

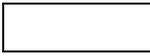
## BAB II

### LANDASAN TEORI

#### 2.1 *Entity Relationship Diagram*

*Entity Relationship Diagram* (ERD) merupakan teknik yang digunakan untuk memodelkan kebutuhan data dari suatu organisasi, biasanya oleh *System Analysts* dalam tahap analisis persyaratan proyek pengembangan sistem. Sementara seolah-olah teknik diagram atau alat peraga memberikan dasar untuk desain database relasional yang mendasari sistem informasi yang dikembangkan. ERD bersama-sama dengan detail pendukung merupakan model data yang pada gilirannya digunakan sebagai spesifikasi untuk database. (Brady dan Loonam : 2010)

Tabel 2.1 *Entity Relationship Diagram* (ERD)

No	Simbol	Keterangan
1		Entitas
2		Atribut
3		Hubungan
4		Garis

##### 1. Entitas

Objek dalam dunia nyata yang dapat dibedakan dengan objek lain. Entitas terdiri atas beberapa atribut mengidentifikasi atau membedakan yang satu dengan yang lainnya. Pada setiap entitas

baru harus memiliki 1 atribut unik atau yang disebut dengan *primary key*.

## 2. Atribut

Isi dari atribut mempunyai elemen yang dapat mengidentifikasi isi elemen satu dengan yang lain. Ada dua jenis atribut, yaitu:

- a. *Identifier (key)* digunakan untuk menentukan suatu *entity* secara unik (*primary key*).
- b. *Descriptor (nonkey attribute)* digunakan untuk menspesifikasi karakteristik dari suatu *entity* yang tidak unik.

## 3. Kardinalitas

Menyatakan jumlah himpunan relasi antar entitas. Pemetaan kardinalitas terdiri dari:

- a. One-to-one, sebuah entitas pada A berhubungan dengan entitas B paling banyak.
- b. One-to-many, sebuah entitas pada A berhubungan dengan entitas B lebih dari satu.
- c. Many-to-many, sebuah entitas pada A berhubungan dengan entitas B lebih dari satu dan entitas B berhubungan dengan entitas A lebih dari satu juga.

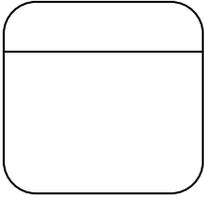
### 2.2 *Data Flow Diagram*

*Data Flow Diagram* (DFD) atau dalam bahasa Indonesia menjadi Data Alir Diagram (DAD) adalah representasi grafis yang menggambarkan

aliran informasi dan tranformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*). (S, Rosa A & M, Shalahuddin, 2011)

Adapun simbol-simbol yang digunakan untuk menggambarkan *Data Flow Diagram* (DFD) adalah sebagai berikut:

Tabel 2.2 *Simbol Data Flow Diagram* (DFD)

Simbol	Keterangan
	<p><b>Proses</b>, kegiatan yang dilakukan oleh orang, mesin atau komputer dari hasil suatu arus data yang dapat berupa masukan untuk sistem.</p>
	<p><b>External Entity</b>, menyatakan kesatuan di lingkungan luar sistem yang dapat berupa orang, organisasi atau sistem yang lainnya, yang berada di lingkungan luarnya yang akan memberikan input atau menerima output dari sistem.</p>
	<p><b>Simpanan Data</b>, simpanan dari data yang dapat berupa suatu file atau data base di sistem komputer, suatu arsip atau catatan maual dan suatu agenda atau buku.</p>
	<p><b>Arus Data</b>, menyatakan arus data yang mengalir diantara data proses, simpan data dan kesatuan luar. Arus data ini menunjukkan arus dari data yang dapat berupa masukan untuk sistem atau hasil dari proses sistem.</p>

Adapun beberapa tahapan dalam pembuatan *Data Flow Diagram* adalah:

1. Diagram level konteks, digunakan untuk menggambarkan sistem secara global.
2. Diagram level nol, digunakan untuk menggambarkan tahapan-tahapan proses data dalam diagram konteks.
3. Diagram detail (level), digunakan untuk menggambarkan arus data yang lebih mendetail dalam proses diagram level nol.

### 2.3 Sistem Pendukung Keputusan

Menurut Raul Valverde (2011), sebuah sistem pendukung keputusan/penunjang keputusan yang digambarkan sebagai sistem interaktif berbasis komputer yang dirancang untuk membantu para pengambil keputusan untuk memecahkan masalah tak terstruktur. Menggunakan kombinasi dari model teknik analisis dan pengambilan informasi sistem seperti membantu mengembangkan dan mengevaluasi atau menjadi sistem alternatif yang benar – benar sesuai.

