

BAB I

PENDAHULUAN

A. Latar Belakang Masalah

Dalam kehidupan, manusia selalu dihadapkan pada beberapa pilihan pengambilan keputusan yang tepat akan sangat berpengaruh pada kehidupan kita kedepannya. Permasalahan pengambilan keputusan juga dialami oleh siswa yang ingin melanjutkan sekolahnya kejenjang yang lebih tinggi. Banyak hal yang perlu dipertimbangkan dalam pemilihan jurusan yang sesuai.

Khususnya para siswa SMA kelas X (sepuluh) yang akan melanjutkan pendidikan di tingkat selanjutnya harus memutuskan pilihan jurusan. Dan ini merupakan sesuatu yang cukup sulit diputuskan oleh kebanyakan siswa. Hal ini dikarenakan masih menggunakan cara manual dan belum ada metode yang objektif untuk memutuskan dengan cepat, berdasarkan data yang ada.

Di SMAN-1 Murung Raya selama ini dalam pemilihan jurusan selalu ditentukan oleh wali kelas masing-masing menggunakan 15 besar peringkat nilai rata-rata tertinggi siswa, maka siswa tersebut akan dianjurkan ke jurusan IPA lalu peringkat 16 ke bawah akan masuk ke IPS. Permasalahan tersebut dapat dijawab menggunakan *Simple Additive Weighting (SAW)* dan *Technique For Order Preference By Similarity (TOPSIS)* adalah proses membentuk skor secara numerik untuk menyusun ranking setiap alternatif keputusan berbasis pada bagaimana sebaiknya alternatif itu dicocokkan dengan kriteria pembuat keputusan.

Dengan permasalahan diatas, maka perlunya sebuah sistem pendukung keputusan untuk membantu pihak sekolah dan siswa menentukan jurusan yang tepat, sesuai dengan minat, bakat, dan kemampuan akademik, maka penulis ingin merancang sebuah “sistem pendukung keputusan seleksi pemilihan jurusan pada SMAN-1 Murung di Kabupaten Murung Raya menggunakan *Simple Additive Weighting (SAW)* dan *Technique for order preference by similarity (TOPSIS)*”. agar mempermudah sekolah dalam memilih jurusan yang tepat bagi siswanya.

B. Rumusan Masalah

Berdasarkan latar belakang masalah yang dikemukakan diatas, maka yang menjadi permasalahan dalam penyusunan proposal skripsi ini adalah bagaimana merancang dan membangun suatu “Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada SMAN-1 Murung Raya Menggunakan Metode *Analytical Hierarchy Process (SAW)* dan *Technique For Order Preference By Similarity (TOPSIS)*” Berbasis *CI*.

C. Batasan Masalah

Agar pembahasan tidak meluas maka penulis hanya akan membahas masalah :

1. Menggunakan metode *Simple Additive Weighting (SAW)* dan *Technique For Order Preference By Similarity (TOPSIS)*.
2. Data – data yang digunakan dalam sistem ini yaitu data siswa, data kriteria, data nilai dan data pengguna.
3. Penelitian dan penjurusan dilakukan hanya untuk penjurusan siswa-siswi kelas X di Sekolah Menengah Atas Negeri 1 Murung Raya.

4. Outputnya adalah hasil keputusan yang hanya menentukan jurusan IPA atau IPS.
5. Kriteria yang dijadikan acuan dalam pengambilan keputusan yaitu :
 - a. Nilai Rata – Rata Raport.
 - b. Nilai Pelajaran.
 - c. Minat dan Bakat
6. Sistem Pendukung Keputusan dibuat dengan bahasa pemrograman *PHP* dan *MySQL* sebagai *datasenya*.

D. Tujuan dan Manfaat

Agar tidak menyimpang dari tujuan dan tidak meluasnya cakupan pembahasan maka batasan masalah adalah :

1. Tujuan :

Tujuan yang akan dicapai dalam penelitian ini adalah :

- a. Untuk memudahkan Sekolah dan Siswa-Siswi dalam menentukan Jurusan yang akan diambil.
- b. Penggabungan metode *SAW (Simple Additive Weighting)* dan *TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)* untuk menentukan jurusan.

2. Manfaat :

- a. Bagi SMAN-1 Murung

Membantu petugas kesiswaan pengolahan data akademik dan juga membantu wali kelas dalam menentukan jurusan bidang minat yang terdiri dari IPA dan IPS.

- b. Bagi penulis

Untuk meningkatkan kemampuan dan pengetahuan mengenai sistem pendukung keputusan.

c. Bagi STMIK Palangka Raya

Bisa menambah pengetahuan bagi mahasiswa dan dapat dijadikan bahan acuan penelitian-penelitian untuk pembuatan laporan yang lebih lanjut.

E. Metode Penelitian

1. Metode Pengumpulan Data

Adapun metode yang digunakan dalam untuk memperoleh/mengumpulkan data dan informasi dalam menyelesaikan penyusunan proposal Skripsi ini adalah :

a. Observasi

Observasi adalah suatu metode yang digunakan untuk mendapatkan data dan informasi dengan cara mengamati secara langsung ke SMAN-1 Murung.

b. Wawancara

Metode pengumpulan data ini, penulis mengumpulkan data dan informasi yang diperlukan dengan melakukan tanya jawab secara langsung kepada pihak-pihak yang terkait yang berbubungan dengan permasalahan dalam penentuan jurusan yang akan diambil di SMAN-1 Murung.

c. Kepustakaan

Metode yang digunakan penulis untuk memperoleh data atau mengumpulkan data dengan cara membaca dan mempelajari

buku-buku atau literatur dan dokumen dari sekolah SMAN-1 Murung Raya terkait dengan penulis proposal tugas akhir ini.

2. Metode Pengembangan Perangkat Lunak

Pengembangan perangkat lunak sistem pendukung keputusan pemilihan jurusan pada SMAN-1 Murung menggunakan model pengembangan *Waterfall Model* (Model Air Terjun).

Menurut Jogiyanto (2010:59) dalam bukunya dikatakan tahapan utama dari model ini adalah :

a. Analisis dan Definisi Persyaratan

Analisis dan definisi persyaratan yang akan diimplementasikan kedalam aplikasi seleksi pemilihan jurusan yang akan penulis kembangkan nantinya adalah :

- 1) Aplikasi ini nantinya akan membantu guru dan siswa-siswi dalam pemilihan jurusan yang akan diambil sesuai dengan kemampuan dan bakat.
- 2) Aplikasi ini dibuat dengan menggunakan metode pengambilan keputusan yaitu SAW (*Simple Additive Weighting*) dan TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*).

b. Perancangan Sistem dan Perangkat Lunak

Proses perancangan yang akan menterjemahkan syarat kebutuhan sebuah perangkat lunak yang dapat diperkirakan sebelumnya, *listing* dan perancangan sistem yang dibuat harus sesuai dengan kebutuhan *user* yang berfungsi untuk membantu

dalam pembuatan aplikasi pemilihan jurusan. Dan proses ini perfokus pada struktur data, arsitektur perangkat lunak, dan representasi *interface*.

c. Implementasi dan Pengujian Unit

Implementasi dan pengujian unit aplikasi yang dikembangkan oleh penulis dilakukan kepada bagian sekolah, guru kelas, dan siswa-siswi di SMAN-1 Murung Raya.

d. Integrasi dan Pengujian Sistem

Integrasi dan pengujian sistem aplikasi yang dikembangkan oleh penulis menggunakan metode pengujian *Black-Box*.

e. Operasi dan Pemeliharaan

Tahap ini mencakup pada proses pemeliharaan dalam manajemen operasinal dalam sistem, sehingga fasilitas yang dimiliki harus dijaga agar digunakan dengan baik sehingga proses operasional dalam seleksi pemilihan jurusan dapat berjalan dengan optimal. Dan juga berguna untuk pengembangan sistem agar mudah untuk dikembang.

F. Sistematika Penulisan

Secara garis besar proposal tugas akhir ini terdiri dari 5 bab dengan beberapa sub bab. Agar mendapat arah dan gambaran yang jelas mengenai hal yang tertulis, berikut ini sistematika penulisan secara lengkap.

Dalam penulisan proposal tugas akhir ini dapat dijelaskan beberapa istilah sebagai berikut :

1. Sistem Pendukung Keputusan

Sistem pendukung keputusan adalah sistem yang dibangun menyelesaikan berbagai masalah yang bersifat atau organisasi perusahaan yang dirancang untuk mengembangkan efektivitas dan produktivitas para manajer untuk menyelesaikan masalah dengan bantuan teknologi komputer.

2. Metode *Simple Additive Weighting (SAW)*

Simple Additive Weighting (SAW) adalah salah satu metode dalam sistem pendukung keputusan yang cukup familiar yang mendukung pengambilan keputusan dengan cara mebobotkan semua kriteria dan alternatif dan mendapatkan nilai preferensi yang tepat.

3. Metode *Technique For Order Preference By Similarity (TOPSIS)*

Technique For Order Preference By Similarity (TOPSIS) adalah metode yang banyak digunakan untuk menyelesaikan pengambilan keputusan secara praktis dan alternatif terbaik yang memiliki jarak terpendek dari solusi ideal positif dan negatif.

4. Penjurusan

Penjurusan merupakan salah satu proses penempatan atau penyaluran dalam pemilihan program pengajaran para siswa di SMA. ketepatan dalam memilih jurusan dapat menentukan keberhasilan belajar siswa sebaliknya, kesempatan yang sangat baik siswa akan hilang karena kekurangtepatan dalam menentukan jurusan. Dalam kurikulum KTSP, penjurusan di SMA akan di mulai pada akhir semester 2 kelas X (sepuluh).

BAB II

LANDASAN TEORI

A. Tinjauan Pustaka

1. Kajian Penelitian yang Relevan

Untuk menghindari duplikasi dari pihak lain, maka penulis melakukan penelusuran terhadap penelitian – penelitian terdahulu. Dari hasil penilitan terdahulu, diperoleh beberapa penelitian yang pernah ada yang berkaitan dengan penelitian yang penulis lakukan, yaitu :

- a. Dalam penelitian yang dilakukan oleh Sucipto (2016) yang berjudul Sistem Pendukung Keputusan Penentuan Jurusan Sekolah Menengah Atas Dengan Metode SAW. Dalam penelitian ini Sistem Pendukung Keputusan (SPK) Pemilihan Jurusan di SMA Tren Sains Tebuireng Jombang dengan menggunakan metode SAW telah berhasil dibangun untuk menghasilkan keputusan berupa rekomendasi jurusan yang terpilih untuk siswa. Semakin banyak sampel data yang digunakan maka semakin tinggi pula tingkat validitas perhitungan yang dihasilkan. Pemberian skala konversi dan bobot preferensi dari setiap bobot kriteria mempengaruhi penilaian dan hasil perhitungan SAW.
- b. Dalam penelitian yang dilakukan oleh Hadi (2015) yang berjudul Sistem Pendukung Keputusan Penjurusan Metode Topsis (Studi Kasus Man Batam). Dalam hal ini Sistem pendukung keputusan yang telah dibangun mampu menentukan jurusan pada siswa MAN berdasarkan nilai raport, nilai tes minat, nilai tes bakat, dan nilai tes

intelegensi. Hasil perbandingan penjurusan sekolah dengan system pendukung keputusan metode *TOPSIS* memiliki tingkat persentase kebenaran 73%.

- c. Dalam penelitian yang dilakukan oleh Dewi (2013) yang berjudul Sistem Pendukung Keputusan menggunakan metode AHP dan TOPSIS (Studi kasus : Pemilihan *Handphone*). Penelitian ini membahas tentang membuat suatu sistem pendukung keputusan untuk menentukan *handphone* yang sesuai dengan keinginan pelanggan. Dari penelitian ini dapat disimpulkan hasil penggabungan metode *SAW* dan *TOPSIS* terbukti mampu memberikan prioritas *handphone* yang tepat sesuai dengan kriteria, subkriteria dan alternatif yang diinginkan karena cepat dan akurat.
- d. Dalam penelitian yang dilakukan oleh Satria (2011) yang berjudul Sistem Pendukung Keputusan Penentuan Pemberian Kredit Angsuran Sistem Fidusia (Kreasi) Dengan Menggunakan Metode *Saw (Simple Additive Weighting Method)* Dan Metode *Topsis (Technique For Order Preference By Similarity To Ideal Solution)*. (Studi kasus : PERUM Pegadaian Kantor Cabang Marpoyan Pekanbaru). Penelitian ini membahas tentang sistem yang dapat menentukan pemberian *KREASI* yang berhak diberikan bantuan dalam pengembangan usaha mikro. Dari penelitian ini dapat disimpulkan hasil penggabungan metode *SAW* dan *TOPSIS* telah berhasil menghasilkan keputusan yang lebih objektif,

terkomputerisasi dan mengurangi terjadinya *human error* karena nilai yang dihasilkan cukup akurat.

Berdasarkan hasil penelusuran terhadap penelitian diatas, untuk lebih jelasnya dapat dilihat pada Tabel 1 :

Tabel 1. Kajian Penelitian yang Relevan

No	Nama	Judul	Metode	Hasil
1.	Sucipto (2016)	Sistem Pendukung Keputusan Penentuan Jurusan Sekolah Menengah Atas Dengan Metode SAW	SAW	Berhasil dibangun untuk menghasilkan keputusan berupa rekomendasi jurusan yang terpilih untuk siswa dan dapat membantu pihak sekolah dalam penjurusan.
2.	Hadi (2015)	Sistem Pendukung Keputusan Penjurusan Metode Topsis (Studi Kasus Man Batam)	TOPSIS	Membantu menentukan jurusan pada siswa MAN berdasarkan nilai raport, nilai tes minat, nilai tes bakat, dan nilai tes intelegensi. Hasil perbandingan penjurusan sekolah dengan sistem pendukung keputusan metode TOPSIS tingkat kebenaran 73%.
Kajian Pustaka Yang Relevan (Lanjutan)				
3.	Dewi (2013)	Sistem Pendukung Keputusan menggunakan metode AHP dan TOPSIS (Studi kasus : Pemilihan Handphone)	SAW dan TOPSIS	Membantu konsumen menentukan pilihan <i>handphone</i> dengan tepat sesuai dengan keinginan, kegunaan dan anggarannya

No	Nama	Judul	Metode	Hasil
4.	Satria (2011)	Sistem Pendukung Keputusan Penentuan Pemberian Kredit Angsuran Sistem Fidusia (Kreasi) Dengan Menggunakan Metode Saw (<i>Simple Additive Weighting Method</i>) Dan Metode <i>Topsis (Technique For Order Preference By Similarity To Ideal Solution)</i> . (Studi kasus : PERUM Pegadaian Kantor Cabang Marpoyan Pekanbaru).	SAW dan TOPSIS	Membantu menentukan pemberian KREASI yang berhak diberikan bantuan dalam pengembangan usaha mikro.

Sehingga dari beberapa penelitian yang dilakukan seperti diatas cukup berbeda dari pembuatan sistem pendukung keputusan yang akan dibuat oleh penulis karena menggunakan penggabungan metode *AHP* dan *TOPSIS*. Sistem pendukung keputusan ini menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai basis datanya

2. Kajian Teori

a. Sistem Pendukung Keputusan (SPK)

Menurut Alter dalam Kusri (2007:15-16) "*Decision Support System (DSS)* merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data".

Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semi terstruktur dan situasi yang

tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

DSS biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. *DSS* yang seperti itu disebut aplikasi *DSS*. Aplikasi *DSS* digunakan dalam pengambilan keputusan. Aplikasi *DSS* menggunakan *CBIS* (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasikan, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Aplikasi *DSS* menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambilan keputusan.

Menurut Kusri (2007:30-31) Saat melakukan pemodelan dalam pembangunan *DSS* dilakukan langkah – langkah sebagai berikut :

- 1) Studi Kelayakan (*Intelligence*)

Pada langkah ini, sasaran ditentukan pencarian prosedur, pengumpulan data, identifikasi masalah, identifikasi kepemilikan masalah, klasifikasi masalah, hingga akhirnya terbentuk sebuah pernyataan masalah. Kepemilikan masalah berkaitan dengan apa yang akan dibangun oleh *DSS* dan apa tugas dari bagian tersebut sehingga model tersebut bisa relevan dengan kebutuhan si pemilik masalah.

2) Perancangan (*Design*)

Pada tahapan ini akan diformulasikan model yang akan digunakan dan kriteria yang ditentukan. Setelah itu, dicari alternative model yang bisa menyelesaikan permasalahan tersebut. Langkah selanjutnya adalah memprediksi keluaran yang mungkin. Kemudian, ditentukan variabel-variabel model.

3) Pemilihan (*Choice*)

Setelah pada tahap design ditentukan berbagai alternatif model beserta variabel – variabelnya, pada tahapan ini akan dilakukan pemilihan modelnya, termasuk solusi dari model tersebut. Selanjutnya dilakukan analisis sensitivitas, yakni dengan mengganti beberapa variabel.

4) Membuat *Decision Support System (DSS)*

Setelah menentukan modelnya, berikutnya adalah mengimplementasikannya dalam aplikasi *DSS*.

b. *SAW (Simple Additive Weighting Method)*

Menurut Kusumadewi, dkk (2006:74-75) metode *SAW* sering juga dikenal dengan istilah penjumlahan berbobot metode ini juga sering digunakan untuk menyeleksi masalah MADM. Konsep dasar metode *SAW* adalah mencari penjumlahan terbobot dari rating kinerja pada setiap alternatif pada semua atribut. Metode *SAW* membutuhkan proses normalisasi matriks keputusan (x) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$r_{ij} \begin{cases} \frac{x_{ij}}{\max x_{ij}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\min x_{ij}}{x_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad (1)$$

Keterangan :

r_{ij} rating kinerja ternormalisasi

$\max x_{ij}$ nilai maksimum dari setiap baris dan kolom

$\min x_{ij}$ nilai minimum dari setiap baris dan kolom

x_{ij} baris dan kolom dari matriks

Dengan r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j , $i = 1, 2, \dots, m$ dan $j = 1, 2, \dots, n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai :

$$v_i = \sum_{j=1}^n w_j r_{ij} \quad (2)$$

Keterangan :

v_i Nilai akhir dari alternatif

w_j Bobot yang telah ditentukan

r_{ij} Normalisasi matriks

Nilai v_i yang lebih besar mengindikasikan bahwa alternative A_i lebih terpilih. Langkah penyelesaian metode SAW antara lain :

- 1) Menentukan kriteria-kriteria yang akan dijadikan acuan dalam pengambilan keputusan, yaitu C_i .
- 2) Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
- 3) Membuat matriks keputusan berdasarkan kriteria(C_i), kemudian melakukan normalisasi matriks berdasarkan

persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi R.

4) Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vektor bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (A_i) sebagai solusi.

c. Metode *Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)*

Menurut Kusuma dewi (2006:87) *TOPSIS* adalah salah satu metode pengambilan keputusan multikriteria atau alternatif pilihan yang merupakan alternatif yang mempunyai jarak terkecil dari solusi ideal positif dan jarak terbesar dari solusi ideal negatif dari sudut pandang geometris dengan menggunakan jarak *Euclidean*. Namun alternatif yang mempunyai jarak terkecil dari solusi ideal positif, tidak harus mempunyai jarak terbesar dari solusi ideal negatif.

Maka dari itu, *TOPSIS* mempertimbangkan keduanya, jarak terhadap solusi ideal positif dan jarak terhadap solusi ideal negatif secara bersamaan. Solusi optimal dalam metode *TOPSIS* didapat dengan menentukan kedekatan relatif suatu alternatif terhadap solusi ideal positif. *TOPSIS* akan merangking alternatif berdasarkan prioritas nilai kedekatan relatif suatu alternatif terhadap solusi ideal positif. Alternatif-alternatif yang telah dirangking kemudian

dijadikan sebagai referensi bagi pengambil keputusan untuk memilih solusi terbaik yang diinginkan.

Berikut adalah langkah – langkah metode *TOPSIS* :

- 1) Membangun sebuah matriks keputusan.

Matriks keputusan X mengacu terhadap m alternatif yang akan dievaluasi berdasarkan n kriteria. Matriks keputusan X dapat dilihat sebagai berikut :

$$X = \begin{matrix} & x_1 & x_2 & x_3 & \dots & x_n \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_m \end{matrix} & \begin{bmatrix} x_{11} & x_{12} & x_{31} & \dots & x_{n1} \\ x_{12} & x_{22} & x_{32} & \dots & x_{n2} \\ x_{13} & x_{32} & x_{33} & \dots & x_{n3} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & \dots & x_{mn} \end{bmatrix} \end{matrix} \quad (3)$$

Keterangan :

a_i ($i = 1, 2, 3, \dots, m$) adalah alternatif-alternatif yang mungkin,

x_j ($j = 1, 2, 3, \dots, n$) adalah atribut dimana performansi alternatif diukur, x_{ij} adalah performansi alternatif a_i dengan acuan atribut.

- 2) Membuat matriks keputusan yang ternormalisasi.

Persamaan yang digunakan untuk mentransformasikan setiap elemen x_{ij} adalah :

$$r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^m x_{ij}^2}} \quad (4)$$

dengan $i = 1, 2, 3, \dots, m$; dan $j = 1, 2, 3, \dots, n$;

Keterangan :

r_{ij} adalah elemen dari matriks keputusan yang ternormalisasi R,

x_{ij} adalah elemen dari matriks keputusan X.

3) Membuat matriks keputusan yang ternormalisasi terbobot.

Dengan bobot $w_j = (w_1, w_2, w_3, \dots, w_n)$, dimana w_j adalah bobot dari kriteria ke - j dan $\sum_{j=1}^n w_j = 1$, maka normalisasi bobot matriks V adalah :

$$V_{ij} = w_j r_{ij}$$

dengan $i = 1, 2, 3, \dots, m$; dan $j = 1, 2, 3, \dots, n$. (5)

Keterangan :

v_{ij} adalah elemen dari matriks keputusan yang ternormalisasi terbobot V.

w_j adalah bobot kriteria ke-j

r_{ij} adalah elemen dari matriks keputusan yang ternormalisasi R.

4) Menentukan matriks solusi ideal positif dan solusi ideal negatif.

Solusi ideal positif dinotasikan A^+ , sedangkan solusi ideal negatif dinotasikan A^- . Berikut ini adalah persamaan dari A^+ dan A^- :

$$\begin{aligned} A^+ &= \{(\max v_{ij} | j \in J), (\min v_{ij} | j \in J), i = 1, 2, 3, \dots, m\} \\ &= \{v_1^+, v_2^+, v_3^+, \dots, v_n^+\} \end{aligned} \quad (6)$$

$$\begin{aligned} A^- &= \{(\min v_{ij} | j \in J), (\max v_{ij} | j \in J), i = 1, 2, 3, \dots, m\} \\ &= \{v_1^-, v_2^-, v_3^-, \dots, v_n^-\} \end{aligned} \quad (7)$$

$J = \{j = 1, 2, 3, \dots, n \text{ dan } J \text{ merupakan himpunan kriteria keuntungan (benefit criteria)}\}$.

$J' = \{j = 1, 2, 3, \dots, n \text{ dan } J' \text{ merupakan himpunan kriteria biaya (cost criteria)}\}$

Keterangan:

v_{ij} adalah elemen dari matriks keputusan yang ternormalisasi terbobot V ,

v_j^+ ($j = 1, 2, 3, \dots, n$) adalah elemen matriks solusi ideal positif,

v_j^- ($j = 1, 2, 3, \dots, n$) adalah elemen matriks solusi ideal negatif.

5) Menghitung separasi.

S^+ adalah jarak alternative dari solusi ideal positif didefinisikan sebagai :

$$Si^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2}, \text{ dengan } i = 1, 2, 3, \dots, m \quad (8)$$

S^- adalah jarak alternative dari solusi ideal negatif didefinisikan sebagai :

$$Si^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2}, \text{ dengan } i = 1, 2, 3, \dots, m \quad (9)$$

Keterangan:

Si^+ adalah jarak alternative ke-I dari solusi ideal positif,

Si^- adalah jarak alternative ke-I dari solusi ideal negatif,

v_{ij} adalah elemen dari matriks keputusan yang ternormalisasi terbobot V ,

v_j^+ adalah elemen matriks solusi ideal positif,

v_j^- adalah elemen matriks solusi ideal negatif.

6) Menghitung kedekaan terhadap solusi ideal positif.

Kedekatan relatif dari setiap alternative terhadap solusi ideal positif dapat dihitung dengan menggunakan persamaan berikut:

$$c_i^+ = \frac{s_i^-}{s_i^- + s_i^+}, 0 \leq c_i^+ \leq 1 \quad (10)$$

dengan $i = 1, 2, 3, \dots, m$

Keterangan :

c_i^+ adalah kedekatan relatif dari alternative ke-I terhadap solusi ideal positif,

s_i^+ adalah jarak alternatif ke-I dari solusi ideal positif,

s_i^- adalah jarak alternative ke-I dari solusi ideal negatif.

7) Merangking alternatif.

Alternative diurutkan dari nilai C^+ terbesar ke nilai terkecil. Alternatif dengan nilai C^+ terbesar merupakan solusi terbaik.

d. Pengertian *Framework Codeigniter*

Menurut Basuki (2014:12) *Framework* adalah paket berisi fungsi-fungsi yang biasa digunakan dalam pembuatan aplikasi. Beberapa contoh fungsi standar yang biasa ada sebuah *Framework* misalnya : *email, paging, kalender, tanggal, bahasa, upload file, session, validasi form, tabel, manipulasi gambar, text, string, captcha, enkripsi, proteksi terhadap XSS, security*, dan lain-lain. Fungsi-fungsi tersebut dapat segera digunakan dengan cara memanggilnya dengan pada program, tentu saja cara

memanggilnya tergantung dari *framework* yang digunakan. Jadi, *programmer* tidak perlu lagi membuat fungsi-fungsi tersebut dari awal.

Metode yang digunakan oleh *Framework Codeigniter* disebut *Model-View-Controller (MVC)*. *MVC* memisahkan antara logika pemrograman dengan presentasi. Hal ini dapat terlihat dengan adanya minimalisir *script* presentasi (*HTML, CSS, JavaScript, dan sebagainya*) yang dipisahkan dari *PHP (Hypertext Preprocessor) script*. Didalam folder *CodeIgniter*, *MVC* dapat kita temukan dalam folder *application*. *CodeIgniter* juga menjadi salah satu *Framework* pilihan yang memungkinkan *developer* untuk membuat sebuah aplikasi *web Codeigniter* dilengkapi dengan fitur-fitur yang cukup banyak. Fitur-fitur utama *Codeigniter* diantaranya:

1) Mendukung beberapa macam *database*

Mendukung “*active record*” untuk mengakses dengan karakter pengembangan *RAD (Rapid Applikation Preprocessor)*, yang memungkinkan untuk digunakan dan dikembangkan menjadi aplikasi lain yang lebih kompleks. *Codeigniter* terdiri dari file-file pustaka, kelas-kelas, dan infrastruktur *run-time* yang terinspirasi oleh *framework Ruby on Rails*.

2) Fitur-fitur *Codeigniter*

a) *Database*

b) *Validasi Form*

- c) *Library* untuk mengolah gambar
- d) *File Upload*
- e) *Data Encryption* (Enkripsi data)
- f) *Full Page Caching*
- g) *Error Logging*
- h) Kalender
- i) *Unit Testing*
- j) Memiliki banyak *Helper* untuk berbagai keperluan

3. *Software/Tools* yang digunakan

a. *MySQL*

Menurut Anhar (2010:45) “*MySQL* adalah salah satu *Database Management System* (DBMS) dari sekian banyak DBMS seperti *Oracle*, *MySQL*, *Postagre SQL* dll. *MySQL* berfungsi untuk mengolah database menggunakan bahasa *SQL*. *MySQL* bersifat *open source* sehingga kita bisa menggunakannya secara gratis”.

MySQL adalah salah satu jenis *database server* yang banyak digunakan dan sangat terkenal. Kepopulerannya disebabkan *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses *database* yang terbagi menjadi tiga bagian, yaitu *DDL*, *DML* dan *DCL*. *MySQL* memiliki beberapa kelebihan yaitu :

1) *Portability* (Portabilitas)

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti *windows*, *Linux*, *FreeBSD*, *Solaris* dan lain-lain.

2) *Open Source* (Sumber Terbuka)

MySQL didistribusikan secara *open source* (gratis), dibawah lisensi *GPL* sehingga dapat digunakan cuma-Cuma.

a) Multi User

MySQL dapat digunakan oleh beberapa user dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

b) *Performance Tuning* (Penyesuaian Kinerja)

MySQL memiliki kecepatan yang menakjubkan dalam menangani *query* sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.

c) *Coloumn Types* (Jenis Kolom)

MySQL memiliki tipe kolom yang sangat kompleks, seperti *integer*, *double*, *char*, *text*, *dated* dan lain-lain.

d) *Command and Function* (Perintah dan Fungsi)

MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah *select* dan *where* dalam *query*.

e) *Security* (Keamanan)

MySQL memiliki beberapa lapisan sekuritas seperti level *subnetmask* nama *host*, dan izin akses *user* dengan sistem perizinan yang mendetail serta *password* terenkripsi.

f) *Scability and Limits* (Skabilitas dan Batas)

MySQL mampu menangani database dalam skala besar, dengan jumlah *records* lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

g) *Connectivity* (Konektifitas)

MySQL dapat melakukan koneksi dengan *clients* menggunakan protokol *TCP/IP*, *Unix socket (UNIX)* atau *Named Pipes (NT)*.

h) *Localisation* (Lokasi)

MySQL dapat mendeteksi pesan kesalahan pada *client* dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

i) *Interface* (Antarmuka)

MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi *API (Application Programming Interface)*.

j) *Clients and Tools* (Klien dan Alat)

MySQL dilengkapi dengan berbagai *tools* yang dapat digunakan untuk administrasi database dan pada setiap *tool* yang ada disertakan petunjuk *online*.

k) Struktur Tabel

MySQL memiliki struktur tabel yang lebih *fleksibel* dalam menangani *Alter Table*, dibandingkan *database* lainnya semacam *PosgreeSQL* ataupun *Oracle*.



Gambar 1. Logo MySQL
Sumber : Anhar, 2010

b. *XAMPP*

Xampp adalah *software (basic package)* yang merupakan *freeware* yang tersedia berbagai *platform*. *Xampp* Merupakan salah satu web server yang lengkap, *Software* yang sejenis adalah *AppServ, Exitami, PWS/IIS (Microsoft)* dan lain-lain.(Jayan, 2010).



Gambar 2. *XAMPP*
Sumber : Jayan, 2010

Adapun isi *Xampp* (Ver.1.7.0) adalah sebagai berikut :

- 1) *Apache 2.2.11*
- 2) *MySQL 5.1.30*
- 3) *PHP 5.2.8 + PHP 4.4.7*
- 4) *PHP-Switch win32 1.0*
- 5) *phpMyAdmin 3.1.1*
- 6) *FileZilla*

c. *UML*

Menurut Nugroho dalam bukunya (2010:6) dikatakan bahwa *UML* adalah salah satu pemodelan untuk sistem atau perangkat lunak yang berparadikma berorientasi objek.

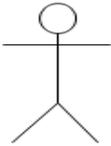
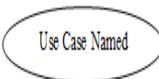
UML merupakan suatu kumpulan teknik terbaik yang telah terbukti sukses dalam pemodelan sistem yang besar dan kompleks. *UML* tidak hanya digunakan dalam proses pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan. Untuk membuat suatu model, *UML* memiliki diagram grafis yang diberi nama berdasarkan sudut pandang yang berbeda-beda terhadap sistem dalam proses analisis atau rekayasa.

Menurut Arbain (2013:38) *UML Behavioral Diagram* terdiri dari:

a. *Use-case diagram*

Use case Diagram digunakan untuk memodelkan fungsionalitas-fungsionalitas sistem atau perangkat lunak dilihat dari pengguna yang ada di luar sistem (yang sering dinamakan sebagai *actor*). Untuk lebih jelasnya dapat dilihat pada Tabel 1 dibawah ini.

Tabel 1. Notasi pada *Use Case Diagram*
(Sumber Nugroho, 2008)

Relasi	Fungsi	Notasi
1	2	3
<i>Actor</i>	<i>Actor</i> adalah pengguna sistem. <i>Actor</i> tidak terbatas hanya untuk manusia saja, jika sebuah sistem berkomunikasi dengan aplikasi lain dan membutuhkan <i>input</i> atau memberikan <i>output</i> , maka aplikasi tersebut juga bisa dianggap sebagai <i>actor</i> .	
<i>Use Case</i>	<i>Use case</i> digambarkan sebagai lingkaran elips dengan nama <i>use case</i> dituliskan didalam <i>elips</i>	

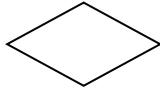
Relasi	Fungsi	Notasi
1	2	3
	tersebut.	
<i>Association</i>	Asosiasi digunakan untuk menghubungkan <i>actor</i> dengan <i>use case</i> . Asosiasi digambarkan dengan sebuah garis yang menghubungkan antara <i>actor</i> dengan <i>use case</i> .	

b. *Activity Diagram*

Activity diagram menggambarkan sebagai alur aktivitas dalam *system* yang sedang dirancang, bagaimana masing-masing alur berawal, *decision* yang memungkinkan terjadi, dan bagaimana mereka berakhir. Untuk lebih jelasnya dapat dilihat pada Tabel 2 dibawah ini :

Tabel 2. Notasi pada *Activity diagram*
(Sumber Nugroho, 2008)

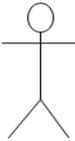
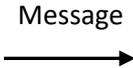
Relasi	Fungsi	Notasi
1	2	3
<i>Initial state</i>	Mempresentasikan dimulainya alur kerja suatu <i>system</i> dalam <i>activity diagram</i> , di notasikan dengan lingkaran <i>solid</i> .	
<i>Final state</i>	Mempresentasikan diakhirinya alur kerja suatu <i>system</i> dalam <i>activity diagram</i> . Dinotasikan dengan lingkaran <i>solid</i> dengan lingkaran diluarnya.	

