**LAPORAN PEMBUATAN APLIKASI**

**PERANCANGAN DAN PEMBUATAN APLIKASI INVENTARIS GUDANG KASUR BERBASIS *WEB***

Laporan Ini Diajukan Untuk Melengkapi Pembelajaran

BDR Mata Pelajaran Pemrograman *Web* Dinamis

Disusun Oleh Kelompok 11:

|  |  |
| --- | --- |
| **Budi Setyawan** | **Imam Satrio Prakoso** |
| NIS: 1819101433 | NIS: 1819101440 |

REKAYASA PERANGKAT LUNAK



**SEKOLAH MENENGAH KEJURUAN**

**MULTI VOCATIONAL PLATFORM ARS INTERNASIONAL**

Jl. Sekolah Internasional 1-6 Antapani, Bandung

2020

# KATA PENGANTAR

Puji syukur atas kehadirat Allah SWT yang telah melimpahkan rahmat dan hidayah-Nya yang telah memberikan banyak kesempatan, sehingga penulis dapat menyelesaikan laporan pembuatan aplikasi ini.

Dalam penyusunan laporan ini, penulis menyadari bahwa selesainya laporan ini tidak terlepas dari dukungan, serta kerja sama dari berbagai pihak, oleh karena itu penulis ingin menyampaikan ucapan terima kasih antara lain kepada:

1. Allah SWT yang telah memberi kemudahan, kelancaran, dan keselamatan kepada penulis.
2. Orang tua, serta keluarga besar yang telah membiayai dan mendidik serta memberi dorongan kepada penulis dalam penyusunan laporan ini.
3. Drs. Tia Sugiri S.T., M.Pd., selaku kepala sekolah SMK MVP ARS Internasional Bandung.
4. Cahyohartono S.T. selaku Ketua Prodi dan pengajar di SMK MVP ARS Internasional Bandung.
5. Aang Miftah Parid S.T., MM., selaku wali kelas XII RPL di SMK MVP ARS Internasional.
6. Seluruh Teman-teman yang telah bekerja sama dan saling membantu satu sama lain.

Penulis menyadari bahwa penyusunan laporan ini masih jauh dari sempurna, untuk itu penulis mengharapkan kritik dan saran yang dapat membangun agar bisa lebih baik lagi.

Bandung, Mei 2020

Kelompok 11

# DAFTAR ISI

[KATA PENGANTAR iv](#_Toc50904796)

[DAFTAR ISI v](#_Toc50904797)

[DAFTAR GAMBAR viii](#_Toc50904798)

[DAFTAR TABEL ix](#_Toc50904799)

[BAB I PENDAHULUAN 10](#_Toc50904800)

[1.1 Latar Belakang 10](#_Toc50904801)

[1.2 Identifikasi Masalah 11](#_Toc50904802)

[1.3 Batasan Masalah 11](#_Toc50904803)

[1.4 Maksud dan Tujuan 11](#_Toc50904804)

[1.4.1 Maksud 11](#_Toc50904819)

[1.4.2 Tujuan 11](#_Toc50904820)

[1.5 Teknik Pengumpulan Data 11](#_Toc50904821)

[1.6 Sistematika Penulisan 12](#_Toc50904824)

[BAB II LANDASAN TEORI 13](#_Toc50904825)

[2.1 Konsep Dasar/teori 13](#_Toc50904829)

[2.1.1 Aplikasi 13](#_Toc50904834)

[2.1.2 *Website* 14](#_Toc50904835)

[2.1.3 *Unified Modeling Language* *(UML)* 14](#_Toc50904836)

[2.1.4 *Use Case Diagram* 15](#_Toc50904837)

[2.1.5 *Activity Diagram* 16](#_Toc50904838)

[2.1.6 *Package Diagram* 17](#_Toc50904839)

[2.1.7 *Sequence Diagram* 18](#_Toc50904840)

[2.1.8 *Class Diagram* 20](#_Toc50904841)

[2.1.9 *Component Diagram* 21](#_Toc50904842)

[2.1.10 *Flowchart* 22](#_Toc50904843)

[2.1.11 *Entity Relationship Diagram* 25](#_Toc50904844)

[2.1.12 *Web Server* 26](#_Toc50904845)

[2.1.13 *XAMPP* 26](#_Toc50904846)

[2.1.14 *PHP* 27](#_Toc50904847)

[2.1.15 *MySQL* 27](#_Toc50904848)

[2.1.16 *HTML* 27](#_Toc50904849)

[2.1.17 *CSS* 27](#_Toc50904850)

[2.1.18 *Bootstrap* 28](#_Toc50904851)

[2.2 Peralatan pendukung (*Tools System*) 28](#_Toc50904852)

[2.2.1 Perangkat Keras 29](#_Toc50904856)

[2.2.2 Perangkat Lunak 29](#_Toc50904857)

[BAB III PEMBAHASAN 30](#_Toc50904861)

[3.1 Prosedur Sistem Berjalan 30](#_Toc50904865)

[BAB IV PERANCANGAN SISTEM 31](#_Toc50904867)

[4.1 Sistem Usulan 31](#_Toc50904871)

[4.2 Prosedur Sistem Usulan 31](#_Toc50904872)

[4.3 Diagram ERD (Entity Relationship diagram) 31](#_Toc50904873)

[4.4 *Object Oriented Analysis and Design (OOAD)* Sistem Usulan 32](#_Toc50904874)

[4.4.1 Sistem Berjalan Inventaris Gudang Kasur 32](#_Toc50904881)

[4.4.2 Proses *Login* dan Registrasi *Admin* 33](#_Toc50904882)

[4.4.3 Proses Fitur *Local Admin* 37](#_Toc50904883)

[4.4.4 Proses Fitur *Super Admin* 38](#_Toc50904884)

[4.5 Spesifikasi Proses Rancangan Sistem Usulan 39](#_Toc50904885)

[4.5.1 Normalisasi *Database* 39](#_Toc50904887)

[4.5.1.1 Normalisasi Transaksi 40](#_Toc50904889)

[4.5.1.2 Normalisasi Barang 41](#_Toc50904890)

[4.5.1.3 Normalisasi Pegawai 42](#_Toc50904893)

[4.5.2 Spesifikasi *Database* 43](#_Toc50904894)

[4.5.2.1 Spesifikasi Normalisasi Transakasi 43](#_Toc50904896)

[4.5.2.2 Spesifikasi Normalisasi Barang 44](#_Toc50904897)

[4.5.2.3 Spesifikasi Normalisasi Pegawai 45](#_Toc50904900)

[4.5.3 Spesifikasi Program 46](#_Toc50904901)

[4.5.3.1 Umum 46](#_Toc50904904)

[4.5.3.2 Perangkat Keras 46](#_Toc50904905)

[4.6 Spesifikasi Sistem Komputer 46](#_Toc50904906)

[4.6.1 Umum 46](#_Toc50904908)

[4.6.2 Perangkat Keras 47](#_Toc50904909)

[4.6.3 Perangkat Lunak 47](#_Toc50904910)

[BAB V PENUTUP 48](#_Toc50904911)

[5.1 Kesimpulan 48](#_Toc50904914)

[DAFTAR PUSTAKA 49](#_Toc50904915)

# 

# DAFTAR GAMBAR

[Gambar 3.1 Flowchart Prosedur Sistem Berjalan 30](#_Toc50904773)

[Gambar 4.1 ERD Diagram Inventaris Gudang Kasur 31](#_Toc50904774)

[Gambar 4.2 Use Case Diagram Inventaris Gudang Kasur 32](#_Toc50904775)

[Gambar 4.3 Package Diagram Smart Restaurant 32](#_Toc50904776)

[Gambar 4.4 Activity Diagram Login Admin 33](#_Toc50904777)

[Gambar 4.5 Sequence Diagram Login Admin 34](#_Toc50904778)

[Gambar 4.6 Activity Diagram Registrasi Admin 35](#_Toc50904779)

[Gambar 4.7 Sequence Diagram Register Admin 36](#_Toc50904780)

[Gambar 4.8 Activity Diagram Fitur Local Admin 37](#_Toc50904781)

[Gambar 4.9 Sequence Diagram Fitur Local Admin 38](#_Toc50904782)

