BAB IV

TUGAS KHUSUS KERJA PRAKTEK

4.1 Pendahuluan

4.1.1 Latar Belakang

PT.Cipta Karya Buana berdiri sejak tahun 2009 dan memulai operasinya pada tahun 2010, perusahan ini menghasilkan berbagai jenis macam sepatu mulai dari bagian *Upper Shoes* (bagian atas sepatu) sampai *Full Shoes* (sepatu utuh) yang tegantung dari permintaan kustomer. PT Cipta Karya Buana dalam menjalankan produksinya juga memperhatikan ketersedian bahan baku yang ada supaya pada saat produksi berjalan tidak sampai kekurangan bahan baku. Oleh karena itu, PT.Cipta Karya Buana dengan cermat melakukan pencataan khususnya pada bagian gudang. Pencataan dimulai dari bahan baku masuk sampai bahan baku keluar atau saat bahan baku telah dikirim ke departemen produksi

Pencatatan yang dilakukan oleh bagian Gudang seperti tanggal bahan baku masuk, quantitas bahan baku yang masuk dan keluar, warna dari bahan baku, jenis dari bahan baku tersebut, ketebalan bahan baku, dan dikirim dari supplier mana. Setelah dilakukan pencataan, bahan baku yang datang kemudian akan diletakan pada rak sesuai dengan kode yang telah diberikan ke bahan baku tersebut. Rak yang tersedia di gudang hanya berjumlah 7 unit dan setiap unit terdiri atas 4 tingkat.

Proses pencataan yang dilakukan pada departemen gudang di PT. Cipta Karya Buana pada saat sekarang masih dilakukan secara manual yang dikerjakan oleh 2 orang operator. Aktivitas pencatatan dari mulai bahan baku masuk ke gudang sampai bahan baku meninggalkan gudang masih dilakukan secara manual. Hal ini dapat menyulitkan operator ketika melakukan pengecekan terhadap persediaan bahan baku yang ada di gudang atau ingin melihat rekap data masa lalu untuk menelusuri kembali data-data yang sudah lama jika terjadi kesalahan, karena harus mencari satu persatu. Disamping itu juga resiko data catatan hilang atau terhapus sangat besar besar.

Berdasarkan hal tersebut di atas, maka tugas khusus yang diberikan oleh perusahaan pada kerja praktek ini adalah merancang sistem informasi dalam gudang untuk memudahkan dalam pencatatan bahan baku mulai datang sampai keluar dari gudang dan juga membantu pencarian informasi mengenai rekap data masa lalu dari bahan baku.

4.1.2 Rumusan Masalah

Bagaimana rancangan sistem informasi yang sesuai untuk memudahkan operator dalam menginput dan mencari informasi mengenai bahan baku di PT. Cipta Karya Buana ?

4.1.3 Tujuan

Merancang sistem informasi yang sesuai untuk operator dalam menginput dan mencari informasi mengenai bahan baku di PT.Cipta Karya Buana.

4.1.4 Batasan Masalah

Program hanya dapat digunakan pada cakupan LAN (*local area network)*

4.1.5 Sistematika Penulisan

Laporan kerja praktek ini memiliki sistematika penulisan sebagai berikut:

1. Bagian 4.1: Pendahuluan,

   Pada bab ini menjelaskan secara umum latar belakang yang diangkat dari tugas khusus di PT. Cipta Karya Buana, rumusan masalah, tujuan penelitian dan sistematika penulisan.
2. Bagian 4.2: Landasan Teori,

   Pada bagian ini berisi tentang teori-teori yang berhubungan dengan tugas khusus yang diberikan perusahaan. Bagian ini akan menunjang serta digunakan untuk menyelesaikan permasalahan serta cara-cara yang akan digunakan dalam melakukan analisis.
3. Bagian 4.3: Metodologi Penelitian

   Pada bagian ini menjelaskan langkah-langkah penelitian yang dilakukan dari awal sampai akhir untuk menyelesikan penelitian.

4. Bagian 4.4: Pengumpulan Data dan Pengolahan Data

   Pada bagian ini berisi penjelasan mengenai tabel relasi, *flowchart* program, *coding* yang digunakan dan bagaimana cara penggunaannya.

5. Bagian 4.5: Analisa Data

   Pada bagian ini berisikan analisa dan interpretasi dari pengolahan data yang diperoleh dari pengolahan data.

6. Bagian 4.6: Kesimpulan dan Saran

   Pada bagian ini berisikan kesimpulan dan saran dari hasil penelitian berdasarkan pengolahan data dan analisa data yang telah dilakukan.

4.2 Landasan Teori

4.2.1 Alat Aplikasi Sistem

Pada perkembangan Sistem Informasi Gudang berbasis website ini, ada beberapa program yang digunakan dalam membuat system. Untuk koding menggunakan bahasa koding PHP, database menggunakan MySQL dengan pengaksesan sistem menggunakan phpMyAdmin, server Apache, dan web browser.

4.2.2 Basis Data

Basis data merupakan suatu data berisi kumpulan data yang saling berhubungan yang mendeskripsikan berbagai entitas dan hubungan antar entitas yang dirancang untuk memenuhi kebutuhan informasi organisasi. *Database* adalah data yang saling terhubung dan deskripsi dari data yang dirancang untuk kebutuhan organisasi (Connolly dan Begg, 2005). Menurut McLeod dan Schell (2004) *database system* adalah sistem penyimpanan informasi yang terorganisasi dengan suatu cara sehingga memudahkan untuk proses pengolahan data.
Tujuan dari pemanfaatan *database* :

1. Efisiensi ruang penyimpanan (*space*)
2. Kecepatan dan kemudahan (*speed*)
3. Keakuratan (*accuracy*)

4. Kelengkapan (*completeness*)
5. Ketersediaan (*availability*)
6. Kebersamaan pemakaian (*sharability*)
7. Keamanan (*security*)

Keuntungan menggunakan *database* :

1. Mengurangi kesalahan yang dapat disebabkan oleh *Humman error*. Tugas mekanis akan lebih baik dilaksanakan oleh mesin.
2. Komputer dapat mengambil maupun mengubah data lebih cepat dari manusia.
3. Akurasi informasi maupun informasi terbaru yang selalu tersedia setiap saat.
4. Menghemat ruangan untuk penyimpanan karena tidak perlu menyediakan ruangan penyimpanan kertas.

4.2.3 *Database Management System* (DBMS)

Database Management System (DBMS) merupakan suatu sistem perangkat lunak yang dapat membantu pemakai / *user* dalam mendefinisikan, merawat, membuat, mengatur dan mengontrol akses pada suatu basis data (Connolly dan Begg,2005).

Fasilitas-fasilitas yang disediakan oleh DBMS adalah sebagai berikut (Connolly dan Begg, 2005) memperbolehkan *user* untuk menambah data, mengubah data, menghapus data, dan mencari data. Biasanya dengan menggunakan suatu Data Manipulation Language (DML). Biasanya ada suatu fasilitas untuk melayani pengaksesan data yang di sebut sebagai Query Language. Bahasa query yang paling diakui adalah Structured Query Language (SQL), yang secara faktanya merupakan standar bagi DBMS.

Database memiliki akses pengkontrolan seperti:

1. Sistem keamanan(*security system*)
   Untuk mencegah file – file pada database tidak dapat diakses oleh pihak yang tidak berwenang.
2. Sistem terintegrasi (*integrity system*)
   Untuk menjaga konsistensi data.
3. Sistem kontrol pengembalian(*Recovery Control System*)
   Untuk memperbaiki data-data jika terjadi kerusakan-kerusakan pada perangkat keras atau perankgat lunak.
4. Katalog yang dapat diakses user (*User accessible catalog*)
   Suatu catatan dari deskripsi data-data dalam database.

Keuntungan menggunakan DBMS meenurut Connolly dan Begg (2005) keuntungan dari DBMS antara lain :

1. Dapat menghindari data yang sama disimpan berulang kali (*control of data redundancy*).
2. Data menjadi konsisten.
3. Informasi yang lebih banyak bisa didapatkan dari kumpulan data yang sama, dengan permasalahan tertentu.
4. Data yang sama dapat di gunakan bersama-sama oleh pemakai yang berbeda-beda (*shared data*).
5. Meningkatkan keamanan akses terhadap data.
6. Penetapan Standarisasi.
7. Kebutuhan-kebutuhan yang berbeda-beda dapat di penuhi dengan mudah.
8. Meningkatkan kemampuan akses data dan kecepatan prosesnya.
9. Meningkatkan produktifitas.
10. Meningkatkan intregitas data.
11. Layanan *back up* dan *recovery* yang baik.

Sedangkan kerugian penggunaan DBMS menurut Connolly dan Begg (2005) antara lain :

1. Proses didalamnya yang rumit.
2. Mempunyai kapasistas yang lebih di memori.
3. Biaya dari suatu DBMS.
4. Membutuhkan biaya tambahan untuk *Hardware* jika DBMS yang digunakan menuntut penggunaan jenis DBMS tertentu / khusus.
5. Kinerja aplikasi tidak dapat berjalan dengan cepat seperti seharusnya karena adanya DBMS
6. Mengurangi performa dari penggunaan aplikasi.
7. Kegagalan dapat berdampak besar

Berikut ini adalah tahap-tahap pengembangan aplikasi database yang terdiri atas beberapa tahapan yang terdiri dari :

1. Perencanaan *database*

   Merancanakan bagaimana tahapan siklus hidup database dapat dilaksanakan dengan baik
2. Definisi *system*

   Menspesifikasikan ruang lingkup dan batasan-batasan dari aplikasi database, penggunaanya dan area cakupan aplikasi.
3. Mengumpulkan dan menganalisis kebutuhan

   Mengumpulkan dan menganalisa kebutuhan dari pemakai dan area aplikasi.
4. Perancangan *database*

   Perancangan database yang dilakukan secara konseptual, logikal, dan fisikal.
5. Pemilihan DBMS

   Memilih DBMS yang sesuai sehingga dapat diaplikasikan ke *system* yang kita buat.
6. Perancangan aplikasi

   Mendesain *user interface* dan program-program aplikasi yang memakai *databse* maupun memproses *database*.

