

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Akuntansi**

Menurut Rudianto (2012 : 4) Akuntansi adalah sistem informasi yang menghasilkan informasi keuangan kepada pihak-pihak yang berkepentingan mengenai aktivitas ekonomi dan kondisi suatu perusahaan.

Hasil dari proses akuntansi disebut dengan laporan keuangan. Informasi yang dihasilkan dari proses akuntansi tersebut harus dapat menjawab kebutuhan umum para pemakainya. Karena itu, laporan keuangan suatu badan usaha harus memiliki kualitas yang diperlukan oleh berbagai pihak yang membutuhkan informasi keuangan tersebut.

#### **II.2. Sistem Informasi Akuntansi**

Sistem informasi akuntansi merupakan sebuah sistem informasi yang mengubah data transaksi bisnis menjadi informasi keuangan yang berguna bagi pemakainya.

Tujuan dari sistem informasi akuntansi adalah :

1. Mendukung operasi sehari-hari.
2. Mendukung pengambilan keputusan manajemen.
3. Memenuhi kewajiban yang berhubungan dengan pertanggungjawaban.

Komponen-komponen yang terdapat dalam sistem informasi akuntansi adalah sebagai berikut :

1. Orang-orang mengoperasikan sistem tersebut.

2. Prosedur-prosedur, baik manual maupun yang terotomatisasi, yang dilibatkan dalam pengumpulan, pemrosesan dan penyimpanan data aktivitas-aktivitas organisasi.
3. Data tentang proses-proses bisnis.
4. *Software* yang dipakai untuk memproses data organisasi.
5. Infrastruktur teknologi informasi.

Di dalam organisasi, sistem informasi akuntansi berfungsi untuk :

1. Mengumpulkan dan menyimpan aktivitas yang dilakukan di suatu organisasi, sumber daya yang dipengaruhi oleh aktivitas-aktivitas tersebut dan para pelaku aktivitas tersebut.
2. Mengubah data menjadi informasi yang berguna bagi manajemen.
3. Menyediakan pengendalian yang memadai (Kusrini ; 2007 : 10).

### **II.3. Harga Pokok Penjualan**

Menurut Zinia Th. A. Sumilat (2013 : 456) Harga pokok penjualan (HPP) adalah salah satu komponen dari laporan laba rugi, yang menjadi perhatian manajemen perusahaan dalam mengendalikan operasional perusahaan. Umumnya terdapat 3 macam harga pokok, yaitu harga pokok persediaan, harga pokok produksi dan harga pokok penjualan. Ketiganya sangat penting, namun dalam pengambilan keputusan perlu jelas harga pokok mana yang dimaksud. Permasalahan tersebut timbul karena perbedaan kebutuhan masing-masing tingkat manajemen. Manajer bagian pembelian (*Purchase Manager*) lebih fokus pada harga pokok persediaan, manajer produksi (*Production Manager*) lebih fokus

pada harga pokok produksi, sedangkan manajemen tingkat puncak tentunya akan lebih cenderung fokus pada harga pokok penjualan.

#### **II.4. Metode Full Costing**

Menurut Eka Nur Khasanah, Prof. DR. Rusdiah Iskandar dan Agus Iwan Kesuma (2012 : 4-5) *Full Costing*, merupakan metode penentuan harga pokok produksi yang memperhitungkan semua unsur biaya produksi kedalam harga pokok produksi. Harga pokok produksi yang dihitung melalui pendekatan *Full Costing* terdiri dari unsur harga pokok produksi (biaya bahan baku, biaya tenaga kerja langsung, biaya overhead pabrik variabel dan biaya overhead pabrik tetap) ditambah dengan biaya non-produksi (biaya pemasaran, biaya administrasi & umum).

Pengertian *Full Costing* adalah metode penentuan harga pokok produksi, yang membebankan seluruh biaya produksi, yang terdiri dari biaya bahan baku, biaya tenaga kerja langsung, biaya overhead pabrik, baik yang berperilaku tetap maupun variabel kepada produk.

