

BAB II

LANDASAN TEORI

2.1. Konsep Sistem Informasi

Pada bab ini berisi penjelasan tentang kategori yang menjadi landasan dalam penyusunan laporan. Secara garis besar akan dijelaskan mengenai pengertian-pengertian dan konsep-konsep dasar yang akan digunakan dalam analisa dan perancangan sistem informasi yang akan dibuat.

2.1.1. Definisi Sistem Informasi

Sistem Informasi dapat didefinisikan sebagai suatu susunan dari orang, data, proses, dan teknologi informasi yang saling berhubungan untuk mengumpulkan, memproses, menyimpan, dan menyediakan keluaran informasi yang diperlukan untuk mendukung suatu organisasi. Sistem informasi dapat digolongkan menurut fungsinya, antara lain adalah sebagai berikut :

- a. *Transaction Processing System* (TPS), suatu sistem informasi yang menangkap dan memproses data tentang transaksi bisnis, seperti pesanan (*order*), kartu catatan waktu, pembayaran, reservasi, dan sebagainya.
- b. *Management Informasi System* (MIS), suatu sistem informasi yang disediakan untuk menghasilkan laporan yang berorientasi pada manajemen yang berdasarkan pada proses transaksi dan operasi dari organisasi. Atau dengan kata lain menggunakan data transaksi untuk menghasilkan informasi yang dibutuhkan oleh manajer untuk menjalankan bisnis.
- c. *Decision Support System* (DSS), suatu sistem informasi yang membantu mengidentifikasi pengambilan keputusan yang mungkin atau menyediakan informasi untuk membantu pengambilan keputusan manajemen.
- d. *Executive Information System* (EIS), suatu sistem informasi yang mendukung perencanaan dan kebutuhan penilaian dari manajer eksekutif. EIS dikhususkan untuk kebutuhan informasi yang unik dari para eksekutif yang merencanakan bisnis dan menilai capaian rencana bisnis tersebut.

- e. *Expert System (ES)*, suatu sistem informasi yang menangkap keahlian dari para pekerja dan kemudian menirukan keahlian tersebut untuk dimanfaatkan oleh orang yang tidak ahli.
- f. *Communications and Collaboration System*, suatu sistem informasi yang memberikan peluang komunikasi yang lebih efektif antara para pekerja, mitra, pelanggan, dan para penyalur untuk meningkatkan kemampuan mereka untuk bekerja sama.
- g. *Office Automation System*, suatu sistem informasi yang mendukung cakupan luas dari aktivitas kantor yang disediakan untuk meningkatkan alur kerja (*work flow*) antara para pekerja dan membantu karyawan membuat dan membagi dokumen yang dapat mendukung aktivitas kantor sehari-hari.

2.1.2. Komponen Sistem Informasi

Menurut Yakub (2012:20), bahwa sistem informasi merupakan sebuah susunan yang terdiri dari beberapa komponen atau elemen^[8]. Komponen sistem informasi disebut dengan istilah blok bangunan (*building block*). Komponen sistem informasi tersebut terdiri dari :

- a. Blok masukan (*Input Block*), input memiliki data yang masuk ke dalam sistem informasi. juga metode-metode untuk menangkap data yang dimasukkan.
- b. Blok model (*Model Block*), blok ini terdiri dari kombinasi prosedur logika dan model matematik yang akan memanipulasi data input dan data yang tersimpan di basis data.
- c. Blok keluaran (*Output Block*), produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.
- d. Teknologi (*Technology Block*), blok teknologi digunakan untuk menerima input, menyimpan dan mengakses data, menghasilkan dan mengirimkan keluaran dari sistem secara keseluruhan. Terdiri dari 3 bagian utama, yaitu teknisi (*brainware*), perangkat lunak (*software*) dan perangkat keras (*hardware*).

- e. Blok basis data (*Database Block*), Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lainnya, tersimpan di perangkat keras komputer dan digunakan perangkat lunak (*software*) untuk memanipulasinya. Basis data diakses atau dimanipulasinya.

2.2. Analisa dan Perancangan Sistem Informasi Berorientasi Obyek dengan UML

Analisa sistem dapat dinyatakan sebagai pemisah suatu hal dalam bagian-bagian tertentu. Bagian-bagian tersebut kemudian dipelajari dan dievaluasi untuk mengetahui apakah terdapat cara-cara yang lebih baik untuk memenuhi kebutuhan manajemen. “Analisa sistem adalah proses menentukan kebutuhan siste, apa yang harus dilakukan sistem untuk memenuhi kebutuhan klien, bukanlah bagaimana sistem tersebut diimplemantasikan”.

