

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **II.1. Sistem**

Secara umum, sistem dapat didefinisikan sebagai sekumpulan objek, ide, berikut saling keterkaitannya (*inter-relasi*) didalam (usaha) mencapai suatu tujuan (atau sasaran bersama tertentu). Atau, dengan kata lain, sistem dapat disebutkan sebagai kumpulan komponen (sub-sistem fisik maupun non-fisil/logika) yang saling berhubungan satu sama lainnya dan bekerja samasecara harmonis untuk mecapai suatu tujuan (Edy Prahasta : 2009 : 89).

#### **II.2. Data Dan Informasi**

Isitlah data dan informasi sering sekali digunakan secara bergantian dan saling bertukar, meskipun kedua istilah ini sebenarnya merujuk pada masing-masing konsep yang berbeda. Data merupakan bahasa, mathematical, dan atau simbol-simbol pengganti lain yang (telah) disepakati secara umum didalam (usaha) menggambarkan suatu objek, manusia, peristiwa, aktivitas, konsep, atau objek-objek penting lainnya. Singkatnya, data merupakan suatu kenyataan apa adanya (*raw facts*). Sedangkan informasi adalah data yang (telah) ditempatkan pada konteks yang penuh arti oleh sipenerimaanya (Edy Prahasta : 2009 : 78).

### **II.3. Sistem Informasi**

Pada saat ini tentunya hampir semua organisasi telah memiliki sistem informasi. Sistem informasi ini merupakan sebuah entitas (kesatuan) formal yang terdiri dari berbagai sumber daya fisik maupun logika. Dari organisasi ke organisasi, sumber daya ini disusun atau distrukturkan dengan beberapa cara (yang bisa belainan satu sama lainnya) karena suatu organisasi dan sistem informasi terkait merupakan sumber daya yang bersifat dinamis (Edy Prahasta : 2009 : 93).

Tujuan sistem informasi adalah untuk menyediakan dan mensistematikkan informasi yang merefleksikan seluruh kejadian atau kegiatan yang diperlukan untuk mengendalikan operasi suatu kegiatan. Sedangkan kegiatan yang dimaksud adalah mengambil, mengolah, menyimpan dan menyampaikan informasi (komunikasi) yang diperlukan didalam mengoperasikan seluruh aktifitas organisasi yang bersangkutan (Edy Prahasta : 2009 : 98).

### **II.4. Sistem Informasi Geografis**

Defenisi SIG (kemungkinan besar) masih berkembang, bertambah, dan sedikit bervariasi. Hal ini terlihat dari banyaknya defenisi SIG yang telah beredar diberbagai sumber pustaka. Lebih dari itu, SIG juga merupakan suatu bidang kajian ilmu dan teknologi yang belum terlalu lama dikembangkan, digunakan oleh berbagai bidang atau disiplin ilmu dan berkembang dengan cepat.

SIG adalah sistem komputer (SBIS) yang digunakan untuk memasukkan (*capturing*), menyimpan, memeriksa, mengintegrasikan, memanipulasi,

menganalisis, dan menampilkan data-data yang berhubungan dengan posisi-posisinya dipermukaan (Edy Prahasta : 2009 : 116).

## II.5. ArcView

Kemampuan *Arcview* GIS pada berbagai serinya tidaklah diragukan lagi. *Arcview* GIS adalah software yang dikeluarkan oleh ESRI (*Environmental Systems Research Institute*). Perangkat lunak ini memberikan fasilitas teknis yang berkaitan dengan pengolahan data spasial. Kemampuan grafis yang baik dan kemampuan teknis dalam pengolahan data spasial tersebut memberikan kekuatan secara nyata pada *Arcview* untuk melakukan analisis spasial. Kekuatan analisis inilah yang pada akhirnya menjadikan *Arcview* banyak diterapkan dalam berbagai pekerjaan, seperti analisis pemasaran, perencanaan wilayah dan tata ruang, sistem informasi persis, pengendalian dampak lingkungan, bahkan untuk keperluan militer. Mengapa *Arcview* dapat memiliki keluwesan yang sedemikian hebat? Hal itu disebabkan oleh adanya dukungan dari skrip *Avenue*. Melalui *avenue* ini dapat dibentuk suatu “kemampuan baru” pada *Arcview*. Tentu saja hal ini membuat *Arcview* menjadi sangat luwes untuk diterapkan pada berbagai permasalahan spasial. *Avenue* dapat digunakan untuk “merombak” wajah *Arcview* sesuai kebutuhan penggunaanya.

