

BAB II

TINJAUAN PUSTAKA

II.1. Penelitian Berdasarkan Sistem Pendukung Keputusan

Penelitian yang di bahas oleh Egi Badar Sambani (2016), Pengambilan Keputusan dan kelayakan penerimaan pengajar karena adanya perbedaan sudut pandang di antara keduanya terhadap personality yang memiliki calon pengajar terutama jika beberapa calon pengajar memiliki kemampuan (dan beberapa pertimbangan lain) yang tidak jauh berbeda. Pengajar memiliki kemampuan (dan beberapa pertimbangan lain) yang tidak jauh berbeda. Dengan adanya sistem pendukung keputusan melalui penerapan metode profile matching di harapkan dapat meningkatkan efektivitas dan objektivitas dalam membantu pemberian keputusan kelayakan penerimaan pengajar.

Heru Purwanto (2017), Dalam penelitian Metode Profile Matching menjadi pilihan guna memberikan penilaian dan evaluasi kinerja karyawan untuk dipromosikan untuk menjadi leader. Terutama dalam kegiatan organisasi dan manajemen sumber daya manusia yang baik akan berdampak positif untuk perkembangan dan kemajuan organisasi.

Andri Anto Tri Susilo (2017), Proses pemilihan ketua program study biasanya di tunjuk langsung oleh ketua yayasan. Kekurangan dari proses penunjukan langsung adalah tidak bisa melihat kemampuan yang tidak di miliki oleh calon ketua program study baik itu miliki oleh calon ketua program study baik itu kemampuan social, SDM, pembelajaran dan keuangan. Hasil akhir dari

penelitian ini adalah di dapatkan informasi peringkat dari setiap calon ketua program studi dengan di dasarkan pada perhitungan profile matching.

Sary Fatimah (2017), Salah satu metode yang digunakan untuk calon beasiswa karyawan adalah penerapan Profile Matching karena mampu menyeleksi alternative terbaik. Hasil perengkingan yang ada akan memiliki nilai Profile Matching tertinggi yang mendapatkan beasiswa seta penerapan Metode Profile Matching kedalam aplikasi, perengkingan karyawaan sesuai kriteria yang telah di tentukan.

II.2. Uraian Teoritis

II.2.1 Definisi Sistem

Defenisi sistem adalah sekelompok komponen dan elemen yang di gabungkan menjadi satu untuk mencapai tujuan tertentu. Pengertian sistem tergantung pada cara pandang dalam mendefensikan nya. Ada beberapa pengertian sistem yaitu :

a. Menurut Hukum

Kumpulan aturan – aturan yang membatasi baik oleh kapasitas hukum itu sendiri maupun lingkungan dimana sistem itu berada, untuk menjamin kseserasian dan keadilan.

b. Menurut Rekayasa

Proses masukan input yang di transformasikan menjadi keluaran atau output tertentu.

c. Menurut Schmidt & Taylor

Adalah suatu kumpulan komponen atau entity yang berinteraksi dan bereaksi. Untuk mencapai tujuan akhir yang logis.

d. Menurut Matematika

Suatu susunan persamaan – persamaan simbolik dengan karakteristik tertentu.

e. Menurut Masyarakat Umum

Cara atau metode untuk mencapai suatu tujuan

f. Menurut Geoffrey Gordin

Suatu agregasi atau kumpulan objek – objek yang terangkai dalam interaksi dan saling ketergantungan yang teratur.

g. Menurut John Burch

Suatu kumpulan dari objek objek dan ide yang saling berhubungan dan di perintahkan untuk mencapai sasaran dan tujuan bersama.

II.2.2. Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses, dan output. Hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. (Tata Sutabri : 2012).

Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen sistem (*component*)

Suatu sistem terdiri dari sejumlah komponen yang berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Suatu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra sistem.

2. Batasan sistem (*Boundary*)

Ruang lingkup sistem merupakan daerah yang membatasi antara sistem dengan sistem lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem di pandang sebagai satu kesatuan yang tidak dapat di pisah-pisahkan.

3. Lingkungan Luar Sistem (*Environment*)

Bentuk apapun yang ada diluar lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut disebut dengan lingkungan luar sistem. Lingkungan luar sistem ini dapat menguntungkan dan dapat juga merugikan sistem tersebut. Lingkungan luar yang menguntungkan merupakan energi bagi sistem tersebut, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar yang merugikan harus dikendalikan. Kalau tidak maka akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (*Interface*)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem interface. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem yang lain. Keluaran suatu subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi sistem yang membentuk satu kesatuan.

5. Masukan Sistem (*Input*)

Energi yang dimasukkan kedalam sistem tersebut masukan sistem, yang dapat berupa pemeliharaan (*maintenance input*) dan sinyal (*signal input*). Sebagai contoh didalam suatu unit sistem komputer, “program” adalah *maintenance input* yang digunakan untuk mengoperasikan komputer. Sementara “data” adalah signal yang akan diolah menjadi informasi.

