

## BAB II LANDASAN TEORI

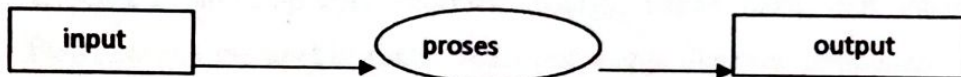
### 2.1 Konsep Sistem Informasi

Dalam menyusun dan merancang sistem informasi terdapat beberapa konsep yang dapat dijadikan sebagai acuan dan landasan, dimana konsep-konsep tersebut merupakan teori yang berhubungan dengan perancangan sistem informasi serta pemecahan masalah yang ada.

#### 2.1.1 Konsep Dasar Sistem dan Informasi

Sistem adalah sebagai suatu kesatuan yang terdiri dari interaksi elemen-elemen (subsistem) yang berusaha mencapai tujuan tertentu Narko(2007:1).

Bentuk umum dari sebuah sistem terdiri atas masukan (*input*), proses (*process*), dan keluaran (*output*), yang mana sistem dapat memiliki satu atau lebih masukan yang akan diproses dan mampu menghasilkan keluaran sesuai dengan rencana yang sudah diharapkan sebelumnya. Gambaran umum mengenai sistem ditujukan pada gambar berikut ini:



Gambar 2.1

#### Bentuk Umum Sistem

Syarat-syarat sistem :

- a. Sistem harus dibentuk untuk menyelesaikan tujuan.
- b. Elemen sistem harus mempunyai rencana yang ditetapkan.
- c. Adanya hubungan diantara elemen system.
- d. Unsur dasar dari proses ( arus informasi, energy dan material) lebih penting dari pada elemen system.

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang yang membentuknya. Berikut adalah karakteristik atau sifat-sifat suatu sistem:

1. Batasan (*boundary*). Pengembangan dari suatu elemen atau unsure mana yang termasuk di dalam sistem dan mana yang diluar sistem.
2. Lingkungan (*environment*). Segala sesuatu yang diluar sistem, lingkungan yang menyediakan asumsi, kendala dan input terhadap suatu sistem.
3. Masukan (*input*). Sumber daya (data, bahan baku, peralatan, energy) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*). Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*). Kegiatan – kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi.
6. Penghubung (*interface*). Tempat dimana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*). Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energy, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga diantara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan memungkinkan komponen yang berbeda dari berbagai data yang sama.

Informasi adalah data yang sudah diolah menjadi bentuk yang berarti bagi pengguna, yang bermanfaat dalam pengambilan keputusan saat ini atau mendukung sumber informasi. Kusrini (2007:7).

Informasi dikatakan baik jika memiliki karakteristik sebagai berikut:

a. Informasi *must be pertinent*

Informasi harus berhubungan, pernyataan informasi harus berhubungan dengan urusan dan masalah bagi penerima informasi (orang yang menerima informasi tersebut).

**b. Informasi *must be accurate***

Informasi harus bebas dari kesalahan-kesalahan dan tidak memiliki bias atau penyesatan. Informasi yang dihasilkan harus mencerminkan maksudnya.

**c. Informasi *must be timely***

Informasi harus ada ketika dibutuhkan. Informasi yang datang pada penerima tidak boleh terlambat. Informasi yang sudah usang tidak akan mempunyai nilai lagi karena informasi merupakan landasan didalam pengambilan keputusan.

**d. Relevan**

informasi tersebut memberikan manfaat untuk pendengar. Relevansi informasi untuk tiap-tiap orang yang satu dengan yang lainnya berbeda.

Nilai dari informasi (*value of information*) ditentukan oleh dua hal yaitu, manfaat dan biaya untuk mendapatkan informasi tersebut. Suatu informasi dapat dikatakan bernilai bila manfaatnya lebih efektif dibandingkan dengan biaya untuk mendapatkannya. Sebagian besar informasi tidak dapat ditaksir secara pasti nilai keuntungannya (dalam satuan uang), tetapi kita dapat menaksir nilai efektifitas dari informasi tersebut. Pengukuran nilai informasi biasanya dihubungkan dengan analisis *cost effectiveness* atau *cost benefit*.