7. Pembuatan *prototype*

   Membuat model kerja untuk aplikasi database yang dapat digunakan oleh perancang atau pengguna untuk memvisualisasikan dan mengevaluasi bagaimana sistem akan terlihat dan bagaimana sistem tersebut akan berfungsi.
8. Implementasi

   Membuat definisi database secara eksternal, konseptual, dan internal melalui program-program aplikasi.
9. Data *conversion* dan *loading*

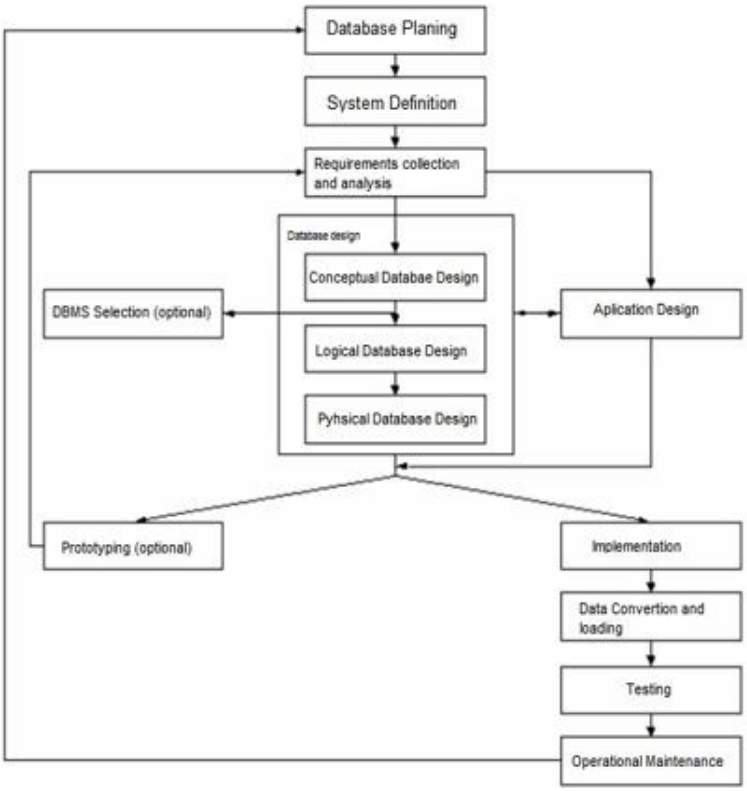
   Memindahkan data dari sistem yang lama ke dalam sistem yang baru.
10. Melakukan *testing*

    Pengujian pada aplikasi *database* yang telah dibuat untuk mencari kesalahan dan menunjukkan apakah database dan DBMS dapat bekerja dengan baik dengan kebutuhan *user*.
11. *Operational Maintenance* (Perawatan Operasional)

    Aplikasi *database* yang telah diimplementasikan akan terus dimonitor dan diajaga. Bahkan jika dibutuhkan, kebutuhan baru akan ditambahakan kedalam aplikasi database melalui tahapan dari siklus hidup sebelumnya.

Berikut ini langkah-langkah dalam rangka perawatan operasional:

1. Mengawasi kinerja sistem.
2. Mempertahankan dan meng-*upgrade* aplikasi basis data (ketika dibutuhkan)



Gambar 4.1 Tahap-tahap pengembangan aplikasi *database*

4.2.4 *Entity Relationship Modeling*

Entity-Relationship Modeling merupakan suatu bentuk pendekatan *top-down* dalam perancangan databse yang dimulai dengan identifikasi data-data penting yang disebut entitas dan relasi antara data-data tersebut harus direpresentasikan dalam bentuk model. (Connolly dan Begg, 2005).

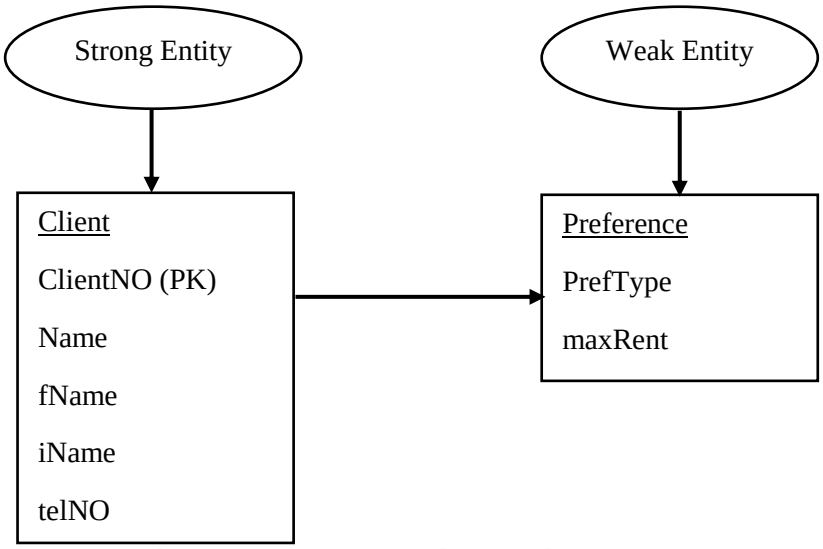
Berikut ini penjelasan konsep dasar dari *ER Modeling*, sebagai berikut:

4.2.4.1 *Entity types*

Menurut Connolly dan Begg (2005), tipe entitas adalah sekumpulan objek dengan *property* yang sama yang di identifikasikan oleh perusahaan dan memiliki keberadaan yang bebas atau tidak tergantung.

Entity Occurance adalah uatu objek dari tipe entitas yang dapat diidentifikasikan secara unik. Tipe entitas dibedakan menjadi dua yaitu :

1. ***Strong Entity type*** yaitu tipe entitas yang keberadaanya tidak tergantung kepada entitas yang lain.
2. ***Weak Entity Type*** yaitu entitas yang keberadaannya bergantung kepada tipe entitas lain.



Gambar 4.2 *Strong Entity* dan *Weak Entity*

Sumber : (Connolly dan Begg, 2005)

4.2.4.2 *Relationship types*

Menurut Connolly dan Begg (2005), tipe *relationship* atau hubungan merupakan suatu kumpulan atas asosiasi yang memiliki arti di antara tipe entitas. Tiap-tiap tipe *relationship* diberikan nama yang menjelaskan fungsinya masing-masing. Derajat dari tipe relationship adalah banyaknya tipe entitas yang ikut serta di dalam suatu *relationship*.

4.2.4.3 *Attributes*

Merupakan sifat-sifat dari sebuah entitas atau *relationship type*. Contohnya: sebuah entitas *Staff* digambarkan oleh atribut staffNo, *name, position,* dan *salary*.

Attribute Domain adalah himpunan nilai yang diperbolehkan untuk satu atau lebih atribut. Macam-macam atribut :

a. *Simple Attribute*, yaitu atribut yang terdiri dari satu komponen tunggal dengan keberadaan yang independen dan tidak dapat dibagi menjadi bagi lagi. Dikenal juga dengan nama *Atomic Attribute*.
b. *Composite Attribute*, yaitu atribut yang terdiri dari beberapa komponen, dimana masing-masing komponennya independen. Misalkan *Attribute Address* dapat terdiri dari *Street, City, PostCode*.
c. *Single-valued Attribute*, yaitu atribut yang mempunyai beberapa nilai untuk setiap kejadian. Misal entitas *Branch* memiliki satu nilai untuk atribut *branchNo* pada setiap kejadian.
d. *Multi-valued Attribute*, yaitu atribut yang mempunyai beberapa nilai untuk setiap kejadian. Misal entitas *Branch* memiliki beberapa nilai untuk atribut telpNo pada setiap kejadian.
e. *Derived Attribute*, yaitu atribut yang memiliki nilai yang dihasilkan dari satu atau beberapa atribut lainnya, dan tidak harus berasal dari satu entitas.

4.2.4.4 *Keys*

Menurut Connolly dan Begg (2005), *Keys* berperan penting dalam menghubungkan suatu obyek dengan obyek yang lainnya. *Keys* diletakkan pada suatu atribut yang telah ditentukan kedudukannya, agar dapat dihubungkan dengan atribut pada entitas lain.

Beberapa jenis *keys* yang sering digunakan, menurut Connolly dan Begg (2005), antara lain:

1. **Candidate key**

   Candidate key yaitu himpunan atribut minimal yang secara unik mengidentifikasikan tiap-tiap keberadaan suatu tipe entitas.

2. **Primary key**

   Primary key yaitu *candidate key* yang dipilih secara unik untuk mengidentifikasikan tiap-tiap keberadaan suatu tipe entitas.

3. **Foreign key**

   Foreign key adalah Sebuah kumpulan field dalam satu relasi yang digunakan untuk menunjuk ke suatu baris pada relasi yang lain (harus berkorespondensi dengan primary key pada relasi yang kedua)

4. **Alternate key**

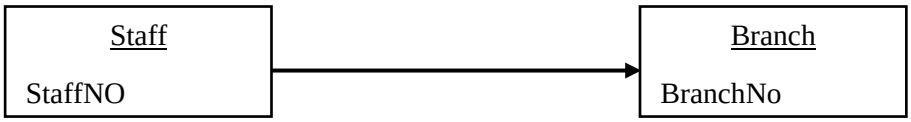
   Alternate key yaitu *candidate key* yang tidak terpilih menjadi *primary key*.

4.2.4.5 *Structural Constraints*

Multiplicity merupakan jumlah dari kejadian yang mungkin dari sebuah tipe entitas yang berhubungan kepada sebuah kejadian tunggal dari tipe entitas yang terasosiasi melalui *relationship* (hubungan) tertentu (Connoly dan Begg, 2005),. Derajat yang bisa digunakan untuk *relationship* (hubungan) adalah *binary*. *Binary relationship* terdiri atas :

1. *One to One* (1:1) *Relationship*

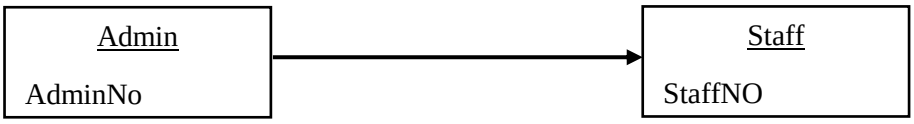
   Hubungan setiap entitas yang ada hanya dapat mempunyai maksimal satu hubungan dengan entitas yang lain. Contohnya entitas *Staff* dan suatu kejadian entitas *Branch*. Dengan kata lain, seorang staf dapat mengatur nol atau satu b*ranch* dan setiap *branch* diatur satu orang *staff*.



Gambar 4.3 *One to One Relationship*

2. *One to Many* (1:*) *Relationship*

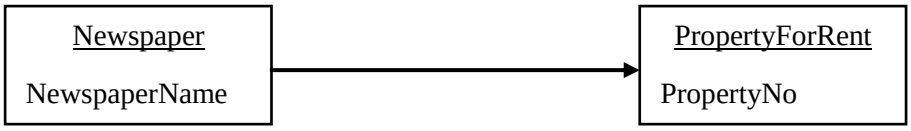
   Hubungan setiap *entity* yang dapat mempunyai satu atau lebih dari satu hubungan dengan *entity* yang lain. Contohnya entitas *Staff* dan satu kejadian entitas admin. Dengan kata lain, seorang admin dapat mengawasi nol atau lebih seorang *staff* tetapi seorang *staff* hanya dapat diawasi oleh seorang admin.



Gambar 4.4 *One to Many Relationship*

3. *Many to Many* (*:*) *Relationship*

   Hubungan setiap *entity* dapat mempunyai lebih dari satu relasi dengan *entity* lainnya. Contohnya entitas *Newspaper* dan satu kejadian entitas *PropertyForRent*. Dengan kata lain, satu *newspaper* mengiklankan satu atau lebih properti dan satu property diiklankan pada nol atau lebih *newspaper*.