*Dialog designer* diperlukan untuk membentuk antarmuka penampil data atribut yang menjadi dasar pemilihan objek. *Dialog designer* yang dipilih adalah bentuk kotak daftar (*listbox*). Dengan menggunakan dialog ini operator akan memilih informasi apa yang akan dicari. Untuk menghubungkan menu dan

tombol dengan berbagai aksi yang diinginkan maka perlu dibentuk skrip atau program. Skrip atau program ini dibentuk menggunakan bahasa Avenue. Setiap aksi yang diperlukan diuraikan menjadi baris-baris perintah pada skrip Avenue dan selanjutnya dikaitkan ke masing-masing menu atau tombol yang bersangkutan (Eko Budiyanto ; 2010 : 178).

## **II.6. PHP & My SQL**

*PHP : Hypertext Processor* adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP banyak dipakai untuk mengoperasikan situs *web* dinamis. PHP dapat digunakan untuk membangun sebuah CMS. Pada awalnya PHP merupakan kependekatan dari personal home page (situs personal). PHP pertama kali dibuat oleh rumus lerdorf pada tahun 1995. Pada waktu itu PHP masih bernama form interorented (FI), yang wujudnya berupa sekumpulan skrip yang digunakan untuk mengolah data formulir web (Alan Nur Aditya : 2011 : 1).

Setelah belajar teknik dasar html kini kita belajar dasar My SQL, My SQL adalah sebuah perangkat lunak sistem berbasis data SQL (bahasa inggirs : *database management system*) atau DBMS yang multithread, multiuser, dengan sekitar 6 juta instalasi di seluruh dunia. My SQL AB membuat My SQL tersedia sebagai perangkat lunak gratis dibawa lisensi GNU general public license (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasis-kasus dimana penggunaanya tidak cocok dengan penggunaan GPL (Alan Nur Aditya : 2011 : 61).

## II.7. Unified Modelling Language

Menurut Prabowo Pudjo Widodo dan Herlawati (2011 : 6-7), *UML* singkatan dari *Unified Modelling Language* yang berarti bahasa permodelan standar. UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

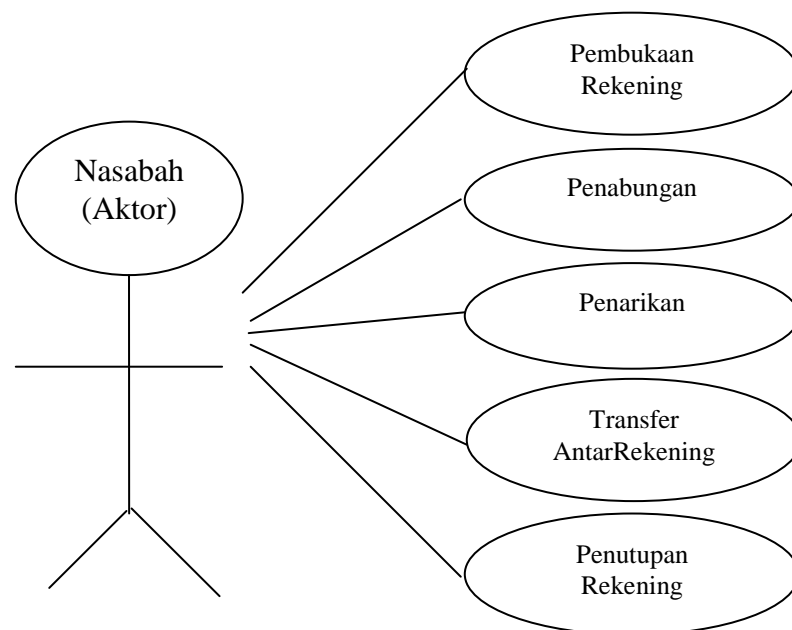
1. Merancang perangkat lunak
2. Sarana Komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