**2.1.2 Konsep Dasar Sistem Informasi**

Definisi Sistem Informasi: "Hubungan informasi antar unit atau arus informasi yang dihasilkan dari kumpulan informasi berbagai unit. Zulkifli Amsyah(2005:3)

Sistem informasi terdiri dari komponen-komponen yang saling berinteraksi satu dengan yang lainnya membentuk suatu kesatuan untuk mencapai sasaran, yaitu:

**a. Blok Masukan**

Input mewakili data yang masuk kedalam sistem informasi. Input disini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, yang dapat berupa dokumen-dokumen dasar.

**b. Blok Keluaran**

Blok dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkatan manajemen serta semua pemakai sistem.

**c. Blok Teknologi**

Teknologi digunakan untuk menerima input, menjalankan model, menyimpan dan mengakses data, menghasilkan dan mengirim keluaran dan membantu pengendalian dari sistem secara keseluruhan. Teknologi terdiri dari 3 bagian utama yaitu teknisi (humanware, brainware), perangkat lunak (software) dan perangkat keras (hardware).

**d. Blok Model**

Blok ini terdiri dari kombinasi prosedur, logika dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara yang sudah tertentu untuk menghasilkan keluaran yang diinginkan.

**e. Blok Basis Data**

Basis data (database) merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya. Tersimpan di perangkat lunak untuk memanipulasinya.

**f. Blok Kendali**

Supaya sistem dapat berjalan sesuai dengan yang diinginkan, maka perlu diterapkan pengendalian –pengendalian didalamnya. Banyak hal yang dapat merusak sistem informasi, seperti misalnya bencana alam, api, temperature dan lain-lain.

## **2.2 Analisa dan Perancangan Sistem Berorientasi Objek dengan UML**

### **2.2.1 UML (Unified Modeling 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)

UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi piranti perangkat 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 lebih cocok untuk penulisan piranti perangkat lunak dalam bahasa-bahasa berorientasi obyek. 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*).

Adapun tujuan dari UML adalah:

- a. Memberikan bahasa pemodelan yang bebas dari berbagai bahasa pemrograman dan proses rekayasa.
- b. Menyatukan praktek-praktek terbaik yang terdapat dalam pemodelan
- c. Memberikan model-model yang siap pakai, bahasa pemodelan yang visual yang ekspresif untuk mengembangkan dan saling menukar model dengan mudah dan dimengerti secara umum.

Cakupan UML diantaranya : Pertama, UML menggabungkan konsep BOOCH, OMT, dan OOSE, sehingga UML merupakan suatu bahasa 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.

Berikut adalah 13 buah diagram UML (*Unified Modelling Language*);

1. *Class Diagram*
2. *Object Diagram*
3. *Component Diagram*

4. *Deployment Diagram*
5. *Composite Diagram*
6. *Package Diagram*
7. *Use Case Diagram*
8. *Activity Diagram*
9. *State Machine Diagram*
10. *Communication Diagram*
11. *Interaction Overview Diagram*
12. *Sequence Diagram*
13. *Timing Diagram*

### **2.2.2 Analisa Sistem Berorientasi Objek**

Analisa sistem adalah : teknik pemecahan masalah yang menggabungkan bagian-bagian komponen dengan mempelajari seberapa bagus bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan. Hanif (2001: 24).

Melakukan kajian dan menemukan berbagai faktor dari produsen penyelenggaraan pengolahan data yang berlangsung saat ini (*present system*) untuk bisa memenuhi kebutuhan akan sistem informasi yang efektif, itulah yang menjadi titik berat dari sebuah proses penganalisaan akan sebuah sistem (yang akan dikomputerisasi).

Keberhasilan dari tahap analisa adalah memahami kebutuhan-kebutuhan sistem dan membuat konsep sistem baru yang menggambarkan apa yang harus dilakukan sistem guna memenuhi kebutuhan-kebutuhan sistem.