Dalam metode *Full Costing*, biaya overhead pabrik, baik yang berperilaku tetap ataupun variabel, dibebankan kepada produk yang diproduksi atas dasar tarif yang ditentukan dimuka, pada kapasitas normal atau atas dasar biaya overhead pabrik sesungguhnya. Oleh karena itu, biaya overhead pabrik tetap akan melekat pada harga pokok persediaan produk dalam proses dan persediaan produk jadi yang belum laku dijual, dan baru dianggap sebagai biaya (unsur harga pokok penjualan) apabila produk jadi tersebut telah dijual.

Karena biaya overhead pabrik dibebankan kepada produk atas dasar tarif yang ditentukan dimuka pada kapasitas normal, maka jika dalam suatu periode biaya overhead pabrik sesungguhnya berbeda dengan yang dibebankan tersebut, akan terjadi pembebanan overhead lebih (*Overapplied Factory Overhead*) atau pembebanan biaya overhead kurang (*Underapplied Factory Overhead*). Jika semua produk yang diolah dalam periode tersebut belum laku dijual, maka pembebanan biaya overhead pabrik lebih atau kurang tersebut digunakan untuk mengurangi atau menambah harga pokok produk yang masih dalam persediaan tersebut (baik yang berupa persediaan dalam proses ataupun barang jadi). Namun jika dalam suatu periode akuntansi tidak terjadi pembebanan overhead lebih atau kurang, maka biaya overhead pabrik tetap tidak mempunyai pengaruh terhadap perhitungan laba/rugi sebelum produknya laku dijual.

Metode *Full Costing*, menunda pembebanan biaya overhead pabrik tetap sebagai biaya sampai saat produk yang bersangkutan dijual. Jadi biaya overhead pabrik yang terjadi, baik yang berperilaku tetap ataupun variabel, masih dianggap sebagai aktiva (karena melekat pada persediaan) sebelum persediaan tersebut terjual menurut Eka Nur Khasanah, Prof. DR. Rusdiah Iskandar dan Agus Iwan Kesuma (2012 : 5).

## **II.5. Java**

Menurut Wahana Komputer (2010 : 1) Java adalah bahasa pemrograman yang dapat dijalankan diberbagai jenis komputer dan berbagai sistem operasi termasuk telepon genggam. Java dikembangkan oleh *Sun Microsystems* dan diliris tahun 1995. Java merupakan suatu teknologi perangkat lunak yang digolongkan

multi platform. Selain itu, Java juga merupakan suatu platform yang memiliki virtual machine dan library yang diperlukan untuk menulis dan menjalankan suatu program.

Bahasa pemrograman Java pertama lahir dari *The GreenProject*, yang berjalan selama 18 bulan, dari awal tahun 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi yang dinamakan Oak. Proyek ini dimotori oleh *Patrick Naughton, Mike Sheridan, James Gosling dan Bill Joy*, serta sembilan programmer lainnya dari *Sun Microsystems*. Salah satu hasil proyek ini adalah maskot Duke yang dibuat oleh *Joe Palrang*.

## **II.6. NetBeans**

Menurut Wahana Komputer ( 2010 : 15 ) NetBeans merupakan salah satu proyek *open source* yang disponsori oleh *Sun Microsystem*. Proyek ini berdiri pada tahun 2000 dan telah menghasilkan 2 produk, yaitu NetBeanss IDE dan NetBeans Platform. NetBeans IDE merupakan produk yang digunakan untuk melakukan pemrograman baik menulis kode, meng-*compile*, mencari kesalahan dan mendistribusikan program. Sedangkan NetBeans Platform adalah sebuah modul yang merupakan kerangka awal / pondasi dalam bangun aplikasi desktop yang besar.

NetBeans juga menyediakan paket yang lengkap dalam pemrograman dari pemrograman standar (aplikasi desktop), pemrograman *enterprise*, dan pemrograman perangkat mobile. Saat ini NetBeans telah mencapai versi 6.8.

