

BAB II

LANDASAN TEORI

2.1 Beasiswa

Pada dasarnya, beasiswa adalah penghasilan bagi yang menerimanya. Hal ini sesuai dengan ketentuan pasal 4 ayat (1) UU PPh/2000. Disebutkan pengertian penghasilan adalah tambahan kemampuan ekonomis dengan nama dan dalam bentuk apa pun yang diterima atau diperoleh dari sumber Indonesia atau luar Indonesia yang dapat digunakan untuk konsumsi atau menambah kekayaan Wajib Pajak (WP). Karena beasiswa bisa diartikan menambah kemampuan ekonomis bagi penerimanya, berarti beasiswa merupakan penghasilan.

Beasiswa dapat dikatakan sebagai pembiayaan yang tidak bersumber dari pendanaan sendiri atau orang tua, akan tetapi diberikan oleh pemerintah, perusahaan swasta, kedutaan, universitas, serta lembaga pendidik atau peneliti, atau juga dari kantor tempat bekerja yang karena prestasi seorang karyawan dapat diberikan kesempatan untuk meningkatkan kapasitas sumber daya manusianya melalui pendidikan. Biaya tersebut diberikan kepada yang berhak menerima, terutama berdasarkan klasifikasi, kualitas, dan kompetensi si penerima beasiswa (Gafur, 2008).

Banyaknya peminat beasiswa dibanding dengan sedikitnya jumlah beasiswa yang diberikan, menjadikan pihak donatur beasiswa memberikan cukup banyak persyaratan yang harus dipenuhi oleh calon penerima beasiswa. Pihak akademik selaku penyalur beasiswa, harus melakukan

seleksi terlebih dahulu sebelum mengajukan penerima beasiswa. Perlu adanya sistem yang mendukung keputusan akademik dalam menyeleksi calon penerima beasiswa.

2.2 Sistem Pendukung Keputusan

Menurut Alter tahun 2000 dalam Kusri, menyatakan Sistem Pendukung Keputusan (SPK) atau *Decision Support System* merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan manipulasi data. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang tidak terstruktur, dimana tidak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. SPK biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. SPK tersebut dinamakan aplikasi SPK. Aplikasi SPK menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan.

Sistem Pendukung Keputusan (SPK) merupakan suatu pendekatan atau metodologi untuk mendukung keputusan. SPK menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi untuk masalah manajemen spesifik yang tidak terstruktur. SPK menggunakan data, memberikan antarmuka pengguna yang mudah dan dapat menggabungkan pemikiran pengambil keputusan. Sebagai tambahan, SPK biasanya menggunakan berbagai model dan dibangun oleh suatu proses interaktif dan

iteratif. SPK mendukung semua fase pengambilan keputusan dan dapat memasukkan suatu komponen pengetahuan. SPK dapat digunakan oleh pengguna tunggal pada satu PC atau bisa menjadi berbasis Web untuk digunakan oleh banyak orang pada beberapa lokasi (Turban dkk, 2005).

Little mendefinisikan SPK sebagai ”sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan”. Little menyatakan bahwa untuk sukses, sistem tersebut haruslah sederhana, cepat, mudah dikontrol, adaptif, lengkap dengan isu-isu penting, dan mudah berkomunikasi.

Bonczek, dan kawan kawan mendefinisikan SPK sebagai sistem berbasis komputer yang terdiri dari tiga komponen yang saling berinteraksi: sistem bahasa (mekanisme untuk memberikan komunikasi antara pengguna dan komponen SPK lain), sitem pengetahuan (repositori pengetahuan domain masalah yang ada pada SPK baik sebagai data atau sebagai prosedur), dan sistem pemrosesan masalah (hubungan antara dua komponen lainnya, terdiri dari satu atau lebih kapabilitas manipulasi masalah umum yang diperlukan untuk pengambilan keputusan). Konsep-konsep yang diberikan oleh defenisi tersebut sangat penting untuk memahami hubungan antara SPK dan pengetahuan.