6. Keluaran Sistem (*Output*)

Hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan bagi subsistem yang lain. Seperti contoh sistem informasi, Keluaran yang dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolahan Sistem (*Procces*)

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, sistem akuntansi, sistem ini akan mengolah

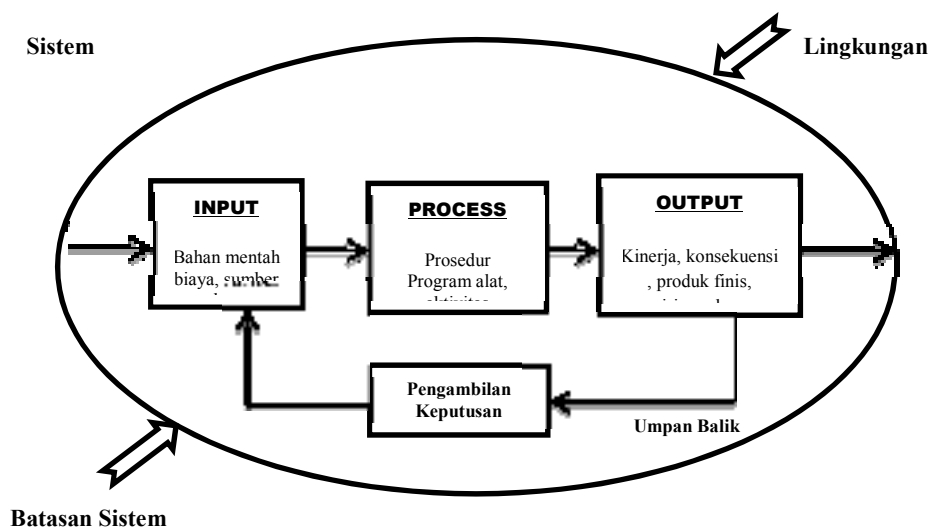
data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem (*Objective*)

Suatu sistem memiliki tujuan dan sasaran yang pasti dan bersifat deterministik. Kalau suatu sistem tidak memiliki sasaran, maka operasi sistem tidak ada gunanya. Suatu sistem dikatakan berhasil bila mengenai sasaran atau tujuan yang telah direncanakan.

II.2.3. Perancangan Sistem

Perancangan sistem yaitu merancang output, input, proses, program prosedur, perangkat keras dan perangkat lunak yang diperlukan untuk mendukung suatu sistem informasi. Perancangan sistem menurut Blanchard dan Fabrycky dalam buku *System Engineering and Analysis* (1989) menjelaskan bahwa istilah daur hidup sistem sering juga dikenal sebagai *consumer to consumer cycle* maksudnya berawal dari pengenalan kebutuhan konsumen terhadap sistem dan berakhir pada penolakan konsumen terhadap sistem tersebut setelah menggunakan sistem sekian lama. Pendekatan daur hidup sistem berawal dari identifikasi kebutuhan yang menjadi alasan mengapa sistem baru atau perbaikan terhadap sistem yang ada sekarang perlu dilakukan. Pada umumnya perlu sistem baru dikembangkan dan sistem lama diperbaiki berhubungan dengan adanya masalah yang sedang atau diperkirakan akan dihadapi. Penjelasan perancangan sistem di atas dapat kita lihat pada Gambar II.1:



Gambar II.1 : Perancangan sistem dan Lingkungan

Berikut di bawah ini penjelasan dari Gambar II.1 yaitu :

- Input adalah elemen yang masuk kedalam sistem input data halaman web untuk query database.
- Proses adalah sebuah elemen yang di perlukan untk mengkonversi atau mentransformasikan input ke dalam output. Pada sebuah Komputer termasuk berbasis web, sebuah proses dapat meliputi perintah – perintah pengaktifan, eksekusi komputasi, dan penyimpanan informasi.
- Output adalah produk finis atau konsekuensi yang ada pada sistem. Server web dapat menghasilkan sebuah halaman web dinamis berdasarkan input dan proses nya.
- Umpan balik adalah aliran informasi dari komponen output ke pengambil keputusan berkenaan dengan output atau perfoma sistem.

- Lingkungan Sistem terdiri dari beberapa elemen yang ada di luar, dalam hal ini bukan lah input, output atau proses. Akan tetapi mempengaruhi performa sistem dan konsekuensi pencapaian tujuan sistem.
- Batasan Sistem biasanya di tentukan dengan mempersempit lingkup sistem untuk mempermudah analisis. Dengan kata lain, batasan sebuah sistem pendukung keputusan berdasarkan desain.

II.3. Sistem Pendukung Keputusan

Decision Support System atau sistem pendukung keputusan didefinisikan juga sebuah sistem yang mampu memberikan kemampuan, baik kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah semi terstruktur. Secara khusus, SPK di defenisikan sebagai sebuah sistem yang mendukung kerja seseorang manager maupun sekelompok manager dalam memecahkan masalah semi terstruktur dengan cara memberikan informasi ataupun usulan menuju pada keputusan tertentu.