## II.7. UML (*Unified Modelling Language*)

Menurut Haviluddin ( 2011 : 15 ) *UML (Unified Modelling Language)* adalah suatu alat untuk memvisualisasikan dan mendokumentasikan hasil analisa dan desain yang berisi sintak dalam memodelkan sistem secara visual. UML juga merupakan satu kumpulan konvensi pemodelan yang digunakan untuk menentukan atau menggambarkan sebuah sistem *software* yang terkait dengan objek.

Sejarah UML sendiri terbagi dalam dua fase, yaitu fase sebelum, UML sebenarnya sudah mulai diperkenalkan sejak tahun 1990 namun notasi yang dikembangkan oleh para ahli analisis dan desain berbeda-beda, sehingga dapat dikatakan belum memiliki standarisasi.

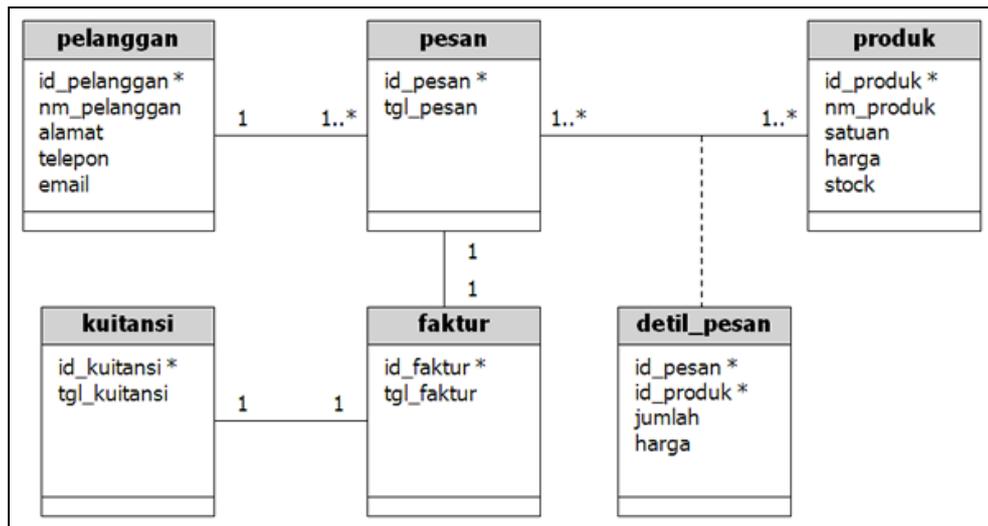
Fase kedua, dilandasi dengan pemikiran untuk mempersatukan metode tersebut dan dimotori oleh *Object Manajement Group* (OMG) maka pengembangan UML dimulai pada akhir tahun 1994 ketika *Grady Booch* dengan metode *Object-Oriented Design* (OOD), *Jim Rumbaugh* dengan metode *Object Modelling Technique* (OMT) mereka ini bekerja pada *Rasional Software Corporetion* dan *Ivar Jacobson* dengan metode *Object-Oriented Software Engineering* (OOSE) yang bekerja pada perusahaan *Objectory Rasional*.

### II.7.1. Diagram – Diagram Pada Metode UML

#### 1. Class Diagram

*Class diagram* dibuat berdasarkan *use case diagram* dan *activity diagram*. *Class diagram* dapat mendeskripsikan kelas-kelas yang digunakan dalam sistem informasi rawat jalan yang akan dibuat. Kelas dengan jenis *actor* diperoleh dari

