

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Informasi

Sistem informasi adalah suatu sistem didalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi organisasi yang bersifat manajerial dengan kegiatan strategis dari suatu organisasi untuk dapat menyediakan laporan-laporan yang diperlukan oleh pihak luar tertentu. Sistem informasi bukan merupakan hal yang baru. Yang baru adalah komputerisasinya. Sebelum ada computer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi telah ada (Tata Sutabri ; 2012 : 22).

II.2. Sistem Informasi Akuntansi

Tujuan dasar suatu sistem tergantung pada jenis sistem itu sendiri. Sebagai contoh, sistem peredaran darah manusia merupakan sistem biologi yang memiliki tujuan untuk mengedarkan darah yang mengandung oksigen dan sari makanan ke seluruh tubuh. Sedangkan sistem buatan manusia seperti sistem yang terdapat di sekolah, organisasi bisnis, atau instansi pemerintah juga mempunyai tujuan yang berbeda-beda. Organisasi bisnis biasanya memiliki tujuan yang lebih jelas (Anastasia Diana dan Lilis Setiawati ; 2011 : 2).

SIA (Sistem Informasi Akuntansi) merupakan kumpulan sumber daya, seperti manusia dan peralatan, yang dirancang untuk mengubah data keuangan

dan data lainnya ke dalam informasi. Informasi tersebut dikomunikasikan kepada para pembuat keputusan. Sistem informasi akuntansi melakukan hal tersebut dengan sistem manual atau melalui sistem terkomputerisasi. (George H. Bodnar;2007:3).

Analog dengan definisi sebelumnya, SIA (Sistem Informasi Akuntansi) adalah sistem berbasis computer yang dirancang untuk mentransformasi data akuntansi menjadi informasi. (George H. Bodnar; 2007: 8).

II. 3. Penyusutan Inventaris

Penyusutan didefinisikan dalam pernyataan standar akuntansi pemerintahan sebagai penyesuaian nilai sehubungan dengan penurunan kapasitas dan manfaat dari suatu asset. Penyusutan asset tetap bukan merupakan metode alokasi biaya untuk periode yang menerima manfaat asset tetap tersebut sebagaimana diberlakukan disektor komersial. Penyesuaian nilai ini lebih merupakan upaya untuk menunjukkan pengurangan nilai karena pengkonsumsian potensi manfaat asset oleh karena pemakaian dan atau pengurangan nilai karena keusangan dan lain-lain (Binsar H. Simanjuntak, dkk ; 2008 : 1).

II. 4. Metode Garis Lurus

Dengan metode ini penyusutan tahunan dapat ditentukan dengan dua cara yaitu :

1. (Cost-nilai residu) : umur

Misalkan sebuah peralatan yang diperoleh tahun 2005 senilai Rp. 16.000.000,00 dan masa manfaat ditentukan 5 tahun dengan nilai sisa Rp. 1.000.000,00 besar penyusutan tahun 2006 dapat dihitung sebagai berikut

$$(16.000.000,00 - 1.000.000,00) / 5 = \text{Rp.} 3.000.000,00$$

2. Ditentukan % penyusutan, kemudian penyusutan tahunan diperoleh dengan cara mengalikan % tersebut dengan cost yang disusutkan sebagai berikut :

- a. Persentase penyusutan tahunan = $100\% : \text{umur}$, jadi = $100\% : 5 = 20\%$
- b. Dihitung penyusutan = $20\% \times (16.000.000,00 - 1.000.000,00) = \text{Rp.} 3.000.000,00$ (Binsar H. Simanjuntak, dkk ; 2008 : 4)

