

BAB II

TINJAUAN PUSTAKA

II.1 Sistem

Menurut (I Putu Agus Eka Pratama; 2014:7) Sistem didefinisikan sebagai sekumpulan prosedur yang saling berkaitan dan saling terhubung untuk melakukan suatu tugas bersama-sama.

II.2 Sistem Informasi Akuntansi

Dengan melihat kata-kata penyusun nama Sistem Informasi Akuntansi, maka nama tersebut dapat dijabarkan sebagai berikut :

1) Sistem

Sebuah sistem adalah suatu himpunan komponen atau variabel yang terorganisasi, saling berinteraksi, saling bergantung satu sama lain dan terpadu. Sebuah sistem mempunyai tujuan dan sasaran. Suatu sistem terdiri dari komponen-komponen yaitu pekerjaan, kegiatan, misi atau bagian-bagian sistem yang dibentuk untuk mewujudkan tujuan.

2) Informasi

Informasi merupakan sesuatu yang nyata atau setengah nyata yang dapat mengurangi derajat ketidak pastian tentang suatu keadaan dan kejadian. Informasi juga sebagai data yang telah diorganisasikan, dan telah memiliki kegunaan dan manfaat. Informasi merupakan hal yang sangat penting di dalam pengambilan keputusan. Informasi tersebut diperoleh dari sistem

informasi (*Information System*). Jadi kesimpulannya, informasi adalah data yang sudah diolah, dibentuk atau dimanipulasikan sehingga dapat dijadikan dasar untuk mengambil keputusan yang tepat.

3) Akuntansi

Menurut Dra. Lanita Winata dalam bukunya yang berjudul Pengantar Akuntansi I mengatakan bahwa Akuntansi adalah suatu sistem keterangan keuangan yang memberikan informasi yang sangat dibutuhkan agar suatu organisasi dapat beroperasi secara efisien dan dapat mengavaluasi aktivitas-aktivitasnya.

Dari penjelasan tersebut dapat diambil simpulan bahwa akuntansi adalah proses pengidentifikasian, pengukuran dan melaporkan informasi ekonomi untuk memungkinkan adanya pembuatan pertimbangan dan keputusan bagi yang menggunakan informasi.

Kutipan diatas menjelaskan bahwa sistem informasi akuntansi adalah kumpulan dari sumber-sumber seperti orang dan peralatan yang dirancang untuk mentransformasikan data keuangan dan data lainnya menjadi informasi, dan informasi ini akan dikomunikasikan kepada para pembuat keputusan.

Manfaat Sistem Informasi Akuntansi tersebut adalah sebagai berikut :

1. Mengamankan harta/kekayaan perusahaan.
2. Menghasilkan beragam informasi untuk pengambilan keputusan.
3. Menghasilkan informasi untuk pihak eksternal
4. Menghasilkan informasi untuk penilaian kinerja karyawan atau divisi.
5. Menyediakan data masa lalu untuk kepentingan audit (pemeriksaan)

6. Menghasilkan informasi untuk penyusunan data evaluasi anggaran perusahaan.
7. Menghasilkan informasi yang diperlukan dalam kegiatan perencanaan dan pengendalian.

II.3 Visual Basic 2005

Microsoft Visual Basic 2005 adalah bahasa pemrograman untuk membuat aplikasi berbasis windows, aplikasi berbasis web, dan aplikasi mobile seperti untuk pocket PC dan smartphone.VB. NET dibangun di atas fondasi Framework.NET (lingkungan kerja .NET). (Ario Suryo Kusumo, 2006:1)

Microsoft menciptakan *.NET Framework* dengan beberapa tujuan, antara lain :

1. Lingkungan pengembangan program multibahasa : *.NET Framework* mendukung beberapa bahasa pemrograman seperti VB, C#, C++, dan lain-lain.
2. Mendukung penuh konsep object oriented programming, semua yang ada di *.NET Framework* adalah object.
3. Menyederhanakan pemrograman pada *platform windows*.
4. Menggunakan *managed code* sehingga penulisan kode menjadi lebih aman.

II.4 Microsoft SQL Server 2005

SQL Server 2005 adalah RDBMS (*Relational Database Management System*) dengan arsitektur *client server* yang disertai dengan berbagai komponen dan

/layanan yang menjadikannya *platform* yang *komprehensif* (memiliki cakupan luas) untuk aplikasi enterprise. (Ario Suryo Kusumo, 2006:257)

Semua DBMS (*Database Management System*) modern saat ini memakai SQL (*Structured Query Language*) sebagai bahasa untuk memprogram *database*. SQL *server* 2005 yang merupakan produk mikrosoft adalah salah satu jenis *database* yang banyak digunakan di Indonesia. Produk ini mudah digunakan dan mendukung aplikasi dengan arsitektur *client/server*.

