BAB II

TINJAUAN PUSTAKA

II.1. Analisis Sistem

Analisis Sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa bagus bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan mereka. Analisis sistem merupakan tahapan yang paling awal dari pengembangan sistem yang menjadi fondasi menentukan keberhasilan sistem informasi yang dihasilkan nantinya. (Hanif Al Fatta ; 2007 : 44)

II.2. Knowledge Management

Sebagai langkah awal untuk memberikan pemahaman dalam manajemen pengetahuan (*knowledge management*), diperlukan peninjauan berbagai literatur. Para pakar dan ahli dalam mengkonsepsikan formulasi definisi satu sama lain berbeda-beda, hal tersebut karena perbedaan sudut pandang dari masing-masing pakar dan ahli.

Knowledge Management adalah usaha untuk meningkatkan pengetahuan yang berguna dalam organisasi, diantaranya membiasakan budaya berkomunikasi antar personil, memberikan kesempatan untuk belajar, dan menggalakan saling berbagi *knowledge*. Dimana usaha ini akan menciptakan dan mempertahankan peningkatan nilai dari inti kompetensi bisnis dengan memanfaatkan teknologi informasi yang ada Hal ini disarikan dari pendapat McInerney sebagai berikut :

"*Knowledge Management (KM) is an effort to increase useful knowledge within the organization. Ways to do this include encouraging communication, offering opportunities to learn, and promoting the sharing of appropriate knowledge artifacts*.". (Winda & Ken ; 2014 : 681-683)

II.2.1. Konversi Knowledge

Nonaka dan Takeuchi mengemukakan bahwa alasan *fundamental* mengapa perusahaan Jepang sukses, kerena ketrampilan dan pengalaman mereka terdapat pada penciptaan *knowledge* organisasi. Penciptaan *knowledge* dicapai melalui pengenalan hubungan sinergik antara *tacit knowledge* dan *explicit knowledge*. Ikujiro Nonaka dan Hirotaka Takeuchi pada tahun 1991 dan 1995, membedakan antara *tacit knowledge* dan *explicit knowledge*, dan membagi model *konversi knowledge* menjadi 4 cara sebagai berikut :

1. *Tacit knowledge* ke *Explicit knowledge* disebut proses *Externalization.*
2. *Tacit knowledge* ke *Tacit knowledge* disebut proses *Socialization*.
3. *Explicit knowledge* ke *Explicit knowledge* disebut proses *Combination*.
4. *Explicit knowledge* ke *Tacit knowledge* disebut proses *Interlization.*

	TACIT	EXPLICIT
TACIT	**(*Socialization*)** E.G, TEAM MEETINGS AND DISCUSSION	**(*Externalization*)** E.G, DIALOG WITHIN TEAM ANSWER QUESTIONS
EXPLICIT	**(*Interlization*)** E.G, LEARN FROM A REPORT	**(*Combination*)** E.G, E-MAIL A REPORT

Gambar II.1. Model *konversi Knowledge* menurut NONAKA
(*Sumber : Winda & Ken ; 2014 : 683*)

II.2.2. Konsep Manajemen Pengetahuan

Konsep dan defenisi manajemen pengetahuan, antara lain dikemukakan oleh Davidson dan Philip Voss (2002), Manajemen pengetahuan sebagai system yang memungkinkan perusahaan menyerap pengetahuan, pengalaman dan kreativitas para stafnya untuk perbaikan perusahaan. Menurut pendapat Betgerson (2003), manajemen pengetahuan merupakan suatu pendekatan yang sistematik untuk mengelola asset intelektual dan informasi lain sehingga memberikan keunggulan bersaing bagi perusahaan. (Prof. DR. H. Ismail Nawawi ; 2012 : 2)

Dalam memperkaya pemahaman, Tannebaum (1998) memberikan definisi dengan berbagai formulasi untuk memberikan pemahaman terhadap manajemen pengetahuan sebagai berikut :