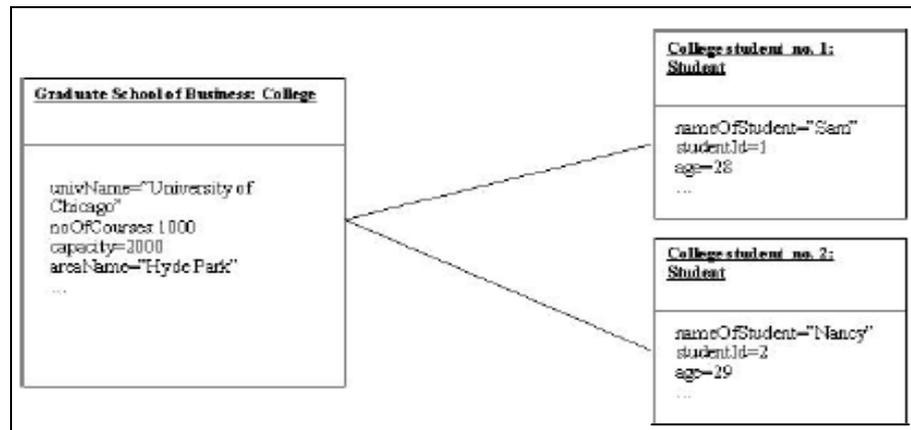
aktor-aktor yang digambarkan dalam *use case diagram*, sedangkan kelas dengan jenis lain seperti *entity*, *boundary*, dan *control* diperoleh dari gambaran alur sistem pada *activity diagram*. Berikut adalah contoh setiap jenis kelas :



**Gambar II.1. Contoh Class Diagram**  
**Sumber : (Haviluddin ; 2011 : 3)**

## 2. Object Diagram

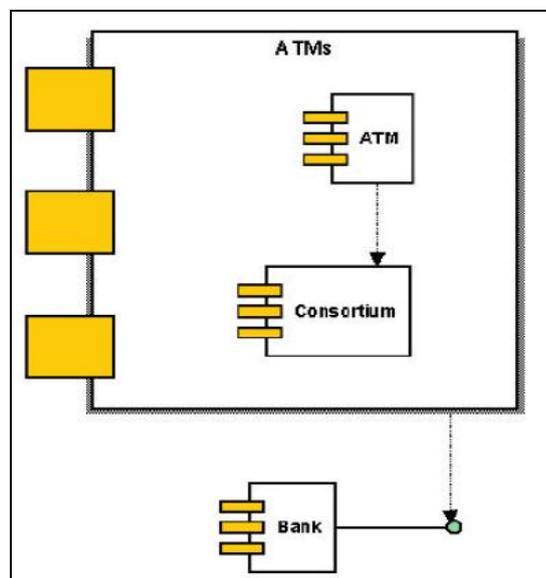
*Object diagram* menggambarkan kejelasan kelas dan warisan dan kadang-kadang diambil ketika merencanakan kelas, atau untuk membantu pemangku kepentingan non-program yang mungkin menemukan diagram kelas terlalu abstrak.



**Gambar II.2. Contoh Objek Diagram**  
 Sumber : (Haviluddin ; 2011 : 3)

### 3. *Component Diagram*

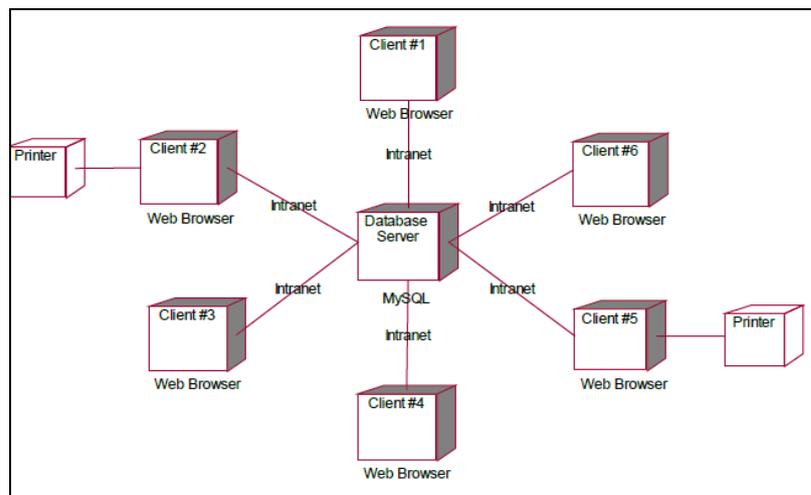
Component Diagram menggambarkan struktur fisik dari kode, pemetaan pandangan logis dari kelas proyek untuk kode aktual di mana logika ini dilaksanakan.



