



BAB II

TINJAUAN PUSTAKA

2.1 Teori Judul

2.1.1 Pengertian Sistem Pendukung Keputusan

Menurut Nofriansyah dan Sarjon (2017:2), “Sistem Pendukung Keputusan adalah suatu sistem informasi spesifik yang ditunjukkan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan yang bersifat semi terstruktur”.

Sedangkan menurut Little dalam Nofriansyah dan Sarjon (2017:1), mendefinisikan sistem pendukung keputusan sebagai suatu informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan sehingga membantu manajemen dalam menangani berbagai permasalahan yang terstruktur maupun tidak terstruktur dengan menggunakan data dan model.

Dari berbagai definisi diatas dapat disimpulkan bahwa sistem pendukung keputusan adalah suatu sistem informasi yang dirancang untuk membantu seseorang maupun sekelompok orang dalam mengambil keputusan terhadap permasalahan yang bersifat terstruktur maupun tidak terstruktur melalui beberapa alternatif dan kriteria yang telah ditentukan sehingga dapat menghasilkan keputusan yang lebih cepat dan tepat.

Komponen-komponen Sistem Pendukung Keputusan (Nofriansyah dan Sarjon, 2017:3) terdiri dari:

- a. *Data Management*. Termasuk *database*, yang mengandung data yang relevan untuk berbagai situasi dan diatur oleh *software* yang disebut *Database Management System (DBMS)*.
- b. *Model Management*. Melibatkan model finansial, *statistical*, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analistis, dan manajemen *software* yang diperlukan



-
- c. *Communication (dialog subsystem)*. *User* dapat berkomunikasi dan memberikan perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.
 - d. *Knowledge Management*. Subsistem *optional* ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri.

Menurut Simon dalam Nofriansyah dan Sarjon (2017:4) ada tiga fase dalam proses pengambilan keputusan diantaranya sebagai berikut:

1. *Intelligence*

Tahap ini merupakan proses penelusuran dan pendektaksian dari ruang lingkup problematika secara proses pengenalan masalah. Data masukan diperoleh, diproses dan diuji dalam rangka mengidentifikasi masalah.

2. *Design*

Tahap ini merupakan proses menemukan, mengembangkan dan menganalisis alternatif tindakan yang bisa dilakukan. Tahap ini meliputi menguji kelayakan solusi.

3. *Choice*

Pada tahap ini dilakukan proses pemilihan diantara berbagai alternatif tindakan yang mungkin dijalankan. Hasil pemilihan tersebut kemudian diimplmentasikan dalam proses pengambilan keputusan.

2.1.2 Metode *Simple Additive Weighting* (SAW)

Menurut Nofriansyah dan Sarjon (2017:33), metode *Simple Additive Weighting* merupakan metode pembobotan sederhana atau penjumlahan terbobot pada penyelesaian masalah dalam sebuah sistem pendukung keputusan. Konsep dalam metode ini adalah dengan mencari peringkat kinerja (skala prioritas) pada setiap alternatif di semua atribut.

Adapun algoritma penyelesaian metode ini (Nofriansyah dan Sarjon, 2017:33) diantaranya sebagai berikut:

1. Langkah 1 : Mendefinisikan terlebih dahulu kriteria-kriteria yang akan di jadikan sebagai tolak ukur penyelesaian masalah.



2. Langkah 2 : Menormalisasi setiap nilai alternatif pada setiap atribut dengan cara menghitung nilai *rating* kinerja.
3. Langkah 3 : Menghitung nilai bobot preferensi pada setiap alternatif.
4. Langkah 4 : Melakukan perankingan.

Rumus yang digunakan pada metode *Simple Additive Weighting* (Nofriansyah dan Sarjon, 2017:33) yaitu:

1. Menormalisasikan setiap alternatif (menghitung nilai *rating* kinerja).

$$r_{ij} = \left\{ \begin{array}{ll} \frac{x_{ij}}{\max_i x_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min_i x_{ij}}{x_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{array} \right\}$$

Dimana r_{ij} adalah *rating* kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$.

Keterangan:

$\max x_{ij}$ = Nilai terbesar dari setiap kriteria i .