Secara spesifik pengertian “Berorientasi Objek” berarti bahwa kita mengorganisasi perangkat lunak sebagai kumpulan dari objek tertentu yang memiliki struktur data dan perilakunya. Hal ini yang membedakan dengan pemrograman konvensional dimana struktur data dan perilaku hanya berhubungan secara terpisah. Terdapat beberapa cara untuk menentukan karakteristik dalam pendekatan berorientasi objek, tetapi secara umum mencakup empat hal, yaitu identifikasi, klasifikasi, *polymorphism* (polimorfisme) dan *inheritance* (pewarisan)

2.2.1. Unified Modelling Language (UML)

Unified Modelling Language (UML) adalah sebuah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka ia lebih cocok untuk penulisan

piranti lunak dalam bahasa-bahasa berorientasi objek. Notasi UML terutama diturunkan dari 3 notasi yang telah ada sebelumnya: *Grady Booch OOD (Object-Oriented Design)*, *Jim Rumbaugh OMT (Object Modelling Technique)*, dan *Ivar Jacobson OOSE (Object-Oriented Software Engineering)*.

Cakupan UML diantaranya: Pertama, UML menggabungkan konsep BOOCH, OMT, dan OOSE, sehingga UML merupakan suatu bahasan pemodelan tunggal yang umum dan digunakan secara luas oleh para user ketiga metode tersebut dan bahkan para user metode lainnya. Kedua, UML menekankan pada apa yang dapat dikerjakan dengan metode-metode tersebut. Ketiga, UML berfokus pada suatu bahasa pemodelan standar, bukan pada proses standar.

2.2.2. Analisa Sistem Berorientasi Objek

Alat-alat bantu yang digunakan dalam analisa berorientasi objek dengan UML antara lain adalah:

a. Activity Diagram

Diagram memodelkan alur kerja (work flow) sebuah proses bisnis dan urutan aktivitas pada suatu proses. Diagram ini sangat mirip dengan flow chart, karena kita dapat memodelkan prosedur logika, proses bisnis dan alur kerja. Perbedaan utamanya adalah flow chart dibuat untuk menggambarkan alur kerja dari sebuah sistem, sedangkan activity diagram dibuat untuk menggambarkan aktivitas dari aktor.

Activity diagram adalah teknik untuk mendeskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. Activity diagram mempunyai persamaan seperti halnya flow chart, akan tetapi perbedaannya dengan flow chart adalah activity diagram bisa mendukung perilaku paralel sedangkan flow chart tidak bisa.

Simbol-simbol yang digunakan pada saat pembuatan activity diagram adalah sebagai berikut :

- 1) *Start Point*, diletakkan pada pojok kiri atas dan merupakan awal aktifitas.



Simbol Start point

2) *End Point*, akhir aktifitas.



Simbol End Point

3) *Activity*, menggambarkan suatu proses / kegiatan bisnis.

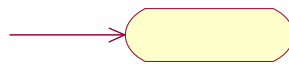


Simbol Activity

Jenis – jenis *Activities* yaitu :

a) *Black Hole Activities*

Ada masukan dan tidak ada keluaran, biasanya digunakan jika dikehendaki ada satu atau lebih transisi.



Simbol Black Hole Activities

b) *Miracle Activities*

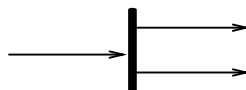
Tidak ada masukan dan ada keluaran, biasanya dipakai pada waktu startpoint dan dikehendaki ada satu atau lebih transisi.



Simbol *Miracle Activities*

c) *Fork* (Percabangan)

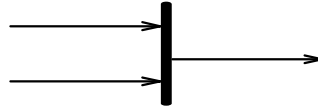
Fork digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu.



Simbol Fork dalam UML

d) *Join* (Penggabungan)

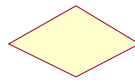
Join (penggabungan)/*Rake*, menunjukkan adanya dekomposisi. Yaitu mempunyai 2 atau lebih transisi masuk dan hanya satu transisi keluar, dan *fork* harus berhubungan dengan *join*.



Simbol Join dalam UML

4) *Decision Point*

Decision digambarkan dengan lambing wajik atau belah ketupat. Mempunyai transisi (sebuah garis dari atau ke *decision point*). Setiap transisi yang ada harus mempunyai *guard* (kunci). Tidak ada keterangan (pernyataan) pada tengah belah ketupat seperti pada *flowchart*.



Simbol Decision point

5) *Guard* (Kunci)

Guard (kunci) adalah kondisi benar sewaktu melewati sebuah transisi. Digambarkan dengan diletakkan diantara tanda []. Tanda (*otherwise*) guard untuk menangkap suatu kondisi yang belum terdeteksi. Setiap transisi dari atau ke *decision point* harus mempunyai *guard* yang harus konsisten dan lengkap serta tidak *overlap*.