II.5 Unified Modeling Language (UML)

Menurut (Adi Nugroho ; 2009 : 10) UML Adalah perkakas untuk analisis dan perancangan yang sesungguhnya digunakan untuk peyederhanaan permasalahan. UML merupakan metodologi kolaborasi antara metoda-metoda *Booch*, *OMT (Object Modeling Technique)*,serta *OOSE (Object Oriented Software Engineering)* dan beberapa metoda lainnya, merupakan metodologi yang paling sering di gunakan pada saat ini untuk mengadaptasi maraknya penggunaan bahasa pemrograman berorientasi objek.

Menurut (prabowo Pudjo Widodo & Herlawati;2011:6) UML diaplikasikan untuk maksud tertentu, biasanya antara lain :

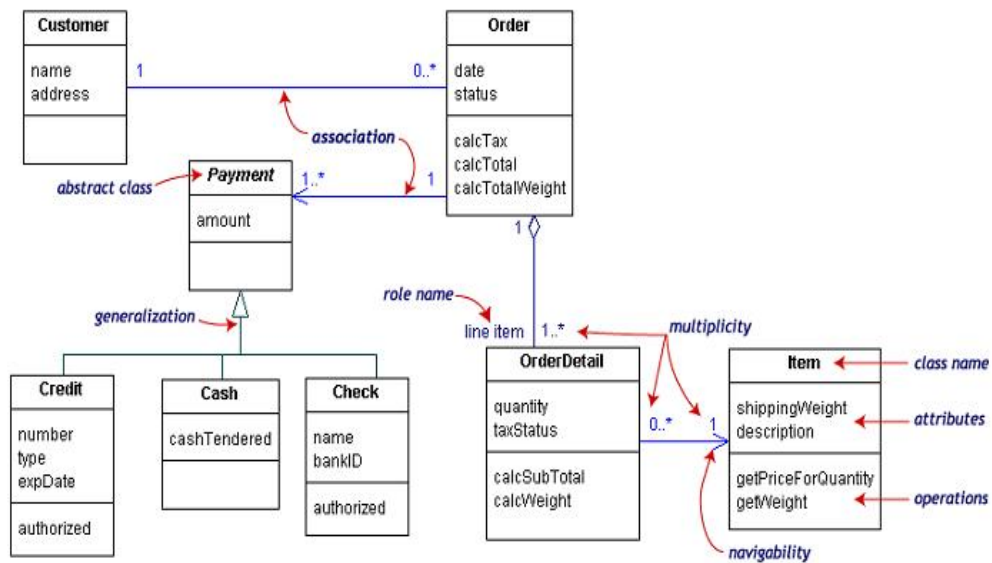
1. Merancang perangkat Lunak.
2. Sarana Komunikasi antaraperangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

Blok pembangunan utama UML adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasi objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. UML memungkinkan para anggota team untuk bekerja sama dengan bahasa model yang sama dengan mengaplikasikan beragam sistem. Intinya UML merupakan alat komunikasi yang konsisten dalam mendukung para pengembang sistem saat ini.

II.5.1 Diagram-Diagram UML

Beberapa literatur menyebutkan bahwa UML menyediakan sembilan jenis diagram, model-model ini dapat dikelompokkan berdasarkan sifatnya yaitu statis atau dinamis. Jenis diagram itu antara lain :

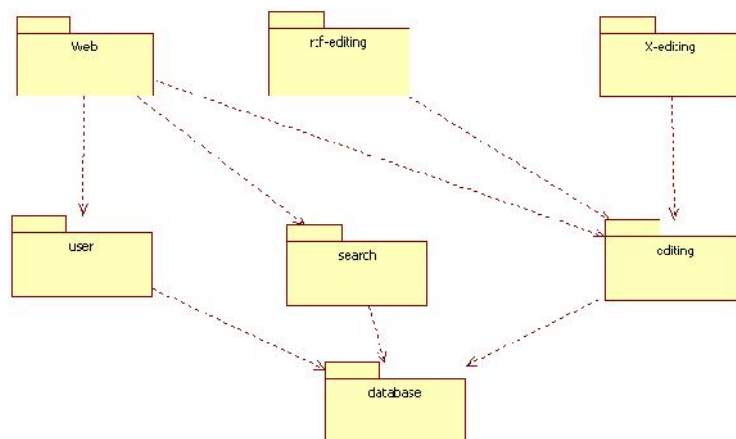
1. Diagram Kelas : Bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka-antarmuka, kolaborasi-kolaborasi, serta relasi-relasi. Diagram ini umum dijumpai pada pemodelan sistem berorientasi objek. Meskipun bersifat statis, sering pula diagram kelas memuat kelas-kelas aktif.