**Gambar II.3. Contoh Component Diagram**  
 Sumber : (Haviluddin ; 2011 : 3)

#### 4. *Deployment Diagram*

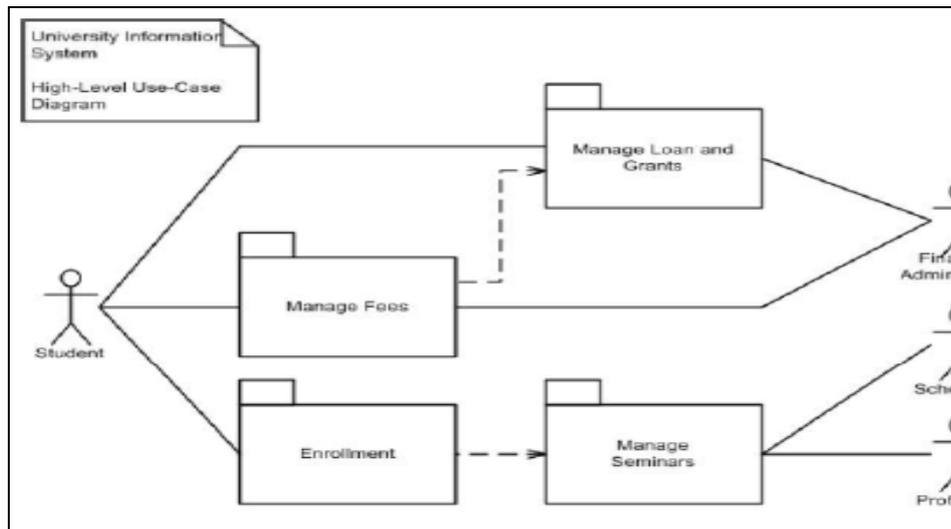
*Deployment diagram* dibuat untuk menunjukkan tata letak sebuah sistem secara fisik, menampakan bagian-bagian *software* yang berjalan pada bagian-bagian *hardware*, dan keterhubungan antara komponen-komponen *hardware* tersebut.



**Gambar II.4. Contoh *Deployment Diagram***  
**Sumber : (Haviluddin ; 2011 : 3)**

#### 5. Package Diagram

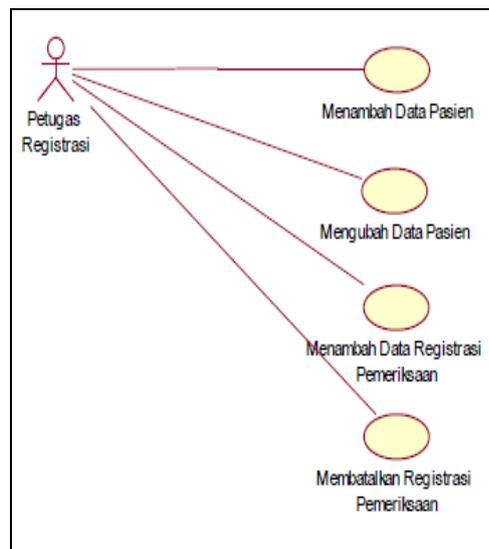
Paket diagram biasanya digunakan untuk menggambarkan tingkat organisasi yang tinggi dari suatu proyek *software*. Atau dengan kata lain untuk menghasilkan diagram ketergantungan paket untuk setiap paket dalam Pohon Model.



