

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Tinjauan Pustaka**

Pendukung penelitian ini diambil dari beberapa jurnal penelitian yang berkaitan dengan sistem pengendalian produk. Adapun jurnal yang digunakan sebagai tinjauan pustaka tersebut yaitu sebagai berikut:

1. Jurnal penelitian menurut Pramarikana, 2015, "Pengembangan Sistem Informasi Penjualan dan Pemesanan Air Berbasis Web pada PDAM WAY". Pada intinya penelitian ini membahas tentang prosedur pendistribusian air dengan berbasis web. Sistem ini digunakan untuk tujuan mempersingkat waktu pemesanan air dan meningkatkan kecepatan dalam penyajian laporan.
2. Kartika Sari, 2014, "Analisis dan Perancangan Sistem Informasi Penjualan dan Media Promosi Gordyn Berbasis Web pada Toko Toti Gordyn". Masalah yang dibahas dalam penelitian ini hanya meliputi analisis dan perancangan sistem penjualan dan media promosi gordyn pada Toko Tuty Gordyn. Sistem penjualan yang dimaksud adalah di mulai dari pemesanan, pembayaran secara tunai, serta pengiriman barang kepada pemesan, sedangkan media promosi dengan mempublikasikan produk melalui halaman *web* untuk memperluas pemasaran gordyn Toko Tuty Gordyn.

#### **2.2 Pengertian Dasar**

##### **2.2.1 Rancang Bangun (Desain Sistem)**

Desain sistem dalam pengembangan perangkat lunak merupakan upaya untuk mengonstruksi suatu sistem yang memberikan suatu kepuasan akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit dari segi informasi maupun segi biaya, waktu dan perangkat. Kualitas perangkat

lunak biasanya dinilai dari segi pengguna terhadap perangkat lunak yang digunakan. (Rosa AS & M. Salahuddin; 2015)

### **2.2.2 Sistem**

Sistem adalah sekumpulan elemen yang saling berkaitan atau terpadu untuk mencapai tujuan tertentu. Sebagai gambaran, jika dalam sebuah sistem terdapat elemen yang tidak memberikan manfaat dalam mencapai tujuan yang sama, maka dapat dipastikan bahwa elemen tersebut bukanlah bagian dari sistem. (Abdul Kadir; 2014)

Elemen Sistem adalah suatu sistem yang terdiri dari sejumlah elemen yang saling berinteraksi, yang artinya saling kerjasama membentuk satu kesatuan. Adapun elemen-elemen yang membentuk sebuah sistem yaitu:

#### 1) Tujuan

Secara spesifik tujuan sistem tergantung pada kegiatan yang ditangani, namun kecenderungan sistem ditujukan untuk menuju keunggulan kompetitif yang artinya mampu bersaing mengungguli pesaing.

#### 2) Masukan

Masukan adalah segala sesuatu yang masuk ke dalam sistem yang selanjutnya menjadi bahan untuk diproses.

#### 3) Proses

Proses melakukan bagian yang melakukan perubahan atau transformasi dari masukan menjadi keluaran yang berguna, misalnya berupa informasi dan produk tetapi bisa juga berupa hal yang tidak berguna seperti sisa pembuangan atau limbah.

#### 4) Keluaran

Keluaran merupakan keluaran dari pemrosesan. Pada sistem informasi keluaran bisa berupa informasi, saran, laporan, dan sebagainya.

#### 5) Mekanisme pengendalian dan umpan balik

Mekanisme pengendalian berupa umpan balik yang mencuplik keluaran. Umpan balik ini digunakan untuk mengendalikan baik masukan ataupun proses.

6) Batasan

Batasan sistem adalah pemisah antara sistem dengan daerah di luar sistem. Batas sistem menentukan konfigurasi, ruang lingkup, atau kemampuan sistem.

7) Lingkungan

Lingkungan adalah segala sesuatu yang ada diluar sistem. Lingkungan dapat berpengaruh terhadap operasi sistem dalam arti dapat merugikan atau menguntungkan suatu sistem itu sendiri.(Abdul Kadir; 2014,)

### **2.2.3 Informasi**

Informasi pada dasarnya adalah sehimpunan data yang telah diolah menjadi sesuatu yang memiliki arti dan kegunaan lebih luas. Menurut Gordon B. Davis, informasi didefinisikan sebagai data yang telah diolah menjadi suatu bentuk yang berguna bagi penerimanya dan memiliki nilai bagi pengambilan suatu keputusan saat ini atau dimasa yang akan datang.

Informasi sangat berhubungan erat dengan suatu data. Data adalah hasil pengukuran dan pencatatan terhadap fakta tentang sesuatu, keadaan, tindakan, atau kejadian. Transformasi data menjadi suatu informasi dilakukan di dalam suatu sistem, dimana data yang merupakan masukan kemudian diolah atau diproses oleh sistem mejadi keluaran atau informasi. Sistem tersebut di sebut sistem informasi. (DR. Bambang Hartono; 2013)

Sehubungan dengan pengertian tersebut, berikut ini adakan dijelaskan mengenai kriteria nilai suatu informasi:

1) Relevansi

Informasi disediakan atau disajikan untuk digunakan. Oleh karena itu, informasi yang bernilai tinggi adalah yang relevan dengan kebutuhan yaitu untuk apa informasi itu akan digunakan.

2) Kelengkapan dan Keluasan

Informasi akan bernilai tinggi jika tersaji secara lengkap dan cakupan yang luas dan tidak terpotong-potong.

3) Kebenaran

Kebenaran informasi ditentukan oleh validitas atau dapat dibuktikan kebenarannya. Informasi dari data, dan data adalah fakta.