1. Manajemen pengetahuan mencakup pengumpulan, penyusunan, penyimpanan, dan pengaksesan informasi untuk membangun pengetahuan, pemanfaatan dengan tepat teknologi informasi, seperti computer yang dapat mendukung manajemen pengetahuan, namun teknologi informasi tersebut bukanlah manajemen pengetahuan.
2. Manajemen pengetahuan mencakup berbagai pengetahuan (*sharing knowledge*). Tanpa berbagi pengetahuan, upaya manajemen pengetahuan akan gagal culture perusahaan, dinamika dan praktik, seperti system penggajian dapat mempengaruhi berbagai pengetahuan. Kultur dan aspek social dari manajemen pengetahuan merupakan tantangan yang signifikan.
3. Manajemen pengetahuan terkait dengan pengetahuan orang. Pada suatu saat, organisasi membutuhkan orang yang kompeten untuk memahami dan

memanfaatkan informasi dengan efektif. Organisasi terkait dengan individu untuk melakukan inovasi dan memberi petunjuk kepada organisasi. Organisasi juga terkait dengan persoalan keahlian yang menyediakan *input* untuk menerapkan manajemen pengetahuan. Oleh karena itu, organisasi mesti mempertimbangkan bagaimana menarik, mengembangkan, dan mempertahankan pengetahuan anggota sebagai bagian dari *domain* manajemen pengetahuan.

4. Manajemen pengetahuan terkait dengan peningkatan efektivitas organisasi. Kita berkonsentrasi dengan manajemen pengetahuan karena dipercaya bahwa manajemen pengetahuan dapat memberikan kontribusi kepada vitalitas dan kesuksesan perusahaan. Upaya untuk mengukur modal intelektual dan untuk menilai efektivitas manajemen pengetahuan harus dapat membantu kita memahami secara luas pengelolaan pengetahuan yang telah dilakukan.

Selain mengusulkan satu consensus mengenai pengertian manajemen pengetahuan, Tannebaum juga memberikan penjelasan mengenai karakteristik berbagai aktivitas manajemen pengetahuan. Manajemen pengetahuan, menurut Tannebaum, paling tidak terdiri atas berikut ini :

1. Pengembangan *database* organisasi mengenai pelanggan, masalah yang bersifat umum dan serta pemecahannya.
2. Mengenali para ahli internal, memperjelas apa yang mereka ketahui, dan mengembangkan kamus yang menjelaskan sumber daya internal kunci dan mengenali bagaimana menemukannya.

3. Mendapatkan dan menangkap pengetahuan dari para ahli tersebut untuk disebar ke yang lain.
4. Mendesain struktur pengetahuan yang membantu mengelola informasi dalam suatu cara yang dapat diakses dan siap untuk diaplikasikan.
5. Menciptakan forum bagi orang-orang yang ada di dalam perusahaan untuk berbagi pengalaman dan ide, baik dalam bentuk tatap muka, berkomunikasi melalui *internet*, *website*, *chating room*, *e-mail*, dan lain-lain.
6. Memanfaatkan *groupware* sehingga memungkinkan berbagai macam orang di lokasi yang berbeda dapat berkomunikasi untuk menyelesaikan masalah secara bersama-sama dan mencatat informasi di dalam suatu *domain* pengetahuan yang telah dipilih.
7. Bertindak untuk mengenali, mempertahankan talenta orang-orang yang memiliki pengetahuan yang diperlukan di dalam bidang kegiatan utama bisnis.
8. Mendesain pelatihan dan aktifitas pengembangan lainnya untuk menilai dan mengembangkan pengetahuan internal.
9. Menerapkan praktik penghargaan pengakuan dan promosi yang mendorong berlangsungnya kegiatan berbagi informasi antar anggota maupun antar unit dalam organisasi.
10. Membantu pekerjaan serta menyediakan alat-alat yang mendukung kinerja sehingga memungkinkan setiap orang menilai dan menerapkan pengetahuan apabila diperlukan.
11. Memaknai *database* pelanggan, produk, transaksi, atau hasil dengan mengenali kecenderungan dan menggali informasi sebanyak mungkin.

12. Mengukur modal intelektual di dalam upaya mengelola pengetahuan yang lebih baik.
13. Menangkap dan menganalisis informasi yang terkait dengan perhatian pelanggan, pilihan-pilihan, dan kebutuhan dari lapangan, *front line* atau personil bagian pelayanan didorong untuk mampu memahami dengan lebih baik terhadap keenderungan pelanggan.

Di pihak lain, ada yang mengkonsepsikan dengan formulasi definisi dikaitkan dengan komponen krisis bahwa manajemen pengetahuan (*knowledge management*) yang sukses tidak hanya karena komputerisasi yang impresif, tetapi sebaiknya ditinjau dari ketiga komponen yang kritis berikut :

1. Alur *knowledge* yang benar dan sumber yang dilimpahkan ke organisasi / institusi.
2. Teknologi tepat yang disimpan dan dapat mengomunikasikan *knowledge* tersebut.
3. Budaya tempat kerja yang benar, sehingga karyawan termotivasi untuk memanfaatkan *knowledge*.