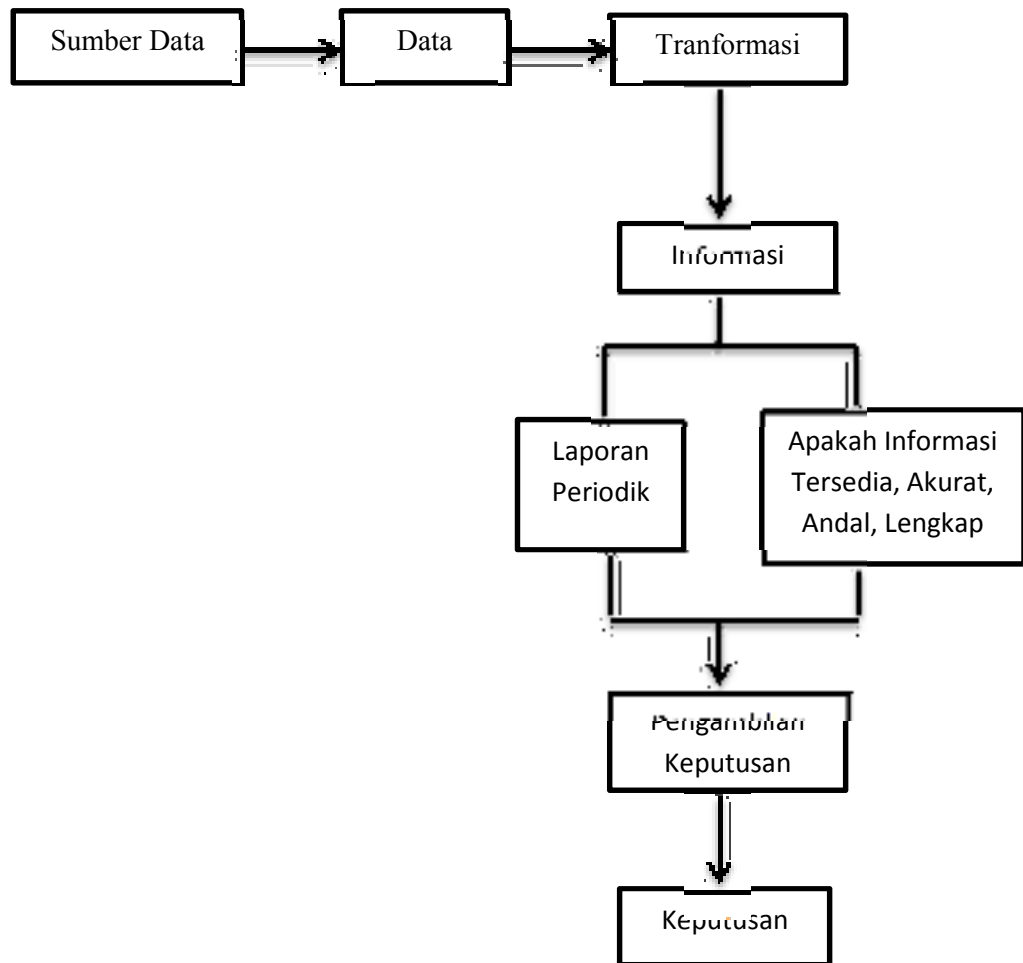
Konsep DSS atau (sistem pendukung keputusan) pertama kali di perkenalkan pada awal tahun 1970-an oleh Michael Scott Morton, yang selanjutnya dengan istilah “Management Decition System”.Konsep DSS merupakan sebuah sistem interaktif berbasis computer yang membantu pembuatan keputusan memanfaatkan data untuk menyelesaikan masalah – masalah yang bersifat tidak tersetruktur dan terstruktur. DSS di rancang untuk menunjang seluruh tahapan pembuatan keputusan, yang di mulai dari tahapan mengidentifikasi masalah, memilih data yang relevan, menentukan pendekatan

yang di gunakan dalam proses pembuatan keputusan sampai pada kegiatan mengevaluasi pemilihan alternative. (Nina Sherly 2013)

Sistem pendukung keputusan adalah bagian dari sistem informasi berbasis computer termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang di pakai untuk mendukung pengambilan keputusan dalam organisasi. Konsep sistem pendukung keputusan di tandai dengan sistem interaktif berbasis computer yang membantu pengambil keputusan memanfaatkan data dan model keputusan untuk menyelesaikan masalah – masalah yang tidak terstruktur. Ketajamaan keputusanyang di hasilkan yang di pengaruhi oleh kelengkapan dank ke akuratan informasi yang di libatkan di dalam proses pengambilan keputusan itu sendiri. Maka peranan sistem informasi sangat penting dalam menyediakan informasi pendukung keputusan (Danis novita perdana,yuni yama sari : 2014)

Jenis Informasi yang dibutuhkan untuk mengambil keputusan ada tiga yaitu:

1. Informasi untuk pengendalian dan pemanfaatan sumber daya yang di miliki.
2. Laporan efisiensi pelaksanaan dan pengelolaan.
3. Laporan efektifitas pencapaian tujuan sebagai fungsi penggunaan sumber



Gambar II.2 : Proses Pengambilan Keputusan

II.3.1. Karakteristik SPK

Sistem pendukung keputusan merupakan sumber intelektual dari individu – individu dengan computer untuk memperbaiki kualitas dari suatu keputusan.