Keen menerapkan istilah SPK ”untuk situasi dimana sistem ’final’ dapat dikembangkan hanya melalui suatu proses pembelajaran dan evolusi yang adaptif.” Keen mendefinisikan SPK sebagai suatu produk dari proses pengembangan dimana pengguna SPK, pembangun SPK, dan SPK itu

sendiri mampu mempengaruhi satu dengan yang lainnya, dan menghasilkan evolusi sistem dan pola-pola penggunaan.

Sebuah sistem dikatakan sebagai Sistem Pendukung Keputusan (SPK) apabila memenuhi karakteristik sebagai berikut:

- a) Mendukung pengambilan keputusan secara cepat dan tepat.
- b) Menggunakan model matematis yang sesuai.
- c) Adanya interface antara manusia dan mesin, dimana manusia yang mengontrol.
- d) Mempunyai kemampuan dialog.

SPK juga memiliki karakteristik dan kemampuan adalah sebagai berikut:

1. Mendukung seluruh kegiatan organisasi
2. Mendukung beberapa keputusan yang saling berinteraksi
3. Dapat digunakan berulang kali dan bersifat konstan
4. Terdapat dua komponen utama, yaitu data dan model
5. Menggunakan baik data eksternal dan internal
6. Memiliki kemampuan *what-if analysis* dan *goal seeking analysis*
7. Menggunakan beberapa model kuantitatif (Kosasi, 2002).

Adapun syarat yang harus dimiliki sebuah SPK adalah:

- a) Membantu manajer membuat keputusan untuk memecahkan masalah semi terstruktur
- b) Mendukung penilaian manajer, bukan mencoba menggantikan
- c) Memungkinkan efektifitas pengambilan keputusan manajer daripada efisiensi.

2.3 *Fuzzy Multiple Attribute Decision Making (FMADM)*

Menurut Kusumadewi (2007) dalam Wibowo dkk, *Fuzzy Multiple Attribute Decision Making* adalah suatu metode yang digunakan untuk mencari alternatif optimal dari sejumlah alternatif dengan kriteria tertentu. Inti dari metode tersebut adalah menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif yang sudah diberikan. Jenis metode yang dapat digunakan untuk menyelesaikan masalah *Fuzzy Multiple Attribute Decision Making* antara lain:

- a. *Simple additive weighting method (SAW)*
- b. *Weight product (WP)*
- c. ELECTRE
- d. *Technique For Order Preference by Similarity to Ideal Solution (TOPSIS)*
- e. *Analytic Hierarchy Process (AHP)*

Metode TOPSIS digunakan untuk menyelesaikan masalah dengan banyak kriteria sebagai komponen penilaian untuk setiap alternatif. Metode TOPSIS merupakan salah satu metode yang bisa membantu proses pengambilan keputusan yang optimal untuk menyelesaikan masalah secara praktis, karena konsepnya sederhana dimana alternatif yang dipilih selain memiliki kedekatan dengan solusi ideal positif, juga jauh dengan solusi ideal negatif. Selain itu metode TOPSIS juga mudah dipahami, karena memiliki kemampuan untuk mengukur kinerja relatif dari alternatif-alternatif keputusan dalam bentuk matematis yang sederhana (Kusumadewi,

2007). Metode TOPSIS sangat sesuai untuk menyelesaikan masalah seperti dalam proses seleksi penerima beasiswa ini.

