih dahulu. Perancangan yang dimaksud adalah *Usecase diagram*, *class diagram*, *sequence diagram*, *collaboration diagram*, *activity diagram*, *statechart diagram*, *component diagram*, *deployment diagram*, *object diagram*, *struktur menu*, dan perancangan antarmuka/ *interface* terlebih dahulu. Aplikasi ini juga memakai *database*, bahasa SQL dan bahasa pemrograman PHP yang berkaitan dengan fitur yang akan dibangun. Bahasa pemrograman yang digunakan adalah *PHP Native*, *Javascript* dan *MySQL*. Proses pembuatan aplikasi ini juga akan mengalami *Debugging*.

#### **2.2 Unified Modelling Language (UML)**

UML (*Unified Modeling Language*) adalah bahasa pemodelan yang menggunakan konsep OOP (*Object Oriented Programming*). UML merupakan bahasa untuk menspesifikasi, memrepresentasi, memvisualisasi, membangun dan mendokumentasikan suatu hal yang

disebut “*artifacts*” (bagian dari informasi yang digunakan agar dihasilkan oleh proses *developing* perangkat lunak lainnya) dari sistem software (perangkat lunak), misalnya pada pemodelan bisnis dan pada sistem non-software lainnya. Artifacts dapat berupa deskripsi, perangkat lunak dan model [6].

UML (*Unified Modeling Language*) menyediakan banyak notasi yang membantu kegiatan pemodelan sistem dari berbagai sudut pandang. Tidak hanya digunakan dalam pemodelan perangkat lunak, namun UML membutuhkan kegiatan pemodelan hampir dalam semua bidang.

UML paling efektif digunakan pada beberapa domain ini, antara lain sebagai berikut :

1. Bidang Pengetahuan
2. Bidang Perdagangan
3. Bidang Telekomunikasi
4. Bidang Transportasi
5. Bidang Penerbangan
6. Bidang Pelayanan Elektronik
7. Bidang Pelayanan Berbasis Web Terdistribusi
8. Sistem Perbankan dan Perekonomian
9. Sistem Informasi Perusahaan

Ada juga yang menjadi bagian-bagian dalam UML, yaitu antara lain :

1. View : Digunakan untuk melihat suatu sistem/ aplikasi yang dimodelkan dari beberapa aspek yang berbeda. Jenis-jenis view dalam UML : Usecase view, Component view, Deployment view, Logical view, Concurrency view.
2. Usecase View : Merupakan deskripsi dari fungsionalitas sistem yang dilakukan sesuai yang diinginkan pada *external actors* dimana *actor* ini yang berinteraksi dengan sistem yang dapat berupa *user* atau sistem lainnya. *Usecase view* ini digambarkan dalam *Usecase Diagram* dan *Activity Diagrams*. View ini digunakan khususnya untuk pelanggan, *developer*, *designer* dan penguji sistem (*system tester*).
3. Logical View : Merupakan deskripsi dari fungsionalitas dari suatu sistem aplikasi, struktur statis (*object*, *class* dan *relationship*) dan kolaborasi dinamis yang terjadi pada saat *object* mengirim pesan ke *object* lain dalam suatu fungsi tertentu. *Logical View* ini digambarkan dalam *Class Diagram* untuk struktur statis dan dalam diagram *Statechart*, *Collaboration*,

*Sequence* dan *Activity* untuk model dinamisnya. Ini umumnya digunakan untuk *designer* dan *developer*.

4. *Component View* : Merupakan deskripsi dari implementasi dan ketergantungan modul dimana *Component View* adalah tipe lainnya dari *code module* yang diperlihatkan dengan struktur dan ketergantungannya juga alokasi sumber daya komponen dan informasi administratif lainnya. *View* yang satu ini digambarkan dalam *Component View* dan digunakan untuk *developer*.
5. *Concurrency View* : Jenis *view* ini deskripsi dari pembagian sistem ke dalam proses dan *processor*, dimana *view* ini digambarkan dalam diagram dinamis, yaitu seperti : *Activity*, *Sequence*, *Collaboration*, *Statechart*) dan pada diagram implementasi, yaitu seperti : *Deployment* dan *Component*). *Concurrency View* ini digunakan untuk *developer*, *integrator* (pengintegrasian), dan *system tester* (penguji sistem).
6. *Deployment View* : Merupakan deskripsi dari bentuk fisik dari sebuah sistem seperti perangkat (*nodes*) dan komputer, dan bagaimana hubungannya dengan yang lainnya. Jenis *view* ini digambarkan dalam diagram *Deployment*. *View* ini digunakan untuk *developer*, *integrator* (pengintegrasian), dan *system tester* (penguji sistem).

Berikut adalah diagram-diagram yang terdapat dalam UML (*Unified Modelling Language*).











#### A. Usecase Diagram

Usecase diagram adalah deskripsi dari fungsi yang disediakan oleh sistem dalam bentuk teks sebagai bentuk dokumentasi dari simbol-simbol usecase. Usecase digambarkan dari keadaan yang dilihat dari luar (lingkungan sistem yang dilihat user), yaitu oleh actor. Hal ini dapat juga dilakukan dalam *Activity Diagrams*.



Gambar 1 Simbol Usecase Diagram

Sumber : Mambaur Roziq [7]

SIMBOL	NAMA	KETERANGAN
	<i>Actor</i>	Menspesifikasikan himpunan peran yang pengguna mainkan ketika berinteraksi dengan <i>use case</i> .
	<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
	<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<i>Include</i>	Menspesifikasikan bahwa <i>use case</i> sumber secara eksplisit.
	<i>Extend</i>	Menspesifikasikan bahwa <i>use case</i> target memperluas perilaku dari <i>use case</i> sumber pada suatu titik yang diberikan.
	<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<i>System</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<i>Use Case</i>	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor.
	<i>Collaboration</i>	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> ).
	<i>Note</i>	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

## B. Class Diagram








*Class Diagram* menggambarkan struktur statis class yang ada pada suatu sistem dimana *Class Diagram* mempresentasikan sesuatu yang dikerjakan oleh sistem. *Class Diagram* dapat berhubungan dengan lainnya melalui beberapa cara, yaitu antara lain :

- *Dependent* : satu class tergantung atau menggunakan *class* lainnya
- *Package* : Grup untuk banyak *class* yang menjadi satu unit
- *Specialized* : satu *class* yang merupakan spesialisasi dari *class* yang lain
- *Associated* : Class yang terhubung satu dengan yang lainnya

## Gambar 2 Simbol Class Diagram

Sumber : Nur Islam Imanudin [8]

**SIMBOL CLASS DIAGRAM**






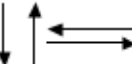
NO	GAMBAR	NAMA	KETERANGAN
1		<i>Generalization</i>	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		<i>Nary Association</i>	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		<i>Class</i>	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		<i>Collaboration</i>	<u>Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu actor</u>
5		<i>Realization</i>	Operasi yang benar-benar dilakukan oleh suatu objek.
6		<i>Dependency</i>	<u>Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri (<i>independent</i>) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri</u>
7		<i>Association</i>	Apa yang menghubungkan antara objek satu dengan objek lainnya

## C. Activity Diagram

*Activity Diagram* mendeskripsikan aktivitas yang dibentuk dalam suatu kegiatan (operasi) yang menggambarkan rangkaian aliran dari suatu aktivitas sehingga dapat digunakan juga untuk interaksi aktivitas yang lain seperti *Usecase Diagram*.

## Gambar 3 Simbol Activity Diagram

Sumber : Nur Islam Imanudin [9]


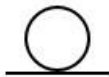
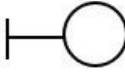



NO	GAMBAR	NAMA	KETERANGAN
1		Activity	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		Action	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		Initial Node	Bagaimana objek dibentuk atau diawali.
4		Activity Final Node	Bagaimana objek dibentuk dan diakhiri
5		Decision	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		Line Connector	Digunakan untuk menghubungkan satu simbol dengan simbol lainnya

#### D. Sequence diagram

Mendeskripsikan gambaran kolaborasi yang dinamis pada sejumlah *object* dimana berguna untuk menunjukkan interaksi antar *object* yang ada, juga rangkaian pesan yang dikirim antara *object-object* tersebut. Ini terjadi pada titik tertentu dalam suatu eksekusi sistem.

*Gambar 4 Simbol Sequence Diagram*

*Sumber : Ansori [10]*

NO	GAMBAR	NAMA	KETERANGAN
1		<i>Actor</i>	Menggambar orang yang sedang berinteraksi dengan sistem.
2		<i>Entity Class</i>	Menggambarkan hubungan yang akan dilakukan
3		<i>Boundary Class</i>	Menggambarkan sebuah gambaran dari form
4		<i>Control Class</i>	Menggambarkan penghubung antara boundary dengan tabel
5		<i>A focus of Control &amp; A Life Line</i>	Menggambarkan tempat mulai dan berakhirnya message
6		<i>A message</i>	Menggambarkan Pengiriman Pesan

#### E. Statechart Diagram

Mendeskripsikan gambaran seluruh kondisi (*state*) yang dimiliki oleh *object* dari suatu *class* dan keadaan yang membuat *state* itu sendiri berubah (kondisi *class* berubah oleh *state* yang berbeda). *Statechart* tidak digambarkan untuk semua *class*, tapi hanya yang mempunyai sejumlah *state* yang terdefinisi dengan baik. Operasi yang terjadi dapat berupa *object* yang mengirim pesan.

*Gambar 5 Simbol Statechart Diagram*

*Sumber : Ahmad Hasanudin [11]*

Simbol	Deskripsi
status awal / kondisi awal 	status awal alur sebuah objek, sebuah diagram status memiliki sebuah status awal
status 	status yang dialami objek selama hidupnya
status akhir / kondisi akhir 	kondisi akhir alur hidup objek, sebuah diagram status memiliki sebuah status akhir
transisi	garis transisi antar status pada daur hidup objek, transisi biasanya diberi nama pesan yang ada pada diagram

Simbol	Deskripsi
nama transisi 	sekuen sehingga pesan pada diagram sekuen menjadi transisi bukan sebagai status, status merupakan kondisi yang dialami objek, bukan merupakan pesan ( <i>message</i> ), transisi juga bisa memutar pada sebuah status
	Transisi internal melingkar / ke status sendiri

## F. Component Diagram

Mendeskripsikan gambaran struktur fisik kode dari suatu komponen dimana komponen ini dapat berupa *biner component*, *executable component* dan *source code*. Komponen berisi informasi tentang *class* yang diimplementasikan (biasa disebut *logic class*) sehingga membuat pemetaan dari *logical view* ke *component view*.

*Gambar 6 Simbol Component Diagram*

*Sumber : Efendi [12]*

Simbol	Deskripsi
<b>Package</b> 	package merupakan sebuah bungkusan dari satu atau lebih komponen
<b>Komponen</b> 	Komponen sistem
<b>Kebergantungan / dependency</b> 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
<b>Antarmuka / interface</b>	sama dengan konsep <i>interface</i> pada pemrograman berorientasi objek, yaitu sebagai antarmuka komponen

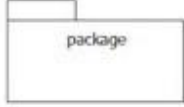
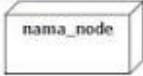


Simbol	Deskripsi
	agar tidak mengakses langsung komponen
<b>Link</b> 	relasi antar komponen
	Dokumen, dapat berupa file, library

## G. Deployment Diagram

Mendeskripsikan gambaran arsitektur fisik dari *hardware* dan *software* dari suatu sistem yang menunjukkan hubungan perangkat computer dengan perangkat (*nodes*) dan jenis hubungan satu dengan lainnya. *Executable component* dan *object* yang ada didalam *nodes*, dialokasikan untuk memperlihatkan *unit software* yang dieksekusi oleh suatu *node* dan ketergantungan suatu komponen.

*Gambar 7 Simbol Deployment Diagram*

*Sumber : Ahmad Hasanudin [13]*



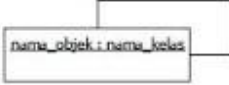


Simbol	Deskripsi
<b>Package</b> 	package merupakan sebuah bungkus dari satu atau lebih <i>node</i>
<b>Node</b> 	biasanya mengacu pada perangkat keras ( <i>hardware</i> ), perangkat lunak yang tidak dibuat sendiri ( <i>software</i> ), jika di dalam <i>node</i> disertakan komponen untuk mengkonsistenkan rancangan maka komponen yang diikutsertakan harus sesuai dengan komponen yang telah didefinisikan sebelumnya pada diagram komponen
<b>Kebergantungan / dependency</b> 	Kebergantungan antar <i>node</i> , arah panah mengarah pada <i>node</i> yang dipakai
<b>Link</b> 	relasi antar <i>node</i>

## H. Collaboration Diagram

Collaboration Diagram (diagram kolaborasi) mendeskripsikan gambaran operasi atau metode yang dijalankan antara suatu objek ke objek lainnya yang mencakup keseluruhan. Penomoran metode/ operasi dapat dilakukan berdasarkan urutan dijalanannya metode tersebut diantara objek satu dengan yang lainnya atau pada objek itu sendiri. Dalam pemodelan Collaboration Diagram, suatu objek mengirim pesan setelah pesan-pesan lainnya terkirim, oleh sebab itu, objek harus menyinkronkan pesan dengan serangkaian pesan yang ada.

*Gambar 8 Simbol Collaboration Diagram*

*Sumber : Ahmad Hasanudin [14]*

Simbol	Deskripsi
<b>Objek</b> 	objek yang melakukan interaksi pesan
<b>Link</b> 	relasi antar objek yang menghubungkan objek satu dengan lainnya atau dengan dirinya sendiri 
<b>Arah pesan / stimulus</b> 	arah pesan yang terjadi, jika pada suatu <i>link</i> ada dua arah pesan yang berbeda maka arah juga digambarkan dua arah pada dua sisi <i>link</i> 








## 2.3 Flowmap

Flowmap atau yang sering disebut sebagai proses bisnis adalah campuran antara peta (map) dan flowchart yang menggambarkan pergerakan suatu benda dari tempat/ lokasi satu ke lokasi lainnya yang dimana sebagai utuh disebut proses dari sebuah operasi. Flowmap banyak menolong programmer dan analysts dalam hal memecahkan masalah kedalam segmen yang lebih kecil dan rinci yang membuat mereka mengetahui penyebab masalah tersebut dari hal yang paling dasar. Flowmap juga membantu para programmer dan analysts dalam hal menganalisis jalan pintas (alternatif) dalam suatu kegiatan pengoperasian [15].

*Gambar 9 Simbol Flowmap*

*Sumber : elib.unikom.ac.id [16]*



Symbol	Nama Simbol	Keterangan
	Dokumen	Digunakan untuk menggambarkan semua jenis dokumen yang merupakan formulir yang digunakan untuk mengentry data keluarga.
	Proses	Merupakan kegiatan proses dari operasi program computer
	Proses manual	Merupakan proses manual pada flowmap
	File Harddisk	Merupakan media penyimpanan dari proses entry data dan proses komputerisasi
	Offline Storage	Merupakan tempat penyimpanan data berupa arsip
	Garis Alir	Merupakan arus data
	Keyboard	Merupakan proses penyimpanan menggunakan keyboard

## 2.4 Teori Umum

Berikut adalah teori teori umum yang dipakai untuk membuat perancangan *Aplikasi Sistem Inventori Berbasis Web* ini.

### a. Pergudangan

Gudang adalah fasilitas khusus yang bersifat tetap, yang dirancang untuk mencapai target tingkat pelayanan dengan total biaya yang paling rendah.

*Gambar 10 Ilustrasi Pergudangan*

*Sumber : CKB Integrated Logistics Services [17]*

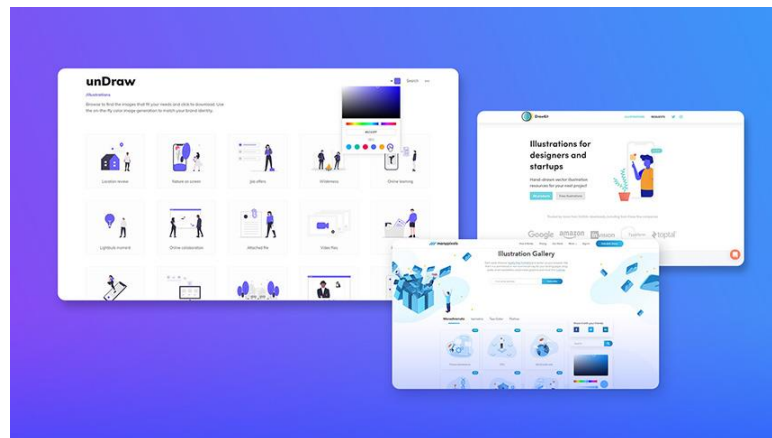


Gudang dibutuhkan dalam proses koordinasi penyaluran barang, yang muncul sebagai akibat kurang seimbangnya proses penawaran dan permintaan. Kurang seimbangnya antara proses permintaan dan penawaran mendorong munculnya persediaan (*inventory*), persediaan membutuhkan ruang sebagai tempat penyimpanan sementara yang disebut sebagai gudang [5].

## **b. Website**

*Gambar 11 Ilustrasi Website*

*Sumber : Alvian Indonesia [18]*



*Website* merupakan kumpulan halaman-halaman yang terdapat pada domain internet yang dibuat dengan tujuan yang jelas dimana ini digunakan untuk menampilkan informasi teks, gambar diam atau gerak, animasi, suara, dan atau gabungan dari semuanya baik yang

bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait, yang masing masing dihubungkan dengan jaringan jaringan halaman [19].

*website* (lebih dikenal dengan sebutan situs) adalah sejumlah halaman *web* yang memiliki topik saling terkait, terkadang disertai pula dengan berkas gambar, video atau berbagai jenis berkas lainnya. *Website* dapat diakses secara menyeluruh melalui halaman *dashboard* (*homepage*) dengan menggunakan URL website pada browser.

### c. SQL

SQL (Structured Query Language) merupakan suatu sistem manajemen database relasional (RDBMS) yang dirancang untuk aplikasi/ sistem yang memiliki arsitektur client/ server. Terdapat tiga (3) jenis dari perintah SQL yang dapat digunakan, antara lain DDL (Data Definition Language), DML (Data Manipulation Language) dan DCL (Data Control Language) [20].

SQL muncul berawal dari sebuah artikel yang membahas tentang ide pembuatan basis data relational pada tahun 1970 oleh seorang peneliti IBM bernama EF Codd. Artikel ini juga membahas tentang kemungkinan pembuatan bahasa standar untuk mengakses data dalam basis data tersebut. Bahasa tersebut diberi nama SEQUEL (*Structured English Query Language*) yang kemudian berganti nama menjadi SQL karena bermasalah dengan hukum.