Relasi	Fungsi	Notasi
1	2	3
<i>Activity</i>	Mempresentasikan perfoma dari beberapa tingkah laku di dalam alur kerja, dinotasikan dengan segi empat.	
<i>Decision Points</i>	Menentukan kapan alur dalam aktivitas menjadi bercabang. Dinotasikan dengan belah ketupat.	
<i>Synchorinization Bars</i>	<i>Fork</i> ; digunakan untuk menunjukkan kegiatan yang dilakukan secara <i>parallel</i> atau untuk menggabungkan dua kegiatan <i>parallel</i> menjadi satu.	
<i>Transition Between Activities</i>	Mengambarkan pertukaran dari <i>control</i> alur antar aktivitas, dinotasikan dengan panah berarah.	

c. *Sequence Diagram*

Sequence Diagram merupakan urutan interaksi antara pengguna dan sistem untuk sebuah *use-case*, juga menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Untuk lebih jelasnya dapat dilihat pada Table 3 dibawah ini :

Tabel 3. Notasi pada *Sequence Diagram*
(Sumber Nugroho, 2008)

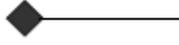
Relasi 1	Fungsi 2	Notasi 3
<i>Object</i>	<i>Object</i> merupakan <i>instance</i> dari sebuah <i>class</i> dan dituliskan tersusun secara <i>horizontal</i> . Digambarkan sebagai sebagai sebuah <i>class</i> (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma.	
<i>Actor</i>	<i>Actor</i> juga dapat berkomunikasi dengan <i>Object</i> , maka <i>Actor</i> juga dapat diurutkan sebagai kolom. Simbol <i>actor</i> sama dengan simbol pada <i>actor use case diagram</i> .	
<i>Lifeline</i>	<i>Lifeline</i> mengindikasikan keberadaan sebuah <i>object</i> dalam basis waktu. <i>Notasi</i> untuk <i>lifeline</i> adalah garis putus-putus <i>vertikal</i> yang ditarik dari sebuah obyek.	
<i>Activation</i>	<i>Activation</i> dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah <i>lifeline</i> . <i>Activation</i> mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.	
<i>Message</i>	<i>Message</i> digunakan dengan anak panah <i>horizontal</i> antara <i>activation</i> . <i>Message</i> mengidentifikasi komunikasi antara <i>object-object</i> .	

d. *Class Diagram*

Class Diagram merupakan struktur *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. *Class* dapat berhubungan dengan yang lain melalui berbagai cara: *associated* (terhubung satu sama lain), *dependent* (satu *class* menggunakan *class* yang lain), *specialized* (satu *class* merupakan spesialisasi dari *class* lainnya), atau *package* (grup bersama sebagai satu unit). Sebuah sistem mempunyai beberapa *class diagram*. Untuk lebih jelasnya dapat dilihat pada Tabel 4 dibawah ini :

Tabel 4. Notasi pada *Class Diagram*
(Sumber Nugroho, 2008)

Relasi	Fungsi	Notasi
1	2	3
<i>Class</i>	<p><i>Class</i> adalah blok-blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagaimana dari <i>class</i>.</p> <p>Bagian tengah mendefinisikan <i>property</i>/atribut <i>class</i>. Bagian akhir mendefinisikan <i>method-method</i> dari sebuah <i>class</i>.</p>	<p>Nama <i>Class</i></p> <p>+atribut</p> <p>+atribut</p> <p>+atribut</p> <p>+<i>method</i></p> <p>+<i>method</i></p>

<i>Association</i>	Sebuah asosiasi merupakan sebuah <i>relationship</i> paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini biasa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum <i>multiplisitas</i> pada sebuah <i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).	
<i>Composition</i>	Jika sebuah <i>class</i> tidak bisa berdiri sendiri, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
<i>Dependency</i>	Kadang kala sebuah <i>class</i> menggunakan <i>class</i> yang lain. Hal ini disebut <i>dependency</i> . Umumnya penggunaan <i>dependency</i> digunakan untuk menunjukkan operasi pada suatu <i>class</i> yang menggunakan <i>class</i> yang lain. Sebuah <i>dependency</i> dilambangkan sebagai sebuah panah bertitik-titik.	
<i>Aggregation</i>	Keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi.	

d. *Entity Relationship Diagram (ERD)*

Menurut Sutanta (2011:91) “*Entity Relationship Diagram (ERD)* merupakan suatu model data yang dikembangkan berdasarkan objek”.

Entity Relationship Diagram (ERD) digunakan untuk menjelaskan hubungan antar data dalam basis data kepada pengguna secara logis. *Entity Relationship Diagram (ERD)* didasarkan pada suatu persepsi bahwa real world terdiri atas obyek-obyek dasar tersebut. Penggunaan *Entity Relationship Diagram (ERD)* relatif mudah dipahami, bahkan oleh para pengguna yang awam. Bagi perancang atau analis sistem, *Entity Relationship Diagram (ERD)* berguna untuk memodelkan sistem yang nantinya, basis data akan di kembangkan.

Model ini juga membantu perancang atau analis sistem pada saat melakukan analis dan perancangan basis data karena model ini dapat menunjukkan macam data yang dibutuhkan dan kerelasian antar data didalamnya.

Menurut Sutanta (2011:91) Komponen *Entity Relationship Diagram* adalah sebagai berikut :

1) Entitas

Entitas merupakan suatu objek yang dapat dibedakan dari lainnya yang dapat diwujudkan dalam basis data. Objek dasar dapat berupa orang, benda, atau hal yang keterangannya

perlu disimpan didalam basis data. Untuk menggambarkan sebuah entitas digunakan aturan sebagai berikut :

- a) Entitas dinyatakan dengan simbol persegi panjang.
- b) Nama entitas dituliskan didalam simbol persegi panjang.
- c) Nama entitas berupa kata benda, tunggal.
- d) Nama entitas sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

2) Atribut

Atribut merupakan keterangan – keterangan yang terkait pada sebuah entitas yang perlu disimpan dalam basis data. Atribut berfungsi sebagai penjelas pada sebuah entitas. Untuk menggambarkan atribut digunakan aturan sebagai berikut :

- a) Atribut digambarkan dengan simbol ellips.
- b) Nama atribut dituliskan didalam simbol ellips.
- c) Nama atribut merupakan kata benda, tunggal.
- d) Nama atribut sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas.

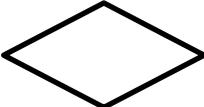
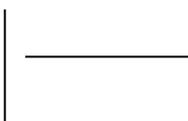
3) Relasi

Relasi merupakan hubungan antara sejumlah entitas yang berasal dari himpunan entitas yang berbeda. Aturan penggambaran relasi adalah sebagai berikut :

- a) Relasi dinyatakan dengan simbol belah ketupat.
- b) Nama relasi dituliskan didalam simbol belah ketupat.
- c) Nama relasi berupa kata kerja aktif.
- d) Nama relasi sedapat mungkin menggunakan nama yang mudah dipahami dan dapat menyatakan maknanya dengan jelas. Simbol

Entity Relationship Diagram bisa dilihat pada Tabel 3.

Tabel 3. Simbol Entity Relationship Diagram

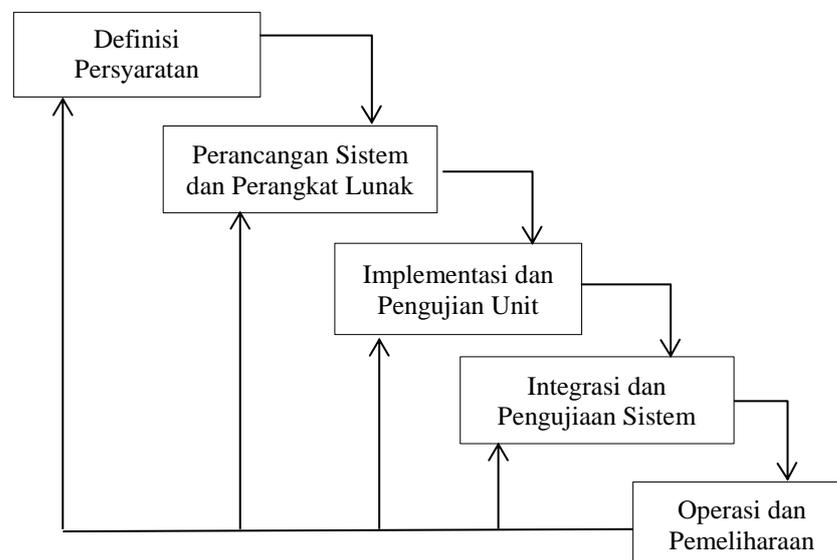
Simbol	Keterangan
	Entitas adalah suatu objek yang dapat diidentifikasi dalam lingkungan pemakai.
	Relasi menunjukkan adanya hubungan di antara sejumlah entitas yang berbeda.
	Atribut adalah karakteristik dari sebuah Entitas.
	Garis Relasi menunjukkan hubungan (keterkaitan antar entitas).

e. Metode *Waterfall*

Menurut Jogiyanto (2010:59) Metodologi Pengembangan Sistem adalah metode – metode, prosedur – prosedur, konsep-konsep pekerjaan, aturan-aturan dan postulat-postulat yang akan digunakan untuk mengembangkan suatu sistem informasi”.

Pengembangan sistem didefinisikan sebagai aktivitas untuk menghasilkan sistem informasi berbasis komputer untuk

menyelesaikan persoalan (problem) organisasi atau memanfaatkan kesempatan (opportunities) yang timbul. Model air terjun (*waterfall*). Biasa juga disebut siklus hidup perangkat lunak. Mengambil kegiatan dasar seperti spesifikasi, pengembangan, validasi, dan evolusi dan merepresentasikannya sebagai fase – fase proses yang berbeda seperti spesifikasi persyaratan, perancangan perangkat lunak, implementasi, pengujian dan seterusnya.



Gambar 3. Siklus Hidup Perangkat Lunak
Sumber : Jogiyanto, 2010

Keterangan Menurut gambar diatas alur dari Model *Waterfall* sebagai berikut :

1) Analisis dan Definisi Persyaratan

Analisis dan definisi persyaratan yang akan diimplementasikan kedalam aplikasi seleksi pemilihan jurusan yang akan penulis kembangkan nantinya adalah :

- a) Aplikasi ini nantinya akan membantu guru dan siswa-siswi dalam pemilihan jurusan yang akan diambil sesuai dengan kemampuan dan bakat.
- b) Aplikasi ini dibuat dengan menggunakan metode pengambilan keputusan yaitu SAW (*Simple Additive Weighting*) dan TOPSIS (*Technique for Order Preference by Similarity to Ideal Solution*).

2) Perancangan Sistem dan Perangkat Lunak

Proses perancangan yang akan menterjemahkan syarat kebutuhan sebuah perangkat lunak yang dapat diperkirakan sebelumnya, *listing* dan perancangan sistem yang dibuat harus sesuai dengan kebutuhan *user* yang berfungsi untuk membantu dalam pembuatan aplikasi pemilihan jurusan. Dan proses ini perfokus pada struktur data, arsitektur perangkat lunak, dan representasi *interface*.

3) Implementasi dan Pengujian Unit

Implementasi dan pengujian unit aplikasi yang dikembangkan oleh penulis dilakukan kepada bagian sekolah, guru kelas, dan siswa-siswi di SMAN-1 Murung Raya.

4) Integrasi dan Pengujian Sistem

Integrasi dan pengujian sistem aplikasi yang dikembangkan oleh penulis menggunakan metode pengujian *Black-Box*.

5) Operasi dan Pemeliharaan

Tahap ini mencakup pada proses pemeliharaan dalam manajemen operasional dalam sistem, sehingga fasilitas yang dimiliki harus dijaga agar digunakan dengan baik sehingga proses operasional dalam seleksi pemilihan jurusan dapat berjalan dengan optimal. Dan juga berguna untuk pengembangan sistem agar mudah untuk dikembangkan.

f. Alat Pengujian

Menurut Pressman (2010:482) tujuan dari pengujian adalah untuk menemukan dan memperbaiki sebanyak mungkin kesalahan dalam program sebelum menyerahkan program kepada *customer*. Salah satu pengujian yang baik adalah pengujian yang memiliki probabilitas tinggi dalam menemukan kesalahan.

1) Pengujian *Black-Box*

Menurut Pressman (2010:495) Pengujian *Black-Box* berfokus pada persyaratan fungsional perangkat lunak yang memungkinkan *engineers* untuk memperoleh set kondisi *input* yang sepenuhnya akan melaksanakan persyaratan fungsional untuk sebuah program. *Black-Box testing* berusaha untuk menemukan kesalahan dalam kategori berikut:

- a) Fungsi yang tidak benar atau fungsi yang hilang
- b) Kesalahan antarmuka
- c) Kesalahan dalam struktur data atau akses *database* eksternal

- d) Kesalahan perilaku (*behavior*) atau kesalahan kinerja
- e) Inisialisasi dan pemutusan kesalahan

Pengujian ini dirancang untuk menjawab beberapa pertanyaan-pertanyaan berikut ini.

- a) Bagaimana validitas fungsional diuji?
- b) Bagaimana perilaku dan kinerja sistem diuji?
- c) Apa kelas *input* akan membuat kasus uji yang baik?
- d) Apakah sistem *sensitive* terhadap nilai input tertentu?
- e) Bagaimana batas-batas kelas data yang terisolasi?
- f) Kecepatan dan volume data seperti apa yang dapat ditolerir sistem?
- g) Efek apakah yang akan menspesifikasikan kombinasi data dalam sistem operasi?

g. *Skala Likert*

Skala Likert digunakan untuk mengukur sikap, pendapat, dan persepsi seorang atau kelompok orang tentang kejadian atau gejala sosial. Dalam penelitian gejala sosial, penelitian telah menetapkan secara spesifik skalanya dan selanjutnya disebut variabel penelitian.

Menurut Guritno, dkk (2011 : 118) *skala likert* disebut juga *summated rating scale*, skala ini sering digunakan karena lebih *fleksibel*, tidak terbatas untuk mengukur sikap saja, tetapi juga persepsi responden terhadap gejala atau fenomena lainnya. Dengan menggunakan *skala likert*, variabel yang akan diukur dijabarkan

menjadi dimensi, lalu dimensi dijabarkan menjadi subvariabel dan subvariabel dijabarkan lagi menjadi indikator yang dapat diukur. Akhirnya indikator-indikator yang terukur dapat menjadi titik tolak untuk membuat item instrumen berupa pernyataan atau pertanyaan yang perlu dijawab responden. Setiap jawaban dihubungkan dengan bentuk pertanyaan atau dukungan sikap yang diungkapkan dengan kata-kata.

menentukan nilai preferensi setiap alternatif, setelah itu dilakukan pengurutan atau perangkingan dari hasil tertinggi ke rendah nilai preferensi yang telah di tentukan.

BAB III

ANALISIS DAN DESAIN SISTEM

A. Tinjauan Umum

1. Sejarah Singkat Sekolah Menengah Atas Negeri 1 Murung Raya

Sekolah Menengah Atas Negeri 1 Murung Raya, Kabupaten Murung Raya didirikan pada tahun 1982. Sebagaimana sejarah berdirinya tercantum dalam surat keputusan dengan nomor : C.172/1982, dan pada tanggal 9 November 1982 ditanda tangani oleh Menteri Pendidikan dan Kebudayaan saat itu Bapak Prof. Dr. Nugroho Notosusanto. Sebelum berpindah lokasi ke kampus Universitas Palangka Raya. Pada mulanya SMA Negeri 1 Murung. Pada tahun pelajaran 1982/1983 dan pada tanggal 20 Mei 1982, Sehingga pada tanggal 28 Oktober ditetapkan sebagai Hari Ulang Tahun SMA Negeri 1 Murung.

B. Analisis

1. Analisis Kelemahan Sistem

Analisa kelemahan sistem yang dilakukan penulis bertujuan untuk membandingkan sistem lama (yang merupakan sistem manual) dengan sistem baru yang dikembangkan penulis dengan memanfaatkan teknologi komputer. Adapun metode yang digunakan penulis pada tahap analisis kelemahan sistem ini adalah metode PIECES. Dimana dengan menggunakan metode ini akan dilakukan analisis secara lebih rinci (*detail*), pada kinerja (*performance*), informasi (*information*), ekonomi (*economy*), kendali (*control*), analisis efisiensi (*effeciency*),

dan pelayanan (*service*) dari sistem seperti pada Tabel 4 Penjelasan Analisis Metode PIECES sebagai berikut.

Tabel 4. Analisis Kelemahan Sistem

No	Analisis	Sistem Lama	Sistem baru yang di usulkan
1	Kinerja (<i>performance</i>)	Penilaian terhadap pemilihan jurusan masih dilakukan secara manual, yaitu wali kelas melakukan dari ranking dan membagi kuisioner minat siswa.	Pengguna cukup menginput nilai raport siswa ke sistem, sehingga mengetahui dimana minat siswa tersebut.
2	Informasi (<i>Information</i>)	Pencarian data siswa yang telah mendaftar pemilihan jurusan dilakukan dengan mencari satu persatu lembaran yang ada, hal tersebut akan membutuhkan waktu yang lama.	Pengguna hanya perlu menjalankan aplikasi sistem pendukung keputusan pemilihan jurusan dan setelah itu pengguna tinggal memasukan nama siswa yang akan di cari.
3	ekonomi (<i>economy</i>)	Untuk mencari data pemilih jurusan pengguna harus melihat kembali satu persatu data yang diinginkan pada tumpukan kertas yang begitu banyak, sehingga data yang tersimpan tidak terlihat rapi serta sering terjadinya kesalahan atau kesamaan data hal ini yang menyebabkan kerugian terutama pengguna harus membuat kembali data	Pengguna cukup menjalan kan aplikasi sistem pendukung keputusan pemilihan jurusan untuk mengedit data yang salah serta data yang tersimpan pun akan terlihat rapi dan biaya yang dikeluarkan tidak terlalu banyak.

		yang rusak atau hilang.	
4	Kendali (<i>control</i>)	Pengguna tentu membutuhkan waktu yang cukup lama untuk melakukan proses perhitungan atau penilaian masih dilakukan secara manual dan tanpa kriteria yang jelas.	Pengguna hanya membutuhkan waktu beberapa menit melakukan penilaian untuk pemilihan jurusan, dengan cara memasukan nilai pada kriteria yang telah di tentukan.
5	Efisiensi (<i>efficiency</i>)	Dari segi efisiensi pada sistem lama pengguna masih manual dengan melihat data-data untuk pemilihan jurusan. Hal ini dinilai kurang efisien, karena akan memakan waktu yang lama untuk menentukan siswa yang akan memilih jurusan.	Pengguna cukup memilih data pemilihan jurusan dan memasukan kriteria yang telah di tentukan, maka dengan sendirinya (otomatis) akan muncul siswa yang sudah memilih jurusan.
6	Pelayanan (<i>service</i>)	Segi pelayanan terhadap pemilihan jurusan yang mendaftar masih menggunakan sistem yang lama maka akan memakan waktu yang lama dan tenaga yang lebih.	Pengguna hanya perlu menjalankan aplikasi sistem pendukung keputusan pemilihan jurusan untuk mempercepat seleksi pemilihan jurusan yang layak, sehingga pelayanan terhadap siswa yang mendaftar akan lebih cepat, serta waktu dan tenaga yang dibutuhkan tidak terlalu banyak.