Tujuan utama dari analisa berorientasi objek adalah memodelkan sistem yang nyata dengan penekanan apa yang harus dilakukan oleh sistem . Pada tahap analisa berorientasi objek, objek bisnis dalam sebuah sistem didentifikasikan seperti siapa atau apa aktornya dan bagaimana mereka bekerja sama dalam aplikasi. Dalam hal ini penulis menggunakan *use case* untuk mengidentifikasi apa yang akan pengguna kerjakan dengan sistem

atau perangkat lunak yang akan dikembangkan dan mengidentifikasi aktor termasuk didalamnya adalah siapa yang akan menggunakan sistem. Mengembangkan proses bisnis sederhana yang memang terjadi diorganisasi yang sedang dianalisis dengan membuat *activity diagram*.

#### a. *Activity Diagram*

*Activity Diagram* menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses, yang mana dipakai pada business modeling untuk memperlihatkan urutan aktifitas proses bisnis secara keseluruhan dalam memodlkan sebuah proses. Adi Nugroho (2010:24)

*Activity Diagram* dibuat berdasarka sebuah atau bebrapa *use case* pada *use case diagram*, atau bahkan tanpa menggunakan *use case diagram*.

Sebuah *activity diagram* mempunyai:

- 1) *Start point (initial mode)* dimana diletakkan pada pojok kiri atas
- 2) *End point (activity final mode)*
- 3) *Activities*, dimana menggambarkan proses bisnis dan dikenla sebagai *activity state*.

Jenis-jenis *activity*:

##### a) *Black hole activities*

Ada masukan dan tidak ada keluaran, biasanya digunakan bila dikehendaki Ada 1 atau lebih transisi.

##### b) *Miracle activities*

Tidak ada masukan tetapi ada keluarannya, biasanya dipakai pada waktu *start point* dan dikehendaki ada 1 atau lebih transisi.

##### c) *Parallel activities*

Suatu *activity* yang berjalan secar berbarengan . terdiri dari:

##### (1) *Fork* (Percabangan)

Mempunyai 1 transisimasuk dan 2 atau lebih transisi keluar.

##### (2) *Join* (Penggabungan)

Mempunyai 2 atau lebih transisi masuk dan hanya 1 transisi keluar.

d) *Decision point*

Digambarkan dengan lambing wajik atau belah ketupat. Mempunyai transisi (sebuah garis dari / kedalam *decision point*). Setiap transisi yang ada harus mempunyai GUARD (kunci). Tidak ada sebuah keterangan (pertanyaan) pada tengah belah ketupat seperti pada *flowchart*.

e) *Guard* (kunci)

Adalah sebuah kondisi benar sewaktu melewati sebuah transisi.

Digambarkan dengan diletakkan di antara [ ]. Tanda [otherwise] *guard* untuk menangkap suatu kondisi yang belum terdeteksi. Setiap transisi dari / ke *decision point* harus mempunyai *guard* yang harus konsisten dan lengkap serta tidak *overlap*.

f) *Swimlane*

Sebuah cara untuk mengelompokkan *activity* berdasarkan *actor* (mengelompokkan *activity* dalam sebuah urutan yang sama). Actor bisa ditulis nama *actor* ataupun sekaligus dalam lambang *actor* (*stick figure*) pada *use case diagram*. *Swimlane* digambarkan secara vertical, walaupun kadang-kadang digambarkan secara horizontal.

g) *Swimarea*

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

**b. Analisa Dokumen Keluaran**

Analisa dokumen keluaran adalah dokumen yang dihasilkan dari proses sistem, yang termasuk proses. Janner (1993:376)

**c. Analisa Dokumen Masukan**

Analisa dokumen masukan adalah dokumen yang mengalami pemrosesan didalam sistem berjalan. Dokumen masukan inilah yang akan mempengaruhi hasil laporan dari dokumen yang akan dihasilkan. Janner (1993:376)



#### **d. Usecase Diagram**

*Usecase diagram* adalah diagram yang menunjukkan fungsionalitas suatu sistem atau kelas dan bagaimana sistem tersebut berinteraksi dengan dunia luar dan menjelaskan sistem secara fungsionalitas yang terlihat user. Julius Hermawan (2004:37)

*Usecase diagram* menggambarkan kebutuhan sistem dari sudut pandang *user* dan memfokuskan pada proses komputerisasi sebuah *use case* dapat menggambarkan hubungan antara *use case* dengan *actor*. Secara umum *use case* adalah pola perilaku sistem dan urutan transaksi yang berhubungan yang dilakukan oleh satu *actor*.