[Gambar 4.10 Sequence Diagram Fitur Super Admin 38](#_Toc50904783)

[Gambar 4.11 Activity Diagram Fitur Super Admin 39](#_Toc50904784)

[Gambar 4.12 Bentuk Tidak Normal Tabel Transaksi 40](#_Toc50904785)

[Gambar 4.13 Bentuk Normal Pertama Tabel Transaksi 40](#_Toc50904786)

[Gambar 4.14 Bentuk Normal Ketiga Tabel Transaksi 40](#_Toc50904787)

[Gambar 4.15 Bentuk Tidak Normal Tabel Barang 41](#_Toc50904788)

[Gambar 4.16 Bentuk Normal Pertama Tabel Barang 41](#_Toc50904789)

[Gambar 4.17 Bentuk Normal Kedua Tabel Barang 41](#_Toc50904790)

[Gambar 4.18 Bentuk Normal Ketiga Tabel Barang 42](#_Toc50904791)

[Gambar 4.19 Bentuk Tidak Normal Tabel Pegawai 42](#_Toc50904792)

[Gambar 4.20 Bentuk Normal Pertama Tabel Pegawai 42](#_Toc50904793)

[Gambar 4.21 Bentuk Normal Kedua Tabel Pegawai 43](#_Toc50904794)

[Gambar 4.22 Bentuk Normal Ketiga Tabel Pegawai 43](#_Toc50904795)

# DAFTAR TABEL

[Tabel 2.1 Simbol Use Case Diagram 15](#_Toc50903801)

[Tabel 2.2 Simbol Activity Diagram 16](#_Toc50903802)

[Tabel 2.3 Simbol Package Diagram 17](#_Toc50903803)

[Tabel 2.4 Simbol Sequence Diagram 18](#_Toc50903804)

[Tabel 2.5 Simbol Class Diagram 20](#_Toc50903805)

[Tabel 2.6 Simbol Component Diagram 22](#_Toc50903806)

[Tabel 2.7 Simbol Flowchart 22](#_Toc50903807)

[Tabel 2.8 Simbol ERD 25](#_Toc50903808)

# 

**BAB** I

# BAB I PENDAHULUAN

## Latar Belakang

*Inventory* (persediaan) atau gudang dalam bidang Teknik Informatika membahas mengenai data persediaan dalam gudang. Perkembangan bisnis di dunia modern tidak akan luput dari bidang Teknik Informatika. Contoh sederhana yang kita temui seperti melakukan proses transaksi secara *online*. Dengan adanya proses transaksi secara *online*, konsumen bisa melakukan pembelian barang dimana saja, kapan saja tidak perlu saling bertemu satu sama lain. Perkembangan teknik informatika di Indonesia terus meningkat. Perusahaan-perusahaan besar di Indonesia menggunakan sistem basis data *inventory*.

Sistem basis data persediaan pada perusahaan membantu *staff* untuk melakukan pendataan, mulai dari pengelompokkan data, *input* data, *update* hingga menghapus data. *Staff* bisa mendata barang yang masuk ke perusahaan, biaya pengeluaran hingga data-data *staff* dalam perusahaan. Dengan demikian data perusahaan bisa diakses secara langsung untuk melihat proses kinerja *karyawan* dan proses keluar masuk barang. Bayangkan apa yang akan terjadi jika perusahaan besar melakukan pendataan hanya menggunakan buku. Data tidak akan akurat dan sulit untuk di akses. Dengan sistem basis data *inventory*, pengguna bisa melihat riwayat data pada hari sebelumnya dan bisa mengontrol secara langsung kegiatan perusahaan.

Dari uraian diatas maka penulis tertarik membuat sistem *inventaris* dengan mengambil judul laporan “PERANCANGAN DAN PEMBUATAN APLIKASIINVENTARISGUDANG KASUR BERBASIS *WEB*”.

## Identifikasi Masalah

Permasalahan yang penulis ajukan ini dapat diidentifikasi pemasalahannya sebagai berikut:

1. Bagaimana membuat perancangan sistem aplikasi inventaris gudang?
2. Bagaimana merancang sebuah *Database* aplikasi inventaris gudang?
3. Bagaimana cara kerja aplikasi inventaris gudang?
4. Bagaimana membuat aplikasi inventaris gudang berbasis *Web*?

## Batasan Masalah

Penulis juga mempunyai sebuah batasan masalah sebagai berikut:

1. Penulis membuat aplikasi berbasis *Web* menggunakan *PHP* *Native* versi 7.
2. Database yang digunakan oleh penulis yaitu *Mysq.l*
3. Aplikasi *Web* *Development* yang digunakan hanya *Sublime.*
4. Penulis tidak merancang/membuat *User Interface*.

## Maksud dan Tujuan



### Maksud

Penulis memiliki maksud untuk mempermudah dan mempercepat alur pendataan dari sebuah gudang.

### Tujuan

Penulis juga mempunyai tujuan untuk:

1. Mengetahui bagaimana membuat rancangan sebuah Aplikasi berbasis *Web*.
2. Memperdalam teknik pembuatan *database*.
3. Mengetahui pembuatan aplikasi berbasis *Web*.

## Teknik Pengumpulan Data



Teknik pengumpulan data yang digunakan penulis dalam membuat laporan ini adalah:

1. Penelitian lapangan, yaitu penelitian yang dilakukan secara langsung untuk mendapatkan informasi yang berhubungan dalam membuat laporan dengan metode yang dilakukan:
   1. Observasi yaitu dengan langsung datang ketempat yang berada objek penelitian atau observasi juga dilakukan dengan mengumpulkan data dengan mencari di *internet*.

## Sistematika Penulisan

Pada bagian ini, penulis ingin memberikan gambaran secara garis besar sistematika isi laporan ini, yang bertujuan untuk mempermudah pembaca dalam memahami isi laporan. Laporan ini terdiri dari beberapa bab, terdiri dari:

BAB I PENDAHULUAN

Bab I berisi tentang latar belakang, identifikasi masalah, batasan masalah, maksud dan tujuan, teknik pengumpulan data, dan lokasi praktek kerja industri.

BAB II LANDASAN TEORI

Bab II berisi tenntang Landasan teori, pengertian aplikasi, pengertian dan penjelasan materi *Website*, pengertian *UML*, Pengertian *Use* *Case* *Diagram*, *Activity* *Diagram*, *Package* *Diagram*, *Sequence* *Diagram*, *Class* *Diagram*, *Component* *Diagram*, *Web* *Server*, *Flowchart*, *PHP*, *XAMPP*, dan *Mysql*.

BAB III PEMBAHASAN

Bab III berisi tentang prosedur sistem berjalan.

BAB VI PERANCANGAN SISTEM

Bab VI berisi tentang sistem usulan, prosedur sistem usulan, *Unified* *Modeling* *Language* (*UML*), *use case* *diagram*, *class* *diagram*, activity *diagram*, *sequence* *diagram*, bentuk dokumen masukan, bentuk dokumen keluaran, spesifikasi file, spesifikasi *design*, dan spesifikasi sistem komputer.

BAB V PENUTUP

Bab V berisi tentang simpulan dan saran dalam proses pebuatan laporan ini.

**BAB II**

# BAB II LANDASAN TEORI



## Konsep Dasar/teori

Sebelum penulis membuat suatu aplikasi maka penulis terlebih dahulu akan menjabarkan tentang definisi, arti dan maksud dari judul yang penulis pilih.



### Aplikasi

Aplikasi dapat diartikan sebagai suatu program berbentuk perangkat lunak yang berjalan pada suatu sistem tertentu yang berguna untuk membantu berbagai kegiatan yang dilakukan oleh manusia. Selain pengertian di atas, ada banyak pengertian dari kata ‘Aplikasi’ yang dikemukakan oleh para ahli.

(bersumber dari <https://pengertiandefinisi.com/pengertian-aplikasi-dan-sejarah-perkembangan-aplikasi/>)

Menurut Syamsu Rizal, Eko Retnadi dan Andri Ikhwana (Vol. 10 No. 1 2013) “aplikasi adalah penggunaan dalam suatu perangkat komputer, instruksi (*instruction*) atau pernyataan (*statement*) yang disusun hingga sedemikian rupa komputer dapat memproses masukan (*input*) menjadi keluaran (*output*)”.

