**IMPLEMENTASI METODE WEIGHTED PRODUCT PADA SISTEM PENDUKUNG KEPUTUSAN PENERIMA BEASISWA BIDIKMISI UNIVERSITAS MATARAM**

Tugas Akhir

Untuk memenuhi sebagian persyaratan

Mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



**Oleh :**

**FARAZ WAHYUDI**

**F1D015023**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MATARAM**

**Juli 2020**

**TUGAS AKHIR**

**IMPLEMENTASI METODE WEIGHTED PRODUCT PADA SISTEM PENDUKUNG KEPUTUSAN PENERIMA BEASISWA BIDIKMISI UNIVERSITAS MATARAM**

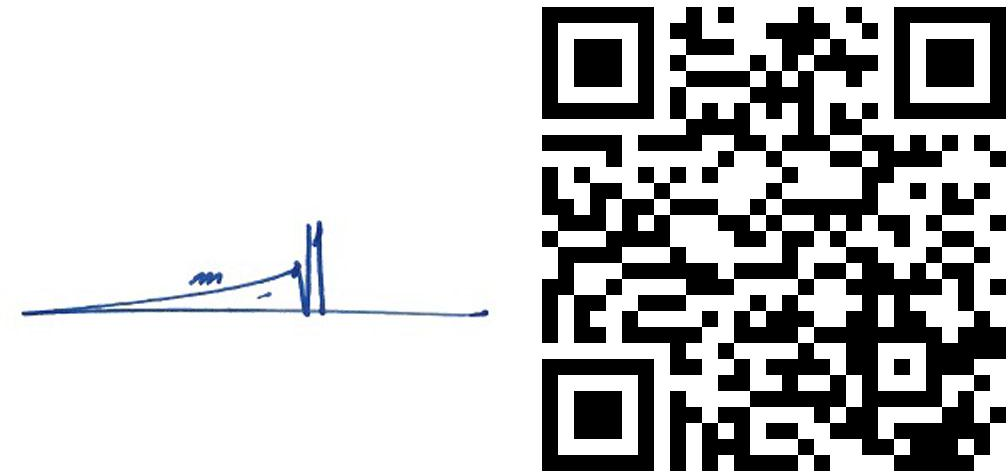
Oleh:

**FARAZ WAHYUDI**

**F1D015023**

Telah diperiksa oleh Tim Pembimbing :