1. Memfokuskan diri dari pada keputusan – keputusan yang tidak terstruktur dan semi terstruktur.
2. Fleksible pada perubahan – perubahan keperluan
3. Mudah untuk di pergunakan.
4. Komputer harus mendukung manager, tetapi tidak menggantikan kebijaksanaan nya.
5. Dukungan utama di berikan untuk masalah yang tidak terstruktur, masalah dapat di analisis secara matematis oleh computer, tetapi kebijaksanaan manjer juga di perlukan untuk proses control.
6. Fleksibel terhadap spesifikasi keperluan keluaran.
7. Memberikan respon yang cepat.
8. Di fokuskan pada Top Managemen.

II.3.2. Tingkatan SPK

Dalam menjalankan fungsinya sebagai suatu Sistem Pendukung Keputusan (SPK) dibagi menjadi tiga tingkatan. Ketiga tingkatan ini dibagi menurut type computer, *hardware*, *software*, dan hirarki penerapan SPK itu sendiri. Ketiga tingkatan itu adalah :

1. SPK Spesifik

Sistem ini mencakup data sistem informasi “aplikasi” namun dengan karakteristik yang berbeda jelas dengan aplikasi pengolah data yang lain. SPK Spesifik adalah *hardware/ software* yang memungkinkan pembuat keputusan spesifik berinteraksi dengan kumpulan hubungan masalah yang spesifik.

2. SPK Generator

SPK Generator merupakan paket dari kumpulan *hardware/ software* yang menyediakan sekumpulan kemampuan untuk membuat SPK Spesifik dengan cepat dan mudah.

3. SPK Tools

SPK Tools merupakan elemen – elemen *hardware/ software* yang mana dapat dipergunakan untuk mengembangkan SPK spesifik maupun SPK Generator. Meskipun SPK *Tools* ini pun mampu membuat SPK Spesifik secara langsung, namun mengembangkan SPK spesifik dengan SPK Generator jauh lebih mudah dan efisien.

II.3.3. Tahapan Proses SPK

Tiga tahapan dalam proses pengambilan keputusan :

- a. Tahap *Intelligen* , adalah tahap proses pengenalan persoalan melalui penyelidikan lingkungan untuk mengetahui ada atau tidaknya masalah. Kesimpulan dari penyelidikan diperoleh dari pengolahan data dengan metode yang telah ditetapkan sebelumnya atau dengan metode khusus. Aliran informasi bergerak dari tingkatan manajemen terendah menuju tingkatan manajemen tertinggi.
- b. Tahap *Design*, merupakan tahap mencari, analisis serta perumusan alternative tindakan yang akan diambil. Pada tahap *design ini*, sistem informasi harus mampu membuat keputusan – keputusan.
- c. Tahap *Choice*, merupakan tahap memilih suatu tindakan yang paling tepat dari beberapa alternative yang telah dirumuskan. Langkah selanjutnya adalah pelaksanaan alternatif terpilih. Bila suatu alternatif telah dilaksanakan, fungsi informasi berubah menjadi pengumpul data untuk selanjutnya, merupakan umpan balik.

II.4. Metode Profile Matching

Profile Matching merupakan suatu proses yang sangat penting dalam manajemen SDM, dimana terlebih dahulu di tentukan kompetensi yang di perlukan oleh suatu posisi. Kemampuan tersebut haruslah dapat di penuhi oleh setiap calon kandidat. Dalam proses Profile Matching secara garis besar merupakan proses membandingkan antara kompetensi individu kedalam kompetensi posisi sehingga

dapat di ketahui perbedaan kompetensinya atau GAP. Semakin kecil GAP yang di hasilakan maka bobot nilainya semakin besar yang berarti memiliki peluang besar untuk menempati posisi tersebut.(Danis Novita Perdana, Yuni Yamasari : 2014)

II.4.1. Langkah – langkah pada metode Profile Matching

Menurut (Nina Sherly : 2013) adapun langkah – langkah dalam penyelesaian metode *Profile Matching* adalah sebagai berikut :

1. Perhitungan Pemetaan Gap Kompetensi.

Yang di maksud Gap adalah beda antara profile beasiswa karyawan dengan profile ideal yang di harapkan atau dapat di tujukan pada rumus di bawah ini:

$$\text{Gap} = \text{Profil Karyawan} - \text{Profile Beasiswa Karyawan}$$

2. Perhitungan Pemetaan Gap Kompetensi berdasarkan Kriteria

Kompetensi berdasarkan Kriteria untuk pengumpulan gap – gap yang terjadi itu sendiri pada tiap aspeknya mempunyai perhitungan yang berbeda beda setelah di dapatkan tiap gap masing – masing karyawan maka tiap profile karyawan di beri bobot nilai dengan menentukan variabel – variabel pemetaan Gap kompetensi menentukan Kriteria yang di gunakan dalam memproses nilai karyawan.