*Gambar 12 Ilustrasi SQL*

*Sumber : [www.blogon.id](http://www.blogon.id) [21]*



Pada tahun 1970-an, sebuah perusahaan yang bernama *Oracle* membuat *server* basis data yang bernama sesuai dengan nama perusahaannya. Seiring naiknya kepopuleran *Oracle*, maka SQL juga ikut populer sehingga menjadi standar bahasa dalam manajemen suatu *database*.

#### d. Basis Data

Basis Data pada dasarnya merupakan sistem yang terkomputerisasi yang tujuan utamanya adalah *me-maintaining* (memelihara) suatu informasi bahkan lebih dimana informasi tersebut disimpan dan tersedia saat dibutuhkan [22]. sistem dari database adalah suatu komponen yang sangat penting dari sistem informasi pada suatu perusahaan. Aplikasi daur hidup *database* adalah pengumpulan pewarisan dengan daur hidup dari sistem informasi. Contohnya adalah dari masalah yang dihadapi selama proses perancangan *database* yang mengharuskan penambahan koleksi dan melakukan analisis kebutuhan. Pada proyek tiga (3) ini, kami menggunakan mysql sebagai manajemen basis data (DBMS).

#### e. JavaScript

*JavaScript* adalah bahasa *script* yang dikembangkan oleh perusahaan *Netscape* untuk membuat dokumen yang dinamis. *JavaScript* merupakan bahasa *script* sederhana yang mempunyai kemiripan dengan bahasa pemrograman C. *JavaScript* juga dikenal sebagai sebuah kode pemrograman berorientasi objek (*Object Oriented Programming*) yang disingkat OOP. Javascript yang digunakan pada proyek tiga (3) kami adalah chart.js.

*Gambar 13 Ilustrasi Kode Javascript*

*Sumber : Ahmad Nur Ubaidah [23]*



*JavaScript* memiliki keistimewaan untuk ditambahkan pada kode HTML dan membuat dokumen menjadi lebih interaktif.

#### f. XAMPP

XAMPP adalah perangkat lunak bebas, yang mendukung banyak sistem operasi, merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP Server*, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.

*Gambar 14 Logo XAMPP*

*Sumber : Warung Komputer [24]*



Nama XAMPP merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan *web server* yang mudah digunakan yang dapat melayani tampilan halaman *web* yang dinamis. XAMPP digunakan saat ingin menampilkan tampilan web dari source code mentahan (raw) yang dibuat. Disini kami menggunakan server MySQL dan Apache untuk koneksi ke phpMyAdmin dan Aplikasi yang dibuat melalui browser perangkat.

#### g. Cascading Style Sheet (CSS)

*Gambar 15 Ilustrasi CSS Tata letak Horizontal dan Vertical*

*Sumber : Tulus [25]*



CSS merupakan suatu teknik design web yang memadukan antara html dengan *cascading style sheet*. CSS menjadikan halaman *web* lebih interaktif, menarik, dan elegan hanya dengan menambahkan beberapa baris kode di dalamnya

#### h. PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman yang berbentuk *scripting*. Pada proyek tiga (3) ini, kami menggunakan PHP versi 5 (PHP5).

*Gambar 16 Logo PHP*

*Sumber : php.net [26]*



Sistem kerja dari program ini adalah sebagai interpreter bukan sebagai *compiler*. Perbedaan antara keduanya adalah :

## 1. Bahasa Compiler

adalah bahasa yang akan mengubah *script script* program kedalam *source code*, selanjutnya dari bentuk *source code* akan diubah menjadi bentuk *object code*, kemudian dari bentuk *object code* akan berubah menjadi sebuah program yang siap dijalankan tanpa adanya program bantu pembuatnya, sehingga hasil dari bahasa pemrograman compiler akan membentuk program dengan ekstensi .exe yang dapat dieksekusi tanpa membutuhkan bantuan program pembuatnya.

## 2. Bahasa Interpreter

Dalam bahasa interpreter, *script* mentahnya tidak harus diubah kedalam *script code*. Sehingga pada saat dijalankan secara langsung akan menjalankan kode dasar tanpa melalui proses pengubahan kedalam bentuk *source code* (Nugroho, 2009)

### i. Sistem

Sistem merupakan gabungan objek atau unsur yang saling teratur sehingga dapat membentuk suatu rangkaian komponen yang kompleks yang dapat berhubungan satu sama lain.

*Gambar 17 Ilustrasi Penggambaran Sebuah Sistem*

*Sumber : Guntoro [27]*



Suatu sistem itu dikatakan sistem yang baik memiliki karakteristik yaitu :

### 1. Komponen



Suatu sistem terdiri dari sejumlah komponen-komponen yang saling berinteraksi, yang artinya saling bekerja sama membentuk satu kesatuan. Komponen sistem terdiri dari komponen yang berupa subsistem atau bagian-bagian dari sistem.

2. Batasan sistem

Batasan sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem yang lain atau dengan lingkungan luarnya.

3. Lingkungan luar sistem (*environment*)

Lingkungan luar sistem adalah diluar batas dari sistem yang mempengaruhi operasi sistem. Lingkungan luar yang bersifat menguntungkan harus tetap dijaga sedangkan lingkungan luar yang bersifat merugikan selain dijaga tapi harus dikendalikan juga, jika tidak akan mengganggu kinerja dari sistem tersebut.

4. Penghubung Sistem(*interface*)

Penghubung sistem merupakan media penghubung antara satu subsistem dengan subsistem lainnya, atau antara subsistem dengan penggunanya. Melalui penghubung ini memungkinkan sumber-sumber daya mengalir dari subsistem ke subsistem lain. Keluaran (*output*) dari subsistem akan menjadi masukan (*input*) untuk subsistem lain melalui penghubung.

5. Masukan sistem(*input*)

Masukkan adalah perintah yang diatur dalam suatu sistem yang perintahnya dapat berupa seperti *maintenance input* dan *signal input*. *Maintenance input* adalah energi yang dimasukkan agar sistem dapat beroperasi. *Signal input* adalah energi yang diproses untuk didapatkan keluaran.

6. Keluaran sistem(*output*)

Keluaran sistem adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan

## **i. Framework**

*Framework* merupakan sebuah software yang berguna untuk memudahkan para *programmer* dan *developer* untuk membuat sebuah aplikasi *web* [4]. Pada *framework* terdapat berbagai fungsi diantaranya *plugin*, dan konsep untuk membentuk suatu sistem tertentu agar tersusun secara terstruktur dan rapih.



Dengan menggunakan framework bukan berarti kita akan terbebas dengan pengkodean/ koding. Sebagai *programmer/ developer* yang menggunakan *framework*, haruslah menggunakan fungsi – fungsi dan variable yang ada di dalam sebuah *framework* yang digunakan.

Dalam proyek tiga (3) ini, kami menggunakan basis *framework* CI (*Codeigniter*) versi 3 (*Codeigniter* 3) dimana ini adalah salah satu poin yang kelompok kami kembangkan pada sistem dari proyek dua (2) sebelumnya, yaitu selain sebagai standar pengkodean, juga untuk mempercepat dan memudahkan dalam membuat *website* sistem inventori ini, dimana disini menghasilkan struktur program yang rapi dari pengkodean dan pembagian serta pemanggilan fungsinya lebih efisien.

## **BAB III**

### **ANALISIS DAN PERANCANGAN SISTEM**

#### **3.1 Analisis Sistem**

Pada bab ini akan membahas dan menganalisis tentang aplikasi sistem inventori berbasis web. Setelah menganalisis, maka data-data yang dikumpulkan dapat dibuat dalam bentuk yang lebih rinci.

##### **3.1.1 Analisis Sistem yang Sedang Berjalan**

Pada bagian ini akan membahas tentang analisis sistem yang sedang berjalan pada pengembangan aplikasi sistem inventori berbasis web dengan framework CI (Codeigniter). Menganalisis sistem yang sedang berjalan pada aplikasi sistem inventori ini merupakan langkah tujuan utama sistem Inventori untuk mengetahui bagaimana proses bisnis yang terjadi pada aplikasi sistem inventori tersebut.

##### **3.1.1.1 Analisis Prosedur/ Flow Map Berjalan**

Pengelolaan data pada aplikasi sistem inventori yang dirancang ini dilakukan dengan menggunakan sistem yang dengan tujuan dapat mempermudah user (admin dan gudang) dapat dengan mudah melakukan kegiatan pengelolaan data barang dalam gudang , dari mulai transaksi barang masuk dan barang keluar, data barang, data supplier. Pada proyek dua (2) sebelumnya, sistem aplikasi yang dibangun belum mumpuni hingga pada prosres pengembangan pada proyek tiga (3) ini. sekarang terdapat beberapa fungsi dan fitur yang ditambahkan serta dengan pembagian user sistem. Berikut adalah actor/ orang-orang yang terlibat di dalam proses sistem inventori barang .

##### **1. User Admin**

Yaitu aktor (user) aktif yang bertugas untuk melakukan pendataan data supplier, transaksi barang masuk, transaksi barang keluar, data barang dan stok, manajemen user dan mencetak laporan transaksi barang (masuk dan keluar).

##### **2. User Gudang**

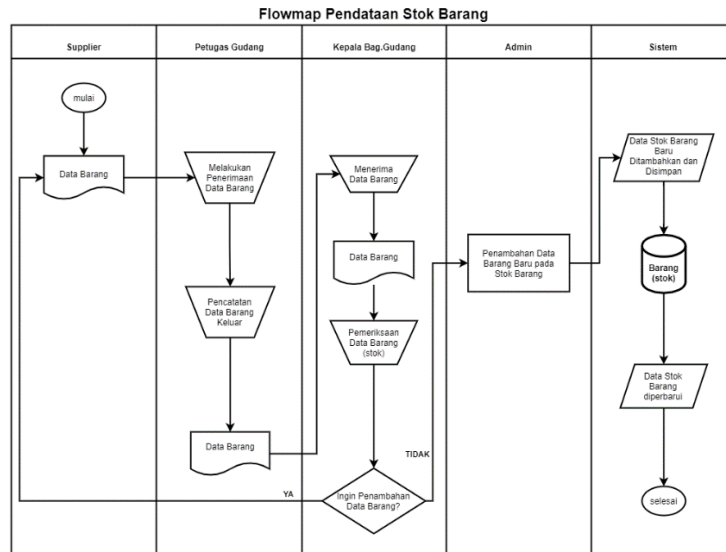
Yaitu actor (user) aktif yang bertugas untuk menerima data barang dari supplier, menerima data barang masuk, data barang keluar dan memberikan cetak laporan transaksi barang (masuk dan keluar) kepada supplier.

### 3. Supplier

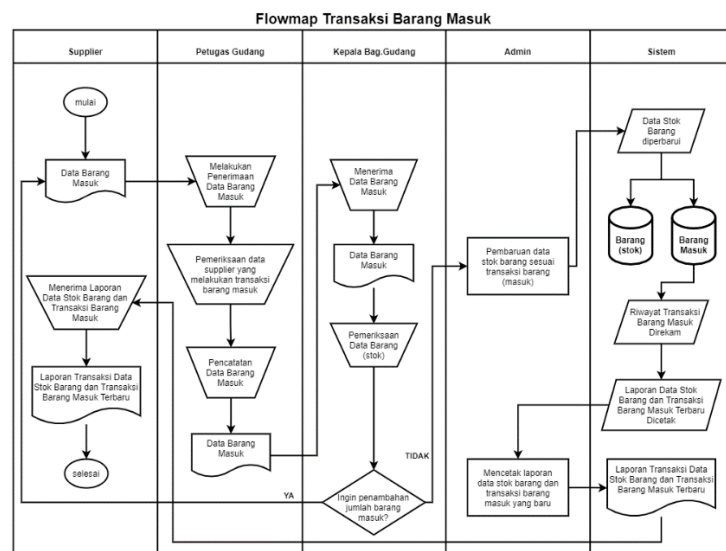
Yaitu actor yang tidak aktif (pasif) yang bertugas untuk memberikan/ mendaftarkan data supplier, memberi data barang yang akan di stok, data barang masuk dan data barang keluar.

Berikut adalah gambar Flowmap dari aplikasi sistem inventori.

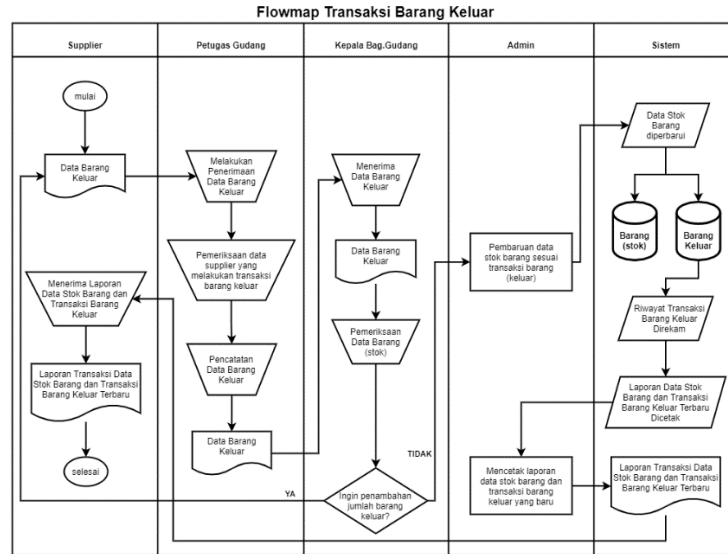
*Gambar 18 Flowmap Pendataan Stok Barang*



*Gambar 19 Flowmap Transaksi Barang Masuk*



*Gambar 20 Flowmap Transaksi Barang Keluar*



### 3.1.1.2 Analisis Dokumen yang Digunakan

Analisis dokumen merupakan kegiatan menganalisis seluruh dokumen dasar yang digunakan dan mengalir pada sistem informasi yang sedang berjalan pada Perancangan Aplikasi Sistem Inventori Berbasis Web untuk menganalisis aplikasi ini. Berikut ini adalah rincian dari tabel dokumen yang digunakan untuk Aplikasi Sistem Inventori Berbasis Web yang sedang sedang berjalan.

Analisa Dokumen sistem yang sedang berjalan dapat dilihat pada tabel berikut :

*Tabel 1 Analisis Dokumen Sistem yang Sedang Berjalan*

No	Nama Dokumen	Uraian
1	Data Supplier	Keterangan : Data Supplier Sumber : User Admin, User Gudang/ Staff Gudang
2	Data Transaksi Barang Masuk	Keterangan : Data Transaksi Barang Masuk Sumber : User Admin, User Gudang/ Staff Gudang
3.	Data Transaksi Barang Keluar	Keterangan : Data Transaksi Barang Keluar

		Sumber : User Admin, User Gudang/ Staff Gudang
4.	Data Stok Barang	Keterangan : Data Stok Barang Sumber : User Admin, User Gudang/ Staff Gudang
5.	Data User Admin	Keterangan : Data User Admin dalam Sistem Aplikasi Sumber : User Admin
6.	Data User Gudang	Keterangan : Data User Gudang dalam Sistem Aplikasi Sumber : User Admin
7.	Satuan Barang	Keterangan : Data Satuan Barang Sumber : User Admin
8.	Jenis Barang	Keterangan : Data Jenis Barang Sumber : User Admin
9.	Status Barang	Keterangan : Data Status Barang Sumber : User Admin
10.	Merek Barang	Keterangan : Data Merek Barang Sumber : User Admin
11.	Harga Barang	Keterangan : Data Harga Barang Sumber : User Admin
12.	Cetak Laporan Data Barang	Deskripsi : Laporan Data Transaksi Barang Masuk, Transaksi Barang Keluar, Barang (stok). Sumber : User Admin Diberikan Kepada : Supplier/ Pemilik Perusahaan

### 3.1.1.3 Analisis User

Aplikasi Sistem Inventori Berbasis Web dengan Framework Codeigniter ini digunakan dalam kegiatan pergudangan yaitu aktivitas pergudangan seperti barang masuk dan barang keluar. Aplikasi ini mempunyai beberapa aktor/ user yaitu antara lain melibatkan User *Admin* (yang mengelola aplikasi sistem inventori tersebut), *User Gudang* (Yaitu pekerja yang berpopulasi dalam Gudang dan bertanggung jawab dalam aktivitas masuk keluar barang) dan *Supplier* (yang memakai jasa inventori dan mengirim persediaan untuk stok barang mereka).

### 3.1.2 Analisis Sistem yang akan Dibangun

Sistem aplikasi yang akan dibangun dan pada proyek lanjutan dari proyek 2(dua) sebelumnya, yakni proyek 3 (tiga) saat ini adalah pengembangan aplikasi sistem inventori berbasis web dengan framework CI (code igniter). Beberapa fitur dan fungsi telah ditambahkan dan dikembangkan yakni yang pada proyek 2 (dua) sebelumnya sistem berbasis native source code, pada proyek 3 (tiga) ini memakai framework CI atau Code Igniter dengan OOP (Object Oriented Programming) dengan membuat setiap fungsi-fungsi yang ada pada sistem menjadi object yang dapat dipanggil kapanpun saat dibutuhkan pada file-file aplikasi sehingga ini dapat memudahkan Kami (Penulis) sebagai developer/ programmer dan analisis dari pengembangan aplikasi sistem inventori pada proyek 3 (tiga) dalam proses code (koding).

#### 3.1.2.1 Analisis Kebutuhan Aplikasi

##### A. Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan yang berhubungan dengan sistem yang berjalan. Pada bagian ini akan menjelaskan tentang apa saja fungsi-fungsi yang akan mendukung berjalannya sistem dari *Perancangan Aplikasi Sistem Inventori Berbasis Web*. Adapun kebutuhan fungsional yang akan dibuat adalah antara lain:

1. Fungsi CRUD data barang
2. Fungsi CRUD data stok barang
3. Fungsi CRUD transaksi barang masuk
4. Fungsi CRUD transaksi barang keluar

5. Fungsi *Trigger* untuk transaksi jumlah barang terhadap stok
6. Fungsi cetak laporan transaksi barang dan laporan data barang
7. Fungsi role user (*user admin* dan *user gudang*)
8. Fungsi aktif dan nonaktif *user*

#### B. Analisis Kebutuhan Nonfungsional

Analisis kebutuhan non-fungsional disini adalah analisa kebutuhan perangkat lunak yang dilakukan yang bertujuan untuk mengetahui spesifikasi kebutuhan yang dibutuhkan untuk sistem bisa dijalankan. Spesifikasi kebutuhan diantaranya adalah analisis perangkat keras/ *hardware* dan analisis perangkat lunak/ *software*.