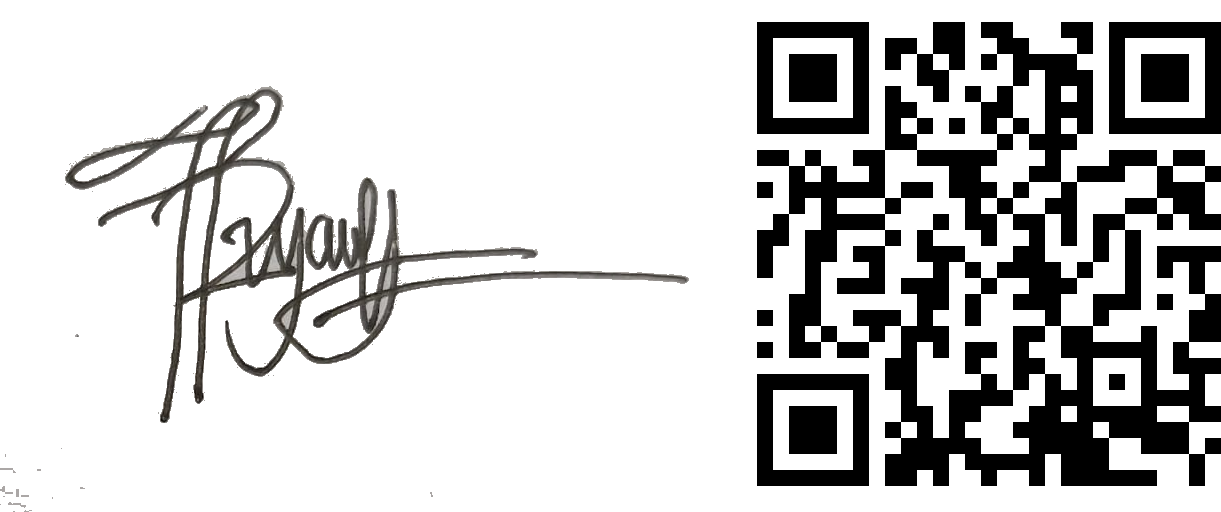
1. Pembimbing Utama

Tanggal: 17/07/2020

**Moh. Ali Albar, ST., M.Eng.**

**NIP. 198311252015041002**

1. Pembimbing Pendamping

Tanggal: 17/07/2020

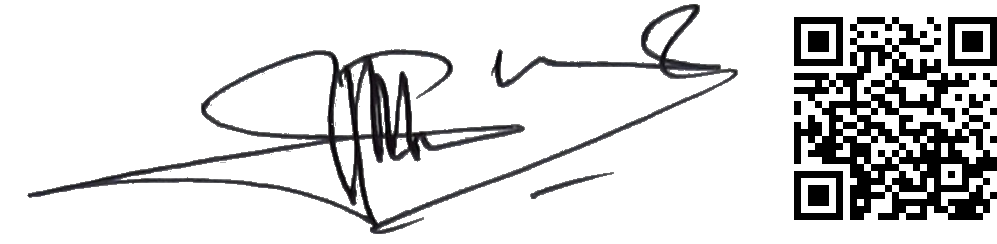
**Royana Afwani, ST., MT.**

**NIP. 198507072014042001**

Mengetahui,

****Ketua Program Studi Teknik Informatika Fakultas Teknik

Universitas Mataram



**Prof. Dr. Eng. I Gede Pasek Suta Wijaya, ST., MT. NIP. 197311302000031001**

**TUGAS AKHIR**

**IMPLEMENTASI METODE WEIGHTED PRODUCT PADA SISTEM PENDUKUNG KEPUTUSAN PENERIMA BEASISWA BIDIKMISI UNIVERSITAS MATARAM**

Oleh:

**FARAZ WAHYUDI**

**F1D015023**

Telah diujikan di depan penguji

Pada tanggal 7 Juli 2020

Dan dinyatakan telah memenuhi syarat mencapai derajat Sarjana S-1

Program Studi Teknik Informatika

Susunan Tim Penguji:

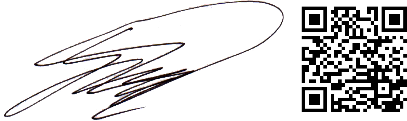
1. Penguji 1



Tanggal: 16/07/2020

**Nadiyasari Agitha, S.Kom.,M.MT**

**NIP. 198608132018032001**

1. Penguji 2

Tanggal: 15/07/2020

**Gibran Satya Nugraha, S.Kom., M.Eng.**

**NIP. 199203232019031012**

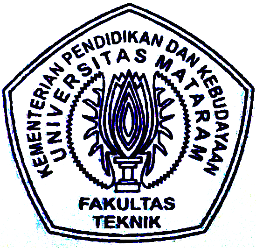
1. Penguji 3

Tanggal: 17/07/2020

**Ir. Sri Endang Anjarwani, M.Kom**

**NIP. 196604032006042001**

Mataram, 21 Juli 2020

Dekan Fakultas Teknik

Universitas Mataram



**Akmaluddin, S.T., M.Sc.(Eng.), Ph.D.**

**NIP. 196812311994121001**

# **PERNYATAAN KEASLIAN TUGAS AKHIR**

Saya yang bertanda tangan di bawah ini bahwa dalam Tugas Akhir ini tidak  
terdapat karya yang pernah diajukan untuk memperoleh gelar kesarjanaan di suatu  
Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau  
pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis  
diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Mataram, Juli 2020

Faraz Wahyudi

# **PRAKATA**

Puji syukur penulis panjatkan ke hadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan  
penyusunan Tugas Akhir dengan judul “Implementasi Metode *Weighted Product* Pada Sistem Pendukung Keputusan Penerima Beasiswa Bidikmisi Universitas Mataram”

Tugas Akhir ini dilaksanakan di Fakultas Teknik Universitas Mataram.  
Tujuan dari Tugas Akhir ini adalah untuk merancang dan membangun sistem untuk menhasilkan perangkingan mahasiswa peserta bidikmisi Universitas Mataram tahun 2018. Tugas Akhir ini juga merupakan salah satu persyaratan kelulusan guna  
mencapai gelar kesarjanaan di Program Studi Teknik Informatika Fakultas Teknik  
Universitas Mataram

Penulisan skripsi ini tentunya tidak terlepas dari kekurangan, baik dari aspek  
kualitas maupun kuantitas dari materi penelitian yang disajikan. Semua ini  
didasarkan dari keterbatasan yang dimiliki penulis. Penulis menyadari bahwa  
skripsi ini jauh dari sempurna sehingga penulis membutuhkan kritik dan saran yang  
bersifat membangun demi kemajuan teknologi di masa yang akan datang.  
Akhir kata semoga tidaklah terlampau berlebihan, bila penulis berharap agar  
karya ini dapat bermanfaat bagi pembaca.