$\min x_{ij}$ = Nilai terkecil dari setiap kriteria i .

x_{ij} = Nilai atribut yang dimiliki dari setiap kriteria.

Benefit = Jika nilai terbesar adalah terbaik.

Cost = Jika nilai terkecil adalah terbaik.

2. Menghitung nilai bobot preferensi pada setiap alternatif.

$$V_i = \sum_{j=1}^n w_j r_{ij}$$

Keterangan:

V_i = Ranking untuk setiap alternatif.

w_j = Nilai bobot kriteria.

r_{ij} = Nilai *rating* kinerja ternormalisasi.

Nilai V_i yang lebih mengindikasikan bahwa alternatif A_i lebih terpilih.

Adapun kelebihan dari metode *Simple Additive Weighting* (Pratiwi, 2016:138), yaitu:

1. Menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses perankingan yang akan menyeleksi alternatif terbaik dari sejumlah



alternatif.

2. Penilaian akan lebih tepat karena didasarkan pada nilai kriteria dari bobot preferensi yang sudah ditentukan.
3. Adanya perhitungan normalisasi matriks sesuai dengan nilai atribut (antara nilai *benefit* dan *cost*).

2.1.3 Pengertian Penyandang Disabilitas

Menurut Kamus Besar Bahasa Indonesia, penyandang merupakan orang yang menyandang (menderita) sesuatu. Sedangkan disabilitas adalah keadaan (seperti sakit atau cedera) yang merusak atau membatasi kemampuan mental dan fisik seseorang.

Menurut Arianti (2018:177), mendefinisikan disabilitas sebagai seseorang yang belum mampu berakomodasi dengan lingkungan sekitarnya sehingga menyebabkan disabilitas.

Dari berbagai definisi diatas dapat disimpulkan bahwa penyandang disabilitas adalah orang yang memiliki keterbatasan mental, fisik, intelektual maupun sensorik yang dapat menjadi penghambat dalam beraktivitas secara penuh dalam aspek kehidupan.

Menurut Ningsih (2014:78), jenis-jenis disabilitas terdiri dari:

1. Disabilitas Fisik
 - a. Tidak dapat melihat (buta);
 - b. Tidak dapat mendengar dan/ kurang dalam mendengar (tuli);
 - c. Tidak dapat berbicara (bisu);
 - d. Cacat tubuh;
2. Disabilitas mental
 - a. Sukar mengendalikan emosi dan sosial;
 - b. Cacat pikiran, lemah daya tangkap.
3. Disabilitas ganda: penderita cacat lebih dari satu kecacatan.



2.1.4 Pengertian Bantuan Sosial

Menurut Arianti (2018:177), bantuan sosial didefinisikan sebagai upaya pemberian bantuan kepada penyandang disabilitas yang tidak mampu yang bersifat tidak tetap untuk meningkatkan taraf kesejahteraan sosialnya.

Bantuan sosial yang diberikan dalam bentuk bantuan langsung berupa uang tunai sebesar Rp2.400.000, yang penyalurannya dilakukan dalam 3 (tiga) tahap selama 1 (satu) tahun. Dalam pemberian bantuan diprioritaskan untuk penyandang disabilitas berat yang kedisabilitasnya sudah tidak dapat direhabilitasi, tidak dapat melakukan aktivitas kehidupannya sehari-hari dan tergantung pada bantuan orang lain atau tidak mampu menghidupi diri sendiri.

2.2 Teori Khusus

2.2.1 *Unified Modelling Language (UML)*

Menurut Sukamto dan Shalahuddin (2018:137), “UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung”.

Pada perkembangan teknologi perangkat lunak, diperlukan adanya bahasa yang digunakan untuk memodelkan perangkat lunak yang akan dibuat dan perlu adanya standarisasi agar orang di berbagai negara dapat mengerti pemodelan perangkat lunak. UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak.

UML hanya berfungsi untuk melakukan pemodelan. Sehingga penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan untuk metodologi berorientasi objek.