#### 3.1.2.2 Analisis Kebutuhan Perangkat Lunak dan Perangkat Keras

Berikut adalah spesifikasi kebutuhan perangkat keras yang digunakan pada sistem.  
Laptop (Asus X540y dan Dell Latitude e6330)

*Tabel 2 Kebutuhan Perangkat Keras*

1	<i>Harddisk</i>	500gb, 500gb	Media penyimpanan data dari aplikasi yang dibuat.
2.	<i>Memory/ RAM</i>	4 gb ddr3, 4 gb ddr3	<i>Memory System</i>
3.	<i>Processor</i>	<ul style="list-style-type: none"> <li>- Amd Quad Core 7110 E2 1.8 Ghz.</li> <li>- Intel Core</li> </ul>	Inti/ otak dari laptop sebagai kunci dari perangkat keras dan perangkat lunak yang digunakan untuk membangun aplikasi

		I 7 Gen. 6 2,9 Ghz.	
--	--	------------------------------	--

Berikut adalah spesifikasi kebutuhan perangkat lunak yang digunakan pada sistem.

*Tabel 3 Kebutuhan Perangkat Lunak*

No	<i>Tools</i>	Fungsi	Keterangan
1.	Windows 10	Sistem Operasi	-
2.	Sublime Text	<i>Text Editor</i>	-
3.	PHP5	Bahasa Pemrograman	-
4.	Mysql	<i>dbms</i>	-
5.	PHPMyAdmin	<i>Web server dbms</i>	-
6.	HTML5	Bahasa Pemrograman	-
7.	Javascript	Bahasa Pemrograman	-
8.	Google Chrome	<i>Web browser</i>	-

### 3.2 Perancangan

Perancangan sistem adalah tahap selanjutnya setelah dilakukannya tahap kegiatan analisis sistem dan setelah mempunyai gambaran jelas akan hal-hal yang dilakukan/ dikerjakan pada tahap analisis sistem. Setelah semua itu telah dilakukan, maka dilanjutkan dengan memikirkan tentang bagaimana merancang sistem tersebut.

Perancangan sistem adalah tahap dimana sangat dibutuhkan skill atau keahlian para programmer dan developer dalam penguasaan setiap elemen pada komputer (baik software dan hardware) yang akan menggunakan sistem tersebut dan dalam hal merancang, mendesain dan mengembangkan suatu sistem.



Adapun tujuan yang hendak dicapai dari tahap perancangan system mempunyai maksud atau tujuan utama, yaitu sebagai berikut:

1. Demi memenuhi kebutuhan *user* dalam hal pemakaian sistem
2. Untuk memberikan gambaran yang jelas dan menghasilkan rancangan bangun yang lengkap kepada pemograman komputer dan para ahli teknik lainnya yang terlibat dalam pengembangan atau pembuatan sistem.

Perancangan sistem dapat dibuat dalam bentuk diagram UML (*Unified Modeling Language*). UML memiliki bermacam bentuk diagram, salah satunya adalah *usecase* diagram dimana diagram itu adalah inti dari diagram UML. Selain itu terdapat beberapa jenis diagram lainnya di diagram UML ini, diantaranya adalah

1. *Usecase Diagram*
2. *Class diagram*
3. *Activity diagram*
4. *Sequence diagram*
5. *Statechart diagram*
6. *Component Diagram*
7. *Deployment diagram*
8. *Collaboration diagram*

### **3.2.1 Usecase Diagram**

*Usecase diagram* adalah sebuah teknik yang digunakan dalam mengembangkan suatu *software* atau sistem informasi untuk menangkap kebutuhan fungsional dari sistem tersebut. Disini *usecase* menjelaskan interaksi yang terjadi antara aktor ke suatu proses yang disebut juga *usecase*. *Usecase* direpresentasikan dengan urutan langkah-langkah yang sederhana.

Berikut adalah gambar *usecase diagram* yang telah dibuat.

Gambar 21 Usecase Diagram Aplikasi Sistem Inventori



Definisi Usecase :

#### 1. Definisi Aktor

Tabel 4 Definisi Aktor

No.	Aktor	Keterangan
-----	-------	------------

1.	User <i>Admin</i>	Aktor yang bisa mengelola data yang ada pada stok barang, kelola barang masuk dan kelola barang keluar, manajemen user (admin dan gudang), kelola supplier, dan kelola laporan transaksi barang.
2.	User Gudang	Aktor yang bisa mengelola data yang ada pada stok barang, kelola barang masuk dan kelola barang keluar, kelola supplier, dan kelola laporan transaksi barang.

## 2. Penjelasan Usecase

*Tabel 5 Penjelasan Usecase*

No.	Usecase	Keterangan
1.	<i>Login</i>	Adalah proses melakukan <i>login</i> untuk user <i>admin</i> dan user gudang
2.	Kelola data barang (stok)	Adalah proses pengelolaan termasuk pendataan barang yang ada pada menu data barang (stok)
3.	kelola barang masuk	Adalah proses pengelolaan data transaksi barang masuk

4.	kelola barang keluar	Adalah proses pengelolaan data transaksi barang masuk
5.	Manajemen <i>User</i>	Adalah proses mengelola atau memanajemen data user ( <i>user admin</i> dan <i>user gudang</i> .) yang ada pada sistem.
6.	Kelola <i>Supplier</i>	Adalah proses pendataan dan pengelolaan data <i>supplier</i> .
7.	Kelola Jenis	Adalah proses pendataan dan pengelolaan data jenis terhadap data barang (stok).
8.	Kelola Satuan	Adalah proses pendataan dan pengelolaan data satuan terhadap data barang (stok).
9.	Kelola Harga	Adalah proses pendataan dan pengelolaan data harga terhadap data barang (stok).
10.	Kelola Merek	Adalah proses pendataan dan pengelolaan data merek terhadap data barang (stok).
11.	Kelola Status	Adalah proses pendataan dan pengelolaan data status terhadap data barang (stok).

12.	Kelola Laporan	Adalah proses pencetakan laporan riwayat transaksi barang terhadap barang masuk dan barang keluar.
13.	<i>Log out</i>	Adalah proses melakukan <i>logout</i> (keluar) untuk user <i>admin</i> dan user gudang

### 3. Skenario *Usecase*

Skenario *usecase* adalah urutan kejadian/ proses dalam suatu aplikasi/ sistem yang sedang berjalan. Skenario usecase menjelaskan tentang langkah-langkah dalam sistem aplikasi yang berjalan, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan sistem terhadap aktor. Berikut adalah penjelasan dari skenario usecase sistem pada pengembangan aplikasi sistem inventori berbasis web dengan *framework* CI (*Codeigniter*).

#### 1. Skenario Usecase Login

*Tabel 6 Skenario Usecase Login*

Aksi Aktor	Aksi Sistem
Skenario Normal	
1. Memasukkan <i>username</i> dan <i>password</i>	Memeriksa validasi data <i>username</i> dan <i>password</i> ke tabel admin
2. Login	Masuk ke aplikasi inventori barang
Skenario Alternatif	
1. Memasukkan <i>username</i> dan <i>password</i>	Memeriksa validasi data <i>username</i> dan <i>password</i> ke tabel admin
	<i>Login</i> gagal, menampilkan pesan <i>alert username</i> atau <i>password</i> salah
Memasukkan <i>username</i> dan <i>password</i> yang benar	Masuk ke aplikasi sistem inventori barang

## 2. Skenario Usecase Kelola Barang (Stok)

*Tabel 7 Skenario Usecase Stok Barang*

Aksi Aktor	Reaksi Sistem
1. Lihat persediaan stok barang	Menampilkan data yang ada pada stok barang
2. Menambahkan data stok barang	Proses penambahan data stok barang
3. Mengubah data stok barang	Proses pengubahan data stok barang
4. Menghapus data stok barang	Proses menghapus data stok barang
5. Mencari data stok barang	Menampilkan data stok barang yang dicari
6. Kelola data jenis	Menambahkan, mengubah, mencari, menghapus data jenis terhadap data barang
7. Kelola data satuan	Menambahkan, mengubah, mencari, menghapus data satuan terhadap data barang
8. Kelola data harga	Menambahkan, mengubah, mencari, menghapus data harga terhadap data barang
9. Kelola data merek	Menambahkan, mengubah, mencari, menghapus data merek terhadap data barang
10. Kelola data status	Menambahkan, mengubah, mencari, menghapus data status terhadap data barang

## 3. Skenario Usecase Kelola Barang Masuk

*Tabel 8 Skenario Usecase Kelola Barang Masuk*

Aksi Aktor	Reaksi Sistem
------------	---------------

1. Lihat halaman transaksi barang masuk	Menampilkan daftar data transaksi barang keluar dan fungsi fungsi untuk mengelola barang masuk
2. Menambahkan data transaksi barang masuk	Proses penambahan data transaksi barang
3. Mengubah data transaksi barang masuk	Proses pengubahan data barang
4. Menghapus data transaksi barang masuk	Proses menghapus data barang
5. Mencari data riwayat transaksi barang keluar	Menampilkan data riwayat transaksi barang yang dicari

#### 4. Skenario Usecase Kelola Barang Keluar

*Tabel 9 Skenario Usecase Kelola Barang Keluar*

Aksi Aktor	Reaksi Sistem
1. Lihat halaman transaksi barang keluar	Menampilkan daftar data transaksi barang keluar dan fungsi fungsi untuk mengelola barang keluar
2. Menambahkan data transaksi barang keluar	Proses penambahan data transaksi barang
3. Mengubah data transaksi barang keluar	Proses pengubahan data barang
4. Menghapus data transaksi barang keluar	Proses menghapus data barang
5. Mencari data riwayat transaksi barang keluar	Menampilkan data riwayat transaksi barang yang dicari

#### 5. Skenario Usecase Kelola Supplier

*Tabel 10 Skenario Usecase Kelola Supplier*

Aksi Aktor	Reaksi Sistem
------------	---------------

1. Lihat halaman data <i>supplier</i>	Menampilkan daftar data <i>supplier</i> dan fungsi fungsi untuk mengelola data <i>supplier</i>
2. Menambahkan data <i>supplier</i>	Proses penambahan data <i>supplier</i>
3. Mengubah data <i>supplier</i>	Proses pengubahan data <i>supplier</i>
6. Menghapus data <i>supplier</i>	Proses menghapus data <i>supplier</i>
7. Mencari data <i>supplier</i>	Menampilkan data <i>supplier</i> yang dibutuhkan

#### 6. Skenario Usecase Manajemen User

*Tabel 11 Skenario Usecase Kelola User*

Aksi Aktor	Reaksi Sistem
1. Lihat halaman <i>User Management</i>	Menampilkan daftar data <i>user</i> dan fungsi fungsi untuk mengelola data <i>user</i>
2. Menambahkan data <i>user</i>	Proses penambahan data <i>user</i>
3. Mengubah data <i>user</i>	Proses pengubahan data <i>user</i>
4. Menghapus data <i>user</i>	Proses menghapus data <i>user</i>
5. Mencari data <i>user</i>	Menampilkan data <i>user</i> yang dibutuhkan

#### 7. Skenario Usecase Kelola Laporan

*Tabel 12 Skenario Usecase Kelola Laporan*

Aksi Aktor	Reaksi Sistem
1. Lihat halaman Cetak Laporan Transaksi	Menampilkan <i>form</i> laporan yang akan dicetak dengan atribut laporan transaksi barang masuk dan barang keluar, pengaturan tanggal riwayat transaksi, lalu ada tombol cetak

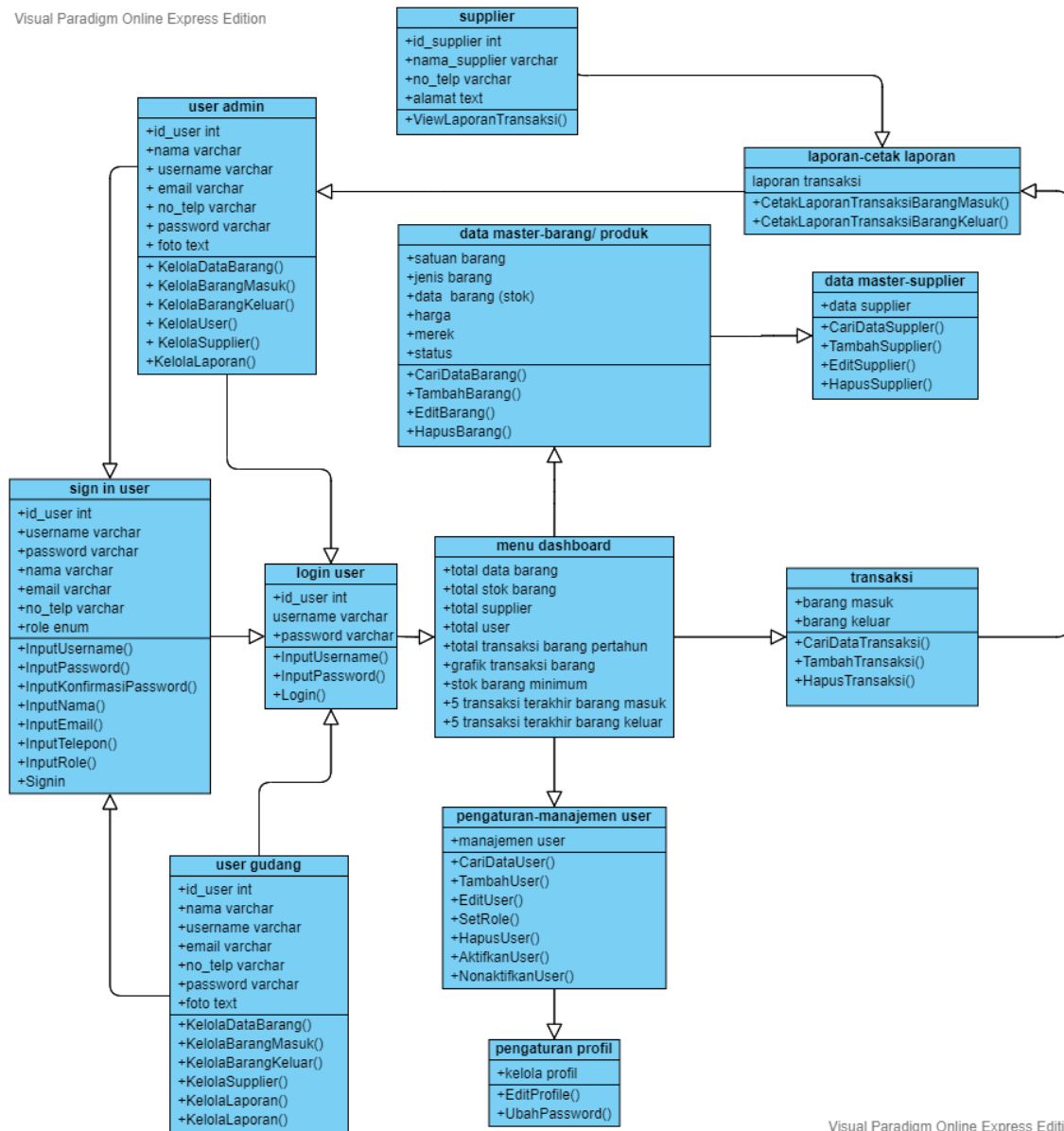


2. Menentukan Laporan Transaksi apa	Proses penentuan laporan transaksi barang masuk atau barang keluar yang akan dicetak dengan memilih opsi <i>radiobutton</i>
3. Mengatur tanggal riwayat transaksi barang	Proses mengatur tanggal riwayat transaksi barang yang akan dicetak
4. Cetak laporan transaksi	Proses mencetak laporan transaksi

### 3.2.2 Class Diagram

Berikut ini adalah *class diagram* dari pengembangan aplikasi sistem inventori berbasis web dengan *framework* CI (*code igniter*) yang telah dibuat.

*Gambar 22 Class Diagram*



Diatas ini adalah gambar Class Diagram yang terdiri objek-objek yang digunakan di dalam pengembangan aplikasi sistem inventori berbasis web dengan framework CI (code igniter), yang meliputi :

1. *Login user* : *Input username, input password, login* (masuk)
2. *Sign in user* : *input username, input password, input konfirmasi password, input nama, input email, input telepon, input role (user admin/ user gudang), sign in* (registrasi)
3. *User admin* : *Kelola data barang, kelola barang masuk, kelola barang keluar, kelola user, kelola supplier, kelola laporan*

4. *User gudang* : kelola data barang, kelola barang masuk, kelola barang keluar, kelola *supplier*, kelola laporan
5. *Menu dashboard* : total data barang, total stok barang, total *supplier*, total user, total transaksi barang pertahun, grafik transaksi barang, stok barang minimum, 5 transaksi terakhir barang masuk, 5 transaksi terakhir barang keluar
6. Data master-barang/ produk : tambah barang, edit barang, cari data barang, hapus barang
7. Data master-*supplier* : tambah *supplier*, edit *supplier*, cari data *supplier*, hapus *supplier*
8. Pengaturan-manajemen *user* : tambah *user*, edit *user*, cari data *user*, hapus *user*, *set role*, aktifkan *user*, nonaktifkan *user*
9. Pengaturan *profile* : *edit profile*, ubah *password*
10. Transaksi : cari data transaksi, tambah transaksi, hapus transaksi
11. Laporan-cetak laporan : cetak laporan transaksi barang masuk, cetak laporan transaksi barang keluar
12. *Supplier* : menerima laporan transaksi

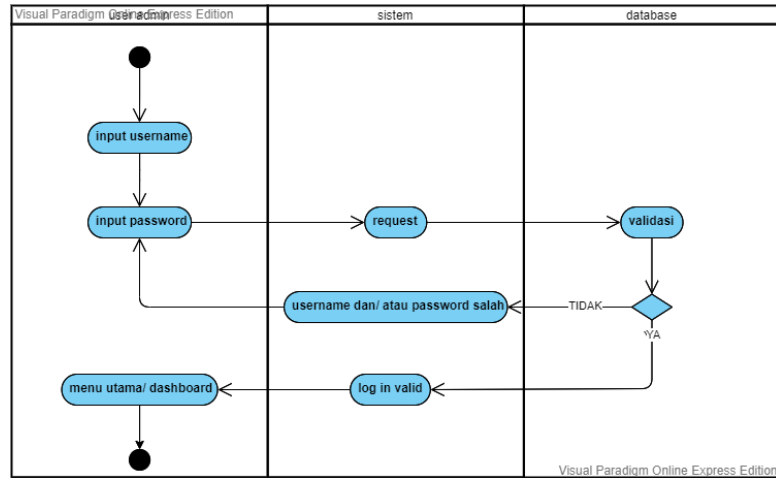
### 3.2.3 Activity Diagram

*Activity diagram* mempresentasikan setiap alur aktivitas dalam sistem yang sedang dirancang dan berjalan, bagaimana masing-masing alur berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Berikut adalah diagram diagram activity dari pengembangan aplikasi sistem inventori berbasis web dengan *framework* CI (*code igniter*) yang telah dibuat.

