



**PERANCANGAN SISTEM KEAMANAN DISTRIBUSI
BARANG PADA PT. CITRA BESTARI MENGGUNAKAN
ALGORITMA MD5**

Disusun dan Diajukan Sebagai Salah Satu Persyaratan Untuk Memenuhi Ujian Akhir Memperoleh
Gelar Sarjana Komputer Pada Fakultas Sains dan Teknologi
Universitas Pembangunan Panca Budi
Medan

SKRIPSI

OLEH

**NAMA : YUSTINUS FIRNANDA
N.P.M : 1514370807
PROGRAM STUDI : SISTEM KOMPUTER**

**PROGRAM STUDI SISTEM KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS PEMBANGUNAN PANCA BUDI
2019**

ABSTRAK

PERANCANGAN SISTEM KEAMANAN DISTRIBUSI BARANG PADA PT. CITRA BESTARI MENGGUNAKAN ALGORITMA MD5

OLEH:

YUSTINUS FIRNANDA

1514370807

Distribusi barang merupakan suatu kegiatan yang utama didalam industri yang bergerak di bidang peralatan kantor PT. Citra Bestari, di dalam pendistribusian barang belum terkomputerisasi sehingga menimbulkan beberapa masalah seperti pengarsipan data barang, pemesanan dan pengiriman barang. Melalui penelitian ini, penulis dalam usaha merancang dan membangun sebuah Sistem Informasi distribusi barang berbasis komputer yang akan menciptakan tidak hanya informasi yang akurat, cepat dan relevan tetapi juga dapat mengatasi masalah-masalah dalam pengarsipan. Dalam melakukan penelitian ini Algoritma pengembangan sistem yang digunakan adalah Algoritma MD5 dimana alat yang digunakan untuk perancangan basis data adalah class diagram, table relasi, normalisasi. Untuk perencanaan Algoritma terstruktur menggunakan alat pengembangan sistem *UML*, *Diagram Activity*, *Sequence Diagram*, sedangkan implementasi dan perangkat lunak yang mendukung menggunakan bahasa pemrograman *PHP* dengan database *MySQL*. Dengan dikembangkan website mengenai sistem informasi pendistribusian barang ini diharapkan dapat membantu pelanggan dalam melakukan proses pemesanan dan juga membantu pihak PT. Citra Bestari dalam memperluas pemesanannya.

Kata Kunci : Pendistribusian, Sistem informasi , *Website*

DAFTAR ISI

	Halaman
COVER	
LEMBAR PENGESAHAN	
ABSTRAK	i
KATA PENGANTAR	ii
DAFTAR ISI	iii
DAFTAR GAMBAR	iv
DAFTAR TABEL	v
DAFTAR LAMPIRAN	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	4
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II LANDASAN TEORI	5
2.1 Keamanan Data	5
2.2 Kriptografi	6
2.3 Macam-Macam Kriptografi.....	7
2.4 Pengertian MD5	8
2.5 Algoritma.....	12
2.6 Pengertian UML.....	14
2.7 Pengertian Informasi.....	21
2.9 Tabel ASCII.....	22
BAB III METODE PENELITIAN	45
3.1 Tahapan Penelitian	45
3.2 Algoritma Pengumpulan Data.....	46
3.3 Analisis Sistem Sedang Berjalan.....	46
3.4 Rancangan Penelitian	47
3.5 Desain Sistem Secara Global	49
3.6 Desain Sistem Secara Detail	62
BAB IV HASIL DAN PEMBAHASAN	63
4.1 Implementasi Sistem	63
4.2 Komponen Utama Dalam Sistem	64
4.3 Tampilan Website	65
BAB V PENUTUP	69
5.1 Kesimpulan.....	69
5.2 Saran.....	69
DAFTAR PUSTAKA	
BIOGRAFI PENULIS	
LAMPIRAN-LAMPIRAN	

DAFTAR GAMBAR

No	Judul	Hal
3.1.	Tahapan Penelitian.....	45
3.2.	<i>Flowmap</i> Usulan.....	47
3.3.	<i>Use Case</i> Diagram.....	48
3.4.	<i>Class</i> Diagram.....	49
3.5.	<i>Activity</i> Diagram <i>Login</i>	50
3.6.	<i>Activity</i> Diagram Data Barang.....	51
3.7.	<i>Activity</i> Diagram Data MD5.....	52
3.8.	<i>Activity</i> Diagram <i>Log Out</i>	53
3.9.	<i>Sequence</i> Diagram <i>Login</i>	54
3.10.	<i>Sequence</i> Diagram Data Barang.....	55
3.11.	<i>Sequence</i> Diagram Data MD5.....	56
3.12.	<i>Sequence</i> Diagram Hasil.....	57
3.13.	<i>Sequence</i> Diagram <i>Log Out</i>	58
3.14.	Tampilan Halaman <i>Login</i>	60
3.15.	Tampilan Halaman <i>Home</i>	61
3.16.	Tampilan Menu <i>Entry Data</i>	61
3.17.	Tampilan Menu <i>Entry Member</i>	62
3.18.	Tampilan Menu <i>View Data</i>	62
4.19.	Halaman <i>Login Admin</i>	50
4.20.	Halaman Utama <i>Website</i>	50
4.21.	Halaman Registrasi Data.....	67
4.22.	Halaman <i>Member User</i>	67
4.23.	Halaman <i>View Data</i>	68

DAFTAR TABEL

No	Judul	Hal
2.1.	Simbol <i>Use Case Diagram</i>	15
2.2.	Simbol <i>Activity Diagram</i>	17
2.3.	Tipe Data Numerik.....	29
2.4.	Tipe Data <i>String</i>	30
2.5.	Tipe Data Tanggal.....	30
2.6.	Tabel ASCII.....	34
3.1	Tabel Data Nasabah.....	58
3.2	Tabel Data <i>Admin</i>	59
3.3	Tabel Data <i>Member</i>	59

DAFTAR LAMPIRAN

	Halaman
Lampiran 1 Listing Program.....	L.1
Lampiran 2 Surat Pengajuan Judul Skripsi.....	L.2
Lampiran 3 Assistensi Bimbingan Doping 1 dan 2.....	L.3
Lampiran 4 Form Permohonan Meja Hijau.....	L.4
Lampiran 5 Kartu Bebas Praktikum.....	L.5
Lampiran 6 Hasil Plagiat Cheker.....	L.6

BAB I

PENDAHULUAN

1.1 Latar Belakang

Berkembangnya dunia teknologi saat ini, berperan penting dalam perkembangan teknologi penyimpanan berkas elektronik. Salah satu contoh dari perkembangan tersebut adalah media penyimpanan berkas elektronik secara *online (online storage)*. Media penyimpanan berkas elektronik secara *online* memiliki tujuan yang menguntungkan karena dapat mengatasi masalah - masalah penyimpanan berkas dengan baik.

PT. Citra Bestari adalah perusahaan yang bergerak dibidang distribusi barang ATK (Alat Tulis Kantor). Pada proses distribusi barang ATK (Alat Tulis Kantor) pada PT. Citra bestari ini sering terjadi kesalahan dalam proses pengiriman, seperti (1) Jumlah barang tidak sesuai kwitansi, (2) Kekurangan barang diterima dikarenakan kurir tidak teliti dalam mengecek barang yang akan didistribusikan, (3) Kesalahan dalam pengiriman. Dari kesalahan pengelolaan data distribusi barang ATK (Alat Tulis Kantor) itu maka penulis ingin melakukan pembuatan sistem yang dapat membantu dalam proses distribusi barang ATK (Alat Tulis Kantor) yang menggunakan Algoritma MD5 sebagai Algoritma yang melakukan keamanan terima barang sehingga tidak terjadi kesalahan pada proses distribusi.

Salah satu kekurangan yang dimiliki implementasi tersebut adalah masalah keutuhan/integritas data, dimana data yang pengguna simpan pada media

penyimpanan berkas elektronik ini akan dikelola atau di ketahui oleh penyedia layanan ini. Oleh karena, itu penyedia layanan memanfaatkan algoritma MD5 (*Message–Digest Algorithm 5*) untuk menjamin keutuhan/ integritas data. Berdasarkan latar belakang dari pembahasan diatas penulis tertarik untuk mengambil judul tentang **“Perancangan Sistem Keamanan Distribusi Barang Pada PT. Citra Bestari Menggunakan Algoritma MD5”**.

1.2 Rumusan Masalah

Berdasarkan penjelasan latar belakang diatas, penulis menyimpulkan rumusan masalah pada penelitian ini adalah sebagai berikut :

1. Bagaimana membuat perancangan keamanan menggunakan MD5 pada distribusi barang?
2. Bagaimana mengimplementasikan Algoritma MD5 pada program keamanan barang?
3. Bagaimana membuat keamanan pada proses distribusi barang menggunakan Algoritma MD5?

1.3 Batasan Masalah

Agar tidak terjadi pembahasan yang meluas maka penulis membuat batasan masalah pada penelitian ini adalah sebagai berikut:

1. Sistem penyimpanan berkas elektronik dibangun menggunakan bahasa pemrograman php dan *user interface* menggunakan HTML (*HyperText Markup Language*).

2. Aplikasi yang dibuat dapat diimplementasikan menggunakan web browser (*Chrome*).
3. Keamanan dalam proses penerimaan barang (keamanan) distribusi barang menggunakan Algoritma MD5.

1.4 Tujuan Penelitian

Adapun Tujuan penelitian ini yang berjudul perancangan sistem keamanan distribusi barang pada PT. Citra Bestari menggunakan Algoritma MD5 adalah :

1. Menganalisa dan merancang sistem informasi keamanan menggunakan MD5 pada distribusi barang PT. Citra Bestari.
2. Memanfaatkan hasil penelitian sebagai referensi dasar untuk mengambil langkah kebijakan yang berhubungan dengan keamanan berkas.
3. Meningkatkan keamanan dalam proses penerimaan barang pada PT. Citra Bestari menggunakan Algoritma MD5.

1.5 Manfaat penelitian

Adapun Manfaat penelitian ini yang berjudul perancangan sistem keamanan distribusi barang pada PT. Citra Bestari menggunakan Algoritma MD5 adalah :

1. Sistem yang dibangun dapat memberikan keamanan bagi PT. Citra Bestari dalam mendistribusikan barang.

2. Sistem yang dibangun dapat memberikan jaminan keutuhan / integritas distribusi barang melalui pemanfaatan Algoritma MD5 pada bahasa pemrograman PHP dalam pencocokan nilai *hash*.
3. Memberikan keamanan pada proses pendistribusian barang dikarenakan proses keamanan menggunakan Algoritma MD5.

BAB II

LANDASAN TEORI

2.1 Keamanan Data

Pada zaman teknologi informasi sekarang, data atau informasi merupakan suatu asset yang sangat berharga dan harus dilindungi. Hal ini juga diikuti oleh kemajuan teknologi komputer. Kemajuan teknologi komputer membantu semua aspek kehidupan manusia. Dengan adanya kemajuan dalam teknologi informasi, komunikasi dan komputer maka kemudian muncul masalah baru, yaitu masalah keamanan akan data dan informasi dan dalam hal ini akan membuka peluang bagi orang-orang yang tidak bertanggung jawab untuk menggunakannya sebagai tindak kejahatan. Dan tentunya akan merugikan pihak tertentu. Dalam keamanan data ada beberapa aspek yang berkaitan dengan persyaratan keamanan yaitu (Pabokory, 2015).

1. *Secrecy*. Berhubungan dengan akses membaca data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diakses dan dibaca oleh orang yang berhak.
2. *Integrity*. Berhubungan dengan akses merubah data dan informasi. Data dan informasi di dalam suatu sistem komputer hanya dapat diubah oleh orang yang berhak.
3. *Availability*. Berhubungan dengan ketersediaan data dan informasi. Data dan informasi yang berada dalam suatu sistem komputer tersedia dan dapat dimanfaatkan oleh orang yang berhak.

4. Lebih lanjut menurut Pabokory, terdapat lima langkah keamanan komputer yang baik untuk diperhitungkan yaitu; aset, analisis resiko, perlindungan, alat dan prioritas.

2.2 Pengertian MD5

Dalam kriptografi, MD5 *94bfbfb41eba4e7150261511f4370f65(Message-Digest algoritim 5)* ialah fungsi hash kriptografik yang digunakan secara luas dengan hash value 128-bit. Pada *standart internet* (RFC 1321), MD5 telah dimanfaatkan secara bermacam-macam pada aplikasi keamanan, dan MD5 juga umum digunakan untuk melakukan pengujian integritas sebuah berkas. (Rivest, 2008).

MD5 di desain oleh Ronald Rivest pada tahun 1991 untuk menggantikan *hash function* sebelumnya, MD4. Pada tahun 1996, sebuah kecacatan ditemukan dalam desainnya, walau bukan kelemahan fatal, pengguna kriptografi mulai menganjurkan menggunakan algoritme lain, seperti SHA-1 (klaim terbaru menyatakan bahwa SHA-1 juga cacat). Pada tahun 2004, kecacatan-kecacatan yang lebih serius ditemukan menyebabkan penggunaan algoritme tersebut dalam tujuan untuk keamanan jadi makin dipertanyakan.

Menunjukkan perputaran bit kiri oleh s ; s bervariasi untuk tiap-tiap operasi. tambahan menunjukkan tambahan modulo 232. MD5 memproses variasi panjang pesan kedalam keluaran 128-bit dengan panjang yang tetap. Pesan masukan dipecah menjadi dua gumpalan blok 512-bit; Pesan ditata sehingga panjang pesan dapat dibagi 512. Penataan bekerja sebagai berikut: bit tunggal pertama, 1,

diletakkan pada akhir pesan. Proses ini diikuti dengan serangkaian nol (0) yang diperlukan agar panjang pesan lebih dari 64-bit dan kurang dari kelipatan 512. Bit-bit sisa diisi dengan 64-bit integer untuk menunjukkan panjang pesan yang asli. Sebuah pesan selalu ditata setidaknya dengan 1-bit tunggal, seperti jika panjang pesan adalah kelipatan 512 dikurangi 64-bit untuk informasi panjang ($\text{panjang} \bmod(512) = 448$), sebuah blok baru dari 512-bit ditambahkan dengan 1-bit diikuti dengan 447 bit-bit nol (0) diikuti dengan panjang 64-bit.

Algoritma MD5 yang utama beroperasi pada kondisi 128-bit, dibagi menjadi empat word 32-bit, menunjukkan A, B, C dan D. Operasi tersebut diinisialisasi dijaga untuk tetap konstan. Algoritma utama kemudian beroperasi pada masing-masing blok pesan 512-bit, masing-masing blok melakukan perubahan terhadap kondisi. Pemrosesan blok pesan terdiri atas empat tahap, batasan putaran; tiap putaran membuat 16 operasi serupa berdasar pada fungsi non-linear F, tambahan modular, dan rotasi ke kiri. Gambar satu mengilustrasikan satu operasi dalam putaran. Ada empat macam kemungkinan fungsi F, berbeda dari yang digunakan pada tiap-tiap putaran:

- $$F(X,Y,Z)=(X \wedge Y) \vee (\neg X \wedge Z)$$

$$F(X,Y,Z)=(X \wedge Y) \vee (\neg X \wedge Z)$$
- $$G(X,Y,Z)=(X \wedge Z) \vee (Y \wedge \neg Z)$$

$$G(X,Y,Z)=(X \wedge Z) \vee (Y \wedge \neg Z)$$
- $$H(X,Y,Z)=X \oplus Y \oplus Z$$

$$H(X,Y,Z)=X \oplus Y \oplus Z$$

- $$I(X,Y,Z)=Y \oplus (X \vee \neg \{Z\})$$

$$I(X,Y,Z)=Y \oplus (X \vee \neg \{Z\})$$

$\{\oplus, \wedge, \vee, \neg\}$ menunjukkan operasi logikan XOR, AND, OR dan NOT.