6) *Swimlane*

Swimlane merupakan sebuah cara untuk mengelompokkan activity berdasarkan *actor*. *Actor* bisa ditulis nama *actor* ataupun sekaligus dalam lambang actor. *Swimlane* digambarkan secara *vertical*, walaupun kadang-kadang digambarkan secara *horizontal*.



Simbol Swimlane

7) Swimarea

Ketika sebuah *activity diagram* mempunyai banyak *swimlane*, perlu dipikirkan dengan pendekatan *swimarea*. *Swimarea* mengelompokkan *activity* berdasarkan kegiatan didalam *use case*.

b. Analisa Dokumen Keluaran

Analisa keluaran adalah bagian dari sistem yang fungsinya menjelaskan dokumen-dokumen apa saja yang dihasilkan sistem berjalan. Dokumen keluaran yang dihasilkan dari sistem berjalan dapat dianalisa dari *activity diagram* yang telah dibuat.

c. Analisa Dokumen Masukan

Analisa masukan adalah bagian dari sistem yang fungsinya menjelaskan dokumen-dokumen apa saja yang dihasilkan sistem berjalan. Dokumen masukan yang dihasilkan dari sistem berjalan dapat dianalisa dari *activity diagram* yang telah dibuat.

d. Use case Diagram

Use case konstruksi digunakan untuk mendeskripsikan bagaimana sistem akan terlihat dimata pengguna potensial. *Use case* terdiri dari sekumpulan skenario yang dilakukan oleh seseorang aktor (Orang, perangkat keras, urutan waktu, atau sistem yang lain). Sedangkan *use case diagram* memfasilitasi komunikasi diantara analis dan klien. Ada hal-hal yang umumnya dipakai untuk menggunakan kembali *use case* yang sudah ada. Untuk itu bisa dipakai <<*include*>> untuk menunjukkan sebuah *use case* adalah bagian dari *use case* yang lain <<*extend*>> digunakan untuk membuat *use case* baru dengan menambahkan langkah-langkah pada *use case* yang sudah ada.

1) Use case

Use case adalah gambaran fungsionalitas dari suatu sistem, sehingga *customer* atau pengguna sistem paham dan mengerti mengenai kegunaan

sistem yang akan dibangun. Cara menentukan *use case* dalam suatu sistem sebagai berikut:

- a) Pola perilaku perangkat lunak aplikasi
- b) Gambaran tugas dari sebuah aktor
- c) Sistem atau “benda” yang memberikan sesuatu yang bernilai kepada aktor
- d) Apa yang dikerjakan oleh suatu perangkat lunak (bukan bagaimana cara mengerjakannya).



Simbol Use Case

2) *Actor*

Pada dasarnya aktor bukanlah bagian dari *use case diagram*, namun untuk dapat terciptanya suatu *use case diagram* diperlukan beberapa aktor. Aktor tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. Sebuah aktor mungkin hanya memberikan informasi inputan pada sistem, hanya menerima informasi dari sistem atau keduanya menerima, dan memberi informasi pada sistem. Aktor hanya berinteraksi dengan *use case*, tetapi tidak memiliki kontrol atas *use case*. Aktor digambarkan secara umum dan spesifik, dimana untuk membedakannya kita dapat menggunakan *relationship*.



Simbol Aktor

3) *Associations* (Asosiasi) / *Relationship* (Relasi)

Associations menggambarkan bagaimana *actor* terlibat dalam *usecase* dan bukan menggambarkan aliran data atau informasi. *Associations* digambarkan dengan sebuah garis berpanah terbuka pada salah satu ujungnya yang menunjukkan arah relasi. Jenis-jenis relasi bisa timbul pada *usecase diagram* adalah sebagai berikut :

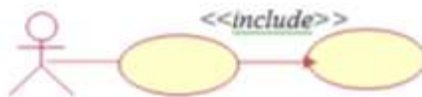
a) *Associations* antara *actor* dengan *usecase*

Ujung panah pada *Associations* antara *actor* dengan *usecase* mengidentifikasi siapa/apa yang meminta interaksi dan bukannya mengidentifikasi aliran data.

b) *Associations* antara *usecase*

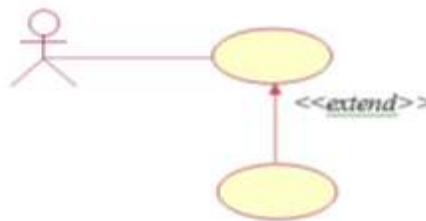
Keterhubungan antara *usecase* lain berupa generalisasi antara *usecase*, yaitu :

- (1) `<<Include>>` digunakan untuk menggambarkan bahwa suatu *usecase* seluruhnya merupakan fungsionalitas dari *usecase* lainnya. Biasanya `<<Include>>` digunakan untuk menghindari pengkopian suatu *usecase* karena sering dipakai.