Oleh karena itu, manajemen pengetahuan (*knowledge management*) akan sukses apabila terjadi interaksi di antara komponennya dan tidak terjadi tumpang tindih (*overlap*) dari ketiga komponen tadi. Meskipun demikian, *knowledge management* memberikan kesempatan pada organisasi tersebut untuk :

1. Menangkap dan menganalisa informasi organisasi dan diaplikasikan secara strategis dalam bentuk *warehousing* dan *dataming*, system pendukung

keputusan (*Decision System Support*/DSS), serta system informasi eksekutif (EIS).

2. Menciptakan proses untuk akses informasi ke seluruh dunia melalui intranet, *groupware*, dan sistem pendukung keputusan kelompok (Group DSS) agar karyawa mendapat informasi secara tepat, informative dan inovatif, menjadikan kekuatan pendorong dari *knowledge* yang terakumulasi dari pengalaman masa lalu seluruh organisasi
3. Membangun dan menyelesaikan proyek dengan meningkatkan kecepatan, ketangkasan, dan keselamatan.

Masih banyak organisasi yang memusatkan usahanya pada pada satu area saja, yaitu mengaplikasikan manajemen pengetahuan melalui teknologi saja. Oleh karena itu, sebaiknya dilakukan melalui pendekatan stok dan alur pengetahuan yang merupakan karakteristik dari manajemen pengetahuan. Stok dan alur pengetahuan tersebut adalah sebagai berikut :

1. Stok pengetahan (*knowledge*) adalah sesuatu yang telah diketahui yang dapat berupa *database* atau perpustakaan, organisasi/institusi, tersebar diseluruh organisasi/institusi dalam berbagai kantor, *filling cabinets*, rak buku (*bookshelves*), dan sebagainya atau di pikiran karyawan.
2. Alur *knowledge*, agar *knowledge* dapat bermanfaat, agar dapat menjamin, bahwa *knowledge* yang ada di manapun dalam organisasi/institusi dapat tersedia di manapun apabila diperlukan, sangat penting untuk menjamin apakah *knowledge* yang ada dalam organisasi/institusi mampu untuk menyebar ke manapun dalam organisasi.

Kedua pendekatan tersebut diperlukan untuk membangun *knowledge sharing* dan *learning organization* dalam organisasi tersebut. Istilah organisasi yang selal belajar (*learning organization*) dimaksudkan sebagai kemampuan organisasi untuk belajar dari pengalaman di masa lalu (Dibell, 1995).

Sebelum organisasi dapat meningkatkan kemampuannya tersebut, harus belajar. Untuk dapat meningkatkan *learning organization*, maka organisasi tersebut harus menanggulangi 3 isu penting / kritis, yaitu :

1. Arti (menentukan visi *learning organization* itu nantinya).
2. Pengelolaan (menentukan bagaiman organisasi tersebut bekerja).
3. Ukuran (mengkaji arah dan tingkat belajar (*learning*)).

II.3. PHP

PHP pertama kali ditemukan pada 1995 oleh seorang Software Developer bernama Rasmus Lerdrof. Ide awal PHP adalah ketika itu Rasmus ingin mengetahui jumlah pengunjung yang membaca resume onlinenya. Script yang dikembangkan baru dapat melakukan 2 pekerjaan, yakni merekam informasi visitor, dan menampilkan jumlah pengunjung dari suatu website. Dan sampai sekarang kedua tugas tersebut masih tetap populer digunakan oleh dunia web saat ini. Kemudian, dari situ banyak orang dimilis mendiskusikan script buatan Rasmus Lerdrof, hingga akhirnya Rasmus mulai membuat sebuah tool/script, bernama Personal Home Page (PHP). (Loka Dwiartara ; 2010 : 4)

PHP adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasiskan kode-kode (*script*) yang digunakan untuk mengolah

suati data dan mengirimkannya kembali ke *web browser* menjadi kode HTML. (Diar Puji Oktavian ; 2010 : 31)

II.4. MySQL

MySQL adalah Database. Database sendiri merupakan suatu jalan untuk menyimpan berbagai informasi dengan membaginya berdasarkan kategori-kategori tertentu. Dimana informasi – informasi tersebut saling berkaitan, satu dengan yang lainnya. (Loka Dwiartara ; 2010 : 6)

II.5. UML *(Unified Modelling Language)*

UML (*Unified Modeling Language*) adalah bahasa pemodelan untuk sistem atau perangkat lunak yang berparadigma 'berorientasi objek'. Pemodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Adi Nugroho, 2010 : 6-7).