4) Terukur

Informasi dari data atau pengukuran dan pencatatan terhadap fakta. Jadi informasi yang bernilai tinggi adalah informasi yang jika dilacak kembali kepa data, data tersebut dapat diukur sesuai dengan faktanya.

5) Keakuratan

Kecermatan dalam mengukur dan mencatat fakta akan menentukan keakuratan data dan nilai informasi yang dihasilkan.

6) Kejelasan

Informasi dapat disajikan dalam bentuk teks, table, grafik, chart, dan lain-lain. Namun apapun bentuk yang dipilih yang penting dapat menjadikan pemakai mudah memahami pemakainya.

7) Keluwesan

Informasi yang baik adalah yang mudah diubah-ubah bentuk penyajiannya sesuai dengan kebutuhan dan situasi yang dihadapi.

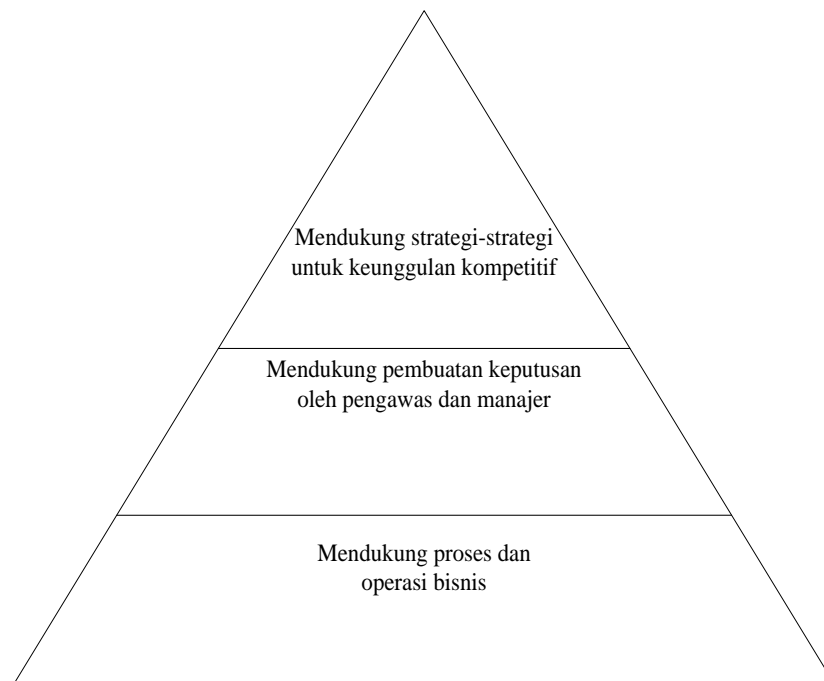
8) Ketepatan Waktu

Informasi yang baik adalah informasi yang disajikan tepat pada saat dibutuhkan.

### 2.2.4 Sistem Informasi

Sistem informasi adalah seperangkat komponen yang saling berhubungan, yang bekerja untuk mengumpulkan dan menyimpan data serta mengolahnya menjadi informasi untuk digunakan. (DR. Bambang Hartono, 2013)

Sistem informasi memiliki peranan yang sangat penting bagi sebuah perusahaan. Adapun peranan sistem informasi tersebut dapat dilihat pada gambar berikut:



Gambar 2.1 Peranan sistem informasi di organisasi

(Abdul Kadir; 2014)

### 2.2.5 Pengendalian

Pengendalian adalah upaya untuk memantau pelaksanaan rencana oleh karena itu pengendalian harus dilaksanakan selagi rencana masih sedang dilaksanakan dan bukan sesudahnya. Terdapat 4 kategori pengendalian yang akan dijelaskan sebagai berikut:

1. Pengendalian produksi

Pengendalian produksi berkaitan dengan perencanaan produksi agar rencana produksi berjalan sesuai dengan yang telah ditetapkan.

2. Pengendalian persediaan

Pengendalian persediaan adalah kegiatan memantau dan menjaga agar persediaan barang, termasuk bahan baku, barang setengah jadi, dan barang jadi, sesuai jumlahnya dengan apa yang tercantum dalam rencana.

3. Pengendalian mutu

Pengendalian mutu berkaitan dengan penjagaan mutu produk agar sesuai dengan standar yang telah ditetapkan, baik berdasarkan spesifikasi dari teknisi perusahaan maupun tuntutan dari konsumen.

4. Pengendalian keuangan

Pengendalian keuangan dilakukan untuk hampir semua kegiatan, karena tidak ada kegiatan yang tidak memerlukan uang.

(DR. Bambang Hartono, 2013)

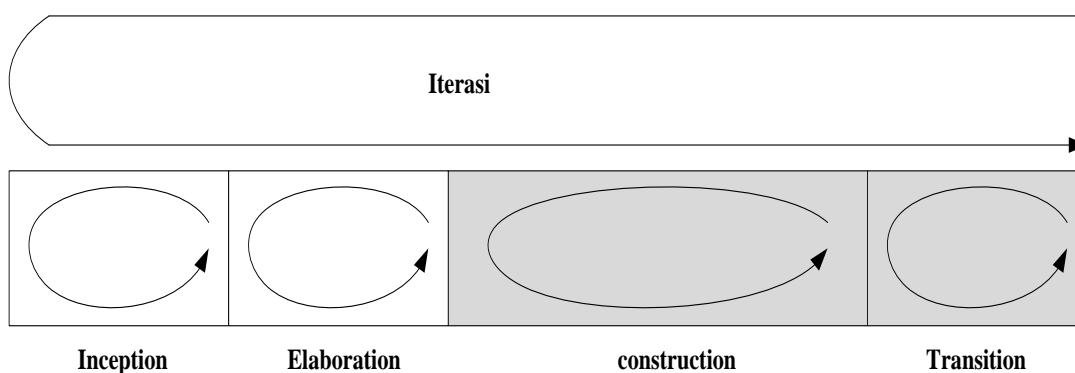
### **2.3 Metodologi Pengembangan Sistem**