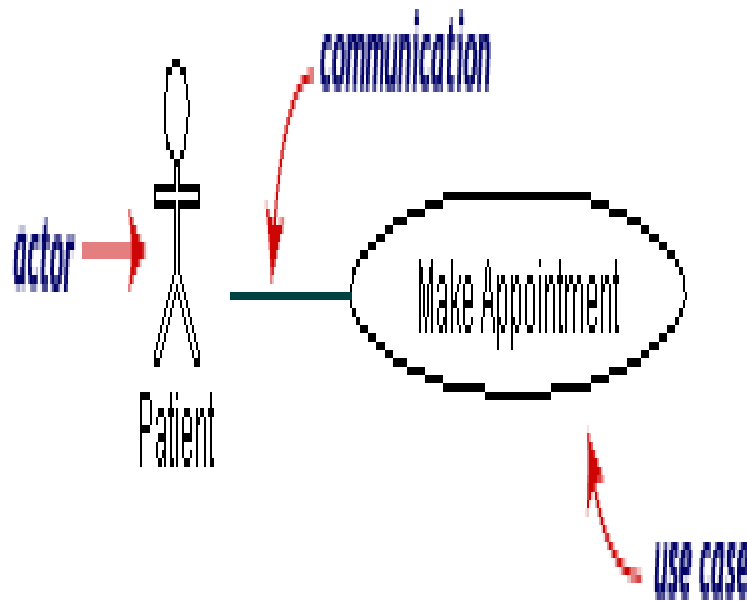
II.5. Unified Modelling Language

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi Setiap sistem yang kompleks seharusnya bisa dipandang dari sudut yang berbeda – beda sehingga bisa mendapatkan pemahaman secara menyeluruh . Untuk upaya tersebut UML menyediakan 9 jenis diagram yang dapat dikelompokkan berdasarkan sifatnya statis atau dinamis. Ke 9 diagram dalam UML itu adalah (Prastuti Sulistyorini 2009).

II.5.1. Use Case Diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas). Diagram ini terutama sangat

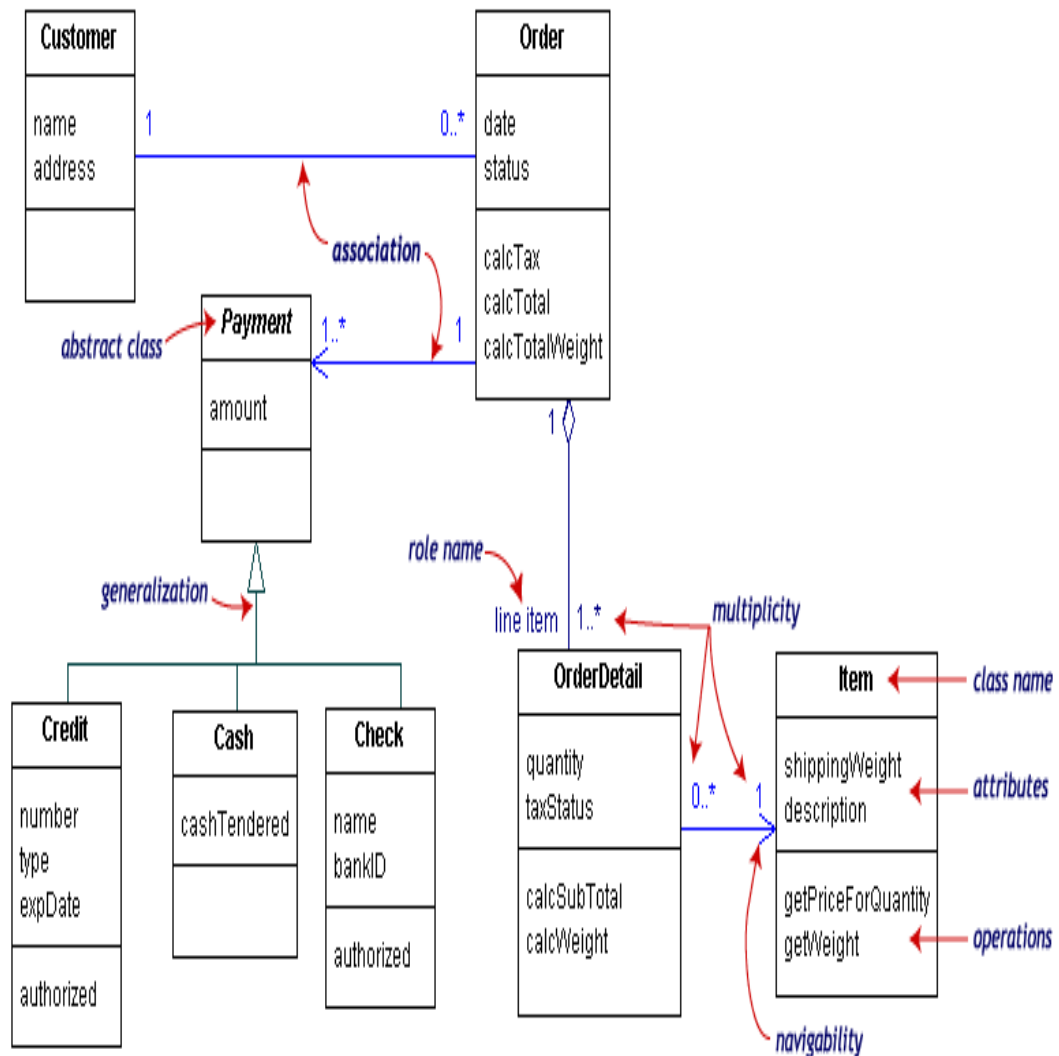
penting untuk mengorganisasi dan memodelkan perilaku dari suatu sistem yang dibutuhkan serta diharapkan pengguna (Prastuti Sulistyorini : 2009).