2. Analisis Kebutuhan Sistem

a. Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yang diperlukan dalam menjalankan Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada Sman-1 Murung Menggunakan Metode *Simple Additive Weighting (Saw)* Dan *Technique For Order Preference By Similarity (Topsis)* Berbasis *Ci* adalah sebagai berikut :

- 1) *Processor minimal Dual Core*
- 2) *Memory 2GB*
- 3) *Monitor 14 inch*
- 4) *Keyboard dan Mouse*
- 5) *Printer*

b. Kebutuhan Perangkat Lunak

Kebutuhan perangkat lunak (*software*) yang di perlukan dalam menjalankan Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada SMAN-1 Murung Menggunakan Metode *Simple Additive Weighting (Saw)* Dan *Technique For Order Preference By Similarity (Topsis)* Berbasis *Ci* adalah sebagai berikut :

- 1) *Windows 7 Ultimate 32-bit*
- 2) *Framework Codeigniter*
- 3) *Notepad ++.*

c. Kebutuhan Informasi

Kebutuhan informasi dalam pembuatan Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada Sman-1 Murung Raya Menggunakan Metode *Simple Additive Weighting (Saw)* Dan *Technique For Order Preference By Similarity (Topsis)*, adalah studi pustaka dan dokumentasi, yaitu dengan membaca buku-buku teori dari para ahli serta dari *internet* yang berhubungan langsung dengan pembuatan perangkat lunak berbasis *web*.

d. Kebutuhan Pengguna (*User*)

Aplikasi Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada Sman-1 Murung Menggunakan Metode *Simple Additive Weighting (Saw)* Dan *Technique For Order Preference By Similarity (Topsis)* Berbasis *Ci* (Studi Kasus di Kabupaten Murung Raya) dibuat hanya digunakan oleh siswa dan guru-guru di SMAN-1 Murung Raya.

3. Analisis Kelayakan Sistem

Analisis kelayakan sistem baru jika diimplementasikan pada obyek penelitian. Analisis ini dilakukan untuk menentukan apakah proyek pengembangan sistem layak dipakai atau tidak.

a. Kelayakan Teknologi

Kelayakan teknologi berkaitan dengan kelayakan teknis, dalam menentukan teknis pada pembuatan sistem pendukung keputusan pemilihan jurusan, dapat dilihat dari ketersediaan

teknologi seperti komputer dan printer yang ada di sekolah SMAN-1 Murung Raya.

b. Kelayakan Hukum

Kelayakan hukum dalam pembuatan sistem pendukung keputusan pemilihan jurusan, dibuat sesuai dengan ijin penelitian yang diajukan penulis kepada sekolah SMAN 1 Murung Raya, sehingga sistem ini tidak melanggar ketentuan yang sudah disetujui dan juga diharapkan agar sistem dapat menjadi alat bantu untuk menyeleksi pemilihan jurusan di sekolah SMAN-1 Murung Raya.

c. Kelayakan Ekonomi

Secara ekonomi sistem ini akan menghemat pengeluaran dalam pengelolaan data siswa yang sangat banyak. Karena pengelolaan data akan dilakukan pasca sistem yang terkomputerisasi.

C. Desain Sistem

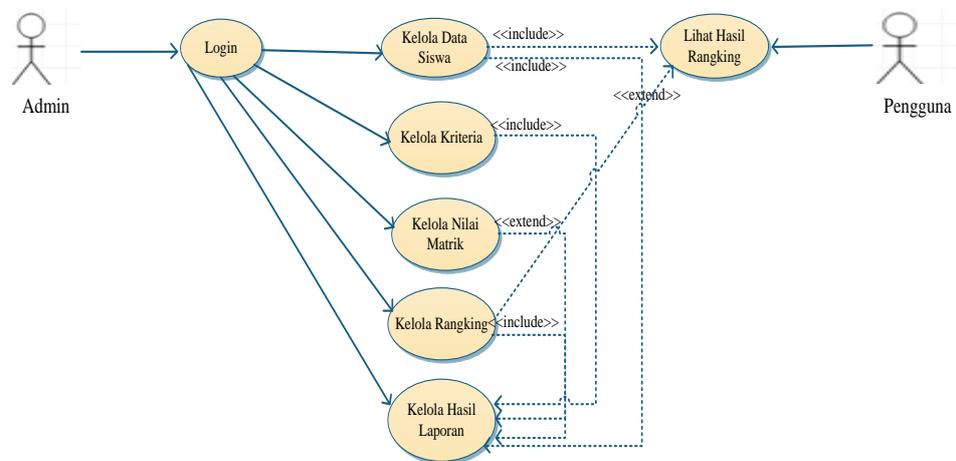
Berdasarkan tahapan pada pemodelan sistem yang penulis gunakan, maka penulis akan menjelaskan tentang *Unified Modelling Language (UML)*.

1. Desain Proses

Pada tahapan ini akan menguraikan dalam beberapa diagram yaitu *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

a. *Use Case Diagram*

Use case diagram yang diperuntukan untuk menggambarkan fungsionalitas sebuah sistem. Sebuah diagram *usecase* mengidentifikasi aktor yang terlibat dalam kegiatan yang dapat dilakukan terhadap aplikasi. Dalam hal ini meliputi pengguna dan *administrator* yang dapat dilihat dengan Gambar 7.



Gambar 7. *Use case Diagram Interaksi Sistem*

Gambar 7 menunjukkan *use case diagram* keseluruhan yang merupakan skenario yang berhubungan satu sama lain dengan satu tujuan yang sama dari pengujung/responden, dan *admin*. *Use case diagram* berisi apa yang dilakukan oleh sistem. Sebuah *Use case diagram* mempresentasikan sebuah interaksi antara aktor dengan sistem. Untuk lebih jelasnya dapat dilihat pada Gambar 7.

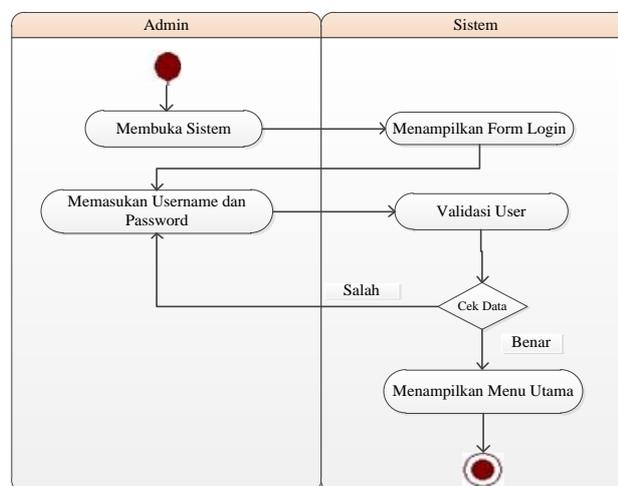
b. *Activity Diagram*

Activity diagram menggambarkan sebagai alur aktivitas dalam sistem yang sedang di rancang, bagaimana masing-masing alur berawal yang memungkinkan terjadi, dan bagaimana mereka berakhir.

1) Activity Diagram Admin

a) Activity Diagram Login

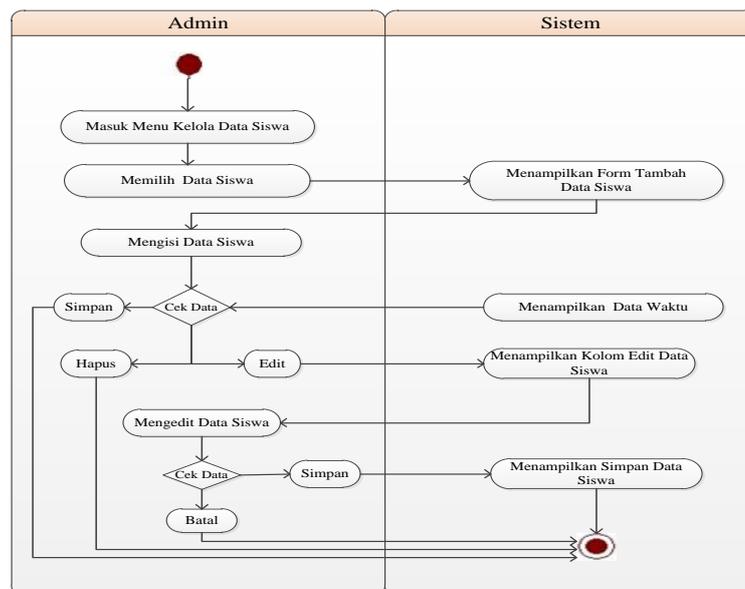
Gambar 8 menunjukkan activity diagram untuk proses *login*, dimulai dari *form login* tampil ketika sistem dibuka, kemudian *user* memasukkan *username* dan *password*, dan masuk dalam proses validasi apa *username* dan *password* sudah *benar*, apabila benar maka sistem akan menampilkan menu utama dan apabila salah maka sistem akan meminta user untuk memasukkan *username* dan *password* kembali. Untuk lebih jelasnya dapat dilihat pada Gambar 8.



Gambar 8 Activity Diagram Login Admin

b) *Activity Diagram* Kelola Data Siswa

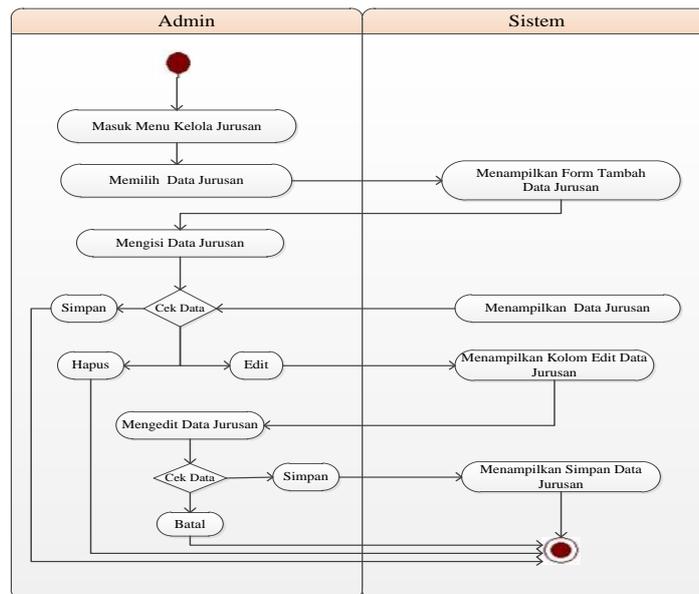
Gambar 9 menunjukkan activity diagram kelola data siswa, dimulai dari *admin* masuk menu data siswa kemudian memilih menu *form* data siswa, maka sistem menampilkan *form* yang berisikan data siswa yang sudah ditambah dan tampil pilihan simpan. Kemudian data siswa akan tampil disistem dan ada pilihan kembali yaitu edit dan hapus. Untuk lebih jelasnya dapat dilihat pada Gambar 9.



Gambar 9 *Activity Diagram* Kelola Data Siswa

c) *Activity Diagram* Kelola Data Jurusan

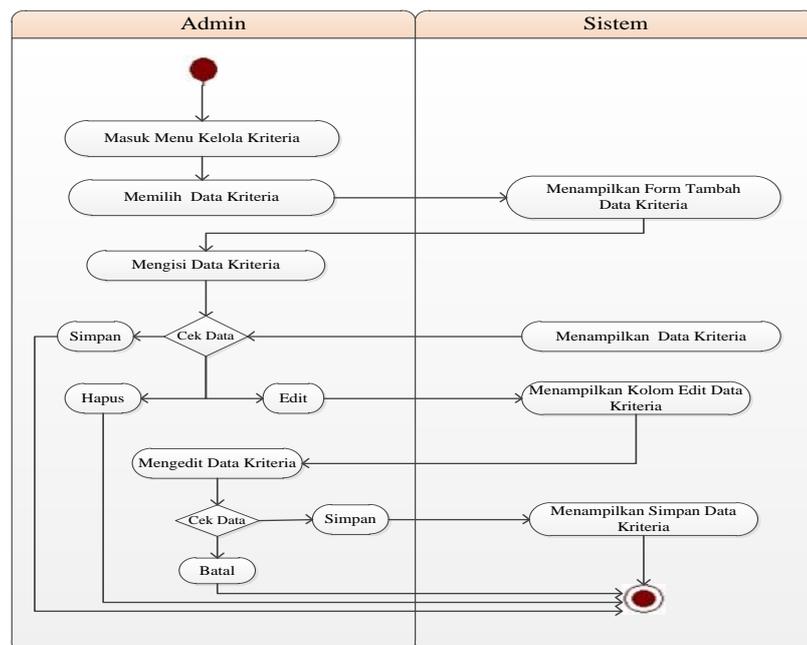
Gambar 10 menunjukkan activity diagram kelola data jurusan, dimulai dari *admin* masuk menu pusat informasi kemudian memilih menu *from* data jurusan, maka sistem menampilkan *form* yang berisikan data jurusan yang sudah ditambah dan tampil pilihan simpan. Kemudian data jurusan akan tampil disistem dan ada pilihan kembali yaitu edit dan hapus. Untuk lebih jelasnya dapat dilihat pada Gambar 10.



Gambar 10 *Activity Diagram* Kelola Data Jurusan

d) *Activity Diagram* Kelola Data Kriteria

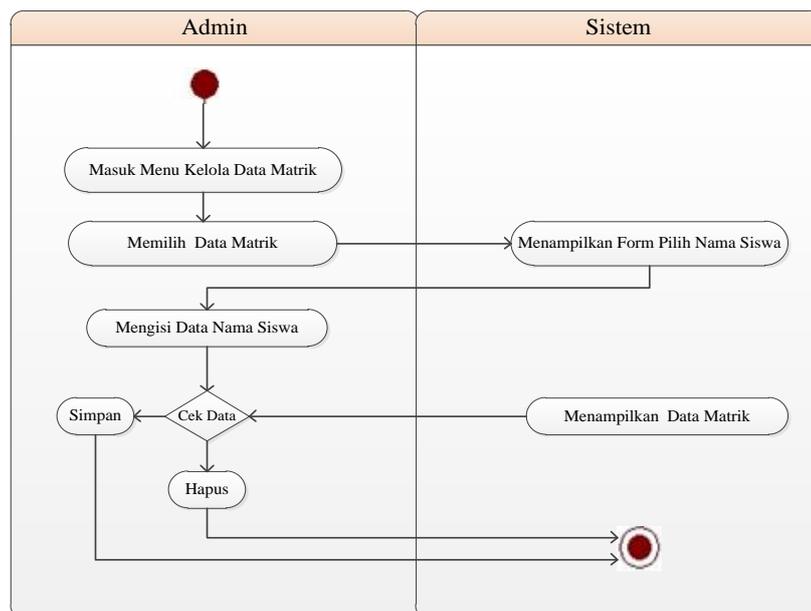
Gambar 11 menunjukkan *activity* diagram kelola data kriteria, dimulai dari *admin* masuk menu data kriteria kemudian memilih menu *from* kriteria, maka sistem menampilkan *form* yang berisikan data kriteria yang sudah ditambah dan tampil pilihan simpan. Kemudian kriteria akan tampil disistem dan ada pilihan kembali yaitu edit dan hapus. Untuk lebih jelasnya dapat dilihat pada Gambar 11.



Gambar 11. *Activity Diagram* Kelola Data Kriteria

e) *Activity Diagram* Kelola Data Matrik

Gambar 12 menunjukkan activity diagram kelola data matrik, dimulai dari *admin* masuk menu data matrik kemudian memilih menu *from* kelola data matrik, maka sistem menampilkan *form* yang berisikan data matrik yang sudah ditambah dan tampil pilihan simpan. Kemudian data matrik akan tampil disistem dan ada pilihan kembali yaitu edit dan hapus. Untuk lebih jelasnya dapat dilihat pada Gambar 12.



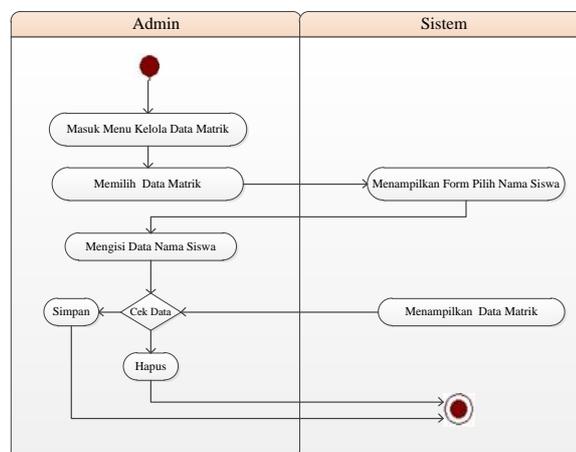
Gambar 12. *Activity Diagram* Kelola Data Matrik

f) *Activity Diagram* Kelola Data Rangking

Gambar 12 menunjukkan activity diagram kelola data rangking, dimulai dari *admin* masuk menu data rangking, kemudian memilih menu *from* kelola data rangking maka sistem menampilkan *form* yang berisikan data layanan yang sudah ditambah dan tampil pilihan simpan. Kemudian data rangking akan tampil disistem dan ada pilihan kembali yaitu edit dan hapus. Untuk lebih jelasnya dapat dilihat pada Gambar 12.

g) *Activity Diagram* Kelola Hasil Laporan

Gambar 12 menunjukkan *activity* diagram kelola hasil laporan, dimulai dari *admin* masuk menu laporan kemudian memilih menu *from* hasil laporan, maka sistem menampilkan *form* yang berisikan data hasil laporan yang sudah ditambah dan tampil pilihan simpan. Kemudian data hasil laporan akan tampil disistem dan ada pilihan kembali yaitu edit dan hapus. Untuk lebih jelasnya dapat dilihat pada Gambar 12.