Untuk mengembangkan suatu sistem informasi, kebanyakan perusahaan menggunakan metodologi yang sering disebut metodologi pengembangan sistem. Metodologi itu sendiri adalah suatu proses standar yang diikuti organisasi untuk melaksanakan seluruh langkah yang diperlukan untuk menganalisa, merancang, mengimplementasikan, dan melihat sistem informasi. (Abdul Kadir; 2014)

Seperti yang dilakukan pada kebanyakan proses, pengembangan sistem juga memiliki daur hidup yang sering disebut dengan Daur Hidup Pengembangan Sistem Informasi yang dalam hal ini adalah pengembangan sistem informasi Analisis Desain Berorientasi Objek (ADBO). Metodologi Berorientasi Objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Alat

yang digunakan dalam metodologi ini meliputi *use case*, *activity diagram*, *sequence diagram*, dan *class diagram*. (Rosa A.S & M.Shalahudin; 2013).

Di dalam metodologi pengembangan sistem berorientasi objek, terdapat metode tahapan dalam penyelesaian proses pengembangan sistem. Adapun metode yang akan digunakan dalam penelitian ini adalah metode *Rational Unified Process* (RUP). Metode RUP merupakan pendekatan perangkat lunak yang dilakukan berulang-ulang, fokus pada arsitektur, lebih diarahkan pada penggunaan kasus. RUP merupakan proses rekayasa perangkat lunak dengan mendefinisikan struktur yang baik untuk alur hidup proyek perangkat lunak. (Rosa As & M.Salahuddin; 2015) *Rational Unified Process* (RUP) memiliki 4 fase atau tahapan yang dapat dilakukan secara interaktif. Berikut ini gambar alur hidup RUP:



Gambar 2.2 Alur Hidup *Rational Unified Process* (RUP)

Berikut ini penjelasan untuk setiap fase pada RUP:

1. *Inception* (Permulaan)

Tahapan ini lebih pada pemodelan proses bisnis yang dibutuhkan dan mendefinisikan kebutuhan akan sistem yang akan dibuat seperti ruang lingkup dari proyek, dan membangun kasus bisnis yang dibutuhkan. Kriteria dari hasil tahapan ini adalah berikut:

- a. Umpan balik dari pendefinisian ruang lingkup, perkiraan biaya, dan perkiraan jadwal.
- b. Kredibilitas dari perkiraan biaya, perkiraan jadwal, penentuan skala prioritas, resiko dan proses pengembangan.
- c. Ruang lingkup prototype yang akan dikembangkan.
- d. Membangun garis dasar dengan membandingkan perencanaan actual dengan perencanaan yang direncanakan

2. *Elaboration* (Perluasan/Perencanaan)

Tahapan ini lebih fokus pada perencanaan arsitektur sistem. Perencanaan ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Tahapan ini lebih pada analisis dan desain sistem serta implementasi sistem yang fokus pada prototype sistem. Hasil yang diharapkan dari tahapan ini sebagai berikut:

- a. Model kasus yang digunakan (use-case) dimana kasus dan actor yang terlibat telah diidentifikasi dan sebagian besar kasus harus dikembangkan.
- b. Deskripsi dari arsitektur perangkat lunak dari proses pengembangan sistem yang telah dibuat.
- c. Rancangan arsitektur yang dibuat diimplementasikan dan mengimplementasikan *use-case*.
- d. Rencana pengembangan untuk seluruh proyek yang telah dibuat.
- e. Purwarupa yang dapat didemonstrasikan untuk mengurangi setiap resiko teknis yang diidentifikasi.

3. *Construction* (Konstruksi)

Tahapan ini fokus pada pengembangan komponen dan fitur-fitur sistem. Tahapan ini fokus pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program.



#### 4. *Transition* (Transisi)

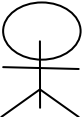

Tahapan ini lebih fokus pada *deployment* atau instalasi perangkat lunak agar dapat dipahami oleh user. Aktifitas ini lebih pada pelatihan pemeliharaan, dan pengujian sistem apakah sistem tersebut sudah sesuai harapan user itu sendiri.

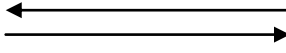
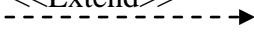
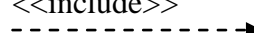
Agar dapat melakukan langkah-langkah sesuai dengan metodologi pengembangan analisis dan desain berorientasi objek, maka dibutuhkan alat dan teknik untuk melaksanakannya. Alat-alat yang digunakan dalam suatu metodologi umumnya berupa suatu gambar diagram atau grafik. Adapun alat pengembangan sistem yang digunakan yaitu sebagai berikut.