Pseudocode pada algoritme MD5 adalah sebagai berikut.

```
var int[64] r, k
r[0..15] := {7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22, 7, 12, 17, 22}
r[16..31] := {5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20, 5, 9, 14, 20}
r[32..47] := {4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23, 4, 11, 16, 23}
r[48..63] := {6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21, 6, 10, 15, 21}
```

//Menggunakan bagian fraksional biner dari integral sinus sebagai konstanta:

```
for i from 0 to 63
    k[i] := floor(abs(sin(i + 1)) × 2^32)
```

//Inisialisasi variabel:

```
var int h0 := 0x67452301
var int h1 := 0xEFCDAB89
var int h2 := 0x98BADCFE
var int h3 := 0x10325476
```

```
//Pemrosesan awal:  
  
append "1" bit to message  
  
append "0" bits until message length in bits  $\equiv 448 \pmod{512}$   
  
append bit length of message as 64-bit little-endian integer to message  
  
  
//Pengolahan pesan paada kondisi gumpalan 512-bit:  
  
for each 512-bit chunk of message  
  
    break chunk into sixteen 32-bit little-endian words  $w(i)$ ,  $0 \leq i \leq 15$   
  
  
  
//Inisialisasi nilai hash pada gumpalan ini:  
  
var int a := h0  
  
var int b := h1  
  
var int c := h2  
  
var int d := h3  
  
  
  
//Kalang utama:  
  
for i from 0 to 63  
  
    if  $0 \leq i \leq 15$  then  
  
        f := (b and c) or ((not b) and d)  
  
        g := i  
  
    else if  $16 \leq i \leq 31$   
  
        f := (d and b) or ((not d) and c)  
  
        g :=  $(5 \times i + 1) \bmod 16$ 
```

```
else if  $32 \leq i \leq 47$ 
```

```
    f := b xor c xor d
```

```
    g :=  $(3 \times i + 5) \bmod 16$ 
```

```
else if  $48 \leq i \leq 63$ 
```

```
    f := c xor (b or (not d))
```

```
    g :=  $(7 \times i) \bmod 16$ 
```

```
temp := d
```

```
d := c
```

```
c := b
```

```
b := ((a + f + k(i) + w(g)) leftrotate r(i)) + b
```

```
a := temp
```

```
//Tambahkan hash dari gumpalan sebagai hasil:
```

```
h0 := h0 + a
```

```
h1 := h1 + b
```

```
h2 := h2 + c
```

```
h3 := h3 + d
```

```
var int digest := h0 append h1 append h2 append h3 //(diwujudkan dalam little-  
endian)
```


2.3 Algoritma

Penyelesaian permasalahan dengan menggunakan alat bantu system computer paling tidak akan melibatkan lima tahapan, yaitu: (Sismoro, 2015)

1. Analisis masalah
2. Merancang algoritma
3. Membuat program computer
4. Menguji hasil program computer
5. Dokumentasi

Poin kedua menerangkan bahwa dalam perancangan sebuah system computer dibutuhkan adanya perancangan algoritma. Sehingga setelahnya dapat dilanjutkan ke tahap-tahap berikutnya hingga dokumentasi.

Algoritma adalah Sistem kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada softwarekomputer. Software terbangun atas susunan program (silahkan baca mengenai pengertian program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkah-langkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka,Algoritma berperan penting dalam penyusunan program atau syntax tersebut.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu.

Pengertian Algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu.

Algoritma adalah kunci dari bidang ilmu komputer, dan pada dasarnya setiap hari kita melakukan aktivitas algoritma. Kata algoritma berasal dari sebutan Algorizm.

Algoritma adalah urutan langkah-langkah berhingga untuk memecahkan masalah logika atau matematika

1. Algoritma adalah logika, metode dan tahapan (urutan) sistematis yang digunakan untuk memecahkan suatu permasalahan.
2. Algoritma adalah urutan langkah-langkah logis penyelesaian masalah yang disusun secara sistematis dan logis.
3. Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah.

Pembuatan algoritma harus selalu dikaitkan dengan:

1. Kebenaran algoritma
2. Kompleksitas (lama dan jumlah waktu proses dan penggunaan memori)

Kriteria Algoritma yang baik:

1. Tepat, benar, sederhana, standar dan efektif
2. Logis, terstruktur dan sistematis
3. Semua operasi terdefinisi
4. Semua proses harus berakhir setelah sejumlah langkah dilakukan
5. Ditulis dengan bahasa yang standar dengan format pemrograman agar mudah untuk diimplementasikan dan tidak menimbulkan arti ganda.

2.4 Pengertian UML

Unified Modelling Language (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML dapat dibuat model untuk semua jenis aplikasi piranti lunak, di mana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, atau VB. NET (Sulistiyorini, 2015).

Unified Modeling Language (UML) adalah kumpulan notasi grafis yang didukung oleh sebuah model tunggal, yang membantu dalam menjelaskan dan merancang sistem perangkat lunak, khususnya sistem perangkat lunak dibangun menggunakan gaya berorientasi objek. UML terdiri atas banyak elemen-elemen grafis yang digabungkan membentuk diagram. Tujuan representasi elemen-elemen grafis ke dalam diagram adalah untuk menyajikan beragam sudut pandang dari sebuah sistem berdasarkan fungsi masing-masing diagram tersebut. Kumpulan dari beragam sudut pandang inilah yang kita sebut sebuah model (A.P Utomo, 2016).

Dengan menggunakan model ini diharapkan pengembangan piranti lunak dapat memenuhi semua kebutuhan pengguna dengan lengkap dan tepat,

termasuk faktor-faktor seperti *scalability*, *robustness*, *security*, dan sebagainya. Untuk melakukan pemodelan sistem perangkat lunak secara visual digunakan UML (*Unified Modelling Language*) yang digambarkan secara elektronik lewat sarana perangkat lunak *Rational Rose*. Sebagai mana telah diterapkan oleh Gufran (2012) di mana UML diterapkan untuk mengukur kinerja mahasiswa menggunakan pendekatan berorientasi objek. Kemudian UML diterapkan juga oleh Sunguk (2012) untuk menerapkan sistem *database* dan aplikasi komputer. Selanjutnya Jakimi dan Koutbi (2009) menerapkan pendekatan UML untuk skenario rekayasa dan kode generasi.

1. Use Case Diagram

Use case merupakan teknik menangkap kebutuhan-kebutuhan fungsional dari sistem baru atau sistem yang diubah. Setiap *use case* terdiri dari satu atau lebih skenario yang menerangkan bagaimana sistem berinteraksi dengan pengguna atau sistem yang lain untuk mencapai suatu sasaran bisnis tertentu. Dalam tehnik ini tidak diterangkan cara kerja sistem secara internal maupun implementasinya. Yang ditunjukkan adalah langkah-langkah yang dilakukan pengguna dalam menggunakan perangkat lunak (Artina, 2016).

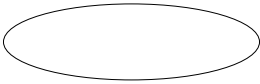
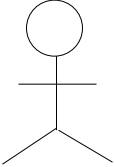
Diagram *Use Case* merupakan diagram yang menggambarkan fungsi berupa komponen, kelas, atau kejadian yang ada dalam *system*. *Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Secara kasar, *use*

case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

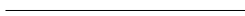
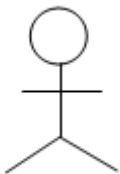


Syarat penamaan pada *use case* adalah nama didefinisikan sesimpel mungkin dan dapat dipahami. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

Use case diagram merupakan pemodelan untuk melakukan (*behavior*) sistem informasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut. Simbol-simbol yang digunakan dalam *Use case Diagram* yaitu :

Tabel 2.1. Simbol *Use case Diagram*

Gambar	Keterangan
	<p><i>Use case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar pesan antar unit dengan aktif, yang dinyatakan dengan menggunakan kata kerja.</p>
	<p><i>Actor</i> atau aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i>, tetapi tidak memiliki kontrol terhadap <i>use case</i>.</p>




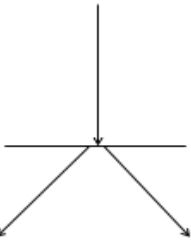
Tabel 2.1. Simbol *Use case Diagram*(Lanjutan)

Gambar	Keterangan
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	<i>Actor</i> atau aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>use case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i> .
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem.
<p>----- <<include>></p>	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya pemanggilan sebuah fungsi program.
<p>←----- <<extends>></p>	<i>Extende</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi.

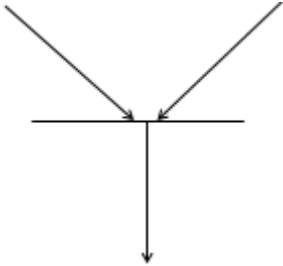
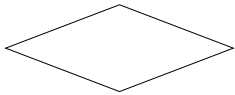
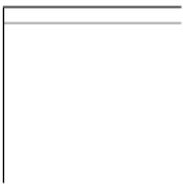
2. Activity Diagram

Activity diagrams menggambarkan *workflow* (aliran kerja) atau aktivitas sari sebuah sistem atau proses bisnis. Yang perlu diperhatikan di sini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh. *Activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol-simbol yang digunakan dalam *activity diagram* yaitu :

Tabel 2.2. Simbol *Activity Diagram*

Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas.
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambarkan suatu proses/kegiatan bisnis.
	<i>Fork</i> /percabangan digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.

Tabel 2.2. Simbol *Activity Diagram*(Lanjutan)

Gambar	Keterangan
	<p><i>Join</i> (penggabungan) atau <i>rake</i>, digunakan untuk menunjukkan adanya dekomposisi.</p>
	<p><i>Decision Points</i>, menggambarkan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>.</p>
	<p><i>Swimlane</i>, pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa</p>

Sumber : Ade Hendini, 2016

3. *Class Diagram*

Class diagram adalah diagram yang menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki 3 bagian utama yaitu *attribute*, *operation*, dan *name*. Kelas-kelas yang ada pada struktur sistem harus dapat melakukan fungsi-fungsi sesuai dengan kebutuhan sistem. Susunan struktur kelas yang baik pada diagram kelas sebaiknya memiliki jenis-jenis kelas berikut :

1. Kelas *Main*. Yaitu kelas yang memiliki fungsi awal dieksekusi ketika sistem dijalankan.

2. Kelas *Interface*. Kelas yang mendefinisikan dan mengatur tampilan ke pemakai. Biasanya juga disebut kelas boundaries.
3. Kelas yang diambil dari pendefinisian usecase. Merupakan kelas yang menangani fungsi-fungsi yang harus ada dan diambil dari pendefinisian usecase.
4. Kelas Entitas. Merupakan kelas yang digunakan untuk memegang atau membungkus data menjadi sebuah kesatuan yang diambil maupun akan disimpan ke basis data.

2.5 Pengertian Informasi

Secara Etimologi, kata informasi ini berasal dari kata bahasa Perancis kuno *informacion* (tahun 1387) mengambil istilah dari bahasa Latin yaitu *informationem* yang berarti “konsep, ide atau garis besar”. Informasi ini merupakan kata benda dari *informare* yang berarti aktivitas dalam “pengetahuan yang dikomunikasikan”.

Informasi adalah hasil pemrosesan data yang diperoleh dari setiap elemen sistem menjadi bentuk yang mudah dipahami dan merupakan pengetahuan yang relevan dan berguna (Yulansari,2013).

Informasi bisa menjadi fungsi penting dalam membantu mengurangi rasa cemas pada seseorang. Menurut pendapat Notoatmodjo (2018) bahwa semakin banyak memiliki informasi dapat memengaruhi atau menambah pengetahuan terhadap seseorang dan dengan pengetahuan tersebut bisa menimbulkan kesadaran

yang akhirnya seseorang itu akan berperilaku sesuai dengan pengetahuan yang dimilikinya.

Informasi adalah data yang telah diolah melalui proses tertentu menjadi sesuatu yang menambah pengetahuan atau temuan yang mempunyai arti baru bagi pemakainya (Melina, 2014).

Adapun fungsi-fungsi informasi adalah sebagai berikut:

1. Untuk meningkatkan pengetahuan bagi si pemakai.
2. Untuk mengurangi ketidakpastian dalam proses pengambilan keputusan pemakai.
3. Menggambarkan keadaan yang sebenarnya dari sesuatu hal. Informasi yang berkualitas harus akurat, tepat dan relevan.

Sumber dari informasi adalah data. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Data merupakan bentuk yang masih mentah, belum dapat bercerita banyak sehingga perlu diolah lebih lanjut. Data diolah melalui suatu metode untuk menghasilkan informasi. Data dapat berbentuk simbol-simbol semacam huruf, angka, bentuk suara, sinyal, gambar, dan sebagainya.

2.6 Pengertian Kriptografi

Kriptografi merupakan kata dari bahasa Yunani yaitu *cryptography*, terdiri dari dua suku kata yaitu kripto dan graphia. Kripto artinya menyembunyikan, sedangkan graphia artinya tulisan. Sehingga, bila digabungkan akan menjadi kata yang berarti menyembunyikan/merahasiakan tulisan. *Kriptografi* adalah suatu

ilmu ataupun seni mengamankan pesan dan dilakukan oleh *cryptographer* (Anonim, 2014).

Kriptografi digunakan untuk memastikan privasi dan autentikasi data dalam komunikasi antar sistem komputer. Terdapat dua proses dasar dalam *kriptografi* yaitu: (Rhee, 2013).

1. *Enkripsi*, adalah sebuah proses yang melakukan perubahan sebuah kode dari yang bisa dimengerti menjadi sebuah kode yang tidak bisa dimengerti (tidak terbaca). (Pabokory, 2015:11).
2. *Deskripsi*, adalah kebalikan dari *Enkripsi* yaitu mengubah kembali bentuk tersamar tersebut menjadi informasi awal. (Pabokory, 2015.)

Sebuah pesan atau data yang masih asli dan belum mengalami penyandian dikenal dengan istilah *plaintext*. Kemudian setelah disamarkan dengan suatu cara penyandian, maka *plaintext* ini disebut sebagai *chipertext*. Proses penyamaran dari *plaintext* ke *Ciphertext* disebut *Enkripsi (encryption)*, dan proses pengembalian dari *Ciphertext* menjadi *plaintext* kembali disebut dekripsi (*decryption*). (Pabokory, 2015). *File* yang dapat di *Enkripsi* dapat berupa teks, gambar maupun audio dan video.

2.6.1 *Kriptografi Klasik*

Kriptografi klasik adalah *kriptografi* yang disebut juga sebagai *kriptografi* kunci tunggal atau *kriptografi* simetris yang menggunakan kunci yang sama untuk *Enkripsi* maupun *Deskripsi*. *Kriptografi* klasik merupakan *kriptografi* yang digunakan pada zaman dahulu sebelum komputer ditemukan atau sudah

ditemukan namun belum secanggih sekarang. *Kriptografi* ini melakukan pengacakan huruf pada kata terang / *plaintext*. (Bishop, 2014).

Algoritma kriptografi klasik digunakan sejak sebelum era komputerisasi dan kebanyakan menggunakan teknik kunci simetris. Metode menyembunyikan pesannya adalah dengan teknik substitusi atau transposisi atau keduanya. Teknik substitusi adalah menggantikan karakter dalam *plaintext* menjadi karakter lain yang hasilnya adalah *Ciphertext*. Sedangkan transposisi adalah teknik mengubah *plaintext* menjadi *Ciphertext* dengan cara permutasi karakter. Kombinasi keduanya secara kompleks adalah yang melatarbelakangi terbentuknya berbagai macam algoritma kriptografi modern. Contoh algoritma kriptografi klasik yaitu: *Caesar Cipher*, *Vigenere Cipher*, dan *Hill Cipher*.