Contoh Include pada UML

- (2) `<<Extend>>` digunakan untuk menunjukkan bahwa satu *usecase* merupakan tambahan fungsional dari *usecase* yang lain jika kondisi satu syarat tertentu yang terpenuhi



Contoh Extend pada UML

e. Deskripsi *Use Case Diagram*

Bagian terbesar dari use case merupakan deskripsi naratif dari urutan utama use case yang merupakan urutan yang paling umum dari interaksi antara aktor dan sistem. Deskripsi tersebut dalam bentuk input dari aktor, diikuti oleh respon pada sistem. Sistem ditandai dengan sebuah kotak hitam (black box) yang berkaitan

dengan apa yang sistem lakukan dalam merespon input aktor, bukan bagaimana internal melakukannya.

2.2.3. Perancangan Sistem Berorientasi Objek

Perancangan berorientasi obyek merupakan tahap lanjutan setelah analisa berorientasi obyek, perancangan berorientasi obyek adalah suatu pendekatan yang digunakan untuk menspesifikasi kebutuhan – kebutuhan sistem dengan mengkolaborasikan obyek–obyek, atribut–atribut, dan *method–method* yang ada. (Whitten 2004:686). Tujuan perancangan sistem itu untuk memahami kebutuhan kepada pemakai sistem (*user*) dan memberikan gambaran yang jelas serta rancang bangun yang lengkap.

a. ERD (*Entity Relationship Diagram*)

Entity Relationship adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis data semantik sistem. Dimana sistem seringkali memiliki basis data relasional dan ketentuannya bersifat *top-down*. Diagram untuk menggambarkan model *Entity Relationshi* ini disebut *Entity Relationship Diagram* (ERD). Notasi-notasi yang sering digunakan dalam *Entity Relationship Diagram* adalah sebagai berikut:

- 1) Entitas adalah segala sesuatu yang digambarkan oleh data. Entitas juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain. Ada dua macam entitas yaitu entitas kuat dan entitas lemah. Entitas kuat merupakan entitas yang tidak memiliki ketergantungan dengan entitas lainnya, contohnya entitas anggota. Sedangkan entitas lemah merupakan entitas yang kemunculannya tergantung pada keberadaan entitas lain dalam suatu relasi.



Simbol Entitas

- 2) Atribut merupakan pendeskripsian karakteristik dari entitas. Atribut digambarkan dalam bentuk lingkaran atau elips. Atribut menjadi kunci

entitas atau *key* diberi garis bawah. Dari setiap atribut entitas terdapat satu atribut yang dijadikan kunci, yaitu:

a) *Primary Key*

Field yang mengidentifikasi sebuah *record* dalam *file* dan bersifat unik.

b) *Secondary Key*

Field yang mengidentifikasi sebuah *record* dalam *file* tetapi tidak bersifat unik.

c) *Candidate Key*

Field yang dapat dijadikan sebagai calon *primary key*.

d) *Alternate Key*

Field yang tidak terpilih menjadi *primary key* dari beberapa *key*.

e) *Composite Key*

Jika tidak ada satupun *field* yang bisa dijadikan *primary key*, maka beberapa *field* dapat digabung menjadi satu.

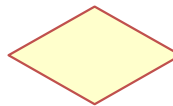
f) *Foreign Key*

Field yang bukan *key* tetapi merupakan *key* pada *file* lain.



Simbol Atribut

- 3) Relasi atau hubungan menunjukkan adanya hubungan diantara sejumlah entitas yang berasal dari himpunan entitas yang berbeda.



Simbol Relasi

- 4) Penghubung antara himpunan relasi dengan himpunan entitas dan himpunan entitas dengan atribut dinyatakan dalam bentuk garis.

b. LRS (Logical Record Structure)

Sebuah model sistem yang digambarkan dengan sebuah Diagram-ER akan mengikuti pola/aturan pemodelan tertentu. Dalam kaitannya dengan konversi ke LRS, maka perubahan yang terjadi adalah mengikuti aturan-aturan berikut ini:

- 1) Setiap entitas akan diubah ke bentuk kotak.
- 2) Sebuah atribut relasi disatukan dalam sebuah kotak bersama entitas jika hubungan yang terjadi pada diagram-ER 1:M (relasi bersatu dengan *cardinality* M) atau tingkat hubungan 1:1 (relasi bersatu dengan *cardinality* yang paling membutuhkan referensi), sebuah relasi dipisah dalam sebuah kotak tersendiri (menjadi entitas baru). jika tingkat hubungannya M:M (*many to many*) dan memiliki *foreign key* sebagai *primary key* yang diambil dari kedua entitas yang sebelumnya saling berhubungan.