Mataram, Juli 2020

Faraz Wahyudi

# **UCAPAN TERIMA KASIH**

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materil dari berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada :

1. Allah SWT atas segala nikmat kesempatan, kesehatan, dan anuerah yang telah diberikan selama pembuatan Tugas Akhir ini.
2. Kedua Orang Tua Penulis, selaku pemberi dukungan utama yang selalu memberikan do’a dan dukungan baik moril maupun materil yang tidak putus-putus kepada penulis sehingga penulis dapat menyelesaikan pembuatan Tugas Akhir ini dengan baik.
3. Keluarga penulis yang telah memberikan dukungan dan doanya selama penulis menjalani perkuliahan sampai dengan pengerjaan Tugas Akhir ini dapat diselesaikan dengan baik.
4. Bapak Moh. Ali Albar, ST., M.Eng. selaku dosen pembimbing utama yang telah memberikan bimbingan dan arahan kepada penulis selama menyusun Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
5. Ibu Royana Afwani, ST., M.T. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan selama menyusun Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.
6. Bapak dan Ibu dosen penguji yang telah memberikan kritik dan saran yang bersifat membangun dalam menyelesaikan Tugas Akhir ini.
7. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan dukungan baik moril maupun materil kepada penulis dalam menyelesaikan Tugas Akhir ini.

Semoga Allah SWT memberikan imbalan yang setimpal atas bantuan yang diberikan kepada penulis.

# **DAFTAR ISI**

[**PERNYATAAN KEASLIAN TUGAS AKHIR** iv](#_Toc45876003)

[**PRAKATA** v](#_Toc45876004)

[**UCAPAN TERIMA KASIH** vi](#_Toc45876005)

[**DAFTAR ISI** vii](#_Toc45876006)

[**DAFTAR GAMBAR** ix](#_Toc45876007)

[**DAFTAR TABEL** xi](#_Toc45876008)

[**ABSTRAK** xii](#_Toc45876009)

[**ABSTRACK** xiii](#_Toc45876010)

[**BAB I** 14](#_Toc45876011)

[**1.1** **Latar Belakang** 14](#_Toc45876012)

[**1.2** **Rumusan Masalah** 16](#_Toc45876013)

[**1.3** **Batasan Masalah** 16](#_Toc45876014)

[**1.4** **Tujuan** 16](#_Toc45876015)

[**1.5** **Manfaat** 17](#_Toc45876016)

[**1.6** **Sistematika** 17](#_Toc45876017)

[**BAB II** 18](#_Toc45876018)

[**2.1** **Tinjauan Pustaka** 18](#_Toc45876019)

[**2.2** **Dasar Teori** 20](#_Toc45876020)

[**2.2.1** **Sistem Pendukung Keputusan** 20](#_Toc45876021)

[**2.2.2** **Metode *Weighted Product*** 20](#_Toc45876022)

[**2.2.3** ***Website*** 21](#_Toc45876023)

[**2.2.4** **UML** 22](#_Toc45876024)

[**2.2.5** ***Framework CodeIgniter*** 27](#_Toc45876025)

[**2.2.6** **ERD** 28](#_Toc45876026)

[**2.2.7** **MySQL** 29](#_Toc45876027)

[**BAB III** 30](#_Toc45876028)

[**3.1** **Metodologi Penelitian** 30](#_Toc45876029)

[**3.2** **Studi Literatur** 30](#_Toc45876030)

[**3.3** **Metode Pengumpulan Data** 31](#_Toc45876031)

[**3.4** **Analisis Sistem** 31](#_Toc45876032)

[**3.4.1** **Analisis Sistem yang Berjalan** 31](#_Toc45876033)

[**3.4.2** **Analisis Prosedur Sistem yang Berjalan** 31](#_Toc45876034)