### II.7.1. Use Case Diagram

Segala sesuatu yang secara akademis dikembangkan pada umumnya berawal dari suatu konsep. Demikian juga halnya dengan pengembangan sistem pada umumnya dikembangkan berdasarkan analisis kebutuhan. Analisis kebutuhan ini adalah tahap konseptualisasi, yaitu suatu tahap yang mengharuskan analis dan perancang sistem untuk berusaha tahu secara pasti mengenai hal yang menjadi kebutuhan dan harapan pengguna sehingga kelak aplikasi yang dibuat memang akan digunakan oleh pengguna (*user*) serta akan memuaskan kebutuhan dan harapannya.

Dalam konteks UML, tahap konseptualisasi dilakukan dengan pembuatan *use case diagram* yang sesungguhnya merupakan deskripsi peringkat tinggi bagaimana perangkat lunak (aplikasi) akan digunakan oleh penggunanya. Selanjutnya, *use case diagram* tidak hanya sangat penting pada saat analisis,

tetapi juga sangat penting dalam tahap perancangan (*design*), untuk mencari kelas-kelas yang terlibat dalam aplikasi, dan untuk melakukan pengujian (*testing*). Saat akan mengembangkan *use case diagram*, hal yang pertama kali harus dilakukan adalah mengenali *actor* untuk sistem yang sedang dikembangkan. Dalam hal ini, ada beberapa karakteristik untuk para *actor*, yaitu *actor* yang ada di luar sistem yang sedang dikembangkan dan *actor* yang berinteraksi dengan sistem yang sedang dikembangkan. (Adi Nugroho ; 2009 : 7)



**Gambar II.1. Contoh Use Case Diagram**

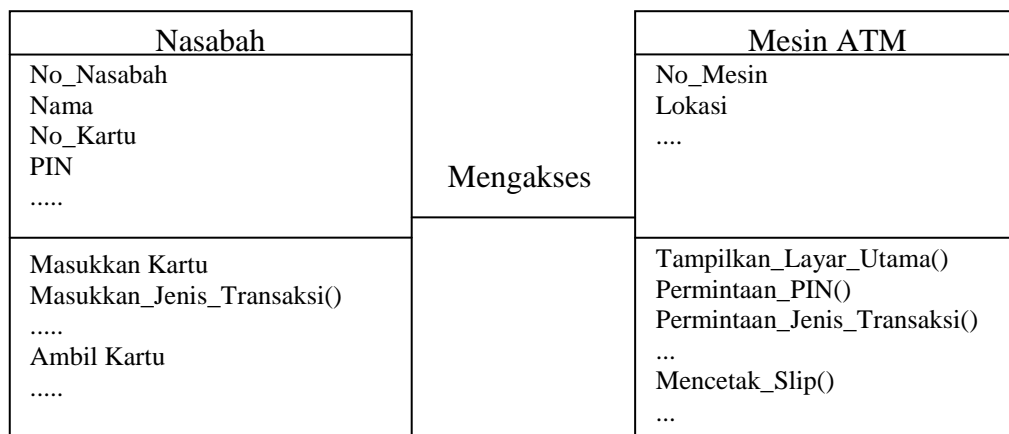
(Sumber : Adi Nugroho ; 2009 : 8)

### II.7.2. Class Diagram

*Class* didefinisikan sebagai kumpulan/himpunan objek yang memiliki kesamaan dalam atribut/properti, perilaku (operasi), serta cara berhubungan dengan objek lain (Adi Nugroho ; 2009 : 18).