##### a. *Use Case Diagram*

Use case atau diagram use case merupakan pemodelan untuk melakukan kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang dibuat. Secara umum, use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang hendak menggunakan fungsi-fungsi itu. (Rosa AS. & M. Salahuddin; 2015) Penjelasan simbol dapat dilihat pada tabel berikut:

Tabel 2.1 Bagan *Use Case Diagram*


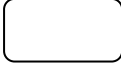
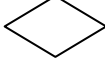


<b>SIMBOL</b>	<b>KETERANGAN</b>
ACTOR 	Mendefinisikan entitas diluar sistem yang memakai sistem.
USE CASE 	Gambaran fungsionalitas dari suatu sistem, sehingga pengguna mengerti kegunaan sistem yang akan dibangun

<p>RELASI</p> 	<p>Menceritakan hubungan antara aktor dan <i>use case</i> sehingga diagram dapat di pahami</p>
<p><i>EKSTENSE / EXTEND</i></p> <p>&lt;&lt;Extend&gt;&gt;</p> 	<p>Relasi Use Case tambahan kesebuah Use Case yang di tambahkan dapat berdiri sendiri walau tanpa Use Case tambahan itu mirip dengan prinsip inheritance pada program berorientasi objek. Biasanya Use Case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan</p>
<p>INCLUDE</p> <p>&lt;&lt;include&gt;&gt;</p> 	<p>Digunakan untuk pengecekan apakah usecase yang ditambahkan telah dijalankan sebelum Use Case tambahan dijalankan.</p>

b. *Activity Diagram*

Diagram aktifitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktifitas dari sebuah sistem atau proses bisnis, atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan di sini bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa AS. & M. Salahuddin; 2015) Sebagai contoh fungsi matematika, pemanggilan perilaku, pemrosesan data. Aktivitas dapat mengakses atribut dan operasi dan operasi *classifier*, tiap objek yang terhubung dan parameter-parameter jika aktivitas memiliki hubungan dengan perilaku. Ketika digunakan untuk model proses bisnis, informasi biasanya disebut *procces-relevant data*. Simbol *activity diagram* dapat dilihat pada tabel berikut:

Tabel 2.2 Bagan *Activity Diagram*

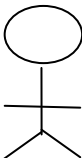
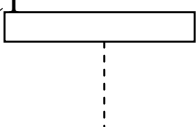

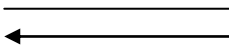
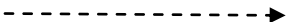
SIMBOL	KETERANGAN
PARTITION 	Memperlihatkan dimana aliran berawal
RELASI 	Merupakan langkah atau aksi-aksi yang terjadi
Decision 	Memperlihatkan dimana keputusan perlu diambil selama terjadi selama terjadi aliran kerja
FORK 	Merupakan percabangan dari sebuah aksi-aksi yang terjadi
FINAL NODE 	Memperlihatkan dimana aliran itu berakhir

c. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek. Oleh karena itu untuk menggambarkan diagram sekuen harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. *Sequence diagram* menjelaskan interaksi objek yang disusun dalam suatu urutan waktu. Diagram ini secara Khusus berasosiasi dengan *use case*. *Sequence Diagram* memperlihatkan tahap demi tahap apa yang sebenarnya terjadi untuk menghasilkan sesuatu didalam *use case*. (Rosa A.S & M.Shalahudin; 2015)

Penjelasan *sequence diagram* dapat dilihat pada tabel 2.3.



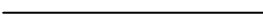

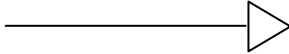
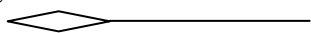
Tabel 2.3 Bagan *Sequence Diagram*

SIMBOL	KETERANGAN
ACTOR 	Prilaku didalam <i>sequence Diagram</i>
OBJECT 	Menambahkan Objek baru pada diagram
OBJECT MESSEGE 	Menggambarkan pesan ( <i>message</i> ) antar dua objek
MESSEGE TO SELF 	Menggambarkan pesan ( <i>message</i> ) yang menuju dirinya sendiri
RETURN MESSEGE 	Menggambarkan pengembalian dari pemanggilan prosedur

d. *Class Diagram*

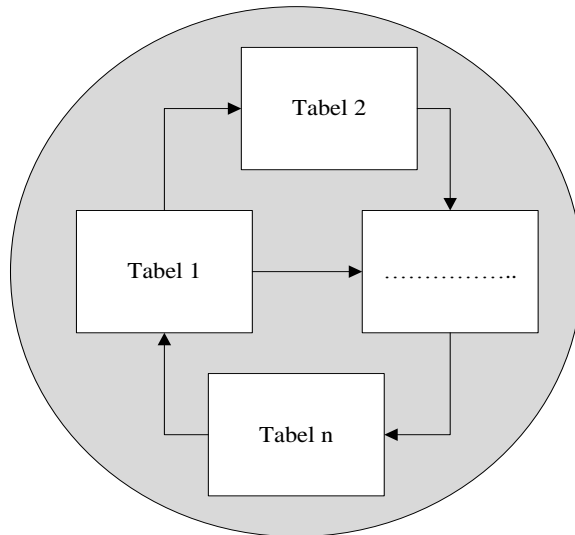
*Class Diagram* atau diagram kelas menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif. (Rosa A.S & M.Shalahudin; 2013) Penjelasan simbol *class diagram* pada tabel berikut:

Tabel 2.4 Bagan *Class Diagram*

SIMBOL	KETERANGAN
Clas 	Menambahkan kelas baru pada diagram
Interface 	Menambahkan kelas antarmuka ( <i>interface</i> ) pada diagram
Association 	Menggambar relasi asosiasi
Association class 	Menghubungkan ( <i>association class</i> ) suatu relasi Asosiasi
Generalization 	Menggambarkan relasi generalisasi
Aggregation 	Menggambarkan relasi agregasi