Gambar II.1. Contoh Use Case Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.2. Class Diagram

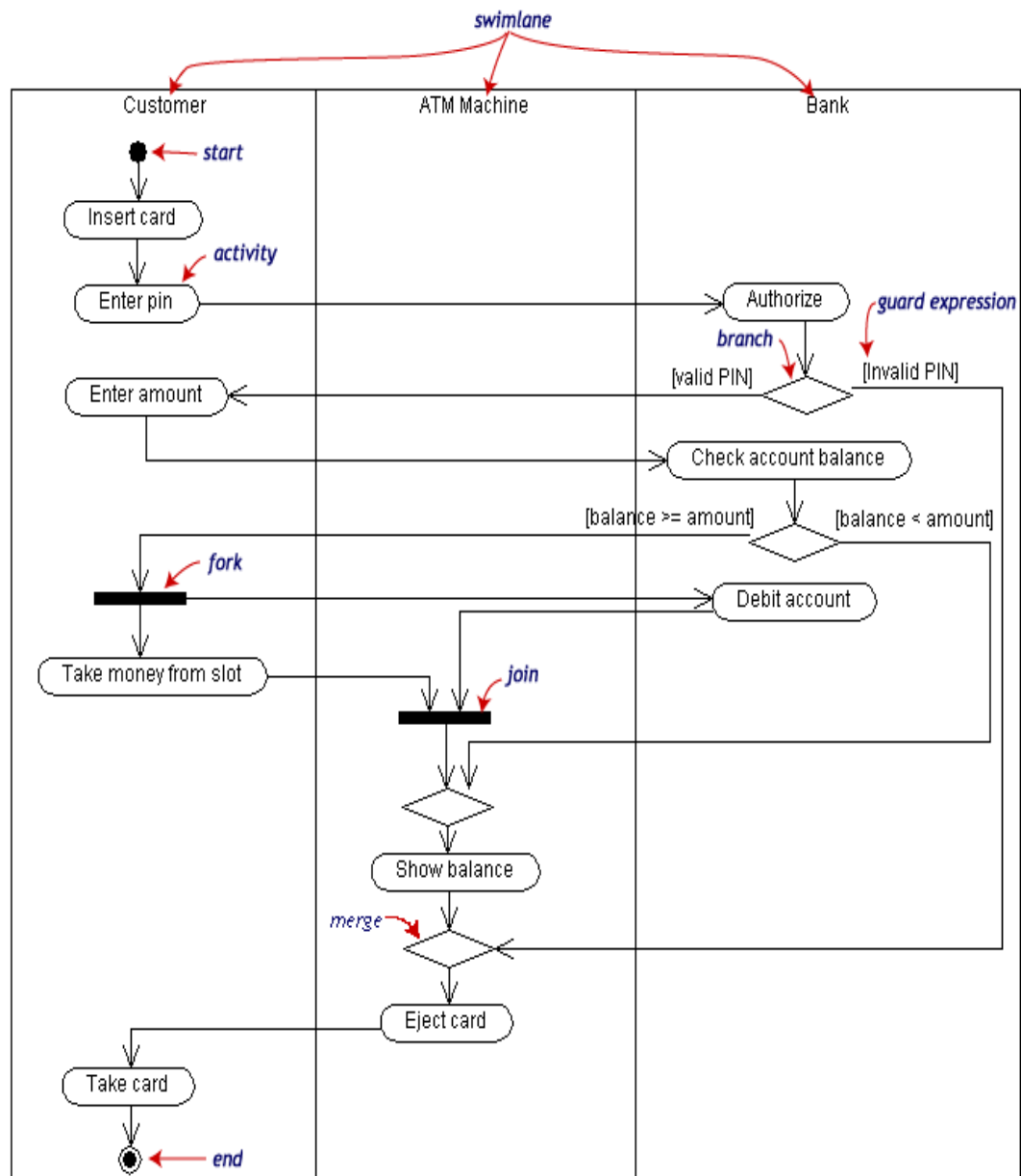
Diagram kelas bersifat statis. Diagram ini memperlihatkan himpunan kelas-kelas, antarmuka antarmuka, kolaborasi-kolaborasi serta relasi (Prastuti Sulistyorini : 2009).