**Gambar II.5. Contoh Package Diagram**  
**Sumber : (Haviluddin; 2011 : 4)**

## 6. Use Case Diagram

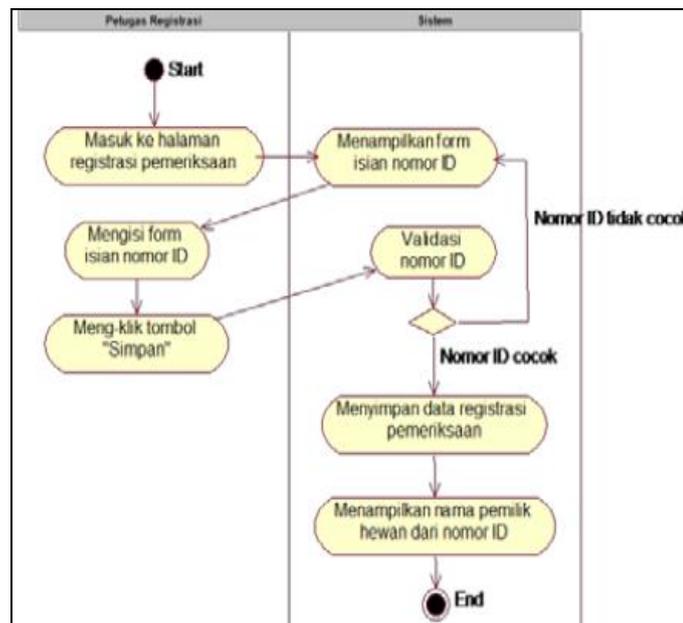
*Use case diagram* dibuat berdasarkan pengguna sistem informasi rawat jalan dan fungsi-fungsi yang ditangani oleh sistem informasi rawat jalan yang didefinisikan pada sub bab analisis sistem rawat jalan yang akan dibuat. Aktor-aktor pada *use case diagram* didapatkan dari pengguna sistem informasi rawat jalan, sedangkan *use case* untuk setiap aktor didapatkan dari fungsi-fungsi yang ditangani oleh sistem informasi rawat jalan. Contoh gambar *use case* untuk aktor petugas registrasi dan dokter pada rancangan sistem informasi rawat jalan yang akan dibuat dapat dilihat pada Gambar.



**Gambar II.6 Contoh Use Case diagram**  
**Sumber : (Haviluddin ; 2011 : 4)**

### 7. Activity Diagram

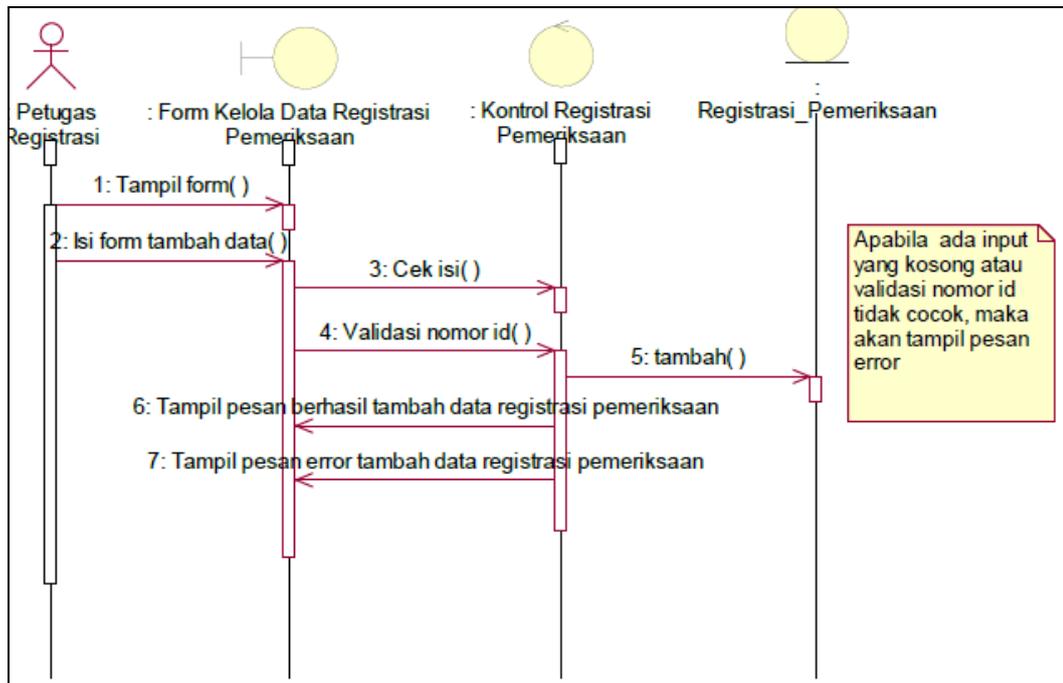
*Activity diagram* dibuat berdasarkan aliran dasar dan aliran alternatif pada skenario *use case diagram*. Pada *activity diagram* digambarkan interaksi antara aktor pada *use case diagram* dengan sistem. Contoh *activity diagram* yang dibuat berdasarkan skenario pada *use case* menambah data registrasi pemeriksaan dan menambah data pemeriksaan medis lanjutan dapat dilihat pada Gambar.