## 2.4 Basis Data

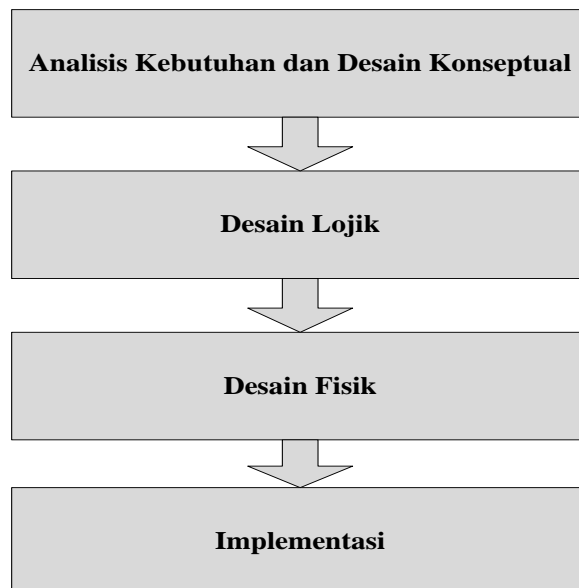
Sistem basis data adalah sistem komputerisasi yang tujuan utamanya adalah untuk memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. Kebutuhan basis data dalam sistem informasi meliputi memasukan, menyimpan, mengambil, dan membuat laporan berdasarkan data yang telah disimpan. Database diimplementasikan dengan bentuk tabel-tabel yang saling memiliki relasi seperti gambar berikut:



Gambar 2.3 Ilustrasi basis data

Sumber : (Rosa AS & M. Salahuddin; 2015)

Tidak hanya perangkat lunak yang memiliki daur hidup, dalam membuat perencanaan basis data juga memiliki alur hidup atau *Database Life Cycle (DBLC)*. (Rosa AS & M. Salahuddin, 2015, 49) Alur hidup basis data dapat dilihat pada gambar berikut:



### Gambar 2.4 Alur Hidup Basis Data

Dari beberapa pendapat tersebut dapat disimpulkan bahwa basis data adalah sekumpulan informasi yang berhubungan dengan subjek atau tujuan tertentu seperti melacak pesanan konsumen, *maintaining inventory* atau pada umumnya mengumpulkan semua data yang berhubungan pada satu tempat. Data tersebut dapat disimpan, dimanipulasi dan digunakan dalam banyak cara. Istilah-istilah yang dipergunakan di dalam basisdata yakni sebagai berikut.

a. Entitas

Suatu objek yang dapat dibedakan dengan objek lainnya yang dapat diwujudkan di dalam basis data.

b. *Attribute/Field*

Karakteristik entitas tertentu.

c. *Data Value*

Merupakan data *actual* atau informasi yang di simpan di tiap data elemen atau *attribute*. Isi *attribute* disebut nilai data.

d. *Record/Tuple*

Kumpulan isi elemen data yang saling berhubungan menginformasikan tentang suatu entitas secara lengkap.

e. *File*

Kumpulan *record* sejenis yang mempunyai panjang elemen dan *attribute* yang sama, namun berbeda-beda data *value*-nya.

f. Kunci elemen data

Sebagai tanda pengenal yang secara unik mengidentifikasi entitas dari suatu kumpulan entitas.

g. *Database management system*

*File* yang saling berkaitan bersama dengan program untuk pengelolaannya.

Didalam *database* juga terdapat juga beberapa *attribute-attribute key (relatinal key)* yang diantaranya adalah sebagai berikut .

1. *Candidate key* adalah *attribute* yang berbeda didalam relasi yang biasanya mempunyai nilai-nilai unik.
2. *Primary key* adalah *candidate key* yang dipilih untuk mengidentifikasi entitas secara unik.
3. *Foreign key* adalah *attribute key* dengan *domain* yang sama yang menjadikan ciri utama dari sebuah relasi tetapi pada relasi lain *attribute* tersebut hanya *attribute* biasa.
4. *Alternative key* adalah *candidate key* yang tidak dipilih sebagai *primary key*.

Pada model data *relational*, hubungan antar file direlasikan dengan kunci relasi (*relation key*) yang meupakan kunci utama dari masing-masing *file*. Relasi antar dua tabel dapat dikatagorikan menjadi 3 macam. Demikian pula untuk membantu menggambarkan relasi secara lengkap terdapat juga beberapa relasi dalam hubungan antribut yang ada didalam satu *file* atau dua *file*, yaitu sebagai berikut:

- a. *One to one relationship* dua file  
Hubungan antara file pertama dengan file kedua adalah satu berbanding satu. Seperti pada pelajaran privat dimana satu guru mengajar satu siswa hanya diajar oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antara keduanya diwakilkan dengan tandah panah tunggal.
- b. *One to many relationship* dua file  
Hubungan antara file pertama dengan file kedua adalah satu berbanding banyak atau dapat pula dibalik, banyak lawan satu. Seperti pada sistem pengajaran di sekolah dasar, dimana satu guru mengajar banyak siswa dan siswa hanya diajar oleh satu guru. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antar keduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.
- c. *Many to many relationship* dua file



Hubungan antara file pertama dengan file kedua adalah banyak berbanding banyak. Seperti pada sistem pengajaran diperguruan tinggi, dimana satu dosen mengajar banyak mahasiswa dan mahasiswa diajar oleh banyak dosen. Hubungan tersebut dapat digambarkan dengan tanda lingkaran untuk menunjukkan tabel dan relasi antarkeduanya diwakilkan dengan tanda panah ganda untuk menunjukkan hubungan banyak tersebut.

d. Relasi *one to one dua atribut* dalam satu file

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan satu. Misalnya atribut nomor pegawai yang unik dan atribut nomor KTP pegawai tersebut mempunyai hubungan satu lawan satu. Satu nomor pegawai hanya satu nomor KTP, tidak ada yang berganda.

e. Relasi *many to one dua atribut* dalam satu file

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan satu lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut, maka hubungan antara atribut alamat pegawai dengan nomor pegawai adalah satu alamat menunjukkan banyak pegawai.

f. Relasi *many to many dua atribut* dalam satu file

Hubungan antara satu atribut dengan atribut yang lain dalam satu file yang sama mempunyai hubungan banyak lawan banyak. Misalnya pada satu alamat pegawai terdapat puluhan pegawai kantor tersebut dan beberapa pegawai punya nama yang sama yang mempunyai alamat berbeda. Maka hubungan antara atribut alamat pegawai dengan nama pegawai adalah beberapa alamat menunjukkan nama pegawai yang sama dan sebaliknya.