Gambar II.1 : Diagram Kelas

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

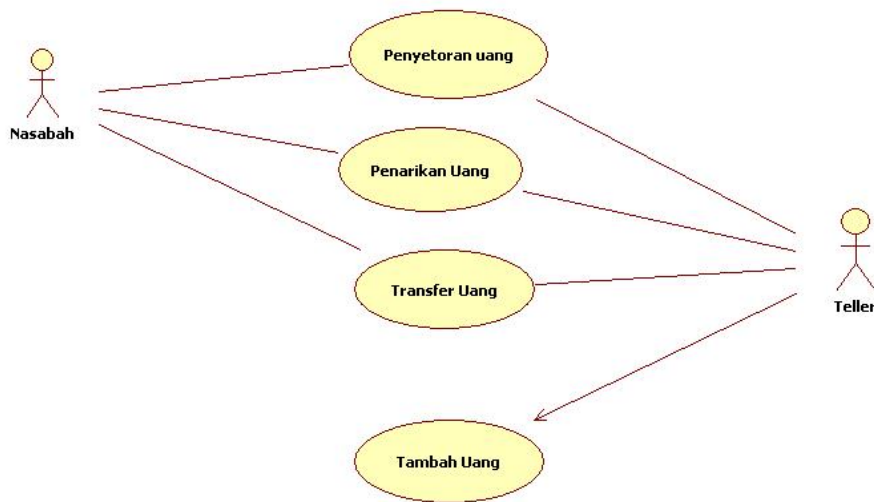
2. Diagram Paket (*Package Diagram*) : Bersifat statis. Diagram ini memperlihatkan kumpulan kelas-kelas, merupakan bagian dari diagram komponen.



Gambar II.2 : Diagram Paket

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

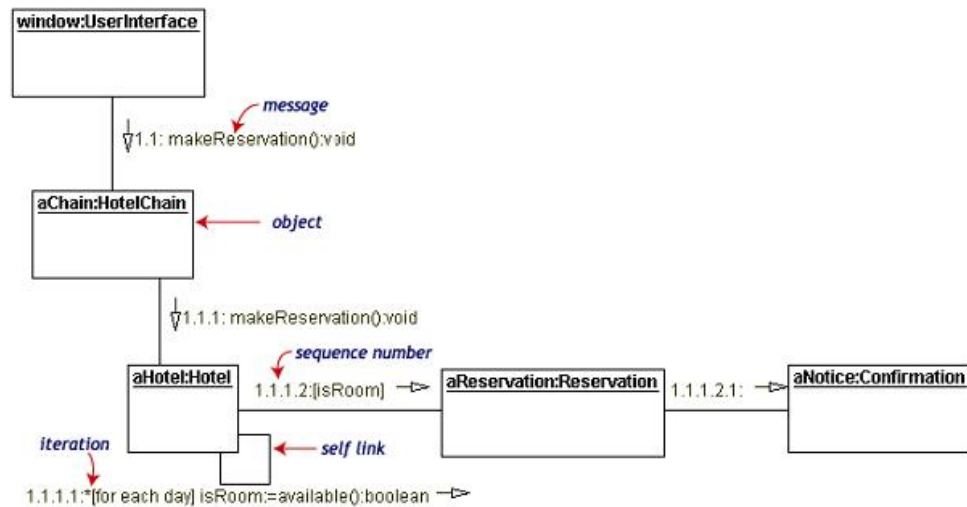
3. Diagram *Use-Case* : Bersifat statis. Diagram ini memperlihatkan himpunan *use-case* dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat penting untuk mengorganisasi dan memodelkan perilaku suatu sistem yang dibutuhkan serta diharapkan pengguna.