c. Tabel / Relasi

Tabel adalah koleksi objek yang terdiri dari sekumpulan elemen yang diorganisasi secara kontinyu, artinya memori yang dialokasi antara satu elmen dengan elmen yang lainnya mempunyai *address* yang berurutan. Pada tabel, pengertian perlu dipahami adalah:

- 1) Nama *table* ; harus unik sehingga dapat dibedakan dengan *table* lain.
- 2) Deskripsi kolom ; (kolom kadang disebut juga atribut, *field* atau data *item*) nama kolom, domain kolom (menyangkut jenis data tergantung *database* yang digunakan), panjang, dan *rang* yang diperbolehkan.
- 3) *Referential Integrity Constraint*
 - a. Definisi apakah kolom tersebut termasuk *primary key*.
 - b. Hubungan *foreign key* pada *table* dengan *primary key* dari *table* lain.

d. Spesifikasi Basis Data

Spesifikasi basis data menggambarkan struktur data fisik pada suatu sistem atau aplikasi. Spesifikasi basis data menyajikan bagaimana penyimpanan data

dilakukan di software basis data. Di dalam dokumentasi sistem, spesifikasi basis data juga perlu ditampilkan. Bentuk dari spesifikasi basis data sendiri secara umum berupa tabel yang menyajikan informasi field untuk seluruh tabel yang digunakan. Informasi field yang ditampilkan antara lain nama field, tipe field, panjang field dan field yang menjadi field kunci (primary key).

e. Rancangan Dokumen Keluaran

Rancangan keluaran merupakan informasi yang akan dihasilkan dari keluaran sistem yang dirancang. Istilah keluaran dapat terdiri dari berbagai jenis. Yang dimaksud keluaran dari rancangan sistem ini adalah keluaran yang berupa tampilan di media kertas atau layar komputer.

f. Rancangan Dokumen Masukan

Rancangan masukan merupakan data yang dibutuhkan untuk menjadi masukan sistem yang dirancang.

g. Rancangan Layar Program

Rancangan tampilan merupakan bentuk tampilan sistem layar komputer sebagai antar muka dengan pemakai yang akan dihasilkan dari sistem yang dirancang.

h. Sequence Diagram

Menurut Munawar (2005 : 63) *Sequence Diagram* digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sebuah contoh objek dan *message* (pesan) yang diletakkan diantara objek-objek ini didalam *use case*. Komponen utama *Sequence Diagram* terdiri atas :

1) *Actor*, menggambarkan orang yang sedang berintraksi dengan sistem.



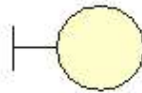
Simbol Actor

- 2) *Entity Object*, suatu obyek yang berisi informasi kegiatan yang terkait yang tetap dan disimpan ke dalam suatu *database*.



Simbol Entity Object

- 3) *Interface/Boundary Object*, sebuah obyek yang menjadi penghubung antara user dengan sistem. Contohnya *window*, *dialogue box* atau *screen* (tampilan layar).



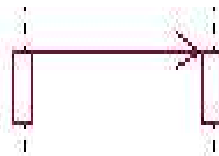
Simbol Boundary Object

- 4) *Control Object*, suatu obyek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai obyek. *Control object* mengkoordinir pesan (*message*) antara *boundary* dengan entitas.



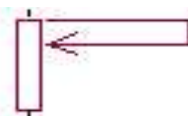
Simbol Control Object

- 5) *Simple Message*, simbol pengiriman pesan dari sebuah obyek ke obyek lain.



Simbol Simple Message

- 6) *Recursive*, sebuah obyek yang mempunyai sebuah *operation* kepada dirinya sendiri.



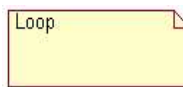
Simbol Recursive

- 7) *Lifeline*, garis titik-titik yang terhubung dengan obyek, sepanjang *lifeline* terdapat *activation*.



Simbol Lifeline

- 8) *Loop*, menggambarkan suatu kegiatan yang dilakukan secara berulang-ulang



Simbol Loop

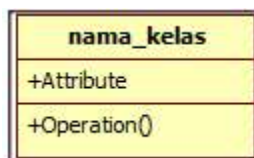
i. Class Diagram (Entity Class)

Class Diagram merupakan diagram paling umum dipakai di semua pemodelan orientasi objek. Pemodelan *class* merupakan pemodelan paling utama di pendekatan berorientasi objek. Pemodelan *class* menunjukkan *class-class* yang ada di sistem dan hubungan antar *class*. *Class Diagram* digambarkan dengan sebuah kotak dengan 3 *section*.