*Usecase diagram* terdiri dari :

##### **1) Use case**

*Use case* adalah deskripsi fungsi dari sebuah sistem dari perspektif pengguna. *Use case* dibuat berdasarkan keperluan actor, merupakan 'apa' yang dikerjakan sistem, bukan 'bagaimana' sistem mengerjakannya. *Use case* diberi nama yang menyatakan apa hal yang dicapai dari hasil interaksinya dengan *actor* dan dinotasikan dengan gambar (*horizontal ellipse*). Julius Hermawan (2004:41)

*Use case* biasanya menggunakan kata kerja dan sebuah nama *use case* boleh terdiri dari beberapa kata tidak boleh ada 2 *use case* yang memiliki nama yang sama. *Use case diagram* tidak terpengaruh urutan waktu, meskipun demikian supaya mudah dibaca perlu penyusunan *use case*.

##### **a) Actor**

*Actor* adalah pemakai sistem, dapat berupa manusia atau sistem terotomatisasi lain. *Actor* adalah sesuatu atau seseorang yang berinteraksi dengan sistem, yaitu siapa atau berinteraksi adalah *actor* mengirim atau menerima pesan ke atau dari sistem, atau mempertukarkan informasi dengan sistem. Dalam perspektif *actor*, *use case* melakukan sesuatu yang berharga bagi *actor*. Julius Hermawan (2004:47)

*Actor* adalah tipe (kelas) instan. *Actor* merepresentasikan peran bukan pemakai individu dari sistem. *Actor* mempunyai nama, nama yang dipilih seharusnya menyatakan peran *actor*.

*Actor* berkomunikasi dengan sistem lewat pengiriman dan penerimaan pesan. *Actor* dinotasikan dengan gambar.

b) *Associations*

*Associations* menggambarkan bagaimana *actor* terlibat dalam *use case* dan bukan menggambarkan aliran data atau informasi. *Associations* digambarkan dengan sebuah garis berpanah terbuka pada salah satu ujungnya yang menunjukkan arah relasi. Adi Nugroho(2003:152)

Jenis-jenis relasi bisa timbul pada *use case diagram* adalah sebagai berikut:

(1) *Associations* antara *actor* dengan *use case*

Ujung panah pada *Associations* antara *actor* dengan *use case* mengidentifikasi aliran data.

(2) *Associations* antara *use case*

Keterhubungan antara *use case* dengan *use case* yang lain berupa generalisasi antara *use case*, yaitu :

`<<include>>`, digunakan untuk menggambarkan bahwa suatu *use case* seharusnya merupakan fungsionalisme dari *use case* lainnya. Biasanya `<<include>>` digunakan untuk menghindari pengkopian suatu *use case* karena sering dipakai.

`<<extend>>`, digunakan untuk menunjukkan bahwa satu *use case* merupakan tambahan fungsional dari *use case* yang lain jika kondisi atau syarat tertentu yang dipenuhi.

#### e. Deskripsi Usecase

Deskripsi dari sekumpulan aksi sekuesial yang ditampilkan sistem yang menghasilkan yang tampak dari nilai ke *actor* khusus. Miftakhul (2010: 138)

*Use case* digunakan untuk menyusun *behavioral things* dalam sebuah model.

*Use case* direalisasikan dengan sebuah *collaboration*.

### 2.2.3 Perancangan Sistem Berorientasi Objek

Selama analisis perhatian kita adalah pada apa yang harus dikerjakan sistem, terlepas dari bagaimana semuanya akan dikerjakan. Selama perencanaan keputusan dibuat tentang bagaimana pemecahan masalah akan dikerjakan.

Perancangan sistem berorientasi objek merupakan proses spesifikasi yang terperinci atau pendefinisian dari kebutuhan –kebutuhan fungsional dan persiapan untuk rancangan bangun implementasi yang menggambarkan bagaimana suatu sistem baru dilakukan dengan hubungan proses-proses dalam bentuk diagram-diagram.

Perancangan sistem berorientasi objek bertujuan untuk:

1. Sistematika proses pendesainan
2. Menghasilkan pendesainan model program
3. Memberikan gambaran pemecahan masukan yang efektif

Focus dari desain objek adalah perencanaan struktur data dan algoritma yang diperlukan untuk implementasi setiap kelas.