Gambar II.3 : Diagram *Use Case*

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

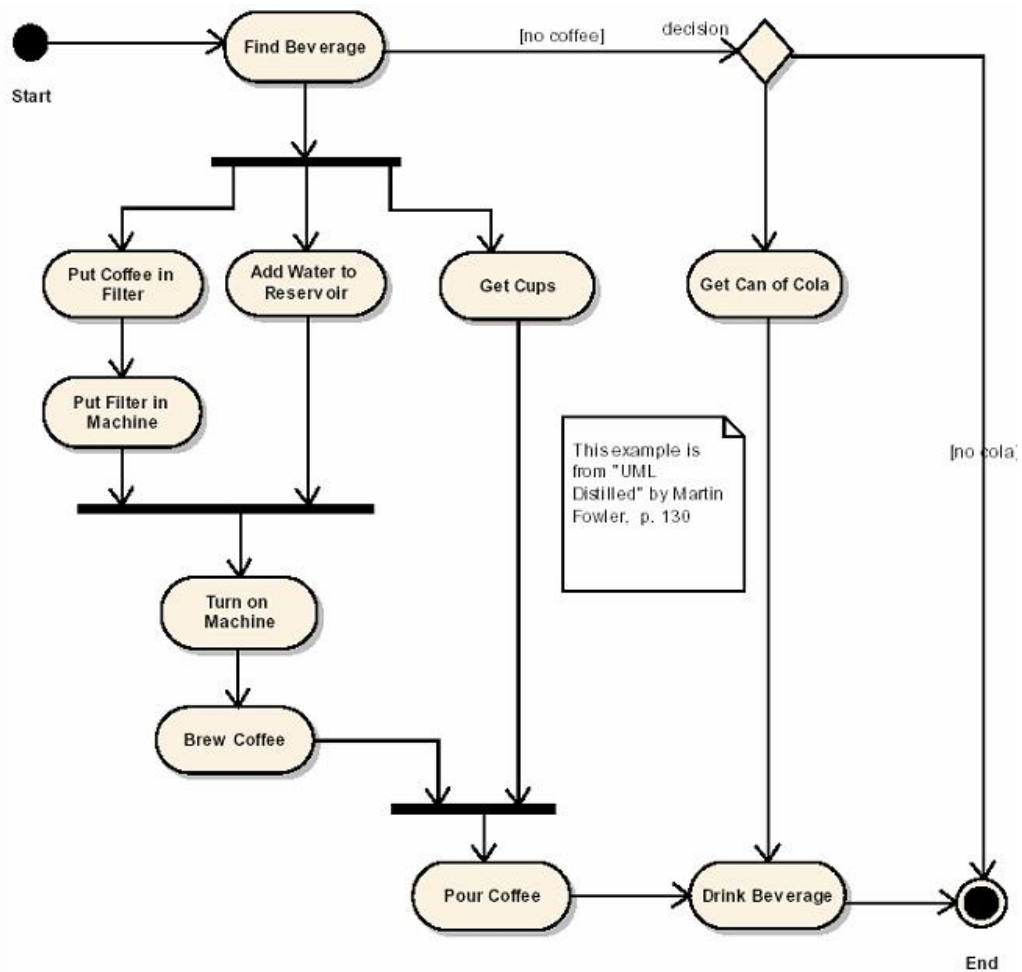
4. Diagram Interaksi dan *Sequence* (Urutan) : Bersifat dinamis. Diagram urutan adalah diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu.
5. Diagram Komunikasi (*Communication Diagram*) : Bersifat dinamis. Diagram sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari objek-objek yang menerima serta mengirim pesan.