Menurut Joni Supriyono Arif Pramadya (2013)“perangkat lunak aplikasi yaitu perangkat lunak yang digunakan untuk membantu pemakai komputer untuk melaksanakan pekerjaannya”.

Dari beberapa pendapat diatas, dapat disimpulkan bahwa aplikasi adalah sebuah program atau perangkat lunak yang dirancang atau dibuat untuk tujuan tertentu dengan melakukan aktifitas tertentu melalui proses dan prosedur aliran data dalam infrastruktur teknologi informasi yang sesuai dengan jenjang dan kebutuhan.

(bersumber dari <http://materipasti.blogspot.co.id/2016/09/pengertian-aplikasi-menurut-para-ahli.html>).

### *Website*

*Website* adalah kumpulan informasi/kumpulan *page* yang biasa diakses lewat jalur *internet*. Setiap orang di berbagai tempat dan segala waktu bisa menggunakannya selama terhubung secara *online* di jaringan *internet*. Secara teknis, *website* adalah kumpulan dari *page*, yang tergabung kedalam suatu *domain* atau *subdomain* tertentu. *Website*-*website* yang ada berada di dalam *World* *Wide* *Web (WWW) Internet*.

(bersumber dari <https://www.dewaweb.com/blog/pengertian-website/>)

Menurut Hakim Lukmanul (2004) *Website* adalah fasilitas *internet* penghubung dokumen dalam lingkup lokal maupun jarak jauh. Dokumen pada *website* disebut dengan *web* *page* sementara *link* dalam *website* memungkinkan pengguna bisa berpindah dari satu page ke *page* lain *(hyper text),* baik diantara *page* yang disimpan dalam *server* yang sama maupun *server* diseluruh dunia. *Pages* diakses dan dibaca lewat *browser* seperti *Netscape* *Navigator*, *Internet* *Explorer*, *Mozila* *Firefox*, *Google* *Chrome* dan aplikasi *browser* lainnya.

(bersumber dari <https://www.dewaweb.com/blog/pengertian-website/>)

### *Unified Modeling Language* *(UML)*

Menurut Hend (2006:5) “*Unified Modeling Language (UML)* adalah bahasa yang telah telah menjadi standard untuk visualisasi, menetapkan, membangun dan mendokumentasikan artifak suatu sistem perangkat lunak”.

Menurut Adi Nugroho (2005:3) “*Unified Modeling Language (UML)* adalah alat bantu analis serta perancangan perangkat lunak berbasis objek”.

Menurut Martin Fowler (2005:1) “*Unified Modeling Language (UML)* adalah keluarga notasi grafis yang didukunng oleh *meta-model* 28 tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek”.

Berdasarkan beberapa pendapat yang dikemukakan dapat ditarik kesimpulan bahwa *Unified Modeling Language (UML)* adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak.

(bersumber dari <http://www.kumpulanpengertian.com/2015/04/pengertian-uml-menurut-para-ahli.html>)

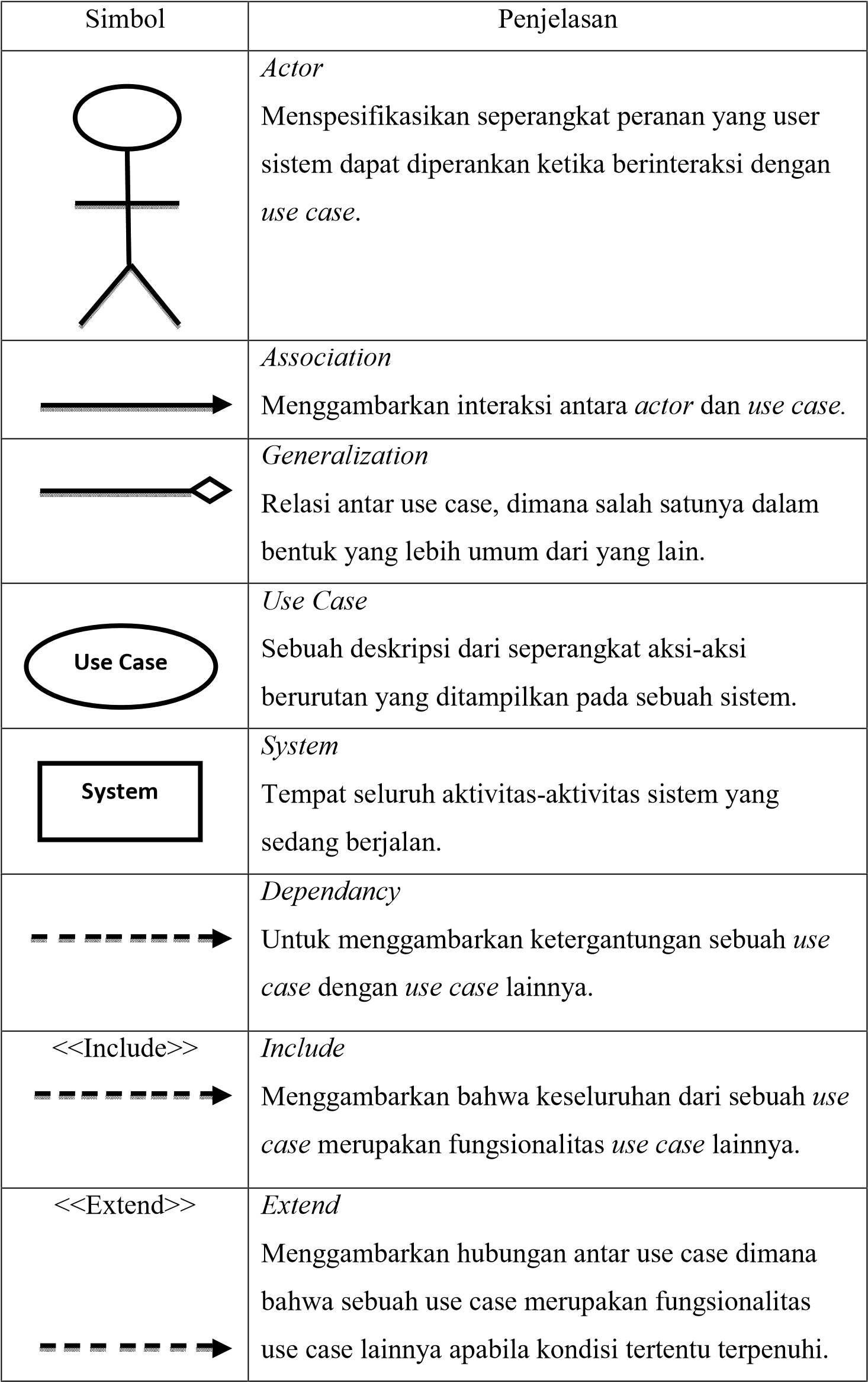
### *Use Case Diagram*

Menurut Satzinger (2011 : 20) “*Use Case Diagram* merupakan rangkaian tindakan yang dilakukan oleh sistem, aktor mewakili user atau sistem lain yang berinteraksi dengan sistem yang dimodelkan”.

Berdasarkan pendapat di atas dapat disimpulkan bahwa *Use case diagram* adalah rangkaian/uraian sekelompok yang saling terkait dan membentuk sistem secara teratur yang dilakukan atau diawasi oleh sebuah aktor.

(bersumber dari <https://library.binus.ac.id/eColls/eThesisdoc/Bab2/2014-2-00823-SI%20Bab2003.pdf>)

Tabel 2.1 Simbol Use Case Diagram



### *Activity Diagram*

Menurut Satzinger (2011 : 23) “Diagram aktivitas adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Diagram aktivitas mempunyai peran seperti halnya *flowchart*, akan tetapi perbedaannya dengan *flowchart* adalah diagram aktivitas bisa mendukung perilaku paralel sedangkan *flowchart* tidak bisa”.