Selain itu, kita juga mendefenisikan objek sebagai konsep, abstraksi dari sesuatu dengan batas nyata, sehingga kita dapat menggambarkan secara sistematis. Pemahaman objek memiliki dua fungsi, yaitu :

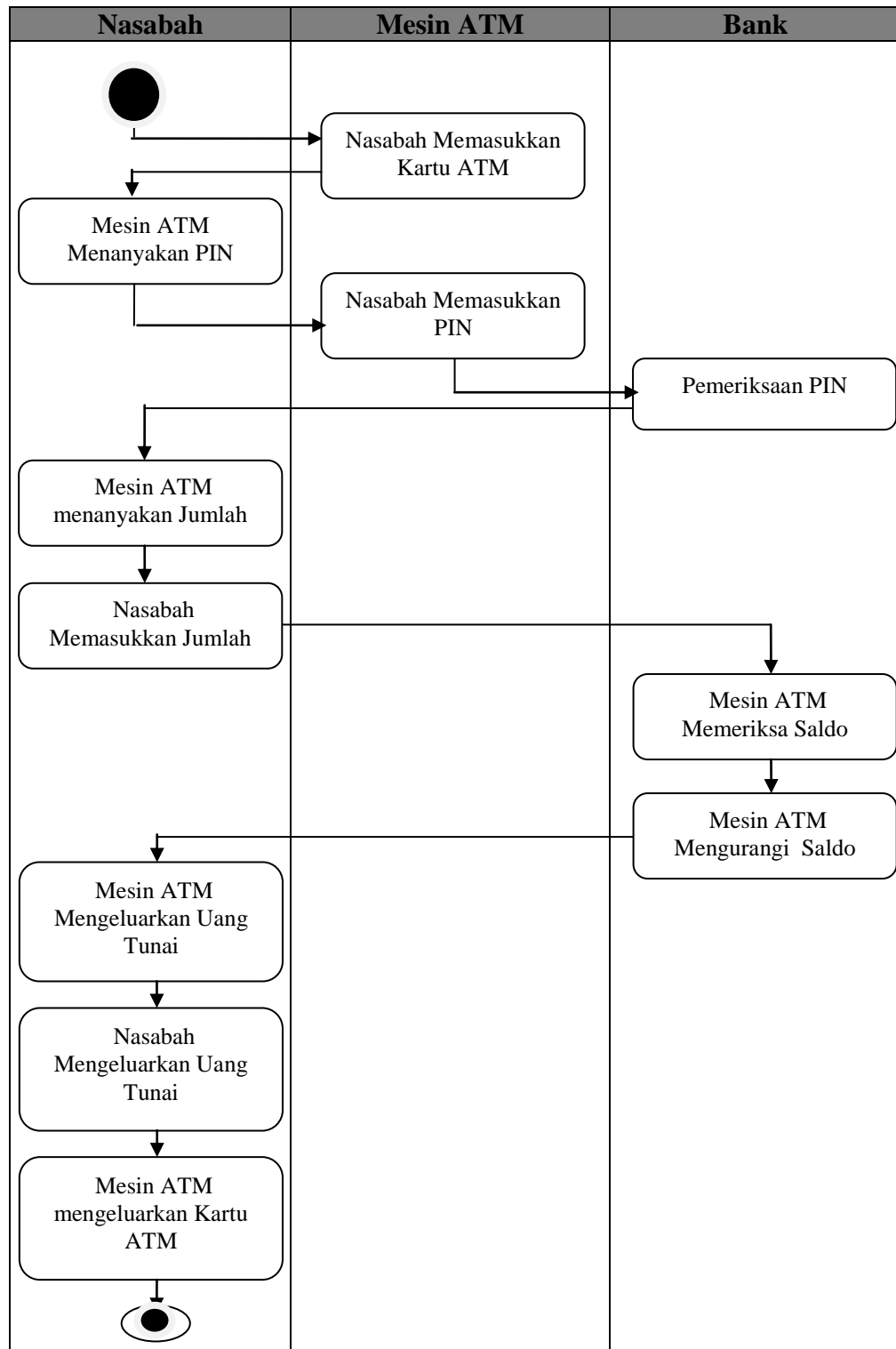
- a. Memudahkan untuk mempelajari secara seksama hal-hal yang ada di dunia nyata.
- b. Menyediakan suatu dasar yang kuat dalam implementasi ke dalam sistem terkomputerisasi (Adi Nugroho ; 2009 :17).