2.6.2 *Kriptografi Modern*

Algoritma kriptografi modern merupakan suatu perbaikan yang mengacu pada kriptografi klasik. Algoritma ini menggunakan pengolahan simbol biner yang dibentuk dari kode *ASCII* (*American Standard Code for Information Interchange*) karena berjalan mengikuti operasi komputer digital, sehingga membutuhkan pengetahuan dasar matematika untuk menguasainya . Algoritma ini memiliki tingkat kesulitan yang kompleks yang menyebabkan kriptanalisis sangat sulit memecahkan *Ciphertext* tanpa mengetahui kuncinya. Adapun jenis kunci dalam kriptografi modern terdiri dari 3 yaitu: *simetri*, *asimetri*, dan hibrida. Pada kriptografi modern terdapat berbagai macam algoritma yang dimaksudkan untuk mengamankan informasi yang dikirim melalui jaringan komputer. Contoh kriptografi modern yaitu *MD5*, *RC4*, *AES* dan lain-lain. (Bishop, 2014).

2.6.3 Proses *Enkripsi dan Deskripsi*

Enkripsi yaitu suatu proses pengaman suatu data yang disembunyikan atau proses konversi data (*plaintext*) menjadi bentuk yang tidak dapat dibaca/dimengerti. Enkripsi telah digunakan untuk mengamankan komunikasi di berbagai negara, namun, hanya organisasi-organisasi tertentu dan individu yang memiliki kepentingan yang sangat mendesak akan kerahasiaan yang menggunakan enkripsi. Di pertengahan tahun 1970an enkripsi kuat dimanfaatkan untuk pengamanan oleh sekretariat agen pemerintah Amerika Serikat pada domain publik, dan saat ini enkripsi telah digunakan pada sistem secara luas, seperti Internet, e-commerce, jaringan telepon bergerak dan ATM pada bank. Enkripsi dapat digunakan untuk tujuan keamanan, tetapi teknik lain masih diperlukan untuk membuat komunikasi yang aman, terutama untuk memastikan integrasi dan autentikasi dari sebuah pesan. Untuk menampilkan enkripsi dan kebalikannya dekripsi, digunakan algoritma yang biasa disebut *Cipher* dengan menggunakan metode serangkaian langkah yang terdefinisi yang diikuti sebagai prosedur. Alternatif lain ialah Enchiperment. Informasi yang asli disebut sebagai *plaintext*, dan bentuk yang sudah dienkripsi disebut sebagai *chiphertext*. Pesan *chiphertext* berisi seluruh informasi dari pesan *plaintext*, tetapi tidak dalam format yang didapat dibaca manusia ataupun komputer tanpa menggunakan mekanisme yang tepat untuk melakukan dekripsi. (Bishop, 2014).

Sedangkan Dekripsi yaitu kebalikan dari proses enkripsi yaitu proses konversi data yang sudah dienkripsi (*ciphertext*) kembali menjadi data aslinya (*Original Plaintext*) sehingga dapat dibaca/ dimengerti kembali. Pesan yang akan

dienkripsi disebut plaintext yang dimisalkan plaintext (P), proses enkripsi dimisalkan enkripsi (E), proses dekripsi dimisalkan dekripsi (D), dan pesan yang sudah dienkripsi disebut ciphertext yang dimisalkan ciphertext (C). (Bishop, 2014).

2.7 Pengertian Web

World Wide Web (www), lebih dikenal dengan *web* yang merupakan salah satu layanan yang didapat oleh pemakai komputer yang terhubung ke internet dengan fasilitas *hypertext* untuk menampilkan data berupa *text*, gambar, suara animasi dan data multimedia lainnya. Sehingga *web* pada awalnya adalah ruang informasi dalam internet dengan menggunakan teknologi *hypertext* pemakai dituntun menemukan informasi dengan mengikuti link yang disediakan dalam dokumen *web* yang ditampilkan dalam *web browser*. Situs *web* dapat dikategorikan menjadi dua yaitu “*web statis*” dan “*web dinamis*”.

Web statis adalah *web* yang menampilkan informasi-informasi yang sifatnya statis (tetap). Disebut statis karena pengguna tidak dapat berinteraksi dengan *web* tersebut. Dengan demikian untuk mengetahui suatu *web* tersebut bersifat statis atau dinamis dapat dilihat dari tampilannya. Jika suatu *web* hanya berhubungan dengan halaman *web* lain dan berisi suatu informasi yang tetap maka *web* tersebut disebut statis.

Web dinamis adalah *web* yang menampilkan informasi serta dapat berinteraksi dengan pengguna. *Web* yang dinamis memungkinkan pengguna untuk

berinteraksi menggunakan *form* sehingga dapat memolah informasi yang ditampilkan. *Web* dinamis bersifat interaktif, tidak kaku dan terlihat lebih indah.

Sehingga dapat disimpulkan bahwa kelebihan dari aplikasi *web* adalah dapat diakses kapan pun dan darimana pun selama ada internet. Dan dapat diakses hanya dengan menggunakan *web browser* perlu menginstal, tidak perlu menginstal aplikasi *web* itu sendiri adalah antarmuka yang dapat dibuat terbatas sesuai spesifikasi standar untuk membuat dokumen *web* dan keterbatasan kemampuan *web browser* untuk menampilkannya. Dan terbatasnya kecepatan internet mungkin membuat respon aplikasi menjadi lambat. (Bunafit Nugroho, 2013)

2.8 Pengertian Web Programming

Dalam *web programming*, terdapat *server-side programming* dan *client-side programming*. *Client-side programming* adalah untuk membuat *web* yang statis, sedangkan untuk membuat *web* yang dinamis (dapat interaktif dengan *user*) diperlukan *server-side* dan *client-side programming*. Program *web* yang tergolong dalam *Client-Side* seperti *Java Script*, *VB Script*, *HTML* dan lain-lain. Hasil *parsing script* pemrograman *client-side* yang berupa *HTML* dari *server web* dapat dilihat dengan memilih menu *view > Source Code*. Sedangkan program *web* yang tergolong *server side* adalah *CGI/Perl*, *ASP*, *JSP*, *PHP*, *CFM*. Hasil *parsing script* pemrograman *server side* yang berupa *HTML* dari *server web* dapat dilihat dengan memilih menu *view > Source Code* juga. Hal ini terjadi karena script hanya diproses di *server web* dan hasilnya dikembalikan dalam bentuk tag-tag *HTML* kemudian ditampilkan pada *browser*.

Berdasarkan basis pengembangan aplikasi (*software*) dapat dibedakan menjadi dua macam, yaitu:

1. Aplikasi berbasis *desktop*

Aplikasi berbasis *desktop* dikembangkan untuk dijalankan dimasing-masing *client* (komputer pengakses aplikasi pengolahan database). *Database* diletakkan di *server* sedangkan aplikasi diinstal dimasing-masing *client*. Bahasa pemrograman yang digunakan untuk aplikasi tipe ini biasanya adalah *Borland Delphi*, *Visual Basic*, *Java netbeans*, dan sebagainya. Pada aplikasi berbasis *desktop*, aplikasi dibangun dengan menggunakan *tool* tertentu, kemudian dikompilasi. Hasilnya dapat langsung digunakan dalam komputer.

2. Aplikasi berbasis *web*

Aplikasi berbasis *web* tidak perlu diinstal dimasing-masing *client* pengakses aplikasi karena aplikasi cukup dikonfigurasi di *server*. Kemudian *client* mengakses dari *browser* seperti *Internet Explorer*, *Opera Mini*, *Firefox*. *Excutor* aplikasi dilakukan oleh *web server* seperti *Apache*, *IIS*, *Xitami* dan lain sebagainya.

Perbedaan lain aplikasi berbasis *desktop* dan *web* adalah bahwa untuk aplikasi dengan mengoptimasi pengguna memori, manajemen proses dan pengaturan *input-output*. Pada aplikasi berbasis *web*, faktor yang menentukan kinerja aplikasi adalah kecepatan akses *database* dan kecepatan akses jaringan dan internet. (Bunafit Nugroho, 2013,)

2.9 Pengertian PHP

PHP (*Hypertext Preprocessor*) merupakan sebuah bahasa *scripting* yang terpasang pada HTML. Sebagian besar sintaks mirip dengan bahasa C, java dan Perl, ditambah beberapa fungsi PHP yang spesifik. Tujuan utama penggunaan bahasa ini adalah untuk memungkinkan perancangan *web* menulis halaman *web* dinamik dengan cepat. Hubungan PHP dengan HTML halaman *web* biasanya disusun dari kode-kode HTML yang disimpan dalam sebuah *file* berekstensi .html. file HTML ini dikirimkan oleh *server* (atau *file*) ke *browser*, kemudian *browser* menerjemahkan kode-kode tersebut sehingga menghasilkan suatu tampilan yang indah. Lain halnya dengan program php, program ini harus diterjemahkan oleh *web server* sehingga menghasilkan kode html yang dikirim ke *browser* agar dapat ditampilkan. Program ini dapat berdiri sendiri ataupun disisipkan di antara kode-kode html sehingga dapat langsung ditampilkan bersama dengan kode-kode html tersebut. Program PHP dapat ditambahkan dengan mengapit program tersebut di antara tanda. Tanda-tanda tersebut biasanya disebut tanda untuk *escaping* (kabur) dari kode html.

PHP merupakan bahasa pemrograman *web* yang bersifat *server-side* HTML *embedded scripting*, dimana *script*-nya menyatu dengan HTML berada di *server*, artinya adalah sintak dan perintah-perintah yang kita berikan akan sepenuhnya dijalankan di *server* tetapi disertakan HTML biasa.

Kelebihan-kelebihan PHP adalah sebagai berikut: (Ardiansyah, S.T, 2014.)

1. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* (pengembang) bisa berkonsentrasi langsung pada penampilan *web*nya.

2. Tidak ada proses *compiling* dan *linking*.
3. Berorientasi obyek (*object oriented*).
4. Sintaksis pemrogramannya mudah dipelajari, mirip *C* dan *Perl*.
5. Integrasi yang sangat luas ke berbagai *server database*. *Database* yang didukung oleh PHP adalah: *Oracle, Sybase, Msql, MSOL, Solid, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, Informix, dBase, UNIXdbm*.

2.10 Pengertian MySql

MySQL merupakan sebuah bentuk *database* yang berjalan sebagai *server*, tidak meletakkan *database* tersebut dalam satu mesin dengan aplikasi yang digunakan, sehingga dapat meletakkan sebuah *database* pada sebuah mesin khusus dan dapat diletakkan ditempat yang jauh komputer pengaksesannya. MySQL merupakan *database* yang sangat kuat dan cukup stabil digunakan sebagai media penyimpanan data. Sebagai *database server* yang mampu memanejemen *database* dengan baik, MySQL terhitung merupakan *database* yang paling banyak digemari dan paling banyak digunakan dibanding *database* yang lain.

MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AD adalah David Axmark, Allan Larsson dan Michael "Monty" Widenius. (Bunafit Nugroho, 2013, Dasar Pemrograman Web PHP-MySQL Dengan Dreamweaver, Gava Media, Yogyakarta)

1. Tipe Data

Data yang terdapat dalam sebuah tabel berupa *field-field* yang berisi nilai dari data tersebut. Nilai data dalam *field* memiliki tipe sendiri-sendiri. MySQL mengenal beberapa tipe data *field* yaitu:

a. Tipe data numerik

Tipe data *numerik* dibedakan dalam dua macam, yaitu *integer* dan *floating point*. Tipe data *numerik* selengkapnya dapat dilihat pada tabel 1.

Tabel 2.3 : Tipe Data Numerik

Tipe Data	Kisaran Nilai
TINYINT	(-128)-127 atau (0-255)
SMALLINT	(-32768)-32767 atau (0-65535)
MEDIUMINT	(-3888608)-8388607 atau 0-16777215
INT, INTEGER	(-2147683648)-(-21447683647) atau 0-4294967295
FLOAT	(-3.4 E+38)-(-1.17E-38), 0 dan 1.175E-38-3.4e+38
DOUBLE	(-1.79E+308)-(-2.225E-308), 0 dan 2.225E-308 – 1.79E+308

b. Tipe data string

String adalah rangkaian karakter. Tipe-tipe data yang termasuk dalam tipe data *string* dapat dilihat pada tabel 2 berikut:

Tabel 2.4 : Tipe Data String

Tipe Data	Kisaran Nilai
CHAR	1-255 karakter
VARCHAR	1-255 karakter
TINYTEXT	1-255 karakter
TEXT	1-65535 karakter
MEDIUMTEXT	1-16777215 karakter
LONGTEXT	1-424967295 karakter

c. Tipe data tanggal

Untuk tanggal dan jam, tersedia tipe-tipe data *field* berupa *DATETIME*, *DATE*, *TIMESTAMP*, *TIME* dan *YEAR*. Masing-masing tipe mempunyai kisaran nilai tertentu. *MYSQL* akan memberikan peringatan kesalahan (*error*) apabila tanggal atau waktu yang dimasukkan salah. Kisaran nilai dan besar memori penyimpanan yang diperlukan untuk masing-masing tipe dapat dilihat pada tabel 3 berikut ini:

Tabel 2.5 : Tipe Data Tanggal

Tipe Data	Kisaran Nilai
DATETIME	1000-01-01 00:00 sampai 9999-12-31 23:59:59
DATE	1000-01-01 sampai 9999-12-31
TIMESTAMP	1970-01-01 00:00:00 sampai 2037
TIME	-839:59:59 sampai 838:59:59

2.11 Pengertian Dreamweaver

Dreamweaver merupakan sebuah HTML profesional untuk mendesain secara visual dan mengelola situs web maupun halaman *web* fasilitas *editing* secara visual dari *dreamweaver* memungkinkan untuk dapat menambahkan *desain* program secara manual. Hanya saja dikarenakan perkembangan perangkat lunak yang semakin cepat, *dreamweaver* semakin membutuhkan spesifikasi komputer yang semakin baik yang cepat sehingga jika dalam penggunaannya menggunakan perangkat keras yang tidak mencukupi akan menjadi hambatan melakukan penelitian.

Dreamweaver cocok bagi *user* yang suka dengan kode – kode HTML secara manual atau lebih menyukai bekerja dalam lingkungan secara visual dalam

melakukan *editing* karena *Dreamweaver* membuatnya menjadi mudah dengan menyediakan *tool – tool* yang sangat berguna dalam meningkatkan kemampuan dalam pengalaman *user* dalam mendesain *web*. *Dreamweaver* mengikut sertakan banyak *tool* untuk kode – kode dalam halaman *web* beserta fasilitas – fasilitasnya, antara lain: Referensi HTML, CSS, *Java Script*, *Java Script Debugger*, dan *editor* kode (tampilan kode dan *code inspector*) yang mengizinkan *user* untuk meng-*edit* kode *Java Script*, XML, dan dokumen teks lain secara langsung dalam *Dreamweaver*. Fasilitas *editing* secara visual dari *Dreamweaver* membuat *user* dapat menambahkan *desain* dan fungsionalitas halaman – halaman *web* tanpa perlu menulis satu baris koden pun. *User* dapat melihat semua aset atau elemen dari situs yang menyeretnya dalam *panel* secara langsung ke dokumen. *User* dapat membuat dan meng-*edit image* dalam *Macromedia Fireworks* kemudian mengimpornya secara langsung ke dalam *Dreamweaver*. Selain itu, *user* juga bisa menambahkan objek *Flash* secara langsung dalam *Dreamweaver*. (Bunafit Nugroho, 2013)

2.12 Pengertian Database

Basis data (*database*) merupakan kumpulan dari data yang saling berhubungan satu dengan lainnya, satu *database* menunjukkan satu kumpulan data yang dipakai dalam satu lingkup perusahaan atau instansi”. *Database* digunakan untuk menyimpan informasi atau data yang terintegrasi dengan baik di dalam komputer. Untuk mengolah *database* diperlukan suatu perangkat lunak yang disebut DBMS (*Database Management System*).