Setelah itu, dalam perancangan *database* terdapat sistem kode yang digunakan untuk mengklasifikasikan data, memasukan data kedalam komputer dan untuk mengambil

bermacam-macam informasi yang berhubungan dengannya. Kode dapat dibentuk dari kumpulan angka, huruf dan karakter-karakter khusus. Beberapa tipe kode yang digunakan diantaranya sebagai berikut.

a. Kode mnemonik (*mnemonic code*)

Kode mnemonik digunakan untuk tujuan supaya mudah diingat dengan dasar singkatan.

b. Kode urut (*sequential code*)

Kode urut disebut juga dengan kode seri, merupakan kode yang nilainya urut antara satu kode dengan kode berikutnya.

c. Kode blok (*block code*)

Kode blok mengklasifikasikan *item* kedalam kelompok blok tertentu yang mencerminkan suatu klasifikasi tertentu atas dasar pemakaian maksimum yang diharapkan.

d. Kode group (*group code*)

Kode group merupakan kode yang berdasarkan *field-field* dan tiap-tiap *field* kode mempunyai arti.

e. Kode desimal (*decimal code*)

Kode desimal mengklasifikasikan kode atas dasar 10 unit angka desimal dimulai dari angka 0 sampai dengan angka 9 atau dari 00 sampai dengan 99 tergantung dari banyaknya kelompok.

Kamus data adalah suatu daftar elemen yang pada sistem perangkat lunak sehingga masukan dan keluaran dapat dipahami (memiliki standar cara penulisan). (Rosa AS & M. Salahuddin; 2015) Pembentukan kamus data didasarkan pada alur data yang terdapat rancangan input dan output. Berikut contoh tampilan format kamus data pada gambar 2.5.

Nama Database :  
 Nama Tabel :  
 Primary Key :  
 Foreign Key :

Nama Field	Type	Size	Kondisi	Keterangan

Gambar 2.5 Format Kamus Data

## 2.5 Perangkat Lunak Pendukung

Perangkat lunak pendukung yaitu *software* yang akan digunakan untuk menganalisis dan merancang sistem yang diusulkan, diantaranya seperti berikut.

### 2.5.1 Pengertian *MySQL*

*MySQL* bukan termasuk bahasa pemrograman. *MySQL* merupakan salah satu database populer yang mendunia. Pada umumnya perintah yang sering digunakan dalam *MySQL* adalah *Select* (mengambil), *Insert* (menambah), *Update* (mengubah), dan *Delete* (menghapus). *MySQL* bekerja dengan menggunakan bahasa basis data atau bahasa yang sering disebut dengan *Database Managemen System (DBMS)*. (Agus Saputra; 2013)

*MySQL* memiliki beberapa keistimewaan, antara lain :

1. Portabilitas.

*MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Open Source.

*MySQL* didistribusikan secara *open source*, dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. Multiuser.

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning’.

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Perintah dan Fungsi.

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).

6. Keamanan

MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta sandi terenkripsi.

7. Antar Muka.

MySQL memiliki antar muka (*interface*) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).

### **2.5.2 Internet**

*Internet* merupakan jaringan terbesar yang menghubungkan jutaan komputer yang tersebar diseluruh penjuru dunia dan tidak terikat pada suatu organisasi apapun. Dengan menggunakan jaringan ini, sebuah perusahaan dapat melakukan pertukaran informasi secara internal ataupun melakukan pertukaran informasi secara eksternal dengan oerorganisasi-organisasi yang lain. Jaringan ini tersusun atas berbagai jenis komputer dan sistem operasi.

Pada awalnya internet dibangun oelh Depatemen Pertahanan Amerika Serikat dalam rangka untuk hubungan dengan para ilmuwan dan professor Universitas seluruh

dunia. Kini internet dapat digunakan oleh siapa saja, untuk mengakses informasi apa saja, dan bahkan untuk melakukan transaksi bisnis.

Supaya dapat terhubung dengan internet, seorang pemakai dapat mengakses komputer pada perusahaan yang telah terkoneksi ke internet atau perlu menjadi pelanggan dari sebuah ISP (*Internet Service Provider*). ISP adalah organisasi komersial yang bergerak dibidang penyedia jasa akses ke internet. Beberapa contoh ISP di Indonesia yaitu seperti INDO.NET, INDOSATNET, LinkNet, Wasantara Net. Abdul Kadir, (2014)

### 2.5.3 WWW (*Word Wide Web*)

Sistem pengakses informasi dalam internet yang paling terkenal adalah WWW (*Word Wide Web*) atau bisa dikenal dengan istilah *web*. WWW pertama kali diciptakan pada tahun 1991 di *CERN*, Laboratorium Fisika Partikel Eropa, Jenewa, Swiss. Tujuan awalnya adalah menciptakan media yang mudah untuk berbagi informasi diantara fisikawan dan ilmuwan.

Web menggunakan protocol yang disebut dengan HTTP (*Hypertext Transfer Protocol*) yang berjalan pada TCP/IP. Dokumen ini diletakkan dalam sebuah *web server* (server yang melayani permintaan halaman *web*) dan diakses oleh klien (pengakses informasi) melalui perangkat lunak yang disebut dengan web browser atau sering disebut *browser* saja.