**Gambar II.2. Contoh Class Diagram**  
(Sumber : Adi Nugroho ; 2009: 39)

### II.7.3. Activity Diagram

Apakah langkah yang harus kita lakukan selanjutnya setelah kita membuat *use case diagram*, *use case diagram* merupakan gambaran menyeluruh dan pada umumnya sangatlah tidak terperinci. Apa perkakas (*tool*) yang bisa kita gunakan, jika kasus kita cukup sederhana, mungkin kita bisa menggunakan skenario seperti yang tercantum berikut, sementara jika kasusnya cukup kompleks, kita mungkin bisa menggunakan *activity diagram* agar bisa mendapatkan gambaran yang lebih menyeluruh (Adi Nugroho ; 2009 : 10).



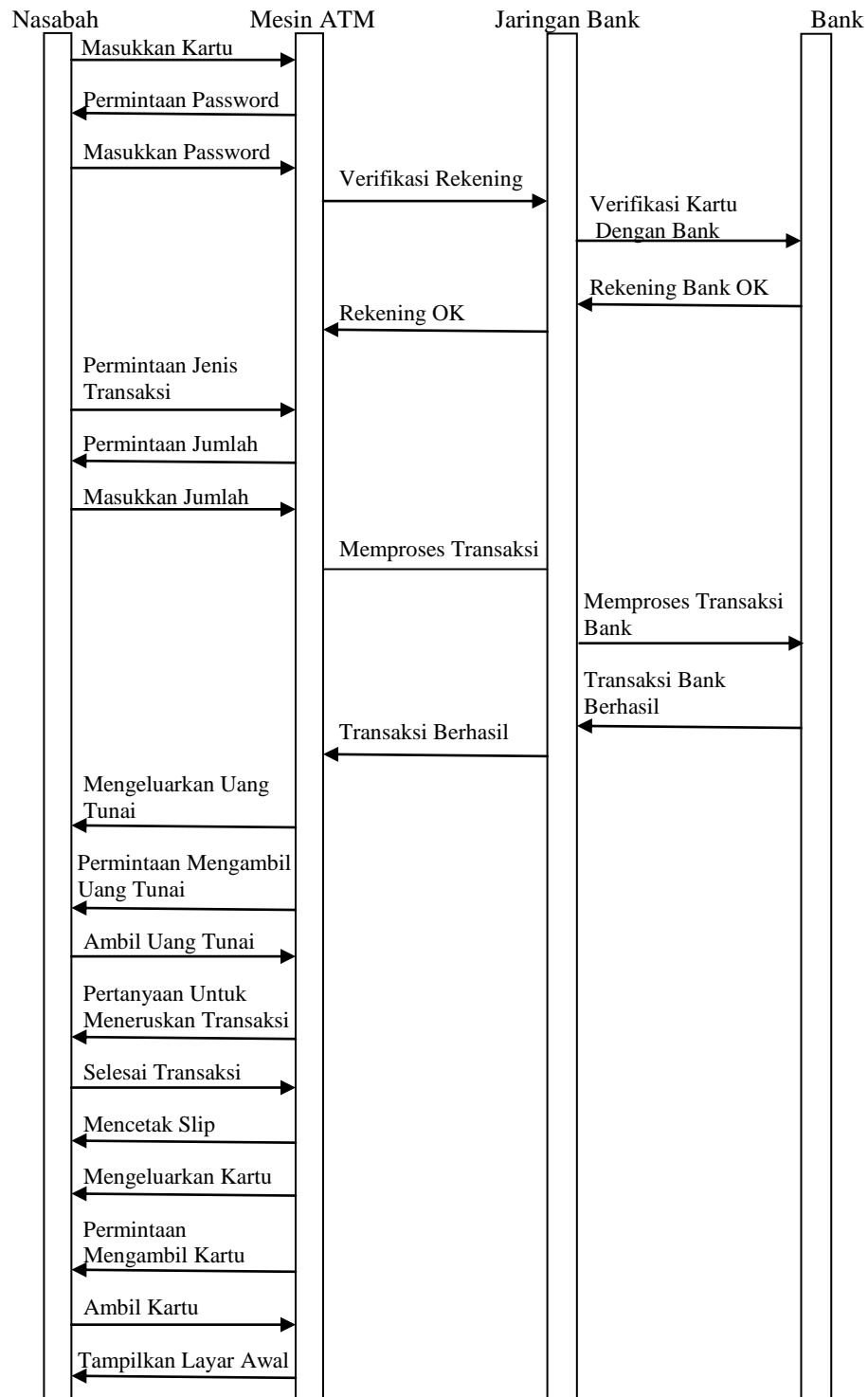
**Gambar II.3. Contoh Activity Diagram**