Gambar II.2. Contoh Class Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.3. Activity Diagram

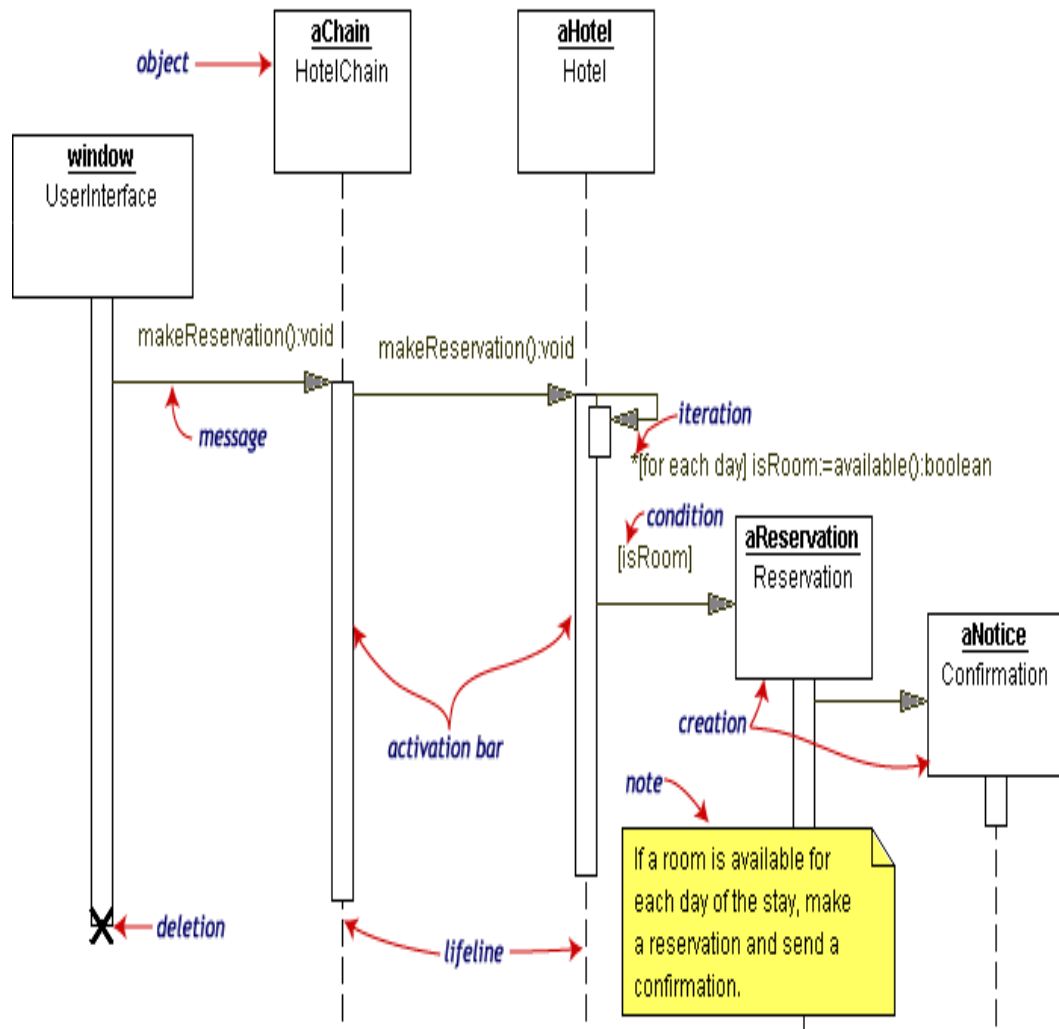
Diagram ini bersifat dinamis. Diagram ini adalah tipe khusus dari diagram state yang memperlihatkan aliran dari suatu aktifitas ke aktifitas lainnya dari suatu sistem. Diagram ini terutama penting dalam pemodelan fungsi – fungsi dalam suatu sistem dan memberi tekanan pada aliran kendali antar objek (Prastuti Sulistyorini : 2009).