DBMS merupakan suatu sistem perangkat lunak yang memungkinkan *user* (pengguna) untuk membuat, memelihara, mengontrol dan mengakses *database* secara praktis dan efisien. Dengan DBMS, *user* akan lebih mudah mengontrol dan memanipulasi data yang ada. Sedangkan RDBMS atau *Relationship Database System* merupakan salah satu jenis DBMS yang mendukung adanya *relationship* atau hubungan antar label. Disamping RDBMS, terdapat jenis DBMS lain, misalnya: *Hierarchy DBMS*, *Object Oriented DBMS* dan sebagainya.

Berdasarkan tingkat kompleksitas nilai data, tingkatan data dapat disusun kedalam sebuah hierarki, mulai dari yang paling sederhana hingga yang paling kompleks. Urutan atau hierarki *database* adalah sebagai berikut:

1. *Database* adalah sekumpulan dari bermacam-macam tipe *record* yang memiliki hubungan antar *record*.
2. *File* adalah sekumpulan rekaman data yang berkaitan dengan suatu objek.
3. *Record* adalah kumpulan elemen-elemen yang saling berkaitan dengan di informasikan tentang suatu *entity* secara lengkap. Suatu record terdiri atas satu atau beberapa *field* yang membentuk satu kesatuan.
4. *Field* adalah unit terkecil yang disebut data yang tidak dapat dipecah lagi menjadi unit lain yang bermakna.
5. *Byte* adalah bagian terkecil yang dialamatkan dalam memori.
6. *Bit* adalah sistem biner yang terdiri atas dua macam nilai, yaitu 0 dan 1. Sistem biner merupakan dasar yang dapat digunakan untuk komunikasi antara manusia dan mesin, yang merupakan serangkaian komponen

elektronik dan hanya dapat membedakan 2 macam keadaan, yaitu ada tegangan dan tidak ada tegangan yang masuk ke rangkaian tersebut.

Hierarki *database* dapat dilihat sebagai berikut:

2.12 Tabel ASCII

ASCII merupakan kepanjangan dari (*American Standard Code for Information Interchange*), dan pengertian dari ASCII sendiri adalah suatu standar internasional dalam kode huruf dan simbol seperti Hex dan Unicode tetapi ASCII lebih bersifat universal, contohnya 124 adalah untuk karakter "|". Ia selalu digunakan oleh komputer dan alat komunikasi lain untuk menunjukkan teks. sedangkan fungsi dari kode ASCII ialah digunakan untuk mewakili karakter-karakter angka maupun huruf didalam komputer, sebagai contoh dapat kita lihat pada karakter 1, 2, 3, A, B, C, dan sebagainya.

ASCCI (*American Standard Code for Information Interchange*) merupakan Kode Standar Amerika untuk Penukaran Informasi atau sebuah standar internasional dalam pengkodean huruf dan simbol seperti Unicode dan Hex tetapi ASCII lebih bersifat universal. Pada materi kali ini sobat akan menemukan 8 bit, 256 karakter ASCII, menurut ISO 8859-1 dan Microsoft Windows Latin-1 dengan peningkatan karakter, yang tersedia dalam program tertentu seperti Microsoft Word. Dalam bahasa computer 0 dan 1 tidak ada cara lain untuk mewakili huruf dan karakter yang bukan nomer. Semuanya harus menggunakan 0 dan 1. Salah satu jalan untuk berbahasa dengan komputer dengan cara menggunakan table ASCII.

Tabel 2.6. Tabel ASCII

DEC	OCT	HEX	BIN	Symbol
0	000	00	00000000	NUL
1	001	01	00000001	SOH
2	002	02	00000010	STX
3	003	03	00000011	ETX
4	004	04	00000100	EOT
5	005	05	00000101	ENQ
6	006	06	00000110	ACK
7	007	07	00000111	BEL
8	010	08	00001000	BS
9	011	09	00001001	HT
10	012	0A	00001010	LF
11	013	0B	00001011	VT
12	014	0C	00001100	FF
13	015	0D	00001101	CR
14	016	0E	00001110	SO
15	017	0F	00001111	SI
16	020	10	00010000	DLE
17	021	11	00010001	DC1
18	022	12	00010010	DC2
19	023	13	00010011	DC3
20	024	14	00010100	DC4
21	025	15	00010101	NAK
22	026	16	00010110	SYN
23	027	17	00010111	ETB
24	030	18	00011000	CAN
25	031	19	00011001	EM
26	032	1A	00011010	SUB

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
27	033	1B	00011011	ESC
28	034	1C	00011100	FS
29	035	1D	00011101	GS
30	036	1E	00011110	RS
31	037	1F	00011111	US
32	040	20	00100000	
33	041	21	00100001	!
34	042	22	00100010	"
35	043	23	00100011	#
36	044	24	00100100	\$
37	045	25	00100101	%
38	046	26	00100110	&
39	047	27	00100111	'
40	050	28	00101000	(
41	051	29	00101001)
42	052	2A	00101010	*
43	053	2B	00101011	+
44	054	2C	00101100	,
45	055	2D	00101101	-
46	056	2E	00101110	.
47	057	2F	00101111	/
48	060	30	00110000	0
49	061	31	00110001	1
50	062	32	00110010	2
51	063	33	00110011	3
52	064	34	00110100	4
53	065	35	00110101	5
54	066	36	00110110	6
55	067	37	00110111	7

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
56	070	38	00111000	8
57	071	39	00111001	9
58	072	3A	00111010	:
59	073	3B	00111011	;
60	074	3C	00111100	<
61	075	3D	00111101	=
62	076	3E	00111110	>
63	077	3F	00111111	?
64	100	40	01000000	@
65	101	41	01000001	A
66	102	42	01000010	B
67	103	43	01000011	C
68	104	44	01000100	D
69	105	45	01000101	E
70	106	46	01000110	F
71	107	47	01000111	G
72	110	48	01001000	H
73	111	49	01001001	I
74	112	4A	01001010	J
75	113	4B	01001011	K
76	114	4C	01001100	L
77	115	4D	01001101	M
78	116	4E	01001110	N
79	117	4F	01001111	O
80	120	50	01010000	P
81	121	51	01010001	Q
82	122	52	01010010	R
83	123	53	01010011	S

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
84	124	54	01010100	T
85	125	55	01010101	U
86	126	56	01010110	V
87	127	57	01010111	W
88	130	58	01011000	X
89	131	59	01011001	Y
90	132	5A	01011010	Z
91	133	5B	01011011	[
92	134	5C	01011100	\
93	135	5D	01011101]
94	136	5E	01011110	^
95	137	5F	01011111	_
96	140	60	01100000	`
97	141	61	01100001	A
98	142	62	01100010	B
99	143	63	01100011	C
100	144	64	01100100	D
101	145	65	01100101	E
102	146	66	01100110	F
103	147	67	01100111	G
104	150	68	01101000	H
105	151	69	01101001	I
106	152	6A	01101010	J
107	153	6B	01101011	K
108	154	6C	01101100	L
109	155	6D	01101101	M
110	156	6E	01101110	N
111	157	6F	01101111	O

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
112	160	70	01110000	P
113	161	71	01110001	Q
114	162	72	01110010	R
115	163	73	01110011	S
116	164	74	01110100	T
117	165	75	01110101	U
118	166	76	01110110	V
119	167	77	01110111	W
120	170	78	01111000	X
121	171	79	01111001	Y
122	172	7A	01111010	Z
123	173	7B	01111011	{
124	174	7C	01111100	
125	175	7D	01111101	}
126	176	7E	01111110	~
127	177	7F	01111111	
128	200	80	10000000	€
129	201	81	10000001	
130	202	82	10000010	,
131	203	83	10000011	ƒ
132	204	84	10000100	„
133	205	85	10000101	...
134	206	86	10000110	†
135	207	87	10000111	‡
136	210	88	10001000	^
137	211	89	10001001	‰
138	212	8A	10001010	Š
139	213	8B	10001011	<

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
140	214	8C	10001100	œ
141	215	8D	10001101	
142	216	8E	10001110	ž
143	217	8F	10001111	
144	220	90	10010000	
145	221	91	10010001	‘
146	222	92	10010010	’
147	223	93	10010011	“
148	224	94	10010100	”
149	225	95	10010101	•
150	226	96	10010110	—
151	227	97	10010111	—
152	230	98	10011000	~
153	231	99	10011001	™
154	232	9A	10011010	š
155	233	9B	10011011	›
156	234	9C	10011100	œ
157	235	9D	10011101	
158	236	9E	10011110	ž
159	237	9F	10011111	Ÿ
160	240	A0	10100000	
161	241	A1	10100001	ı
162	242	A2	10100010	ç
163	243	A3	10100011	£
164	244	A4	10100100	¤
165	245	A5	10100101	¥
166	246	A6	10100110	ı
167	247	A7	10100111	§

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
168	250	A8	10101000	¨
169	251	A9	10101001	©
170	252	AA	10101010	^a
171	253	AB	10101011	«
172	254	AC	10101100	¬
173	255	AD	10101101	
174	256	AE	10101110	®
175	257	AF	10101111	ˉ
176	260	B0	10110000	°
177	261	B1	10110001	±
178	262	B2	10110010	²
179	263	B3	10110011	³
180	264	B4	10110100	´
181	265	B5	10110101	μ
182	266	B6	10110110	¶
183	267	B7	10110111	·
184	270	B8	10111000	¸
185	271	B9	10111001	¹
186	272	BA	10111010	°
187	273	BB	10111011	»
188	274	BC	10111100	¼
189	275	BD	10111101	½
190	276	BE	10111110	¾
191	277	BF	10111111	ı
192	300	C0	11000000	À
193	301	C1	11000001	Á
194	302	C2	11000010	Â
195	303	C3	11000011	Ã
196	304	C4	11000100	Ä

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
197	305	C5	11000101	Å
198	306	C6	11000110	Æ
199	307	C7	11000111	Ç
200	310	C8	11001000	È
201	311	C9	11001001	É
202	312	CA	11001010	Ê
203	313	CB	11001011	Ë
204	314	CC	11001100	Ì
205	315	CD	11001101	Í
206	316	CE	11001110	Î
207	317	CF	11001111	Ï
208	320	D0	11010000	Ð
209	321	D1	11010001	Ñ
210	322	D2	11010010	Ò
211	323	D3	11010011	Ó
212	324	D4	11010100	Ô
213	325	D5	11010101	Õ
214	326	D6	11010110	Ö
215	327	D7	11010111	×
216	330	D8	11011000	Ø
217	331	D9	11011001	Ù
218	332	DA	11011010	Ú
219	333	DB	11011011	Û
220	334	DC	11011100	Ü
221	335	DD	11011101	Ý
222	336	DE	11011110	Þ
223	337	DF	11011111	ß
224	340	E0	11100000	À
225	341	E1	11100001	Á
226	342	E2	11100010	Â

Tabel 2.6. Tabel ASCII (Lanjutan)

DEC	OCT	HEX	BIN	Symbol
227	343	E3	11100011	Ã
228	344	E4	11100100	Ä
229	345	E5	11100101	Å
230	346	E6	11100110	Æ
231	347	E7	11100111	Ç
232	350	E8	11101000	È
233	351	E9	11101001	É
234	352	EA	11101010	Ê
235	353	EB	11101011	Ë
236	354	EC	11101100	Ì
237	355	ED	11101101	Í
238	356	EE	11101110	Î
239	357	EF	11101111	Ï
240	360	F0	11110000	Ð
241	361	F1	11110001	Ñ
242	362	F2	11110010	Ò
243	363	F3	11110011	Ó
244	364	F4	11110100	Ô
245	365	F5	11110101	Õ
246	366	F6	11110110	Ö
247	367	F7	11110111	÷
248	370	F8	11111000	Ø
249	371	F9	11111001	Ù
250	372	FA	11111010	Ú
251	373	FB	11111011	Û
252	374	FC	11111100	Ü
253	375	FD	11111101	Ý
254	376	FE	11111110	Þ
255	377	FF	11111111	ÿ

2.13 Bahasa pemrograman

Bahasa pemrograman adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas tertentu. Bahasa pemrograman merupakan sebuah instruksi untuk memerintah komputer agar bisa menjalankan fungsi tertentu, namun hanya instruksi standar saja. Bahasa pemrograman juga memiliki perhimpunan dari aturan sintaks dan semantik yang tugasnya untuk mendefinisikan program komputer. Bahasa pemrograman komputer yang kita kenal antara lain adalah *Java*, *Visual Basic*, *C++*, *C*, *PHP*, dan bahasa pemrograman lainnya. Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Menurut generasi bahasa pemrograman digolongkan menjadi 4 generasi, yaitu:

1. Generasi ke-1: *machine language*
2. Generasi ke-2: *assembly language: Assembler*
3. Generasi ke-3: *high level programming language*, contoh: *C* dan *Pascal*
4. Generasi ke-4: *4 GL (fourth-generation language)*, contoh: *SQL*
5. Generasi ke-5: *Programming Language Based Object Oriented & Web Development*

Secara umum bahasa pemrograman dibagi menjadi 4 kelompok, yaitu :

1. *Object Oriented Language* : Seperti bahasa *Visual C*, *Delphi*, *Visual dBase*, *Visual FoxPro*.
2. *Low Level Language* : Bahasa *Assembly*.

3. *Middle Level Language* : Bahasa C.

4. *High Level Language* : Bahasa Basic dan Pascal.

Menurut tingkat kedekatannya dengan mesin komputer, bahasa pemrograman terdiri dari:

- a. Bahasa Mesin, yaitu memberikan perintah kepada komputer dengan memakai kode bahasa biner, contohnya 01100101100110.
- b. Bahasa Tingkat Rendah, atau dikenal dengan istilah bahasa rakitan (bah.Inggris *Assembly*), yaitu memberikan perintah kepada komputer dengan memakai kode-kode singkat (kode *mnemonic*), contohnya *MOV, SUB, CMP, JMP, JGE, JL, LOOP*, dsb.
- c. Bahasa Tingkat Menengah, yaitu bahasa komputer yang memakai campuran instruksi dalam kata-kata bahasa manusia (lihat contoh Bahasa Tingkat Tinggi di bawah) dan instruksi yang bersifat simbolik, contohnya {, }, ?, <<, >>, &&.
- d. Bahasa Tingkat Tinggi, yaitu bahasa komputer yang memakai instruksi berasal dari unsur kata-kata bahasa manusia, contohnya *begin, end, if, for, while, and, or, dsb*. Komputer dapat mengerti bahasa manusia itu diperlukan program *compiler* atau *interpreter*.