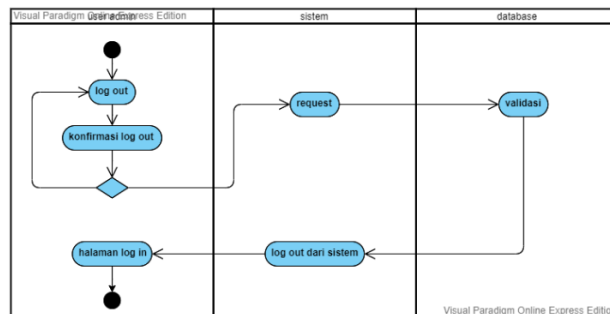
.

Gambar 23 Activity Diagram Log in



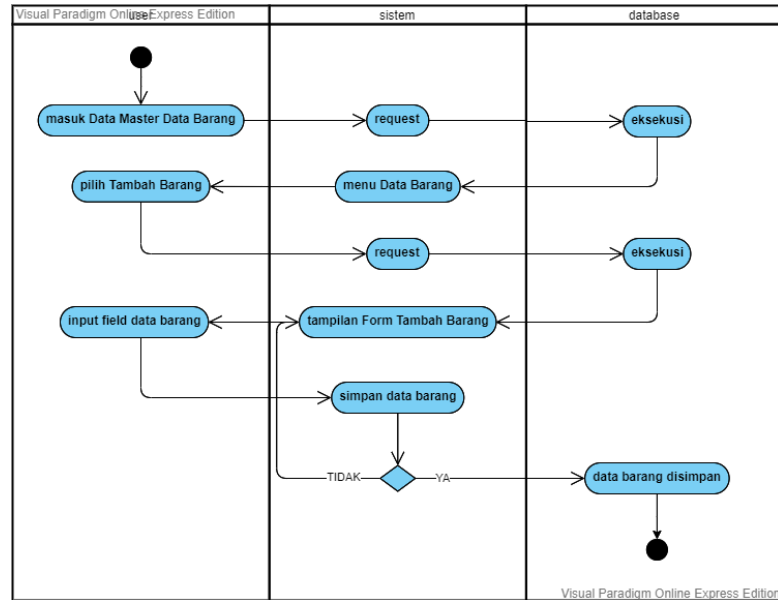
Disini user (*admin* atau *gudang*) melakukan *login* dengan meng-*input*-kan *username* dan *password*. Pada sistem melakukan *request* ke *database* terhadap data *login user*, jika valid maka *user* akan dilempar ke menu utama/ *dashboard* (masuk ke *session* aplikasi *web*). Jika tidak valid atau *user* tidak terdaftar di *database*, maka validasi gagal dan *user* akan dikembalikan ke halaman *login*.

Gambar 24 Activity Diagram Log out



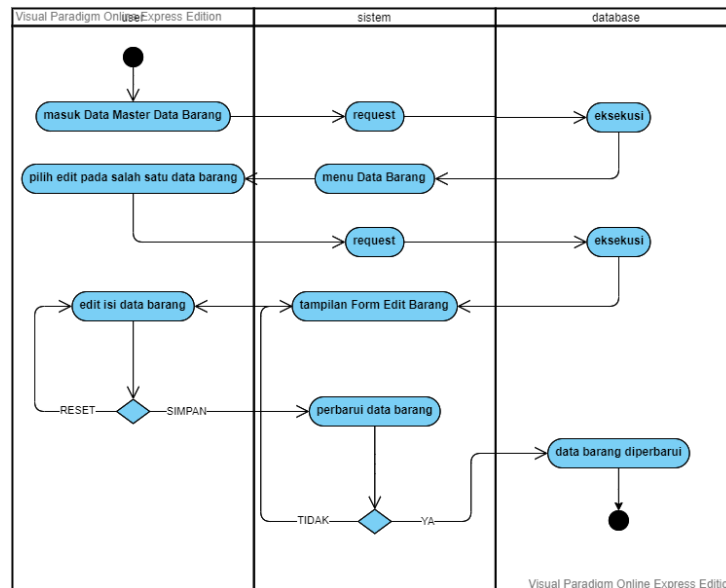
User (*admin* atau *gudang*) melakukan *logout* dimana jika *user* konfirmasi untuk *logout*, maka sistem akan melakukan *request* ke *database* untuk memvalidasi proses *logout* dari *user* tersebut, setelah itu *session* dari *user* tersebut berhenti, dan *user* akan dilempar ke halaman *login*.

Gambar 25 Activity Diagram Tambah Data Barang



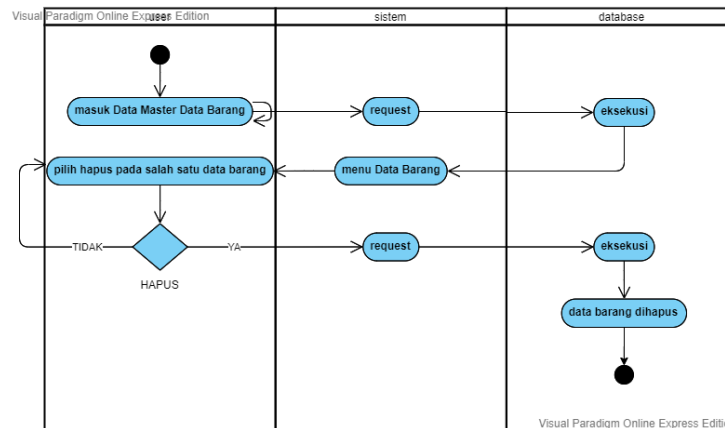
User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu data master, submenu data barang, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel barang, disini *user* akan masuk ke menu data barang. Saat *user* ingin memilih *input* tambah barang, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *user* masuk ke halaman *form* tambah barang dimana *user* dapat menambah data pada tabel barang. Lalu *user* dapat menyimpan data barang, jika *user* membatalkannya, maka *user* kembali ke *form* tambah barang. Jika *user* menyimpan data barang pada sistem, maka data barang akan disimpan pada tabel barang di *database*.

Gambar 26 Activity Diagram Edit Data Barang



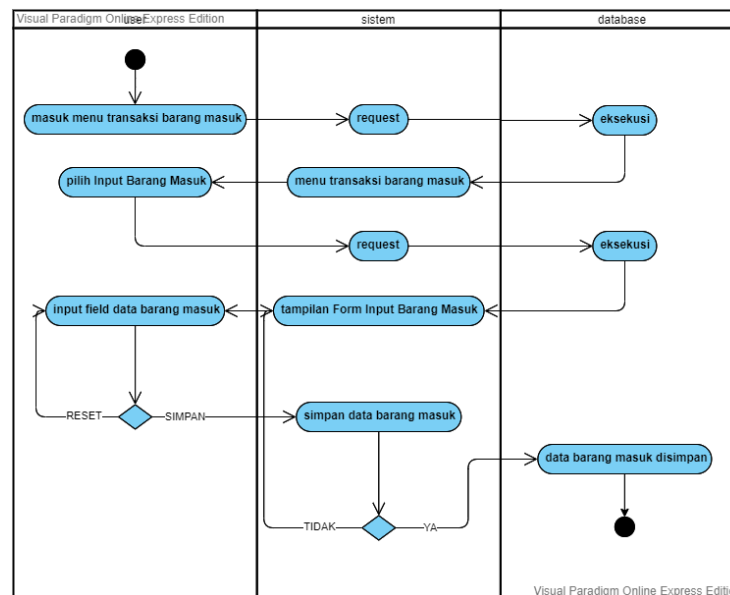
User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu data master, submenu data barang, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel barang, disini *user* akan masuk ke menu data barang. Saat *user* ingin memilih *edit* data barang, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *user* masuk ke halaman *form edit* barang dimana *user* dapat mengubah data pada tabel barang. Lalu *user* dapat menyimpan data barang yang diperbarui, jika *user* membatalkannya, maka *user* kembali ke *form edit* barang. Jika *user* menyimpan data barang pada sistem, maka data barang tersebut akan diperbarui pada tabel barang di *database*.

Gambar 27 Activity Diagram Hapus Data Barang



User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu data master, submenu data barang, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel barang, disini user akan masuk ke menu data barang. Saat user ingin memilih hapus data barang, Jika user membatalkannya, maka, *request* tidak akan dijalankan dari sistem ke *database*, jika user ingin menghapus data barang maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga data barang tersebut dihapus dari tabel barang di *database*.

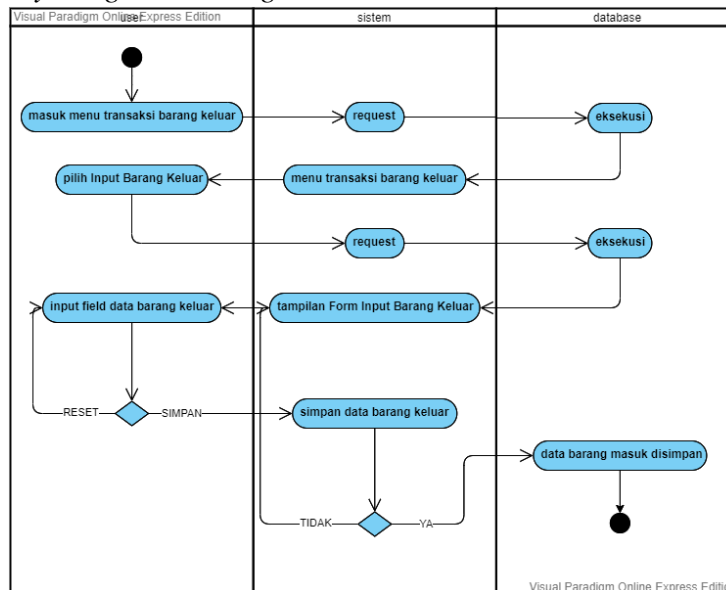
Gambar 28 Activity Diagram Barang Masuk



User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu transaksi, submenu transaksi barang masuk maka sistem akan melakukan *request* ke

*database* dan akan dieksekusi untuk menampilkan konten pada tabel *barang\_masuk*, disini *user* akan masuk ke menu transaksi barang masuk. Saat *user* ingin memilih *input* barang masuk, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *user* masuk ke halaman *form* barang masuk. *User* dapat meng-*input*-kan data barang masuk baru. Jika data yang diisi tidak sesuai, maka *user* bisa me-*reset field* yang diisi dan bisa mengulang *inputan*-nya. Ketika *user* ingin menyimpannya, maka data akan disimpan didalam tabel *barang\_masuk* pada *database*. Jika *user* membatalkannya, maka *field* akan ter-*reset* dan kembali ke *form input* barang masuk.

Gambar 29 Activity Diagram Barang Keluar

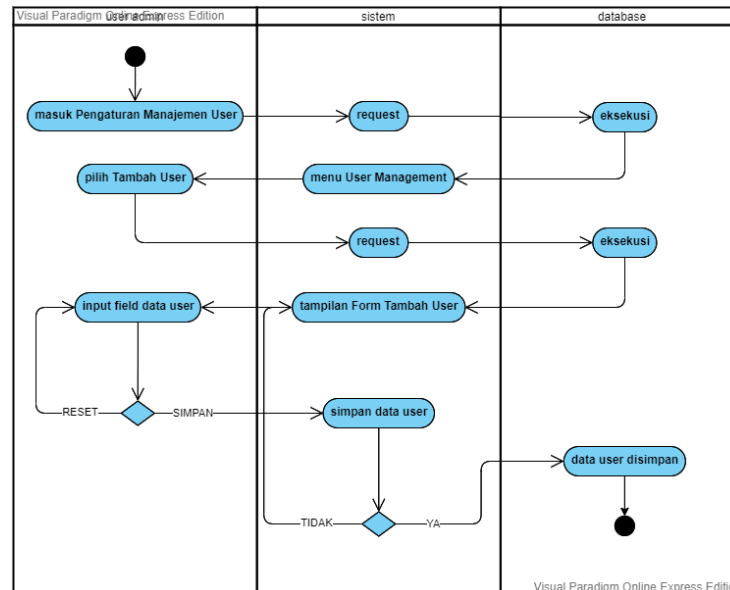


*User* (*admin* atau *gudang*) sudah *login* di *web* aplikasi sistem inventori. *User* masuk ke menu transaksi, submenu transaksi barang keluar maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel *barang\_keluar*, disini *user* akan masuk ke menu transaksi barang keluar. Saat *user* ingin memilih *input* barang keluar, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *user* masuk ke halaman *form* barang keluar. *User* dapat meng-*input*-kan data barang keluar baru. Jika data yang diisi tidak sesuai, maka *user* bisa me-*reset field* yang diisi dan bisa mengulang *inputan*-nya. Ketika *user* ingin menyimpannya, maka data akan disimpan



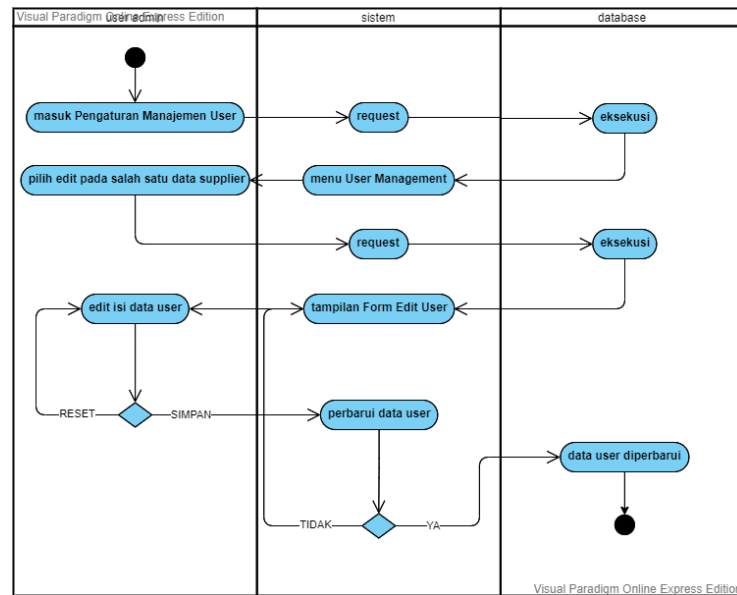
didalam tabel barang\_keluar pada *database*. Jika *user* membatalkannya, maka *field* akan ter-reset dan kembali ke *form input* barang keluar.

Gambar 30 Activity Diagram Tambah User



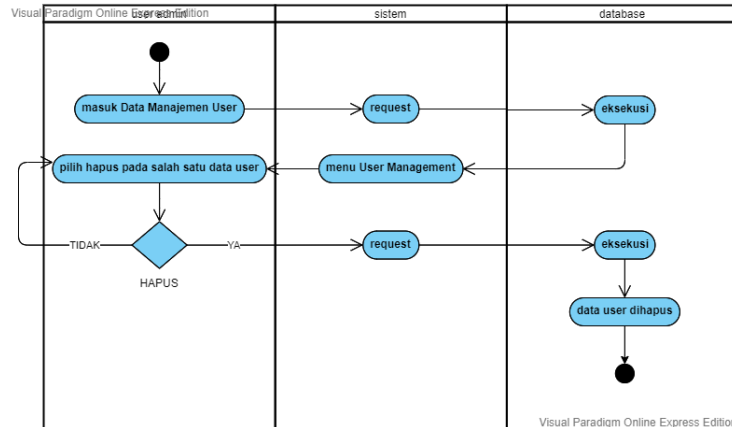
*Admin* sudah *login* di *web* aplikasi sistem inventori. *Admin* masuk ke menu pengaturan manajemen *user*, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel *user*. *Admin* memilih tambah *user*, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *admin* masuk ke halaman *form* tambah *user*. *Admin* dapat meng-*input*-kan data *user* baru. Jika data yang diisi tidak sesuai, maka *admin* bisa me-*reset* field yang diisi dan bisa mengulang *inputan*-nya. Ketika *admin* ingin menyimpannya, maka data akan disimpan didalam tabel *user* pada *database*. Jika *admin* membatalkannya, maka *field* akan ter-*reset* dan kembali ke *form* tambah *user*.

Gambar 31 Activity Diagram Edit User



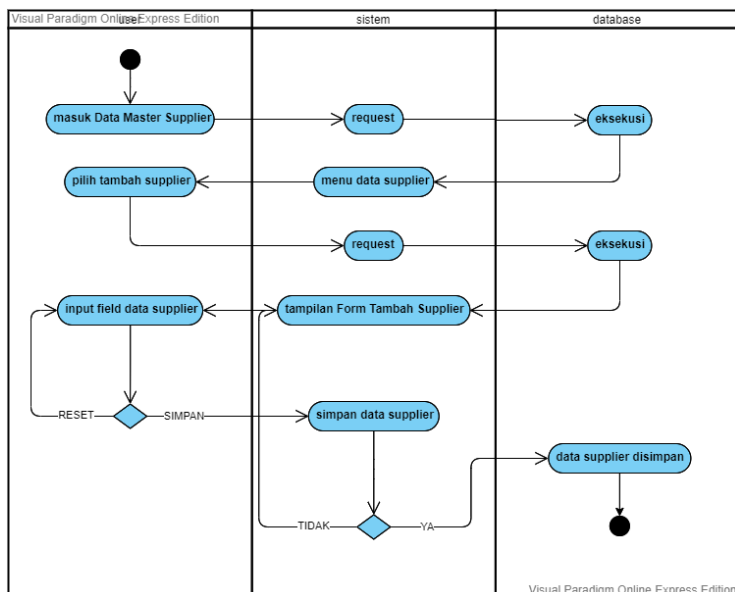
Admin sudah login di web aplikasi sistem inventori. Admin masuk ke menu pengaturan manajemen user, maka sistem akan melakukan request ke database dan akan dieksekusi untuk menampilkan konten pada tabel user. Admin memilih edit data user, maka sistem akan melakukan request ke database dan dieksekusi sehingga admin masuk ke halaman form edit user dimana admin dapat mengubah data pada tabel user. Jika data yang ingin diperbarui tidak sesuai, maka admin bisa me-reset field yang diisi dan bisa mengulang inputan-nya. Setelah itu admin dapat menyimpan data user yang diperbarui, jika admin membatalkannya, maka admin kembali ke form edit user. Jika admin menyimpan data user pada sistem, maka data user tersebut akan diperbarui pada tabel user di database.

Gambar 32 Activity Diagram Hapus User



Admin sudah login di web aplikasi sistem inventori. admin masuk ke menu pengaturan manajemen user, maka sistem akan melakukan request ke database dan akan dieksekusi untuk menampilkan konten pada tabel user. Admin memilih hapus data user, Jika admin membatalkannya, maka, request tidak akan dijalankan dari sistem ke database, jika admin ingin menghapus data user maka sistem akan melakukan request ke database dan dieksekusi sehingga data user tersebut dihapus dari tabel user di database.