(bersumber dari <https://library.binus.ac.id/eColls/eThesisdoc/Bab2/2014-2-00823-SI%20Bab2003.pdf>)

Tabel 2.2 Simbol Activity Diagram

|  |  |
| --- | --- |
| **Simbol** | **Penjelasan** |
|  | *Initial State*  Mempresentasikan dimulainya alur kerja suatu sistem dalam *activity diagram*. |
|  | *Action State* |
| Action State | Sebuah *state* yang menggambarkan eksekusi dari aksi *atomic*. |
| ► | *Transition Between Activities*  Mengidentifikasikan bahwa suatu objek dari state pertama akan menampilkan aksi-aksi tertentu dan memasuki state kedua ketika peristiwa terjadi pergerakan dari aksi ke aksi lainnya. |
|  | *Decision Point*  Menentukan kapan alur dalam aktivitas menjadi bercabang. |
|  | *Final State*  Mempresentasikan bahwa telah diakhirinya alur suatu sistem dalam *activity diagram.* |

### *Package Diagram*

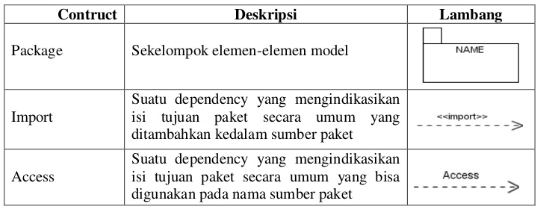
*Package Diagram* menggambarkan dependensi antara paket yang membentuk model. sebuah paket dapat berisi elemen-elemen model yang berlainan, termasuk paket-paket untuk menciptakan/menggambarkan sifat hiraki.

(bersumber dari <https://triutina55.wordpress.com/2019/04/08/pengertian-package-diagram-dan-deployment-diagram/>)

Menurut Satzinger (2012, 353) “*Package diagram* di *UML* hanyalah sebuah diagram tingkat tinggi yang memungkinkan desainer untuk mengasosiasikan kelompok terkait”.

(bersumber dari <https://adoc.tips/bab-2-landasan-teori-21-teori-umum-pengertian-sistem-informa914b4c586529871acbd3217f687d155a9338.html>)

Tabel 2.3 Simbol Package Diagram



### *Sequence Diagram*

Menurut Satzinger (2011 : 26) “*Sequence* adalah satu dari dua interaksi diagram yang mengilustrasikan objek-objek yang berhubungan dengan *use* *case* dan *message* atau pesan-pesannya”.

(bersumber dari <https://library.binus.ac.id/eColls/eThesisdoc/Bab2/2014-2-00823-SI%20Bab2003.pdf>)

Tabel 2.4 Simbol Sequence Diagram

|  |  |
| --- | --- |
| **Simbol** | **Penjelasan** |
| https://3.bp.blogspot.com/-kTP2sefYGJk/W1QEDPEvUEI/AAAAAAAABm4/0_ZLcuynX2oqM88pW0ZjLxKrJWYmC19mQCLcBGAs/s1600/aktor.PNG | Aktor  Aktor merepresentasikan entitas yang berada di luar sistem dan berinteraksi dengan sistem . Mereka bisa berupa manusia , perangkat keras ataupun sistem yang lain. |
| https://1.bp.blogspot.com/-g5fbhRROlhg/W1QEENnRxcI/AAAAAAAABnM/MSafkkITyF89ToYh4Ohf61X_GUnqh5xAACEwYBhgL/s1600/lifeline.PNG | *Lifeline*  Fungsi dari simbol ini adalah Mengeksekusi objek selama *sequence* (*message* dikirim atau diterima dan aktifasinya). |
| https://2.bp.blogspot.com/-NvB3bAWmmMo/W1QED5ylnYI/AAAAAAAABnI/NTtKLTy0p_ksAzizVCPzTObu4Cc17_3yQCEwYBhgL/s1600/general.PNG | *General*  Merepresentasikan entitas tunggal dalam *sequence* *diagram* . Entitas ini memiliki nama , *stereotype* atau berupa *instance* *(class).* |
| https://2.bp.blogspot.com/-mE_TwQ2AFQc/W1QEC_Gp18I/AAAAAAAABm8/cu3rWG7FD6g5YIMNpPUNMU0RW4LxBadqgCEwYBhgL/s1600/boundary.PNG | *Boundary*  *Boundary* biasanya berupa tepi dari sistem , seperti *user interface* atau suatu alat yang berinteraksi dengan sistem yang lain. |
| https://4.bp.blogspot.com/-JskDUjXKCio/W1QEDRcpXlI/AAAAAAAABnA/0Y8e4c3XBEYaJsLZz7adidfqZDmw9WAawCEwYBhgL/s1600/control.PNG | *Control*  *Control* elemen mengatur aliran dari informasi untuk sebuah skenario . Objek ini umumnya mengatur perilaku dan perilaku bisnis. |
| https://2.bp.blogspot.com/-2AeNkppCY-k/W1QED9LKiAI/AAAAAAAABnE/qahfyYGoQdwzO35xEAGPxrhZDWa7tvvFACEwYBhgL/s1600/entity.PNG | *Entity*  Entitas biasanya elemen yang bertanggung jawab menyimpan data atau informasi . Ini dapat berupa *beans* atau *model* *object*. |
| https://3.bp.blogspot.com/-RCb0izxkfjo/W1QECS9z_7I/AAAAAAAABnc/jISu_TlSN9UxGvMwLpUj24Ces4aQkI0qgCEwYBhgL/s1600/activation.PNG | *Activation*  Yaitu suatu titik dimana sebuah objek mulai berpartisipasi di dalam sebuah *sequence* yang menunjukkan kapan sebuah objek mengirim atau menerima objek. |
| **Message** | *Message*  *Message* berfungsi sebagai komunikasi antar objek yang menggambarkan aksi yang akan dilakukan . *Message* terjadi antara dua buah objek dimana satu objek *(client)* dan meminta objek *(supplier)* untuk melakukan sesuatu. |
| https://4.bp.blogspot.com/-3SNDolfsNSo/W1QECAnzuEI/AAAAAAAABnU/4bdtJEfnax88r1pFChMrSEnNqIZiEXxFwCEwYBhgL/s1600/Message%2Bto%2Bentry.PNG | *Message Entry*  Simbol ini berfungsi untuk menggambarkan pesa/hubungan antar objek yang menunjukkan urutan kejadian yang terjadi. |
| https://1.bp.blogspot.com/-2PKb-czxzsw/W1QECmTsICI/AAAAAAAABnc/A9Z4lq4jQhEQD-9urMz6SBzTzcmx8WrHQCEwYBhgL/s1600/Message%2Bretrurn.PNG | *Message to Self*  Simbol ini menggambarkan pesa/hubungan objek itu sendiri , yang menunjukkan urutan kejadian yang terjadi. |
| https://2.bp.blogspot.com/-cbKKfklEeL4/W1QEEaXcpFI/AAAAAAAABng/7jqqihvkqNcHct41fqeScwnU03YkDfOeQCEwYBhgL/s1600/message%2Bto%2Bself.PNG | *Message Return*  Simbol ini menggambarkan hasil dari pengiriman *message* dan digambarkan dengan arah dari kanan ke kiri. |

### *Class Diagram*

Menurut Satzinger (2011:28) “*Class Diagram* adalah deskripsi kelompok obyek-obyek dengan *property*, operasi dan relasi yang sama”.

(bersumber dari <https://library.binus.ac.id/eColls/eThesisdoc/Bab2/2014-2-00823-SI%20Bab2003.pdf>)

Tabel 2.5 Simbol Class Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Keterangan** | **Simbol** |
| ***Class*** | *Class* adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah *class* digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari *class*. Bagian tengah mendefinisikan *property/atribut* *class*. Bagian akhir mendefinisikan *method-method* dari sebuah *class*. | |  | | --- | | Nama *Class* | | + atribut  + atribut  + atribut | | + *method*  + *method* | |
| ***Association*** | Sebuah asosiasi merupakan sebuah *relationship* paling umum antara 2 *class* dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 *class*. Garis ini bisa melambangkan tipe-tipe *relationship* dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah *relationship.* (Contoh: *One-to-one, one-to-many, many-to-many*). |  |
| ***Composition*** | Jika sebuah *class* tidak bisa berdiri sendiri dan harus merupakan bagian dari *class* yang lain, maka *class* tersebut memiliki relasi *Composition* terhadap *class* tempat dia bergantung tersebut. Sebuah *relationship composition* digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid. |  |
| ***Dependency*** | Kadangkala sebuah *class* menggunakan *class* yang lain. Hal ini disebut *dependency.* Umumnya penggunaan *dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* yang lain. Sebuah *dependency* dilambangkan sebagai sebuah panah bertitik-titik. |  |
| ***Aggregation*** | *Aggregation* mengindikasikan keseluruhan bagian *relationship* dan biasanya disebut sebagai relasi |  |

### *Component Diagram*

*Component diagram* menurut Munawar (2005:119) “mempresentasikan

dunia riil item yaitu *component software*”.