Gambar II.3. Contoh Activity Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.4. Sequence Diagram

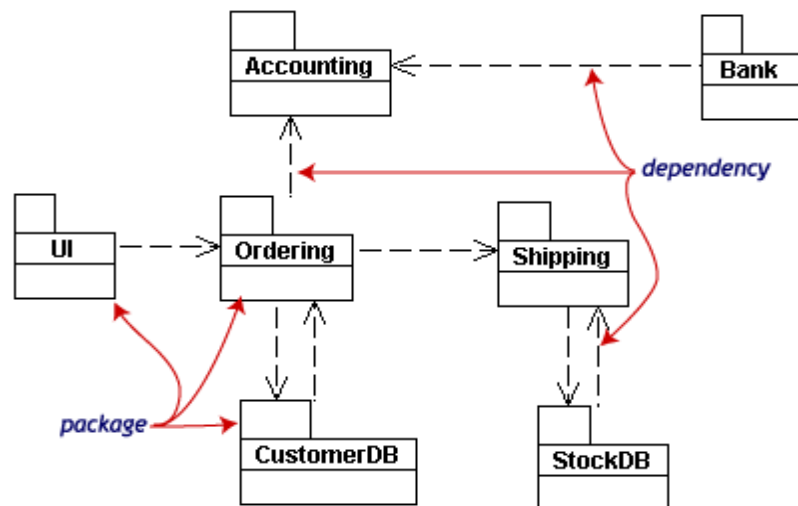
Diagram ini bersifat dinamis. Diagram sequence merupakan diagram interaksi yang menekankan pada pengiriman pesan (message) dalam suatu waktu tertentu (Prastuti Sulistyorini : 2009).



Gambar II.4. Contoh Sequence Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.5. Diagram Package Object

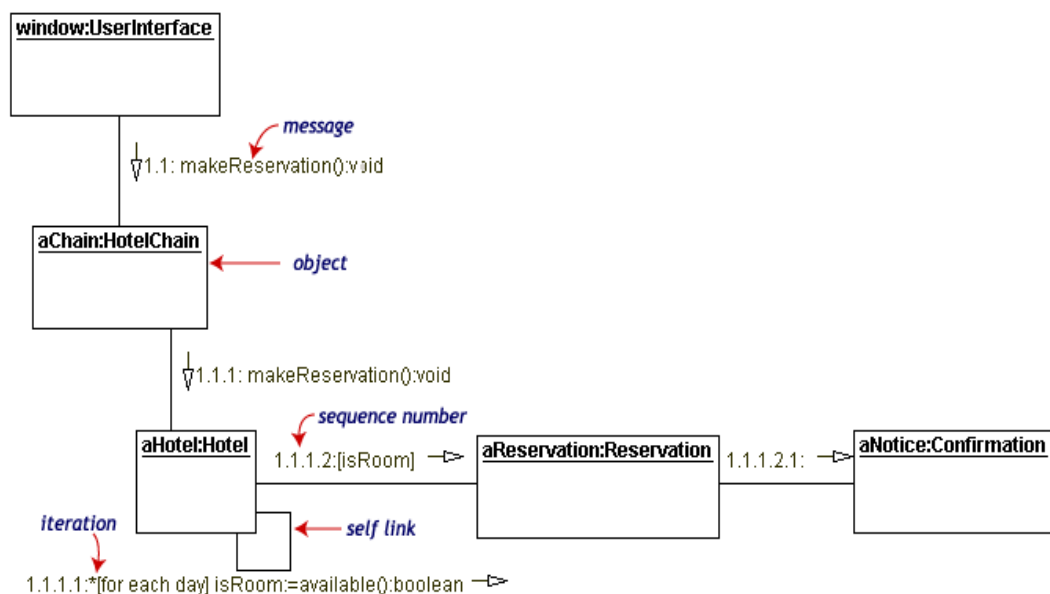
Diagram objek bersifat statis. Diagram ini memperlihatkan objek-objek serta relasi antar objek. Diagram objek memperlihatkan instansiasi statis dari segala sesuatu yang dijumpai pada diagram kelas (Prastuti Sulistyorini : 2009).