2.4 *Technique For Order Preference By Similarity To Ideal Solution* (TOPSIS)

TOPSIS adalah salah satu metode pengambilan keputusan multi kriteria yang pertama kali diperkenalkan oleh Yonn dan Hwang pada tahun 1981. Dengan ide dasarnya adalah bahwa alternatif yang dipilih memiliki jarak terdekat dengan solusi ideal positif dan memiliki jarak terjauh dari solusi ideal negatif. Sebuah matriks dengan alternatif dan kriteria dapat dinotasikan sebagai berikut:

$$D = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{m1} & \cdots & x_{mn} \end{bmatrix}$$

dimana: D = matriks

m = alternatif

n = kriteria

X_{ij} = alternatif ke-i dan kriteria ke-j

Konsep ini banyak digunakan pada beberapa model FMADM karena konsepnya sederhana dan mudah dipahami, komputasinya efisien, dan memiliki kemampuan untuk mengukur kinerja alternatif.

Langkah-langkah penyelesaian masalah FMADM dengan TOPSIS:

- 1) Membuat matriks keputusan yang ternormalisasi

Setiap elemen pada matriks D (kriteria C) dinormalisasikan untuk mendapatkan matriks normalisasi R . Setiap normalisasi dari nilai R_{ij} dapat dilakukan dengan perhitungan sebagai berikut:

$$r_{ij} = \frac{X_{ij}}{\sqrt{\sum_{i=1}^m X_{ij}^2}} \quad \text{dengan } i = 1, 2, \dots, m \text{ dan } j = 1, 2, \dots, n.$$

2) Membuat matriks keputusan yang ternormalisasi terbobot

Diberikan bobot $W = (w_1, w_2, \dots, w_n)$, sehingga *weighted normalized matrix* V dapat dihasilkan sebagai berikut:

$$V = \begin{vmatrix} W_{11}R_{11} & \dots & W_{1n}R_{1n} \\ \dots & \dots & \dots \\ W_{m1}R_{m1} & \dots & W_{nm}R_{nm} \end{vmatrix}$$

dimana : $i=1,2,3,\dots,m$, $j=1,2,3,\dots,n$, dan $w_{ii} r_{ij} = y_{ij}$.

3) Menentukan matriks solusi ideal positif & matriks solusi ideal negatif

Solusi ideal positif dinotasikan dengan A^+ dan solusi ideal negatif dinotasikan dengan A^- , sebagai berikut :

$$A^+ = (y_1^+, y_2^+, \dots, y_n^+)$$

$$A^- = (y_1^-, y_2^-, \dots, y_n^-)$$

dengan $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$

Syaratnya:

$$y_j^+ = \begin{cases} \max_i y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \min_i y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

$$y_j^- = \begin{cases} \min_i y_{ij}; & \text{jika } j \text{ adalah atribut keuntungan} \\ \max_i y_{ij}; & \text{jika } j \text{ adalah atribut biaya} \end{cases}$$

- 4) Menentukan jarak antara nilai setiap alternatif dengan matriks solusi ideal positif & matriks solusi ideal negatif (*Separation measure*).

Jarak antara alternatif A_i dengan solusi ideal positif dirumuskan

$$\text{sebagai: } D_i^+ = \sqrt{\sum (Y_j^+ - Y_{ij})^2}$$

Jarak antara alternatif A_i dengan solusi ideal negatif dirumuskan

$$\text{sebagai: } D_i^- = \sqrt{\sum (Y_{ij} - Y_j^-)^2}$$

- 5) Menentukan nilai preferensi untuk setiap alternatif.

Kedekatan relatif dari alternatif A^+ dengan solusi ideal A^- direpresentasikan dengan:

$$V_i = \frac{D_i^-}{D_i^- + D_i^+} \quad \text{dimana } 0 < D_i < 1 \text{ dan } i=1,2,3,\dots,m$$

Alternatif dapat dirangking berdasarkan urutan D_i . Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terdekat terhadap solusi ideal positif dan berjarak terjauh dengan solusi ideal negatif. Dengan kata lain, yang memiliki nilai preferensi V_i terbesar yang merupakan alternatif pilihannya.

Sebelum melakukan pembuatan program, dilakukan perancangan model sistem dan database dengan menggunakan *Schematic flowchart*, *Context Diagram*, HIPO, DFD, dan ERD.