Gambar II.4 : Diagram Komunikasi

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

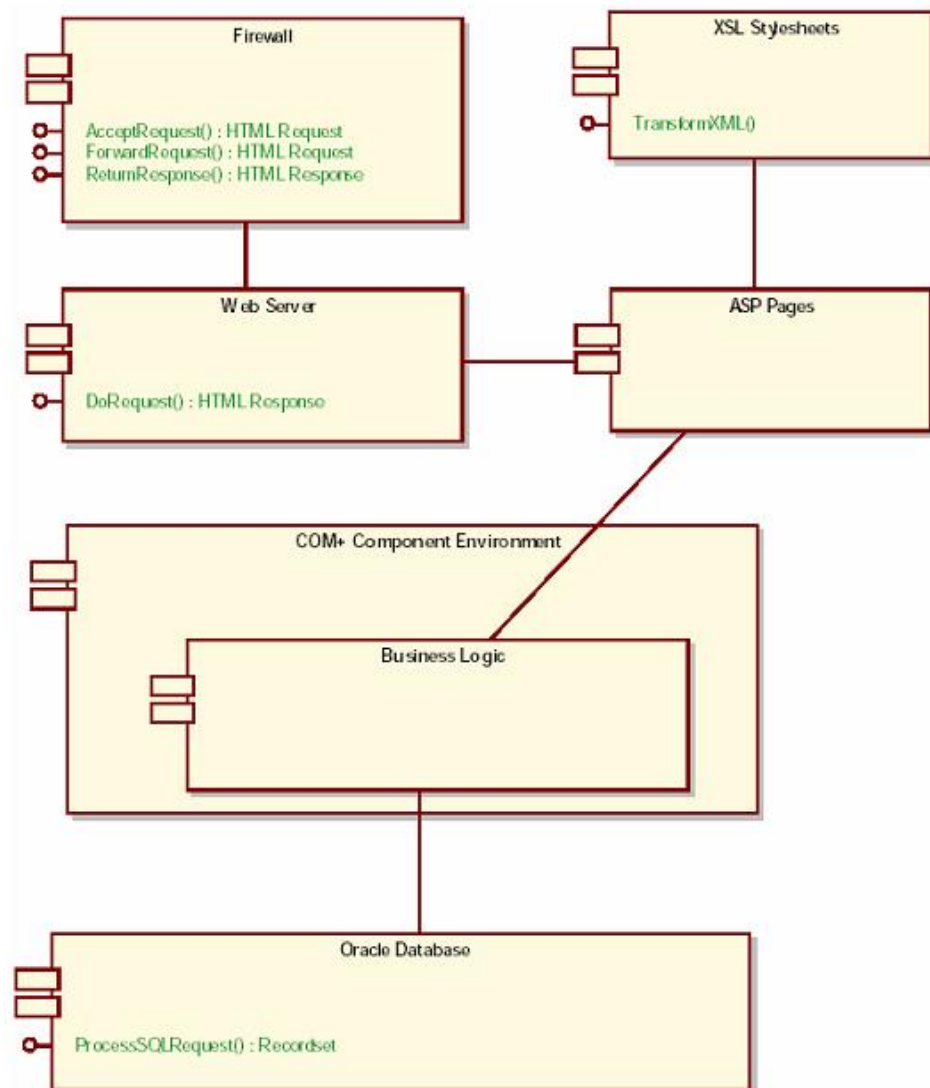
6. Diagram *Statechart* (*Statechart Diagram*) : Bersifat dinamis. Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (*state*), transisi, kejadian serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka (*interface*), kelas, kolaborasi dan terutama penting pada pemodelan sistem-sistem yang reaktif.
7. Diagram Aktivitas (*Activity Diagram*). Bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antar objek.



Gambar II.5 : Diagram Aktivitas

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

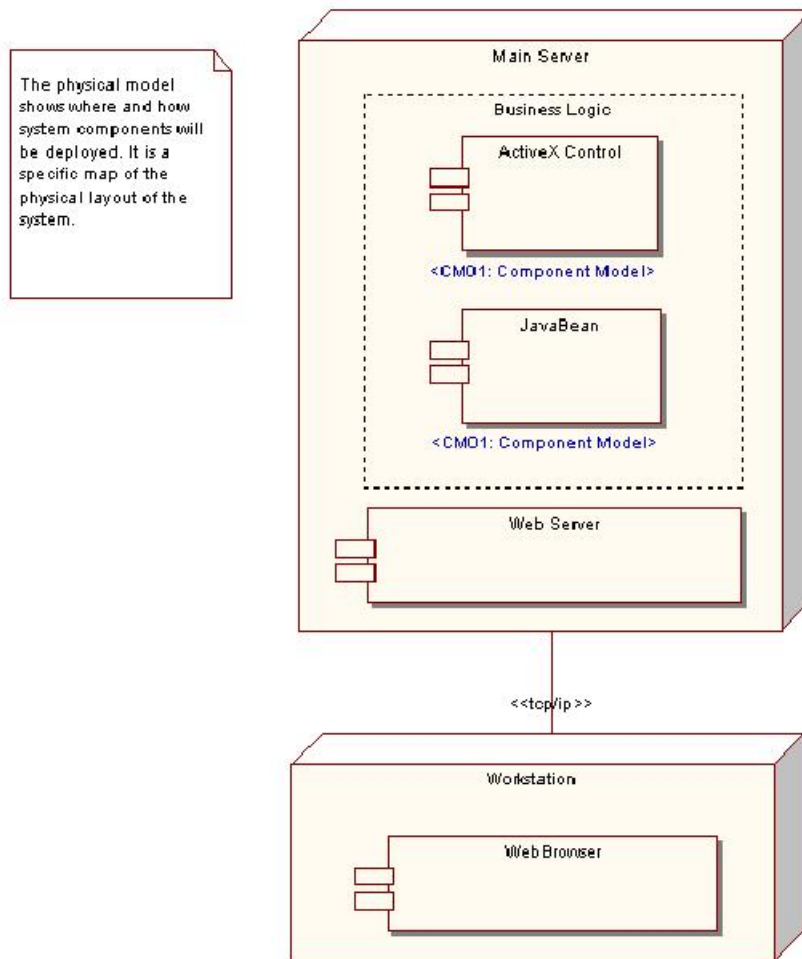
8. Diagram Komponen (*Component Diagram*) : Bersifat statis. Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem/perangkat lunak pada komponen-komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan kedalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.