Komponen – komponen *Class Diagram* :

1) *Class Name*

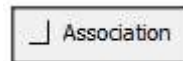
Nama *class* menggunakan huruf besar di awal kalimatnya dan diletakkan diatas kotak. Bila *class* mempunyai nama yang terdiri dari 2 suku kata atau lebih, maka semua suku kata digabungkan tanpa spasi dengan huruf awal tiap suku kata menggunakan huruf besar.



Simbol *Class*

2) *Association*

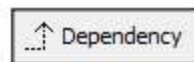
Association adalah konsep dasar hubungan antar *class*. Setiap *class* pada asosiasi memainkan sebuah peran dan *multiplicity* memberikan spesifikasi berapa banyak objek pada suatu *class* berhubungan dengan suatu *class* pada asosiasi *class*.



Simbol *Association*

3) *Dependency*

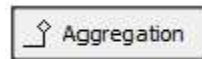
Umumnya penggunaan *Dependency* digunakan untuk menunjukkan operasi pada suatu *class* yang menggunakan *class* lain.



Simbol *Dependency*

4) *Aggregation*

Relasi antar kelas dengan makna semua-bagian



Simbol *Aggregation*

2.3. Model Prototype

Dalam perancangan Tugas Akhir ini penulis menggunakan metode *Prototype*. *Prototype Model* adalah salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan Metode Prototyping ini pengembangan dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem. Sering terjadi seorang pelanggan hanya mendefinisikan secara umum apa yang dibutuhkan, Pemrosesan dan data-data apa saja yang dibutuhkan. Sebaliknya disisi pengembang Kurang memperhatikan efisiensi Algoritma. Kemampuan sistem oprasi dan interface yang menghubungkan manusia dengan computer.

Pengumpulan Kebutuhan Pada Prototyping model kadang – kadang klien hanya memberikan beberapa kebutuhan umum software tanpa detile input,

proses atau detail output dilain waktu mungkin tim pembangun (developer) tidak yakin terhadap efesiensi dari algoritma yang digunakan, tingkat adaptasi terhadap sistem operasi atau rancangan form user interface. Ketika situasi seperti ini terjadi model prototyping sangat membantu proses pembangunan software.

Proses pada prototyping bisa dijelaskan sebagai berikut

a. Pengumpulan kebutuhan

developer dan klien bertemu dan menentukan tujuan umum, kebutuhan yang diketahui dan gambaran bagian-bagian yang akan dibutuhkan berikutnya. Detail kebutuhan mungkin tidak dibicarakan disini, pada awal pengumpulan kebutuhan.

b. Perancangan

Perancangan dilakukan cepat dan rancangan mewakili aspek software yang diketahui. Dan rancangan ini menjadi dasar pembuatan prototype.

c. Evaluasi Prototype

klien mengevaluasi prototype yang dibuat dan dipergunakan untuk memperjelas kebutuhan software.

2.3.1. Tahapan-Tahapan *Prototype*

Tahap-tahap pengembangan *Prototype* model menurut *Roger S. Pressman, Ph.D.* adalah :

a. Mendengarkan pelanggan

Pada tahap ini dilakukan pengumpulan kebutuhan dari system dengan cara mendengar keluhan dari pelanggan. Untuk membuat suatu system yang sesuai kebutuhan, maka harus diketahui terlebih dahulu bagaimana system yang sedang berjalan untuk kemudian mengetahui masalah yang terjadi.

b. Merancang dan Membuat Prototype

Pada tahap ini, dilakukan perancangan dan pembuatan prototype system. *Prototype* yang dibuat disesuaikan dengan kebutuhan system yang telah didefinisikan sebelumnya dari keluhan pelanggan atau pengguna.

c. Uji coba

Pada tahap ini, *Prototype* dari system di uji coba oleh pelanggan atau

pengguna. Kemudian dilakukan evaluasi kekurangan-kekurangan dari kebutuhan pelanggan. Pengembangan kemudian kembali mendengarkan keluhan dari pelanggan untuk memperbaiki *Prototype* yang ada.

2.3.2. Kelebihan Model *Prototype*

Kelebihan Model *Prototype* :

- a. Adanya komunikasi yang baik antara pengembang dan pelanggan
- b. Pengembangan dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
- c. Lebih menghemat waktu dalam pengembangan system
- d. Penerapan menjadi lebih mudah karna pemakai mengetahui apa yang diharapkannya

2.3.3. Kekurangan Model *Prototype*

Kekurangan model *Prototype* :

- a. Resiko tinggi yaitu untuk masalah-masalah yang tidak terstruktur dengan baik, ada perubahan yang besar dari waktu ke waktu, dan adanya persyaratan data yang tidak menentu.
- b. Interaksi pemakai penting, sistem harus menyediakan dialog on-line antara pelanggan dengan komputer.
- c. Hubungan yang di sediakan antara pelanggan dengan komputer mungkin tidak mencerminkan teknik perancangan yang baik.