#### a. Konsep SPK

SPK adalah sistem yang dapat dikembangkan, mampu mendukung analisis data dan pemodelan keputusan, berorientasi pada perencanaan serta tidak bisa direncanakan interval (periode) waktu pemakaiannya. SPK terdiri dari 3 komponen yang berinteraksi satu dengan yang lainnya, yaitu :

- 1) *Language System*, adalah suatu mekanisme untuk menjembatani pemakai dan komponen lainnya.

- 2) *Knowledge System*, adalah *repository* pengetahuan yang berhubungan dengan masalah tertentu baik berupa data maupun prosedur.
- 3) *Problem Processing System*, adalah sebagai penghubung kedua komponen lainnya, berisi satu atau beberapa kemampuan manipulasi atau menyediakan masalah secara umum yang diperlukan dalam pengambilan keputusan.

b. Karakteristik SPK

Karakteristik dari Sistem Pendukung Keputusan yang membedakan dari sistem lainnya adalah:

- 1) SPK dirancang untuk membantu pengambilan keputusan dalam memecahkan masalah yang sifatnya semi terstruktur maupun tidak terstruktur.
- 2) Dalam proses pengolahannya, SPK mengkombinasikan menggunakan model-model analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari.
- 3) SPK dirancang sedemikian rupa sehingga dapat digunakan dengan mudah oleh orang-orang yang tidak memiliki dasar kemampuan pengoperasian komputer yang tinggi.
- 4) SPK dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi. Sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi dan kebutuhan pemakai.

c. Keuntungan Sistem Pendukung Keputusan

Keuntungan dari Sistem Pendukung Keputusan antara lain:

- 1) Memperluas kemampuan pengambilan keputusan dalam memproses data/informasi bagi pemakainya.
- 2) Membantu pengambilan keputusan dalam hal penghematan waktu yang dibutuhkan untuk memecahkan masalah terutama berbagai masalah yang sangat kompleks dan tidak terstruktur.
- 3) Dapat menghasilkan solusi dengan lebih cepat serta hasilnya dapat diandalkan.

**2.4 Metode *Simple Multi Attribute Rating Technique (SMART)***

Metode *SMART (Simple Multi Attribute Rating Technique)* merupakan metode pengambilan keputusan multi atribut yang dikembangkan oleh Edward pada tahun 1977. Metode SMART merupakan teknik pengambilan keputusan multi kriteria ini berdasarkan pada teori bahwa setiap alternatif terdiri dari sejumlah kriteria yang memiliki nilai-nilai dan setiap kriteria memiliki bobot yang menggambarkan seberapa penting dia dibandingkan dengan kriteria lain. Pembobotan ini digunakan untuk menilai setiap alternatif agar diperoleh alternatif terbaik.

Teknik pembuatan keputusan multi atribut ini digunakan untuk mendukung pembuat keputusan dalam memilih antara beberapa alternatif. Setiap pembuat keputusan harus memilih sebuah alternatif yang sesuai dengan tujuan yang telah dirumuskan. Setiap alternatif terdiri dari sekumpulan atribut dan setiap atribut mempunyai nilai-nilai. Nilai ini dirata-rata dengan skala tertentu. Setiap atribut mempunyai bobot yang

menggambarkan seberapa penting dibandingkan dengan atribut lain. Pembobotan dan pemberian alternatif terbaik. Pembobotan pada *SMART* (*Simple Multi Attribute Rating Technique*) menggunakan skala antara 10 sampai 100. Sehingga mempermudah perhitungan dan perbandingan nilai pada masing-masing alternatif. Model yang digunakan dalam *SMART* (*Simple Multi Attribute Rating Technique*) yaitu:

$$u(a_i) = \sum_{j=1}^m W_j U_j(a_i), \quad i=1,2,\dots,m \dots\dots\dots(1)$$

Keterangan:

$w_j$  = nilai pembobotan kriteria ke-j dan  $k$  kriteria

$u(a_i)$  = nilai *utility* kriteria ke-i untuk kriteria ke-i

Pemilihan keputusan adalah mengidentifikasi mana dari  $n$  alternatif yang mempunyai nilai fungsi terbesar.

Teknik Metode SMART :

Langkah 1 : Tentukan kriteria yang digunakan dalam pemilihan siswa berprestasi

Langkah 2 : Tentukan alternatif siswa yang berprestasi.

Langkah 3 : Melakukan perbandingan terhadap kedudukan kepentingan kriteria.

Langkah 4 : Memberikan bobot berdasarkan kriteria paling penting dan kriteria paling rendah tingkat kepentingannya dengan skala 10-100.

Langkah 5 : Mencari nilai rata-rata bobot, berdasarkan prioritas yang telah diinputkan kemudian dilakukan normalisasi.

Normalisasi =

$$\frac{w_j}{\sum w_j} \dots\dots\dots(2)$$

Keterangan :

$w_j$  : bobot suatu kriteria

Langkah 6 : Memberikan nilai kriteria untuk setiap alternatif.