Informasi yang terdapat pada *Web* disebut halaman *web* (*web page*). Untuk mengakses sebuah halaman *web* dari browser, pemakai perlu menyebutkan URL (*Uniform Resource Locator*) sebagai contoh URL “ <http://persediaan.com/index.html>: (Abdul Kadir, 2014)

#### **2.5.4 PHP**

PHP memiliki kepanjangan *Hypertext Preprocessor*, merupakan suatu bahasa pemrograman yang difungsikan untuk membangun salah satu website dinamis. *PHP* menyatu dengan kode *HTML*, dengan maksud adalah beda kondisi. *HTML* digunakan sebagai pondasi atau pembangun dari kerangka layout web, sedangkan *PHP* difungsikan sebagai proses nya, sehingga dengan adanya *PHP* tersebut sebuah web dapat dengan mudah untuk di-*maintenance*.

*PHP* adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi *web*. *PHP* merupakan salah satu dari bahasa skrip yang dapat dibangun untuk membangun aplikasi *web*. Bahasa lain yang dapat digunakan untuk keperluan yang sama yaitu *ASP* (*Active Server Pages*) *JSP* (*Java Server Page*), *Java Servlet*, dll.

Program *PHP* memiliki kerangka yang sangat sederhana. Kita dapat menggunakannya atau menyisipkan secara langsung di dalam kode *HTML*. Kode *PHP*, ditandai dengan tanda disebut tag `<?php` dan `?>`. Kode `<?php` digunakan untuk mengawali kode *PHP*, sedangkan `?>` untuk mengakhiri kode *PHP*. (Budi Raharjo,2011)

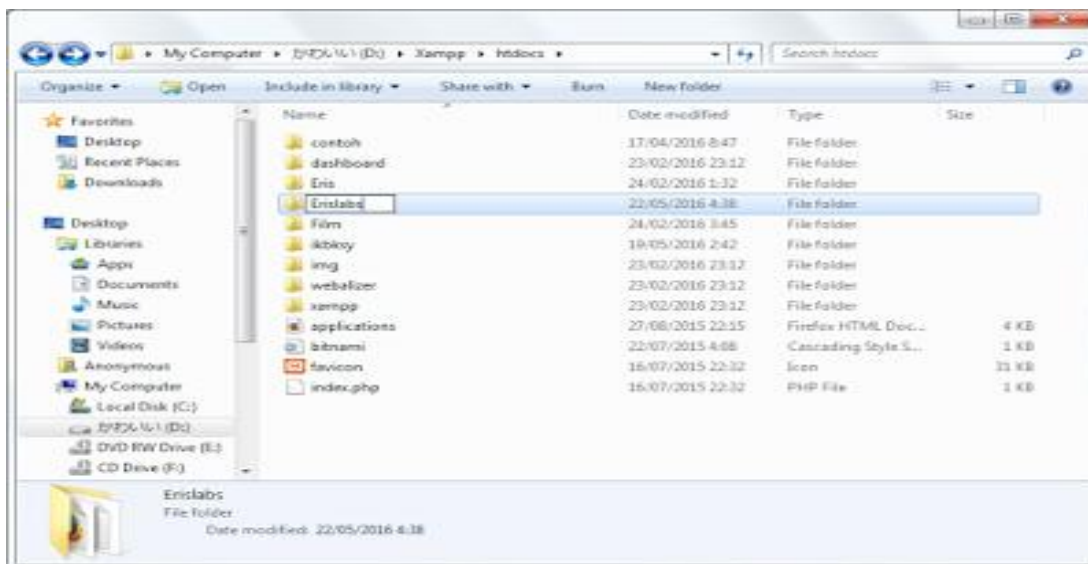
#### **2.5.5 Xampp**

*Xampp* merupakan singkatan dari **X** (empat system operasi apapun), *Apache*, *MySQL*, *PHP*, *Perl*. *Xampp* merupakan tool yang menyediakan paket perangkat lunak ke dalam satu buah paket. Dalam paketnya sudah terdapat *Apache* (*web server*), *MySQL* (*database*), *PHP* (*server side scripting*), *Perl*, *FTP server*, *phpMyAdmin* dan berbagai pustaka bantu lainnya. Dengan menginstall *Xampp* maka tidak perlu lagi melakukan instalasi dan konfigurasi *web server Apache*, *PHP* dan *MySQL* secara manual. *Xampp* akan menginstalasi dan mengkonfigurasikannya secara otomatis untuk Anda.

Versi Xampp yang ada saat ini adalah Versi 1.7.3 yang terdiri atas :

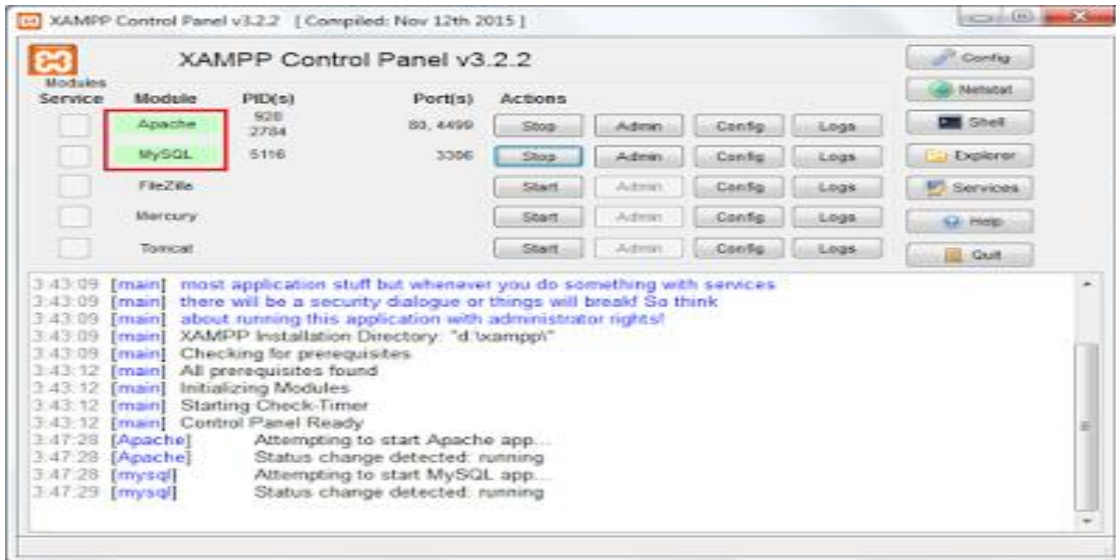
- *Apache 2.2.14 (IPv6 enabled) + OpenSSL 0.9.8l*
- *MySQL 5.1.41 + PBXT engine*
- *PHP 5.3.1*
- *phpMyAdmin 3.2.4*
- *Perl 5.10.1*
- *FileZilla FTP Server 0.9.33*
- *Mercury Mail Transport System 4.72*