Gambar 4.5 *Many to Many Relationship*

4.2.5 PHP (Hypertext Preprocessor)

4.2.5.1 Definisi PHP

PHP *Hypertext Preprocessor* merupakan bahasa *script server-side* yang digunakan untuk mengembangkan Web yang disisipkan pada dokumen HTML. Penggunaan PHP memungkinkan Web agar dapat dibuat dinamis sehingga perawatan situs Web tersebut menjadi lebih mudah dan efisien. PHP merupakan

Software Open-Source yang disebarkan dan dilisensikan secara gratis serta dapat didownload secara bebas dari situs resminya *http://www.php.net*. PHP ditulis menggunakan bahasa pemrograman C (Peranginangin, 2006)

4.2.5.2 Keuntungan Penggunaan PHP

PHP memiliki banyak kelebihan yang tidak dimiliki oleh bahasa *script* sejenis, sehingga beberapa keuntungan yang didapatkan dari penggunaan PHP sebagai *development tool*, yaitu (Peranginangin, 2006):

a. PHP adalah teknologi yang paling superior. Teknologi lain yaitu *PERL, Phyton, Tcl, VB script, ASP* merupakan teknologi yang lebih rendah / dibawah php dan lebih tua. Bahkan Java/Jsp lebih rendah dari PHP.
b. Open Source
c. Sintaks-sintaksnya lebih mudah dipelajari olah pemakai, sangat menyerupai C dan *Perl*.
d. Integrasi yang sangat luas ke berbagai server *database*. Menulis web yang terhubung ke database menjadi sangat sederhana. Database yang didukung oleh PHP: Oracle, Sybase, mSQL, MySQL, Solid, ODBC, PostgreSQL, Adabas D, FilePro, Velocis, Informix, dBase, dan UNIX dbm.
e. *Script* (kode program) terintegrasi dengan file HTML, sehingga *developer* bisa berkonsentrasi langsung pada penampilan dokumen webnya.
f. Tidak ada proses compiling dan linking.
g. Berorientasi obyek *(object oriented).*

4.2.5.3 Tipe Data

PHP memiliki beberapa tipe data. Empat adalah tipe skalar (*single-value*); *integer, floating-point number, string* ,dan *booleans*. Dua adalah tipe campuran (koleksi) ; *array* dan *object*. Sisanya adalah jenis khusus ; *resource* dan NULL.

4.2.6 My SQL

4.2.6.1 Definisi MySQL

MySQL merupakan salah satu jenis untuk database server yang sangat terkenal. Kepopuleran MySQL disebabkan dari penggunaan SQL sebagai bahasa dasar untuk mengakses *database*-nya (Abdul Kadir, 2008). MySQL adalah *Relational Database Management Sistem* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL *(General Public License).* Setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat closed source atau komersial (Prasetyo, 2003). MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: basis data *management system*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai *software* gratis di bawah lisensi GNU *General Public License* (GPL), tetapi programer juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

4.2.6.2 Fitur MySQL

Sebagai software DBMS, MySQL memiliki sejumlah fitur seperti berikut (Kadir, 2008):

1. *Multiplatform* MySQL tersedia untuk beberapa sistem operasi seperti: Windows, Linux, dan Unix.
2. Cepat dan Mudah digunakan. MySQL tergolong sebagai database server (server yang melayani permintaan terhadap *database)* yang handal, dapat menangani *database* besar dengan kecepatan tinggi, mendukung banyak sekali fungsi untuk mengakses *database*, sekaligus mudah untuk digunakan.
3. Jaminan Keamanan akses MySQL mendukung pengamanan database dengan berbagai kriteria pengaksesan. Sebagai gambaran, dimungkinkan untuk mengatur user tertentu agar bisa mengakses data yang bersifat rahasia (misalnya gaji pegawai), sedangkan user lain tidak

boleh. MySQL juga mendukung konektivitas ke berbagai *software*. Sebagai contoh, dengan menggunakan ODBC *(Open Database Connectivity)*, *database* yang ditangani MySQL dapat diakses melalui program yang dibuat dengan *Visual Basic*. MySQL juga dapat mendukung program klien yang berbasis Java untuk berkomunikasi dengan *database* MySQL melalui JDBC *(Java Database Connectivity).* MySQL juga bisa diakses melalui aplikasi berbasis Web, misalnya dengan menggunakan PHP

4. Dukungan SQL Seperti tersirat dalam namanya, MySQL mendukung perintah SQL (*Structured Query Language*). Sebagaimana diketahui, SQL merupakan standar dalam pengaksesan *database* relasional.

4.2.7 phpMyAdmin

PhpMyAdmin merupakan sebuah perangkat lunak gratis yang ditulis dengan PHP yang digunakan untuk menangani administrasi MySQL di World Wide Web seperti yang dituliskan dalam website resminya www.phpmyadmin.net. Aplikasi phpMyAdmin dapat mengatur sebuah server MySQL (membutuhkan sebuah super-user) sama sepeti sebuah database tunggal. Untuk menjalankan script MySQL, pengguna perlu mengatur *privilege* bagi *user* yang menggunakan aplikasi ini, agar user tersebut dapat membaca/mengubah hanya database tertentu.

4.2.8 Apache

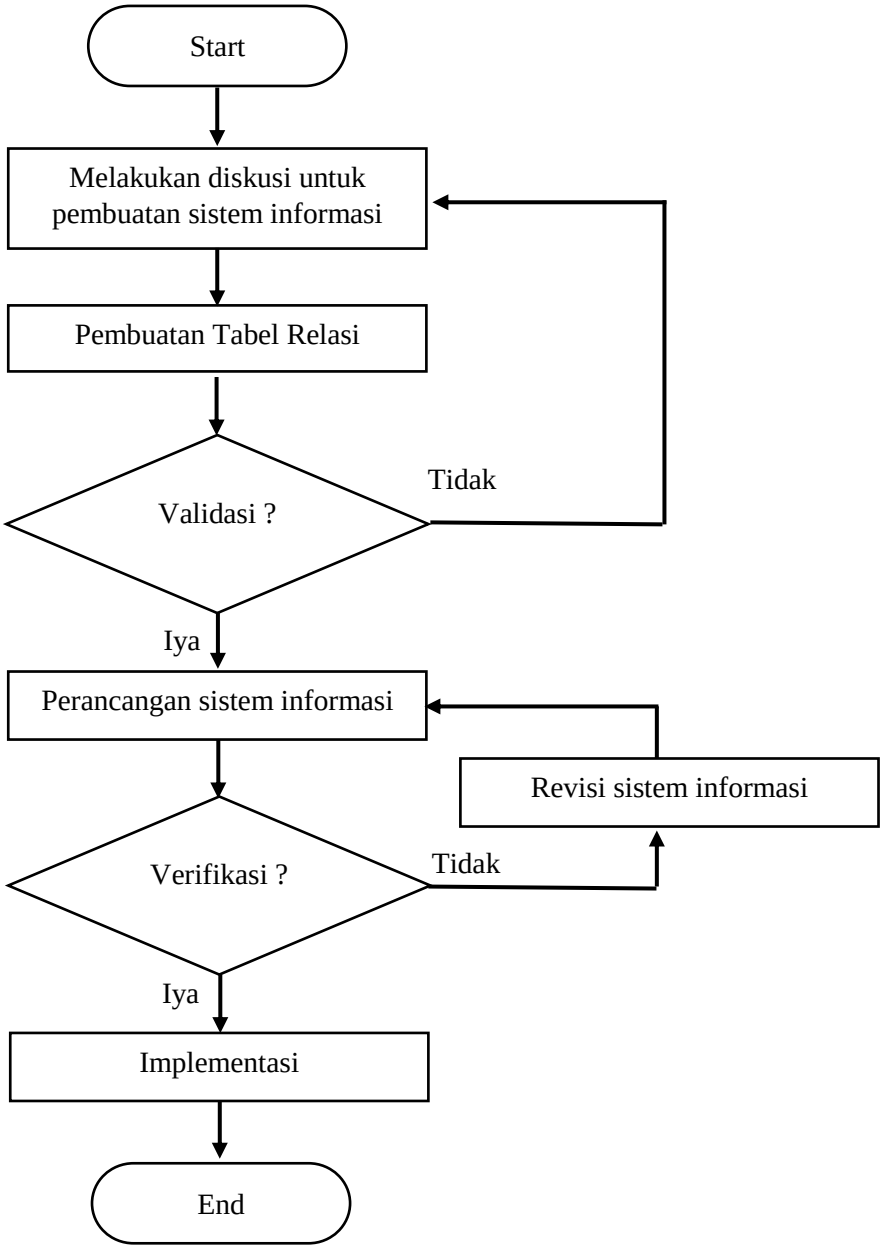
Web server Apache berbasiskan *Open Source* dan mulai populer di Internet sejak tahun 1996. *Software* ini dapat diunduh pada situs http://www.apache.org dan tersedia untuk berbagai platform (Windows, Linux, dan UNIX). Supaya dokumen-dokumen web baik berupa HTML ataupun PHP bisa diakses oleh browser maka dokumen-dokumen tersebut perlu diletakkan dalam direktori khusus yang diatur oleh Apache. Oleh karena itu, semua *script* PHP yang digunakan perlu diletakkan di dalamnya (Kadir, 2008).

4.2.9 Web Browser

Web browser sebagai perangkat lunak yang berguna untuk mengakses informasi web ataupun untuk melakukan transaksi via web. Beberapa contoh browser yang ada saat ini seperti : Internet Explorer, Mozilla Firefox, Opera, Netscape, dan Safari (Abdul Kadir, 2003).

4.3 Metodologi Penelitian

Dalam penelitian diperlukan langkah-langkah yang dilakukan dari awal sampai akhir untuk menyelesikan penelitian, ini dapat dilihat dalam bentuk *flowchart* dibawah ini.



Gambar 4.6 *flowchart* Metodologi Penelitian

4.3.1 Diskusi pembuatan sistem informasi

Tahap awal dilakukan diskusi dengan pembimbing pabrik mengenai proses pencatatan yang ada digudang dari barang masuk sampai barang meninggalkan gudang. Juga mengenai apa saja yang harus dicatat jika barang masuk dan barang keluar seperti kuantitas, warna, tebal, nama barang, kode barang, jenis dll. Lalu output yang diperlukan oleh divisi lain.

4.3.2 Pembuatan Tabel Relasi

Tahap ini dilakukan dengan mengelompokan informasi-informasi yang ada untuk pembuatan sistem informasi dalam bentuk tabel-tabel. Tabel-tabel tersebut disebut entitas yang saling berhubungan satu dengan yang lain, dalam tabel terdapat entitas dan atribut.

4.3.3 Validasi

Tahap ini dilakukan untuk mengetahui apakah tabel relasi yang telah dibuat sudah menggambarkan secara nyata keadaan sistem pencatatan yang ada digudang. Jika tidak tervalidasi maka akan dilakukan diskusi dengan pembimbing lapangan kerja praktek untuk mendapat tabel relasi yang sesuai.