Tabel II.1. View dan Diagram dalam UML

Major Area	View	Diagrams	Main Concepts
Structural	Static view	Class diagram	Class, association, generalization, dependency, realization, interface.
	Use case view	Use case diagram	Use case, actor, association, extend, include, use case generalization

	Implementation view	Component diagram	Component, interface, dependency, realization
	Deployment view	Deployment diagram	Node, component, dependency, location
Dynamic	State machine	Statechart diagram	State, event, transition, action
	Activity view	Activity diagram	State, activity, completion transition, fork, join
	Interaction view	Squence diargam	Interaction, object, message, activation
		Collaboration diagram	Collaboration, interaction, collaboration role, message
Model management	Model management view	Class diagram	Package, subsystem, model
Extensibility	All	All	Constraint, stereotype, tagged values

(*Sumber : Adi Nugroho ; 2010 : 9-10*)

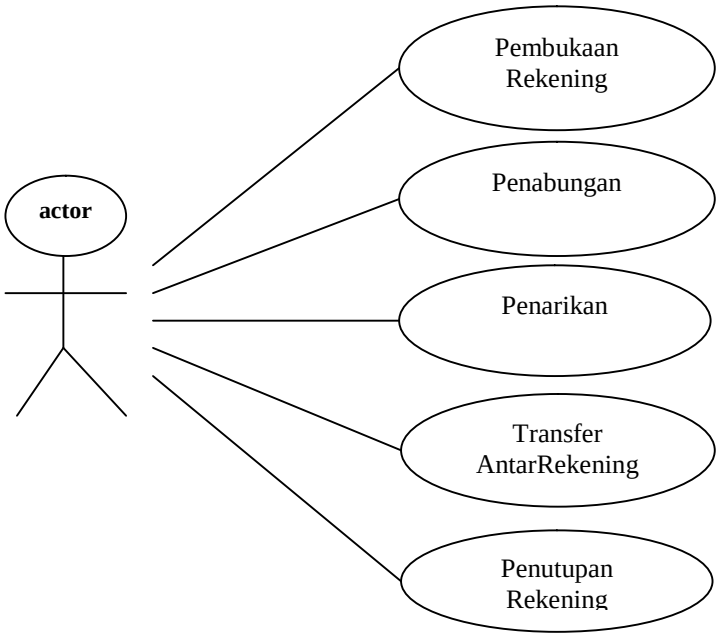
II.5.1. *Use Case Diagram*

Use Case adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. Use case bekerja dengan cara mendeskripsikan tipikal interaksi antara user (pengguna) sebuah sistem dengan sistemnya sendiri melalui sebuah cerita bagaimana sebuah sistem dipakai. Urutan langkah-langkah yang menerangkan antara pengguna dan sistem disebut scenario. Setiap scenario mendeskripsikan urutan kejadian. Setiap urutan diinisialisasikan oleh orang, sistem yang lain,

perangkat keras atau urutan waktu. Dengan demikian secar singkat bisa dikatakan use case adalah serangkain scenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

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 alam 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)