(Sumber : Adi Nugroho ; 2009 : 11)



#### II.7.4. Sequence Diagram

Dalam terminologi penggunaan UML, kita bisa menggambarkan *sequence diagram/collaboration diagram* untuk menentukan metoda/fungsi yang kelak dapat diterapkan pada suatu kelas/objek. Dengan gambar tersebut, kita bisa melakukan pemetaan aliran event antarkelas/objek. *Event-event* itu dapat kita anggap sebagai metode/fungsi yang kelak dapat diterapkan dalam suatu kelas/objek. Kita tinggal melihat ke mana arah event terjadi. Sebagai contoh, *event* “masukkan kartu” mengalir dari nasabah ke mesin ATM sehingga *event* ini dapat dianggap sebagai metode untuk kelas nasabah, *event* “permintaan pin” mengalir dari mesin ATM ke nasabah sehingga *event* ini dapat dianggap sebagai metoda untuk kelas mesin Atm, dan seterusnya. Dalam hal ini, tentunya tidak semua metoda yang dijumpai dalam skenario merupakan metoda yang bisa diimplementasikan dalam bentuk program komputer. *Event* “memasukkan kartu”, misalnya bersifat mekanis sehingga tidak perlu diimplementasikan dalam program komputer (kecuali jika *event* ini memerlukan program komputer untuk menggerakkan mesin ATM untuk menerima kartu).



**Gambar II.4. Contoh Sequence Diagram**  
(Sumber : Adi Nugroho ; 2009 : 36)

## II.8. Kamus Data

Menurut Edhy Sutanta (2011 : 25) Sebelum memperoleh defenisi formal basis data, kita akan mencoba memahaminya secara sederhana terlebih dahulu. Istilah basis data tersusun atas dua suku kata, yaitu basis dan data (basis data = basis + data). Dalam sistem bilangan biner, kita dapat menuliskan beberapa contoh bilangan sebagai berikut.

0 → sama dengan 0 dalam sistem bilangan desimal

1 → sama dengan 1 dalam sistem bilangan desimal

10 → sama dengan 2 dalam sistem bilangan desimal

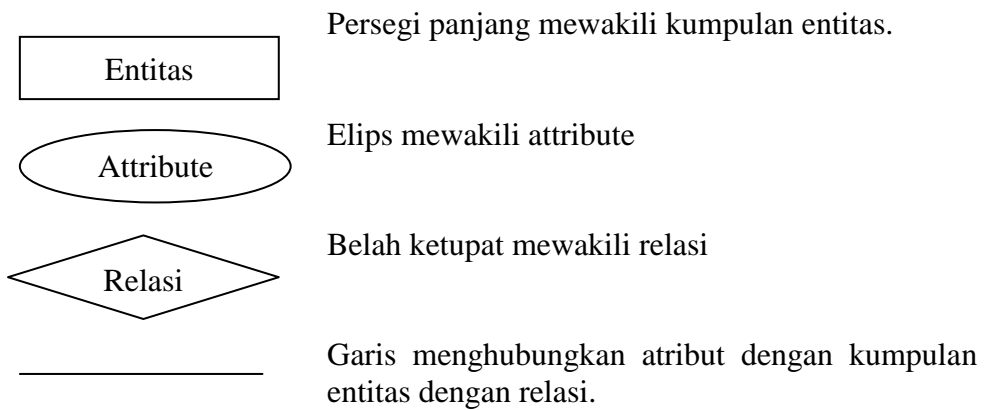
11 → sama dengan 3 dalam sistem bilangan desimal