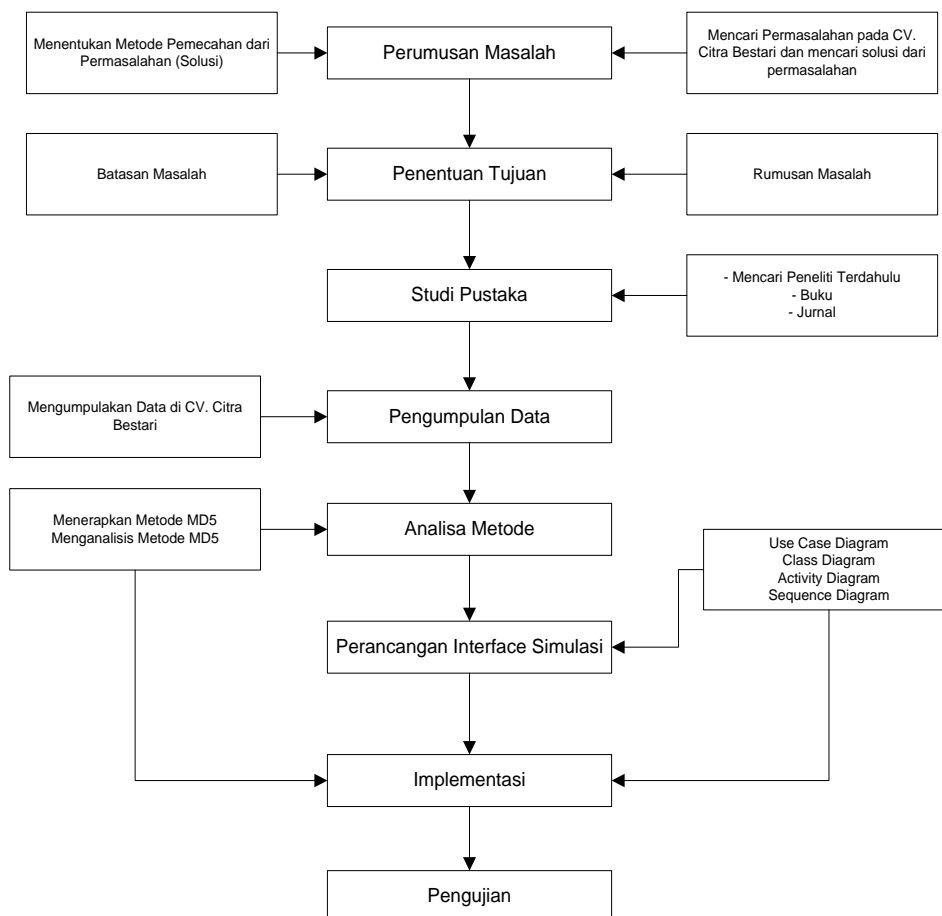
Fungsi dari bahasa pemrograman adalah untuk memerintahkan sebuah komputer agar dapat mengolah data yang sesuai dengan di inginkan. *Output* dari bahasa pemrograman ini dapat berupa aplikasi ataupun program khusus. Contoh sederhananya seperti lampu lalu lintas di jalan raya.

BAB III

METODE PENELITIAN

3.1 Tahapan Penelitian

Analisis sistem dilakukan untuk mendapatkan informasi penelitian terhadap sistem yang berjalan dan menentukan bagaimana sistem yang diusulkan agar menjawab permasalahan yang terjadi pada sistem yang berjalan.



Gambar 3.1. Tahapan Penelitian

3.2 Metode Pengumpulan Data

Pengumpulan data adalah pencarian terhadap sesuatu karena ada perhatian dan keinginan terhadap hasil suatu aktivitas. Metode pengumpulan data dalam penulisan ini dibagi menjadi 2, yaitu :

1. Pengamatan (*Observation*)

Penulis melakukan pengamatan langsung pada setiap proses pendistribusian data barang di PT. Citra Bestari.

2. Penelitian Kepustakaan (*Library Research*)

Merupakan cara untuk mencari referensi dengan mengumpulkan bahan-bahan pustaka yang dilakukan di perpustakaan kampus, maupun perpustakaan umum, juga melakukan pencarian lewat internet, dengan mengunjungi situs-situs seperti *google Book online* yang dapat membantu pembahasan materi.

3.3 Analisis Sistem yang Berjalan

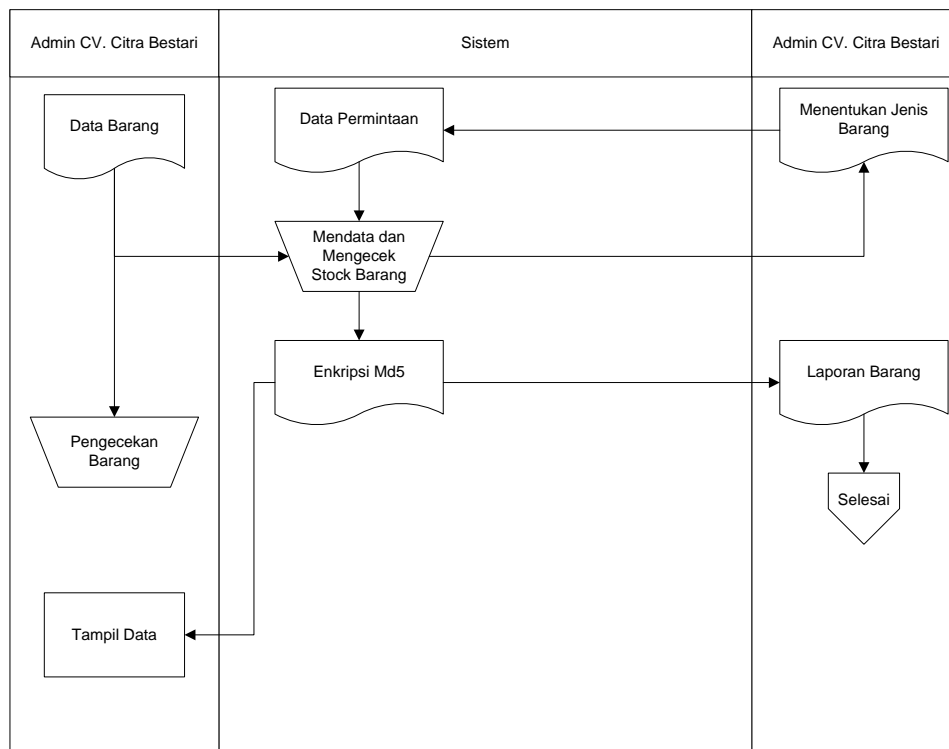
Sistem yang berjalan di PT. Citra Bestari dalam pengelolaan data barang masih kurang efektif karena mengandalkan berkas yang sebelumnya dikelola menggunakan aplikasi Microsoft Office Excel. Data yang disimpan juga tergolong minim karena hanya menyimpan teks dan angka, dan tidak menyimpan file media seperti photo. Untuk membuat kesimpulan dari data barang, dibutuhkan analisa secara manual oleh *staff* administrasi.

3.4 Rancangan Penelitian

Berdasarkan kelemahan dari sistem yang berjalan saat ini, penulis bermaksud untuk membuat suatu aplikasi yang dapat diterapkan untuk pengolahan data barang, dengan mengutamakan penyimpanan yang lebih lengkap dan terstruktur, serta penggunaan aplikasi yang mudah dan cepat. Penulis menuangkan maksud tersebut dalam pembuatan sistem informasi bagi PT. Citra Bestari.

3.4.1 Perancangan Sistem Usulan

Adapun sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:

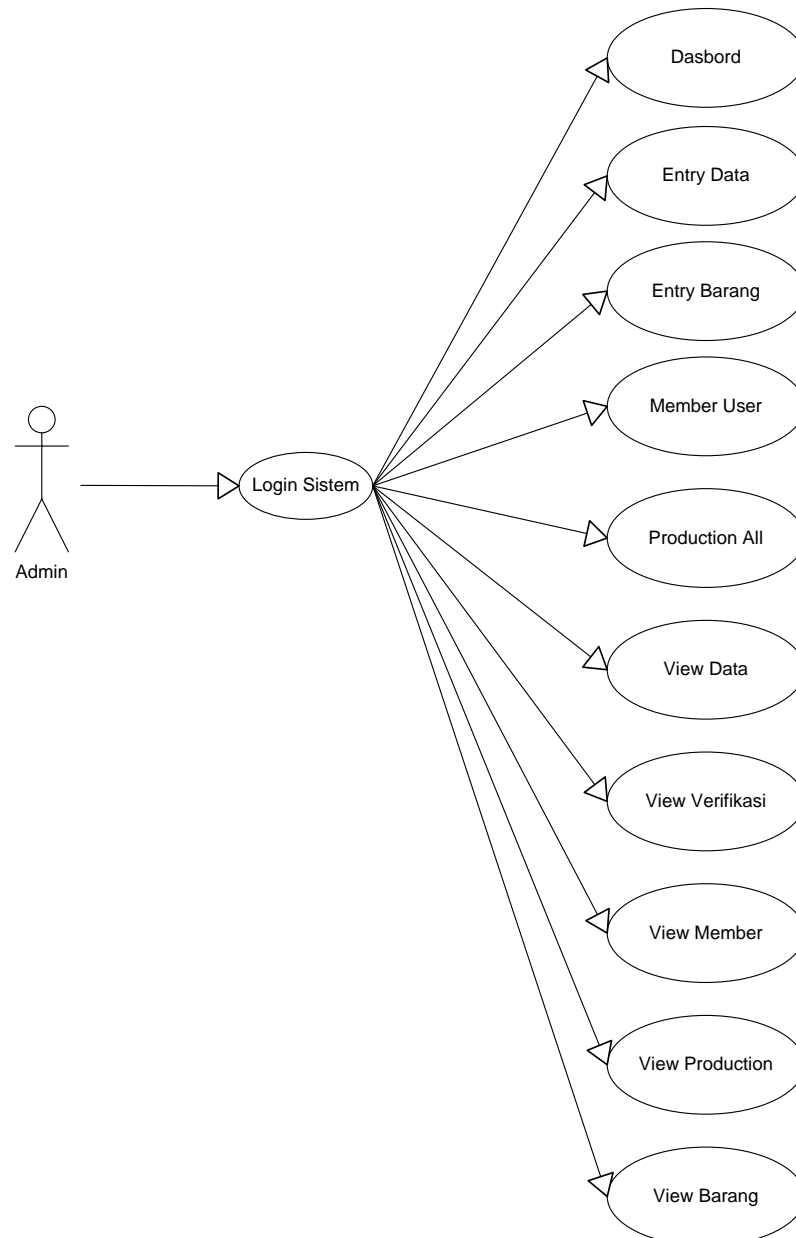


Gambar 3.2. Flowmap Usulan

3.4.2 Desain Sistem Secara Global

1. Use Case Diagram

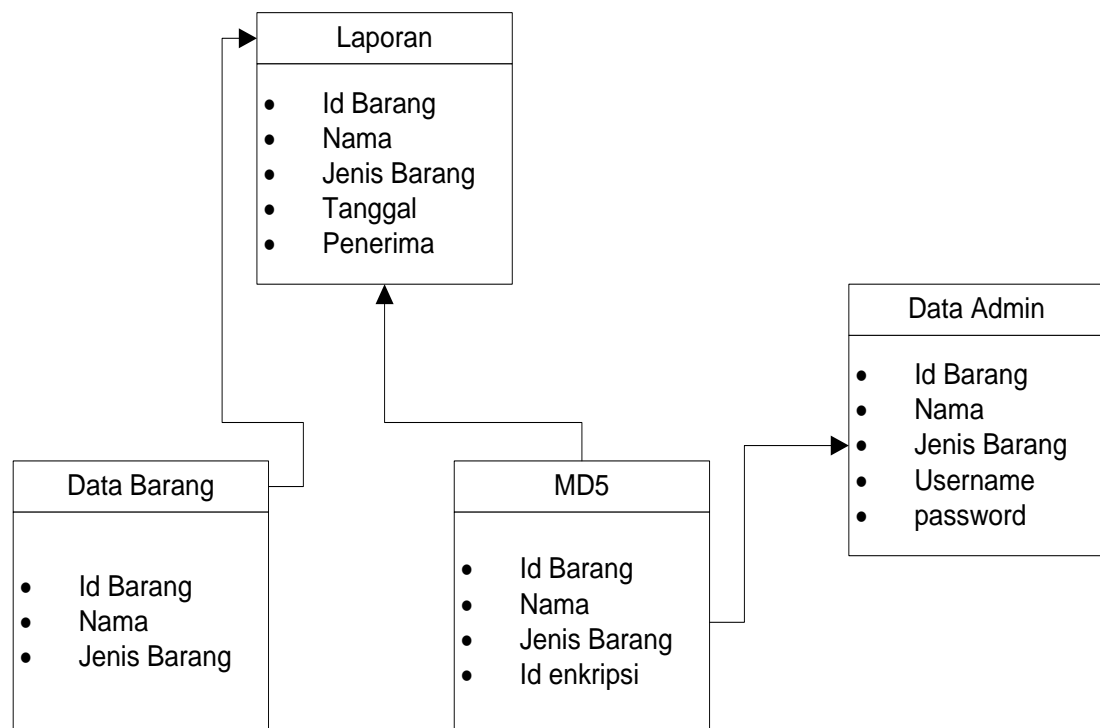
Adapaun use case digaram sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:



Gambar 3.3. Use Case Diagram

2. *Class Diagram*

Adapun class diagram sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:



Gambar 3.4. *Class Diagram*

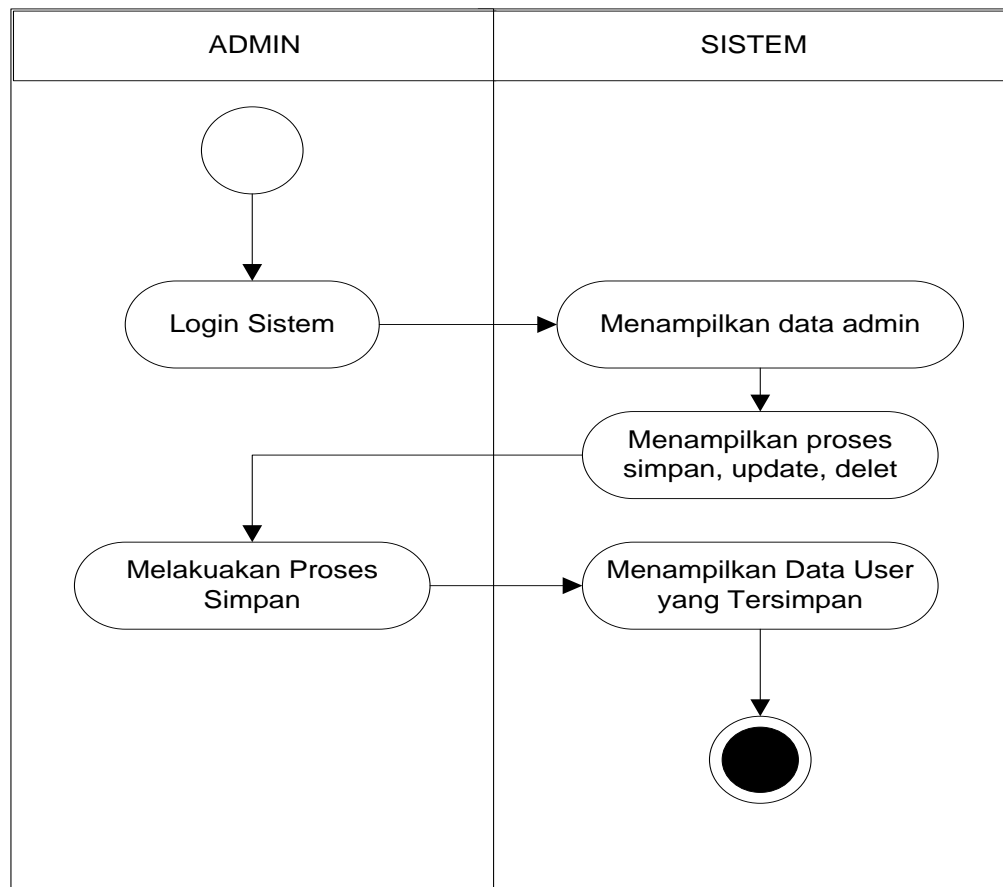
Pada sistem yang diusulkan memiliki 4 tabel data, adapun tabel sebagai berikut: tabel admin, tabel barang, tabel md5, tabel laporan.