2.5 Bagan Alir Skematik (*Schematic Flowchart*)

Bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur


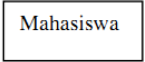


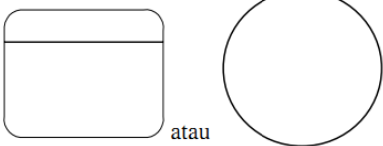

di dalam sistem. Perbedaannya adalah, bagan alir skematik selain menggunakan simbol-simbol bagan alir sistem, juga menggunakan gambar-gambar komputer dan peralatan lainnya yang digunakan. Maksud penggunaan gambar-gambar ini adalah untuk memudahkan komunikasi kepada orang yang kurang paham dengan simbol-simbol bagan alir. Penggunaan gambar-gambar ini memudahkan untuk dipahami, tetapi sulit dan lama menggambarinya.

Menurut Jogiyanto (2005) “bagan alir skematik (*schematic flowchart*) merupakan bagan alir yang mirip dengan bagan alir sistem, yaitu untuk menggambarkan prosedur di dalam sistem”

2.6 *Context Diagram*

Context Diagram merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem secara keseluruhan. Proses tersebut diberi nomor nol. Semua entitas eksternal yang ditunjukkan pada diagram konteks berikut aliran data-aliran data utama menuju dan dari sistem. Diagram tersebut tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan, begitu *entitas-entitas* eksternal serta aliran data-aliran data menuju dan dari sistem diketahui menganalisis dari wawancara dengan user dan sebagai hasil analisis dokumen.

Menurut Sutejo (2002), *contexti diagram* merupakan pola penggambaran yang berfungsi untuk memperlihatkan interaksi sistem informasi dengan lingkungan dimana sistem itu berada. Simbol *context diagram* sebagai berikut.

Simbol	Arti	Contoh
	Terminator	
	Aliran Data/ Data flow	Informasi mahasiswa baru 
	Proses/Process	

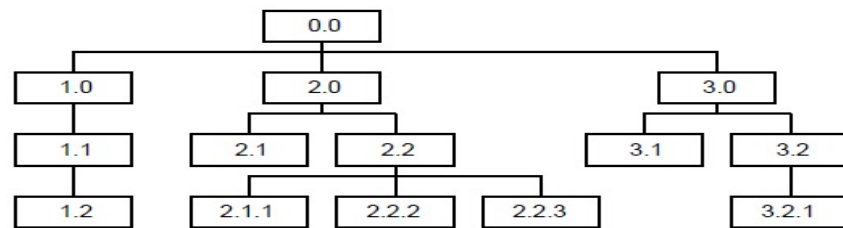
Gambar 2.1 Simbol *context diagram*

2.7 HIPO

HIPO (*Hierarchy plus Input-Proses-Output*) merupakan metodologi yang dikembangkan dan didukung oleh IBM. HIPO sebenarnya adalah alat dokumentasi program. “HIPO (*Hierarchy Plus Input Process Output*) adalah alat desain dan teknik dokumentasi dalam siklus pengembangan sistem”. HIPO merupakan akronim dari *Hierarchy plus Input-Process-Output*. HIPO merupakan paket yang berisikan suatu set diagram yang secara grafis menjelaskan fungsi suatu sistem dari tingkat umum ketingkat khusus.(Jogiyanto HM, 2007)

Ada 3 jenis diagram HIPO, salah satunya Daftar Isi Visual/ *Visual Tabel of Contents (VTOC)*, yang terdiri dari satu diagram hirarki atau lebih. Diagram ini menggambarkan hubungan dari fungsi-fungsi secara berjenjang *Visual tabel of contents* menggambarkan seluruh program HIPO baik rinci maupun ringkasan yang terstruktur. Pada diagram ini nama dan nomor dari program HIPO diidentifikasi. Struktur paket diagram dan hubungan fungsi juga diidentifikasi dalam bentuk hirarki. Keterangan

masing-masing fungsi diberikan pada bagian penjelasan yang diikutsertakan dalam diagram ini.