Gambar II.6 : Diagram Komponen

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

9. Diagram *Deployment (Deployment Diagram)* : Bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul-simpul beserta komponen-komponen.



Gambar II.7 : Diagram *Deployment*

(Sumber : Prabowo Pudjo Widodo & Herlawati ; Menggunakan UML)

Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*), (prabowo Pudjo Widodo & Herlawati ; 2011: 6).

II.6. Database

Menurut (Abdul Kadir ; 2009:10) terdapat butir-butir yang bisa diambil dari definisi database yaitu :

1. Sebuah database menghimpun data yang terkait atau data yang saling berhubungan.
2. Kumpulan data tersebut terorganisasi.
3. Bisa melibatkan lebih dari satu organisasi.

Butir ketiga mengisyaratkan bahwa sebuah database bisa digunakan oleh lebih dari satu organisasi. Sebuah database mencatat berbagai data yang diperlukan oleh suatu organisasi, rekaman-rekaman data tersebut pada suatu saat akan diambil dan melalui suatu pemrosesan akan diperoleh suatu informasi yang dikehendaki oleh pengguna.

Database berbeda dengan sistem pemrosesan berbasis berkas. Sistem pemrosesan berbasis berkas adalah suatu model penyimpanan data yang mendasarkan pada penyimpanan data dalam bentuk *file* (berkas), yang memiliki banyak kelemahan dibanding database.sistem seperti ini banyak dipakai dimasa lalu, salah satu perangkat lunak yang biasa dipakai untuk mengimplementasikannya adalah COBOL.

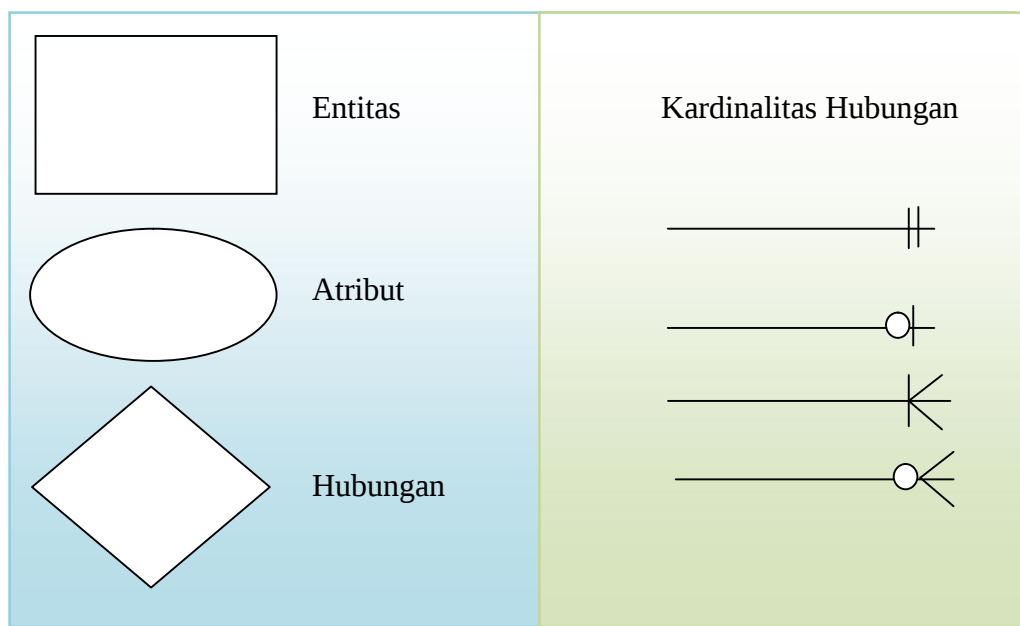
II.6.1. Pemodelan Data

Pada Perancangan konseptual diperlukan suatu pendekatan yang digunakan untuk menggambarkan hubungan antar data. Hubungan tersebut dapat dinyatakan dalam bentuk E-R, karena model E-R adalah dasar penting dalam merancang database maka akan dijelaskan tentang gambaran tentang model E-R, penjelasan mengenai komponen-komponen yang menyusun model E-R, hingga cara penyusunan model E-R.