Gambar 12 *Activity Diagram* Kelola Data Hasil Laporan

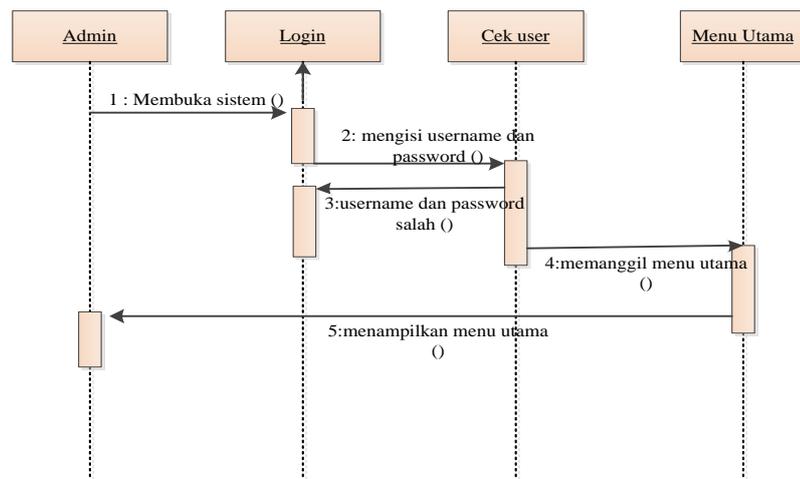
c. *Sequence Diagram*

Sequence diagram merupakan gambaran secara grafis sebuah skenario yang menunjukkan interaksi antar objek dalam sebuah urutan waktu. Untuk gambar *Sequence diagram* sistem yaitu, sebagai berikut :

1) *Sequence Diagram Admin*

a) *Login Admin*

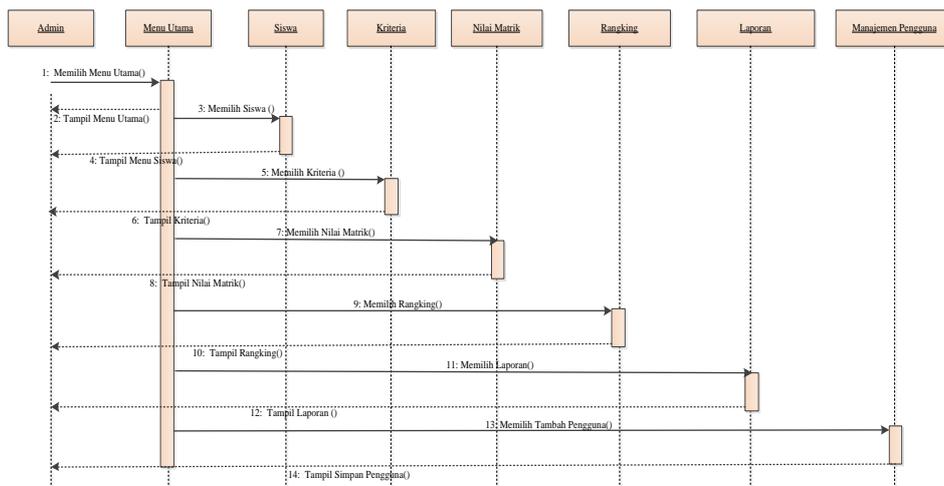
Gambar 13 menunjukkan *Sequence diagram* dimulai dari *admin* membuka sistem, maka sistem akan masuk ke *login*, kemudian *dmin* harus mengisi *username* dan *password*, apabila *username* dan *password* salah maka sistem akan kembali ke proses *login*, dan apabila benar maka sistem akan memanggil menu utama dan menampilkan menu utama . Untuk lebih jelasnya dapat dilihat pada Gambar 13



Gambar 13 *Sequence Diagram Login Admin*

b) Menu Utama *Admin*

Gambar 14 menunjukkan *Sequence* diagram menu utama *admin* dimulai dari mengakses menu dan memilih sub menu apa yang ingin diakses, terdapat tujuh sub menu dalam sistem. Kemudian *admin* memanggil sub yaitu, menu utama, menu data siswa, menu data kriteria, menu data nilai matrik, menu data rangking, menu hasil laporan, dan manajemen pengguna, kemudian sistem akan menampilkan satu persatu *formnya* sesuai dengan inputan *admin*. Untuk lebih jelasnya dapat dilihat pada gambar 14.

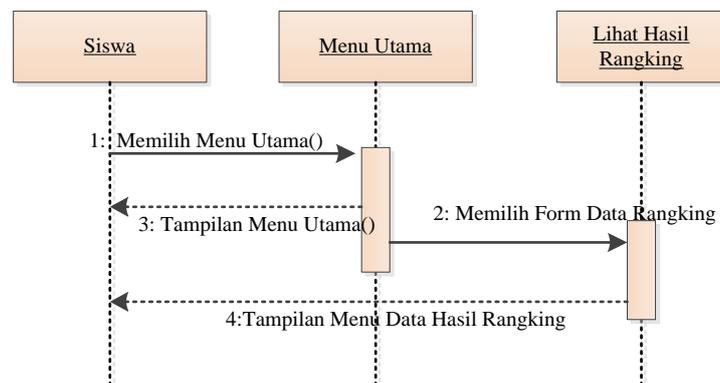


Gambar 14 *Sequence* Diagram Menu Utama *Admin*

2) *Sequence* Diagram Untuk Pengguna/Siswa

Gambar 15 menunjukkan *Sequence* diagram pengguna/siswa dimulai dari pengguna/siswa mengakses menu dan memilih sub menu apa yang ingin diakses, terdapat dua sub menu dalam sistem. Kemudian pengguna/siswa memanggil sub menu utama, dan lihat hasil rangking,

kemudian sistem akan menampilkan satu persatu *formnya* sesuai dengan inputan pengguna/siswa. Untuk lebih jelasnya dapat dilihat pada Gambar 15



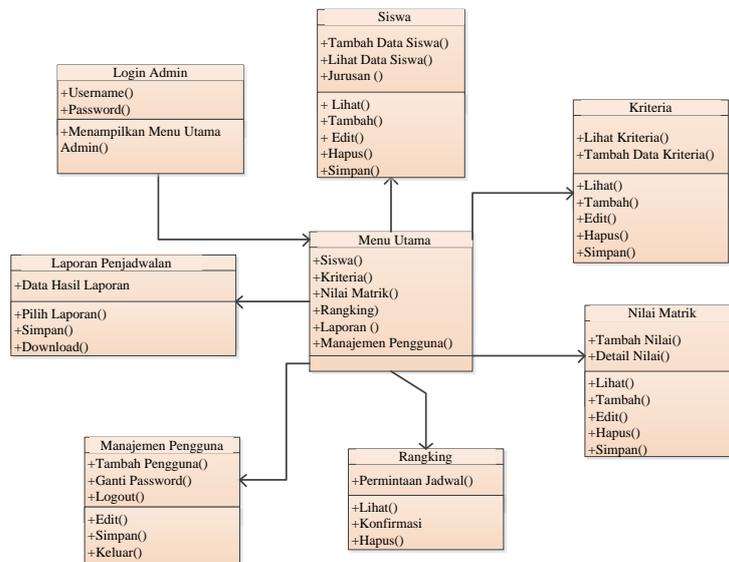
Gambar 15 *Sequence Diagram* Menu Pengguna/Siswa

d. *Class Diagram*

Class diagram menggambarkan struktur dan deskripsi yang menunjukkan *class-class* yang ada dari sebuah sistem dan hubungannya secara logika. *Class diagram* menggambarkan struktur statis dari sebuah sistem. Karena itu, *class diagram* merupakan tulang punggung atau kekuatan dasar dari hampir setiap metode berorientasi objek. Desain *class diagram*, yaitu sebagai berikut :

1) *Class Diagram Admin*

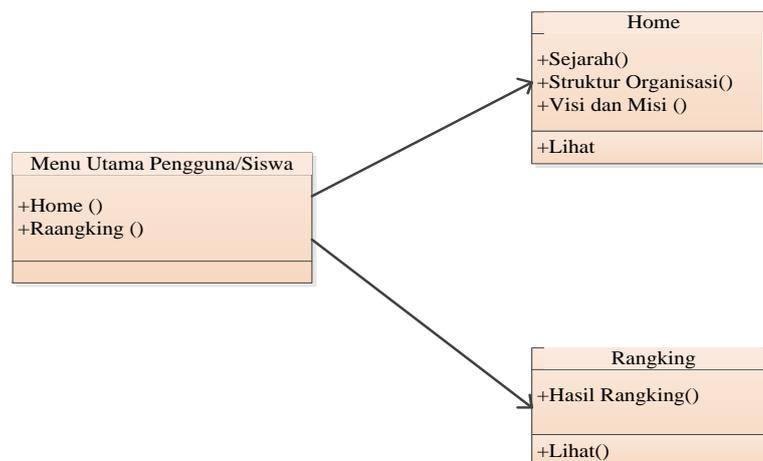
Gambar 16 *class diagram admin* dapat dilihat pada beberapa *class* yang memiliki hubungan yang mana terdapat fasilitas-fasilitas yang disediakan sistem dan pengunjung dapat memanggil menu sesuai dengan yang dibutuhkan. Untuk lebih jelasnya dapat dilihat pada Gambar 16.



Gambar.16 Class Diagram Admin

2) Class Diagram Pengguna/Siswa

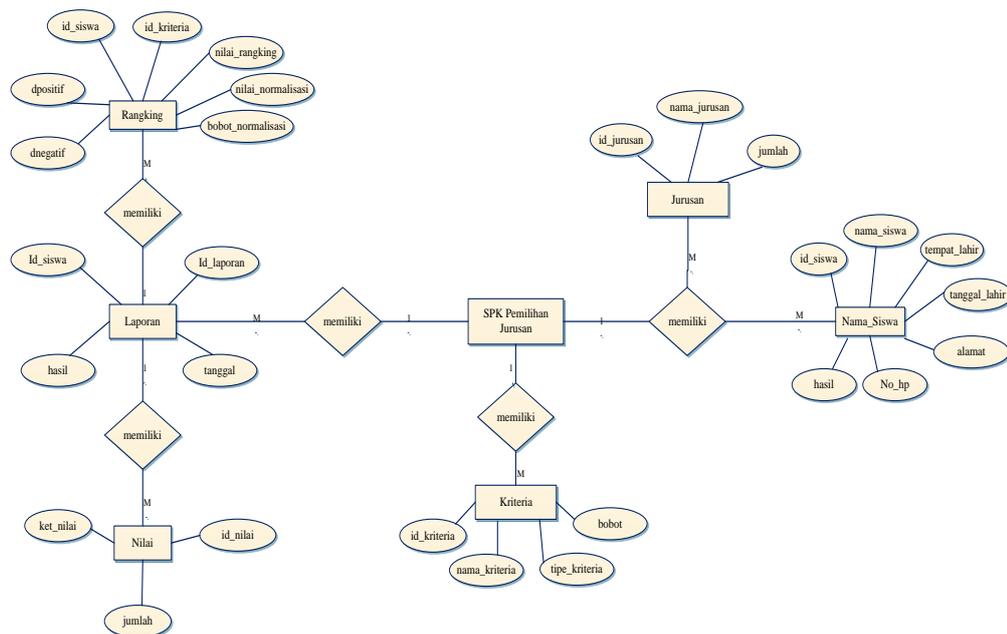
Gambar 17 class diagram pengguna/siswa dapat dilihat pada beberapa class yang memiliki hubungan yang mana terdapat fasilitas-fasilitas yang disediakan sistem untuk pengguna/siswa dapat memanggil menu sesuai dengan yang dibutuhkan. Untuk lebih jelasnya dapat dilihat pada Gambar 17



Gambar 17 Class Diagram Pengguna/Siswa

3. Desain Basis Data

Pada tahapan desain basis data ini dibuat relasi antar entitas dan perancangan tabel sebagai penunjang pembuatan Aplikasi Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada Sman-1 Murung Raya Menggunakan Metode *Simple Additive Weighting (SAW)* Dan *Technique For Order Preference By Similarity (Topsis)* Berbasis *Ci*. Sistem informasi penjualan secara online menggunakan alur diagram yang sering disebut dengan istilah *ERD*. Diagram *ERD* berguna untuk memastikan alur proses yang terjadi pada sebuah sistem. Untuk lebih jelasnya diagram *ERD* dapat dilihat pada Gambar..



Gambar 24.ERD (Entity Relationship Diagram)

a. Desain Database

Perancangan Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada Sman-1 Murung Menggunakan Metode *Simple Additive Weighting (Saw)* Dan *Technique For Order*

Preference By Similarity (Topsis) Berbasis *Ci*, ini terdapat beberapa struktur tabel basis data yang akan digunakan sebagai tempat penyimpanan data dan informasi. Berikut tabel yang digunakan :

1) Tabel Admin

Tabel 12 memuat data – data pengguna yang akan dijalankan oleh sistem.

Tabel 12. Tabel Admin

Field	Type	Size	Keterangan
Id	Varchar	3	Primary key
username	Varchar	50	Username sebagai login
Password	Varchar	50	Kata sandi sebagai login

2) Tabel Siswa

Tabel 13 memuat data – data peserta yaitu siswa yang mengikuti seleksi.

Tabel 13. Tabel Siswa

Field	Type	Size	Keterangan
Nis	Varchar	11	Primary key
nama_s	Varchar	30	Nama siswa
Tempat	Varchar	30	Tempat Lahir
tgl_lahir	Date	-	Tanggal Lahir siswa
j_kel	enum('L','P')	-	
No_hp	Double	20	Nomer hp
Hasil	Double	100	Hasil

3) Tabel Kriteria

Tabel 14 memuat data-data kriteria dan bobot untuk masing- masing kriteria.

Tabel 14. Tabel Kriteria

Field	Type	Size	Keterangan
id_k	Varchar	3	Primary key
nama_kriteria	Varchar	30	Nama Kriteria
bobot	Int	11	Bobot Nilai Kriteria

4) Tabel Keputusan

Tabel 15 memuat data – data hasil keputusan yaitu siswa yang terpilih mengikuti LKS – SMK.

Tabel 15. Keputusan

Field	Type	Size	Keterangan
Nis	Varchar	20	Primary key
nilai	Double	-	Hasil Perhitungan
hasil	Varchar	100	Keterangan dari hasil perhitungan

5) Tabel Jurusan

Tabel 16 memuat data – data hasil jurusan

Field	Type	Size	Keterangan
Id_jurusan	Int	11	Primary key
Nm_jurusan	Varchar	50	Nama Jurusan
jumlah	Int	4	Jumlah

6) Tabel Nilai

Tabel 17 memuat data – data hasil nilai siswa yang mengikuti seleksi, Tipe data double adalah tipe data yang bertujuan untuk menyimpan nilai dengan angka desimal. Bilangan double digunakan untuk angka desimal terbesar. Tipe data double **dapat menyimpan data** pembagian yang nilai dibelakang koma tidak habis (tak berhingga) seperti $22/7$, $1/3$, $20/6$ dan lainnya

Field	Type	Size	Keterangan
Id_nilai	Int	11	Primary key
Nilai	Varchar	30	Hasil Perhitungan
Hasil	double	-	Keterangan dari hasil perhitungan

7) Tabel ranking

Tabel 18 memuat data – data hasil ranking siswa yang mengikuti seleksi

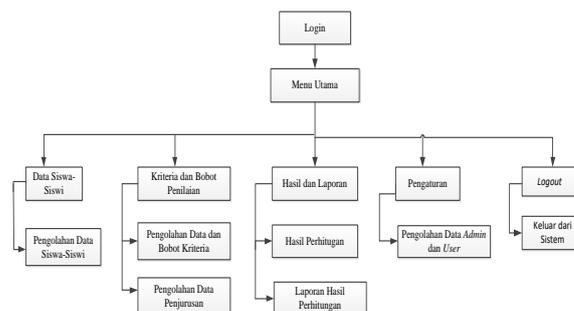
Field	Type	Size	Keterangan
Id_siswa	Int	11	Primary key
Id_kriteria	Varchar	30	Id_kriteria
Nilai_ranking	double	-	Keterangan dari hasil perhitungan
Nilai_Normalisasi	double	-	Nilai normalisasi
Bobot_normalisasi	double	-	Bobot bormalisasi
Dpositif	double	-	Positif
Dnegatif	double	-	Negatif

5. Desain *Interface* / Antarmuka

Desain *interface* merupakan perancangan *form-form* yang ada didalam sistem. Desain yang akan dibangun pada aplikasi ini adalah sebagai berikut :

a. Perencanaan Struktur Menu

Perencanaan struktur menu merupakan rancangan menu yang digunakan untuk memudahkan pemakaian dalam menjalankan program sehingga pada saat menjalankan program, *user* tidak mengalami kesulitan dalam memilih menu – menu yang diinginkan. Struktur menu pada sistem ini dapat dilihat pada gambar



Gambar 6. Struktur Menu

b. Desain *Interface*

1) Desain *Form Login*

Proses login admin dengan memasukkan *username* dan *password* sistem akan mengecek kesesuaian *username* dan *password* yang ada di dalam *database*, *admin* juga dapat mengubah data.

a) *Login Admin*

Proses *login admin* dengan memasukkan *username* dan *password* sistem akan mengecek kesesuaian *username* dan *password* yang ada di dalam *database*, *admin* juga dapat mengubah data.

	
	Username
	Password
<input type="button" value="Login"/>	
<input type="button" value="Reset"/>	

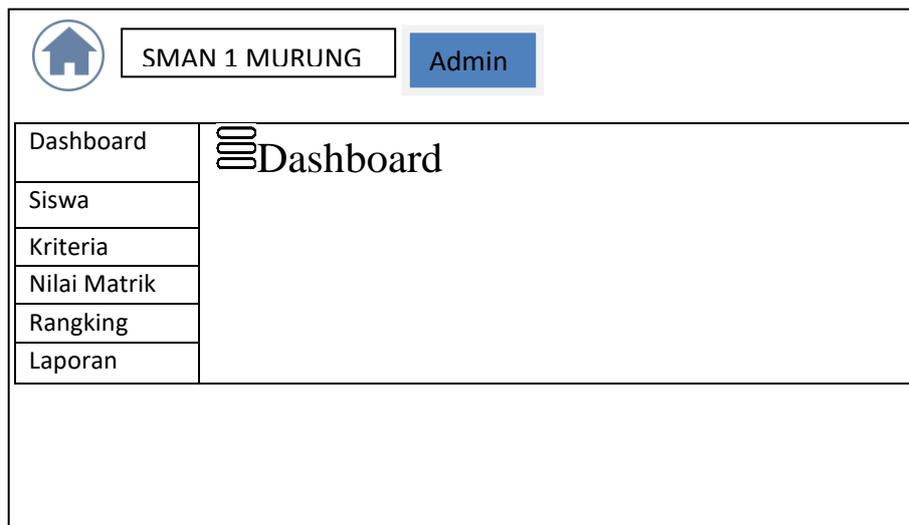
b) Menu Utama

Pengertian *homepage* adalah halaman muka dari suatu situs web (web site), atau lokasi dari situs web di mana dokumen web dapat diakses. Setelah melakukan proses login dengan memasukkan *username* dan *password* akan muncul tampilan home seperti gambar 16

	SMAN 1 MURUNG
Home	
Rangking	
	<input type="button" value="Selamat Datang"/>

c) *Dashboard*

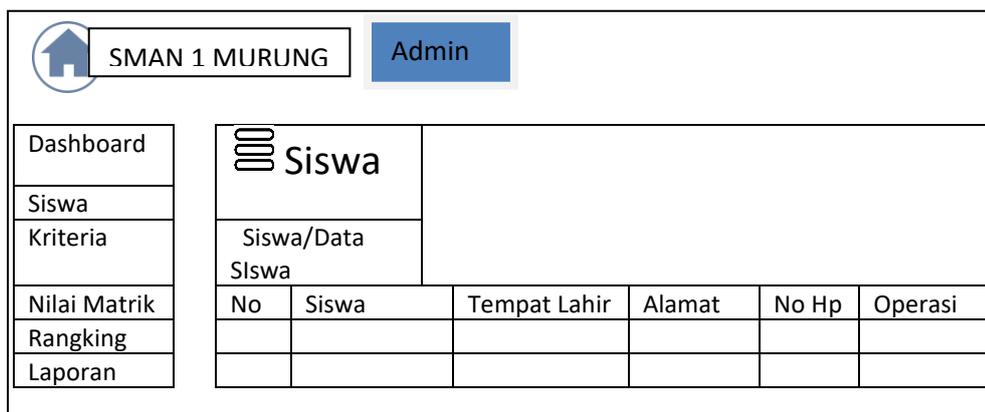
Dashboard adalah pusat control panel *berplatform* yang berfungsi untuk mengatur semua kegiatan di sebuah situs atau website.