4.3.4 Perancangan Sistem Informasi

Tahap ini merupakan tahap perancangan sistem informasi dari tabel-tabel relasi yang sudah tervalidasi supaya sistem informasi yang ada dapat di implementasikan. Sistem informasi yang dirancang berupa web browser yang dapat menyimpan data-data di gudang. *Software* yang digunakan pada perancangan sistem informasi ini adalah Dreamweaver, basis data MySql dan bahasa pemrograman menggunakan PHP

4.4.4 Verifikasi

Tahap ini dilakukan untuk mengetahui apakah sistem informasi yang dirancang sudah dapat di aplikasikan atau belum, dan melakukan pengecekan logika pembuatan sistem informasi apakah masih error atau tidak. Jika tidak

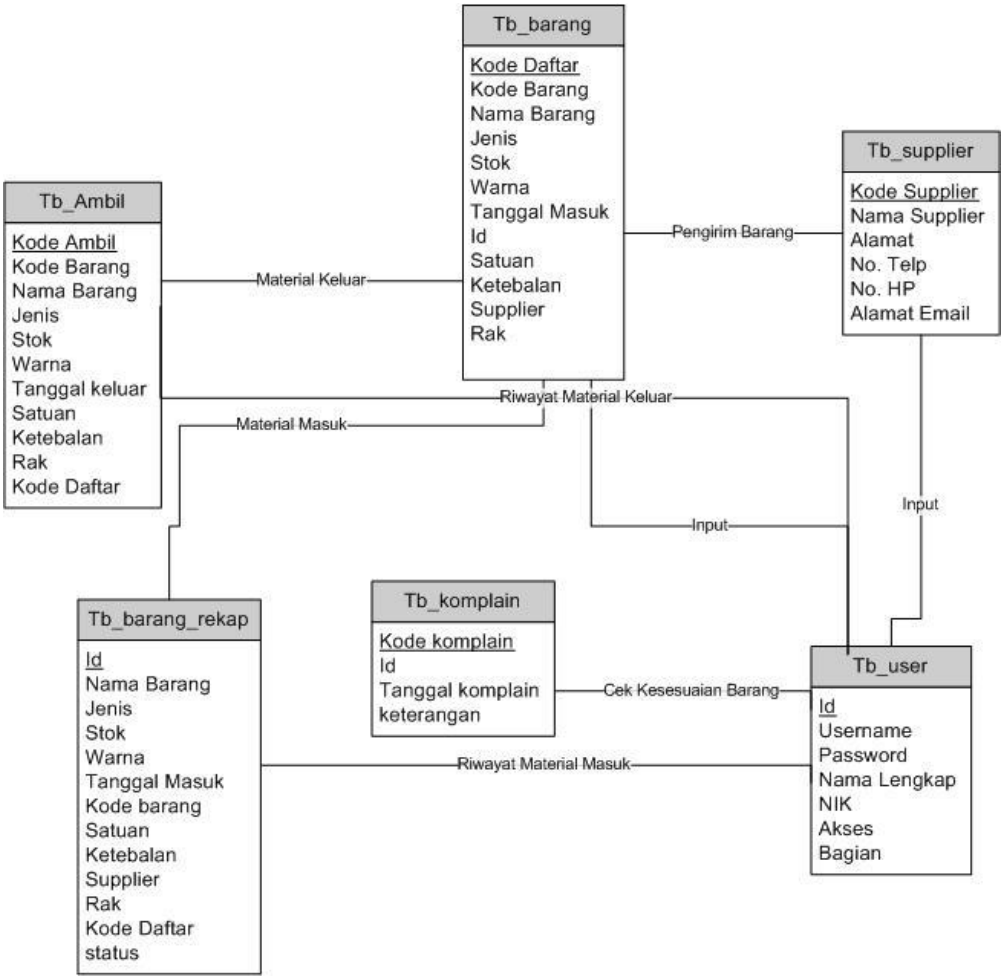
terverifikasi maka akan dilakukan revisi ulang terhadap koding sistem informasi yang telah dikembangkan.

4.4.5 Implementasi

Seletah sistem informasi terverifikasi tahap terakhir adalah implementasi. Pada tahap ini sistem informasi yang telah dirancang dengan berbagai tahap yang telah dilalui akan di implementasikan pada perusahaan (divisi gudang) agar dapat langsung di operasikan oleh operator atau karyawan.

4.4 Pengumpulan Data dan Pengolahan Data

4.4.1 ERM (*Entity Relationship Modeling)*



Gambar 4.7 *Entity Relationship Modeling*

Gambar diatas merupakan gambar *ERM* / *Entity Relationship Modeling*. Pada gambar diatas, terdapat beberapa tabel entitas yang digunakan pada program yang dibuat. Terdiri dari Tb_ambil, Tb_barang, Tb_barang_rekap, Tb_supplier, Tb_komplain dan Tb_user. Setiap Tabel tersebut memiliki attributnya masing – masing. Pada tabel di atas, atrribut yang diberi garis *underline* merupakan *primary key* untuk tabel tersebut. Dimana *primary key* sendiri menjadi identitas utama yang membedakan suatu data dengan data yng lainnya. Berikut adalah jenis relasi antar tabel satu dengan tabel yang lain.

Tabel 4.1 Jenis Relasi Antar Tabel

Hubungan	Jenis Relasi
Tb_barang rekap dan Tb_user	Many to Many
Tb_supplier dan Tb_user	Many to Many
Tb_supplier dan Tb_barang	Many to Many
Tb_ambil dan Tb_barang	Many to Many
Tb_ambil dan Tb_user	Many to Many
Tb_barang_rekap dan Tb_barang	Many to Many
Tb_barang dan Tb_user	Many to Many
Tb_user dan Tb_Komplain	Many to Many

4.4.2 Tabel *Database*

Tabel 4.2 tb_ambil

tb_ambil			
No	**Name**	**Type**	**Length**
1	kode_ambil	varchar	10
2	kode_barang	varchar	10
3	nama_barang	varchar	255
4	Jenis	varchar	50
5	Stok	int	10
6	Satuan	varchar	50
7	Warna	varchar	30
8	Ketebalan	varchar	50
9	tanggal_keluar	date	
10	Rak	varchar	100
11	kode_daftar	varchar	20

Tabel 4.3 tb_barang

tb_barang			
No	**Name**	**Type**	**Length**
1	kode_daftar	varchar	10
2	kode_barang	varchar	10
3	nama_barang	varchar	255
4	Jenis	varchar	50
5	Stok	int	10

Tabel 4.3 lanjutan

tb_barang			
No	**Name**	**Type**	**Lenght**
6	Satuan	varchar	50
7	Warna	varchar	30
8	Ketebalan	varchar	50
9	tanggal_masuk	date	
10	Rak	varchar	100
11	Supplier	varchar	100
12	Id	int	100

Tabel 4.4 tb_barang_rekap

tb_barang_rekap			
No	**Name**	**Type**	**Length**
1	Id	int	100
2	kode_barang	varchar	10
3	nama_barang	varchar	255
4	Jenis	varchar	50
5	Stok	int	10
6	Satuan	varchar	50
7	Warna	varchar	30
8	Ketebalan	varchar	50
9	tanggal_masuk	date	
10	Rak	varchar	100
11	Supplier	varchar	100
12	kode_daftar	varchar	10

Tabel 4.5 tb_supplier

tb_supplier			
No	**Name**	**Type**	**Length**
1	kode_supplier	varchar	100
2	nama_supplier	varchar	50
3	Alamat	varchar	200
4	no_telp	varchar	50
5	bo_hp	varchar	20
6	alamat_email	varchar	50

Tabel 4.6 tb_user

Tb_user			
No	**Name**	**Type**	**Length**
1	Id	int	11
2	Username	varchar	50
3	Password	varchar	50
4	nama_lengkap	varchar	50
5	NIK	int	10
6	Akses	varchar	100
7	Bagian	varchar	100

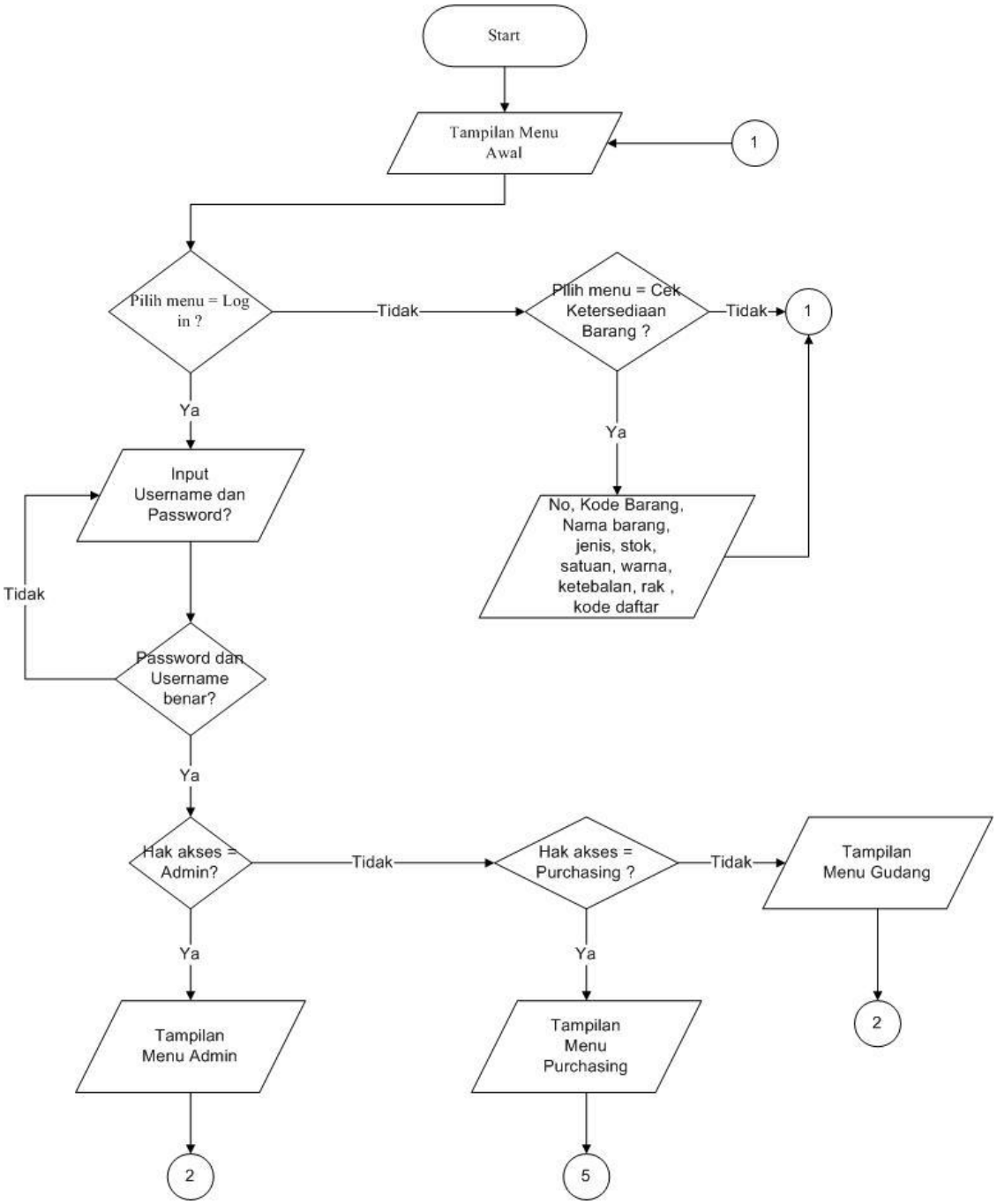
Tabel 4.7 tb_komplain

tb_komplain			
No	**Name**	**Type**	**Length**
1	Kode_komplain	varchar	10
2	Tanggal_komplain	date	
3	id	int	10
4	keterangan	varchar	500