2.4. Sistem Informasi Akademik

a. Pengertian Sistem Informasi Akademik

Sistem Informasi Akademik merupakan suatu sistem yang berfungsi untuk mengolah data-data yang berhubungan dengan bagian akademik pada suatu lembaga pendidikan. Pada umumnya data-data yang diolah pada suatu sistem informasi akademik diantaranya meliputi data siswa, data pengajar, data mata pelajaran, dan data-data lain yang bersifat intern berdasarkan kebutuhan masing-

masing lembaga pendidikan. Didalam sistem akademik terdapat sistem lain yang saling mendukung dalam rangka mencapai suatu tujuan. Sistem akademik dirancang sesuai dengan visi misi agar mampu mendukung dalam pencapaian tujuan lembaga pendidikan.

b. Pengertian Pendidikan

Untuk mendapatkan Sumber Daya Manusia yang berkualitas, dibutuhkan suatu pendidikan bagi manusia baik yang bersifat formal maupun non formal, karena pendidikan merupakan sarana utama untuk meningkatkan tingkat kualitas seseorang. Pengertian pendidikan menurut beberapa pakar adalah sebagai berikut :

“Driyarkara (1980) mengatakan bahwa pendidikan adalah memanusiakan manusia muda. Pengangkatan manusia muda ketaraf mendidik”.

Dalam *Dictionary of Education*, Pendidikan adalah :

- 1) Proses seseorang mengembangkan kemampuan, sikap, dan tingkah laku lainnya di dalam masyarakat tempat mereka hidup.
- 2) Proses sosial yang terjadi pada orang yang diharapkan pada pengaruh lingkungan yang terpilih dan terkontrol (khususnya yang datang dari sekolah). Sehingga mereka dapat memperoleh perkembangan kemampuan sosial dan kemampuan individu yang optimum. Dengan kata lain, pendidikan dipengaruhi oleh lingkungan atas individu untuk menghasilkan perubahan-perubahan yang sifatnya permanen (tetap) dalam tingkah laku, pikiran dan sikapnya.

Dari pengertian diatas, dapat disimpulkan bahwa pendidikan berusaha mengembangkan potensi individu agar mampu berdiri sendiri. Untuk itu individu perlu diberi berbagai kemampuan dan pengembangan dalam berbagai hal untuk membentuk kepribadian manusia seutuhnya yaitu mengembangkan manusia sebagai makhluk individu, makhluk sosial, dan makhluk beragama.

Ciri pendidikan adalah :

- 1) Pendidikan mengandung tujuan, yaitu kemampuan untuk berkembang sehingga bermanfaat untuk kepentingan hidup.

- 2) Untuk mencapai tujuan itu, pendidikan melakukan usaha yang terencana dalam memiliki isi (materi), strategis, dan teknik penilaiannya yang sesuai.
- 3) Kegiatan pendidikan dilakukan dalam lingkungan keluarga, sekolah, dan masyarakat (formal dan non formal).

“Ciri sistem pengajaran adalah mempunyai rencana, kesalingketergantungan (unsur suatu sistem merupakan bagian yang koheren dalam keseluruhan, masing-masing bagian bersifat esensial, satu sama lain saling memberikan sumbangan tertentu), dan mempunyai tujuan”.

c. Pengertian Sistem Pengajaran

“Sistem pengajaran adalah suatu kombinasi yang terorganisasi yang meliputi unsur-unsur manusiawi, material, fasilitas, perlengkapan, dan prosedur-prosedur yang berinteraksi untuk mencapai suatu tujuan”. “Kegiatan Belajar Mengajar adalah tahap-tahap kegiatan yang dilakukan pengajar dan mahasiswa untuk menyelesaikan materi kuliah, yaitu tahap pendahuluan.

“Proses Belajar Mengajar adalah suatu penataan yang memungkinkan guru dan siswa berinteraksi satu sama lain untuk memberi kemudahan bagi siswa belajar”. Proses Belajar Mengajar (PBM) sangat erat hubungannya dengan pendidikan, dimana PBM ditempuh untuk mendapatkan suatu hasil belajar yang nantinya akan digunakan untuk melakukan proses evaluasi.

“Belajar dan mengajar adalah suatu proses yang mengandung tiga unsur yang dapat dibedakan, yaitu tujuan pengajaran (instruksional), pengalaman (proses), belajar mengajar, dan hasil belajar.