Langkah 7 : Hitung nilai utility untuk setiap kriteria masing-masing.

Langkah 8 : Hitung nilai akhir masing-masing.

## 2.5 My SQL

MySQL adalah sebuah *server database open source* yang digunakan sebagai aplikasi terutama untuk *server*. MySQL sendiri merupakan sebuah implementasi dari sistem manajemen basis data relasional yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan

batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. SQL (Structured Query Language) adalah sebuah konsep pengoperasian basisdata, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. (Prasetyo, 2009)

## 2.6 Visual Basic .Net

Microsoft Visual Basic .Net adalah visual basic yang direkayasa kembali untuk digunakan pada *platform.NET* sehingga aplikasi yang dibuat menggunakan *visual basic.NET* dapat berjalan pada sistem komputer apapun, dan dapat mengambil data dari server dengan tipe apapun asalkan terinstal *.NET framework*. (Priyanto, 2014)

Berikut ini perkembangan *visual basic .NET*

- a. *Visual basic .NET 2002 (VB 7.0)*
- b. *Visual basic .NET 2003 (VB 7.1)*
- c. *Visual basic 2005 (VB 8.0)*
- d. *Visual basic 2008 (VB 9.0)*
- e. *Visual basic 2010 (VB 10.0)*
- f. *Visual basic 2012 (VB 11.0)*
- g. *Visual basic 2013*

*Visual Studio 2008* sebuah *software* buatan *microoft* yang berfungsi untuk membuat program berbasis *Desktop* maupun berbasis *Web*, di *Visual Studio .Net* ini selain ada *Vb.net* terdapat juga bahasa pemrograman lainnya seperti *C#.Net*, *ASP.Net* dan yang lainnya dengan adanya aplikasi ini proses pengerjaan Aplikasi Penetapan Angka Kredit ini menjadi lebih singkat dan

mempermudah *user* yang sudah mengerti akan bahasa pemrograman *VB.net* maupun *VB*. Dengan adanya *Visual Studio 2008* ini para programmer bisa mudah membuat sebuah aplikasi berbasis *desktop* maupun aplikasi berbasis *web*, *Visual Studio 2008*.

Mempunyai MSDN yang berfungsi sebagai alat bantu jika ketika kita mempunyai kesulitan dalam pembuatan *coding* maka MSDN bisa membantu akan codingan yang sedang kita buat sehingga para *user* yang membuat aplikasi *desktop* maupun aplikasi *web* menjadi lebih mudah dengan adanya MSDN.

Pada pemrograman *Visual*, pengembangan aplikasi dimulai dengan pembentukan *user interface*, kemudian mengatur properti dari objek – objek yang digunakan dalam *user interface*, dan baru dilakukan penulisan kode program untuk menangani kejadian – kejadian (*event*). Tahap pengembangan aplikasi demikian dikenal dengan istilah pengembangan aplikasi dengan pendekatan *Bottom Up*.

## 2.7 Pengujian Mccall

Model ini merupakan model kualitas tertua yang dikembangkan pada tahun 1976. Model ini digunakan dengan tujuan agar sebuah kualitas dapat diukur secara eksplisit dengan menjelaskan 11 faktor kualitas atau karakteristik yang memiliki pengaruh penting terhadap kualitas. Faktor-faktor tersebut antara lain:

(sumber dari: [http://mkti-si-its.blogspot.co.id/2013/03/model-mccall\\_14.html](http://mkti-si-its.blogspot.co.id/2013/03/model-mccall_14.html))

1. Product Revision (Kemampuan untuk mengalami perubahan)
  - *Maintainability*: Usaha yang dibutuhkan untuk menempatkan dan menyelesaikan kesalahan program dalam lingkungan operasi.
  - *Flexibility*: Kemudahan membuat perubahan yang dibutuhkan oleh perubahan dalam lingkungan pengoperasian
  - *Testability*: Kemudahan pengujian program, untuk meyakinkan bahwa program bebas error dan memenuhi spesifikasinya.
2. Product Transition (Beradaptasi pada lingkungan yang baru)
  - *Portability*: Usaha yang dibutuhkan untuk memindahkan program dari satu lingkungan ke lingkungan lain
  - *Reusability*: Kemudahan penggunaan kembali software dalam konteks yang berbeda.
  - *Interoperability*: Usaha yang dibutuhkan untuk memasang sistem dengan sistem yang lain
3. Product Operations (Karakteristik pengoperasian)
  - *Correctness*: Tingkatan dimana software mencapai spesifikasinya
  - *Reliability*: Kemampuan sistem untuk tidak gagal
  - *Efficiency*: Efisien dalam eksekusi dan penyimpanan. Umumnya diartikan penggunaan resource, seperti: waktu, processor, memori, dsb.

- *Integrity*: Perlindungan program dari akses yang tidak berkepentingan
- *Usability*: Kemudahan penggunaan software