Gambar 17. Tampilan Dashboard

d) *Siswa*

Dalam *homepage* ini, terdapat beberapa data yang perlu di isi oleh calon pemilih jurusan. Seperti, nama siswa, tanggal lahir, alamat dan nomer hp.



e) Data Siswa

Pada tampilan *homepage* ini siswa calon pemilih jurusan diharapkan mengisi data siswa dengan sebenarnya. Data yang diperlukan seperti ; nama siswa, tanggal lahir, alamat dan nomer hp.

The screenshot shows the 'Tambah Data Siswa' form. On the left is a navigation menu with options: Dashboard, Siswa, Kriteria, Nilai Matrik, Rangking, and Laporan. The main content area has a header with 'SMAN 1 MURUNG' and 'Admin'. The form fields are: Nama Siswa* (text input), Tempat Lahir* (text input), Tanggal Lahir* (text input), Alamat* (text input), and No Hp* (text input). Below the form are 'Save' and 'Cancel' buttons.

Gambar 19. Tampilan Data Siswa

f) Kriteria

Homepage kriteria ini berisikan nama kriteria, jenis, bobot dan operasi. Dalam *homepage* ini juga bisa melakukan reset kriteria.

The screenshot shows the 'Kriteria' page. On the left is a navigation menu with options: Dashboard, Siswa, Kriteria, Nilai Matrik, Rangking, and Laporan. The main content area has a header with 'SMAN 1 MURUNG' and 'Admin'. Below the header is a 'Kriteria' title and a 'Reset Kriteria' button. At the bottom is a table with the following structure:

No	Nama Kriteria	Jenis	Bobot	Operasi

Gambar 20. Tampilan Kriteria

g) Nilai Matrik

Pada tampilan nilai matrik, admin memilih siswa dan melakukan input nilai kriteria, yaitu harga, kualitas, populer, purna jual dan keawetan. Untuk menyimpan data

Gambar 21 Tampilan Halaman Nilai Matrik

h) Ranking

Pada halaman perankingan akan muncul nama siswa yang tergolong kepada jurusan IPA atau IPS.

Gambar 22. Tampilan halaman perankingan

i) Laporan

Pada halaman laporan berisikan laporan yang sudah ada di database. Untuk melihat laporan, pilih bulan dan tahun laporan lalu laporan dapat dicetak.

	SMAN 1 MURUNG	Admin	
Dashboard			
Siswa			
Kriteria			
Nilai Matrik			
Rangking			
Laporan	Januari	2018	Cetak Laporan

Gambar 23. Tampilan Halaman Laporan

BAB IV

IMPLENTASI DAN PEMBAHASAN

A. Implementasi

Pengimplementasian sistem yang telah dirancang dan dibangun ditunjukkan untuk melihat dan menganalisa mengenai kesesuaian sistem terhadap rancangan yang dibuat, dimana sistem tersebut akan melalui proses uji coba untuk melihat apakah sistem dapat bekerja sesuai yang diharapkan atau tidak.

1. Uji Coba Sistem dan Program

Uji coba aplikasi dan *source code* perangkat lunak dari Aplikasi Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada Sman-1 Murung Raya Menggunakan Metode *Simple Additive Weighting (SAW)* Dan *Technique For Order Preference By Similarity (Topsis)* Berbasis *Ci*.

a) Skenario Pengujian *Black Box Testing*

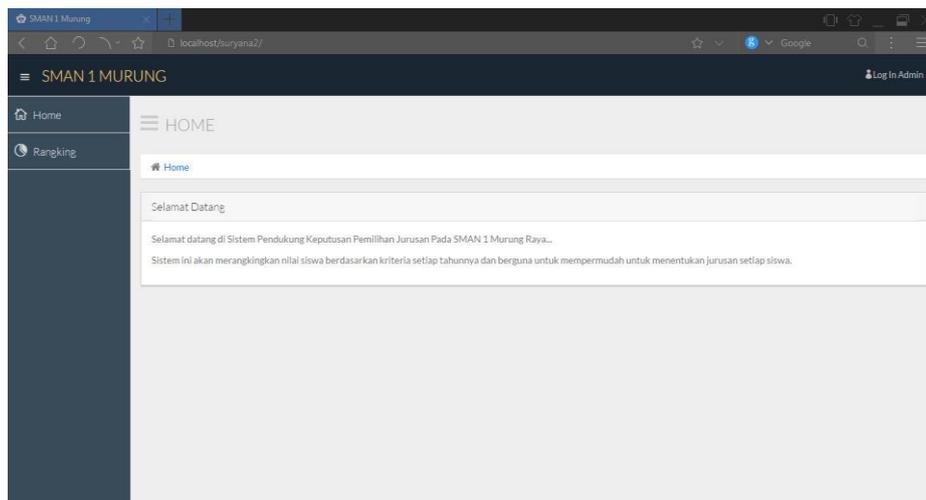
Pengujian aplikasi yang penulis gunakan yaitu menggunakan *black box Testing* dengan menjabarkan pengujian. Dengan pengujian ini maka dapat diketahui kesalahan pada aplikasi. *Black box testing* merupakan pengujian yang dilakukan untuk mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari aplikasi perangkat lunak.

1) Pengujian Halaman Pengunjung

Halaman beranda merupakan halaman yang pertama kali akan muncul saat pengunjung membuka sistem pendukung keputusan seleksi pemilihan jurusan pada SMAN 1 Murung Kab.Murung Raya

Tabel 6. Pengujian Halaman Pengunjung

No	Pengujian	Data Masukan	Hasil Yang diharapkan	Hasil	Kesimpulan
1.	Menu Halaman Pengunjung	Klik menu beranda	Tampil halaman beranda	Halaman beranda tampil	Sesuai
		Klik menu pemilihan jurusan	Tampil halaman pemilihan jurusan	Halaman pemilihan jurusan tampil	Sesuai



Gambar 24. Beranda Pengunjung

Siswa	Kriteria			Hasil
	nilai rapor (Benefit)	nilai pelajaran (Benefit)	perilaku (Benefit)	
Rizal	75	80	75	1
Riky	75	80	75	1

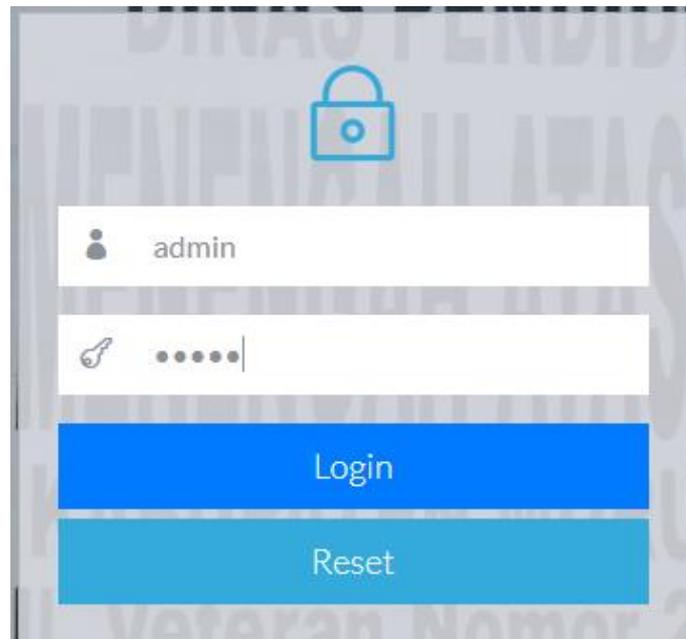
Gambar. 25 Data Hasil Rangking

2) Penguji Masuk Admin

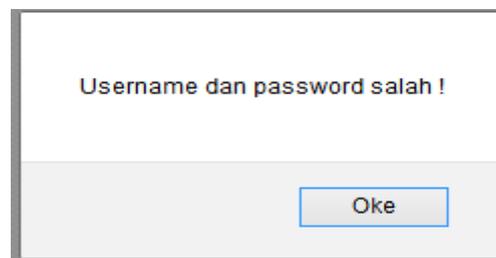
Halaman *Login admin* digunakan untuk *login* ke halaman admin, dengan mengisi *username* dan *password* yang sudah terdaftar di database lalu klik tombol *login*.

Tabel 7. Pengujian Masuk Admin

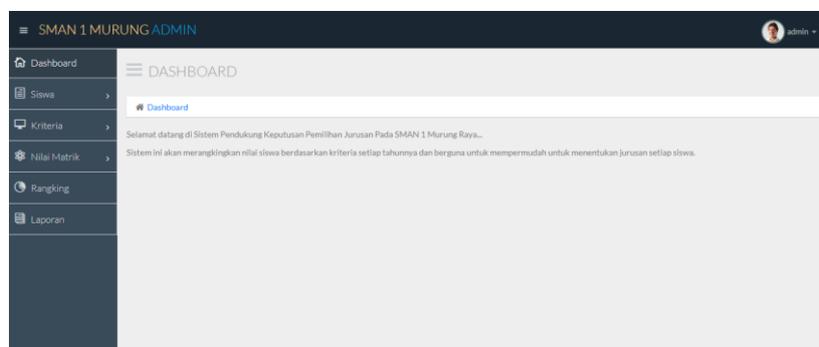
No	Pengujian	Data Masukan	Hasil yang diharapkan	Hasil	Kesimpulan
1	Menu Masuk Admin	<i>Username</i> dan <i>password</i> tidak diisi	Gagal masuk dan kembali ke halaman admin	Admin tidak dapat masuk dan kembali ke halaman masuk admin	Sesuai
		<i>Username</i> dan <i>password</i> tidak benar	Gagal login dan kembali ke halaman masuk admin	Admin tidak dapat masuk dan kembali ke halaman masuk	Sesuai
		<i>Username</i> dan <i>password</i> benar	Berhasil login, muncul halaman beranda admin	Admin berhasil masuk beranda admin muncul	Sesuai



Gambar 25. Pengujian *Login Admin*



Gambar 26. *Username dan Pasword* salah



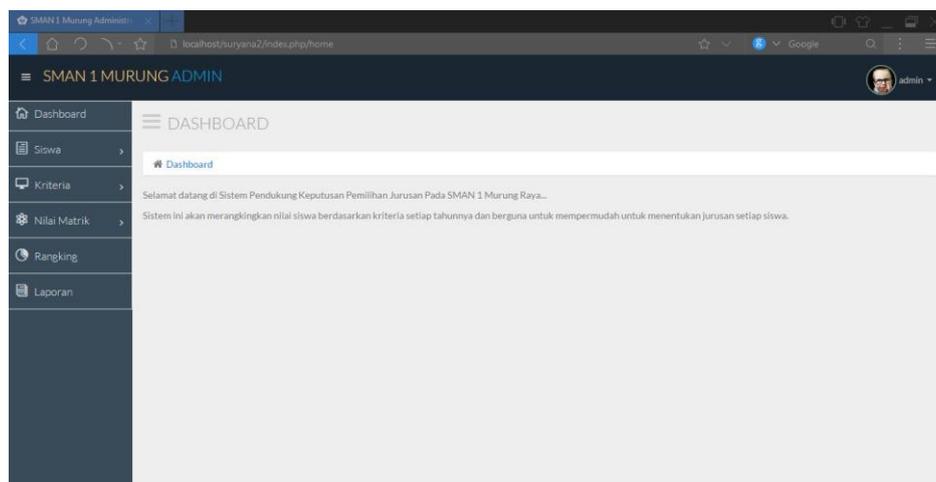
Gambar 27. Halaman Login Berhasil Akan Masuk Kehalaman Beranda

3) Pengujian Halaman Pemilihan Jurusan

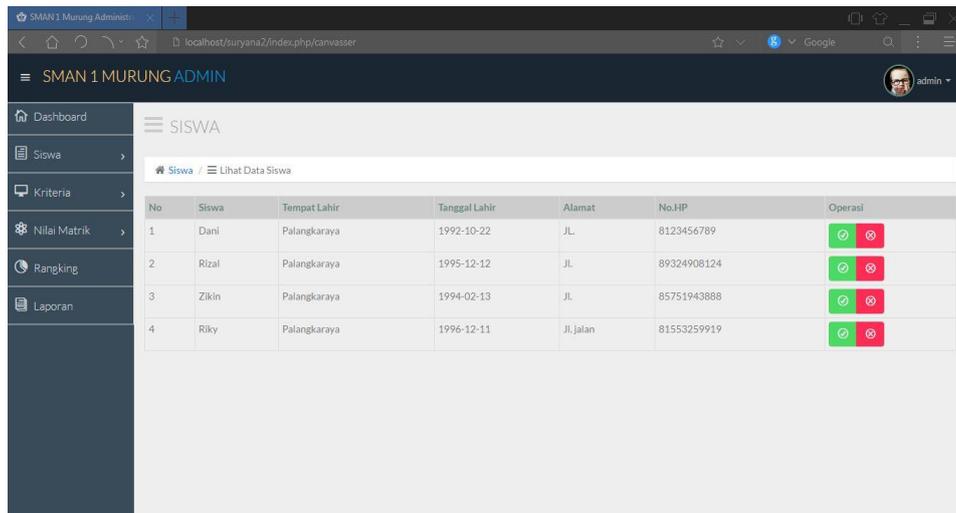
Halaman beranda merupakan halaman yang pertama kali akan muncul saat pengunjung membuka sistem pendukung keputusan seleksi jurusan pada SMAN 1 Murung Raya.

Tabel 8. Pengujian Halaman Pemilihan Jurusan

No	Pengujian	Data Masukan	Hasil Yang diharapkan	Hasil	Kesimpulan
1	Menu Beranda	Klik menu beranda	Tampil halaman beranda	Halaman beranda tampil	Sesuai
2	Menu Pemilihan Jurusan	Klik menu data Pemilihan Jurusan	Tampil halaman Pemilihan Jurusan	Halaman Pemilihan Jurusan tampil	Sesuai
3	Tombol Tambah Pemilihan Jurusan	Klik tombol tambah Pemilihan Jurusan	Tampil halaman tambah Pemilihan Jurusan	Halaman tambah Pemilihan Jurusan tampil	Sesuai
4	Tombol Edit	Klik tombol edit	Tampil halaman edit	Halaman edit tampil	Sesuai
5	Tombol hapus	Klik tombol hapus	Tampil halaman hapus	Halaman hapus tampil	Sesuai



Gambar 28. Beranda Admin

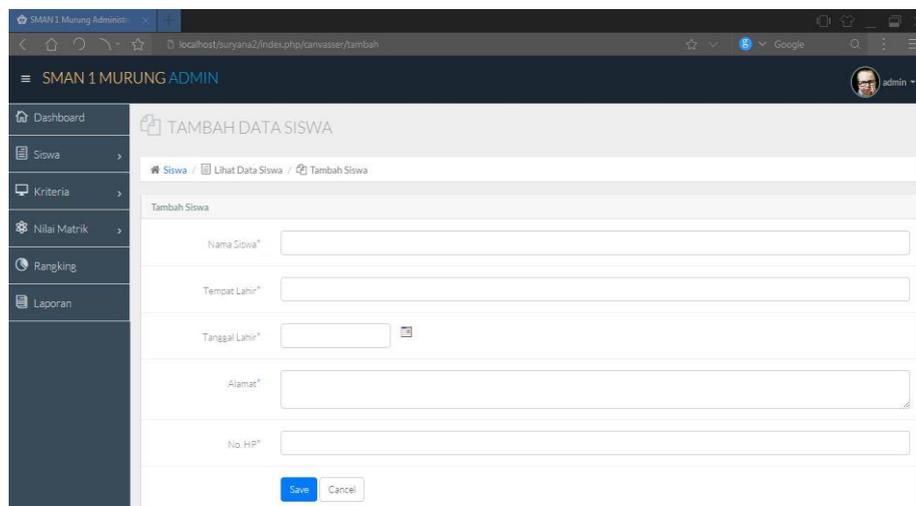


The screenshot shows the SMAN 1 MURUNG ADMIN dashboard. The left sidebar contains navigation options: Dashboard, Siswa, Kriteria, Nilai Matrik, Rangkings, and Laporan. The main content area is titled 'SISWA' and displays a table with the following data:

No	Siswa	Tempat Lahir	Tanggal Lahir	Alamat	No.HP	Operasi
1	Dani	Palangkaraya	1992-10-22	JL.	8123456789	+ -
2	Rizal	Palangkaraya	1995-12-12	JL.	89324908124	+ -
3	Zikin	Palangkaraya	1994-02-13	JL.	85751943888	+ -
4	Riky	Palangkaraya	1996-12-11	Jl. Jalan	81553259919	+ -

Gambar 27. Data Pemilihan Jurusan

Pada halaman ini pengunjung sebelum melihat data seleksi jurusan pada SMAN-1 Murung Raya.