2.5. Tinjauan Pustaka

Dalam Penyusunan Skripsi penulis melakukan peninjauan terhadap 3 (tiga) Penelitian dengan judul yang sama guna mendapatkan pemahaman dalam penyusunan skripsi, yaitu :

- a. **Tinjauan Studi Penelitian Alexander Setiawan, Liliana, Bryan Andhika Universitas Kristen Petra 2012**

Penelitian ini berjudul **“Aplikasi Sistem Informasi Akademik dan Penjadwalan Mata Pelajaran Sekolah Menengah Atas di Sampit”**

Penulis dalam pembuatan aplikasi sistem informasi ini menggunakan Data Flow Diagram dan Entity Relationship Diagram. Aplikasi ini juga dibuat menggunakan berbasis web. Adapun masalah yang penulis angkat dalam Penelitian ini adalah masih dipakainya sistem manual dalam pengolahan pengkreditan barang furniture dan data transaksi simpan pinjam sehingga informasi yang dihasilkan tidak tepat waktu dan kurang efisien dan pengetikan berulang mengakibatkan kemungkinan tingkat kesalahan manual yang tinggi. Berikut ini hasil pembahasan Penelitian ini adalah :

1. Adanya kesepakatan antara pihak sekolah dan siswa dapat melihat jadwal mata pelajaran secara online.
2. Dari evaluasi yang telah dilakukan, penilaian untuk aplikasi ini dapat membantu proses akademik dalam kesehariannya, serta aplikasi ini memiliki kesesuaian dengan sistem yang ada dan dapat menghasilkan informasi yang baik.
3. Tools penjadwalan masih diproses dengan sederhana melalui proses random dan pemberian warning bentrok antar jadwal pelajaran, sehingga bagian penjadwalan ini dapat dikembangkan lebih baik dengan algoritma lainnya.

b. Tinjauan Studi Penelitian Sulastri Universitas Komputer Indonesia Bandung 2012

Penelitian ini berjudul **“Sistem Informasi Akademik Pada Sekolah Menengah Kejuruan Negeri 3 Cimahi”**. Penulis dalam pembuatan sistem informasi ini menggunakan Data Flow Diagram dan Entity Relationship Diagram Adapun masalah yang penulis angkat dalam Penelitian ini adalah bagaimana sistem informasi pengolahan data siswa, guru, jadwal, dan nilai yang berjalan pada SMKN 3 Cimahi, bagaimana perancangan sistem informasi data siswa, guru, jadwal dan nilai, bagaimana pengujian sistem informasi data siswa, guru, jadwal dan nilai,

bagaimana implementasi sistem informasi data siswa, guru, jadwal dan nilai pada SMKN 3 Cimahi. Berikut ini hasil pembahasan Penelitian ini adalah :

1. Pengolahan data akademik menjadi lebih efektif dan efisien karena adanya sistem informasi akademik pada SMKN 3 Cimahi yang sudah terkomputerisasi dan terotomatisasi ke database
2. Penginputan dan pencarian data-data siswa, data guru, data penjadwalan, dan data kelas menjadi lebih cepat karena adanya sistem informasi akademik pada SMKN 3 Cimahi yang terkomputerisasi.
3. Dengan sistem penyimpanan data akademik pada database dengan kapasitas tamping data yang besar, memungkinkan data yang disimpan lebih besar, aman, efektif dan efisien, serta menghindari terjadinya duplikasi data.

c. **Tinjauan Studi Penelitian Andri Sunaryo, Bambang Eka Purnama, Sukadi Universitas sa 2011**

Penelitian ini berjudul “**Pembuatan Sistem informasi Pengolahan Data Siswa Pada SD Negeri Sirnobojo 1 Pacitan**”. Adapun masalah yang penulis angkat dalam Penelitian ini adalah Proses pengolahan data dan nilai siswa masih menggunakan metode manual sehingga proses pendataan berjalan sangat lama, sering terjadi kesalahan data, sulitnya dalam pencarian data dan menyebabkan keterlambatan dalam penyajian laporan data-data pada saat laporan data tersebut diperlukan. Berikut ini hasil pembahasan Penelitian ini adalah :

1. Pengolahan data siswa dan nilai siswa masih manual yaitu tanpa terkomputerisasi sehingga menimbulkan beberapa masalah seperti proses pendataan berjalan sangat lama, sering terjadi kesalahan data, sulitnya dalam pencarian data dan kurang cepat sehingga menyebabkan keterlambatan dalam menyajikan informasi dan laporan yang dibutuhkan.

2. Dengan dibuatnya sistem terkomputerisasi maka memberikan kemudahan bagi pihak sekolah dalam proses penolahan data dan nilai siswa.
3. Memberikan informasi data dan nilai siswa serta mempercepat penyajian laporan saat diperlukan