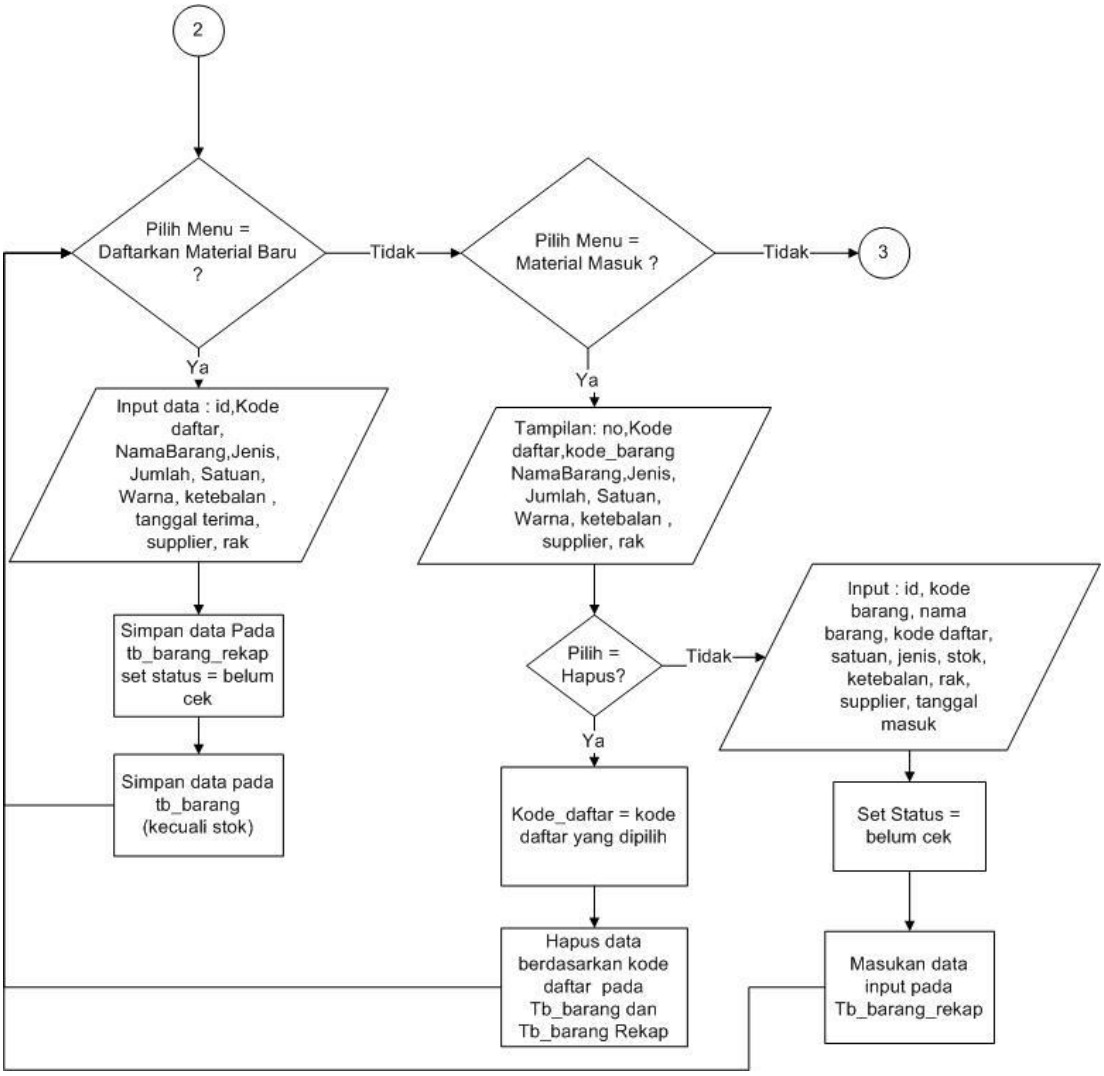
Keterangan:

- Varchar : Tipe data yang terdiri dari huruf maupun angka
- Int (*interger)* : Tipe data berupa data numerik
- Date : Data berupa format tanggal
- Length : Merupakan batasan banyaknya jumlah karakter yang dapat di tulis