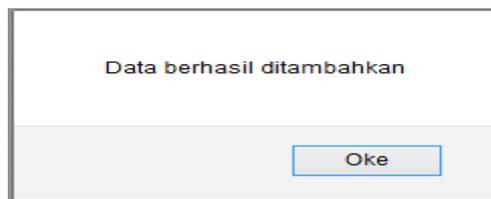
The screenshot shows the 'TAMBAH DATA SISWA' form in the SMAN 1 MURUNG ADMIN dashboard. The form contains the following fields:

- Nama Siswa*
- Tempat Lahir*
- Tanggal Lahir*
- Alamat*
- No. HP*

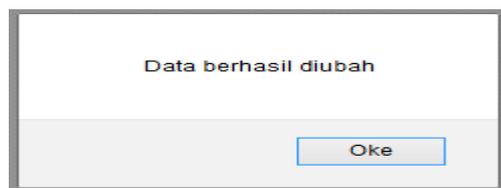
At the bottom of the form, there are 'Save' and 'Cancel' buttons.

Gambar 28. Tambah Data Pemilihan Jurusan

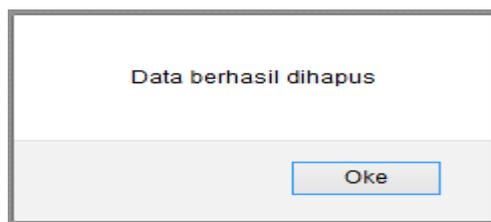
Pada halaman *admin* dapat menambahkan data data siswa pada halaman tambah data siswa pada SMAN-1 Murung Raya.



Gambar 29. Data Berhasil Ditambahkan



Gambar 30. Data Berhasil Diubah



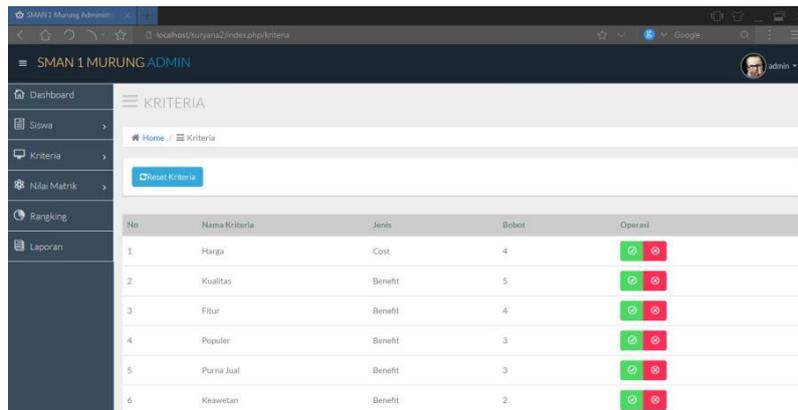
Gambar 31. Data Berhasil Dihapus

4) Pengujian Halaman Menu Kriteria

Pada halaman ini *admin* dapat menambahkan data kriteria jurusan pada sekolah SMAN 1 Murung Raya.

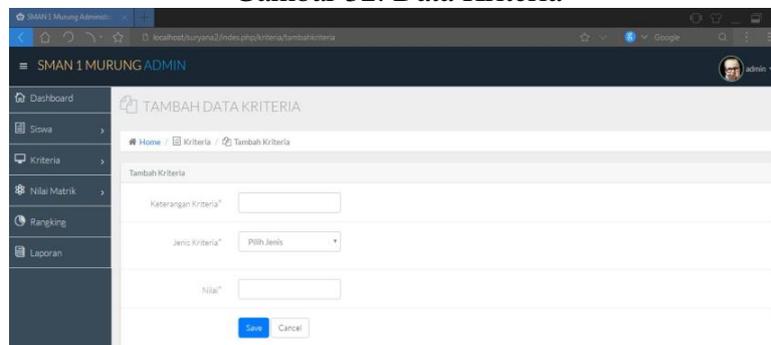
Tabel 9. Pengujian halaman Menu Kriteria

1	Menu Kriteria	Klik menu kriteria	Tampil halaman kriteria	Halaman kriteria tampil	Sesuai
2	Menu Tambah Kriteria	Klik menu tambah kriteria	Tampil halaman tambah kriteria	Halaman tambah kriteria tampil	Sesuai
3	Tombol Edit	Klik tombol edit	Tampil halaman edit	Halaman edit tampil	Sesuai
4	Tombol hapus	Klik menu hapus	Tampil halaman hapus	Halaman hapus tampil	Sesuai

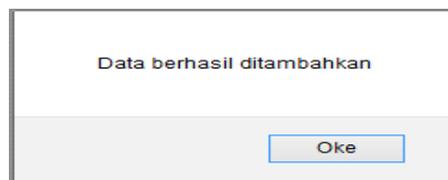


No	Nama Kriteria	Jenis	Bobot	Operasi
1	Harga	Cost	4	 
2	Kualitas	Benefit	5	 
3	Fitur	Benefit	4	 
4	Populer	Benefit	3	 
5	Purna Jual	Benefit	3	 
6	Keawetan	Benefit	2	 

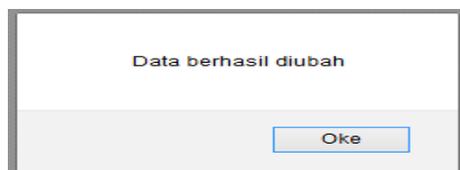
Gambar 32. Data Kriteria



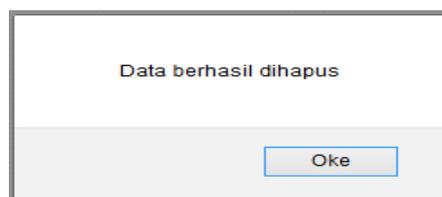
Gambar 33. Tambah Data Kriteria



Gambar 34. Data Berhasil Ditambahkan



Gamabr 35. Data Berhasil Diubah



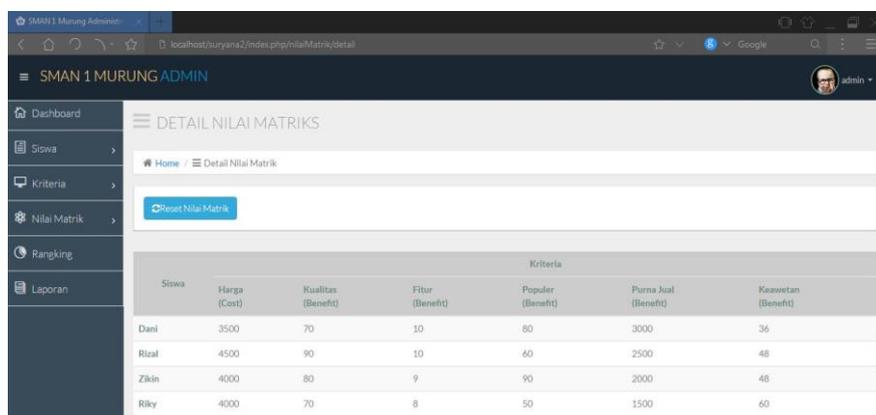
Gambar 36. Data Berhasil Dihapus

Pada halaman ini *admin* dapat melihat data kriteria yang dalam *database*. *Admin* juga dapat dapat menambah data kriteria, lalu bagian operasi *admin* dapat mengubah data kriteria dan menghapus data kriteria.

5) Pengujian Halaman Menu Nilai Matrik

Tabel 10. Halaman Menu Nilai Matrik

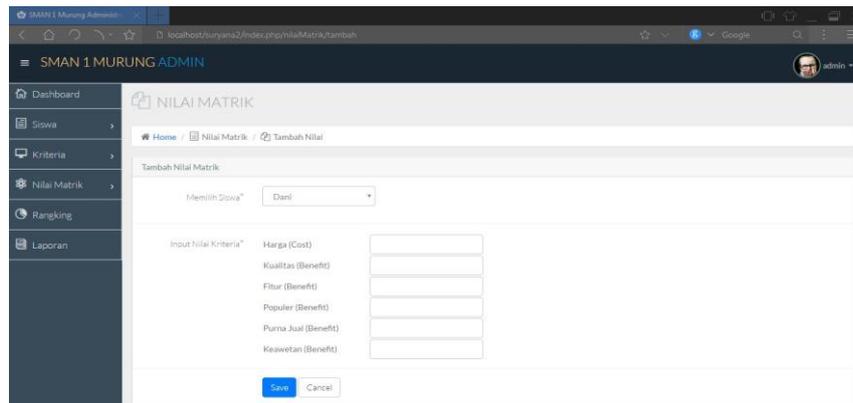
1	Menu Tabel	Klik menu tabel	Tampil halaman tabel	Halaman tabel tampil	Sesuai
2	Tombol Tambah Tabel	Klik tombol tambah tabel	Tampil halaman tambah tabel	Halaman tambah tabel tampil	Sesuai
3	Tombol <i>Reset</i> Tabel	Klik tombol <i>reset</i> tabel	Tampil halaman <i>reset</i> tabel	Halaman <i>reset</i> tabel tampil	Sesuai



Siswa	Kriteria					
	Harga (Cost)	Kualitas (Benefit)	Fitur (Benefit)	Populer (Benefit)	Harga Jual (Benefit)	Keawetan (Benefit)
Dani	3500	70	10	80	3000	36
Rizal	4500	90	10	60	2500	48
Zikri	4000	80	9	90	2000	48
Riky	4000	70	8	50	1500	60

Gambar 37. Tabel Nilai Matrik

Pada halaman ini *admin* dapat melihat data proses SPK tabel Nilai Matrik yang ada dalam *database*. Lalu bagian operasinya *admin* juga dapat menambah data percocokan kriteria dan juga dapat mereset data percocokan kriteria.



Gambar 38. Tambah Nilai Matrik

Pada halaman ini *admin* menambah Nilai Matrik dengan mengisi data yang diperlukan lalu klik save , jika ingin menghapus data yang terisi klik reset dan data dapat diisi kembali.

6) Pengujian Halaman Menu Ranking

Halaman ini merupakan hasil rekomendasi yang sebelumnya dilakukan perhitungan bobot dalam tabel Ranging.

Tabel 11. Halaman Menu Ranking

1	Menu Ranking	Klik menu ranking	Tampil menu ranking	Halaman ranking tampil	Sesuai
2	Menu Grafik	Klik menu ranking	Tampil halaman menu ranking	Halaman ranking tampil	Sesuai
3	Menu Tabel Ranking	Klik menu tabel ranking	Tampil halaman tabel ranking	Halaman tabel ranking tampil	Sesuai

Siswa	Kriteria						Hasil
	Harga (Cost)	Kualitas (Benefit)	Fitur (Benefit)	Populer (Benefit)	Purna Asal (Benefit)	Keawetan (Benefit)	
Rizal	3.111111111111111	5	4	2	2.5	1.6	0.62557097895135
Zikin	3.5	4.444444444444444	3.6	3	2	1.6	0.54099198706757
Dani	4	3.888888888888889	4	2.666666666666667	3	1.2	0.54208385783509

Gambar 39. Menu Halaman Ranking

Siswa	Kriteria			Hasil
	nilai rapot (Benefit)	nilai pelajaran (Benefit)	perilaku (Benefit)	
Rizal	75	80	75	1
Riky	75	80	75	1

Gambar 40. Menu Halaman Tabel IPA

Siswa	Kriteria			Hasil
	nilai rapot (Benefit)	nilai pelajaran (Benefit)	perilaku (Benefit)	
Zikin	75	65	70.3125	0.4001067188456
Dani	65.625	70	65.625	0.2314100998277

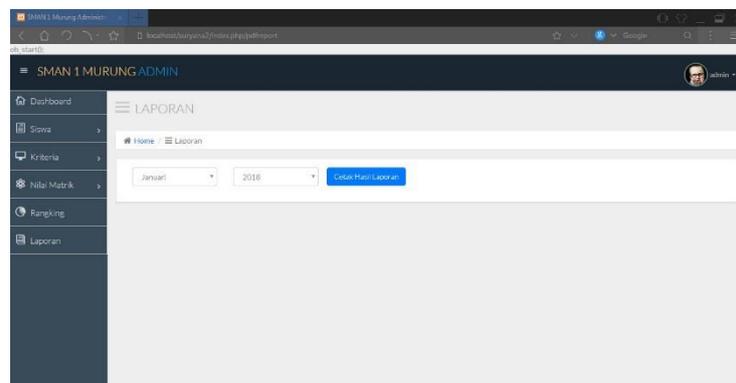
Gambar 41. Menu Tabel Halaman IPS

7) Pengujian Halaman Menu Laporan

Pada halaman ini pengujung dapat melihat hasil laporan penjurusan bulan dan tahun.

Tabel 12. Halaman Menu Laporan

1	Menu Laporan	Klik menu rekomedasi	Tampil menu laporan	Halaman laporan tampil	Sesuai
2	Menu datar Pemilihan Jurusan	Klik menu datar Pemilihan Jurusan	Tampil halaman menu datar Pemilihan Jurusan	Halaman menu datar Pemilihan Jurusan	Sesuai

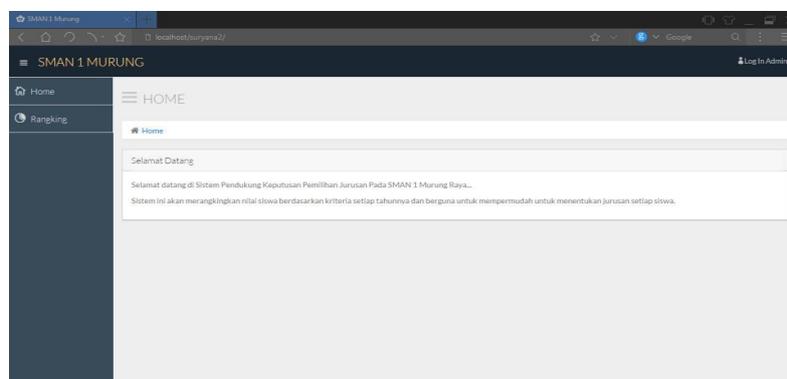


Gambar 34. Beranda Pengunjung Menu Laporan

2. Manual Pogram

a. Halaman Beranda Pegunjung

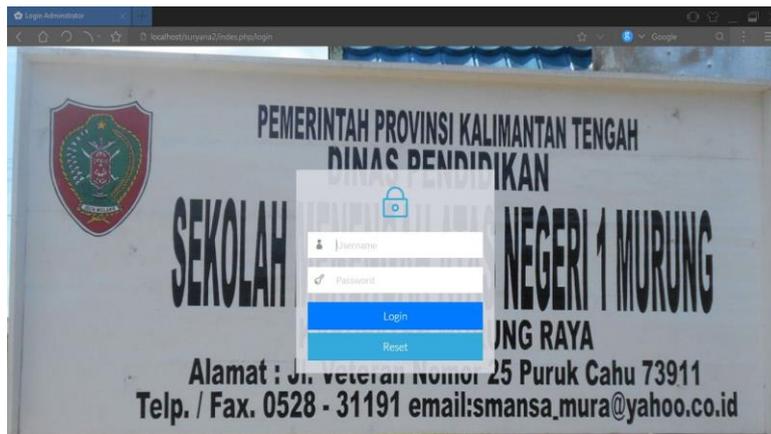
Halaman beranda merupakan haaman yang pertama kali akan muncul saat pengunjung membuka sistem pendukung keputusan pemilihan jurusan.



Gambar 35. Beranda Pengunjung

b. Halaman *Login Admin*

Halaman *Login admin* digunakan untuk *login* kehalaman admin, dengan mengisi *username* dan *password* yang sudah terdaftar di database lalu klik tombol *login*.



Gambar 36. Halaman *Login Admin*

c. Halaman Calon Pemilihan Jurusan

Pada halaman ini *admin* dapat melihat data calon Pemilih Jurusan yang ada dalam *database*. *Admin* juga dapat menambah data Pemilihan Jurusan, menghapus data Pemilih Jurusan dan mengubah data Pemilih Jurusan.

No	Siswa	Tempat Lahir	Tanggal Lahir	Alamat	No.HP	Operasi
1	Dani	Palangkaraya	1992-10-22	JL.	8123456789	  
2	Rizal	Palangkaraya	1995-12-12	JL.	89324908124	  
3	Zikin	Palangkaraya	1994-02-13	JL.	85751943888	  
4	Riky	Palangkaraya	1996-12-11	Jl. jalan	81553259919	  

Gambar 37. Halaman calon siswa Pemilihan Jurusan

d. Halaman Tambah Kriteria

Pada halaman ini *admin* menambah data kriteria dengan mengisi data yang di perlukan lalu klik *save*. Jika tidak ingin menambahkan data yang telah terisi klik *cancel* dan data tidak akan ditambahkan.

The screenshot shows a web browser window with the URL `localhost/suryana2/index.php/kanvas/tambah`. The page title is 'SMAN 1 MURUNG ADMIN' and the user is logged in as 'admin'. The main content area is titled 'TAMBAH DATA SISWA' and contains a form with the following fields:

- Nama Siswa*
- Tempat Lahir*
- Tanggal Lahir*
- Alamat*
- No. HP*

At the bottom of the form are two buttons: 'Save' (blue) and 'Cancel' (white).

Gambar 38. Halaman Tambah kriteria

e. Halaman Data Kriteria

Pada halaman ini *admin* dapat melihat data kriteria yang dalam *database*. *Admin* juga dapat dapat menambah data kriteria, lalu bagian operasi *admin* dapat mengubah data kriteria dan menghapus data kriteria.