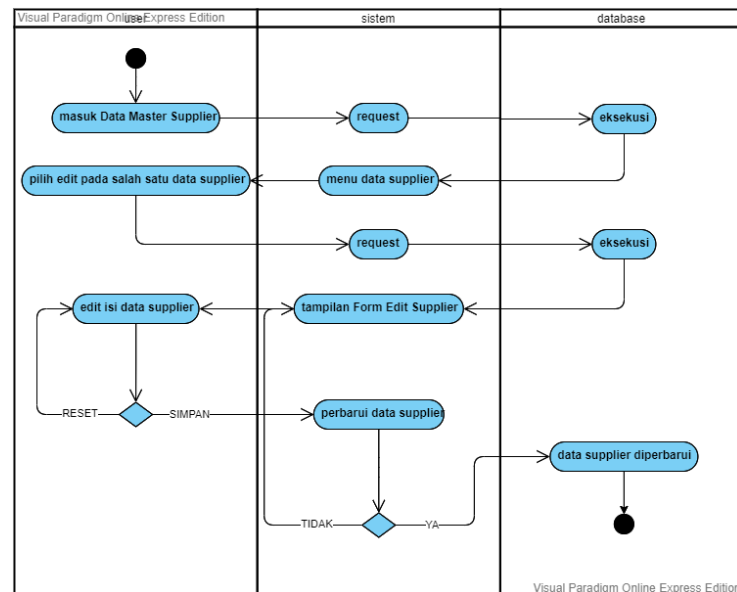
Gambar 33 Activity Diagram Tambah Supplier



User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu data master, submenu supplier, maka sistem akan melakukan request ke database dan akan dieksekusi untuk menampilkan konten pada tabel supplier. User memilih tambah

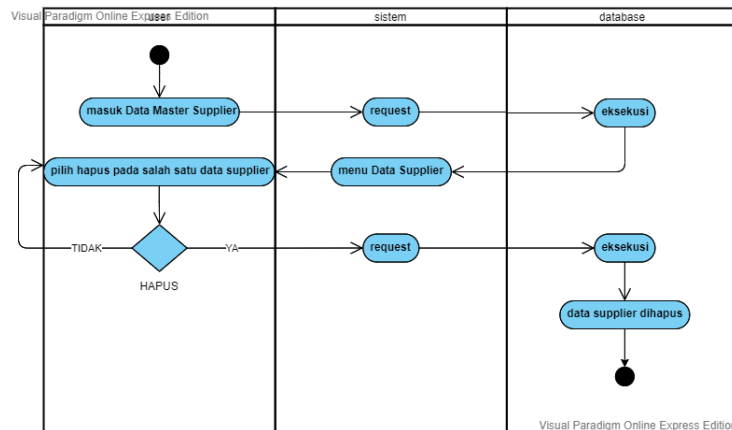
*supplier*, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *user* masuk ke halaman *form* tambah *supplier*. *User* dapat meng-*input*-kan data *supplier* baru. Jika data yang diisi tidak sesuai, maka *user* bisa me-*reset field* yang diisi dan bisa mengulang *inputan*-nya. Ketika *user* ingin menyimpannya, maka data akan disimpan didalam tabel *supplier* pada *database*. Jika *user* membatalkannya, maka *field* akan ter-*reset* dan kembali ke *form* tambah *supplier*.

Gambar 34 Activity Diagram Edit Supplier



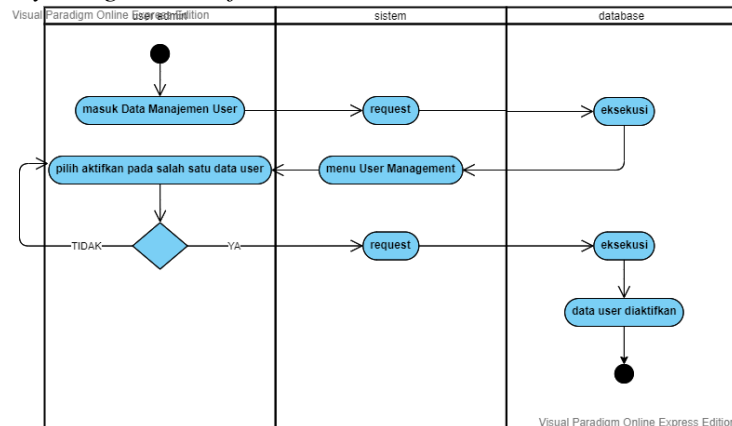
*User* (admin atau gudang) sudah *login* di *web* aplikasi sistem inventori. *User* masuk ke menu data master, submenu *supplier*, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel *supplier*. *User* memilih *edit* data *supplier*, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *user* masuk ke halaman *form edit supplier* dimana *user* dapat mengubah data pada tabel *supplier*. Jika data yang ingin diperbarui tidak sesuai, maka *user* bisa me-*reset field* yang diisi dan bisa mengulang *inputan*-nya. Setelah itu *user* dapat menyimpan data *supplier* yang diperbarui, jika *user* membatalkannya, maka *user* kembali ke *form edit supplier*. Jika *user* menyimpan data *supplier* pada sistem, maka data *supplier* tersebut akan diperbarui pada tabel *supplier* di *database*.

Gambar 35 Activity Diagram Hapus Supplier



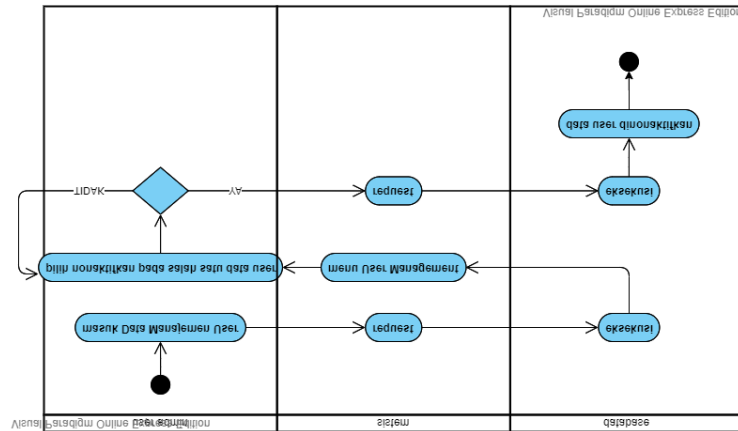
User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu data master, submenu *supplier*, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel *supplier*. User memilih hapus data *supplier*, Jika user membatalkannya, maka *request* tidak akan dijalankan dari sistem ke *database*, jika user ingin menghapus data *supplier* maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga data *supplier* tersebut dihapus dari tabel *supplier* di *database*.

Gambar 36 Activity Diagram Aktifkan User



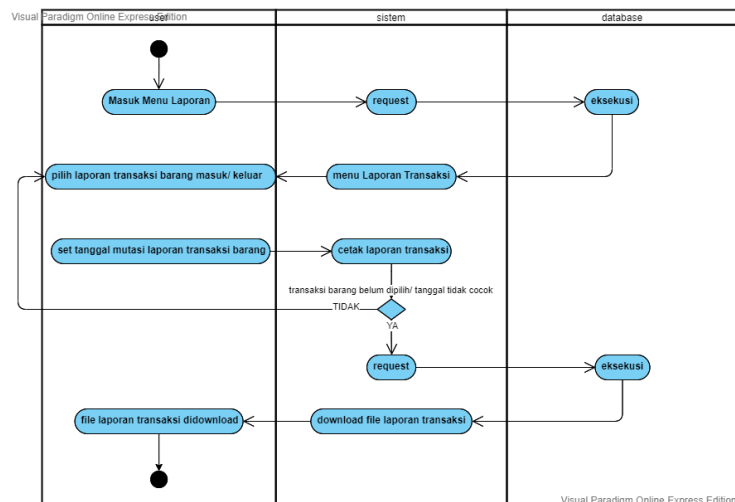
User Admin sudah login di web aplikasi sistem inventori. Admin masuk ke menu data manajemen user, maka sistem akan melakukan *request* ke *database* dan akan dieksekusi untuk menampilkan konten pada tabel *user*. Admin mempunyai akses untuk mengaktifkan data *user* yang sudah ditambahkan (terdaftar). Jika admin mengaktifkan data *user* tersebut, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga data *user* tersebut dapat diaktifkan.

Gambar 37 Activity Diagram Nonaktifkan User



User Admin sudah login di web aplikasi sistem inventori. Admin masuk ke menu data manajemen user, maka sistem akan melakukan request ke database dan akan dieksekusi untuk menampilkan konten pada tabel user. Admin mempunyai akses untuk menonaktifkan data user yang sudah ditambahkan (terdaftar). Jika admin menonaktifkan data user tersebut, maka sistem akan melakukan request ke database dan dieksekusi sehingga data user tersebut dapat dinonaktifkan.

Gambar 38 Activity Diagram Cetak Laporan



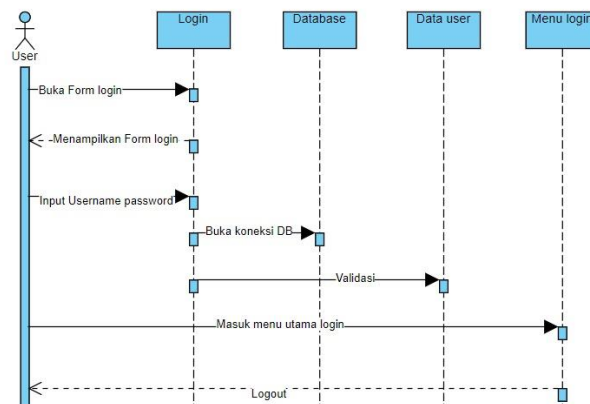
User (admin atau gudang) sudah login di web aplikasi sistem inventori. User masuk ke menu cetak laporan transaksi, maka sistem akan melakukan request ke database dan akan dieksekusi. User berada di menu laporan transaksi, User memilih laporan transaksi barang masuk atau barang keluar yang akan dicetak dan menge-set tanggal mutasi transaksi

barang masuk atau barang keluar yang sudah dilakukan. Selanjutnya *user* akan mencetak laporan transaksi, jika *user* membatalkannya, maka cetak laporan tidak akan dieksekusi. Jika *user* mencetak laporan transaksi, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *file* transaksi laporan tersebut di-*generate* didalam sistem dan siap di-*download* oleh *user*.

### 3.2.4 Sequence Diagram

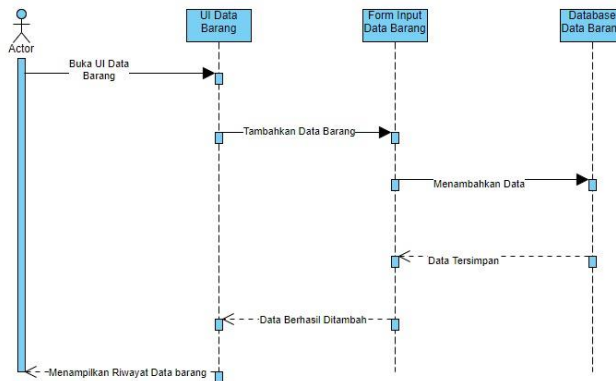
Berikut adalah diagram diagram activity dari pengembangan aplikasi sistem inventori berbasis web dengan *framework* CI (*code igniter*) yang telah dibuat.

Gambar 39 Sequence Diagram Login



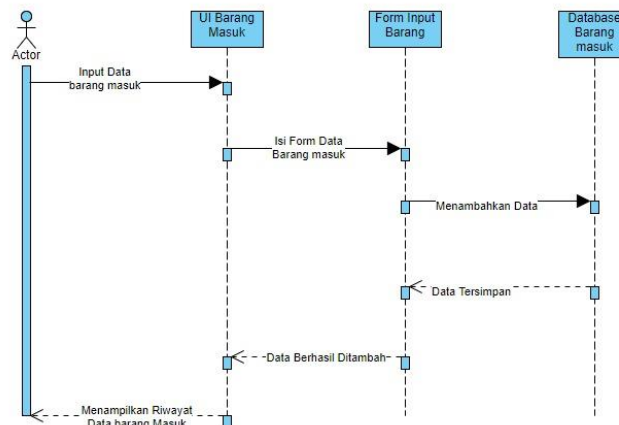
*User* (admin dan gudang) membuka *form login* lalu sistem menampilkan *form login*. Lalu *user* meng-*input*-kan *username* dan *password*, maka disini sistem akan membuka koneksi *database* dan memvalidasi apakah *user* tersebut sudah terdaftar atau tidak. Jika tidak valid maka *login* tidak berjalan. Jika valid, maka *user* masuk ke menu utama aplikasi *web* dan *session* dibuka. Jika *user logout*, maka *session* user berakhir.

Gambar 40 Sequence Diagram Data Barang (Stok)



User (admin dan gudang) membuka dan masuk ke UI menu data barang. User dapat menambahkan data barang dengan meng-input-kan data pada *form input* data barang. User menambah data barang maka data barang disimpan ke *database* tabel barang. Maka data barang yang baru ditambahkan akan ditampilkan di menu data barang (stok) dan bisa di-manage oleh user (admin dan gudang).

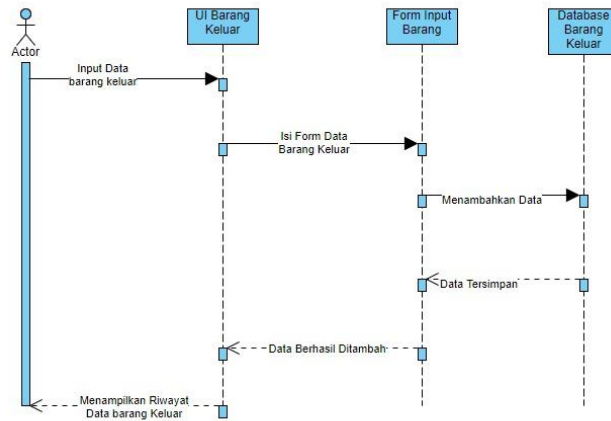
Gambar 41 Sequence Diagram Barang Masuk



User (admin dan gudang) membuka dan masuk ke UI menu barang masuk. User dapat menambahkan data barang masuk dengan meng-input-kan data pada *form input* barang masuk. User menambah data barang masuk maka data barang masuk disimpan ke *database* tabel barang\_masuk. Maka data barang masuk yang baru ditambahkan akan ditampilkan di menu barang masuk dan bisa di-manage oleh user (admin dan gudang).

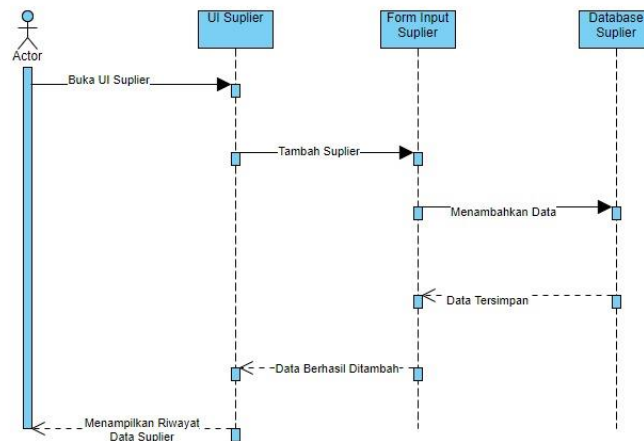


Gambar 42 Sequence Diagram Barang Keluar



User (*admin* dan gudang) membuka dan masuk ke UI menu barang keluar. User dapat menambahkan data barang keluar dengan meng-*input*-kan data pada *form input* barang keluar. User menambah data barang keluar maka data barang keluar disimpan ke *database* tabel *barang\_keluar*. Maka data barang keluar yang baru ditambahkan akan ditampilkan di menu barang keluar dan bisa di-*manage* oleh user (*admin* dan gudang).

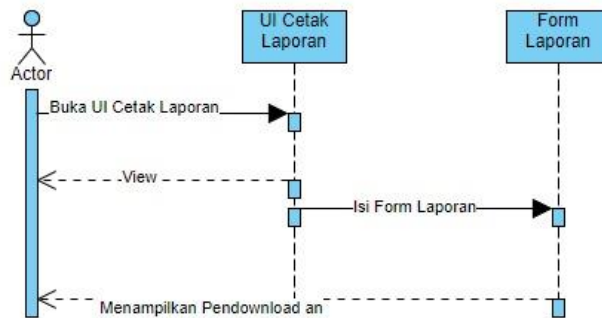
Gambar 43 Sequence Diagram Supplier



User (*admin* dan gudang) membuka dan masuk ke UI menu *supplier*. User dapat menambahkan data *supplier* dengan meng-*input*-kan data pada *form input supplier*. User menambah data *supplier* maka data *supplier* disimpan ke *database* tabel *supplier*. Maka

data *supplier* yang baru ditambahkan akan ditampilkan di menu *supplier* dan bisa di-*manage* oleh user (*admin* dan gudang).

Gambar 44 Sequence Diagram Laporan

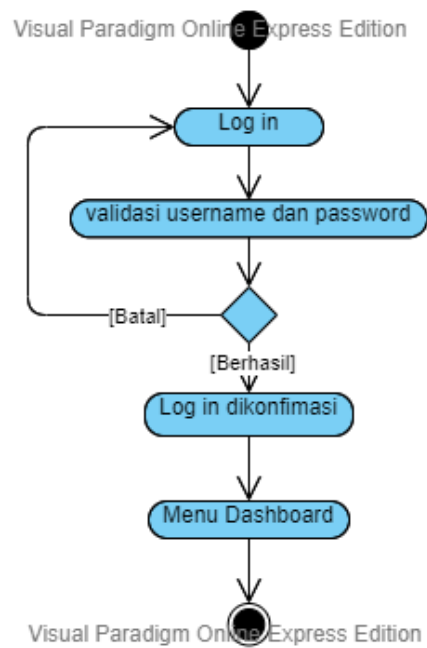


User (*admin* dan gudang) membuka dan masuk ke UI menu cetak laporan. User dapat melakukan cetak laporan transaksi dengan memilih transaksi barang masuk atau barang keluar dan dan menge-*set* tanggal mutasi transaksi barang masuk atau barang keluar yang sudah dilakukan. User akan mencetak laporan transaksi, maka sistem meng-*generate file* transaksi laporan tersebut dan siap di-*download* oleh user.

### 3.2.5 Statechart Diagram

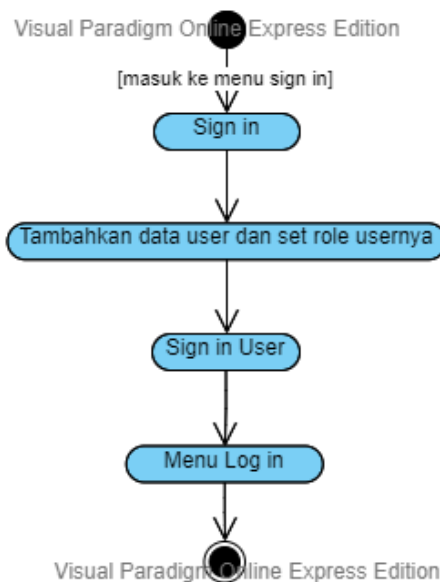
Berikut adalah diagram diagram activity dari pengembangan aplikasi sistem inventori berbasis web dengan *framework* CI (*code igniter*) yang telah dibuat. Selanjutnya user akan mencetak laporan transaksi, jika user membatalkannya, maka cetak laporan tidak akan dieksekusi. Jika user mencetak laporan transaksi, maka sistem akan melakukan *request* ke *database* dan dieksekusi sehingga *file* transaksi laporan tersebut di-*generate* di dalam sistem dan siap di-*download* oleh user.