Diagram-diagram UML yang digunakan penulis dalam merancang sistem berorientasi adalah:

#### a. *Entity Relationship Diagram* (ERD)

ERD (*Entity Relationship Diagram*) merupakan diagram yang digunakan untuk menggambarkan secara sistematis hubungan antar entity-entity yang ada dalam suatu sistem database menggunakan symbol-symbol sehingga lebih mudah dipahami. ERD (*Entity Relationship Diagram*) untuk memodelkan struktur data dan hubungan antar data, untuk menggambarkan digunakan beberapa notasi dan symbol. Yuniar Supardi(2004: 78)

b. *Logical Record Structure* (LRS)

*Logical Record Structure* dibentuk dengan nomor dari tipe record, beberapa tipe record digambarkan oleh kotak persegi panjang dan dengan nama yang unik, *Logical Record Structure* terdiri dari link-link diantara tipe *record*, link ini menunjukkan arah dari satu *record* ke *record* yang lainnya

Banyaknya link dari link dari *Logical Record Structure* yang diberi nama oleh field-field yang kelihatan pada kedua link tipe record. Penggambaran *Logical Record Structure* dimulai dengan bagaimana memperoleh model yang dimengerti, dua metode yang dapat digunakan dimulai dengan hubungan model data yang dikonversikan ke *Logical Record Structure*.

c. **Tabel**

Tabel adalah kumpulan data yang tersusun secara kolom dan dan berhubungan dengan topik tertentu. (kamus besar bahasa Indonesia)

Tabel-tabel saling berhubungan dengan data base pada saat di butuhkan. Tabel juga mempunyai tipe-tipe data yang yang actual. Hal-hal penting yang harus ada pada tabel antara lain :

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

d. **Spesifikasi Basis Data**

Basis data merupakan kumpulan dari data yang saling berhubungan satu dengan yang lain dan tersipam diluar komputer serta digunakan perangkat lunak (Software) tertentu untuk memanipulasinya.

Sedangkan sistem berbasis data adalah suatu sistem penyusunan dan pengolahan record-record dengan menggunakan komputer, dengan tujuan untuk menyimpan atau merekam serta melihat data operasional lengkap pada sebuah organisasi, sehingga mampu menyediakan informasi yang diperlukan untuk kepentingan proses pengambilan keputusan.

Spesifikasi basis data adalah mendefinisikan table-tabel secara detail yang terbentuk pada tahap-tahap sebelumnya. Jogiyanto(2008)

**e. Rancangan Dokumen Keluaran**

Rancangan dokumen keluaran merupakan informasi yang akan dihasilkan dari keluaran sistem yang dirancang.

**f. Rancangan Dokumen Masukan**

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

**g. Rancangan Layar**

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

**h. Sequence Diagram**

Menurut Jhon Satzinger, 2010 dalam buku *System Analysis and Design In a Changing World*, "Sequence Diagram adalah diagram yang digunakan untuk mendefinisikan input dan output serta urutan interaksi antara pengguna dan sistem untuk sebuah use case".

Interaksi objek yang tersusun dalam urutan waktu/kejadian. Diagram ini secara khusus berasosiasi dengan use case diagram, memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu didalam use case.

Beberapa symbol yang sering digunakan pada sequence diagram, yaitu:

- 1) *Entity Object*, suatu objek yang berisi informasi kegiatan yang berkaitan dan disimpan kedalam suatu database.
- 2) *Interface/Boundary Object*, sebuah object yang menjadi penghubung antara user dengan sistem.

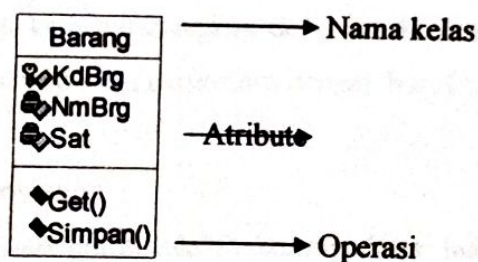
- 3) *Control Object*, suatu object yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas. Contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai object. *Control Object* mengkoordinir pesan (message) antara boundary dengan entitas.
- 4) *Simple Message*, symbol pengiriman pesan dari sebuah objek ke objek lain.
- 5) *Activation*, mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi.
- 6) *Lifeline*, garis titik-titik yang terhubung dengan objek, sepanjang lifeline terdapat activation.