3. Pembobotan

Setelah di peroleh Gap pada masing – masing calon penerima beasiswa karyawan di beri bobot nilai dengan ketentuan pada tabel Gap. Ketentuan bobot nilai gap dapat di lihat pada Tabel II.1

Tabel II.1 :Bobot Nilai GAP

No	Selisih	Bobot Nilai	Keterangan
1	0	5	Tidak ada selisih (Kompetensi sesuai dengan yang di butuhkan)
2	1	4,5	Kompetensi Individu kelebihan 1 tingkat / level
3	-1	4	Kompetensi Individu kekurangan 1 tingkat / level
4	2	3,5	Kompetensi Individu kelebihan 2 tingkat / level
5	-2	3	Kompetensi Individu kekurangan 2 tingkat / level
6	3	2,5	Kompetensi Individu kelebihan 3 tingkat / level
7	-3	2	Kompetensi Individu kekurangan 3 tingkat / level
8	4	1,5	Kompetensi Individu kelebihan 4 tingkat / level
9	-4	1	Kompetensi Individu kekurangan 4 tingkat / level

4. Core Factor dan Secondary Factor

Core Factor atau factor utama dan *Secondary Factor* atau factor pendukung memiliki bobot penilaian yang berbeda. *Core Factor* yang menjadi factor yang utama dalam Karyawan yang di seleksi akan memiliki persentase nilai yang lebih besar dibandingkan dengan *Secondary Factor*. Perhitungan *Core*

Factor dapat menggunakan rumus di bawah ini:

$$\mathbf{NCF} = \frac{\sum NC}{\sum IC}$$

Keterangan :

NCF : Nilai Rata – Rata *Core Factor*.

NC : Jumlah total nilai *Core Factor*.

IC : Jumlah item *Core Factor*.

Sementara untuk perhitungan *Secondary Factor* dapat menggunakan rumus di bawah ini:

$$\mathbf{NSF} = \frac{\sum NS}{\sum IS}$$

Keterangan :

NSF : Nilai Rata – Rata *Secondary Factor*.

NS : Jumlah total nilai *Secondary Factor*.

IS : Jumlah item *Secondary Factor*.

5. Perhitungan Nilai Akhir

Setelah perhitungan *Core Factor* dan *Secondary Factor*, kemudian menghitung nilai total berdasarkan dari persentase dari *Core Factor* dan *Secondary Factor* yang di perkirakan berpengaruh terhadap beasiswa karyawan. Perhitungan dapat di lihat pada rumus di bawah ini:

$$\text{NCF} * (\text{X})\% + \text{NSF} * (\text{X})\% = \text{N}$$

Keterangan :

NCF : Nilai Rata – Rata *Core Factor*

NSF : Nilai Rata – Rata *Secondary Factor*.

N : Nilai Total

(x)% : Nilai Persen yang di inputkan.

6. Perhitungan Penentuan Rangkaing.

Terakhir perhitungan rangkaing, perhitungan tersebut dapat di tentukan setelah menghitung Nilai Akhir.

II.4.2. Model Database

Model database yang saat ini banyak digunakan adalah model database relational. Imam (2008) menyebutkan “Model database” disusun dalam bentuk tabel dua dimensi yang terdiri dari baris (*Record*) dan (*field*), pertemuan antara baris dengan kolom disebut item data (*data value*). Tabel – tabel yang ada dihubungkan (*relationship*) sedemikian rupa menggunakan *field – field* kunci (*key field*) sehingga dapat memindahkan duplikasi data.

Model database relational ini dapat kita kenal konsepnya mulai dari yang paling sederhana misalnya dengan penerapan program aplikasi *excel*. Meskipun untuk pengelolaan database secara luas *excel* jarang digunakan dan kurang mencukupi , Namun untuk melihat konsep database dan konsep membangunnya

program ini dapat dimanfaatkan. *Excel* mempunyai baris yang disebut *Row* dan mempunyai kolom. Kemudian item data merupakan sel atau pertemuan antara baris dan kolom. Tabel – tabel dapat diumpamakan apabila kita menggunakan tabel dalam satu *sheet* tertentu. Data dari berbagai tabel dapat diambil dari tabel lain menggunakan perintah *look up* yang berdasarkan kode kunci tertentu. Kode kunci tersebut berada pada suatu kolom tertentu, Yang dalam konsep database relational disebut sebagai *key field*. (Agustin Mujilan : 2012).

II.4.3. Struktur Database

Untuk memahami konteks *database* kita perlu memahami istilah dan hal – hal yang terkait dengan database. Dalam berbagai program aplikasi database terdapat kesamaan ataupun sedikit perbedaan di dalamnya. Seseorang yang mempelajari database dengan program aplikasi tertentu harus memperhatikan struktur dan karakteristik sesuai dengan bahasa dalam aplikasi tersebut. Namun demikian, secara umum terdapat karakteristik sebagai berikut :

1. *Nama file*

Nama file adalah nama yang digunakan untuk mengidentifikasi adanya data yang disimpan dalam computer dan digunakan untuk pemanggilan data. File yang dikelola akan muncul dalam computer dengan ekstensi sesuai dengan program aplikasinya. File tersebut dapat digunakan untuk menandakan adanya file database, ataupun *file table*. Database dan table akan saling terkait, meskipun cara menyimpan dalam computer akan mengalami sedikit perbedaan pada beberapa aplikasi. Misalnya *Ms. Access* akan menyimpan

dengan *field* yang dapat kita lihat adalah *field* databasenya. Di program *My SQL* nama database ini akan menjadi *folder*. Sementara di *foxpro* nama database dapat menjadi *file*

tersendiri. *Table* cara menyimpannya juga berbeda, dalam *Ms. Access* mungkin kita tidak melihat nama *table*. Secara kasat mata karena akan dikelola di dalam *file* database. Didalam *My SQL* kita bisa melihat beberapa nama *file* terkait dengan pengelolaan *table*. Dan didalam *foxpro table* ini dapat menjadi nama *file* terpisah dan dapat dikenali pula sebagai *free table*.