Bagian penjelasan



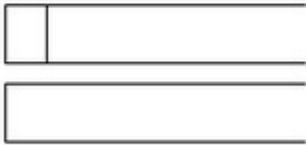

0.0	
1.0	
2.0	
3.0	

Gambar 2.2 Diagram HIPO (*Hierarchy plus Input-Proses-Output*)

2.8 DFD (*Data Flow Diagram*)

Data Flow Diagram (DFD) merupakan gambaran suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut mengalir, dan akan disimpan, karena dapat menggambarkan arus data di dalam sistem dengan terstruktur dan jelas. Maka DFD juga merupakan dokumentasi dari sistem yang baik (HM. Jogyanto, 2005).

Terdapat empat komponen utama dalam pemodelan ini dapat dilihat pada gambar sebagai berikut:

Simbol	Keterangan
	<i>External Entity</i> , merupakan kesatuan di lingkungan luar sistem yang bisa berupa orang, organisasi atau sistem lain.
	<i>Process</i> , merupakan proses seperti perhitungan aritmatik penulisan suatu formula atau pembuatan laporan
	<i>Data Store</i> (Simpan Data), dapat berupa suatu file atau database pada sistem komputer atau catatan manual
	<i>Data Flow</i> (arus data), arus data ini mengalir diantara proses, simpan data dan kesatuan luar

Gambar 2.3 Simbol DFD

2.9 ERD

Entity-Relationship Model (ERM) merupakan abstrak dan konseptual representasi data. *Entity-Relationship* adalah salah satu metode pemodelan basis data yang digunakan untuk menghasilkan skema konseptual untuk jenis/model data semantik sistem. Dimana sistem seringkali memiliki basis data relasional, dan ketentuannya bersifat *top-down*. Diagram untuk menggambarkan model *Entity-Relationship* ini disebut *Entity-Relationship diagram*, *ER diagram*, atau ERD.

Menurut Bin ladjamudin (2005) “ERD merupakan model jaringan data yang menekankan pada struktur-struktur dan relationship data”

Notasi-notasi simbolik yang digunakan dalam *Entity Relationship Diagram* seperti pada gambar 2.4 adalah sebagai berikut :

- a) *Entitas*, Adalah segala sesuatu yang dapat digambarkan oleh data. *Entitas* juga dapat diartikan sebagai individu yang mewakili sesuatu yang nyata (eksistensinya) dan dapat dibedakan dari sesuatu yang lain (Fathansyah, 1999). 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*.
- b) *Atribut*, *Atribut* merupakan pendeskripsian karakteristik dari entitas. *Atribut* digambarkan dalam bentuk lingkaran atau elips. *Atribut* yang menjadi kunci *entitas* atau *key* diberi garis bawah.
- c) *Relasi* atau Hubungan, *Relasi* menunjukkan adanya hubungan diantara sejumlah *entitas* yang berasal dari himpunan *entitas* yang berbeda.
- d) Penghubung antara himpunan *relasi* dengan himpunan *entitas* dan himpunan *entitas* dengan *atribut* dinyatakan dalam bentuk garis.
- e) Derajat *relasi* atau kardinalitas menunjukkan jumlah maksimum *entitas* yang dapat berelasi dengan *entitas* pada himpunan *entitas* yang lain. Macam-macam kardinalitas adalah:
 - a. Satu ke satu (*one to one*), Setiap anggota *entitas* A hanya boleh berhubungan dengan satu anggota *entitas* B, begitu pula sebaliknya
 - b. Satu ke banyak (*one to many*), Setiap anggota *entitas* A dapat berhubungan dengan lebih dari satu anggota *entitas* B tetapi tidak sebaliknya.