3.5 **Desain Sistem Secara Detail**

3.5.1 *Activity Diagram*

Adapun *activity* diagram sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:

1. Activity Diagram Login

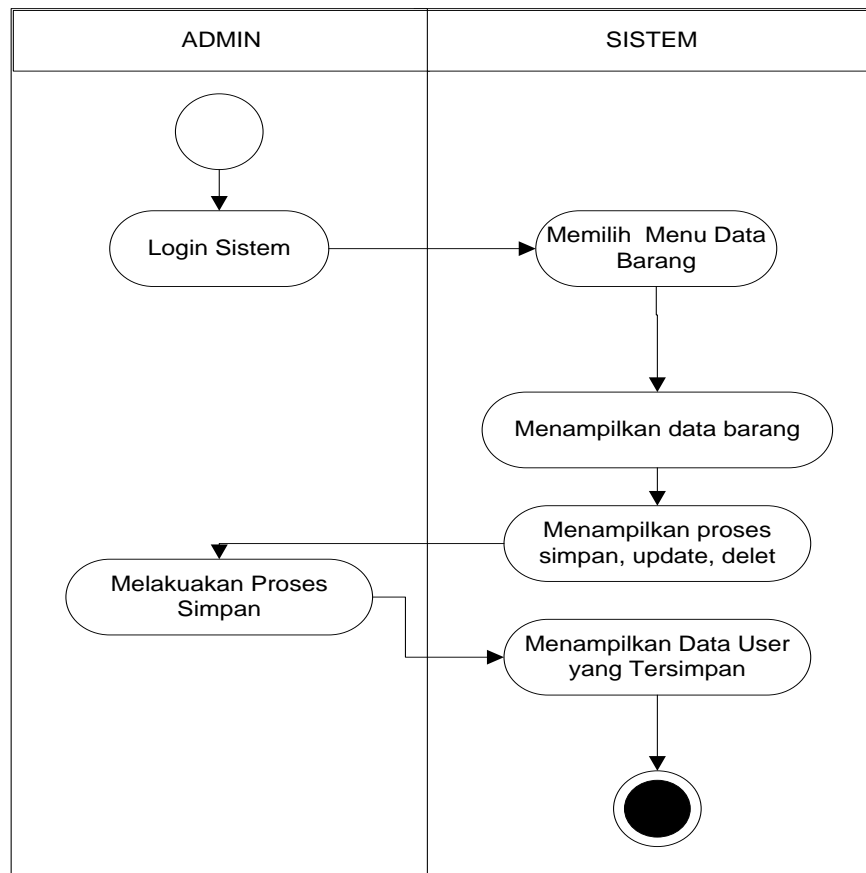


Gambar 3.5. Activity Diagram Login

Keterangan:

1. Admin melakukan login sistem
2. Setelah admin melakukan login, maka akan menampilkan data user
3. Jika ingin menambahkan user, maka pilih proses simpan.
4. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem.
5. Lalu proses simpan user login selesai.

b. Activity Diagram Data Barang

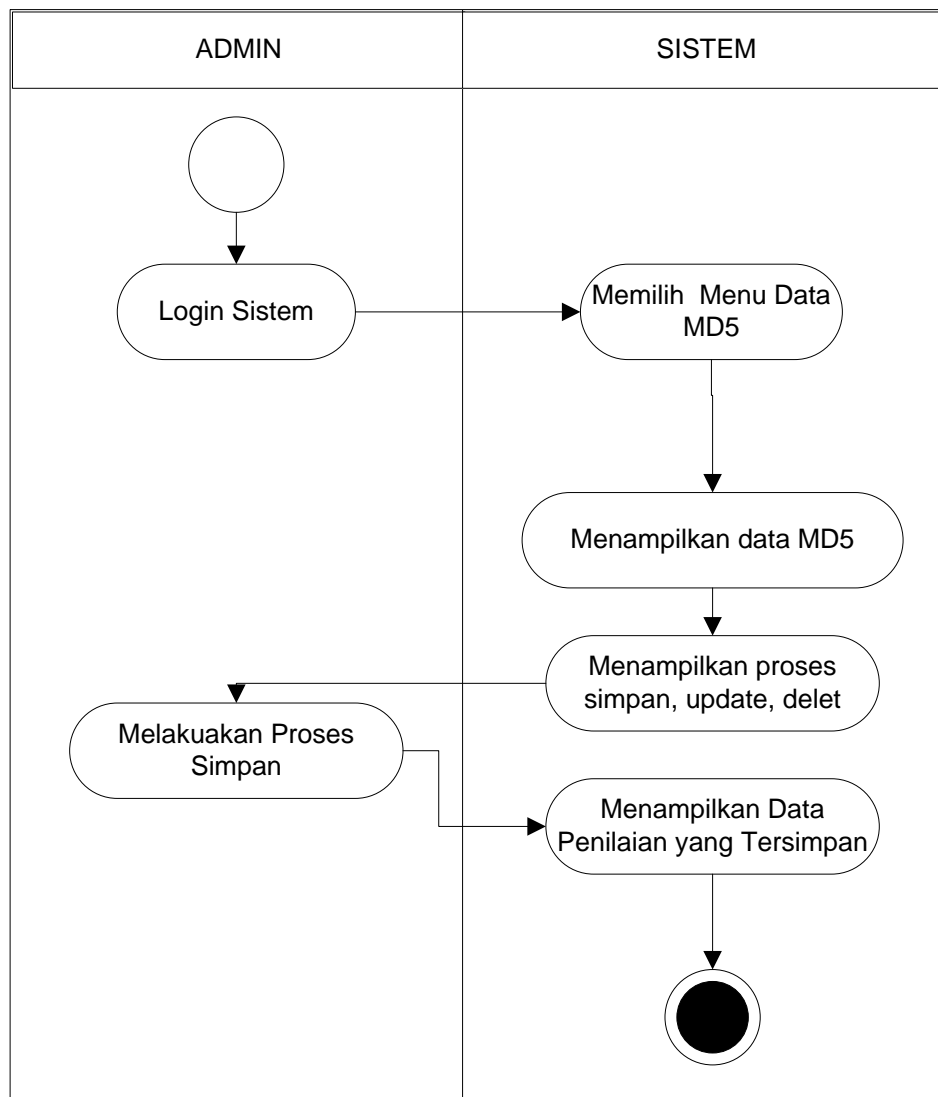


Gambar 3.6. Activity Diagram Data Barang

Keterangan:

1. Admin melakukan login sistem
2. Setelah admin melakukan memilih menu data barang, maka akan menampilkan data barang
3. Jika ingin menambahkan barang, maka pilih proses simpan barang.
4. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem.
5. Lalu proses simpan barang selesai.

c. **Activity Diagram Data MD5**



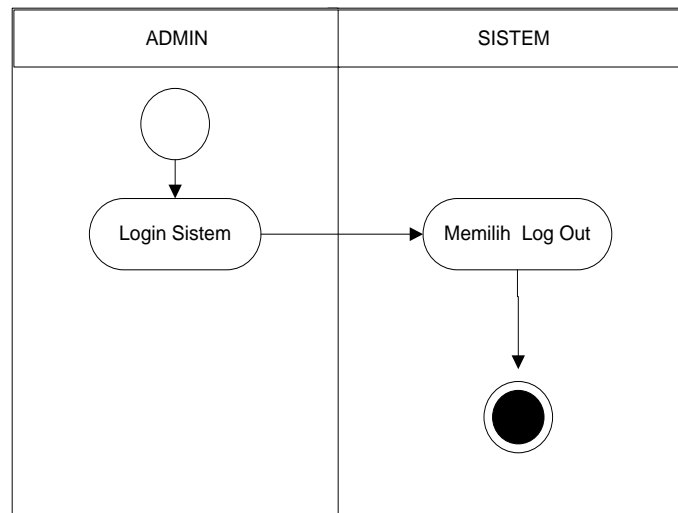
Gambar 3.7. Activity Diagram Data MD5

Keterangan:

1. Admin melakukan login sistem
2. Setelah admin melakukan memilih menu data kriteria, maka akan menampilkan data MD5
3. Jika ingin menambahkan MD5, maka pilih proses simpan MD5.

4. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem.
5. Lalu proses simpan MD5 selesai.

e. **Activity Diagram Log Out**



Gambar 3.8. *Activity Diagram Log Out*

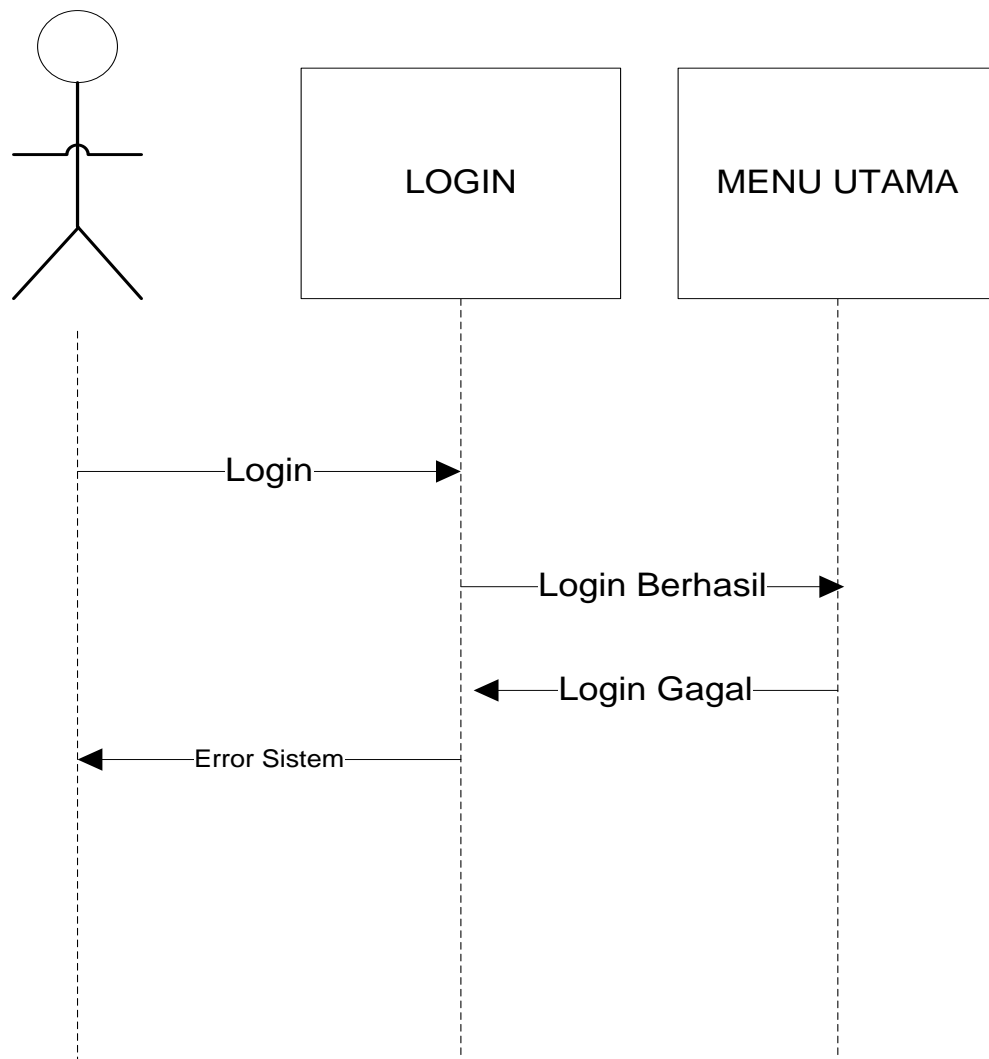
Keterangan:

1. Admin melakukan logut sistem
2. Setelah admin melakukan memilih menu log out, maka sistem langsung shutdown.

3.5.2 Sequence Diagram

Adapaun sequence digram sistem usulan yang dilakukan oleh penulis adalah sebagai berikut:

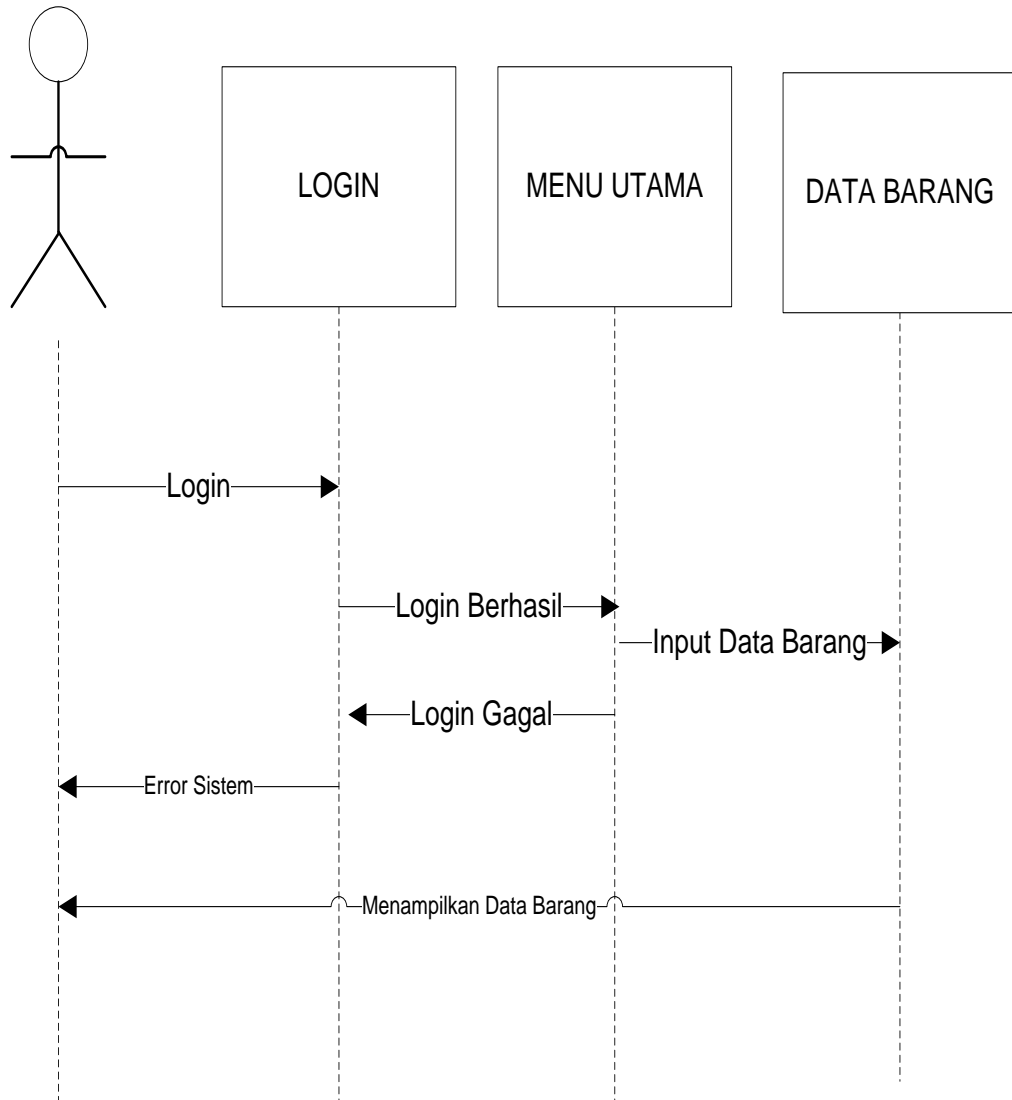
1. *Sequence Diagram Login*



Gambar 3.9. *Sequence Diagram Login*

Admin melakukan login sistem, Setelah admin melakukan login, maka akan menampilkan data user, Jika ingin menambahkan user, maka pilih proses simpan. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem maka proses simpan user login selesai.