#### i. Class Diagram

Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain. Djon Irwanto(2005: 60)

Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan.

Class diagram digambarkan dengan sebuah kotak dengan 3 section.



Gambar 2.2  
Class Diagram

## Komponen-komponen Class Diagram:

### 1) *Class name*

Nama class menggunakan huruf besar diawal kelimatnya 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.

### 2) *Attribute*

*Attribute* adalah property dari sebuah class, *Attribute* ini melukiskan nilai yang mungkin ada pada proyek dari class. Sebuah class mungkin mempunyai nol atau lebih *attribute*, Secara konversi, jika nama attribute terdiri dari satu suku kata maka ditulis dengan huruf kecil. Akan tetapi jika semua attribute mengandung lebih dari satu suku kata maka semua suku kata digabung dengan suku kata pertama menggunakan huruf kecil dan awal suku kata menggunakan huruf besar.

### 3) *Operation*

Operation adalah sesuatu yang bisa dilakukan oleh sebuah class atau class yang lain. Seperti halnya attribute, nama operation juga menggunakan huruf kecil semua terdiri dari satu suku kata, Maka semua suku kata digabungkan dengan suku kata pertama huruf kecil dan huruf awal tiap suku berikutnya dengan huruf besar.

### 4) *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.

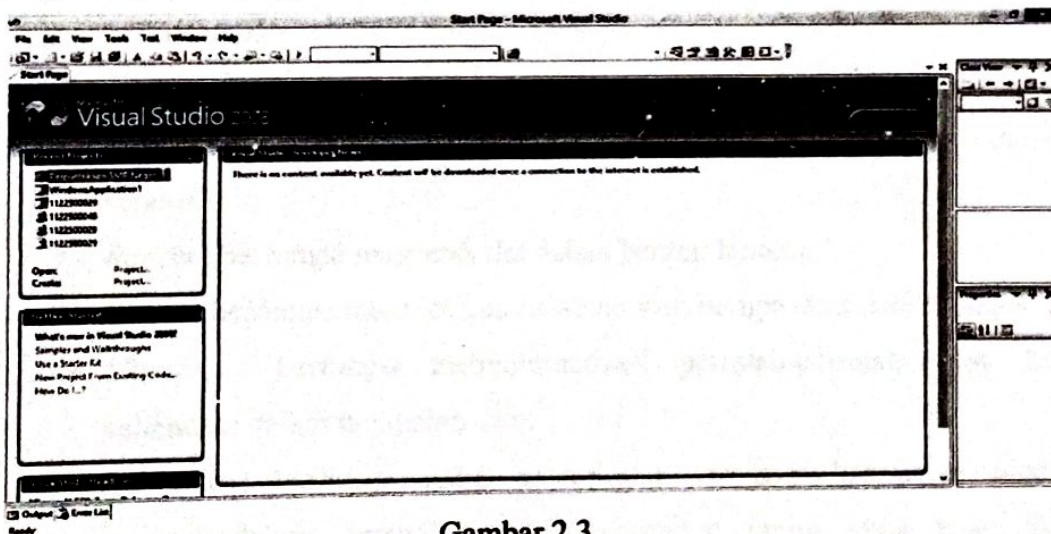
## 2.3 Perangkat Lunak

Perangkat lunak yang digunakan untuk merancang Sistem Informasi pengolahan nilai pada SMP N 5 Sungailiat yaitu:

### 2.3.1 Visual Basic NET 2008

Menurut wahana komputer (2007 : 1), Visual Basic NET merupakan salah satu bahasa pemrograman yang dapat digunakan untuk membuat program aplikasi. Bahasa pemrograman ini menyediakan beberapa *tool* untuk otomatisasi proses pengembangan, yaitu *visual tool* yang digunakan untuk melakukan beberapa operasi pemrograman dan desain umum dan juga fasilitas-fasilitas lain yang dapat menunjang dalam pemrograman.