**Gambar II.7. Contoh Activity Diagram**  
 Sumber : (Haviluddin ; 2011 : 4)

## 8. Sequence Diagram

*Sequence diagram* dibuat berdasarkan *activity diagram* dan *class diagram*. *Sequence diagram* menggambarkan aliran pesan yang terjadi antar kelas yang dideskripsikan pada *class diagram* dengan menggunakan operasi yang dimiliki kelas tersebut. Untuk aliran pesan, *sequence diagram* merujuk pada alur sistem *activity diagram* yang telah dibuat sebelumnya. Berikut adalah contoh *sequence diagram* menambah data registrasi pemeriksaan :



**Gambar II.8. Contoh *Sequence Diagram***  
(Sumber : Haviluddin, 2011 ; 3)

## II.8. Database

Menurut Dermawan Wibisono (2012 : 129-130) *Database* adalah sekumpulan data mentah yang disusun menurut logika tertentu dan terorganisasi dalam bentuk yang dapat disimpan dan diproses oleh komputer. Contoh *database* dapat berisi data pegawai, data penjualan, pembayaran, dan lain-lain. Data internal dari akunting, keuangan, penjualan, dandidang-bidang bisnis lainnya yang disimpan dalam suatu sistem komputer dan disusun menurut logika tertentu disebut sebagai internal *database*.

*Database* seringkali disimpan dalam suatu perangkat tertentu pada komputer, seperti hard disk, compact disk, dan sebagainya. hubungan antarsistem *database* dan sistem software sangat kuat karena sistem *database* yang dipakai

sangat menentukan kemudahan aksesnya data sementara *software* sendiri memungkinkan peneliti memanipulasi data untuk dianalisis.

## II.9. MySQL

Menurut M. Agus dan J.Alam (2010: 181-182) MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau yang dikenal dengan DBMS (*database management system*), database ini *multithread, multi-user*. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License (GPL)*. Kekuatan MySQL tidak ditopang oleh sebuah komunitas, seperti Apache, yang dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh pemilik masing-masing, tetapi MySQL didukung penuh oleh pemilik sebuah perusahaan profesional dan komersial, yakni MySQL AB dari Swedia.

MySQL adalah *Relational Database Management System (RDBMS)* yang didistribusikan secara gratis di bawah lisensi GPL (*General Public License*). Di mana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam database sejak lama, yaitu SQL (*Structured Query Language*).

Sebagai database server, MySQL dapat dikatakan lebih unggul dibandingkan database server lainnya, terutama dalam kecepatan. Berikut ini beberapa keistimewaan MySQL, antara lain :

1. Portability

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti windows, linux, freeBSD, Mac Os X Server, Solaris, Amiga dan masih banyak lagi.

2. Multiuser

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

3. Security

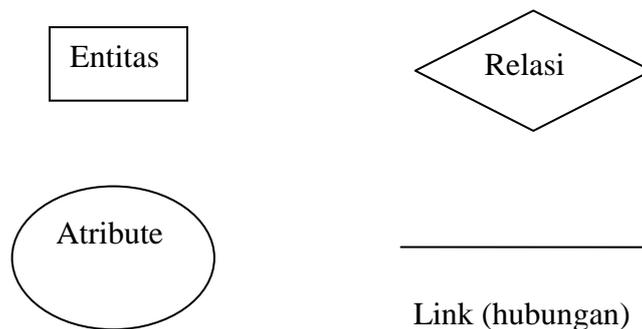
MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta password terenkripsi.