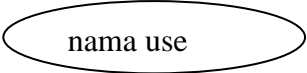
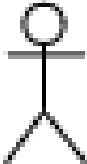

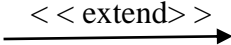
2.2.2 *Use Case Diagram*

Menurut Sukamto dan Shalahuddin (2018:155), “*Use case* atau diagram *use case* merupakan pemodelan untuk kelakuan (*behavior*) sistem informasi yang akan dibuat”. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. *Use case* digunakan untuk

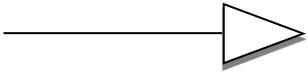
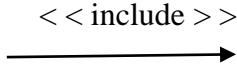


mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu.

Tabel 2.1 Simbol *Use Case Diagram*

| No | Simbol | Deskripsi |
|----|--|---|
| 1 | <p><i>Use case</i></p>  | Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan anta unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja di awal fase nama <i>use case</i> |
| 2 | <p>Aktor / <i>actor</i></p>  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal fase nama aktor |
| 3 | <p>Asosiasi / <i>association</i></p>  | Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor |
| 4 | <p>Ekstensi / <i>extend</i></p>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemrograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> |

Lanjutan **Tabel 2.1** Simbol *Use Case Diagram*



| | | |
|---|--|--|
| 5 | Generalisasi / <i>generalization</i>  | Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya. |
| 6 | Menggunakan / <i>include</i> / <i>uses</i>  | Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> di mana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini |

(Sumber: Sukamto dan Shalahuddin, 2018:155)

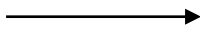

2.2.3 Activity Diagram

Menurut Sukamto dan Shalahuddin (2018:161), “Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2.2 Simbol *Activity Diagram*

| No | Simbol | Deskripsi |
|----|---|--|
| 1 | <i>Start</i> / status awal (<i>Intial State</i>)  | <i>Start</i> atau <i>intial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup |
| 2 | <i>End</i> / status akhir (<i>final state</i>)  | <i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu sistem. |

Lanjutan **Tabel 2.2** Simbol *Activity Diagram*

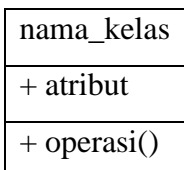


| | | |
|---|---|---|
| 3 | <p><i>Event</i></p>  | <i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin. |
| 4 | <p><i>State</i></p>  | Sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu yang memicu perubahan tersebut. |

(Sumber: Sukamto dan Shalahuddin, 2018:161)

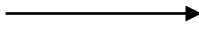
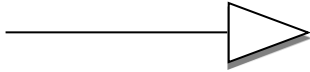


2.2.4 *Class Diagram*

Menurut Sukamto dan Shalahuddin (2018:143), “Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi”.

Tabel 2.3 Simbol *Class Diagram*

| No | Simbol | Deskripsi |
|----|---|---|
| 1 | <p>Kelas</p>  | Kelas pada struktur sistem |
| 2 | <p>Antarmuka / <i>interface</i></p>  <p>name_interface</p> | sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek. |
| 3 | <p>Asosiasi / <i>association</i></p>  | Relasi antarkelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |

Lanjutan **Tabel 2.3** Simbol *Class Diagram*

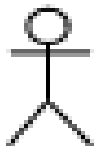
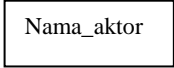
| | | |
|---|---|---|
| 4 | Asosiasi berarah / <i>directed association</i>  | Relasi antarkelas dengan makna kelas yang saat digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> . |
| 5 | Generalisasi  | Relasi antarkelas dengan makna generalisasi-spesialisasi (umum khusus). |
| 6 | Kebergantungan / <i>dependency</i>  | Relasi antarkelas dengan makna kebergantungan antar kelas. |
| 7 | Agregasi / <i>aggregation</i>  | Relasi antarkelas dengan makna semua-bagian (<i>whole-part</i>) |

(Sumber: Sukamto dan Shalahuddin, 2018:143)


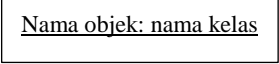

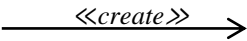
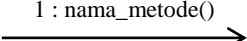
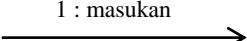
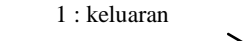
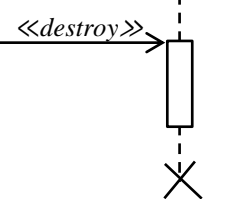
2.2.5 *Sequence Diagram*

Menurut Sukamto dan Shalahuddin (2018:165), “Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek”. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diintansiasi menjadi objek itu dan juga dibutuhkan untuk melihat skenario yang ada pada use case.