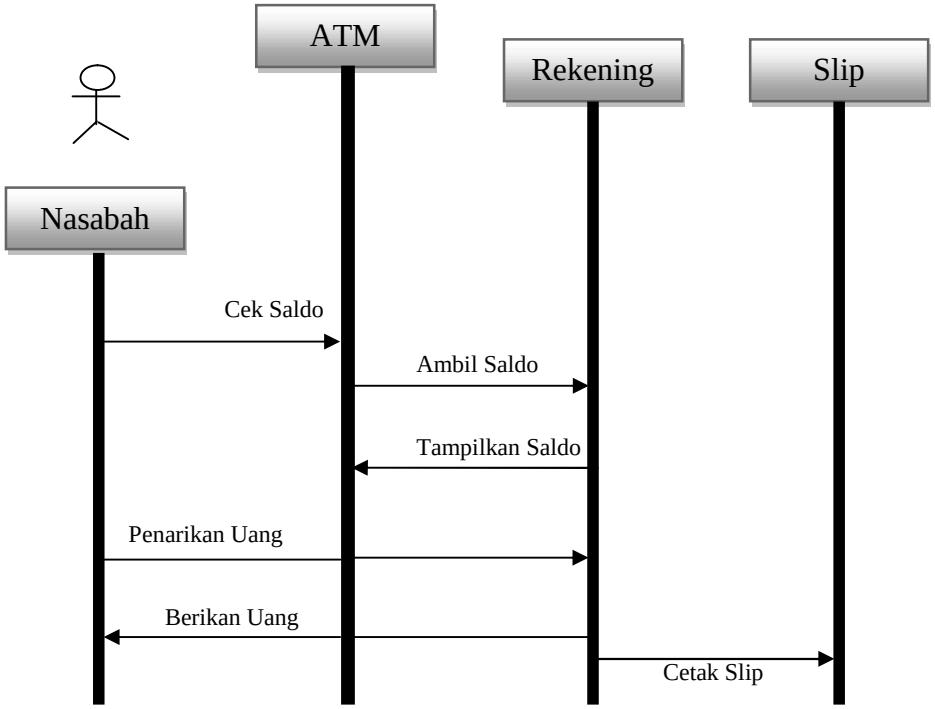


Gambar II.2. Contoh *Use Case Diagram*
(*Sumber : Adi Nugroho ; 2009 : 8*)

II.5.2. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh obyek dan message (pesan) yang diletakkan diantara obyek-obyek ini di dalam use case.

Sequence diagram juga menampilkan interaksi antar suatu kelas dengan kelas yang lainnya, bagaimana suatu *message* (pesan) dikirimkan dari suatu kelas ke kelas yang lainnya, dengan penekanan lebih pada urutan kejadian menurut waktu. Keunggulan dari *Sequence diagram* memperlihatkan dengan baik urutan interaksi yang terjadi antara suatu kelas dengan kelas lainnya, tetapi mengabaikan pengorganisasiannya. (Adi Nugroho ; 2009 : 101)



Gambar II.3. ***Sequence Diagram***
(*Sumber : Adi Nugroho ; 2009 : 102*)

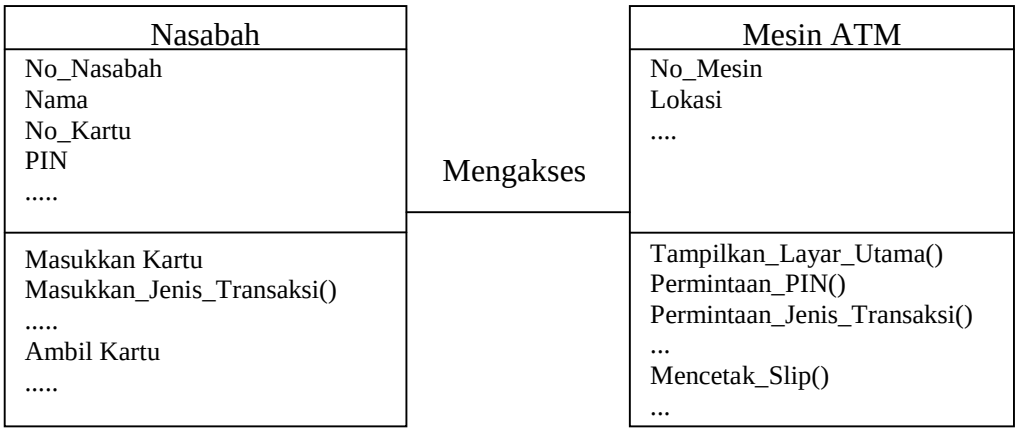
II.5.3. *Class Diagram*

Class didefenisikan 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 :

1. Memudahkan untuk mempelajari secara seksama hal-hal yang ada di dunia nyata.

2. Menyediakan suatu dasar yang kuat dalam implementasi ke dalam sistem terkomputerisasi. (Adi Nugroho ; 2009 : 17)

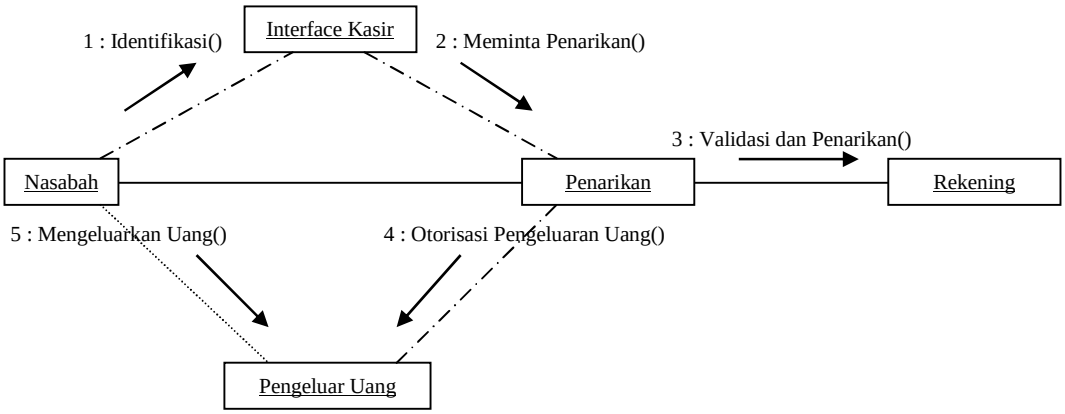


Gambar II.4. Contoh Class Diagram
(*Sumber : Adi Nugroho ; 2009 : 39*)