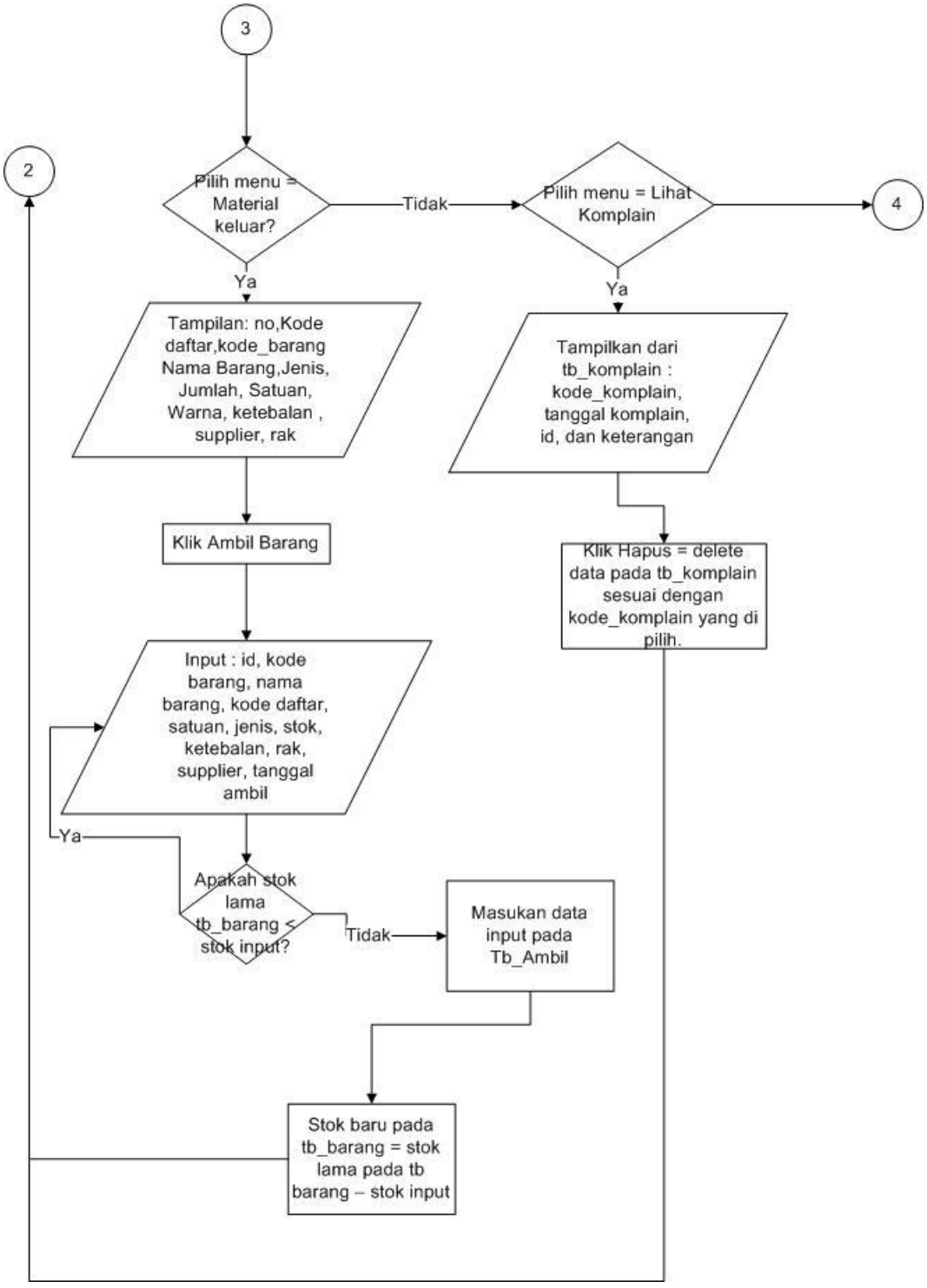
4.4.3. *Flowchart* Program



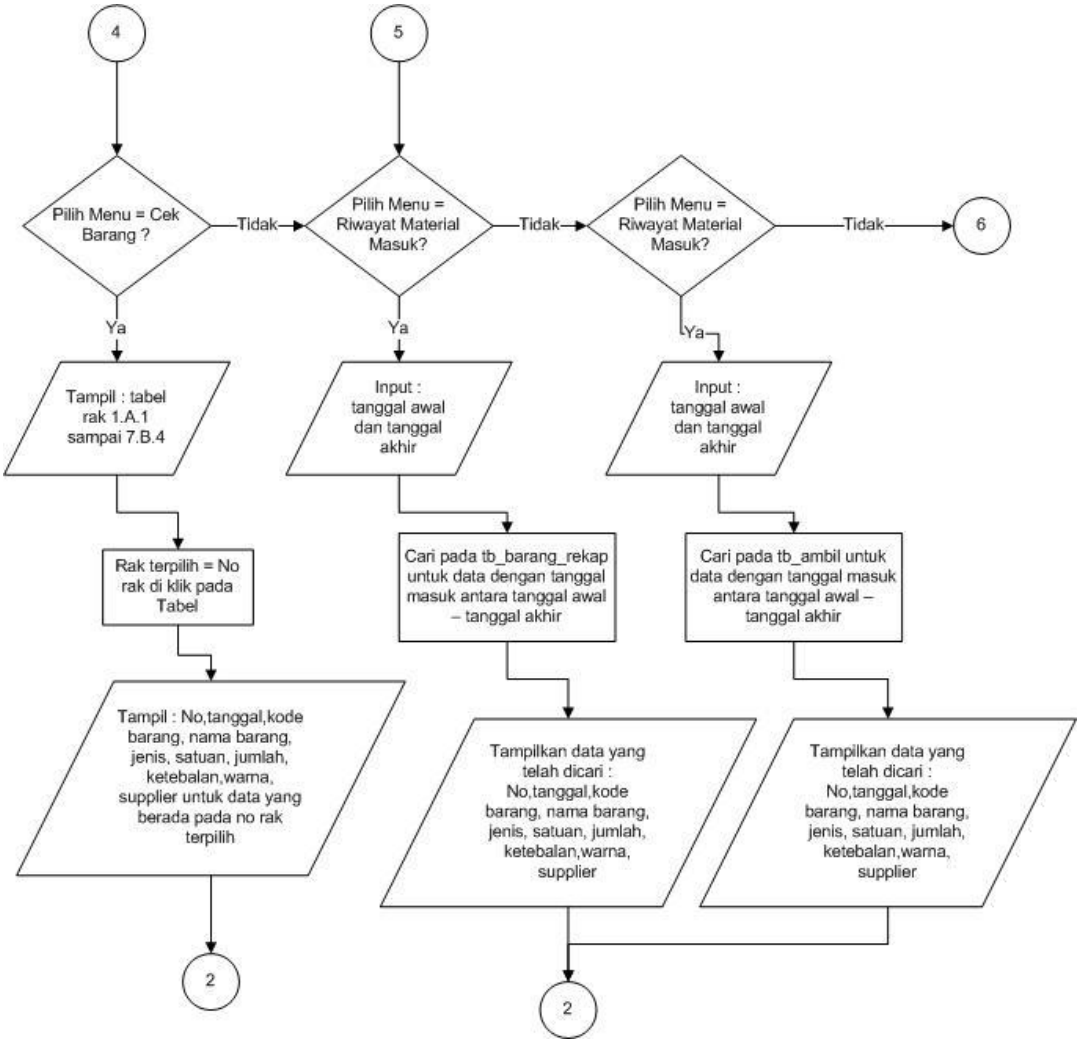
Gambar 4.8 *Flowchart* Program 1



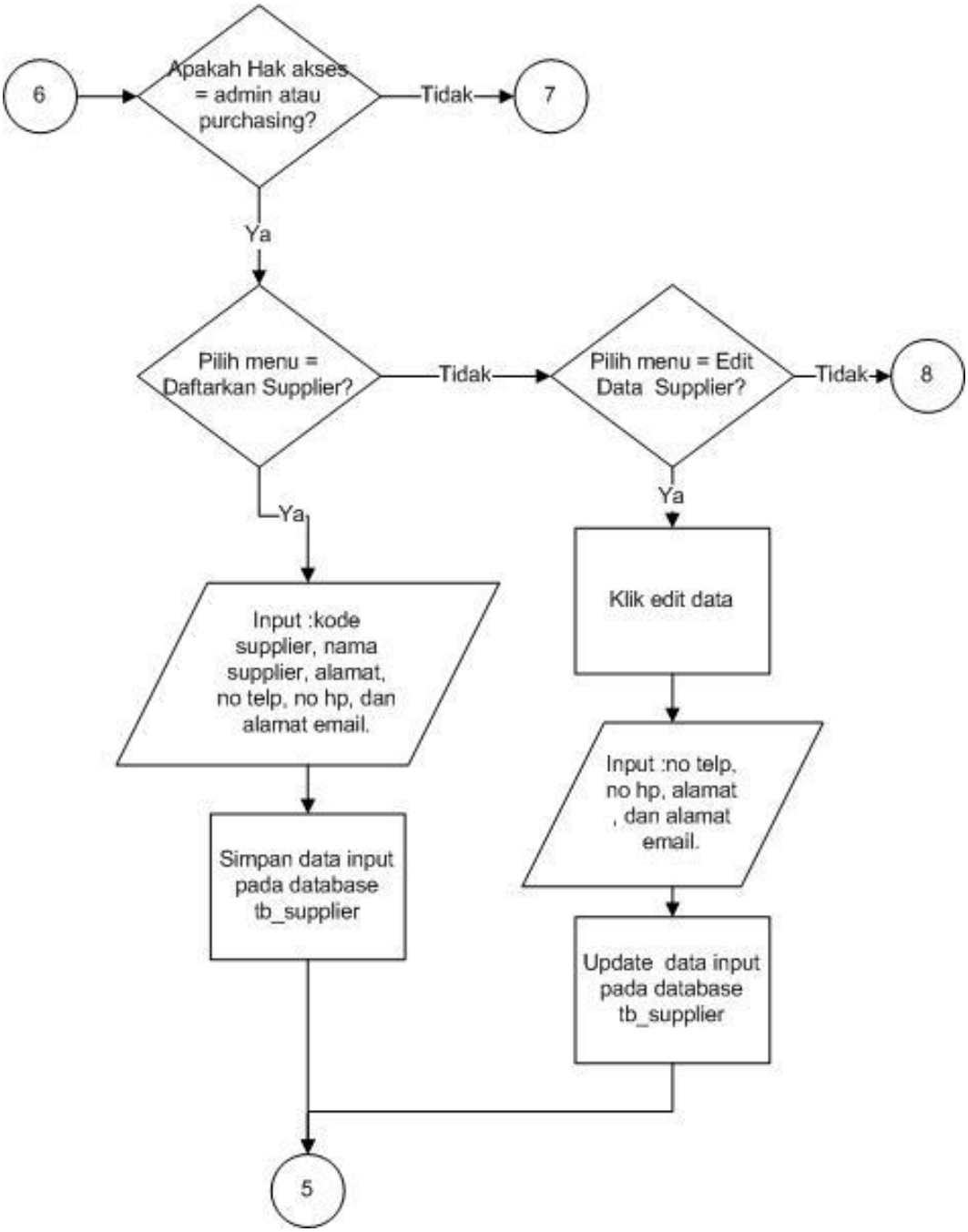
Gambar 4.9 *Flowchart* Program 2



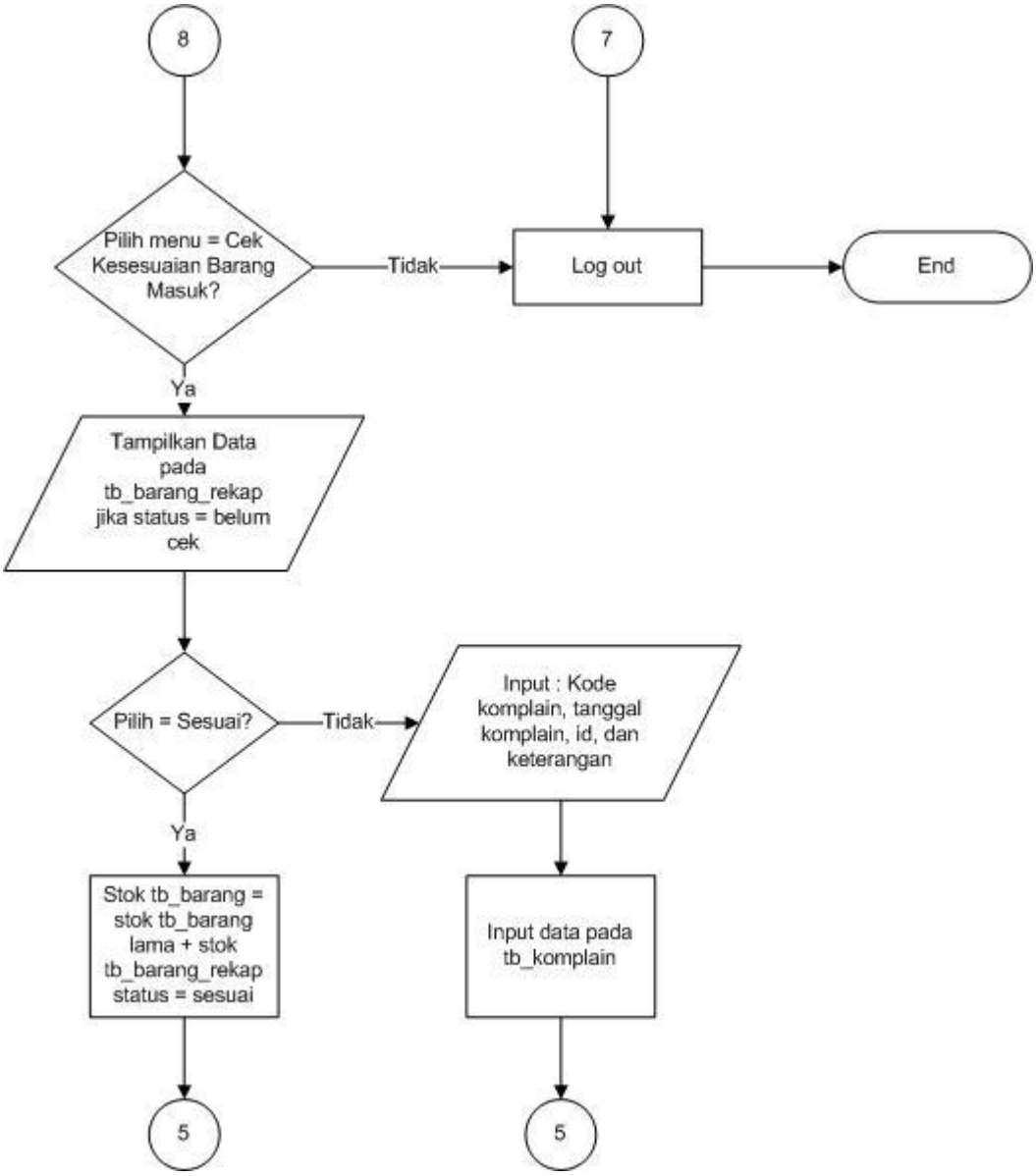
Gambar 4.10 *Flowchart* Program 3



Gambar 4.11 *Flowchart* Program 4



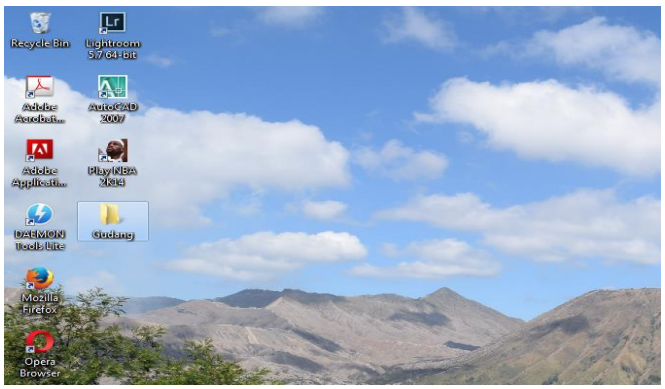
Gambar 4.12 *Flowchart* Program 5



Gambar 4.12 *Flowchart* Program 6

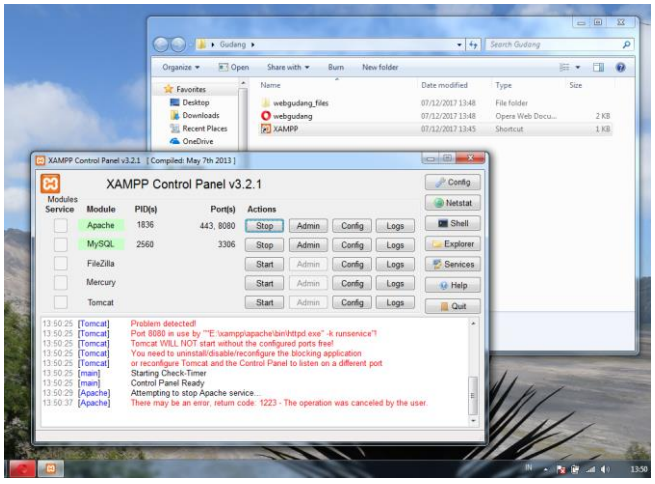
4.4.4. SOP *(standart operation process)* Sistem Informasi Gudang

A. Membuka Program



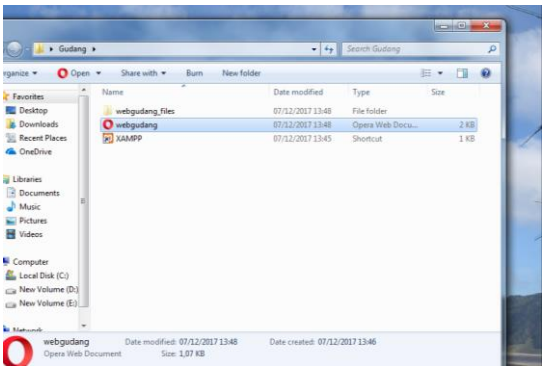
Gambar 4.13 Tampilan Folder Gudang

1. Buka Folder “Gudang” yang ada pada desktop dengan mengklik 2 kali pada folder.

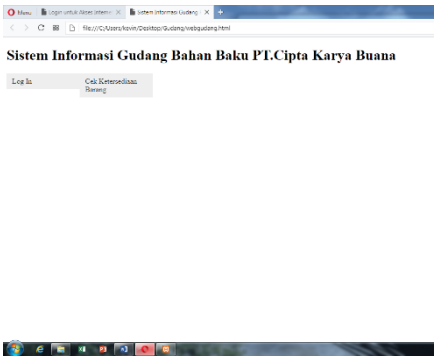


Gambar 4.14 Tampilan *Software* Xampp

2. Buka aplikasi XAMPP
3. Saat program XAMPP terbuka, silahkan mengklik tombol Start untuk module Apache dan MySQL.Lalu minimize program XAMPP (Jangan di close!).