(bersumber dari <https://elib.unikom.ac.id/download.php?id=40183>)

Tabel 2.6 Simbol Component Diagram

|  |  |  |
| --- | --- | --- |
| **Nama Komponen** | **Keterangan** | **Simbol** |
| ***Component*** | Sebuah komponen melambangkan sebuah entitas *software* dalam sebuah sistem. Sebuah komponen dinotasikan sebagai sebuah kotak segiempat dengan dua kotak kecil tambahan yang menempel disebelah kirinya. |  |
| ***Dependency*** | Sebuah *Dependency* digunakan untuk menotasikan relasi antara dua komponen. Notasinya adalah tanda panah putusputus yang diarahkan kepada komponen tempat sebuah komponen itu bergantung. |  |

### *Flowchart*

Menurut Indrajani (2011) “*Flowchart* merupakan gambaran secara grafik dari langkah-langkah dan urutan prosedur suatu program”.

(bersumber dari <https://definisimenurutparaahli.blogspot.com/2017/05/5-definisi-flowchart-menurut-para-ahli.html>)

Tabel 2.7 Simbol Flowchart

|  |  |
| --- | --- |
| **Simbol** | **Penjelasan** |
|  | *Arus / flow*  Menyatakan jalannya arus suatu proses. |
|  | *Communication link*  Menyatakan transmisi data dari satu lokasi ke lokasi lain. |
|  | *Connector*  Menyatakan sambungan dari proses ke proses lainnya dalam halaman yang sama. |
|  | *Offline connector*  Sambungan dari proses ke proses lainnya dalam halaman yang berbeda. |
|  | *Process*  Menyatakan suatu tindakan (proses) yang dilakukan oleh komputer. |
|  | *Manual*  Menyatakan suatu tindakan (proses) yang tidak dilakukan oleh komputer. |
|  | *Decision*  Menujukkan suatu kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban : ya / tidak. |
|  | *Predefined process*  Menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberi harga awal. |
|  | *Terminator*  Menyatakan permulaan atau akhir suatu program. |
|  | *Operation*  Menyatakan segal jenis operasi yang diproses dengan menggunakan suatu mesin yang mempunyai *keyboard*. |
|  | *Offline-storage*  Menunjukkan bahwa data dalam simbol ini akan disimpan ke suatu media tertentu. |
|  | *Manual input*  Memasukkan data secara *manual* dengan menggunakan *online* *keyboard*. |
|  | *Input/output*  Menyatakan proses *input* atau *output* tanpa  tergantung jenis peralatannya. |
|  | *Punched card*  Menyatakan *input* berasal dari kartu atau *output* ditulis ke kartu. |
|  | *Magnetic tape*  Menyatakan *input* berasal dari pita magnetis atau *output* disimpan ke pita magnetis. |
|  | *Disk storage*  Menyatakan *input* berasal dari dari *disk* atau *output* disimpan ke *disk*. |
|  | *Document*  Mencetak keluaran dalam bentuk dokumen (melalui *printer*). |
|  | *Display*  Mencetak keluaran dalam layar monitor |

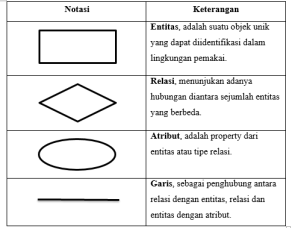
### *Entity Relationship Diagram*

*ERD* merupakan suatu model untuk menjelaskan hubungan data dalam basis data berdasarkan objek-objek dasar data yang mempunyai hubungan antar relasi.

Menurut Brady dan Loonam (2010) *Entity Relationship diagram (ERD)* merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analyst* dalam tahap analisis persyaratan proyek pengembangan system. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain *database* relasional yang mendasari sistem informasi yang dikembangkan. *ERD* bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk *database*.

(bersumber dari <https://rahmatagusblog.wordpress.com/2018/11/11/pengertian-entity-relationship-diagram/>)

Tabel 2.8 Simbol ERD



### *Web Server*

Secara umum *web server* dapat diartikan sebagai sebuah perangkat lunak atau *software* yang memberikan pelayanan berbasis data yang digunakan untuk menerima permintaan atau *request* dari pengguna *internet* atau biasa disebut *client* berupa *http* atau *https* yang kemudian akan ditampilkan kedalam bentuk halaman *website*.

Menurut Khairil “*web server* adalah perangkat lunak yang menjadi sandaran utama dari *www (world wide web)* yang memroses permintaan dari *user* dan mengirimkannya dalam sebuah tampilan halaman *browser*”.

(bersumber dari <https://idcloudhost.com/pengertian-web-server-beserta-fungsinya>)

### *XAMPP*

Menurut Nugroho (2013:1), “*XAMPP* adalah paket program web lengkap yang dapat Anda pakai untuk belajar pemrograman *web*, khususnya *PHP* dan *MySQL*”.

Menurut Buana (2014:4), “*XAMPP* adalah perangkat lunak *opensource* yang diunggah secara gratis dan bisa dijalankan di semua semua operasi seperti *windows, linux, solaris,* *dan mac*”.

(bersumber dari <http://eprints.polsri.ac.id/1068/3/Bab%202.pdf>)

### *PHP*

Menurut Arief (2011c:43) “*PHP* adalah bahasa *server-side –scripting* yang menyatu dengan *HTML* untuk membuat halaman *Web* yang dinamis”.

(bersumber dari <https://bangpahmi.com/pengertian-bahasa-pemrograman-php-menurut-para-pakar/>)

### *MySQL*

Menurut Arief (2011e:151) “*MySQL (My Structure Query Languange)* adalah salah satu jenis *database server* yang sangat terkenal dan banyak digunakan untuk membangun aplikasi *Web* yang menggunakan *database* sebagai sumber dan pengelolaan datanya”.

(bersumber dari <http://batubenjo.blogspot.com/2018/01/mysql-dikembangkan-oleh-perusahaan.html>)

### *HTML*

Menurut Nugroho (2006c:48) ”*HTML* adalah bahasa pemfomatan teks untuk dokumen-dokumen pada jaringan komputer yang sering disebut sebagai *world wide web*”.

Menurut Suyanto (2007:83) ”*HTML* itu adalah bahasa yang digunakan untuk menulis halaman *Web*, biasanya menggunakan extensi .*htm*, .*html* atau .*shtml*”.

(bersumber dari <https://bangpahmi.com/pengertian-html-menurut-para-ahli-dan-pakar/>)

### *CSS*

*CSS (Cascading Style Sheets)* adalah bahasa desain *Web* yang memungkinkan Anda untuk mengatur serta mengontrol tampilan *Web* Anda yang sebelumnya dibuat dengan bahasa *mark up.*

(bersumber dari <https://bootup.ai/blog/css-cascading-style-sheets-adalah-pengertian-belajar-desain-web/#more-306>)

Menurut Jayan (2010:2) mengemukakan bahwa “*CSS* merupakan singkatan dari *Cascading Style Sheet*. Kegunaannya adalah untuk mengatur tampilan dokumen *HTML*, contohnya seperti pengaturan jarak antar baris, teks, warna dan format *border* bahkan penampilan *file* gambar”.

(bersumber dari <https://aguslilinkecil.wordpress.com/tag/landasan-teori/page/2/>)

### *Bootstrap*

*Bootsrap* merupakan *Framework* ataupun *Tools* untuk membuat aplikasi *Web* ataupun situs *Web* *responsive* secara cepat, mudah dan gratis.

(bersumber dari <https://elib.unikom.ac.id/download.php?id=256496>)

Menurut Khadafi (2015:9) “*Bootstrap* adalah sebuah alat bantu untuk membuat sebuah tampilan halaman *website* yang dapat mempercepat pekerjaan seorang pengembang *website* atau pendesain halaman *website*”.