`

II.5.4. Collaboration Diagram

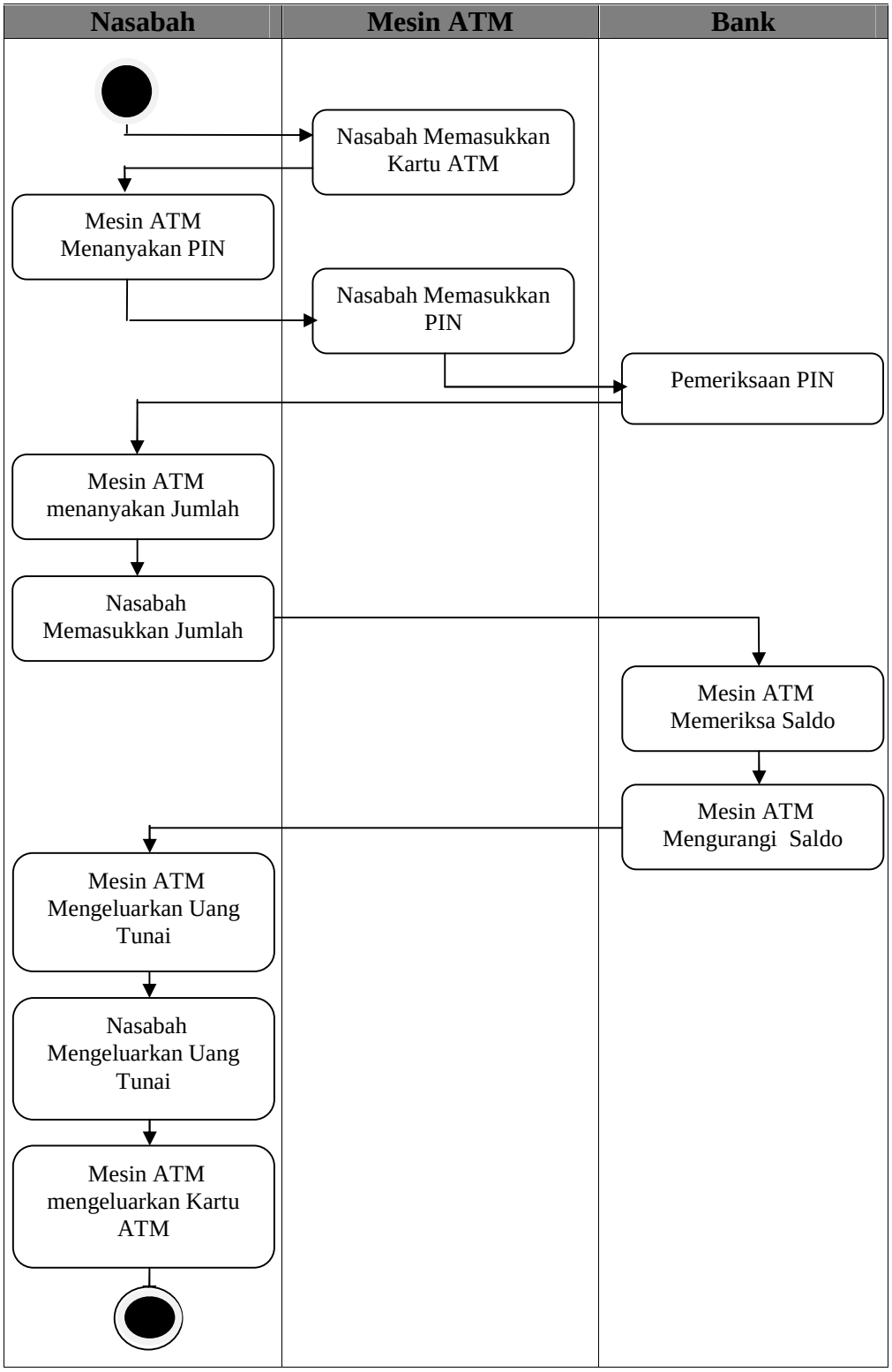
Collaboration diagram pada dasarnya merupakan diagram kelas yang memuat peran-peran pengklasifikasi dan peran-peran asosiasi, alih-alih hanya menampilkan pengklasifikasi-pengklafikasi serta asosiasi-asosiasi. Peran pengklasifikasi dan peran asosiasi mendeskripsikan konfigurasi objek-objek dan tautan-tautan yang mungkin terjadi saat suatu *instance* kolaborasi dieksekusi. (Adi Nugroho ; 2010 : 44)