- c. Banyak ke banyak (*many to many*), Setiap *entitas* A dapat berhubungan dengan banyak *entitas* himpunan *entitas* B dan demikian pula sebaliknya

Berikut adalah gambar notasi-notasi simbolik yang digunakan dalam

ERD:

Notasi	Keterangan
	Entitas , adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi , menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut , berfungsi mendeskripsikan karakter entitas (atribut yg berfungsi sebagai key diberi garis bawah)
	Garis , sebagai penghubung antara relasi dengan entitas, relasi dan entitas dengan atribut.

Gambar 2.4 Simbol ERD

2.10 Borland Delphi 7

Borland Delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi untuk mengolah teks, grafik, angka, database dan aplikasi web. Program ini mempunyai kemampuan luas yang terletak pada produktifitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta bahasa pemrogramannya terstruktur dan lengkap.

Menurut Kadir (2004), Delphi merupakan sebuah peranti pengembangan aplikasi berbasis windows yang dikeluarkan oleh *Borland International*. Perangkat lunak ini sangat terkenal di kalangan pengembang aplikasi karena mudah untuk dipelajari dan dapat digunakan untuk menangani berbagai hal, dari aplikasi matematika, permainan, hingga database. Pada penanganan database, Delphi menyediakan fasilitas yang memungkinkan pemrogram dapat berinteraksi dengan database seperti, dBase, Paradox, Oracle, MySQL, dan Access. Pendapat serupa juga dikemukakan oleh Alam (2005), Delphi merupakan bahasa pemrograman yang mempunyai cakupan kemampuan yang luas dan sangat canggih. Berbagai jenis aplikasi dapat dibuat dengan delphi, termasuk aplikasi untuk mengelolah teks, grafik, angka, database dan aplikasi web.

IDE Delphi merupakan lingkungan pemrograman terpadu yang terdapat dalam Delphi. Dengan IDE semua yang diperlukan dalam pengembangan, dalam kondisi normal, semuanya telah tersedia. Adapun bagian-bagian IDE Delphi yang biasa ditampilkan yaitu :

1) Jendela Utama

Di dalam jendela utama Delphi terdapat menu-menu sebagaimana menu aplikasi Windows umumnya, *toolbar* yang merupakan langkah cepat dari beberapa menu, dan component palette yaitu gudang komponen yang akan digunakan untuk membuat aplikasi.

2) Objek *Treeview*

Fasilitas ini berguna untuk menampilkan daftar komponen yang digunakan dalam pengembangan aplikasi sesuai dengan penempatannya.

3) Objek *Inspector*

Objek ini digunakan untuk mengatur properti dan event suatu komponen. Akan tetapi tidak dapat mengubah langsung properti-properti yang tidak ditampilkan kecuali melalui penulisan kode program.

4) Form *Designer*

Form adalah komponen utama dalam pengembangan aplikasi. Form designer adalah tempat melekatnya komponen yang lain, dengan arti lain tempat komponen-komponen lain diletakkan.

5) *Code Editor, Explorer* dan *Component Diagram*

Code Editor adalah tempat kode program yang diperlukan untuk mengatur tugas aplikasi ditulis. *Code Explorer* adalah fasilitas yang membantu penjelajahan kode program menjadi lebih mudah. *Component Diagram* adalah fasilitas yang dapat digunakan untuk membuat diagram komponen-komponen yang digunakan dalam aplikasi.

2.11 MySQL

MySQL adalah salah satu jenis database server yang menggunakan bahasa SQL untuk mengakses databasenya. Lisensi MySQL adalah FOSS

License Exception dan ada juga yang versi komersialnya. Tag MySQL adalah “*The World's most popular open source database*”. MySQL tersedia untuk beberapa *platform*, diantaranya adalah untuk versi windows dan versi linux. Untuk melakukan administrasi secara lebih mudah terhadap MySQL, dapat menggunakan software tertentu, diantaranya adalah PhpMyAdmin dan MySQL yog.