Gambar 45 Statechart Diagram Login



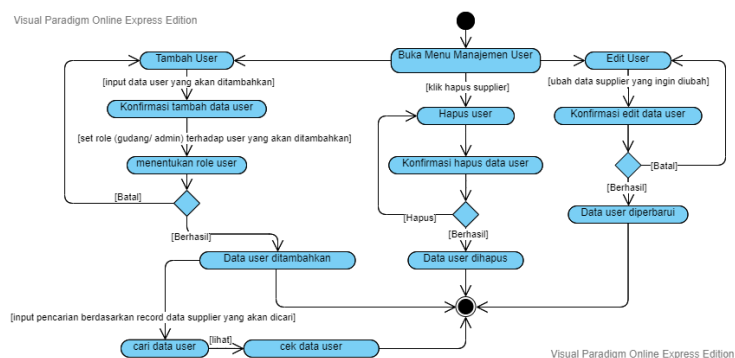
*Login* dilakukan, sistem memvalidasi *username* dan *password* yang di-*input*-kan, jika informasi *login* user tidak terdaftar di *database*, maka *login* dibatalkan. Jika valid maka *login* berhasil dan dikonfirmasi setelah masuk ke menu utama *dashboard*.

Gambar 46 Statechart Diagram Sign in



*Sign in* dilakukan, *user* menambahkan data *user* dan *set role user*. Setelah itu *sign in user*, disini *user* yang baru didaftarkan (*sign in*) belum diaktifkan oleh *user admin*, maka disini *admin* yang mempunyai hak dan akses terhadap manajemen data *user* termasuk mengaktifkan dan menonaktifkan *user*. Setelah itu akan dilempar ke menu *login*.

Gambar 47 Statechart Diagram User



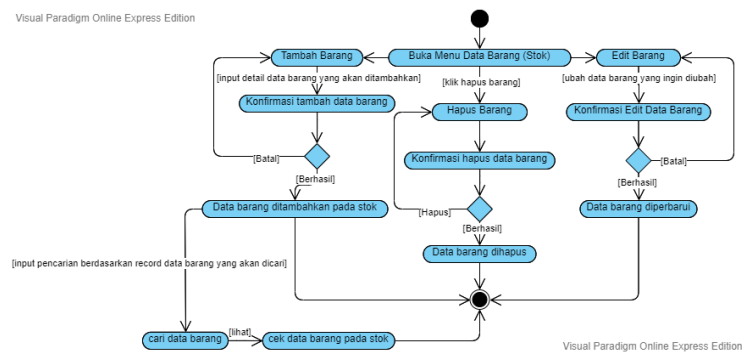
User admin membuka menu manajemen user.

- *Admin* dapat menambah *user* baru dengan *input* data *user* yang akan ditambahkan, *set role* (gudang atau *admin*) terhadap *user* baru yang akan ditambahkan. Jika *admin* batal untuk menyimpan data *user* baru, maka proses tambah *user* tidak akan dijalankan. Jika

*admin* menyimpan data *user*, maka data *user* akan tersimpan, selesai. *Admin* dapat mengecek data *user* yang baru disimpan.

- *Admin* dapat menghapus data *user*. Konfirmasi hapus data *user*, jika tidak (batal), maka proses hapus tidak akan dijalankan. Jika berhasil maka data *user* akan terhapus, selesai.
- *Admin* dapat meng-*edit* data *user*. Konfirmasi *edit* data *user*, jika tidak (batal), maka proses *edit* tidak akan dijalankan. Jika berhasil maka data *user* sukses diperbarui, selesai.

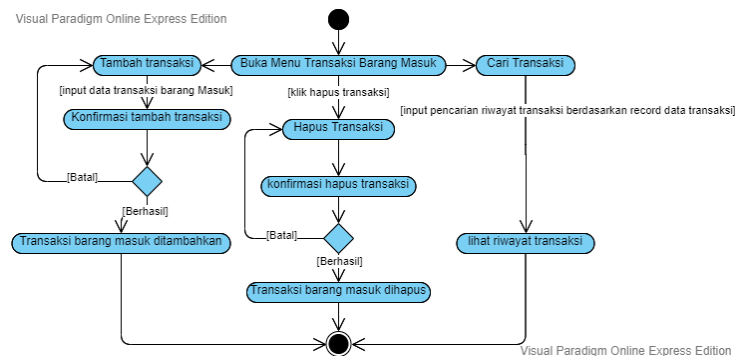
Gambar 48 Statechart Diagram Data Barang (Stok)



*User* (*admin* atau gudang) membuka menu data barang (stok).

- *User* dapat menambah data barang baru dengan *input* data barang yang akan ditambahkan. Jika *user* batal untuk menyimpan data barang baru, maka proses tambah barang tidak akan dijalankan. Jika *user* konfirmasi untuk menyimpan data barang, maka data barang akan tersimpan, selesai. *User* dapat mengecek data barang (stok) yang baru disimpan.
- *User* dapat menghapus data barang. Konfirmasi hapus data barang, jika tidak (batal), maka proses hapus tidak akan dijalankan. Jika berhasil maka data barang akan terhapus, selesai.
- *User* dapat meng-*edit* data barang. Konfirmasi *edit* data barang, jika tidak (batal), maka proses *edit* tidak akan dijalankan. Jika berhasil maka data barang sukses diperbarui, selesai.

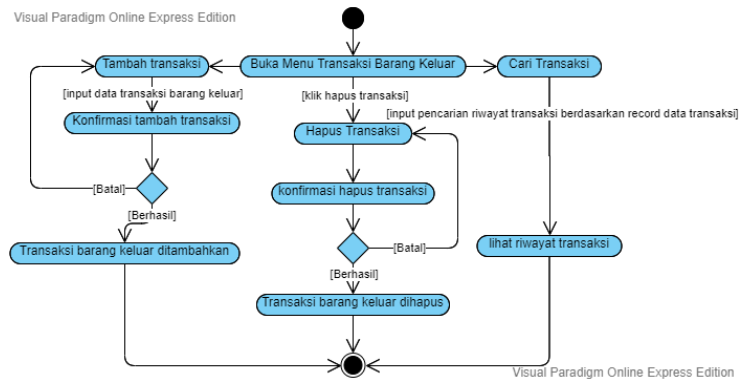
Gambar 49 Statechart Diagram Barang Masuk



User (*admin* atau gudang) membuka menu transaksi barang masuk.

- User dapat menambah transaksi barang masuk baru dengan *input* data transaksi. Jika user batal untuk menyimpan data transaksi barang masuk baru, maka proses tambah transaksi tidak akan dijalankan. Jika user konfirmasi untuk menyimpan data transaksi barang masuk, maka data transaksi akan tersimpan, selesai.
- User dapat menghapus data transaksi barang masuk. Konfirmasi hapus data transaksi barang masuk, jika tidak (batal), maka proses hapus tidak akan dijalankan. Jika berhasil maka data transaksi akan terhapus, selesai.
- User dapat meng-*edit* data transaksi barang masuk. Konfirmasi *edit* data transaksi barang masuk, jika tidak (batal), maka proses *edit* tidak akan dijalankan. Jika berhasil maka data transaksi sukses diperbarui, selesai.

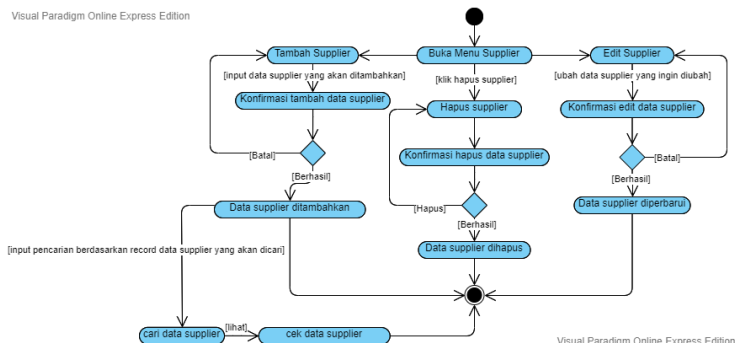
Gambar 50 Statechart Diagram Barang Keluar



User (admin atau gudang) membuka menu transaksi barang keluar.

- User dapat menambah transaksi barang keluar baru dengan *input* data transaksi. Jika user batal untuk menyimpan data transaksi barang keluar baru, maka proses tambah transaksi tidak akan dijalankan. Jika user konfirmasi untuk menyimpan data transaksi barang keluar, maka data transaksi akan tersimpan, selesai.
- User dapat menghapus data transaksi barang keluar. Konfirmasi hapus data transaksi barang keluar, jika tidak (batal), maka proses hapus tidak akan dijalankan. Jika berhasil maka data transaksi akan terhapus, selesai.
- User dapat meng-*edit* data transaksi barang keluar. Konfirmasi *edit* data transaksi barang keluar, jika tidak (batal), maka proses *edit* tidak akan dijalankan. Jika berhasil maka data transaksi sukses diperbarui, selesai.

Gambar 51 Statechart Diagram Supplier



User (admin atau gudang) membuka menu *supplier*.

- User dapat menambah *supplier* baru dengan *input* data *supplier* yang akan ditambahkan. Jika user batal untuk menyimpan data *supplier* baru, maka proses tambah

- supplier* tidak akan dijalankan. Jika *user* menyimpan data *supplier*, maka data *supplier* akan tersimpan, selesai. *User* dapat mengecek data *supplier* yang baru disimpan.
- *User* dapat menghapus data *supplier*. Konfirmasi hapus data *supplier*, jika tidak (batal), maka proses hapus tidak akan dijalankan. Jika berhasil maka data *supplier* akan terhapus, selesai.
  - *User* dapat meng-*edit* data *supplier*. Konfirmasi *edit* data *supplier*, jika tidak (batal), maka proses *edit* tidak akan dijalankan. Jika berhasil maka data *supplier* sukses diperbarui, selesai.

Gambar 52 Statechart Diagram Laporan



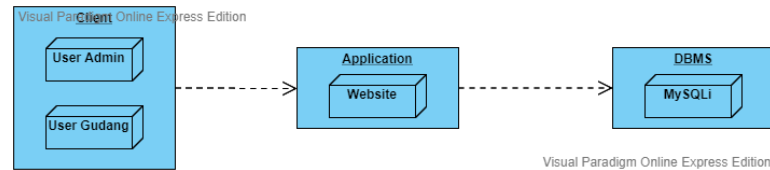
*User* (*admin* atau gudang) membuka cetak laporan transaksi. *User* dapat memilih transaksi barang masuk atau barang keluar dan menge-*set* tanggal mutasi riwayat transaksi barang masuk atau barang keluar yang sudah dilakukan. Selanjutnya *user* dapat mencetak informasi laporan barang masuk atau barang keluar tersebut, selesai.



### 3.2.6 Deployment Diagram

Berikut adalah diagram diagram activity dari pengembangan aplikasi sistem inventori berbasis web dengan *framework* CI (*code igniter*) yang telah dibuat.

*Gambar 53 Deployment Diagram*



Pada lingkungan package Client disini memiliki dua node yaitu enduser admin dan gudang yang dapat mengakses langsung ke lingkungan package application yang berisi node website yaitu sistem aplikasi web tersebut dimana memiliki akses ke lingkungan package DBMS (database manaegement system) yang berisi node MySQLi.

## BAB IV

### IMPLEMENTASI DAN PENGUJIAN

#### 4.1 Implementasi

Implementasi sistem ini meliputi kebutuhan perangkat lunak dan perangkat keras yang digunakan, pemrograman dan pengujian program dan pengujian sistem yang telah dirancang sesuai kebutuhan dalam pengerjaan aplikasi sistem inventori berbasis web dengan framework CI (code igniter) ini.

##### 4.1.1 Implementasi Perangkat Keras

Perangkat keras yang diperlukan untuk keperluan menjalankan sistem inventori ini diantaranya adalah sebagai berikut.

*Tabel 13 Tabel Implementasi Perangkat Keras*

Perangkat Keras	Keterangan
<i>Processor</i>	Intel(R) Core(TM) i7-3520M CPU @ 2.90GHz (4 CPUs), ~2.9GHz dan sejenis, AMD E2 7015 dan sejenis.
<i>Memory</i>	Minimum 1 GB Memory RAM, Recommended 4 GB Memory RAM.
<i>Storage</i>	150 mb recommended requirement storage
<i>Monitor</i>	Standarisasi alat pendukung monitor dari personal computer (PC) dan laptop sesuai kebutuhan.
<i>Keyboard &amp; Mouse</i>	Standarisasi alat pendukung penggunaan personal computer (PC) dan laptop sesuai kebutuhan.

#### 4.1.2 Implementasi Perangkat Lunak

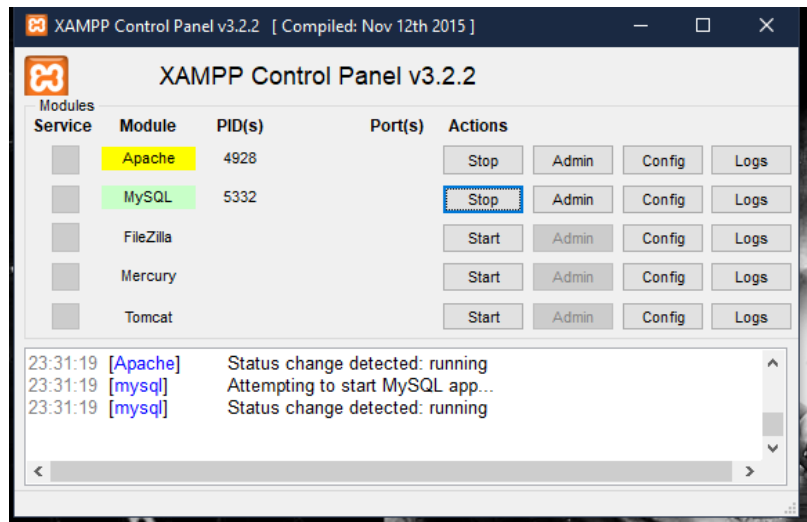
Untuk mendukung pengembangan aplikasi sistem inventori berbasis web dengan framework CI (code igniter) ini dibutuhkan perangkat lunak sebagai pendukung untuk menjalankan aplikasi dan tool pengerjaannya, diataranya adalah sebagai berikut.

*Tabel 14 Tabel Implementasi Perangkat Lunak*

Perangkat Lunak	Keterangan
Windows 10 Home	Sistem Operasi
XAMPP 3.2.2	<i>Web Server</i>
Code Igniter 3	<i>Framework</i>
Bootstrap 4	Bagian desain aplikasi (css)
Visual Studio Code	<i>Tool text editor</i>
Sublime Text	<i>Tool text editor</i>
<i>Browser (Chrome)</i>	Untuk mejalankan <i>source code</i>
<i>Database dan DBMS MySQL</i>	Memanaajemen penyimpanan data sistem ( <i>database</i> )

## 4.2 Perancangan Database

Gambar 54 Tampilan XAMPP saat mulai

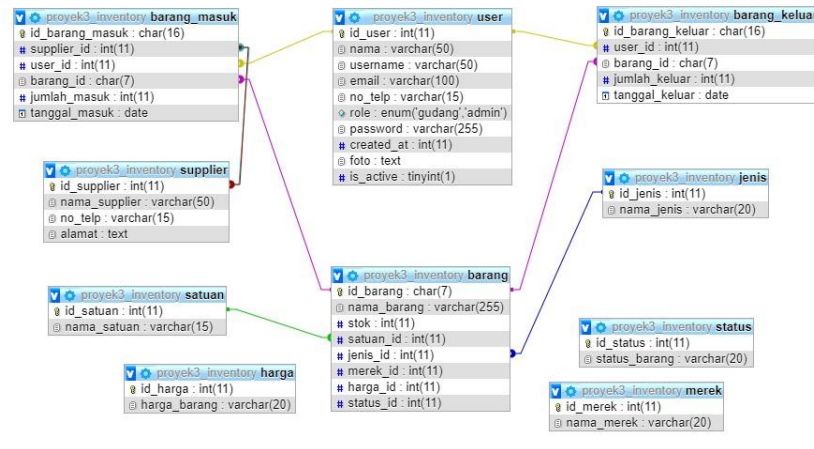


Sebelum membuat perancangan aplikasi, hal yang dilakukan pertama kali adalah mengaktifkan XAMPP terlebih dahulu, yaitu menjalankan fitur service server *Apache* dan *MySQL*.

Gambar 55 Database Aplikasi Sistem Inventori dan Tabel-Tabelnya

Table	Action	Rows	Type	Collation	Size	Overhead
barang	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_general_ci	48 K1B	-
barang_keluar	Browse Structure Search Insert Empty Drop	6	InnoDB	utf8_general_ci	48 K1B	-
barang_masuk	Browse Structure Search Insert Empty Drop	8	InnoDB	utf8_general_ci	64 K1B	-
harga	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 K1B	-
jenis	Browse Structure Search Insert Empty Drop	7	InnoDB	utf8_general_ci	16 K1B	-
merek	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 K1B	-
satuan	Browse Structure Search Insert Empty Drop	4	InnoDB	utf8_general_ci	16 K1B	-
status	Browse Structure Search Insert Empty Drop	2	InnoDB	latin1_swedish_ci	16 K1B	-
supplier	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8_general_ci	16 K1B	-
user	Browse Structure Search Insert Empty Drop	5	InnoDB	utf8_general_ci	16 K1B	-
10 tables	Sum				272 K1B	0 B

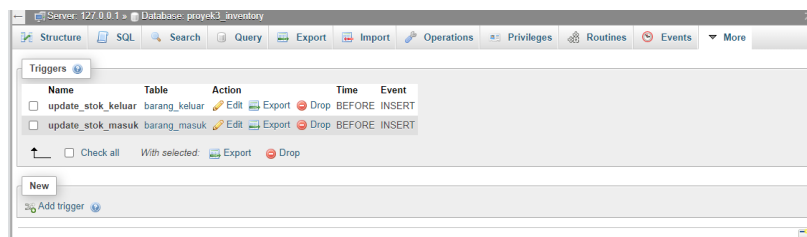
Gambar 56 Desain Database Aplikasi Sistem Inventori



Pembuatan *database* pada *Perancangan Aplikasi Sistem Inventori Berbasis Codeigniter* ini bertujuan untuk menaruh dan menyimpan data barang yang nantinya akan berelasi dengan program / *script* yang dibuat.