Xampp tersedia untuk *Linux, Windows, Mac OS X* maupun *Solaris* sehingga sangat memudahkan membuat *web server multiplatform*. Selain itu Xampp adalah 100% *open source*, tersedia bebas dan legal. Untuk lebih jelasnya tampilan dari Xampp dapat dilihat pada gambar berikut:



Gambar 2.6 Lokasi Penyimpanan *Htdocs*

Control panel utama xampp adalah bagian terpenting karena tanpanya xampp tidak akan berjalan.



Gambar 2.7 Kontrol panel utama xampp

### 2.5.6 Adobe Dreamweaver

*Adobe Dreamweaver* merupakan program *web editor* yang dapat digunakan untuk membangun halaman *web*. *Dreamweaver* merupakan perangkat lunak yang ditujukan untuk membuat suatu situs web. Versi pertama dirilis pada tahun 1997, dan sejak itu *Dreamweaver* menjadi *web editor* yang banyak digunakan oleh para *web developer*. Hal itu antara lain karena kemudahan dalam penggunaannya, kelengkapan fiturnya dan juga dukungannya terhadap teknologi terkini. *Dreamweaver* merupakan salah satu perangkat lunak yang dikembangkan oleh *Macromedia Inc* dan sekarang resmi Milik *Adobe*. (MadCom.2013)

Setelah *Dreamweaver* terinstall di komputer, kita dapat mengaksesnya di menu Program Files. Untuk membukanya kita dapat membuka dengan memilih menu Start > All Programs > Macromedia > Macromedia Dreamweaver 8. Maka akan terbuka *window start-up Dreamweaver*. Dari *window start-up Dreamweaver*, terlihat 3 (tiga) kolom, yaitu *Open a Recent Item*, *Create New* dan *Create From Samples*. Pada



kolom *Open a Recent Item* ditampilkan file-file yang terakhir dibuka (*recent files*). Kita dapat membuka file tersebut dengan mengklik salah satu nama file. Dan pada kolom ini juga terdapat menu *Open* yang dapat dipilih jika ingin membuka file atau dokumen yang belum ada di *recent item*.

Selanjutnya pada kolom *Create New* terdapat pilihan untuk membuat file baru. Pilihan file baru yang dapat dibuat dengan *Dreamweaver* antara lain file HTML, *ColdFusion*, PHP, ASP *VBScript*, *Javascript*, *XML*, *CSS* dan *Dreamweaver Site*. Pada kolom ketiga terdapat kategori *Create From Samples* dimana dapat kita pilih jika ingin membuat halaman *web* berdasarkan tampilan (*template*) yang sudah tersedia. Jadi pilihan ini dapat digunakan jika menginginkan membuat halaman web dengan cepat dan mudah. Jika jenis file yang akan dibuat tidak terdapat di dalam list, kita dapat memilih menu *More* untuk melihat daftar jenis file yang dapat dibuka dengan *Dreamweaver*.

Pada kolom *Category* terdapat beberapa kategori file baru yang dapat dibuat, mulai dari *Basic page* hingga *Page Designs*. Berikut ini penjelasan singkat mengenai kategori tersebut:

a) *Basic page*

*Basic page* digunakan untuk membuat file-file dasar situs *web*. File yang termasuk dalam kategori ini antara lain file HTML, *CSS*, *Javascript* dan *XML*.

b) *Dynamic page*

*Dynamic page* digunakan untuk membuat file-file dinamis. Yang dimaksud dengan file dinamis adalah file atau *script* berbasis *server* (*server-sidescripting*). Masuk dalam kategori ini yaitu PHP, ASP, *ColdFusion* dan *JSP*.

c) *Tempate page*

Kategori ini menampilkan pilihan untuk membuat file baru yang berupa *template*. *Template* tersebut dapat berupa *template* untuk HTML, PHP atau yang lainnya.

d) *Other*

Kategori *Other* menampilkan jenis file yang tidak ada di tiga kategori sebelumnya, antara lain *file Text, Java*, dan seterusnya.

e) *CSS Style Sheets*

Pilihan kategori ini berisi beberapa contoh halaman CSS siap pakai.

f) *Framesets*

Jika kita ingin membuat situs dengan menggunakan frame, maka pilihlah kategori ini dan kita dapat memilih tampilan *frame* yang sesuai.

g) *Page Designs (CSS)*

Kategori ini menampilkan beberapa contoh *layout* halaman situs yang didesain dengan CSS.

h) *Starter Page*

Pada kategori *Starter Page* ditampilkan pilihan beberapa *layout* halaman depan situs. Kita dapat memilih sesuai keinginan kita.

i) *Page Designs*

*Page Designs* menampilkan berbagai format atau *template* dasar halaman *web*