2. *Database*

Database sebenarnya merupakan nama untuk menampung berbagai *table* didalamnya. Konsep ini akan sama dalam berbagai program aplikasi. Misalnya kita membangun database akuntansi dengan nama database "*akun_base*". Di dalam *akun_base* akan diorganisasikan sebagai *table* yang terkait dengan kegiatan akuntansi misalnya tabel : rekening, pelanggan, jurnal, buku induk, dan administrator program, dan sebagainya. Setiap data yang masuk tidaklah dicatat dalam database, Namun didalam masing – masing *table* yang sesuai.

3. *Table*

Table merupakan tempat untuk menyimpan data sesuai dengan kelompok data. Setiap isi tabel mengandung data yang mempunyai karakteristik dalam penggunaannya. Untuk mempermudah pengolahan biasanya pembangun database mengkategorikan *table* sesuai dengan data isinya sebagai berikut :

a. *Master table*

Master table berisi data tentang hal – hal utama dalam kegiatan database tabel ini berisi record yang relative permanen atau seringkali menjadi acuan ketika mengoperasikan transaksi. Dalam master tabel identitas record menjadi penting dan diusahakan merupakan data atau kode yang bersifat unik. Unik dapat diartikan bahwa tidak ada dalam satu tabel berisi kode yang sama. Desain kode menjadi penting disini . Misalnya dalam mendesain nama akun dalam database akutansi, Maka kode akun menjadi sangat penting artinya *table* berisi nama barang, maka kode barang menjadi hal penting. Contoh lain dalam database akademik, tabel *master* dapat berupa mahasiswa, daftar dosen, daftar kurikulum.

b. *Transaction table*

Tabel traksaksi digunakan untuk menyimpan data dalam menjalankan suatu kegiatan atau bisnis. Data ini seringkali akan bertambah dalam kesehariannya ketika terjadi transaksi yang sesuai dengannya. Secara lebih mudah dapat dipahami dalam akutansi seringkali mencatat transaksi dalam jurnal.Terkait hal tersebut, transaksi ini dicatat dalam tabel jurnal.Dalam mencatat transaksi ini, Kita harus menyesuaikan kode data tertentu dengan kode yang terdapat dalam *master table*.

a. *Tabulation Table*

Tabulasi data dapat digunakan untuk menyimpan data seperti halnya master data namun bersifat sebagai data pembantu ketika menginput

formulir baik untuk data master maupun transaksi. Misalnya untuk memetakan keterangan hobi, jenis kelamin, nama golongan, nama level manajemen, dan sebagainya. Dengan konsep penamaan field yang baik mungkin saja tabel tabulasi ini dapat digunakan untuk memuat berbagai kelompok data. Misalnya fieldnya berupa kode dan keterangan. Contoh kelompok gender dengan L = laki – laki; P = perempuan.

Kelompok level dengan M = Manajer, O = Operator, S = Seller.

b. *Temporary table*

Temporary adalah data sementara yang digunakan untuk membantu ketika terjadi proses transaksi. Data ini dapat saja langsung dihapus ketika transaksi selesai terproses. Misalnya digunakan untuk mempermudah perhitungan, penyimpanan data sementara sebelum di proses setuju ke database.

4. *Field*

Field adalah penanda untuk kolom data. Jika dalam *excel* penanda tersebut adalah kolom A,B, dan seterusnya, sementara dalam konsep *Table* dalam database maka nama *Field* memegang peranan penting. Dalam konsep table dalam database, ketika memanggil dengan nama *field* tentu maka data – data di dalamnya akan muncul. Pengolahan dapat dilakukan dengan membuat *filter* misalnya berdasarkan kode tertentu, berdasarkan kode tertentu, Misalnya kita ingin memanggil record terkait.

Dalam mengatur setting *field* biasanya akan terkait hal – hal sebagai berikut

a. *Field type*

Tipe field ini terkait apakah field tersebut akan berisi data berupa key field (Primary key) atau *descriptor*. *Primary key* akan berisi ID atau kode pokok yang akan digunakan dalam mengidentifikasi record, sehingga data dalam field tersebut diizinkan untuk memiliki lebih dari data yang sama. *Secondary* adalah subset dari key utama.

b. *Data type*

Tipe data merupakan jenis data yang dapat dimasukkan dalam *field*. Hal ini dapat dibagi secara umum sebagai karakter/*text*, numerik, tanggal dan sebagainya.

c. *Field Size*

Penting untuk memahami ukuran field yang akan digunakan dalam menampung data. Dalam pengembangan sistem harus dapat memperkirakan berapa lebar ukuran field yang efektif. Apabila terlalu lebar akan terjadi banyak spasi kosong dan berpengaruh pada ukuran file yang disimpan. Sementara apabila terlalu sempit akan terdapat data yang tidak akan tersimpan. Misalnya ketika kita menghitung bahwa nama menggunakan ukuran 40 karakter sudah memenuhi untuk field data kita. Konsekuensinya, apabila terdapat nama diatas 40 karakter maka akan terpotong menjadi 40 karakter. Konsekuensi lain adalah nama tersebut di input sehingga memuat maksimal 40 karakter yaitu dengan mengadakan singkatan nama.