Gambar 4.15 Tampilan file Webgudang



Gambar 4.16 Tampilan program

4. Setelah itu dapat membuka file "webgudang" yang ada pada folder "gudang" untuk membuka program. Atau dapat dilakukan dengan membuka web browser yang ada pada komputer, lalu menuliskan alamat web: localhost/gudang

B. Log In

Gambar 4.17 Tampilan menu Awal

1. Log ini dilakukan jika ingin melakukan memasukkan data barang keluar, barang masuk, dll. Untuk melakukan log in dapat dilakukan dengan cara klik "log in" pada menu bar di tampilan awal.
2. Lalu akan muncul halaman baru dan akan diminta untuk mengisikan username dan password.

Gambar 4.18 Tampilan menu *log in*

3. Isikan username pada kolom username dan password pada kolom password dengan benar (perhatikan penggunakan huruf besar dan kecil maupun tanda baca).

Gambar 4.19 Tampilan tombol *log in*

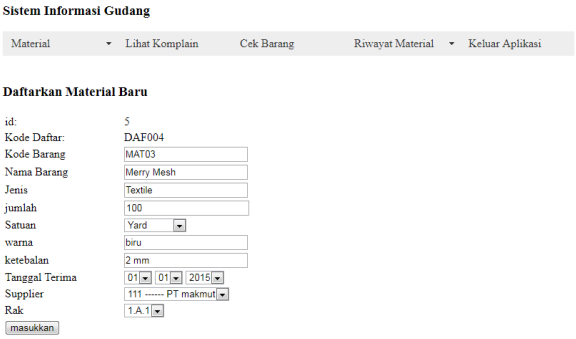
4. Jika telah terisi, lalu dapat menekan enter / klik tombol “masuk” yang ada pada halaman web.

A. Daftarkan Data Material Baru

1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.

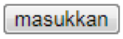
Gambar 4.20 Tampilan sub menu pada material

2. Pada Menu Bar, klik “Daftarkan Material Baru” untuk mengisi data material yang belum pernah didaftarkan sebelumnya.



Gambar 4.21 Tampilan *form* daftarkan material baru

3. Lalu isikan data material sesuai dengan tabel yang telah disediakan (Kode barang, Nama barang, Jenis, Jumlah, Satuan, Warna, Ketebalan,Tanggal Terima, Supplier, dan Rak). Form harus di isi dengan lengkap dan jelas.

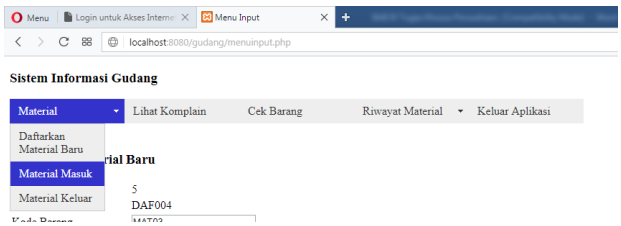


Gambar 4.22 Tampilan tombol *input* material baru

4. Setelah Memasukkan data dengan benar dan lengkap, lalu klik tombol “masukkan” yang berada di bawah form.
5. Lalu setelah itu akan ada pemberitahuan apakah data masuk atau tidak. Jika data tidak masuk, maka ulangi langkah memasukkan data dengan baik dan benar.

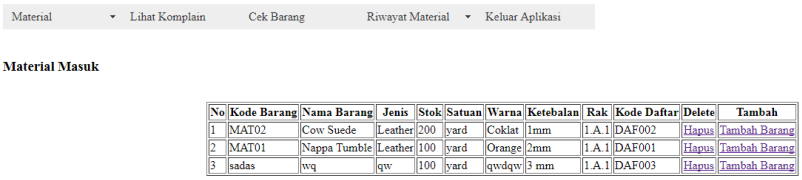
B. Tambah Material

1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.



Gambar 4.23 Tampilan sub menu Material

2. Pada Menu Bar, klik “Material Masuk” untuk menambah material yang telah didaftarkan sebelumnya.



No	Kode Barang	Nama Barang	Jenis	Stok	Satuan	Warna	Ketebalan	Rak	Kode Daftar	Delete	Tambah
1	MAT02	Cow Suede	Leather	200	yard	Coklat	1mm	1.A.1	DAF002	Hapus	Tambah Barang
2	MAT01	Nappa Tumble	Leather	100	yard	Orange	2mm	1.A.1	DAF001	Hapus	Tambah Barang
3	sadas	wq	qw	100	yard	qwdqw	3 mm	1.A.1	DAF003	Hapus	Tambah Barang

Gambar 4.24 Tampilan menu material masuk

3. Lalu klik “tambah barang “ pada material yang ingin di tambah

Gambar 4.25 Tampilan tambah material

4. Lalu isikan data material sesuai dengan tabel yang telah disediakan (Jumlah & Tanggal Terima). Form harus di isi dengan lengkap dan jelas.

Gambar 4.26 Tampilan tombol tambah material

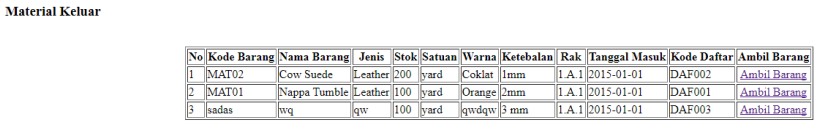
5. Setelah Memasukkan data dengan benar dan lengkap, lalu klik tombol "masukkan" yang berada di bawah form.
6. Lalu setelah itu akan ada pemberitahuan apakah data masuk atau tidak. Jika data tidak masuk, maka ulangi langkah memasukkan data dengan baik dan benar.
7. Jika ingin membatalkan memasukkan data, dapat menggunakan tombol "kembali" yang ada pada pojok kiri atas

C. Material Keluar

1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.

Gambar 4.27 Tampilan menu material

2. Pada Menu Bar, klik "Material Keluar" untuk menambah material yang telah didaftarkan sebelumnya.



Gambar 4.28 Tampilan sub menu material Keluar

3. Lalu klik "ambil barang" pada material yang ingin diambil.

Gambar 4.29 Tampilan menu ambil barang

4. Lalu isikan data material sesuai dengan tabel yang telah disediakan (Jumlah & Tanggal Keluar). Form harus di isi dengan lengkap dan jelas.

Gambar 4.30 Tampilan tombol ambil barang

5. Setelah Memasukkan data dengan benar dan lengkap, lalu klik tombol "masukkan" yang berada di bawah form.
6. Lalu setelah itu akan ada pemberitahuan apakah data masuk atau tidak. Jika data tidak masuk, maka ulangi langkah memasukkan data dengan baik dan benar.
7. Jika ingin membatalkan memasukkan data, dapat menggunakan tombol "kembali" yang ada pada pojok kiri atas

D. Daftarkan Supplier Baru

1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.

Gambar 4.31 Tampilan sub menu data supplier

2. Pada Menu Bar, klik "Data Supplier" Lalu pilih "Daftarkan Supplier Baru"

Gambar 4.32 Tampilan menu daftar supplier baru

3. Lalu isikan data material sesuai dengan tabel yang telah disediakan (Kode Supplier, Nama Suplier, Alamat, Nomor Telepon, Nomor Handphone, dan Alamat Email). Form harus di isi dengan lengkap dan jelas.
4. Setelah Memasukkan data dengan benar dan lengkap, lalu klik tombol "masukkan" yang berada di bawah form.

E. Edit Data Supplier

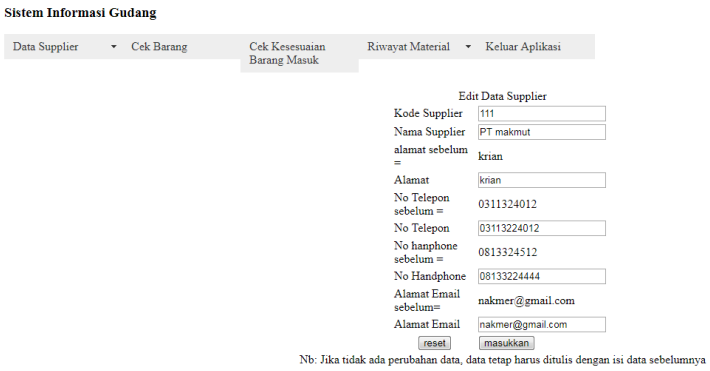
1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.

Gambar 4.33 Tampilan sub menu Data Supplier

2. Pada Menu Bar, klik "Data Supplier" Lalu pilih "Edit Data Supplier"

Gambar 4.34 Tampilan menu edit data supplier

3. Lalu klik "edit data" pada data supplier yang ingin di edit datanya.



Gambar 4.35 Tampilan menu Edit Data

4. Lalu masukkan data pada form dengan lengkap pada tabel yang telah disediakan (Alamat, Nomor Telepon, Nomor Handphone, dan Alamat Email). Form harus di isi dengan lengkap dan jelas. Jika tidak ada perubahan data, data tetap harus di isi pada form sesuai dengan data yang sebelumnya.
5. Setelah Memasukkan data dengan benar dan lengkap, lalu klik tombol "masukkan" yang berada di bawah form.
6. Lalu setelah itu akan ada pemberitahuan apakah data masuk atau tidak. Jika data tidak masuk, maka ulangi langkah memasukkan data dengan baik dan benar.
7. Jika ingin membatalkan memasukkan data, dapat menggunakan tombol "kembali" yang ada pada pojok kiri atas

F. Riwayat Material Masuk

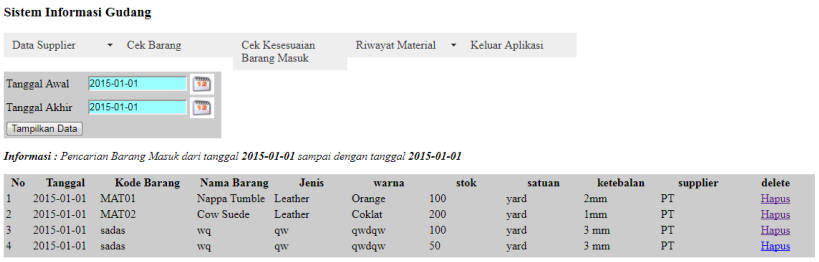
1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.

Gambar 4.36 Tampilan sub menu Riwayat Material

2. Pada Menu Bar, klik "Riwayat Material" lalu pilih "Riwayat material masuk"

Gambar 4.37 Tampilan menu Material masuk

3. Lalu pilih tanggal awal dan tanggal akhir. Dengan memilih tanggal awal dan tanggal akhir, maka data histori akan ditampilkan berdasarkan dasar waktu yang ditentukan.
4. Setelah itu, klik "tampilkan data" , maka data pencarian akan ditampilkan.



Gambar 4.38 Tampilan data dari Material Masuk

5. Pada Bagian paling kanan dari data terdapat fungsi "hapus" yang digunakan untuk menghapus data jika terjadi kesalahan penginputan data. Fungsi ini dapat dijalankan dengan mengklik "hapus" pada data yang ingin dihilangkan.