Gambar II.5. Contoh Package Object Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.6. Collaboration Diagram

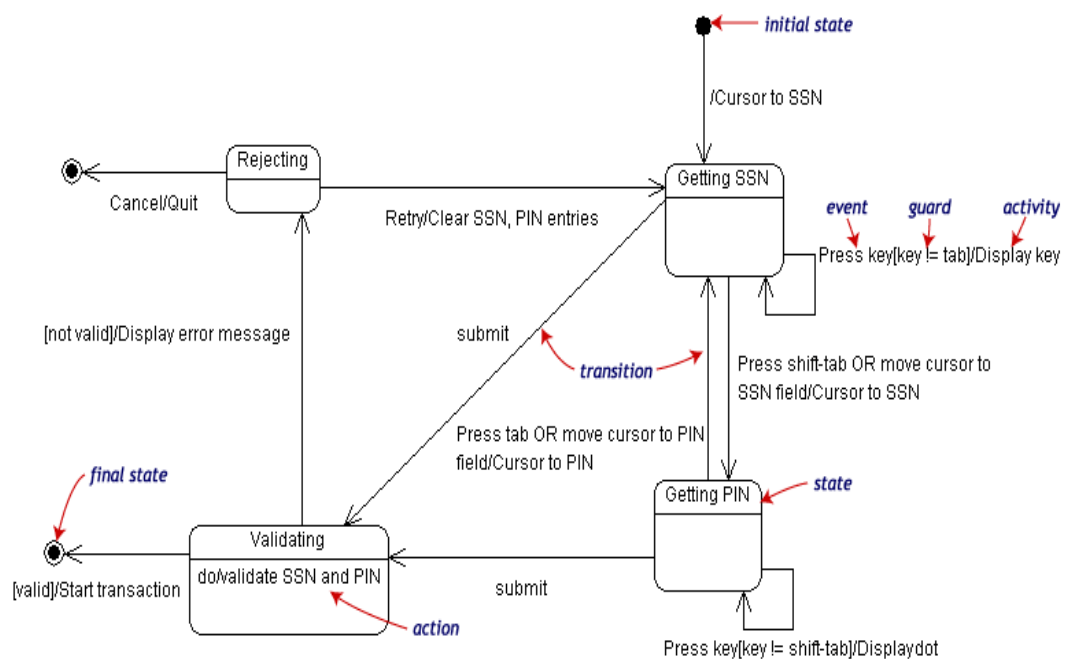
Diagram ini bersifat dinamis. Diagram kolaborasi adalah diagram interaksi yang menekankan organisasi struktural dari objek – objek yang menerima serta mengirim pesan (message) (Prastuti Sulistyorini : 2009).



Gambar II.6. Contoh Collaboration Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.7. Statechart Diagram