II.6.2. Model E-R

Model E-R adalah suatu model yang digunakan untuk menggambarkan data dalam bentuk entitas, atribut dan hubungan antar entitas. Huruf E sendiri menyatakan entitas dan R menyatakan hubungan (dari kata *Relationship*). Model ini dinyatakan dalam bentuk diagram, itulah sebabnya model E-R sering disebut sebagai diagram E-R.

Model E-R melibatkan sejumlah notasi, beberapa notasi dasar dalam model E-R ditunjukkan pada gambar II.9, notasi-notasi tersebut diberikan hanya untuk memberikan suatu pengetahuan dasar.



Gambar II.8 : Sejumlah notasi pada model E-R

(Sumber : Abdul Kadir ; Dasar Perancangan & Implementasi Database Relasional)

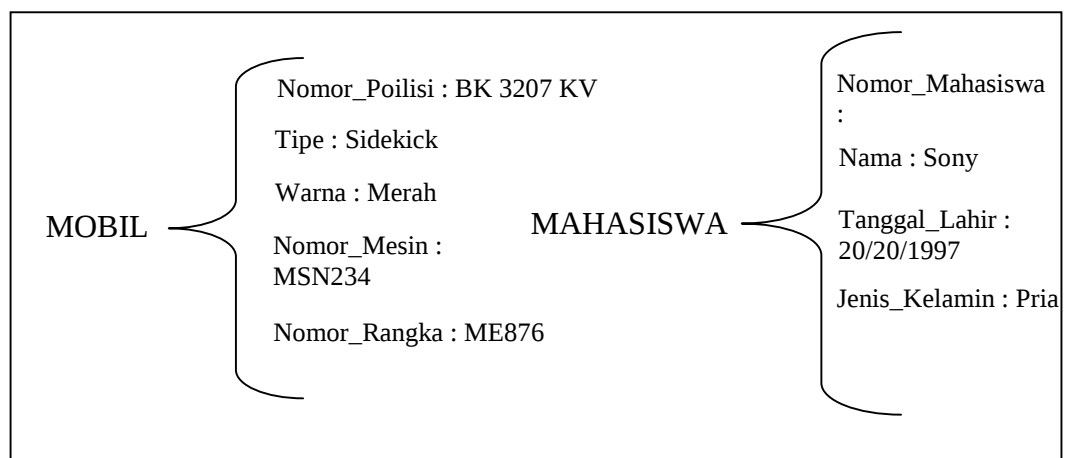
1. Entitas

Yang dimaksud dengan entitas adalah sesuatu dalam dunia nyata yang keberadaannya tidak bergantung pada yang lain. Sebagai contoh, setiap

pegawai dalam sebuah organisasi adalah sebuah entitas. Entitas dapat berupa sesuat yang nyata ataupun abstrak (berupa suatu konsep). Secara lebih rinci dijelaskan bahwa entitas dapat berupa seseorang, sebuah tempat, sebuah objek, sebuah kejadian atau suatu konsep.

2. Atribut

Setiap entitas dinyatakan dalam sejumlah atribut. Atribut adalah properti atau karakteristik yang terdapat pada setiap entitas. Sebagai contoh, pada gambar II.10, terdapat entitas MOBIL yang mengandung atribut Nomor_Polisi, Tipe, Warna, Nomor_Rangka dan Nomor_Mesin. Selain itu terdapat entitas MAHASISWA yang mengandung atribut Nomor_Mahasiswa, Nama, Tanggal_Lahir, dan Jenis_Kelamin.



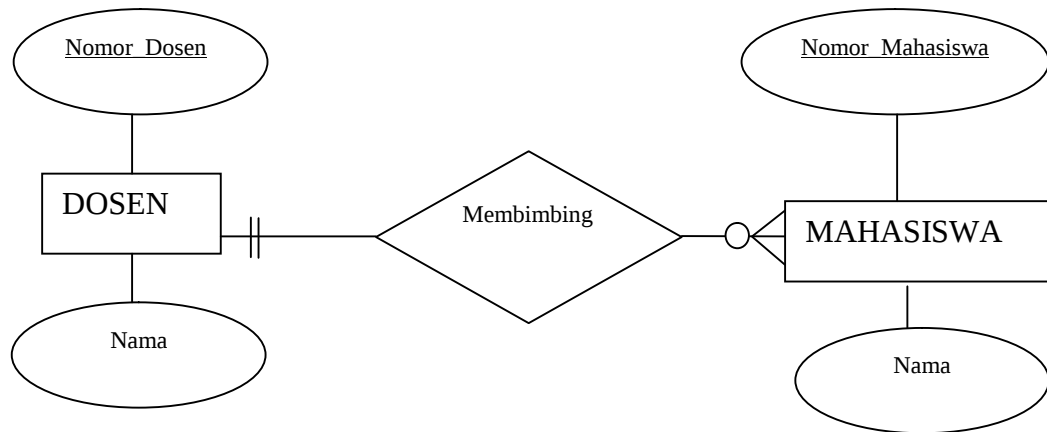
Gambar II.9 : Contoh Entitas dan Atribut