Menurut Fauzi (2008:14) “*Bootstrap* adalah suatu metode berbasis komputer yang sangat potensial untuk dipergunakan pada masalah ketidakstabilan dan keakurasian, khususnya dalam menentukan interval konfendesi”.

(bersumber dari <https://docplayer.info/78430080-Bab-ii-landasan-teori.html>)

Dari beberapa pendapat diatas dapat disimpulkan bahwa Bootstrap merupakan salah satu *framework* untuk pengembangan sebuah aplikasi *website* yang menarik dan responsif. *Bootstrap* merupakan sebuah alat bantu yang dapat mempercepat para pendesain membuat sebuah tampilan *website*.

## Peralatan pendukung (*Tools System*)

Untuk memnuat sebuah aplikasi “*Smart Restaurant*” berbasis *Android* dan *Web* penulis menggunakan peralatan sebagai berikut:



### Perangkat Keras

Perangkat keras yang penulis gunakan untuk pembuatan aplikasi “*Smart Restaurant*” berbasis *Android* dan *Web* yaitu:

1. Komputer/laptop
2. Koneksi *Internet*

### Perangkat Lunak

Sedangkat perangkat lunak yang penulis gunakan yaitu:

1. *Web* *Browser*
2. *Edraw* *Max*
3. *Sublime* *Text* *3*
4. *XAMPP*

**BAB II****I**





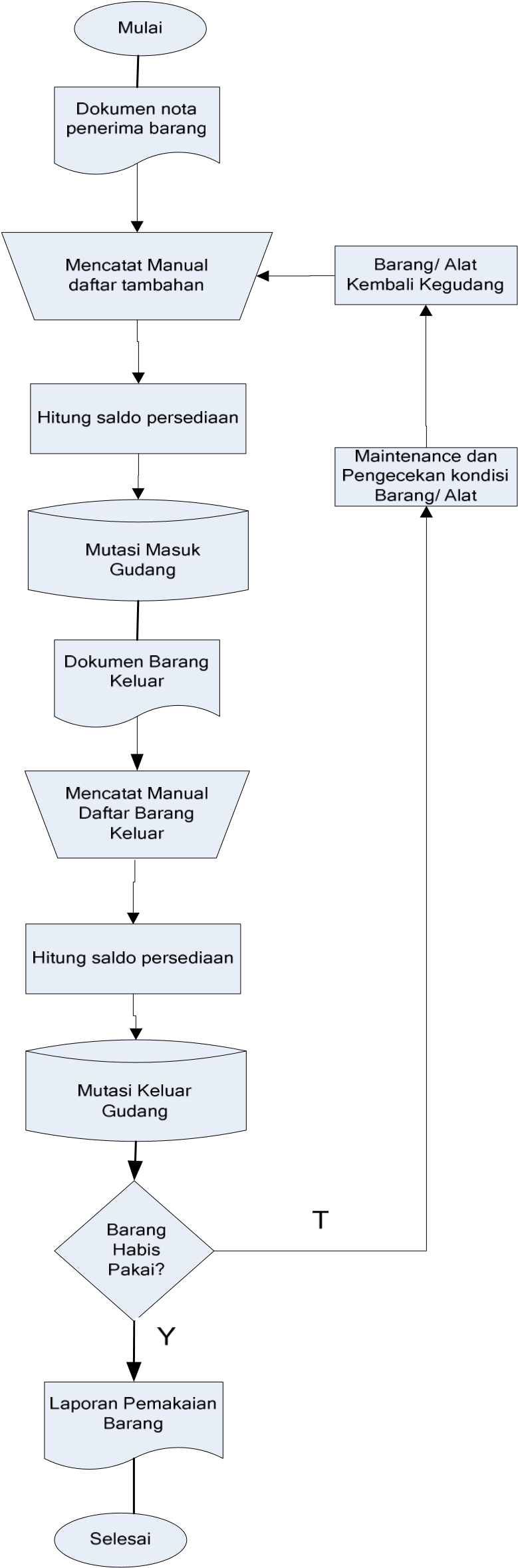


# BAB III PEMBAHASAN



## Prosedur Sistem Berjalan

Dalam pendataan pengelolaan barang inventori masih dilakukan secara manual, yakni mencatat tanggal dan hari masuknya barang serta jumlah barang yang ada digudang, pencatatan semua barang yang masuk dan keluar dicatat dikertas dan disimpan kedalam map, sebagai arsip. Berikut adalah flowchart dari analisa sistem berjalan dibagian gudang inventori sebagai berikut:



Gambar 3.1 Flowchart Prosedur Sistem Berjalan



**BAB IV**

# BAB IV PERANCANGAN SISTEM



## Sistem Usulan

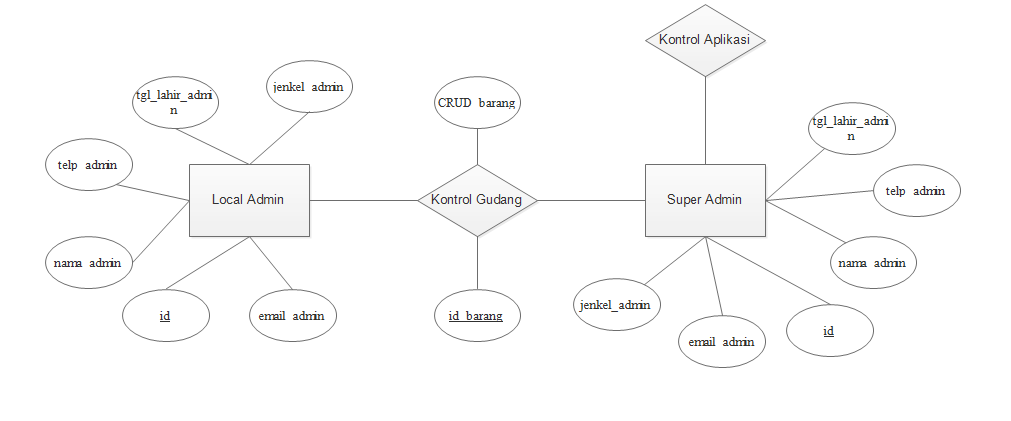
Sistem yang digunakan untuk aplikasi inventaris gudang kasur ini merupakan sistem yang menurut penulis cukup cepat dan efisien jika digunakan sesuai fungsinya. Sistem yang akan penulis buat pun sudah penulis diskusikan dengan tim penulis.

## Prosedur Sistem Usulan

Prosedur sistem usulan untuk aplikasi inventaris gudang kasur dimulai dari membagi *role*. *Role* dibagi menjadi 2 *role*, yaitu *Super Admin* dan *Local Admin*. Masing-masing pengelola/*admin* mempunyai fitur dan tugas yang berbeda-beda diantaranya, *Local Admin* bertugas untuk mengelola seluruh kegiatan inventaris dalam gudang kasur. *Super Admin* dapat melakukan tugas milik *Local Admin* tanpa batasan tertentu dan juga dapat mengontrol kegiatan seluruh aplikasi.