5. *Records*

Records merupakan baris data. Karena satu baris data biasanya mengindikasikan satu kesatuan data tertentu, maka satu record ada yang menyebut satu data. Dalam konsep database masing-masing records memiliki nomor identitas tersendiri baik itu identitas yang diberikan computer ataupun yang diinputkan secara manual. Sehingga dalam konteks tertentu dapat digunakan konsep nomor records, ID otomatis, ID *primary key*. (Agustinus Mujilan: 2012).

II.5 Beasiswa

Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan yang bertujuan untuk digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan pemberian beasiswa pun dapat dikategorikan pada pemberian cuma – cuma ataupun pemberian dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan, lama ikatan dinas ini berbeda- beda, tergantung pada lembaga ataupun yayasan yang memberikan beasiswa tersebut.

Menurut Nasution (2007) Pendidikan adalah suatu proses, teknis dan metode belajar mengajar dengan maksud mentrasfer suatu pengetahuan dari seseorang kepada orang lain sesuai dengan standard yang telah ditetapkan sebelumnya. Menurut Pont (2007) Pelatihan adalah mengembangkan orang – orang sebagai individu dan mendorong mereka lebih percaya diri dan berkemampuan dalam hidup dan pekerjaannya.

II.6. Visual Studio 2010

Microsoft Visual Basic 2010 adalah salah satu komponen *Microsoft Visual Studio 2010*. Software ini di luncurkan Microsoft pada tanggal 12 April 2010 dengan nama kode Dev10 dan menggunakan Net Framework 4.0. Intergrated Development Envirotment (IDE) pada visual studio 2010 telah di design ulang sehingga lebih enak di pandang dan digunakan programmer. Untuk kode editor nya, Visual Basic 2010 telah menambah fitur highlights refrence. Ketika satu symbol / kode dalam bahasa pemrograman nya di pilih, maka symbol / kode yang sama, meskipun penggunaan nya berbeda akan terlihat berwarna sama. Misal jika kode math di pilih, seluruh kode math akan terlihat berwarna sama pelayanan perpustakaan menggunakan Visual Basic 2010 di kembangkan untuk melihat apakah terdapat kelayakan pada pengembangan media sesuai dengan kriteria kelayakan media, kelayakan media pada perpustakaan, serta apakah terdapat kelayakan dalam penggunaan nya.setelah melalui tahap validasi pustaka dan media di dapat hasil bahwa nilai untuk pustaka mencapai 90 % atau terkategori “Sangat Baik”. Sedangkan nilai media memperoleh persentase 83.75% atau terkategori “Baik”. Serta uji Praktikalitas mendapat hasil 88.5% atau terkategori “Sangat Baik”.(Bayu Fadilah: 2014).

II.7. SQL Server 2008

SQL (Structured Query Language) adalah bahasa prusedural untuk mengakses data pada database relational. SQL adalah bahasa database yang di pergunakan dalam menyelesaikan permasalahan dalam database serta mempunyai kelebihan dalam mengolah data. Standart SQL mula – mula di defenisikn oleh ISO (*International Standart Organization*) dan ANSI (*The American National Standart Institute*) yang di kenal dengan sebutan SQL 86. (Eka Iswandy : 2015).

Dengan menggunakan SQL, kita dapat melakukan hal-hal berikut :

1. Memodifikasi Struktur Database.
2. Mengubah, mengisi, menghapus isi database.
3. Mentrasnfer data antara database yang berbeda. SQL ada yang di kembangkan untuk PC da nada juga yang di kembangkan untuk dapat mengakomodasi database yang sangat besar.

Beberapa contohnya antara lain:

1. *Microsoft Acsess*

Diguanakn untuk PC, sangat mudah di pakai dimana perintah SQL dapat langsung di masukan atau melalui fasilitas yang di gunakan.

2. *Microsoft Query*

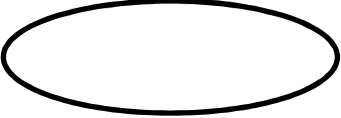
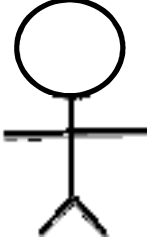




SQL yang di paket dengan produk lain dari Microsoft Windows, yaitu Microsoft Visual Studio seperti Visual basic dan Visual C++. Untuk terhubung dengan database lain menggunakan ODBC.

II.8. UML(*Unified Modeling Language*)

Hasil pemodelan pada OOAD terdokumentsaikan dalam bentuk *Unified Modeling Language* (UML). UML adalah bahasa spesifikasi standart yang di pergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodologi dalam mengembangkan sistem berorientasi objek dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini sangat banyak di pergunakan dalam dunia industry yang merupakan standart pemodelan umum dalam industry perangkat lunak dan pengembangan sistem. (Windu Gata : 2013).