Pada gambar 4.2 yaitu kumpulan *database* yang sudah dibuat untuk memenuhi kebutuhan pada projet 3 yaitu *Pengembangan Aplikasi Sistem Inventori Berbasis Web dengan framework CI (Code Igniter)*, dan pada gambar selanjutnya yaitu desain *database* pada aplikasi sistem inventori dimana terdapat 10 tabel didalamnya.

Gambar 57 Fungsi Trigger pada Aplikasi Sistem Inventori



Setelah itu ada implementasi *trigger* pada aplikasi sistem inventori yang dibuat. Yaitu terdapat dua (2) *trigger* masing masing dengan nama : `update_stok_keluar` dan `update_stok_masuk`.

Gambar 58 Fungsi Trigger *update\_stok\_keluar*

The screenshot shows a window titled "Edit trigger" with a "Details" tab. The fields are as follows:

Field	Value
Trigger name	update_stok_keluar
Table	barang_keluar
Time	BEFORE
Event	INSERT
Definition	1 UPDATE `barang` SET `barang`.`stok` = `barang`.`stok` - NEW.jumlah_keluar WHERE `barang`.`id_barang` = NEW.barang_id
Definer	root@localhost

At the bottom right, there are "Go" and "Close" buttons.

Pada *trigger* *update\_stok\_keluar* mempunyai relasi dengan tabel *barang keluar* dimana fungsinya ini adalah jika *user* menambah transaksi data *barang keluar*, maka fungsi *trigger* *update\_stok\_keluar* akan berjalan dimana data *barang* yang ada pada tabel *barang* (*stok*) akan diperbaharui sesuai jumlah transaksi *barang keluar* yang dibuat. Seperti gambar diatas, memiliki detail, yakni :

- Trigger name : *update\_stok\_keluar*
- Time : BEFORE
- Event : INSERT
- Definition :

```
UPDATE `barang` SET `barang`.`stok` = `barang`.`stok` - NEW.jumlah_keluar WHERE  
`barang`.`id_barang` = NEW.barang_id
```

Disini query ini dapat dibaca dengan *trigger* *update\_stok\_keluar* dilakukan saat sebelum data dimasukkan (*insert*) dimana data *stok barang* akan di-update pada tabel *barang* dengan jumlah transaksi *barang keluar* tertentu yang baru dimasukkan dimana *id barang* dari tabel *barang* didefinisikan menjadi *NEW.barang\_id*.

Gambar 59 Fungsi Trigger *update\_stok\_masuk*

The screenshot shows a window titled "Edit trigger" with a close button. Inside, there's a "Details" tab. The fields are as follows:

Field	Value
Trigger name	update_stok_masuk
Table	barang_masuk
Time	BEFORE
Event	INSERT
Definition	<pre>1 UPDATE `barang` SET `barang`.`stok` = `barang`.`stok` + NEW.jumlah_masuk WHERE `barang`.`id_barang` = NEW.barang_id</pre>
Definer	root@localhost

At the bottom right, there are "Go" and "Close" buttons.

Pada *trigger* *update\_stok\_masuk* mempunyai relasi dengan tabel barang masuk dimana fungsinya ini adalah jika *user* menambah transaksi data barang masuk, maka fungsi *trigger* *update\_stok\_masuk* akan berjalan dimana data barang yang ada pada tabel barang (*stok*) akan diperbaharui sesuai jumlah transaksi barang masuk yang dibuat. Seperti gambar diatas, memiliki detail, yakni :

- Trigger name : *update\_stok\_masuk*
- Time : BEFORE
- Event : INSERT
- Definition :

```
UPDATE `barang` SET `barang`.`stok` = `barang`.`stok` - NEW.jumlah_masuk WHERE `barang`.`id_barang` = NEW.barang_id
```

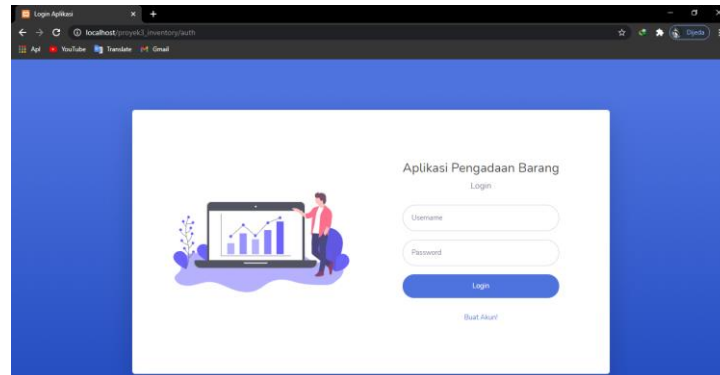
Disini query ini dapat dibaca dengan trigger *update\_stok\_masuk* dilakukan saat sebelum data dimasukkan (*insert*) dimana data *stok* barang akan di-update pada tabel barang dengan jumlah transaksi barang masuk tertentu yang baru dimasukkan dimana *id* barang dari tabel barang didefinisikan menjadi *NEW.barang\_id*.

### 4.3 Tampilan Antar Muka Web

Berikut ini adalah tampilan antar muka (*user interface*) dari aplikasis sistem inventori yang dibuat.

### 4.3.1 Tampilan Halaman Login

*Gambar 60 Tampilan Login*

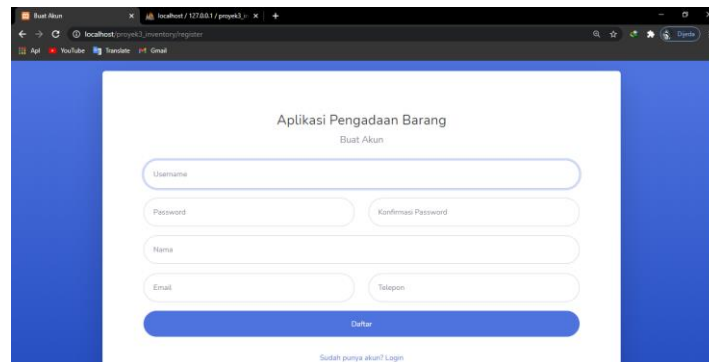


Disini terdapat tampilan halaman *login* yang akan digunakan oleh *user* admin maupun *use5* gudang untuk masuk ke menu utama aplikasi sistem inventori ini. Pada menu *login*, terdapat relasi dengan tabel *user*, dimana disini, *user* sebagai *admin* yang akan mengelola data barang dan manajemen user (admin dan gudang) di dalam sistem aplikasi ini lalu *user* gudang juga bertugas untuk mengelola barang. Di dalam table *user* sudah ada beberapa *field* yang mempunyai *attribute id*, *username* dan *password*. Dimana didalam fungsi *login* ini akan ada suatu *method* “IF” yang fungsinya ketika dijalankan apakah *input-an username* dan *password* nya sudah sesuai dengan data yang ada pada *database* atau belum.



### 4.3.2 Tampilan Registrasi

Gambar 61 Tampilan Menu Sign In (Registrasi) pada Aplikasi

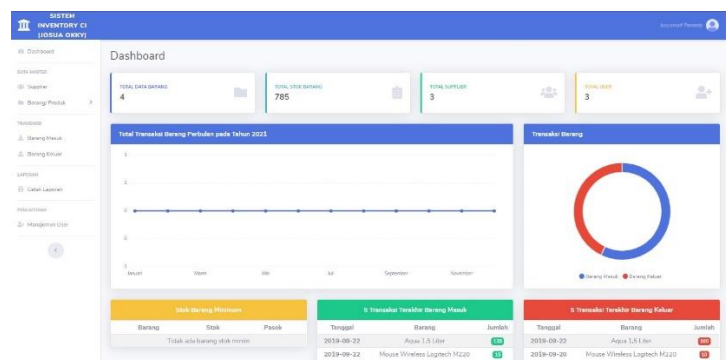


The screenshot shows a web browser window displaying a registration form for 'Aplikasi Pengadaan Barang'. The form is titled 'Buat Akun' and includes the following fields: Username, Password, Confirm Password, Name, Email, and Telephone. A blue 'Daftar' button is at the bottom. Below the button, there is a link that says 'Sudah punya akun? Login'.

Berikut ini adalah menu *sign in* (registrasi). User yang mendaftar disini nantinya data tersebut akan tersimpan di database tabel *user*. . Pada menu *Sign in*, terdapat relasi dengan tabel *user*, dimana disini, *user* sebagai *admin* yang akan mengelola data barang dan manajemen user (admin dan gudang) di dalam sistem aplikasi ini lalu user gudang juga bertugas untuk mengelola barang.

### 4.3.3 Tampilan Utama (User Admin Login)

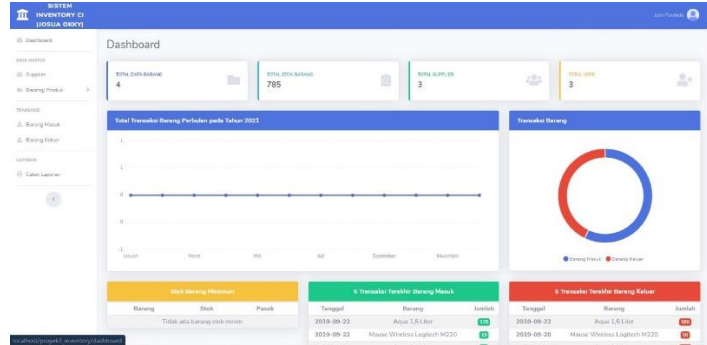
Gambar 62 Tampilan Utama Aplikasi Sebagai User Admin



Tampilan utama *user admin* ketika *login* yang ditampilkan oleh sistem yang didalamnya meliputi *dashboard*, setelah itu ada menu *supplier*, barang/produk, barang masuk, barang keluar, cetak laporan, dan manajemen *user* untuk mengatur user. Disini hanya *role user admin* yang dapat mengelola *user* pada menu manajemen *user*. *Admin* disini bertugas untuk memanajemen *user* (gudang dan *admin*), *set role* gudang dan *admin* pada *user* yang baru mendaftar dan mengaktifkan/ menonaktifkan *user* tersebut.

#### 4.3.4 Tampilan Utama (User Gudang Login)

Gambar 63 Tampilan Utama Aplikasi Sebagai User Gudang









Tampilan utama *user* gudang ketika *login* yang ditampilkan oleh sistem yang didalamnya meliputi *dashboard*, setelah itu ada menu *supplier*, barang/produk, barang masuk, barang keluar, dan cetak laporan. Disini *role user* gudang tidak dapat mengelola *user* pada menu manajemen *user* oleh karena itu tidak ada menu manajemen *user* pada *login user* gudang.

#### 4.3.5 Menu Supplier

Gambar 64 Tampilan Menu Data Supplier

The screenshot shows the 'Supplier' menu with a 'Data Supplier' table. The table has columns for 'No.', 'Nama', 'Nomor Telepon', 'Alamat', and 'Aksi'. There are three entries listed. Above the table, there are buttons for 'Tambah Supplier', 'Copy', 'CSV', 'Print', 'Excel', and 'PDF', along with a search bar. Below the table, there is a pagination bar showing 'Showing 1 to 3 of 3 entries' and 'Previous 1 Next'.

No.	Nama	Nomor Telepon	Alamat	Aksi
1	Ahmad Hasanudin	085688772971	Kec. Cigudeg, Bogor - Jawa Barat	 
2	Asep Salahudin	081341879246	Kec. Ciamepa, Bogor - Jawa Barat	 
3	Filo Lial	087728164328	Kec. Ciomas, Bogor - Jawa Barat	 

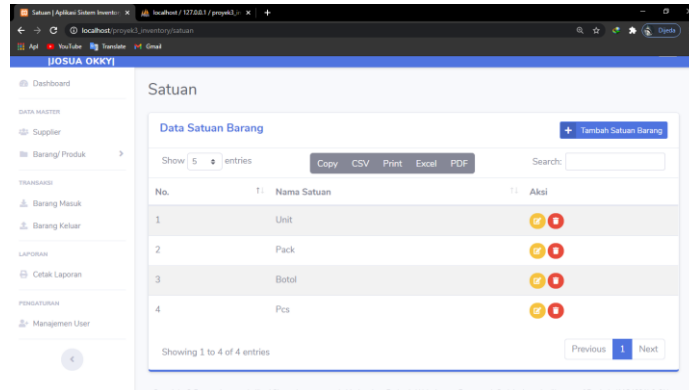
Pada tampilan *supplier* disini terdapat beberapa fitur untuk mengelola data *supplier*, yaitu tambah, ubah dan hapus data *supplier*. Tabel *supplier* disini berelasi dengan tabel Barang Masuk. Pendataan data *supplier* akan dilakukan oleh *user* sendiri saat ada *supplier* yang ingin menyimpan data barangnya pada sistem inventori ini.

#### 4.3.6 Barang / Produk

Menu Barang/ Produk adalah *sub menu* dari *Data Master* yang berisi *sub-sub menu* seperti : Satuan Barang, Jenis Barang, Merek Barang, Harga Barang, Status Barang, Data Barang.

##### a. Satuan Barang

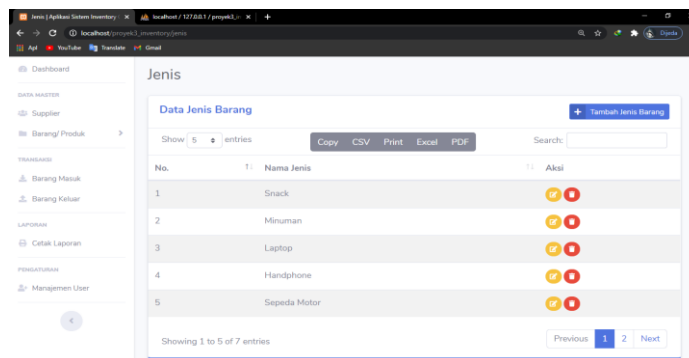
Gambar 65 Menu Satuan Barang



Di dalam *Menu data master-barang/ produk*, terdapat beberapa bagian antara lainya yaitu satuan. Pada *menu data master* satuan barang, *user* dapat mengelola data satuan barang. Terdapat fitur tambah, hapus dan ubah satuan barang. Tabel satuan disini berelasi pada tabel barang.

##### b. Jenis Barang

Gambar 66 Menu Jenis Barang







Di dalam *Menu data master-barang/ produk*, terdapat beberapa bagian antara lainya yaitu satuan. Pada *menu data master* satuan barang, *user* dapat mengelola data satuan

barang. Terdapat fitur tambah, hapus dan ubah satuan barang. Tabel satuan disini berelasi pada tabel barang.

### c. Merek Barang





*Gambar 67 Menu Merek Barang*

No.	Nama Merek	Aksi
1	Aqua	 
2	MSI (Laptop)	 

Di dalam *Menu data master-barang/ produk*, terdapat beberapa bagian antara lainya yaitu merek. Pada *menu data master* merek barang, *user* dapat mengelola data merek barang. Terdapat fitur tambah, hapus dan ubah merek barang. Tabel merek disini berelasi pada tabel barang.

### d. Harga Barang

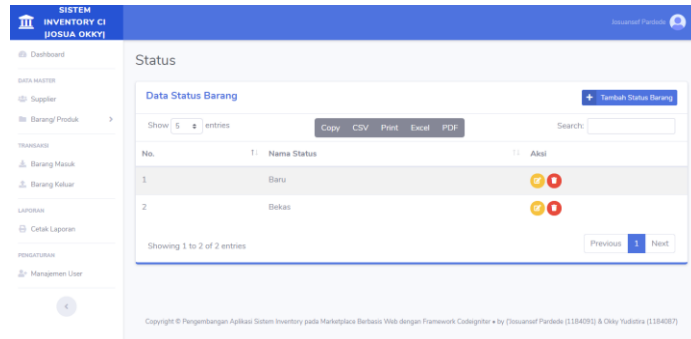
*Gambar 68 Menu Harga Barang*

No.	Nama Harga	Aksi
1	100.000	 
2	14.999.000	 

Di dalam *Menu data master-barang/ produk*, terdapat beberapa bagian antara lainya yaitu harga. Pada *menu data master* harga barang, *user* dapat mengelola data harga barang. Terdapat fitur tambah, hapus dan ubah harga barang. Tabel harga disini berelasi pada tabel barang.

## e. Status Barang

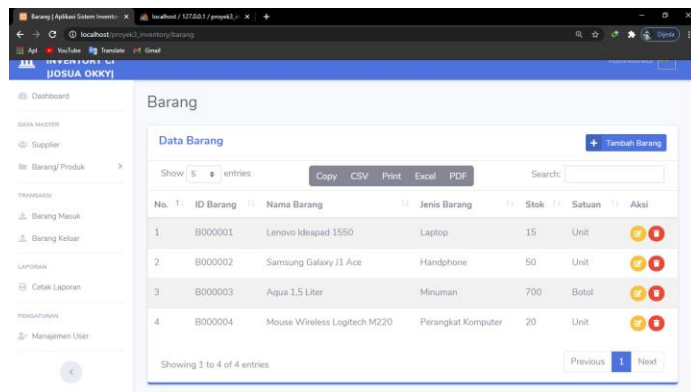
*Gambar 69 Menu Status Barang*



Di dalam *Menu data master-barang/ produk*, terdapat beberapa bagian antara lainya yaitu status. Pada *menu data master* status barang, *user* dapat mengelola data status barang. Terdapat fitur tambah, hapus dan ubah status barang. Tabel status disini berelasi pada tabel barang.

## f. Data Barang

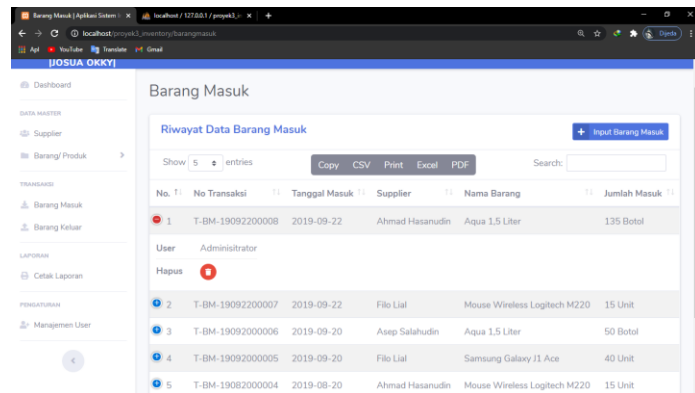
*Gambar 70 Tampilan Data barang*



Data barang adalah tempat dimana tampilan untuk melihat stok. Didalam tampilan data barang sama seperti semua tampilan didalam line Barang atau produk. Dimana terdapat beberapa menu yang sama yaitu fitur delete, ubah dan tambah barang untuk mengelolah. Table data barang ini paling banyak berelasi, yaitu pada satuan, jenis barang, barang masuk dan barang keluar.