Gambar II.5. Contoh Collaboration Diagram
(*Sumber : Adi Nugroho ; 2010 : 44*)

II.5.5. Activity View

Diagram aktivitas (activity diagram) sesungguhnya merupakan bentuk khusus dari state machine yang bertujuan memodelkan komputasi-komputasi dan aliran-aliran kerja yang terjadi dalam sistem/perangkat lunak yang sedang dikembangkan. Biasanya, suatu diagram aktivitas mengasumsikan komputasi-komputasi dilaksanakan tanpa adanya interupsi-interupsi eksternal berbasis event terjadi padanya (Adi Nugroho, 2010 : 62).



Gambar II.6. Contoh Activity Diagram
(*Sumber : Adi Nugroho ; 2009 : 11*)

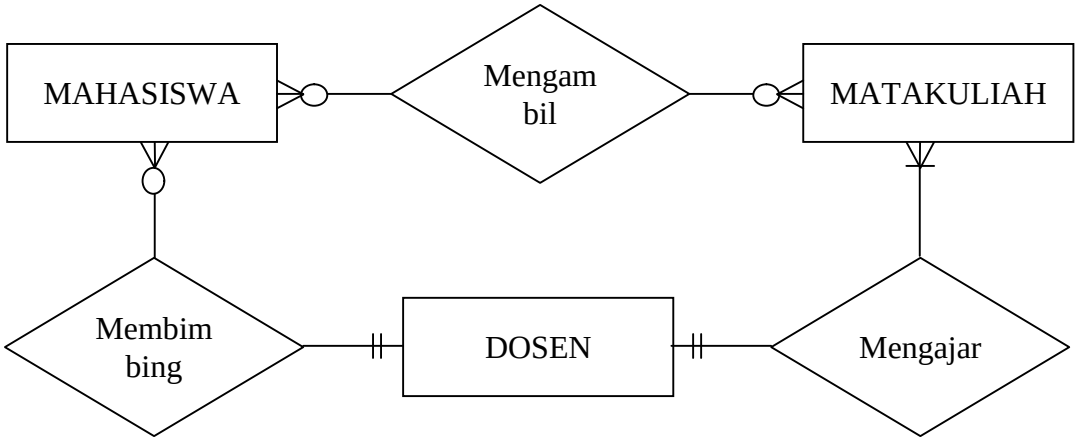
II.6. *Entity Relational Diagram* (ERD)

Entity Relationship Diagram (ERD) digunakan untuk mengidentifikasi data yang akan diambil, disimpan, dan dipanggil kembali (*retrieve*) untuk keperluan-keperluan tertentu dalam mendukung kegiatan yang dilakukan oleh organisasi. ERD juga digunakan untuk mengidentifikasi asal data yang dibutuhkan dan dilaporkan. (Prof. Dr. Ir. Marimin ; 111)

ERD (model data) merupakan alat yang digunakan untuk menggambarkan kebutuhan data dan asumsi-asumsi dalam sistem yang akan dibangun/dikembangkan secara terstruktur dari atas ke bawah. Model data ini juga diatur pada tahapan SDLC dalam mendesain *database*. Pembuatan ERD membutuhkan pemahaman terhadap sistem dan komponen-komponen yang menyusunnya.

Untuk mempermudah dalam perancangan *database*, maka digunakan *Entity Relationship Diagram* (ERD). ERD diutamakan untuk pemodelan dari desain konseptual. ERD menggambarkan struktur dan keterkaitan tabel-tabel data yang menyusun *database* secara detail. ERD merupakan representasi data sebagai entitas, attribute, dan relasi.

Model ini dinyatakan dalam bentuk diagram. Itulah sebabnya model ERD acapkali juga disebut sebagai diagram ERD. perlu diketahui bahwa model seperti ini tidak mencerminkan bentuk fisik yang nantinya akan disimpan dalam database, melainkan hanya bersifat konseptual. Itulah sebabnya model ERD tidak bergantung pada produk DBMS yang akan digunakan. Contoh sebuah model ERD ditunjukkan pada gambar II.7.