4. Scalability dan Limits

MySQL mampu menangani database dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

## **II.10. Entity Relationship Diagram (ERD)**

*Entity Relationship Diagram* atau ERD merupakan salah satu alat (tool) berbentuk grafis yang populer untuk *desain database*. Tool ini relatif lebih mudah dibandingkan dengan Normalisasi. Kebanyakan sistem analis memakai alat ini, tetapi yang jadi masalah, kalau kita cermati secara seksama, tool ini mencapai 2NF (Yuniar Supardi ; 2010 : 448).



**Gambar II.9. Komponen – Komponen ERD**  
**Sumber : Ir. Yuniar Supardi ( 2010 : 448 )**

### II.11. Kamus Data

Kamus data (data dictionary) adalah suatu ensiklopedik dari informasi yang berkaitan dengan data perusahaan, atau dapat juga kita katakan bahwa kamus data adalah katalog atau *directory* yang berbasis komputer (*computer base catalog or directory*) yang berbasis data perubahan (metadata). Yang berkenaan dengan tahapan penjelasan data ini adalah sistem kamus data (*data description language/DDDL*). Sistem kamus data berbentuk perangkat lunak yang fungsinya adalah penciptaan dan pemeliharaan serta menyediakan kamus data agar dapat digunakan. Kamus data dapat berbentuk kertas ataupun arsip (*file*) komputer (Ian Sommerville ; 2010 : 344).

### II.12. Teknik Normalisasi

Menurut Janner Simarmat dan Imam Payudi (2010 : 76) Normalisasi adalah teknik perancangan yang banyak digunakan sebagai pemandu dalam merancang basis data relasional. Pada dasarnya, normalisasi adalah proses dua

langkah yang meletakkan data dalam bentuk tabulasi dengan menghilangkan kelompok berulang lalu menghilangkan data yang terduplikasi dari tabel rasional.

Teori normalisasi didasarkan pada konsep bentuk normal. Sebuah tabel relasional dikatakan berada pada bentuk normal tertentu jika tabel memenuhi himpunan batasan tertentu. Ada lima bentuk normal yang telah ditemukan

### II.12.1. Bentuk-bentuk Normalisasi

#### a. Bentuk tidak normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi. Data dikumpulkan apa adanya sesuai keadaanya.

#### b. Bentuk normal tahap pertama (1<sup>st</sup> Normal Form)

Definisi :

Sebuah table disebut 1NF jika :

- Tidak ada baris yang duplikat dalam tabel tersebut.
- Masing-masing cell bernilai tunggal

Catatan: Permintaan yang menyatakan tidak ada baris yang duplikat dalam sebuah tabel berarti tabel tersebut memiliki sebuah kunci, meskipun kunci tersebut dibuat dari kombinasi lebih dari satu kolom atau bahkan kunci tersebut merupakan kombinasi dari semua kolom.

#### c. Bentuk normal tahap kedua (2<sup>nd</sup> normal form)

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh.

d. Bentuk normal tahap ketiga (3<sup>rd</sup> normal form)

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi  $X \rightarrow A$ , dimana A mewakili semua atribut tunggal di dalam tabel yang tidak ada di dalam X, maka :

- X haruslah *superkey* pada tabel tersebut.
- Atau A merupakan bagian dari *primary key* pada tabel tersebut.

e. Bentuk Normal Tahap Keempat dan Kelima

Penerapan aturan normalisasi sampai bentuk normal ketiga sudah memadai untuk menghasilkan tabel berkualitas baik. Namun demikian, terdapat pula bentuk normal keempat (4NF) dan kelima (5NF). Bentuk Normal keempat berkaitan dengan sifat ketergantungan banyak nilai (*multivalued dependency*) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal tahap kelima merupakan nama lain dari *Project Join Normal Form* (PJNF).

f. Boyce Code Normal Form (BCNF)

- Memenuhi 1<sup>st</sup> NF
- Relasi harus bergantung fungsi pada atribut superkey (Janner Simarmat dan Imam Payudi ; 2010 : 78-86).