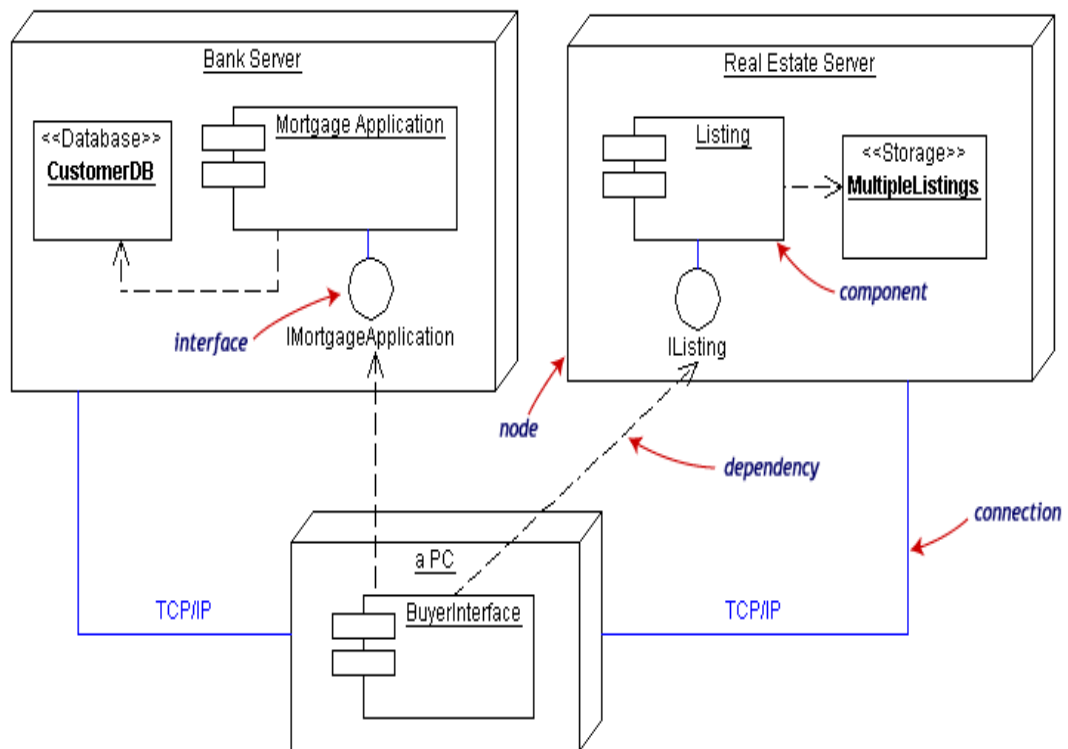
Diagram ini bersifat dinamis. Diagram ini memperlihatkan state – state pada sistem, memuat state, transisi, event, serta aktifitas. Diagram ini terutama penting untuk memperlihatkan sifat dinamis dari antarmuka, kelas, kolaborasi dan terutama penting pada pemodelan sistem – sistem yang reaktif (Prastuti Sulistyorini : 2009).



Gambar II.7. Contoh Statechart Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.5.8. Component dan Deployment Diagram

Diagram ini bersifat statis. Diagram ini memperlihatkan organisasi serta ketergantungan pada komponen – komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal ditempatkan ke dalam satu atau lebih kelas, antarmuka – antarmuka serta kolaborasi – kolaborasi (Prastuti Sulistyorini : 2009).



Gambar II.8. Contoh Component dan Deployment Diagram
(Sumber : Prastuti Sulistyorini : 2009)

II.6. Microsoft Visual Basic

Visual basic merupakan salah satu bahasa pemrograman yang andal dan banyak digunakan oleh pengembang untuk membangun berbagai macam aplikasi *windows*. *Visual basic 2008* merupakan aplikasi pemrograman yang menggunakan teknologi. *NET Framework 3.5*. Teknologi. *NET Framework 3.5* merupakan komponen *windows* yang terintegrasi serta mendukung pembuatan, penggunaan aplikasi, dan halaman *web*. Teknologi *.Net Framework 3.5* mempunyai 2 komponen utama, yaitu *CLR (Common Language Runtime)* dan *Class Library*. *CLR* digunakan untuk menjalankan aplikasi yang berbasis *.NET*, sedangkan *Library* adalah kelas pustaka atau perintah yang digunakan untuk membangun aplikasi (Wahana Komputer ; 2010 : 2)

II.7. Microsoft SQL Server

Bahasa query merupakan bahasa khusus yang digunakan untuk melakukan manipulasi dan menanyakan pertanyaan (query) yang berhubungan dengan bahasa pemrograman, dimana bahasa query tidak memiliki kemampuan untuk menyelesaikan banyak masalah seperti bahasa pemrograman pada umumnya. Dalam pemrograman basis data, salah satu bahasa yang harus kita kuasai adalah SQL. SQL merupakan bahasa komputer standar yang digunakan untuk berkomunikasi dengan sistem manajemen basis data relasional (RDBMS) (Ema Utami dan Anggi Dwi Hartanto ; 2012 : 63)