G. Riwayat Material Keluar

1. Untuk mengakses fungsi ini user di harapkan telah log in terlebih dahulu.

Gambar 4.39 Tampilan menu Riwayat Material

2. Pada Menu Bar, klik "Riwayat Material" lalu pilih "Riwayat material Keluar"

Gambar 4.40 Tampilan menu material keluar

3. Lalu pilih tanggal awal dan tanggal akhir. Dengan memilih tanggal awal dan tanggal akhir, maka data histori akan ditampilkan berdasarkan dasar waktu yang ditentukan.
4. Setelah itu, klik "tampilkan data" , maka data pencarian akan ditampilkan.

Gambar 4.41 Tampilan data material keluar

5. Pada Bagian paling kanan dari data terdapat fungsi "hapus" yang digunakan untuk menghapus data jika terjadi kesalahan penginputan data. Fungsi ini dapat dijalankan dengan mengklik "hapus" pada data yang ingin dihilangkan.

H. Lihat Komplain

1. Untuk Mengakses fungsi ini user diharapkan telah log in terlebih dahulu

Gambar 4.42 Pilih lihat Komplain

2. Klik lihat komplain pada menu bar

Gambar 4.43 Tampilan menu Lihat Komplain

3. Akan ditampilkan komplain yang telah dikirimkan.

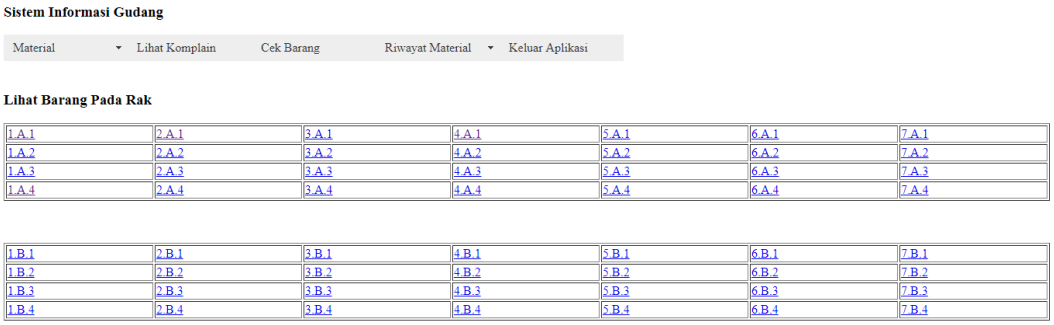
4. Tombol “delete” yang terdapat pada pojok kanan tabel dapat digunakan untuk menghapus komplain jika sudah ada tindakan lanjutan dari komplain yang dihadapi.

I. Cek Barang

1. Untuk Mengakses fungsi ini user diharapkan telah log in terlebih dahulu

Gambar 4.44 Pilih cek Barang

2. Klik cek barang pada menu bar



Gambar 4.45 Tampilan menu Cek barang

3. Klik pada no rak yang ingin di cek.

Gambar 4.46 Tampilan data rak

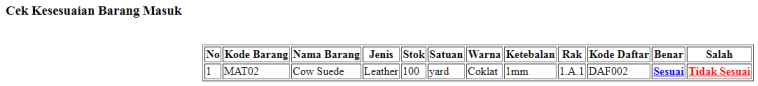
4. Data barang pada rak akan di tampilkan. Klik tombol kembali jika ingin kembali pada halaman sebelumnya.

J. Cek Kesesuaian Barang Masuk

1. Untuk Mengakses fungsi ini user diharapkan telah log in terlebih dahulu.

Gambar 4.47 Pilih cek Ketersediaan Barang

2. Klik Cek Kesesuaian barang masuk pada menu bar
3. Akan ditampilkan informasi barang yang masuk pada gudang.



Gambar 4.48 Tampilan menu cek ketersediaan barang

4. Klik sesuai jika barang masuk sesuai dengan yang dipesan dari supplier

Gambar 4.49 Tampilan barang Tidak Sesuai

5. Klik Tidak sesuai jika barang masuk tidak sesuai dengan barang yang dipesan dari supplier. Silahkan masukkan tanggal komplain dan Keterangan ketidak sesuaian pada kolom input yang disediakan. Lalu klik masukkan.

4.4.5 Penggunaan Coding PHP pada Sistem Informasi

1. echo "<h1>Sistem Informasi Gudang Bahan Baku PT. Cipta Karya Buana</h1>";.

   Penggunaan koding echo digunakan untuk memprint kata-kata pada halaman web. Dengan kode di atas maka akan terprint kata “sistem informasi gudang bahan baku PT.Cipta karya Buana” pada halaman web.
2. Cek Ketersediaan Barang.

   Penggunaan “a href” adalah untuk mehubungkankan suatu halaman dengan halaman yang lain. Maksud dari koding diatas adalah jika kita mengklik “Cek Ketersediaan Barang” maka kita akan dipindahkan ke halaman “cekbarang.php”. Dimana pada “cekbarang.php” akan memiliki koding lainnya lagi.

3. <form action="periksa_login.php" method="post">

   <label>Username:</label>
   <input type="text" id="user" name="user"/>

   <label>Password:</label>
   <input type="password" id="pass" name="pass"/>

   <input type="submit" id="btn" name="btn" value="masuk"/>

   </form>

   Fungsi dari koding Form adalah untuk membuat suatu wadah untuk mengisi data secara sementara kedalam form. Pada form terdapat action=”periksa_login.php” dimana berarti jika form sudah diisi maka akan diproses pada periksa_login.php. Pada form ada input, yang berfungsi untuk memunculkan box untuk mengisi data yang kita inginkan. Id digunakan untuk menamai setiap inputan yang kita punya agar membedakan inputan yang satu dengan yang lain. Kegunaan dari type input sendiri pun ada bermacam-macam mulai dari text, password, submit, reset, dll. Setiap type input memiliki fungsinya masing-masing dan berbeda- beda
4. $kode_barang=$_POST['kode_barang'];
   $nama_barang=$_POST['nama_barang'];
   $jenis=$_POST['jenis'];
   $stok=$_POST['stok'];
   $query_insert2="INSERT INTO tb_barang_rekap (kode_barang,nama_barang,jenis,stok)VALUES('".$kode_barang."','".$nama_barang."','".$jenis."','".$stok.")";

   Lalu ada tanda $ yang menandakan sebagai variabel. Misal $stok berarti sebuah variabel dinamai stok. Lalu untuk $_POST, maksudnya adalah untuk

mengkap nilai yang telah di post melalui fungsi “form”. $jenis=$_POST['jenis']; maka arti dari koding ini adalah untuk menamai variabel dengan nama jenis. Variabel jenis ini di isi nilai “jenis” yang telah di input dari form. Lalu ada juga INSERT INTO , yang digunakan untuk memasukkan suatu nilai ke database. Untuk contoh INTO tb_barang_rekap (kode_barang,nama_barang,jenis,stok)VALUES('".$kode_barang."','".$nama _barang."','".$jenis."','".$stok.")"; maksudnya adalah memasukan data ke tb_barang_rekap yang di isi pada tabel kode_barang,nama_barang,jenis, dan stok yang di isi dengan nilai /values berupa variabel –variabel yang telah dinput dengan form dan nilainya didapatkan dari S_POST.

5. $query_delete="DELETE FROM tb_barang WHERE kode_daftar='".$kode_daftar."'";

   Berbeda dengan fungsi INSERT INTO, guna dari DELETE adalah untuk menghapus yang ada pada database. DELETE FROM tb_barang WHERE kode_daftar='".$kode_daftar , maksud dari kode ini adalah untuk menghapus data dari tb_barang dimana data yang dihapus berdasarkan kode_daftar yang dipilih / diinput oleh user.

6. <?php

   $host="localhost";

   $user="root";

   $pass="";

   $ambildb="gudang1";

   $koneksi=mysql_connect("$host","$user","$pass","$ambildb") or die("gagal mengkoneksikan database");

   $pilihdb=mysql_select_db("$ambildb");

   ?>

   Lalu selanjutnya ada penjelasan mengenai mengkoneksikan program dengan database. Dimana jika ingin mengkoneksikan kita memerlukan kode mysql_connect("$host","username database","password database","database

yang dikoneksikan"). Jika di isi dengan password dan username yang benar maka kita dapat mengkoneksikan program yang telah dibuat dengan database yang dipilih. Untuk contoh kode di atas, database yang digunakan adalah database “gudang1”.

4.5 Analisa

4.5.1 Analisa

Pada sistem yang sebelumnya pencatatan data yang ada di gudang masih dilakukan secara manual, sehingga dapat menyebabkan banyak terjadinya kesalahan. Beberapa masalah yang dapat terjadi pada proses pencatatan pada gudang adalah

1. Data barang dibuat menggunakan Ms. Excel, biasanya pencarian data hanya mengacu pada nama barang sehingga diperlukan ketelitian karena banyak nama barang yang sama tapi mempunyai spesifikasi yang bervariasi.
2. Diperlukan waktu yang lama saat mencari rekapan data
3. Tidak ada sistem pengaman sehingga semua orang dapat masuk
4. Kesalahan dalam menginputkan data yang kurang lengkap sering terjadi karena pada excel tidak ada peringatan jika kita tidak mengisi data pada salah satu tabel.
5. Kesulitan untuk barang yang berada dalam rak

Dengan adanya sistem informasi yang telah dirancangan dalam penelitian ini maka :

1. Pencarian data semakin cepat karena hanya klik menu cek ketersediaan barang maka akan menampilakan barang yang tersedia digudang.
2. Pemasukan dan pengeluran barang lebih mudah karena adanya menu rak yang akan menampilkan barang apa saja yang ada dalam satu rak.
3. Adanya sistem login yang dapat menjaga keamanan informasi data.
4. Pada program untuk setiap pengisian data selalu dilakukan pengecekan secara otomatis agar data yang diisi lengkap. Hal ini meminimalkan pekerja untuk lupa mengisi data.
5. Pencarian data lebih mudah dan dapat di cari sesuai tanggal. Sehingga tidak memerlukan waktu yang lama.

4.5.2 Spesifikasi Kebutuhan Perangkat Lunak

Pada pembuatan sistem informasi ini dibutuhkan perangkat lunak untuk mensupport pengaplikasian program. Spesifikasi perangkat lunak yang digunakan adalah :

1. Windows XP Profesional SP2 atau lebih
2. Dreamweaver CS 3
3. Xampp V3.2.2
4. Web Browser (Mozzila, Chrome, Atau Opera)

4.6 Penutup

4.6.1 Kesimpulan

Dengan adanya aplikasi ini dapat mempermudah kerja dalam mengolah data barang antara *purchasing* dan gudang barang jadi. Penggunaan sistem database dapat mempermudah pihak pengelola untuk mengetahui informasi barang. Proses pengelolaan data dengan komputerisasi meningkatkan kecepatan penyampaian informasi.

4.6.2 Saran

Berdasarkan program yang telah kami rancang terdapat kekurangan yang dapat dikembangkan oleh penelitian selanjutnya yaitu mengembangkan dari sistem dengan cakupan LAN(*Local Area Network)* menjadi sistem yang dapat diakses secara *online*. Program dapat dikembangkan pada divisi lain.

DAFTAR PUSTAKA

Kasmirin, Agus Rahmat. 2016. Perancangan Sistem Informasi Perpustakaan Berbasis Web (Studi Kasus SMAN 1 Penengahan)

Connolly, Thomas and Carolyne, Begg. (2005). *Database System : A Practical Approach to Design, Implementation, and Management, Fourth Edition, Addison Wesley. English.*