Visual Basic NET merupakan bagian dari Visual Studio.NET. Visual Studio.NET merupakan suatu lingkungan (*Environment*) terintegrasi untuk membangun dan melakukan uji coba (*Testing and Debugging*) berbagai macam aplikasi. Diantaranya adalah aplikasi Windows, Web, Control, Class serta aplikasi console. Dengan Visual Studio NET, anda akan dapat lebih mudah membuat aplikasi karena dalam Visual Studio NET ada dukungan fasilitas baru yang ditambahkan antara lain *Integrated Development Environment (IDE)*, *Microsoft Intellisense*, *debugging* yang lebih baik dan kemampuan dalam *XML Web Services*. Dengan Visual Basic NET, user dapat mendesain, mengeksekusi dan men-debug program aplikasi yang telah dibuat.



Gambar 2.3  
Visual Basic NET 2008



### 2.3.2 Microsoft Access

Microsoft Office Access adalah sebuah program aplikasi pengelolaan database secara elektronik yang memungkinkan disusunnya data dan informasi yang banyak secara sistematis dan disimpan/direkam kedalam sebuah komputer.

Microsoft Office Access mempunyai banyak keunggulan, diantaranya yaitu:

- a. Mempunyai semua fungsi fasilitas yang dimiliki oleh software database klasik seperti dBase dan Foxpro. Hal ini berarti jika user berpindah dari aplikasi database klasik ke Microsoft Office Access, maka tidak ada masalah dengan segala fungsi/fasilitas yang pernah digunakan dengan aplikasi database yang terdahulu.
- b. Merupakan program aplikasi database yang sangat mudah digunakan. Seorang user harus bisa memahami teknik-teknik pemrograman agar bisa bekerja dengan optimal. Dengan Microsoft Office Access, User tersebut akan merasa jauh lebih mudah dalam membuat sebuah sistem informasi tanpa mengurangi kualitas program tersebut.
- c. Komponen-komponen yang terdapat pada Microsoft Office Access adalah sebagai berikut:
  - 1) *Tables* : merupakan kumpulan data sebagai komponen utama dalam database.
  - 2) *Queries* : berfungsi menyaring data dari berbagai kriteria data urutan yang dikehendaki.
  - 3) *Forms* : berfungsi memasukkan data, menampilkan data serta mengedit data dari suatu tabel dengan tampilan formulir yang telah kita rancang sendiri.
  - 4) *Report* : berfungsi mencetak data dalam bentuk laporan.
  - 5) *Pages* : berfungsi menciptakan halaman web berupa *data access pages*.
  - 6) *Macros* : berfungsi mengotomatisasi perintah-perintah yang kita kehendaki dalam mengolah data:
  - 7) *Modules* : berfungsi untuk merancang bermacam-macam modul aplikasi dalam mengolah database tingkat lanjut yang telah kita kehendaki.

### **2.3.3 Crystal Reeport**

Crystal Report merupakan program yang dapat digunakan untuk membuat, menganalisa dan menerjemahkan informasi yang terkandung dalam database atau program kedalam berbagai jenis laporan yang sangat fleksibel. Beberapa kelebihan dari Crystal Report adalah :

- a. Pembuatan laporannya tidak terlalu rumit sehingga memungkinkan pemrogram pemula sekalipun untuk membuat laporan tanpa harus melibatkan banyak kode pemrograman.
- b. Terintegritas dengan berbagai bahasa pemrograman lain sehingga memungkinkan pemrogram memanfaatkannya dengan keahlinya sendiri-sendiri.
- c. Fasilitas import hasil laporan yang mendukung format yang sudah populer seperti Microsoft Word, Excel, Access, Adobe Acrobat Reader, HTML, dan sebagainya. Elemen layar Crystal Report tidak jauh dengan elemen layar Data Report (salah satu fasilitas default yang disediakan Visual Basic untuk membuat laporan). Hanya saja Crystal Report dilengkapi dengan 12fasilitas yang lebih banyak untuk mengembangkan berbagai jenis laporan.  
Pada umumnya sebuah laporan sedikitnya terdiri dari lima bagian (section) utama, yaitu:
  1. Report Header yang berisi informasi yang hanya akan terlihat sekali, pada awal laporan, misalnya judul dan tanggal laporan.
  2. Page Header yang berisi informasi yang akan terlihat diatas setiap halaman laporan, seperti label heading kolom.
  3. Details yang berisi informasi yang akan terlihat satu kali setiap record dalam tabel atau query yang terkait dengan laporan.
  4. Page Footer yang berisi informasi yang akan terlihat didasar setiap halaman laporan, seperti nomor halaman.
  5. Report Footer yang berisi informasi yang akan terlihat hanya sekali diakhir laporan, seperti ringkasan atau rata-rata yang ada diakhir laporan.