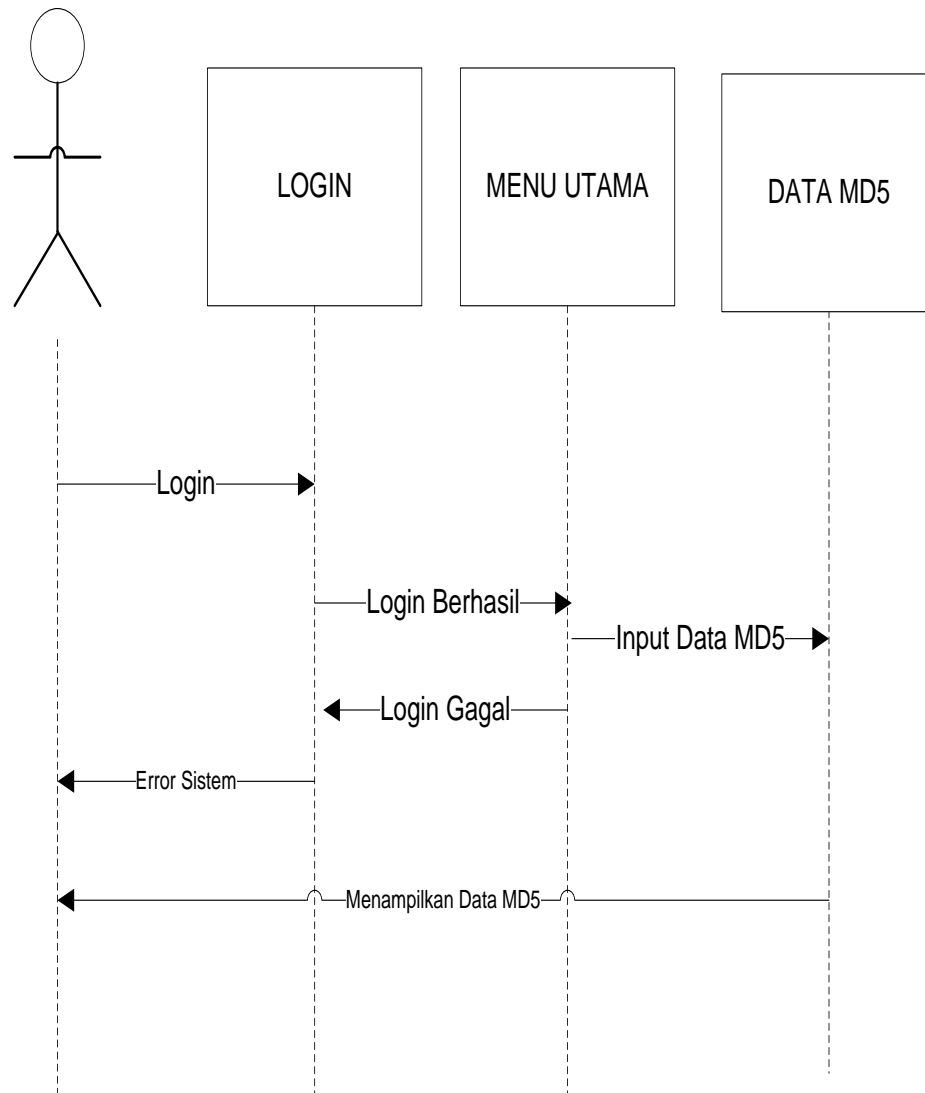
3.13. Sequence Diagram Data Barang



Gambar 3.10 Sequence Diagram Data Barang

Admin melakukan login sistem, Setelah admin melakukan memilih menu data barang, maka akan menampilkan data barang. Jika ingin menambahkan barang, maka pilih proses simpan barang. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem maka proses simpan barang selesai.

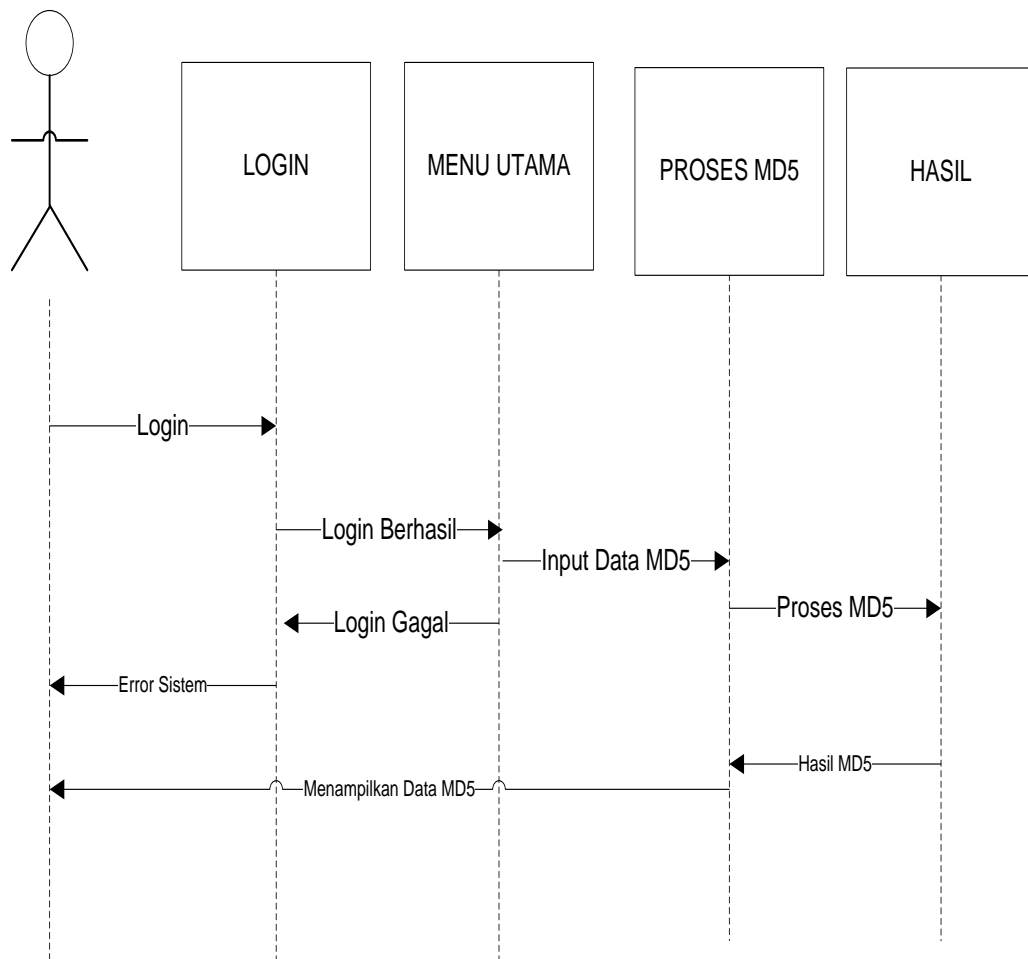
3.14. Sequence Diagram Data MD5



Gambar 3.11 Sequence Diagram Data MD5

Admin melakukan login sistem, Setelah admin melakukan memilih menu data MD5, maka akan menampilkan data MD5. Jika ingin menambahkan MD5, maka pilih proses simpan MD5. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem maka proses simpan data MD5 selesai.

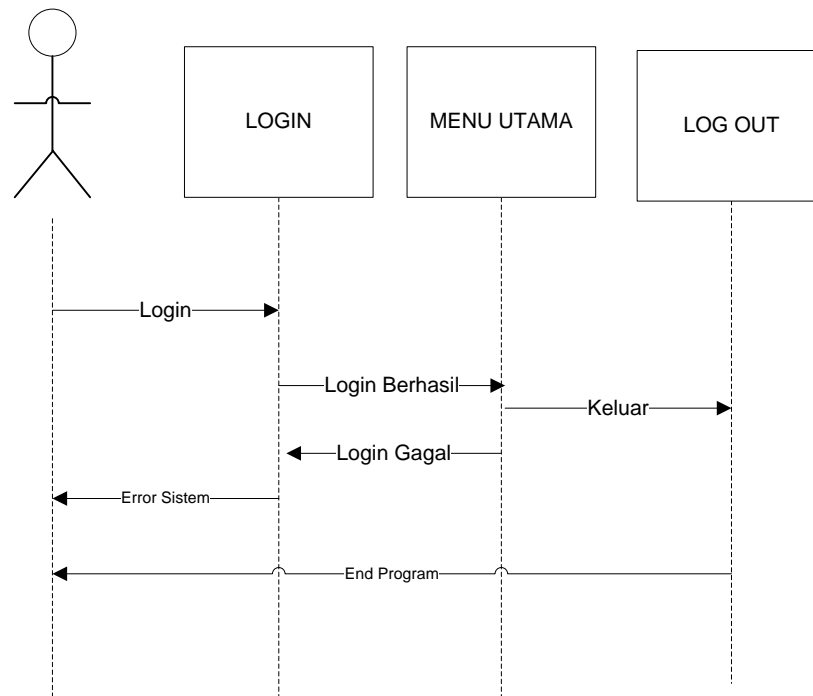
3.15. Sequence Diagram Tampil MD5



Gambar 3.12 Sequence Diagram Hasil

Admin melakukan login sistem, Setelah admin melakukan memilih menu proses hasil, maka akan menampilkan hasil. Lalu memilih proses hasil, maka pilih proses simpan penilaian lalu klik proses. Setelah dilakukan proses simpan, maka akan tersimpan kedalam database sistem maka proses selesai.

3.16. Sequence Diagram Log Out



Gambar 3.13 Sequence Diagram Log Out

3.5.6 Perancangan Database

Proses untuk menentukan isi data pengaturan data yang dibutuhkan untuk mendukung berbagai rancangan *system*.

Berikut adalah struktur tabel dari website PT. Citra Bestari

1. Tabel Data Nasabah

Merupakan data yang diperlukan pada saat nasabah melakukan penginputan data.

Tabel 3.1. Tabel Data Nasabah

Field	Type	Null	Default	Keterangan
Producer_name	char(50)	Yes	NULL	Primay Key
Tanggal_submit	datetime	Yes	NULL	Tanggal Input

Tabel 3.1. Tabel Data Nasabah (Lanjutan)

Field	Type	Null	Default	Keterangan
Tanggal_issued	datetime	Yes	NULL	Tanggal Terbit
status	char(80)	Yes	NULL	Status barang
mop	char(20)	Yes	NULL	Proses pembayaran
Pemegang_polis	char(60)	Yes	NULL	Penanggung jawab
nospaj	char(20)	Yes	NULL	Nomor kwintansi
alp	char(20)	Yes	NULL	Kode kwintansi

2. Tabel Data Admin

Merupakan seorang yang mempunyai tugas dalam manajemen sumber daya data dalam Perusahaan.

Tabel 3.2. Tabel Data Admin

Field	Type	Null	Default	Keterangan
username	char(50)	Yes	NULL	Primay Key
password	char(20)	Yes	NULL	password

3. Tabel Data Member

Merupakan suatu data yang memudahkan para pelanggan dalam mengakses barang pada Perusahaan.

Tabel 3.3. Tabel Data Member

Field	Type	Null	Default	Keterangan
Nm_member	char(100)	Yes	NULL	Primay Key
levelsub	char(80)	Yes	NULL	Level member
direkrut	char(20)	Yes	NULL	Pimpinan

Tabel 3.3. Tabel Data Member (Lanjutan)

Field	Type	Null	Default	Keterangan
Kode_unit	char(20)	Yes	NULL	Kode unit
Kode_kpm	char(20)	Yes	NULL	Kode kpm
Nohp	char(80)	Yes	NULL	Nomor telepon

3.6 Desain Sistem Secara Detail

Perancangan *interface* dapat dilihat sebagai berikut :

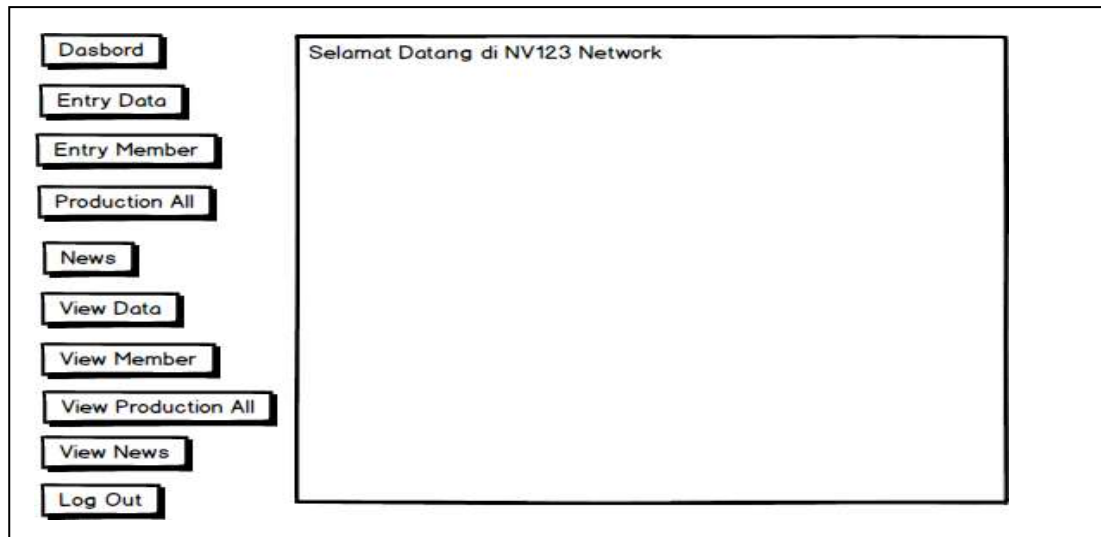
1) Tampilan Halaman *Login*

PT. CITRA BESTARI

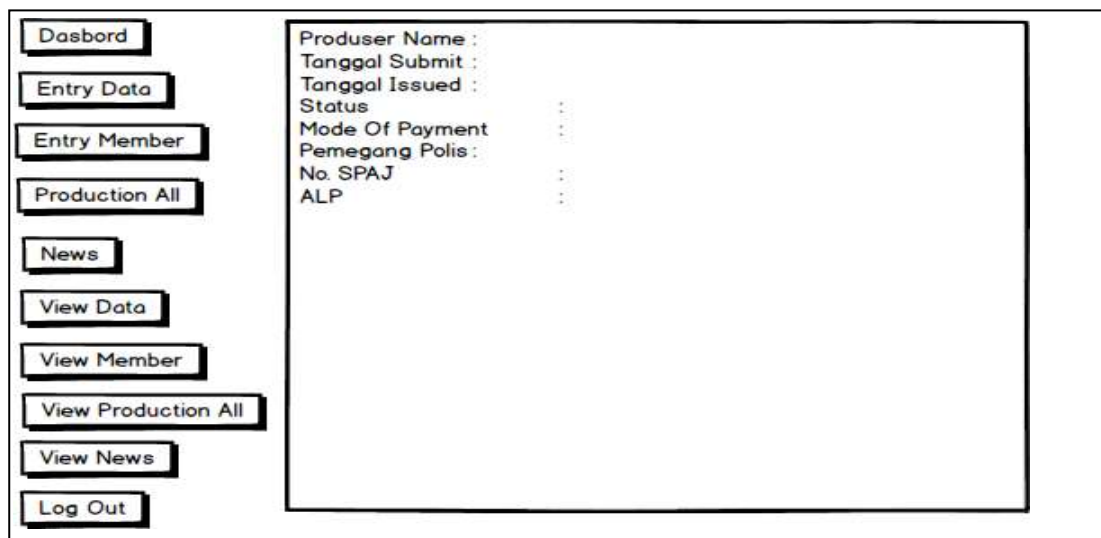
Username :

Password :

Gambar 3.14. Tampilan Halaman *Login*

2) Tampilan Halaman *Home*

Gambar 3.15. Tampilan Halaman Home

3) Tampilan *Entry Data*Gambar 3.16. Tampilan *Entry Data*

4) Tampilan Entry Member

Gambar 3.17. Tampilan *Entry Member*5) Tampilan *View Data*

Producer Name	Tanggal Submit	Tanggal Issued	Status	MOD	Pemegang Polis	NO. SPAJ	ALP

Gambar 3.18. Tampilan *View Data*

BAB IV

HASIL DAN PEMBAHASAN

4.1 Implementasi Sistem

Implementasi sistem adalah langkah-langkah atau prosedur-prosedur yang dilakukan dalam menyelesaikan desain sistem yang telah disetujui, untuk menguji, menginstal dan memulai sistem baru atau yang diperbaiki untuk menggantikan sistem yang lama.