Tabel 2.4 Simbol *Sequence Diagram*

| No | Simbol | Deskripsi |
|----|---|---|
| 1 | Aktor  atau  | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor. |

Lanjutan **Tabel 2.4** Simbol *Sequence Diagram*

| | | |
|---|---|--|
| 2 | Garis Hidup/ <i>Lifeline</i>  | Menyatakan kehidupan suatu objek |
| 3 | Objek  | Menyatakan objek yang berinteraksi pesan |
| 4 | Waktu aktif  | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya. |
| 5 | Pesan tipe <i>create</i>  | Menyatakan sebuah objek membuat objek yang lain. Arah panah mengarah pada objek yang dibuat |
| 6 | Pesan tipe <i>call</i>  | Menyatukan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri |
| 7 | Pesan tipe <i>send</i>  | Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim |
| 8 | Pesan tipe <i>return</i>  | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian |
| 9 | Pesan tipe <i>destroy</i>  | Menyatakan suatu objek mengakhiri hidup objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada <i>create</i> maka ada <i>destroy</i> |

(Sumber: Sukamto dan Shalahuddin, 2018:165)

2.3 Teori Program

2.3.1 PHP

Menurut Gunawan (2010:12), “PHP merupakan bahasa pemrograman yang memungkinkan para *web developer* untuk membuat aplikasi *web* yang dinamis dengan cepat dan mudah”.



Menurut Winarno, dkk. (2014:49), “PHP atau PHP *Hypertext Preprocessor* adalah sebuah bahasa pemrograman *web* berbasis *server* (*server-side*) yang mampu memarsing kode PHP dari kode *web* dengan ekstensi *.php*, sehingga menghasilkan tampilan *website* yang dinamis di sisi *client* (*browser*)”.

Dari pengertian di atas dapat disimpulkan bahwa PHP merupakan sebuah bahasa pemrograman yang digunakan untuk membuat aplikasi *web*.

2.3.2 MySQL

Menurut Gunawan (2010:15), MySQL adalah aplikasi atau sistem untuk mengelola *database* atau manajemen data. MySQL bertugas mengatur dan mengelola data-data pada *database*. MySQL menggunakan struktur atau kerangka yang berbentuk tabel. Dalam tabel-tabel itulah data diatur dan dikelompokkan.

Menurut Winarno, dkk. (2014:49), “MySQL merupakan tipe data relasional yang artinya MySQL menyimpan datanya dalam bentuk table-table yang saling berhubungan”.

Dari pengertian di atas dapat disimpulkan bahwa MySQL merupakan suatu aplikasi atau sistem yang digunakan untuk menyimpan dan memanajemen data dalam tabel-tabel yang saling berhubungan.

2.3.3 XAMPP

Menurut Enterprise (2017:93), XAMPP merupakan *tool* yang digunakan untuk membantu kita bekerja menggunakan MySQL *Server*, namun dengan tampilan antarmuka grafis yang lebih ramah untuk siapa pun, terlebih bagi kalangan pemula.

Menurut Gunawan (2010:17), 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 penterjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl.



Dari pengertian di atas dapat disimpulkan bahwa XAMPP merupakan suatu aplikasi *web server* yang digunakan untuk mengelola *MySQL Server* dan membantu dalam pemrograman PHP.

2.4 Referensi Penelitian Sebelumnya

Berdasarkan penelitian yang telah dilakukan oleh Ardhi Bagus Primahudi, Fajar Agustinus Suciono, dan Anang Aris Widodo pada tahun 2016 dengan judul **Sistem Pendukung Keputusan untuk Pemilihan Karyawan dengan Metode *Simple Additive Weighting* di PT. Herba Penawar Alwahida Indonesia** ISSN : 2503-1945, terdapat 5 (lima) kriteria dalam melakukan pemilihan calon karyawan yaitu pendidikan terakhir, IPK, pengalaman kerja, akreditasi, status perkawinan, kesesuaian program studi, dan usi yang diberi bobot tersendiri serta dilakukan perhitungan menggunakan metode *Simple Additive Weighting* yaitu melakukan normalisasi dengan nilai himpunan pada setiap kriteria yang telah ditentukan. Kemudian melakukan proses perangkingan dengan mengalikan nilai hasil normalisasi dari setiap kriteria dengan bobot kriteria dan menjumlahkan semua hasil perkalian. Dari proses tersebut didapatkan hasil perangkingan alternatif terbaik dari sejumlah alternatif.