## **2.4 Teori Pendukung**

### **2.4.1 Pengertian Nilai**

Nilai adalah konsepsi abstrak tentang sesuatu yang berharga dalam diri manusia mengenai apa yang dianggap baik dan apa yang dianggap buruk (Saptina dkk. 2010: 7). Sedangkan Nursal Luth dan Daniel Fernandes mengatakan bahwa nilai adalah perasaan-perasaan tentang apa yang diinginkan atau tidak diinginkan yang mempengaruhi perilaku social dari orang yang memiliki nilai itu. Nilai bukanlah soal benar atau salah, tetapi soal dikehendaki atau tidak, disenangi atau tidak, nilai merupakan kumpulan sikap dan perasaan-perasaan yang selalu diperhatikan melalui perilaku oleh manusia. Dari beberapa pengertian nilai diatas, dapat dipahami bahwa nilai adalah kualitas ketentuan yang bermakna bagi kehidupan manusia perorangan, masyarakat, bangsa, Negara.

#### **a. Ciri-ciri Nilai:**

##### **1) Nilai-nilai yang mendarah daging (internalized value)**

Yaitu nilai yang telah menjadi kepribadian bawah sadar atau yang mendorong timbulnya tindakan tanpa berfikir lagi. Bila dilanggar maka akan timbul perasaan malu atau bersalah yang mendalam dan sukar dilupakan, misalnya orang yang taat beragama akan menderita beban mental apabila, melanggar salah satu norma yang ada dalam agamanya.

##### **2) Nilai yang Dominan**

Merupakan nilai yang dianggap lebih penting dari pada nilai-nilai lainnya.

Tampak pada pilihan yang dilakukan seseorang pada waktu berhadapan dengan beberapa alternative tindakan yang harus diambil. Beberapa pertimbangan dominan atau tidaknya nilai tersebut adalah sebagai berikut.

- a) Banyaknya orang yang menganut nilai tersebut
- b) Lamanya nilai tersebut dirasakan anggota kelompok tertentu.
- c) Tingginya usaha untuk mempertahankan nilai itu.
- d) Tingginya kedudukan (prestige) orang-orang yang membawakan.

**b. Macam-macam nilai**

Nilai berhubungan erat dengan budaya dan masyarakat. Menurut prof Dr. Notonegoro (2006), nilai dibagi menjadi 3 bagian yaitu:

- 1) Nilai material, yaitu segala sesuatu yang berguna bagi manusia.
- 2) Nilai vital, yaitu segala sesuatu yang berguna manusia untuk dapat mengadakan aktivitas.
- 3) Nilai kerohanian, yaitu segala sesuatu yang berguna bagi jiwa/rohani manusia. Nilai kerohanian dapat dibagi atas 4 macam, yaitu:
  - a) Nilai kebenaran atau kenyataan yang bersumber dari undur akal manusia.
  - b) Nilai keindahan yang bersumber dari unsure rasa manusia.
  - c) Nilai moral/kebaikan yang berunsur dari kehendaki/kemauan.
  - d) Nilai religious, yaitu merupakan nilai ketuhanan, kerohanian yang tinggi dan mutlak yang bersumber dari keyakinan/kepercayaan manusia.

**2.4.2 Raport**

Raport berasal dari kata report yang berarti laporan. Raport merupakan laporan hasil dari suatu kegiatan yang disusun secara benar. Materi yang dilaporkan dalam hal ini adalah hasil ulangan harian, tugas harian, ujian tengah semester, ujian akhir semester, kepribadian, ekstrakurikuler dan beserta data yang diperlukan yang berkaitan dengan raport.