## Diagram ERD (Entity Relationship diagram)

Berdasarkan prosedur sistem usulan yang sebelum nya penulis jelaskan, prosedur tersebut dapat dijabarkan menjadi *ERD* *Diagram* sebagai berikut:



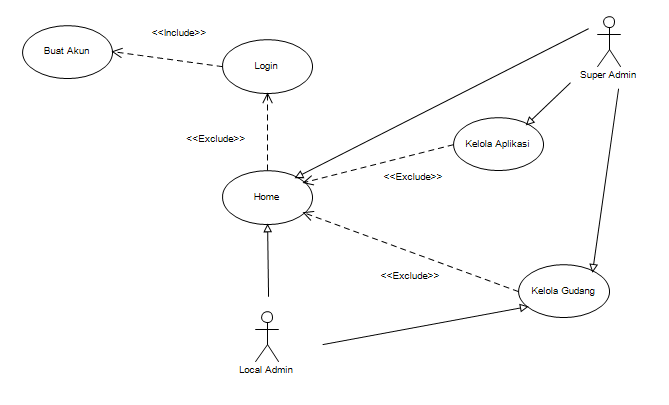
Gambar 4.1 ERD Diagram Inventaris Gudang Kasur

## *Object Oriented Analysis and Design (OOAD)* Sistem Usulan

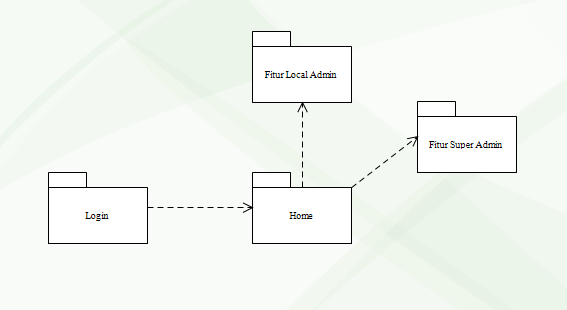
Berdasarkan prosedur sistem usulan yang sebelum nya penulis jelaskan, prosedur tersebut dapat dijabarkan menjadi beberapa diagram *UML* diantaranya sebagai berikut:



### Sistem Berjalan Inventaris Gudang Kasur



Gambar 4.2 Use Case Diagram Inventaris Gudang Kasur



Gambar 4.3 Package Diagram Smart Restaurant

### Proses *Login* dan Registrasi *Admin*



Gambar 4.4 Activity Diagram Login Admin



Gambar 4.5 Sequence Diagram Login Admin



Gambar 4.6 Activity Diagram Registrasi Admin



Gambar 4.7 Sequence Diagram Register Admin

### Proses Fitur *Local Admin*



Gambar 4.8 Activity Diagram Fitur Local Admin



Gambar 4.9 Sequence Diagram Fitur Local Admin

### Proses Fitur *Super Admin*



Gambar 4.10 Sequence Diagram Fitur Super Admin



Gambar 4.11 Activity Diagram Fitur Super Admin

## Spesifikasi Proses Rancangan Sistem Usulan



### Normalisasi *Database*

Normalisasi merupakan peralatan yang digunakan untuk melakukan proses pengelompokkan data menjadi tabel-tabel yang menunjukkan entitas dan

relasinya.

Dalam proses normalisasi, persyaratan sebuah tabel masih harus dipecah didasarkan adanya kesulitan kondisi pengorganisasian data seperti untuk menambah atau menyisipkan, menghapus atau mengubah, serta pembacaan data dari tabel tersebut. Bila masih ada kesulitan, maka tabel harus dipecah menjadi beberapa lagi, dan dilakukan proses normalisasi kembali sampai diperoleh tabel yang optimal.

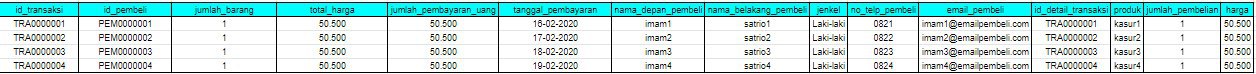
Secara umum proses normalisasi dibagi dalam tiga tahap, yaitu tahap tidak normal (*Unnormal*), normalisasi tahap 1, normalisasi tahap 2 dan normalisasi tahap 3. pada tahap yang ketiga biasanya sudah akan diperoleh tabel yang optimal.

Adapun bentuk normalisasi dari aplikasi inventaris gudang kasur adalah sebagai berikut:



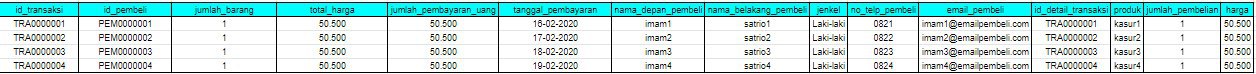
#### Normalisasi Transaksi

1. **Bentuk Tidak Normal**

****

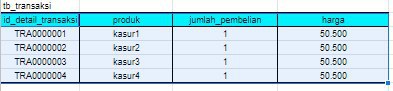
Gambar 4.12 Bentuk Tidak Normal Tabel Transaksi

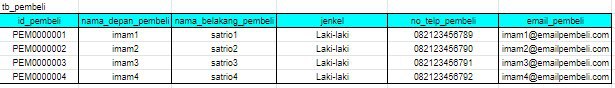
1. **Bentuk Normal Pertama (1st NF)**

****

Gambar 4.13 Bentuk Normal Pertama Tabel Transaksi

1. **Bentuk Normal Kedua (2nd NF)**

****

****

****

Gambar 4.14 Bentuk Normal Ketiga Tabel Transaksi

1. **Bentuk Normal Ketiga (3rd NF)**

**-**

#### Normalisasi Barang

1. **Bentuk Tidak Normal**





Gambar 4.15 Bentuk Tidak Normal Tabel Barang

1. **Bentuk Normal Pertama (1st NF)**





Gambar 4.16 Bentuk Normal Pertama Tabel Barang

1. **Bentuk Normal Kedua (2nd NF)**

tb\_produk



Tb\_kategori



Tb\_supplier



Gambar 4.17 Bentuk Normal Kedua Tabel Barang

1. **Bentuk Normal Ketiga (3rd NF)**

Tb\_produk



Tb\_kategori



Tb\_supplier



Gambar 4.18 Bentuk Normal Ketiga Tabel Barang



#### Normalisasi Pegawai

1. **Bentuk Tidak Normal**





Gambar 4.19 Bentuk Tidak Normal Tabel Pegawai

1. **Bentuk Normal Pertama (1st NF)**





Gambar 4.20 Bentuk Normal Pertama Tabel Pegawai

1. **Bentuk Normal Kedua (2nd NF)**

Tb\_pegawai



Tb\_login



Tb\_kode\_perusahaan



Gambar 4.21 Bentuk Normal Kedua Tabel Pegawai

1. **Bentuk Normal Ketiga (3rd NF)**

Tb\_pegawai



Tb\_login



Tb\_kode\_perusahaan



Gambar 4.22 Bentuk Normal Ketiga Tabel Pegawai

### Spesifikasi *Database*



#### Spesifikasi Normalisasi Transakasi

1. **Bentuk Tidak Normal**

{ id\_transaksi, id\_pembeli, jumlah\_barang, total\_harga, jumlah\_pembayaran\_uang, tanggal\_pembayaran, nama\_depan\_pembeli, nama\_belakang\_pembeli, jenkel, no\_telp\_pembeli, email\_pembeli, id\_detail\_transaksi, produk, jumlah\_pembelian, harga }

1. **Bentuk Normal Pertama (1st NF)**

{ id\_transaksi, id\_pembeli, jumlah\_barang, total\_harga, jumlah\_pembayaran\_uang, tanggal\_pembayaran, nama\_depan\_pembeli, nama\_belakang\_pembeli, jenkel, no\_telp\_pembeli, email\_pembeli, id\_detail\_transaksi, produk, jumlah\_pembelian, harga }

1. **Bentuk Normal Kedua (2nd NF)**

**tb\_transaksi =** { id\_transaksi, jumlah\_barang, total\_harga, jumlah\_pembayaran\_uang, tanggal\_pembayaran }

**tb\_pembeli** = {id\_pembeli, nama\_depan\_pembeli, nama\_belakang\_pembeli, jenkel, no\_telp\_pembeli, email\_pembeli}

**tb\_transaksi** = {id\_detail\_transaksi, produk, jumlah\_pembelian, harga}

1. **Bentuk Normal Ketiga (3rd NF)**

-

#### Spesifikasi Normalisasi Barang

1. **Bentuk Tidak Normal**

{ id\_produk, id\_supplier, id\_kategori, kode\_produk, nama\_produk, deskripsi\_produk, jumlah\_produk, harga\_produk, tanggal\_persediaan nama\_kategori , nama\_perusahaan, kota\_perusahaan , no\_telp\_perusahaan, email\_perusahaan, provinsi\_perusahaan }

1. **Bentuk Normal Pertama (1st NF)**

{ id\_produk, id\_supplier, id\_kategori, kode\_produk, nama\_produk, deskripsi\_produk, jumlah\_produk, harga\_produk, tanggal\_persediaan nama\_kategori , nama\_perusahaan, lokasi\_perusahaan, no\_telp\_perusahaan, email\_perusahaan }

1. **Bentuk Normal Kedua (2nd NF)**

**tb\_produk** = { id\_produk, id\_supplier, id\_kategori, kode\_produk, nama\_produk, deskripsi\_produk, jumlah\_produk, harga\_produk tanggal\_persediaan }