Menurut penelitian yang telah dilakukan oleh Nalsa Cintya Resti pada tahun 2017 yang berjudul **Penerapan Metode *Simple Additive Weighting* (SAW) pada Sistem Pendukung Keputusan Pemilihan Lokasi untuk Cabang Baru Toko Pakan UD. Indo Multi Fish** P-ISSN : 2580-409X, E-ISSN : 2549-6824, pemilihan lokasi yang digunakan untuk pembangunan cabang baru UD. Indo Multi Fish dilakukan dengan memberikan 6 (enam) kriteria yaitu harga tanah untuk lokasi, jarak dengan toko lama, jarak dengan gudang penyimpanan, tingkat persaingan dengan toko lain, nilai investasi 3 (tiga) tahun kedepan, dan tingkat masyarakat yang memelihara ikan. Setelah menghitung masing-masing nilai alternatif menggunakan rumus dengan mengkonverensikan antara nilai *utility* dengan nilai normalisasi bobot kriteria sehingga diperoleh nilai terakhir. Nilai dengan peringkat tertinggi merupakan daerah yang dapat direkomendasikan untuk dibuka cabang baru.



Berdasarkan penelitian yang telah dilakukan oleh Elsen Ronando dan Enny Indasyah pada tahun 2017 yang berjudul **Sistem Pendukung Keputusan untuk Penentuan Penerima Bantuan Langsung Masyarakat PNPM Mandiri Menggunakan *Simple Additive Weighting*** P-ISSN : 2088-3943, E-ISSN : 2580-9741, dengan memberikan kriteri kesesuaian terhadap ketentuan PNPM, mendesak untuk dilaksanakan, lebih bermanfaat untuk kelompok miskin, bila dikerjakan masyarakat, tingkat keberhasilan pengembangan dan berkelanjutan, dan didukung oleh sumber daya yang ada yang diberi bobot tertentu maka didapatkan nilai dan *ranking* terbesar dalam pemilihan daerah calon penerima bantuan langsung.

Menurut penelitian yang telah dilakukan oleh Reza Fauzan, Yoenie Indrasary, dan Nonik Muthia pada tahun 2017 yang berjudul **Sistem Pendukung Keputusan Penerima Beasiswa Bidik Misi di POLIBAN dengan Metode SAW berbasis *Web*** P-ISSN : 2527-1682, E-ISSN : 2527-9165, didapatkan daftar mahasiswa yang mendapat beasiswa Bidik Misi dari proses perhitungan, yaitu penjumlahan dari matriks ternormalisasi dengan bobot per kriteria yang menunjukkan peringkat alternatif penerima beasiswa yang paling mendekati kriteria. Adapun kriteria acuan dalam pemilihan mahasiswa penerima beasiswa Bidik Misi adalah penghasilan orang tua, jumlah tanggungan, rata-rata nilai *raport* semester 4-5, bukti rekening listrik, dan bukti pembayaran PBB. Nilai dengan peringkat tertinggi merupakan mahasiswa yang dapat direkomendasikan untuk mendapat beasiswa Bidik Misi

Berdasarkan penelitian tahun 2018 yang ditulis oleh Irfan Fandinata dan Budi Serasi Ginting dengan judul **Sistem Pendukung Keputusan Pemilihan Bibit Unggul Tanaman Jambu Madu Menggunakan Metode SAW** ISSN : 2448-9712, dalam penentuan bibit jambu madu yang unggul dapat dilihat dari beberapa kriteria yaitu tekstur tanah, suhu, ketahanan, dan kelembapan. Penelitian ini menggunakan metode SAW yang berdasarkan kriteria dan bobot yang telah ditentukan sehingga didapatkan *ranking* bibit jambu madu yang paling bagus diantara bibit jambu madu yang ada.