## g. Barang Masuk

Gambar 71 Tampilan dari barang masuk

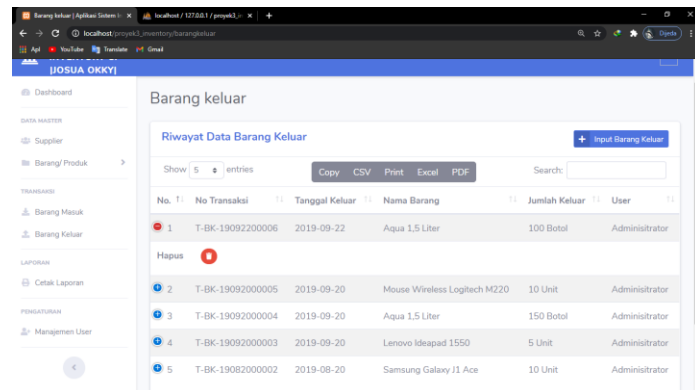


No.	No Transaksi	Tanggal Masuk	Supplier	Nama Barang	Jumlah Masuk
1	T-BM-19092200008	2019-09-22	Ahmad Hasanudin	Aqua 1,5 Liter	135 Botol
User: Administrator					
Hapus					
2	T-BM-19092200007	2019-09-22	Filo Lial	Mouse Wireless Logitech M220	15 Unit
3	T-BM-19092000006	2019-09-20	Asep Salehudin	Aqua 1,5 Liter	50 Botol
4	T-BM-19092000005	2019-09-20	Filo Lial	Samsung Galaxy J1 Ace	40 Unit
5	T-BM-19082000004	2019-08-20	Ahmad Hasanudin	Mouse Wireless Logitech M220	15 Unit

Tampilan *menu* barang masuk ini mempunyai beberapa fitur yang sedikit berbeda, disini tidak dapat di ubah, hanya tersedia fitur tambah dan hapus pada transaksi barang masuk yang dapat dilakukan oleh *user*. Table barang masuk disini berelasi dengan table barang, *supplier* dan *user*.

## h. Barang keluar

Gambar 72 Tampilan Barang keluar



No.	No Transaksi	Tanggal Keluar	Nama Barang	Jumlah Keluar	User
1	T-BK-19092200006	2019-09-22	Aqua 1,5 Liter	100 Botol	Administrator
Hapus					
2	T-BK-19092000005	2019-09-20	Mouse Wireless Logitech M220	10 Unit	Administrator
3	T-BK-19092000004	2019-09-20	Aqua 1,5 Liter	150 Botol	Administrator
4	T-BK-19092000003	2019-09-20	Lenovo Ideapad 1550	5 Unit	Administrator
5	T-BK-19082000002	2019-08-20	Samsung Galaxy J1 Ace	10 Unit	Administrator

Tampilan *menu* barang keluar ini mempunyai beberapa fitur yang sedikit berbeda, disini tidak dapat di ubah, hanya tersedia fitur tambah dan hapus pada transaksi barang keluar yang dapat dilakukan oleh *user*. Table barang keluar disini berelasi dengan table barang, *supplier* dan *user*.

### 4.3.7 Cetak Laporan / Laporan Transaksi

Gambar 73 Menu Cetak Laporan Transaksi

SISTEM INVENTORY CI (JOSUA OKKY)

Laporan Transaksi

Form Laporan

Laporan Transaksi ☒ Barang Masuk ☐ Barang Keluar

Tanggal 12/21/2020 - 01/19/2021

Cetak

Pada menu cetak laporan ini terdiri dari beberapa pilihan apakah ingin membuat laporan transaksi untuk barang masuk atau barang keluar . serta ada pilihan *Date* atau tanggal laporan yang berfungsi untuk melihat beberapa perkembangan barang mereka (mutasi transaksi barang). Biasanya laporan transaksi ini digunakan saat supplier meminta bukti laporan fisik dan akan diberikan secara manual oleh pihak gudang. Ketika laporan akan dicetak akan mengeluarkan laporan yang outputnya berupa PDF.

### 4.3.8 Manajemen User

Gambar 74 Tampilan Manajemen user

User Management | Aplikasi Sistem Inventory CI

Administrator

User Management

Data User

Show 5 entries

Copy CSV Print Excel PDF

Search:

No.	Foto	Nama	Username	Email	No. telp	Role
1		Josuansel Pardede	josua18	josuansel18@gmail.com	085233535495	admin
2		Okky Yudistira	okky	okky Yudistira06@gmail.com	088886666	gudang

Showing 1 to 2 of 2 entries

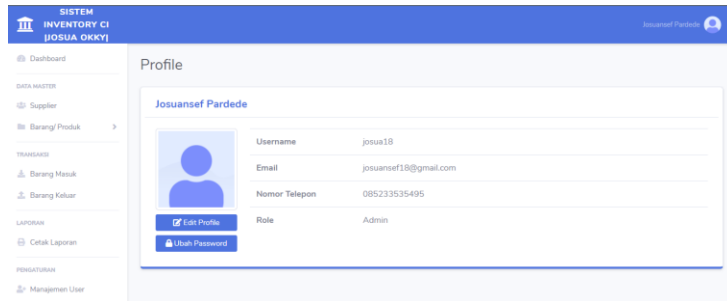
Previous 1 Next

Copyright © Pengembangan Aplikasi Sistem Inventory pada Marketplace Berbasis Web dengan Framework Codeigniter • by (Josuansel Pardede (1184091) & Okky Yudistira (1184087))

Menu manajemen user ini hanya ada ketika user admin login pada sistem aplikasi, tidak terdapat pada user gudang login. Dimana fitur yang tersedia disini adalah seperti hapus, edit dan tambah user.

### 4.3.9 User Profile

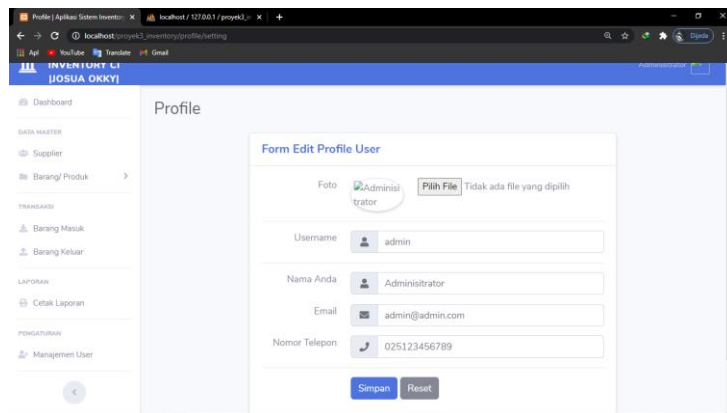
*Gambar 75 Menu Profile*



Menu profile ini terdiri dari dua (2) fitur penting yaitu edit profile (pengaturan) dan ubah password.

### 4.3.10 Pengaturan

*Gambar 76 Tampilan menu Setting atau pengaturan pada profile*



Menu setting ini ada di profile untuk mengubah data untuk mengganti beberapa data yang ada pada profile kita seperti username, email, nomor telepon, dan foto.



### 4.3.11 Ubah Password

*Gambar 77 Tampilan Fitur Password pada menu profile*

Ubah Password / Aplikasi Sistem - X localhost / 127.0.0.1 / proyek3 X

SISTEM INVENTORY CI (JOSUA OKKY) Administrator

Ubah Password

Form Ubah Password

Password Lama Password Lama...

Password Baru Password Baru...

Konfirmasi Password Konfirmasi Password...

Simpan

Copyright © Pengembangan Aplikasi Sistem Inventory pada Marketplace Berbasis Web dengan Framework Codeigniter 4 by Cloussant Pande (1354091) & Okky Yudhira (1354097)

Tampilan menu ubah password ini untuk mengubah password pada user saat ini. Dapat dilakukan jika user ingin mengubah password

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Setelah melakukan perancangan, implementasi dan pengujian pada pengembangan aplikasi sistem inventori berbasis *web* dengan *framework* CI (*code igniter*) ini, terdapat beberapa kesimpulan, yaitu :

1. Telah dihasilkan aplikasi Pergudangan yang memberikan kemudahan untuk mengelolah suatu gudang atau inventory dengan menggunakan aplikasi sistem inventori berbasis web dengan framework CI (*Code igniter*) ini
2. Dengan menggunakan aplikasi sistem inventori berbasis web dengan framework CI (*Code igniter*), mempercepat dan mempermudah kita dalam pembuatan *website*.
3. Dalam pembuatan aplikasi pergudangan berbasis CI ini Menghasilkan struktur pemrograman yang sangat rapi, baik dari segi kode maupun struktur *file* phpnya. Memberikan standar *coding* sehingga memudahkan kita atau orang lain untuk mempelajari kembali sistem aplikasi yang dibangun.
4. Data aplikasi sistem inventori dapat diolah secara terstruktur dalam proses pencarian data barangnya.

#### 5.2 Saran

Dalam mengembangkan aplikasi ini, ada beberapa saran yang berhubungan dengan pengembangan aplikasi yang dibuat, diantaranya sebagai berikut:

1. Aplikasi sistem inventori ini dapat diperluas fungsinya dengan memperbaharui teknologi sesuai perkembangannnya dan sesuai dengan kebutuhan yang diinginkan.
2. Aplikasi sistem inventori dapat dikembangkan lagi dengan menggabungkan/ mengintegrasikan teknologi lain seperti *finger print* untuk *login user* kedepanya.

## DAFTAR PUSTAKA

- [1] s. E-Commerce, "Kelola Stok di Marketplace secara Otomatis, Caranya?," Sirclo, 12 March 2020. [Online]. Available: <https://www.sirclo.com/kelola-stok-di-marketplace-secara-otomatis-caranya/>. [Accessed 29 October 2020].
- [2] K. kemdikbud, "KBBI Daring pengertian pengembangan," KBBI kemdikbud, 2016. [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/pengembangan>. [Accessed 29 October 2020].
- [3] K. kemdikbud, "KBBI Daring pengertian sistem," KBBI kemdikbud, 2016. [Online]. Available: <https://kbbi.kemdikbud.go.id/entri/sistem>. [Accessed 29 October 2020].
- [4] "mengenal apa itu framework beserta jenisnya," CloudHost, 2 August 2017. [Online]. Available: <https://idcloudhost.com/mengenal-apa-itu-framework-beserta-jenisnya/>. [Accessed 29 October 2020].
- [5] M. Riadi, "pengertian tujuan dan manfaat gudang," Kajian Pustaka, 30 April 2016. [Online]. Available: <https://www.kajianpustaka.com/2016/04/pengertian-tujuan-dan-manfaat-gudang.html>. [Accessed 29 October 2020].
- [6] A. Rahmatullah, "Belajar Unified Modeling Language (UML)," CodePolitan, 21 January 2019. [Online]. Available: <https://www.codepolitan.com/unified-modeling-language-uml>. [Accessed 15 January 2021].
- [7] M. Roziq, "Pengertian UseCase Diagram (UML), Komponen dan Simbolnya," Caraguna.com, 16 July 2020. [Online]. Available: <https://caraguna.com/pengertian-use-case-diagram-adalah/>. [Accessed 04 02 2021].
- [8] N. I. Imanudin, "Pengertian, Manfaat Hingga Contoh Class Diagram," TISUCODING, 01 12 2020. [Online]. Available: <https://tisucoding.com/contoh-class-diagram/>. [Accessed 04 02 2021].

- [9] N. I. Imanudin, "Pengertian, Fungsi, Cara Membuat hingga Contoh Activity Diagram," Tisucoding, 07 12 2020. [Online]. Available: <https://tisucoding.com/contoh-activity-diagram/>. [Accessed 04 02 2021].
- [10] A. "Pengertian Sequence Diagram : Tujuan, Simbol, dan Contohnya," Ansoriweb, 04 01 2020. [Online]. Available: <https://www.ansoriweb.com/2020/04/pengertian-sequence-diagram.html>. [Accessed 04 02 2021].
- [11] A. Hasanudin, "Contoh Statechart Diagram," Guratgarut, 14 01 2021. [Online]. Available: <https://guratgarut.com/contoh-statechart-diagram/>. [Accessed 04 02 2021].
- [12] E. "Apa itu Component Diagram? Mengenal Component Diagram," Nesabamedia, 12 01 2021. [Online]. Available: <https://www.nesabamedia.com/component-diagram/>. [Accessed 04 02 2021].
- [13] A. Hasanudin, "Contoh Deployment Diagram," Guratgarut, 13 01 2021. [Online]. Available: <https://guratgarut.com/contoh-deployment-diagram/>. [Accessed 04 02 2021].
- [14] A. Hasanudin, "Contoh Collaboration Diagram," Guratgarut, 14 01 2021. [Online]. Available: <https://guratgarut.com/contoh-collaboration-diagram/>. [Accessed 04 02 2021].
- [15] H. Bagir and B. E. Putro, "Analisis Perancangan Sistem Informasi Pergudangan di CV. Karya Nugraha," *Jurnal Media Teknik dan Sistem Industri*, vol. 2, no. 1, p. 22, 2018.
- [16] elib.unikom.ac.id, "DAFTAR SIMBOL FLOW MAP," [Online]. Available: <https://elib.unikom.ac.id/download.php?id=82483>. [Accessed 04 02 2021].
- [17] C. I. L. Services, "Manajemen Pergudangan," CKB Integrated Logistics Services, [Online]. Available: <https://www.ckb.co.id/id/warehouse-management>. [Accessed 04 02 2021].
- [18] A. Indonesia, "Website Penyedia Gambar Ilustrasi Gratis," ALVIAN INDONESIA, 2020. [Online]. Available: <https://www.alvianindonesia.com/2020/05/website-penyedia-gambar-ilustrasi-gratis.html>. [Accessed 04 02 2021].

- [19] Waryanto, "Pengertian Website Lengkap dengan Jenis dan Manfaatnya," Niagahoster, 22 01 2018. [Online]. Available: [niagahoster.co.id/blog/pengertian-website/](https://niagahoster.co.id/blog/pengertian-website/). [Accessed 15 01 2021].
- [20] IdCloudHost, "Structured Query Language (SQL)," IdCloudHost, 22 01 2019. [Online]. Available: <https://idcloudhost.com/kamus-hosting/structured-query-language/>. [Accessed 2021 01 15].
- [21] admin, "BELAJAR QUERY DASAR PADA MYSQL," Blogon.id, [Online]. Available: <https://www.blogon.id/2017/01/belajar-query-dasar-pada-mysql.html>. [Accessed 04 02 2021].
- [22] A. S. Putra and O. M. Febriani, "Sistem Informasi Monitoring Inventori Barang pada Balai Riset Standarisasi Industri Bandar Lampung," *Jurnal Informatika*, vol. 13, no. 1, p. 92, 2013.
- [23] A. N. Ubaidah, "Pengertian JavaScript, Keuntungan dan Kekurangan Menggunakannya," LOGIQUE, 18 11 2020. [Online]. Available: <https://www.logique.co.id/blog/2020/11/18/keuntungan-pengertian-javascript/>. [Accessed 04 02 2021].
- [24] W. Komputer, "Tips Mudah Menjalankan XAMPP," Warung Komputer, 12 04 2019. [Online]. Available: <https://warungkomputer.com/tips-mudah-menjalankan-xampp/>. [Accessed 04 02 2021].
- [25] Tulus, "Framework CSS yang Siap Membantu Membuat Website," AGOAGA, [Online]. Available: <https://www.agoaga.com/framework-css-yang-siap-membantu-membuat-website/>. [Accessed 04 02 2021].
- [26] php.net, "Download Logos and Icons PHP Logo," php.net, [Online]. Available: <https://www.php.net/download-logos.php>. [Accessed 04 02 2021].
- [27] Guntoro, "10 Framework PHP yang Paling Populer di Tahun 2020," Badoy Studio, 09 09 2020. [Online]. Available: <https://badoystudio.com/framework-php/>. [Accessed 04 02 2021].

- [28] T. Owl, "Pengertian dan Cara Menggunakan Sublime Text," [Online]. Available: <https://www.teknoowl.com/2019/03/pengertian-dan-cara-menggunakan-sublime.html>.
- [29] nesabamedia, "Pengertian Bootstrap Beserta Fungsi, Kekurangan, dan Kelebihan Bootstrap," [Online]. Available: <https://www.nesabamedia.com/pengertian-bootstrap/>.
- [30] info@thidiweb.com, "Mengenal Framework Twitter Bootstrap dan Penggunaannya Pada Website," [Online]. Available: <https://thidiweb.com/pengertian-bootstrap/>. [Accessed 20 1 2020].
- [31] M. Arumsari, "Microsoft Visual Studio Code," [Online]. Available: <https://www.dicoding.com/blog/microsoft-visual-studio-code/>.
- [32] D. P. 3, "XAMPP adalah," [Online]. Available: <https://www.dosenpendidikan.co.id/xampp-adalah/>.
- [33] A. ÄMNEN, "Yang perlu Anda ketahui tentang Microsoft Edge Browser pada Windows 10," [Online]. Available: <http://id.wondershare.com/windows10/microsoft-edge-browser-on-windows-10.html>.
- [34] B. Winarso, "Apa Itu Google Chrome dan Sepenggal Sejarahnya," [Online]. Available: [dailysocial.id/post/apa-itu-google-chrome/](https://dailysocial.id/post/apa-itu-google-chrome/).
- [35] A. C., "Pemahaman Dasar Tentang Bahasa Markup Hypertext," [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-html/>.
- [36] Andre, "Tutorial Belajar PHP Part 1: Pengertian dan Fungsi PHP dalam Pemrograman Web," [Online]. Available: [duniailkom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/](https://duniailkom.com/pengertian-dan-fungsi-php-dalam-pemrograman-web/).
- [37] A. C., "Pemahaman Dasar Mengenai JavaScript bagi Para Pemula," [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-javascript/>.
- [38] N. A., "Apa itu PHP? Tutorial Tentang PHP untuk Pemula," Hostinger, 13 07 2020. [Online]. Available: <https://www.hostinger.co.id/tutorial/apa-itu-php/>. [Accessed 2 1 2021].

- [39] A. C., "Perbedaan HTML dan HTML5," Hostinger, 14 12 2020. [Online]. Available: <https://www.hostinger.co.id/tutorial/perbedaan-html-dan-html5/>. [Accessed 2 1 2021].
- [40] B. Winarso, "Apa Itu Google Chrome dan Sepenggal Sejarahnya," DailySocial Indonesia, 23 3 2016. [Online]. Available: <https://dailysocial.id/post/apa-itu-google-chrome>. [Accessed 2 1 2021].
- [41] G. Kaonang, "Microsoft Edge Adalah Nama Resmi Browser Baru Windows 10," Daily Social Indonesia, 30 4 2015. [Online]. Available: <https://dailysocial.id/post/microsoft-edge-adalah-nama-resmi-browser-baru-windows-10>. [Accessed 2 1 2021].
- [42] Gurat, 14 01 2021. [Online]. Available: <https://guratgarut.com/contoh-statechart-diagram/>. [Accessed 02 04 2021].