(Sumber : Abdul Kadir ; Dasar Perancangan & Implementasi Database Relasional)

3. Hubungan (*Relationship*)

Hubungan (*Relationship*) menyatakan ketertarikan antara beberapa tipe entitas. Sebagai contoh , tipe entitas MAHASISWA dan DOSEN mempunyai

hubungan yang mencerminkan bahwa seorang mahasiswa memiliki dosen pembimbing akademis. Gambar II.10 menunjukkan hubungan tersebut.



Gambar II.10 : Contoh Hubungan antara tipe entitas

(Sumber : Abdul Kadir ; Dasar Perancangan & Implementasi Database Relasional)

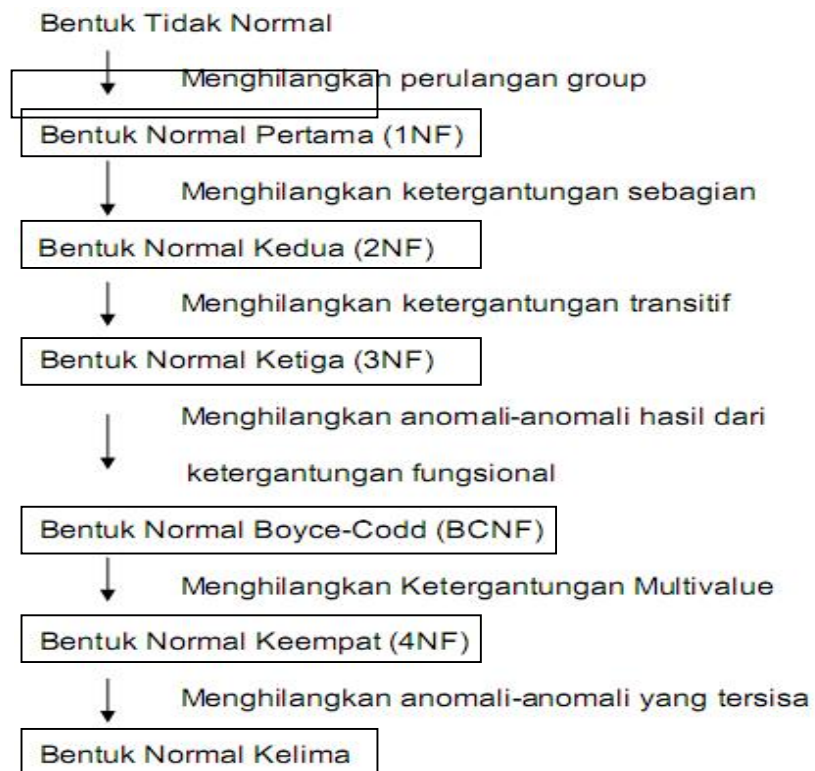
II.6.3. Normalisasi

Normalisasi adalah proses yang digunakan untuk menentukan pengelompokan atribut-atribut dalam sebuah relasi sehingga diperoleh relasi yang berstruktur baik. Dalam hal ini yang dimaksud dengan relasi yang berstruktur baik adalah relasi yang memenuhi dua kondisi berikut

1. Mengandung redundansi sedikit mungkin, dan
2. Memungkinkan baris-baris dalam relasi disisipkan, dimodifikasi dan dihapus tanpa menimbulkan kesalahan atau ketidakkonsistenan.

Normalisasi sendiri dilakukan melalui sejumlah langkah. Setiap langkah berhubungan dengan bentuk normal (*normal form*) tertentu. Dalam hal ini yang disebut bentuk normal adalah suatu keadaan relasi yang dihasilkan oleh penerapan aturan-aturan sederhana yang berhubungan dengan dependensi fungsional

terhadap relasi tersebut. Gambar II.11 berikut ini akan memperlihatkan hubungan keenam bentuk normal tersebut.



Gambar II.11 : Langkah-langkah dalam normalisasi

(Sumber : Abdul Kadir ; Dasar Perancangan & Implementasi Database Relasional)