**tb\_kategori** = { nama\_kategori }

**tb\_supplier** = { nama\_perusahaan, lokasi\_perusahaan, no\_telp\_perusahaan, email\_perusahaan }

1. **Bentuk Normal Ketiga (3rd NF)**

**tb\_produk** = { id\_produk, id\_supplier, id\_kategori, kode\_produk, nama\_produk, deskripsi\_produk, jumlah\_produk, harga\_produk tanggal\_persediaan }

**tb\_kategori** = { id\_kategori, nama\_kategori }

**tb\_supplier** = { id\_supplier, nama\_perusahaan, lokasi\_perusahaan, no\_telp\_perusahaan, email\_perusahaan }



#### Spesifikasi Normalisasi Pegawai

1. **Bentuk Tidak Normal**

{ id\_pegawai, nama\_depan\_pegawai, nama\_belakang\_pegawai , email\_pegawai, no\_telepon\_pegawai, jenis\_kelamin\_pegawai, tanggal\_rekrut, id\_login, id\_kode, password, status, kode\_perusahaan }

1. **Bentuk Normal Pertama (1st NF)**

{ id\_pegawai, nama\_lengkap , email\_pegawai, no\_telepon\_pegawai, jenis\_kelamin\_pegawai, tanggal\_rekrut, id\_login, id\_kode, password, status, kode\_perusahaan }

1. **Bentuk Normal Kedua (2nd NF)**

**tb\_pegawai** = { id\_pegawai, nama\_lengkap , email\_pegawai, no\_telepon\_pegawai, jenis\_kelamin\_pegawai, tanggal\_rekrut }

**tb\_login** = { id\_login, id\_kode, password, status }

**tb\_kode\_perusahaan** = { kode\_perusahaan }

1. **Bentuk Normal Ketiga (3rd NF)**

**tb\_pegawai** = { id\_pegawai, nama\_lengkap , email\_pegawai, no\_telepon\_pegawai, jenis\_kelamin\_pegawai, tanggal\_rekrut }

**tb\_login** = { id\_pegawai, id\_login, id\_kode, password, status }

**tb\_kode\_perusahaan** = { id\_kode, kode\_perusahaan }

### Spesifikasi Program



#### Umum

Secara keseluruhan, untuk dapat menggunakan aplikasi Inventaris Gudang Sepatu diperlukan spesifikasi khusus yang dapat menunjuang penggunaan aplikasi tersebut

#### Perangkat Keras

Untuk menggunakan aplikasi tersebut, diperlukan sebuah perangkat *Android/Smartphone* yang mendukung dengan minimal sepesifikasi sebagai berikut:

* *Android* versi 4.4 “*KitKat*”
* Memori *Internal*/*ROM* sebesar 8 *GB*
* *RAM* sebesar 512 *MB*

## Spesifikasi Sistem Komputer



### Umum

Secara garis besar, untuk mendukung berjalannya pembuatan aplikasi Inventaris Gudang Kasur diperlukan tambahan dari perangkat pendukung yang dapat mempermudah dalam pembuatan aplikasi tersebut.

### Perangkat Keras

Untuk mengimplementasikan rancangan dan pembuatan aplikasi Inventaris Gudang Kasur, diperlukan laptop atau komputer yang mendukung dengan minimal spesifikasi sebagai berikut:

* *OS Microsoft® Windows® 8/10 (64-bit)*
* *Intel core i3-2330 CPU @ 2.20GHz (4 CPUs),2.20GHz*
* 2 *GB RAM*
* 3 *GB* Penyimpanan *HardDisk*
* Resolusi Layar 1280 x 800

### Perangkat Lunak

Perangkat lunak yang dapat menunjang pembuatan aplikasi Inventaris Gudang Sepatu, dapat diurutkan sebagai berikut:

* *Sublime Text 3*
* *XAMPP*
* *OS Microsoft® Windows® 8/10 (64-bit)*

**BAB V**

# BAB V PENUTUP



## Kesimpulan

Penggunaan *Web Browser* kini sudah ada dimana-mana, yang pastinya dapat digunakan dalam berbagai bidang, salah satunya aplikasi Inventaris Gudang Kasur, dimana bagian gudang dapat mudah mengelola barang melalui *Smartphone* mereka, dirumah atau didalam gudang. Untuk itu alangkah baiknya teknologi *Web Browser* dapat dikembangkan menjadi banyak hal yang berguna bagi kehidupan manusia.

# DAFTAR PUSTAKA

Adzikra Ibrahim, (2015), Pengertian Aplikasi dan Sejarah Perkembangan Aplikasi, https://pengertiandefinisi.com/pengertian-aplikasi-dan-sejarah-perkembangan-aplikasi/ (15/3/ 2020).

Dian Hardianto, (2016), Pengertian Aplikasi Menurut Para Ahli dengan Sumbernya, https://materipasti.blogspot.com/2016/09/pengertian-aplikasi-menurut-para-ahli.html (15/3/ 2020).

Rifqi Mulyawan, (2019), Pengertian Android: Menurut Para Ahli, Sejarah, Manfaat dan jenisnya, https://rifqimulyawan.com/blog/pengertian-android/ (15/3/ 2020).

PT Dewaweb, (2018), Pengertian Website: Panduan Lengkap Soal Website https://www.dewaweb.com/blog/pengertian-website/ (15/3/ 2020).

Andi, (2015), Pengertian UML Menurut Para Ahli, http://www.kumpulanpengertian.com/2015/04/pengertian-uml-menurut-para-ahli.html (15/3/ 2020).

BINUS Library, (2014), bab ii landasan teori, https://library.binus.ac.id/eColls/eThesisdoc/Bab2/2014-2-00823-SI%20Bab2003.pdf (15/3/ 2020).

Tri Septiansyah, (2019), Pengerian Package Diagram dan Deployment Diagram, https://triutina55.wordpress.com/2019/04/08/pengertian-package-diagram-dan-deployment-diagram/ (15/3/ 2020).

Jujun Juhar, (2018), BAB II LANDASAN TEORI, https://triutina55.wordpress.com/2019/04/08/pengertian-package-diagram-dan-deployment-diagram/ (15/3/ 2020).

Ryusnita, (2017), 5 Definisi Flowchart Menurut Para Ahli https://definisimenurutparaahli.blogspot.com/2017/05/5-definisi-flowchart-menurut-para-ahli.html (15/3/ 2020).

Rahmat Agus Sandika, (2018), Pengertian Entity Relationship Diagram https://rahmatagusblog.wordpress.com/2018/11/11/pengertian-entity-relationship-diagram/ (15/3/ 2020).

PT Cloud Hosting Indonesia, (2019), Pengertian Web Server Beserta Fungsinya, https://idcloudhost.com/pengertian-web-server-beserta-fungsinya (15/3/ 2020).

Y Fadilla Samawi, (2014), BAB II TINJAUAN PUSTAKA 2.1 Teori Umum, http://eprints.polsri.ac.id/1068/3/Bab%202.pdf (15/3/ 2020).

Pahmi Ritonga, (2015), Pengertian Bahasa Pemrograman PHP menurut Para Ahli & Pakar, https://bangpahmi.com/pengertian-bahasa-pemrograman-php-menurut-para-pakar/ (15/3/ 2020).

Dwi Kariawan, (2018), Pengertian MySQL Menurut Para Ahli, https://batubenjo.blogspot.com/2018/01/mysql-dikembangkan-oleh-perusahaan.html (15/3/ 2020).

D Hidayah, (2017), BAB II TINJAUAN PUSTAKA, https://batubenjo.blogspot.com/2018/01/mysql-dikembangkan-oleh-perusahaan.html (15/3/ 2020).

Pahmi Ritonga, (2016), Pengertian HTML Menurut Para Ahli dan Pakar, https://bangpahmi.com/pengertian-html-menurut-para-ahli-dan-pakar/ (16/3/ 2020).

Boot up academy ai, (2019), CSS (Cascading Style Sheets) Adalah? Pengertian – Belajar Desain Web https://bootup.ai/blog/css-cascading-style-sheets-adalah-pengertian-belajar-desain-web/#more-306 (16/3/ 20