II.8. Entity Relationship Diagram

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* terdiri 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 digunakan simbol-simbol grafis tertentu (Edhy Sutanta ; 2011 : 91)

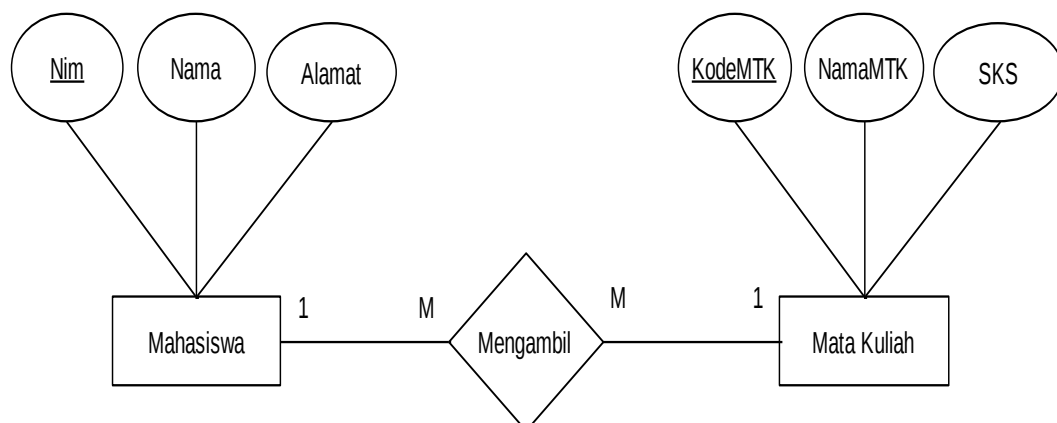
Struktur *logis* (skema *database* dapat ditunjukkan secara *grafis* dengan ER yang dibentuk dari komponen-komponen berikut ini (Rosa A.S dan M. Shalahuddin ; 2011 : 60)

Tabel II. 1. Komponen ERD

Simbol	Keterangan
Entitas	Persegi panjang mewakili kumpulan entitas.
Attribute	Elips mewakili attribute
	Belah ketupat mewakili relasi
	Garis menghubungkan atribut dengan kumpulan entitas dengan relasi.

(Sumber : Rosa A.S dan M. Shalahuddin ; 2011)

Sebagai ilustrasi, bayangan anda mengambil bagian sistem basis data universitas yang terdiri dari mahasiswa dan mata kuliah. Gambar II.9. menunjukkan ER Diagram dari contoh. Diagram menunjukkan bahwa ada dua kumpulan entitas, yaitu mahasiswa dan mata kuliah, dan bahwa relasi mengambil mahasiswa dan mata kuliah.

**Gambar II.9. Diagram ER**

(Sumber : Rosa A.S dan M. Shalahuddin ; 2011)

II.9. Kamus Data

Kamus data adalah suatu file yang terpisah yang menyimpan informasi seperti :

1. Nama setiap *item*/jenis/kolom data
2. Struktur data untuk tiap *item*
3. Program yang menggunakan tiap *item*
4. Tingkat keamanan untuk setiap *item* (Zulkifli Amsyah : 2008 : 17)

Sebelum memperoleh definisi 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 (Edhy Sutanta ; 2011 : 25)

II.10. Normalisasi

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 (Edhy Sutanta ; 2011 : 174).

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 Depedency*) 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 (Yuniar Supardi ; 2008 : 11)

Tabel II.2. Normalisasi

Pelanggan

*Kode Pelanggan	Nama_Pelanggan	Alt_Pelanggan	Tlp_Pelanggan

Transj

*No_Faktur	Qty_Jual	Harga_Jual	**Kode_Barang

Faktur

*No_Faktur	Tgl_Jual	Total_Jual	Pembuat	Penerima	**Kode_Pelanggan	**No_SO

Barang

*No_Barang	Nama_Barang	Harga_Beli	Harga_Jual	Quantity

(Sumber : Yuniar Supardi ; 2008)