MySQL (*My Structure Query Language*) adalah salah satu database management sistem (DBMS) yang berfungsi mengolah database dengan menggunakan bahasa SQL” (Anhar, 2010).

MySQL merupakan perangkat lunak untuk sistem manajemen database yang menggunakan bahasa SQL (*Struktur Query Language*). SQL adalah bahasa standar yang digunakan untuk mengakses database server.

Keunggulan MySQL dibandingkan dengan database lainnya yaitu:

- a) MySQL dapat berjalan dengan stabil pada berbagai sistem operasi, seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, dan masih banyak lagi.
- b) MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah.
- c) MySQL memiliki kecepatan yang baik dalam menangani *query* (perintah SQL). Dengan kata lain, dapat memroses lebih banyak SQL per satuan waktu.
- d) Dari segi *security* atau keamanan data, MySQL memiliki beberapa lapisan *security*, seperti level *subnet mask*, nama *host*, dan izin akses

user dengan sistem perizinan yang mendetail serta password yang terenskripsi.

2.12 XAMPP

XAMPP adalah aplikasi web *server* instan yang dibutuhkan untuk membangun aplikasi berbasis web. Fungsi XAMPP adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache* HTTP *server*, MySQL database, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X=cross platform, Apache, MySQL,PHP, dan Perl. Program ini tersedia dalam lisensi GNU General Public License dan merupakan *open source* atau gratis. (Gunawan,Wahyu, 2010)

Menggunakan XAMPP, tidak perlu menginstall aplikasi-aplikasi tersebut satu persatu. Paket aplikasi perlu diekstrak dan di install terlebih dahulu, dengan memilih jenis XAMPP sesuai dengan jenis OS nya. Setelah sukses menginstall XAMPP, dapat langsung diaktifkan MySQL dengan cara mengaktifkan XAMPP. Masuk ke PhpMyAdmin, dengan cara mengetik di jendela browser *http://localhost/phpmyadmin* atau klik tombol admin di XAMPP *control panel application*.

PhpMyAdmin adalah sebuah aplikasi *open source* yang berfungsi untuk memudahkan manajemen MySQL. PhpMyAdmin, dapat membuat database, membuat tabel, menginsert, menghapus dan mengupdate data dengan GUI dan terasa lebih mudah, tanpa perlu mengetikkan perintah SQL

secara manual. PhpMyadmin dapat di download secara *free* di <http://www.phpmyadmin.net>. PhpMyAdmin dapat di jalankan di banyak OS, selama dapat menjalankan webserver dan Mysql karena berbasis web. (Gunawan,Wahyu, 2010)

2.13 Tinjauan Pustaka

Tri Handayani (2012) dalam skripsinya yang berjudul Sistem Pendukung Keputusan Beasiswa Diklat Di Solo Technopark dengan *Fuzzy MADM*. Aplikasi ini dibuat untuk membantu penentuan penerima beasiswa diklat di solo technopark. Variabel yang digunakan antara lain: nilai ujian masuk, jumlah penghasilan orang tua, jumlah tanggungan orang tua, kondisi rumah. Metode penyelesaian masalah yang digunakan adalah metode *Simple Additive Weighting Method (SAW)*. Alternatif dengan nilai V_i paling besar merupakan alternatif yang disarankan.

Supriyadi (2014) dalam skripsinya yang berjudul Sistem Pendukung Keputusan Penerimaan Beasiswa Untuk Siswa Kurang Mampu Menggunakan Metode MADM (studi kasus:SMK KARYA NUGRAHA BOYOLALI). Metode penyelesaian masalah yang digunakan adalah *Weighted Product (WP)*, yang mana kriteria bobot yang digunakan yaitu: penghasilan orang tua, jumlah tanggungan orang tua, status rumah, rata-rata tagihan per bulan, dan nilai raport. Penerima beasiswa dengan vector V paling tinggi merupakan alternatif yang dipilih sebagai penerima beasiswa.