Adapun langkah-langkah yang dibutuhkan dalam implementasi sistem adalah:

1. Mendapatkan *software* dan *hardware* yang tepat/sesuai untuk merancang *website*.
2. Menyelesaikan rancangan sistem.
3. Menulis, menguji, mengontrol dan mendokumentasikan *website*.
4. Mendapatkan persetujuan.

4.1.1 Tujuan Implementasi Sistem

Tujuan implementasi sistem adalah sebagai berikut:

1. Menyelesaikan desain sistem yang telah disetujui sebelumnya.
2. Memastikan bahwa pemakai (*user*) dapat mengoperasikan sistem baru.
3. Menguji apakah sistem baru tersebut sesuai dengan pemakai.

Memastikan bahwa konversi ke sistem baru berjalan yaitu dengan membuat rencana, mengontrol dan melakukan instalasi baru secara benar.

4.2 Komponen Utama Dalam Sistem

Untuk menjalankan sistem yang telah dirancang, dibutuhkan beberapa komponen, antara lain:

4.2.1 *Hardware*

Merupakan suatu komponen yang sangat dibutuhkan dalam mewujudkan sistem yang diusulkan. Dalam hal ini, dapat dirincikan spesifikasi komponen *hardware* yaitu:

1. PC dengan processor minimal Core i31733 MHz
2. *Harddisk* 20 GB
3. Monitor Super VGA
4. *Memory* minimal 128 MB
5. *Keyboard*
6. *Mouse*
7. *Printer*.

4.2.2 *Software*

Hardware tidak akan memecahkan suatu masalah tanpa adanya komponen *software*. Adapun *software* yang sering digunakan dalam pembuatan *website* ini adalah:

1. Sistem operasi Ms. *Windows XP*.
2. *XAMPP Version 1.7.7 software* yang merangkum Apache 2.2.21 sebagai *web server*, PHP 5.3.8 sebagai *web programming* dan *MySQL 5.0.8* sebagai *database server*.
3. Adobe *Dreamweaver CS5* sebagai web editor.

4. Adobe *Photoshop CS5* sebagai *desain layout*.

4.2.3 *Brainware*

Brainware adalah semua pihak yang bertanggung jawab dalam pengembangan informasi, pemrosesan dan penggunaan keluaran informasi.

Brainware dalam sistem ini terbagi atas:

1. Sistem analis, yaitu orang yang menganalisa sistem dengan mempelajari masalah- masalah yang timbul dan menentukan kebutuhan-kebutuhan pemakai dan mengidentifikasi pemecahan yang beralasan.
2. Programmer, yaitu orang yang membuat sistem dengan menggunakan salah satu bahasa pemrograman yang dikuasainya.
3. Operator, yaitu orang yang menggunakan dan memanfaatkan sistem.

4.3 Tampilan Website

4.3.1 Halaman *Log In Admin*

Halaman ini apabila admin ingin melakukan *entry data*, maka admin harus login terlebih dahulu.



Gambar 20. Halaman Log In Admin

4.3.2 Halaman Utama Website

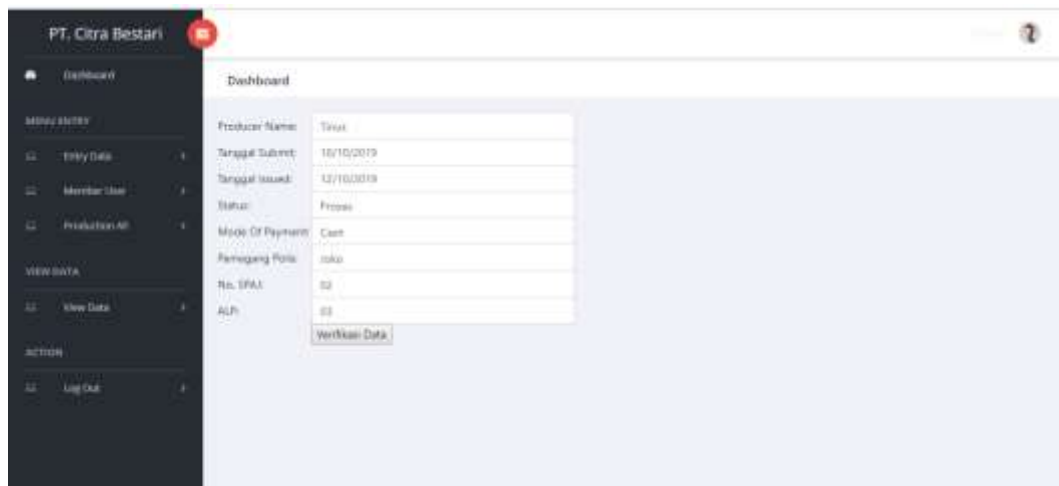
Merupakan halaman utama yang dapat diakses oleh admin dari PT. Citra Bestari .



Gambar 19. Halaman Utama Website

4.3.3 Halaman Entry Data

Halaman ini adalah *entry data* untuk menginputkan data barang awal untuk dilakukannya proses *production*.



Gambar 21. Halaman Registrasi Data

4.3.4 Halaman Member User

Halaman ini adalah halaman member *user* untuk menginputkan member yang berhasil mencari barang untuk dilakukannya proses *production*.

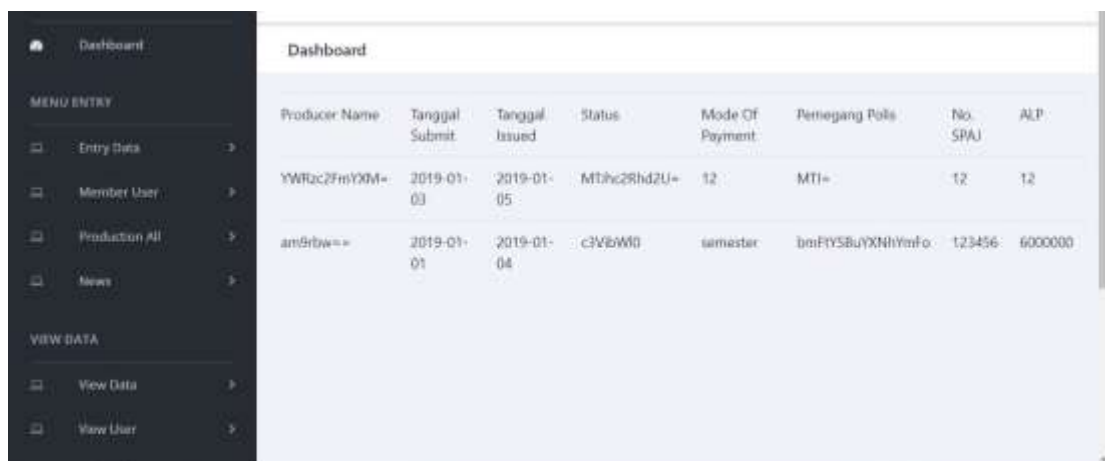


The screenshot shows a dashboard interface with a dark sidebar on the left containing menu options like 'Entry Data', 'Member User', 'Production All', 'News', 'View Data', and 'View User'. The main content area is titled 'Dashboard' and contains a form for adding a member. The form fields are: Nama Member, Level Sub, Di Rekrut Oleh, Kode Producer, Kode Unit, Kode KPM, and No Handphone. A 'Save' button is located at the bottom of the form.

Gambar 22. Halaman *Member User*

4.3.5 Halaman View Data

Halaman ini adalah view data untuk menampilkan data-data barang berserta member *user*.



The screenshot shows the 'View Data' page in the dashboard. It displays a table with the following columns: Producer Name, Tanggal Submit, Tanggal Issued, Status, Mode Of Payment, Pemegang Polis, No. SPAJ, and ALP. The table contains two rows of data.

Producer Name	Tanggal Submit	Tanggal Issued	Status	Mode Of Payment	Pemegang Polis	No. SPAJ	ALP
YWRUc2FhYXNM=	2019-01-03	2019-01-05	MTIhc2Rhd2U=	12	MTI=	12	12
am8rbw=	2019-01-01	2019-01-04	c3VibW0	semester	bmFY5BUYXNhYmFo	123456	6000000

Gambar 23. Halaman *View Data*

BAB V

PENUTUP

5.1 Kesimpulan

Dari penjelasan bab-bab sebelumnya, penulis merangkum beberapa kesimpulan, yaitu:

1. Dengan adanya website dalam perancangan sistem distribusi barang ini, dapat mempermudah proses distribusi barang.
2. Keamanan dengan menggunakan MD5 dalam proses distribusi barang menjadi lebih aman dikarenakan tampilan data distribusi tidak dapat dimanupilasi oleh pihak yang tidak diinginkan.
3. Php dan *MySql* dapat mempermudah dengan menerapkan metode MD5, dikarenakan *Mysql* menyediakan keamanan penyandian ini pada saat proses simpan data di database.

5.2 Saran

Adapun saran dapat diberikan pada Laporan Skripsi ini adalah :

1. Dalam proses distribusi barang sebaiknya dapat di monitor menggunakan android.
2. Website diharapkan dapat di upload dan bersifat online agar dapat di update secara online.

DAFTAR PUSTAKA

- Akbar, A. (2018). Pembangunan Model Electronic Government Pemerintahan Desa Menuju Smart Desa. *Jurnal Teknik dan Informatika*, 5(1), 1-5.
- Ariyus, Kadir. 2006. *Computer Security*. Yogyakarta: Penerbit Andi.
- Arjana, Putu H. dkk. 2012. Implementasi Enkripsi Data Dengan Algoritma LSB. Yogyakarta:
- Azmi, Fadhillah, And Winda Erika. "Analisis Keamanan Data Pada Block Cipher Algoritma Kriptografi Rsa." *Cess (Journal Of Computer Engineering, System And Science)* 2.1: 27-29.
- Bishop, Matt. 2005. *Introduction to Computer Security*. Boston: Addison-Wesley.
- Christensen, Chris.2006. *Steganografi and LSB*. Diakses pada 5 November 2016.
- Erika, Winda, Heni Rachmawati, and Ibnu Surya. "Enkripsi Teks Surat Elektronik (E-Mail) Berbasis Algoritma Rivest Shamir Adleman (RSA)." *Jurnal Aksara Komputer Terapan* 1.2 (2012).
- Hartanto, S. (2017). Implementasi fuzzy rule based system untuk klasifikasi buah mangga. *TECHSI-Jurnal Teknik Informatika*, 9(2), 103-122.
- Harumy, T. H. F., & Sulistianingsih, I. (2016). Sistem penunjang keputusan penentuan jabatan manager menggunakan metode mfep pada cv. Sapo durin. In *Seminar Nasional Teknologi Informasi dan Multimedia* (pp. 6-7).
- Havena, M., & Marlina, L. (2018). The Technology of Corn Processing as an Effort to Increase The Income of Kelambir V Village. *Journal of Saintech Transfer*, 1(1), 27-32.
- Herdianto, H. (2018). Perancangan Smart Home dengan Konsep Internet of Things (IoT) Berbasis Smartphone. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
- Khairul, K., Haryati, S., & Yusman, Y. (2018). Aplikasi Kamus Bahasa Jawa Indonesia dengan Algoritma Raita Berbasis Android. *Jurnal Teknologi Informasi dan Pendidikan*, 11(1), 1-6.
- Leong, Marlon. 2006. *Dari Programmer Untuk Programmer Visual Basic*. Yogyakarta:
- Marlina, L., Muslim, M., Siahaan, A. U., & Utama, P. (2016). Data Mining Classification Comparison (Naïve Bayes and C4. 5 Algorithms). *Int. J. Eng. Trends Technol*, 38(7), 380-383.

- Marlina, L., Putera, A., Siahaan, U., Kurniawan, H., & Sulistianingsih, I. (2017). Data Compression Using Elias Delta Code. *Int. J. Recent Trends Eng. Res*, 3(8), 210-217.
- Martin, Keith. 2012. *Everyday Cryptography*. Oxford: Oxford University Press. McGraw-Hill Book Co.
- Mulyana, Teady. 2012. *Steganografi Citra Digital Menggunakan Spreadsheet*. Vol: 8 No 2 Agustus 2012.
- Muttaqin, Muhammad. "Analisa Pemanfaatan Sistem Informasi E-Office Pada Universitas Pembangunan Panca Budi Medan Dengan Menggunakan Metode Utaut." *Jurnal Teknik Dan Informatika* 5.1 (2018): 40-43.
- Muttaqin, Muhammad. "Portal Academic Portal Innovation Based On Website In The Era Of Digital 4.0 Technology Now."
- Pabokory, Fresly Nandar dkk. 2015. *Implementasi LSB Pengamanan Data Pada Pesan Teks, Isi File Gambar Menggunakan Algoritma Advanced Encryption Standard*. Vol: 10 No 1 Februari 2015.
- Perwitasari, I. D. (2018). Teknik Marker Based Tracking Augmented Reality untuk Visualisasi Anatomi Organ Tubuh Manusia Berbasis Android. *INTECOMS: Journal of Information Technology and Computer Science*, 1(1), 8-18.
- Putri, R. E., & Siahaan, A. (2017). Examination of document similarity using Rabin-Karp algorithm. *International Journal of Recent Trends in Engineering & Research*, 3(8), 196-201.
- Ramadhani, S., Suherman, S., Melvasari, M., & Herdianto, H. (2018). Perancangan Teks Berjalan Online Sebagai Media Informasi Nelayan. *Jurnal Ilmiah Core IT: Community Research Information Technology*, 6(2).
- Rhee, Man Young. 1994. *Library of Congress Cataloging-in-Publication Data*. Singapore: Seminar Nasional Teknologi Informasi dan Komunikasi 2012 (SENTIKA 2012).
- Sutanto, Edhy. 2004. *Algoritma: Teknik Penyelesaian Permasalahan Untuk Komputasi*.
- Wahana Komputer. 2003. *Memahami Model Enkripsi dan Security Data*. Yogyakarta