100 → sama dengan 4 dalam sistem bilangan desimal

## II.9. Entity Relationship Diagram

Menurut Edhy Sutanta (2011 : 91) *Entity Relationship Diagram/ER\_M* merupakan suatu model data yang dikembangkan berdasarkan objek. ER\_M digunakan untuk menjelaskan hubungan antara data dalam basis data kepada pengguna secara logik. ER\_M didasarkan pada suatu persepsi bahwa *real world* terdi atas objek-objek dasar yang mempunyai hubungan/kerelasian antar objek-objek dasar tersebut. ER\_M digambarkan dalam bentuk diagram yang disebut dengan ER (*ER\_Diagram/ER\_D*). Untuk menggambarkan ER\_D diguna-kan simbol-simbol grafis tertentu.

Menurut Rosa A.S dan M. Shalahuddin (2011 : 60), Struktur *logis* (skema *database* dapat ditunjukkan secara *grafis* dengan ER yang dibentuk dari komponen-komponen berikut ini :



## II.10. Normalisasi

Menurut Edhy Sutanta (2011 : 174) Normalisasi diartikan sebagai suatu teknik yang menstrukturkan atau mendekomposisikan data dalam cara-cara tertentu untuk mencegah timbulnya pemasalahan dalam pengolahan data dalam basis data. Permasalahan yang dimaksud adalah berkaitan dengan penyimpangan-penyimpangan (*anomalies*) yang terjadi akibat adanya kerangkapan data dalam relasi dan in-efisiensi pengolahan. Proses normalisasi menghasilkan relasi yang optimal yaitu :

1. Memiliki struktur record yang konsisten secara logik
2. Memiliki struktur record yang mudah untuk dimengerti
3. Memiliki struktur record yang sederhana dalam pemeliharaan
4. Memiliki struktur record yang mudah ditampilkan kembali untuk memenuhi kebutuhan pengguna
5. Minimalisasi kerangkapan data guna meningkatkan kinerja sistem.

Menurut Yuniar Supardi (2008 : 10) tahapan normalisasi terdiri dari beberapa bentuk, yaitu:

1. Bentuk Tak Normal (UNF / *Un Normal Form*).

2. Bentuk Normal Pertama (1 NF / *First Normal Form* ).

Bentuk Normal pertama memiliki ciri: Data berbentuk *file-file* (*file* datar), *record* disusun sesuai kedatangan, masih mungkin terjadi penyimpangan data (anomali data ). Anomali data dapat berupa insert *anomali*, *delete anomali*, *update, anomali*, dan *redundancy data* (data duplikat).

3. Bentuk Normal Kedua (2 NF / *Second Normal Form*).

Bentuk Normal kedua memiliki ciri; Tidak terjadi anomali data, setiap *field*/atribut bukan kunci harus tergantung fungsi (*Functional Dependency*) terhadap *field*/atribut kunci, masih mungkin terjadi *transitive dependency* (*field* bukan kunci tergantung pada *field* bukan kunci dalam satu *table*). Model objek mencapai bentuk normal kedua, sehingga penulis mendesain mulai bentuk normal ketiga dan bentuk normal boyce codd. Sedangkan untuk bentuk tak normal sudah dari dokumen dasar berupa Faktur, Nota, dan laporan *Stock of Name*.

4. Bentuk Normal Ketiga (3 NF / *Third Normal Form*).

Table yang memenuhi Bentuk Normal Ketiga harus tidak terdapat *Transitive Dependency*. Bentuk normal ketiga dari sistem *inventory* :

5. Bentuk Normal *Boyce Codd* (BCNF / *Boyce Codd Normal Form*).

Karena tak ada *field* bukan kunci tergantung secara parsial (bagian) kunci dalam satu tabel, maka bentuk normal ketiga juga merupakan bentuk BCNF.

6. Normal yang lebih tinggi.

Setelah 3NF, semua masalah normalisasi hanya melibatkan tabel yang mempunyai tiga kolom atau lebih dan semua kolom adalah kunci. Bentuk Normal Boyce-Code (BCNF) adalah versi 3NF yang lebih teliti dan berhubungan dengan tabel relasional yang mempunyai banyak kunci kandidat, kunci kandidat gabungan, dan kunci kandidat yang saling tumpang tindih.