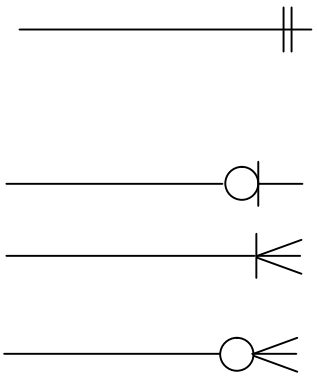
Gambar II.7. Contoh model E-R
(*Sumber: Abdul kadir;2009:30*)

Secara garis besar model E-R di atas menerangkan hubungan antara :

- Mahasiswa dan mata kuliah yang diambil,
- Dosen dan mahasiswa yang dibimbing,
- Dosen dan mata kuliah yang diajar.

Dalam hal ini MAHASISWA, MATA KULIAH, dan DOSEN menyatakan tipe entitas dan Mengambil, Membimbing, dan Mengajar menyatakan hubungan. Sekedar untuk diketahui, model E-R melibatkan sejumlah notasi. Beberapa notasi dasar dalam model E-R ditunjukkan pada tabel II.2.

Tabel II.2 Sejumlah notasi pada model ERD

Simbol	Kardinalitas hubungan
Entitas	
Attribute	
Relasi	

(*Sumber: Abdul Kadir;2009:31*)

II.7. Kamus Data

Kamus data (KD) atau data *dictionary* (DD) atau disebut juga dengan istilah systems data dictionary adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan KD, analisis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap. KD dibuat pada tahap analisis sistem dan digunakan baik pada tahap analisis maupun pada tahap perancangan sistem.

Pada tahap analisis, KD dapat digunakan sebagai alat komunikasi anatara analisis sistem dengan pemakai sistem tentang data yang mengalir di sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem, KD digunakan untuk merancang input, merancang laporan-laporan dan database. KD dibuat berdasarkan arus data

yang di DAD. Arus data di DAD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja. (Jogiyanto;2005:725)

II.8. Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsng berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standar untuk menghasilkan struktur tabel yang normal. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari model ERD ke bentuk fisik. (Kusrini ; 2007 ; 39-43)

Dalam perspektif normalisasi sebuah *database* dikatakan baik jika setiap tabel yang membentuk basis data sudah berada dalam keadaan normal. Suatu tabel dikatakan normal, jika :

1. Jika ada dekomposisi/pengurauan tabel, maka dekomposisinya dijamin aman (*lossless-join decomposition*).
2. Terpeliharanya ketergantungan *functional* pada saat perubahan data (*dependency preservation*).
3. Tidak melanggar *Boyce Code Normal Form* (BCNF), jika tidak bias minimal tidak melanggar bentuk normalisaasi ketiga.

II.8.1 Bentuk-Bentuk Normalisasi

1. 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.

2. Bentuk normal tahap pertama (1st Normal Form)

   Defenisi :

   Sebuah tabel disebut 1NF jika :

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

   Contoh normalisasi dari tabel kuliah yang memiliki atribut : kode_kul, nama_kul, sks, semester, waktu, tempat, dan nama_dos.

   Tabel kuliah tersebut tidak memenuhi normalisasi pertama, karena terdapat atribut waktu yang tergolong ke dalam atribut bernilai banyak. Agar tabel tersebut dapat memenuhi 1 NF, maka solusinya adalah dengan mendekomposisi tabel kuliah menjadi :

   a. Tabel kuliah (kode_kul, nama_kul, sks, semester, nama_dos).
   b. Tabel jadwal (kode_kul, waktu, ruang).

3. Bentuk normal taham kedua (2 Normal Form)

   Bentuk normal kedual (2NF) terpenuhi jika pada sebuah tabel semua atribut yang tidak termasuk dalam *primary key* memiliki ketergantungan fungsional pada primary key secara utuh. Sebuah tabel dikatakan tidak memenuhi 2NF,

jika keterantungannya hanya bersifat parsial (hanya tergantung pada sebagian dari primary key).

4. Bentuk normal tahap ketiga (3rd Normal Form)

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

   a. X haruslah *superkey* pada tabel tersebut.
   b. A merupakan bagian dari *primary key* pada tabel tersebut.

5. 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 5NF merupakan nama lain dari *Project Join Normal Form* (PJNF).