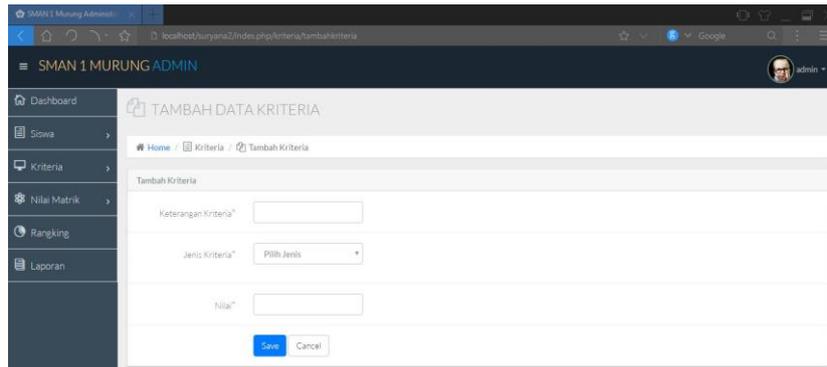
The screenshot shows a web browser window with the URL `localhost/suryana2/index.php/kriteria`. The page title is 'SMAN 1 MURUNG ADMIN' and the user is logged in as 'admin'. The main content area is titled 'KRITERIA' and contains a table with the following data:

No	Nama Kriteria	Jenis	Bobot	Operasi
1	Harga	Cost	4	 
2	Kualitas	Benefit	5	 
3	Fitur	Benefit	4	 
4	Populer	Benefit	3	 
5	Purna Jual	Benefit	3	 
6	Kawetan	Benefit	2	 

Gambar 39. Halaman Data Kriteria

f. Halaman Tambah Data Kriteria

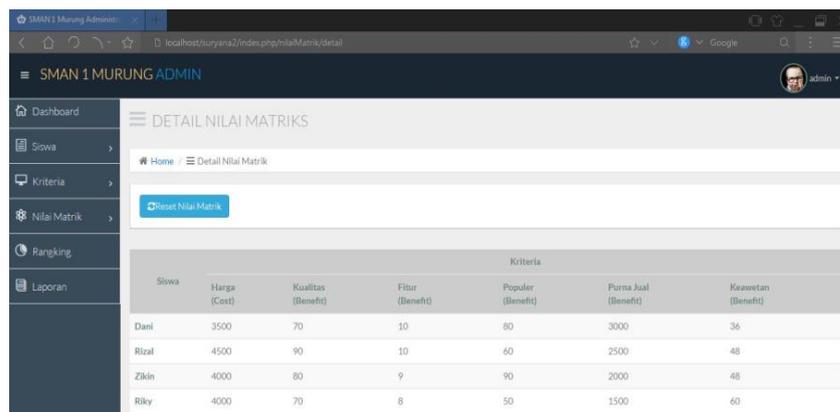
Pada halaman ini *admin* menambah data sub kriteria dengan mengisi data yang diperlukan lalu klik *save*, jika tidak ingin menambahkan data yang telah terisi klik *cancel* dan data tidak akan ditambahkan.



Gambar 40. Halaman Tambah Data Kriteria

g. Halaman Nilai Matrik

Pada halaman ini *admin* dapat melihat data tabel nilai matrik yang ada dalam *database*. Lalu bagian operasinya *admin* juga dapat menambah data nilai matrik dan juga dapat mereset data nilai matrik.



Siswa	Kriteria					
	Harga (Cost)	Kualitas (Benefit)	Fitur (Benefit)	Populer (Benefit)	Purna Jual (Benefit)	Kuawatan (Benefit)
Dani	3500	70	10	80	3000	36
Rizal	4500	90	10	60	2500	48
Zikri	4000	80	9	90	2000	48
Riky	4000	70	8	50	1500	60

Gambar 41. Halaman Tabel Nilai Matrik

h. Halaman Tambah Alternatif

Pada halaman ini *admin* menambah alternatif dengan mengisi data yang diperlukan lalu klik *save* , jika tidak ingin menambahkan data yang terisi klik *cancel* dan data tidak akan ditambahkan.

Gambar 42. Halaman Tambah Nilai Matrik

i. Halaman Rangkaian

Halaman ini *admin* dapat melihat data rangkaian yang ada dalam *database*.

Siswa	Kriteria						Hasil
	Harga (Cost)	Kualitas (Benefit)	Fitur (Benefit)	Populer (Benefit)	Purna Jual (Benefit)	Keawetan (Benefit)	
Rizal	3.11111111111111	5	4	2	2.5	1.6	0.42557097895135
Zikri	3.5	4.44444444444444	3.6	3	2	1.6	0.54099199705757
Dani	4	3.88888888888889	4	2.66666666666667	3	1.2	0.54208385763509

Gambar 43. Halaman Ranking

3. Perhitungan SAW dan TOPSIS

Perhitungan penelitian pemiklian calon pemilihan jurusan dilakukan berdasarkan kriteria yang telah ditentukan dan setiap kriteria memiliki bobot perhitungan. Dimisalkan ada 50 siswa SMAN 1 Murung yang menjadi alternatif dalam pemilihan seleksi Jurusan terbaik ada 4 (empat) kriteria yang menjadi acuan dalam pengambilan keputusan yaitu:

a. Analisis Proses Perhitungan

Analisa acuan dalam membangun sistem pendukung keputusan ini berdasarkan penilaian secara umum dilakukan pada proses pemilihan jurusan. Dimana dalam penilaian ini setiap siswa akan dinilai berdasarkan kriteria dan *alternative*, di bawah ini adalah table kriteria dan alternatif yang akan di uji dengan metode SAW (*Simple Additive Weighting*) . Diperlukana kriteria-kriteria dan bobot untuk melakukan perhitungannya sehingga akan di dapat *alternative* terbaik.

Dalam metode penelitian ini ada bobot dan kriteria yang dibutuhkan untuk menentukan siapa yang akan terseleksi sebagai calon pemilih jurusan. Dimana dari suatu variable tersebut akan dirubah kedalam bilangan fuzzy, metode ini dipilih karena mampu menyeleksi alternatif terbaik dari sejumlah alternatif. Data alternatif yang di gunakan :

- 1) Sangat Rendah = 5
- 2) Rendah = 4
- 3) Sedang = 3
- 4) Tinggi = 2
- 5) Sangat Tinggi = 1

Tabel 19 : Tabel Ranting Kecocokan Dari Setiap Alternatif Pada Setiap Kriteria

Alternatif	Kriteria			
	C1	C2	C3	C4
Ryan	4	3	3	3
Fatimah	2	4	3	3
Andi	3	5	2	3
Yenita	4	3	2	3

1. Metode SAW

Langkah 1 : Membuat Matrik Keputusan X,

$$X = \begin{bmatrix} X_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \dots & \dots & \dots & \dots \\ x_{m1} & x_{m2} & \dots & x_{mn} \end{bmatrix}$$

Sehingga diperoleh :

$$X = \begin{bmatrix} 4 & 3 & 3 & 3 \\ 2 & 4 & 3 & 3 \\ 3 & 5 & 2 & 3 \\ 4 & 3 & 2 & 3 \end{bmatrix}$$

Langkah 2 : Mementukan Normalisasi Matriks keputusan R,

$$R_{11} = \frac{4}{\text{Max}(4234)} = \frac{4}{4} = 1$$

$$R_{12} = \frac{2}{\text{Max}(4234)} = \frac{2}{4} = 0,5$$

$$R_{13} = \frac{3}{\text{Max}(4234)} = \frac{3}{4} = 0,75$$

$$R_{14} = \frac{4}{\text{Max}(4234)} = \frac{4}{4} = 1$$

$$R = \begin{pmatrix} 0.30 & 0.12 & 0.30 & 0.20 \\ 0.15 & 0.16 & 0.30 & 0.20 \\ 0.23 & 0.20 & 0.20 & 0.20 \\ 0.30 & 0.12 & 0.20 & 0.20 \end{pmatrix}$$

2. Metode TOPSIS

Langkah 1 : Normalisasi Matriks Terbobot y berdasarkan nilai setiap elemen pada matrik R yang di peroleh metode SAW.

$$Y_{ij} = W_i r_{ij}$$

Sehingga :

$$Y_{11} = 0.3 \times 1.00 = 0.30$$

$$Y_{12} = 0.15 \times 0.50 = 0.07$$

$$Y_{13} = 0.75 \times 0.23 = 0.17$$

$$Y_{14} = 0.3 \times 0.10 = 0.23$$

$$Y = \begin{bmatrix} 0.30 & 0.12 & 0.30 & 0.20 \\ 0.15 & 0.16 & 0.30 & 0.20 \\ 0.23 & 0.20 & 0.20 & 0.20 \\ 0.30 & 0.12 & 0.20 & 0.20 \end{bmatrix}$$

Langkah 1 ; Penggabungan Metode SAW dan TOPSIS, menentukan solusi idea positif (A^+)

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

$$Y_1^+ = \max\{0.30; 0.15; 0.23; 0.30\} = 0.30$$

$$Y_2^+ = \max\{0.12; 0.16; 0.20; 0.12\} = 0.20$$

Diperoleh hasil :

$$A^+ = \{0.30; 0.20; 0.30; 0.20\}$$

Langkah 2 : : menentukan Solusi idea negatif (A^-), dengan persamaan

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

$$Y_1^- = \min\{0.30; 0.15; 0.23; 0.30\} = 0.15$$

$$Y_2^- = \min\{0.12; 0.16; 0.20; 0.12\} = 0.12$$

Di peroleh hasil :

$$A^- = \{0.15; 0.12; 0.20; 0.20\}$$

Langkah 3 : Menentukan Jarak Terbobot setiap alternatif terhadap solusi

idea positif ($S1^+$), dengan persamaan

$$D_i^+ = \sqrt{\sum_{j=1}^n (y_i + -y_{ij})^2};$$

$$D_1^+ = \sqrt{(0.30 - 0.30)^2 + (0.12 - 0.20)^2 + (0.30 - 0.30)^2 + (0.20 - 0.20)^2} = 0.08$$

$$D_2^+ = \sqrt{(0.15 - 0.30)^2 + (0.16 - 0.20)^2 + (0.30 - 0.30)^2 + (0.20 - 0.20)^2} = 0.16$$

Di peroleh hasil :

$$D_1^+ \text{ Ryan} = 0.08$$

$$D_2^+ \text{ Fatima} = 0.16$$

$$D_3^+ \text{ Andy} = 0.13$$

$$D_4^+ \text{ Yenita} = 0.33$$

Langkah 4 : Menentukan Jarak Terbobot setiap alternatif terhadap solusi
idea negatif ($S1^-$), dengan persamaan

$$D_i^- = S$$

$$D_1^- = \sqrt{(0.30 - 0.30)^2 + (0.12 - 0.20)^2 + (0.30 - 0.30)^2 + (0.20 - 0.20)^2} = 0.18$$

$$D_2^- = \sqrt{(0.15 - 0.15)^2 + (0.12 - 0.20)^2 + (0.30 - 0.30)^2 + (0.20 - 0.20)^2} = 0.11$$

Hasil di peroleh :

$$D_1^- \text{ Ryan} = 0.18$$

$$D_2^- \text{ Fatima} = 0.11$$

$$D_3^- \text{ Andy} = 0.11$$

$$D_4^- \text{ Yenita} = 0.15$$

Langkah 5 : Menentukan Nilai preferensi untuk setiap alternatif, dengan persamaan

$$V_i = \frac{D_i^-}{D_i^- + D_i^+};$$

Di peroleh :

$$V_1 \text{ Ryan} = \frac{0.18}{0.18+0.08} = 0.69$$

$$V_2 \text{ Fatima} = \frac{0.11}{0.11+0.16} = 0.41$$

$$V_3 \text{ Andy} = \frac{0.11}{0.11+0.18} = 0.47$$

$$V_4 \text{ Yenita} = \frac{0.15}{0.15+0.33} = 0.32$$

4. Manual Instansi

a. *XAMPP*

XAMPP adalah sebuah perangkat lunak atau software yang didalamnya sudah *membundle* banyak aplikasi lain yang dibutuhkan dalam pengembangan web, seperti Apache, Mysql, Filezilla, dll. Nama *XAMPP* merupakan singkatan dari aplikasi utama yang ada didalamnya:

- 1) *X (cross-platform)*, dimana aplikasi *XAMPP* tersedia untuk banyak Sistem Operasi karena *XAMPP* bisa dijalankan di 4 OS besar yang sering digunakan oleh pengguna komputer saat ini. Dan 4 OS tersebut tidak lain dan tidak bukan adalah Windows, Linux, Mac OS dan Solaris.
- 2) *A (Apache)* merupakan aplikasi web server. *Apache* ini bersifat opensource yang berarti gratis dan bisa diedit oleh penggunanya. Tugas utama *Apache* adalah menghasilkan

halaman web yang benar kepada user berdasarkan kode PHP yang dituliskan oleh pembuat halaman web. jika diperlukan juga berdasarkan kode PHP yang dituliskan, maka dapat saja suatu database diakses terlebih dahulu (misalnya dalam *MySQL*) untuk mendukung halaman web yang dihasilkan.

- 3) *M(MySQL)*, merupakan aplikasi database server. Perkembangannya disebut SQL yang merupakan kepanjangan dari Structured Query Language. SQL merupakan bahasa terstruktur yang digunakan untuk mengolah database. MySQL dapat digunakan untuk membuat dan mengelola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam database.
- 4) *P(PHP)*, bahasa pemrograman *web*. Bahasa pemrograman *PHP* merupakan bahasa pemrograman untuk membuat web yang bersifat server-side scripting. *PHP* memungkinkan kita untuk membuat halaman web yang bersifat dinamis. Sistem manajemen basis data yang sering digunakan bersama *PHP* adalah *MySQL*. Namun *PHP* juga mendukung sistem manajemen database Oracle, Microsoft *Access*, *Interbase*, *dbase*, *PostgreSQL*, dan sebagainya.
- 5) *P(Perl)*, bahasa pemrograman, pertama kali dikembangkan oleh Larry Wall di mesin Unix. Perl pertama kali dirilis pada tanggal 18 Desember 1987 ditandai dengan keluarnya Perl 1.

Dua diantara karakteristik utama perl adalah penanganan teks dan berbagai jalan pintas untuk meyelesaikan persoalan-persoalan umum. Perl sangat populer di gunakan dalam program-rogram CGI (Common Gateway Interface) dan protokol internet lainnya.

b. Fungsi *Xampp*

Xampp Server berfungsi untuk pengembangan *website* berbasis *PHP* dan *MySQL*, bahkan *Software* ini memiliki kelebihan untuk bisa berperan sebagai *server web Apache* untuk simulasi pengembangan *website*. Tool pengembangan web ini mendukung teknologi web populer seperti *PHP*, *MySQL*, dan *Perl*. Melalui program ini, *programmer web* dapat menguji aplikasi web yang dikembangkan dan mempresentasikannya ke pihak lain secara langsung dari komputer, tanpa perlu terkoneksi ke internet. *XAMPP* juga dilengkapi fitur manajemen database *PHP MyAdmin* seperti pada *server hosting* sungguhan, sehingga pengembang web dapat mengembangkan aplikasi web berbasis database.



Gambar 44. *XAMPP V3.22*

5. Pemeliharaan Sistem

Dalam melakukan pemeliharaan pada sistem informasi ini ada beberapa hal yang akan penulis lakukan, diantaranya adalah sebagai berikut:

- a. Melakukan perbaikan pada sistem apabila ketika saat sistem tersebut sudah digunakan terjadi *error* atau ada *bug* pada struktur kode program penyusun sistem.
- b. Memastikan bahwa sistem berjalan dengan baik, dengan cara mengecek satu persatu fungsi-fungsi tombol pada sistem apakah sudah sesuai dengan yang diharapkan atau belum.
- c. Sistem operasi disini memegang peran yang sangat penting karena berpengaruh pada kinerja dari sistem yang dibuat. Agar sistem dapat berjalan dengan baik tentunya didukung pula dengan sistem operasi yang baik, dalam hal ini sistem operasi yang penulis gunakan untuk membangun sistem pendukung keputusan pemilihan jurusan di SMAN-1 Murung Raya.

BAB V

KESIMPULAN DAN SARAN

A. Kesimpulan

Berdasarkan hasil dan pembahasan pada bab-bab sebelumnya, seperti analisis dan pengujian aplikasi yang dibangun ini maka penulis dapat menarik kesimpulan yaitu :

1. Penggabungan Metode *Simple Additive Weighting (SAW)* Dan *Technique For Order Preference By Similarity (TOPSIS)* Berbasis *CI* dapat diimplementasikan dalam Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada SMAN 1 Murung dan memberikan rekomendasi kepada pengambil keputusan berupa ranking calon siswa berdasarkan bobot yang diperoleh masing masing siswa.
2. Sistem Pendukung Keputusan Seleksi Pemilihan Jurusan Pada SMAN 1 Murung Menggunakan Metode *Simple Additive Weighting (SAW)* Dan *Technique For Order Preference By Similarity (TOPSIS)* Berbasis *CI* berhasil dibangun .
3. Hasil perhitungan manual dibandingkan dengan hasil keluaran Sitem Pendukung Keputusan memiliki selisih yang sangat kecil. Hal ini menunjukkan bawah Sistem Pendukung Keputusan yang dibangun dapat memproses perhitungan bobot
4. Pada sistem ini kriteria dan bobot penilaian dapat ditambahkan, dihapus, diperbaharui, atau dikelola sesuai dengan ketentuan penilaian yang diperlukan.

B. Saran

Dari kesimpulan diatas, adapun saran yang dapat diberikan penulis untuk kedepannya, penulis mengharapkan sebagai berikut :

1. Untuk mencegah rusak ataupun hilangnya data pada *file database*, sebaiknya dapat dilakukan *back up* data secara berkala dan melakukan *scaning* terhadap virus yang merusak.
2. Diharapkan kedepannya sistem ini dapat dikembangkan dengan pemodelan sistem pendukung keputusan (*SPK*) yang lain, seperti metode penggabungan *WP (Weighted Pruduct)* dan *AHP (Analitic Hierarchy Process)* atau *Profil Maching* dan *TOPSIS (Technicue for Order Prefence by Similarity to Ideal Solution)*.
3. Diharapkan sistem ini dapat dikembangkan tidak hanya untuk memberikan keputusan pada pemilihan jurusan saja tetapi dapat dikembangkan untuk memberikan keputusan pada lomba lain yang terdapat pada sekolah.