1. *Diagram Use Case (Use Case Diagram)*





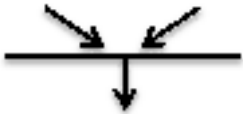
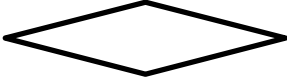

Use Case diagram merupakan pemodelan untuk kelakuan (*Behavior*) Sistem informasi yang akan dibuat. *Use Case* mendeskripsikan sebuah interaksi anatara satu atau lebih actor dengan sistem informasi yang akan di buat. Dapat dikatakan use case di gunakan untuk mengetahui fungsi apa saja yang ada di dalam sistem informasi dan siapa saja yang berhak menggunakan fungsi – fungsi tersebut. Simbol yang digunakan dalam use diagram, yaitu:

Gambar	Keterangan
	<p>Use case menggambarkan fungsionalitas yang disediakan sistem sebagai unit – unit yang bertukar pesan antar unit dengan actor, biasanya dinyatakan dengan menggunakan kata kerja di awal nama use case.</p>
	<p>Aktor adalah <i>abstraction</i> dari orang atau sistem yang lain mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi actor, harus ditentukan pembagian tenaga kerja dan tugas – tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa actor berinteraksi dengan use case, tetapi tidak memiliki control terhadap use case.</p>
	<p>Asosiasi antara actor dan use case, digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan aliran data.</p>
	<p>Asosiasi antara actor dan use case yang menggunakan panah terbuka untuk mengindikasikan bila actor berinteraksi secara pasif dengan sistem.</p>
	<p><i>Include</i>, merupakan didalam use case lain(required) atau pemanggilan use case lain, contohnya adalah pemanggilan sebuah fungsi program.</p>
	<p><i>Extend</i>, merupakan perluasan dari use case lain jika kondisi atau syarat terpenuhi.</p>

Gambar II.3 : Simbol Use Case Diagram

2. Diagram Aktivitas (*Activity Diagram*)

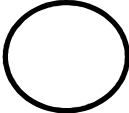
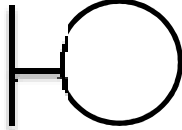
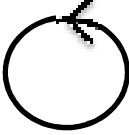




Activity Diagram mengGambarkan *Workflow* (Aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Simbol yang di gunakan dalam *Activity Diagram*, yaitu :

Gambar	Keterangan
	Start point, diletakkan pajak pojok kiri atas dan merupakan awal aktivitas.
	<i>End point</i> , akhir aktivitas.
	<i>Activites</i> , mengGambarkan suatu proses/ kegiatan bisnis.
	<i>Fork</i> (Percabangan), digunakan untuk menunjukkan kegiatan yang dilakukan secara parallel atau untuk menggabungkan dua kegiatan parallel menjadi satu.
	<i>Join</i> (Penggabungan) atau rake , digunakan untuk menunjukkan adanya dekomposisi.
	<i>Decision Points</i> , mengGambarkan pilihan untuk pengambilan keputusan, <i>true, false</i>
	<i>Swimlane</i> , merupakan <i>activity diagram</i> untuk menunjukkan siapa melakukan apa.

Gambar II.4 : Simbol Diagram Aktivitas

3. *Sequence Diagram*

Sequence Diagram mengGambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan pesan yang di kirimkan dan di terima antar objek. Simbol yang di gunakan *Sequence Diagram*, yaitu:

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas – entitas yang membentuk Gambaran awal sistem dan menjadi landasan untuk menyusun basis data.
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interface</i> atau interaksi antara satu atau lebih actor dengan sistem, seperti tampilan <i>formentry</i> dan <i>form</i> cetak.
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi atau aturan bisnis yang melibatkan berbagai objek.
	<i>Message</i> , Simbol mengirim pesan antar <i>Class</i> .
	<i>Recursive</i> , mengGambarkan pengiriman pesan yang dikirim untuk dirinya sendiri.
	<i>Activation</i> , activation mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivitas sebuah operasi.
	<i>Lifeline</i> , garis titik – titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

Gambar II.5 : Simbol Sequence Diagram

4. *Class Diagram* (Diagram Kelas)

Merupakan hubungan antar kelas dan penjelasan detail tiap- tiap kelas di dalam model design dari suatu sistem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang melakukan perilaku sistem. *Class Diagram* juga menunjukkan atribut-atribut dan operasi-operasi dari sebuah kelas dan constraint yang berhubungan dengan objek yang di koneksikan.

Class Diagram secara khas meliputi: kelas (*Class*), relasi, *Associations*, *Generalization* dan *Aggregation*, Atribut (*Attributes*) operasi (*Operation Method*), *Visibility*, Tingkat Akses objek eksternal kepada suatu operasi atau atribut. Hubungan antar kelas mempunyai keterangan yang di sebut dengan *Multiplicity* atau kardinaliti.

Multiplicity	Penjelasan
1	Satu dan hanya satu
0..*	Boleh tidak ada atau 1 atau lebih
1..*	1 atau lebih
0..1	Boleh tidak ada, maksimal 1
n..n	Batasan antara, Contoh 2..4 mempunyai arti minimal 2 maksimum 4

Gambar II.6 : Simbol Class diagram