[**3.4.3** **Analisis Prosedur Sistem yang Akan Dibangun** 33](#_Toc45876035)

[**3.4.4** **Analisis Kebutuhan Sistem** 33](#_Toc45876036)

[**3.5** **Metode *Weighted Product*** 34](#_Toc45876037)

[**3.6** **Perancangan Sistem** 41](#_Toc45876038)

[**3.6.1** ***Usecase Diagram*** 41](#_Toc45876039)

[**3.6.2** ***Class Diagram*** 42](#_Toc45876040)

[**3.6.3** ***Sequence Diagram*** 43](#_Toc45876041)

[**3.6.4** ***Activity Diagram*** 47](#_Toc45876042)

[**3.6.5** ***Entinity Relationship Diagram* (ERD)** 53](#_Toc45876043)

[**3.6.6** **Desain Tampilan Sistem** 55](#_Toc45876044)

[**3.7** **Pembuatan Sistem** 58](#_Toc45876045)

[**3.8** **Pengujian Sistem** 58](#_Toc45876046)

[**3.8.1** **Metode *White Box Testing*** 58](#_Toc45876047)

[**3.8.2** **Metode *Black Box Testing*** 58](#_Toc45876048)

[**BAB IV** 59](#_Toc45876049)

[**4.1** **Implementasi Sistem** 59](#_Toc45876050)

[**4.1.1** **Implementasi *database*** 59](#_Toc45876051)

[**4.1.2** **Implementasi *class*** 61](#_Toc45876052)

[**4.1.3** **Implementasi *interface* sistem** 66](#_Toc45876053)

[**4.1.4** **Implementasi Hasil Perangkingan** 73](#_Toc45876054)

[**4.2** **Pengujian Sistem** 77](#_Toc45876055)

[**4.2.1** **Pengujian *black box*** 77](#_Toc45876056)

[**4.2.2** **Pengujian *white box*** 78](#_Toc45876057)

[**4.2.3** **Pengujian dengan kuisioner** 83](#_Toc45876058)

[**BAB V** 86](#_Toc45876059)

[**DAFTAR PUSTAKA** 87](#_Toc45876060)

# **DAFTAR GAMBAR**

[Gambar 2. 1 *Fish bone* tinjauan pustaka. 19](#_Toc40445813)

[Gambar 2. 2 Arsitektur MVC. 28](#_Toc40445814)

[Gambar 3. 1 Diagram alir penelitian…………………………………………………...30](#_Toc45514946)

[Gambar 3. 2 *Flowmap* sistem yang sedang berjalan 32](#_Toc45514947)

[Gambar 3. 3 Alur sistem yang akan dibangun 33](#_Toc45514948)

[Gambar 3. 4 Alur perhitungan metode *weighted product.* 34](#_Toc45514949)

[Gambar 3. 5 *Usecase diagram* 42](#_Toc45514950)

[Gambar 3. 6 *Class diagram* 42](#_Toc45514951)

[Gambar 3. 7 *Sequence diagram login*. 43](#_Toc45514952)

[Gambar 3. 8 *Sequence diagram* tambah data mahasiswa. 43](#_Toc45514953)

[Gambar 3. 9 *Sequence diagram* hapus data mahasiswa. 44](#_Toc45514954)

[Gambar 3. 10 *Sequence diagram* edit data mahasiswa. 44](#_Toc45514955)

[Gambar 3. 11 *Sequence diagram* tambah data admin. 45](#_Toc45514956)

[Gambar 3. 12 *Sequence diagram* hapus data admin. 45](#_Toc45514957)

[Gambar 3. 13 *Sequence diagram* edit data admin. 46](#_Toc45514958)

[Gambar 3. 14 *Sequence diagram* edit data bobot. 46](#_Toc45514959)

[Gambar 3. 15 *Activity diagram login.* 47](#_Toc45514960)

[Gambar 3. 16 *Activity diagram* tambah data mahasiswa. 48](#_Toc45514961)

[Gambar 3. 17 *Activity diagram edit* data mahasiswa. 49](#_Toc45514962)

[Gambar 3. 18 *Activity diagram* hapus data mahasiswa. 49](#_Toc45514963)

[Gambar 3. 19 *Activity diagram* tambah admin. 50](#_Toc45514964)

[Gambar 3. 20 *Activity diagram edit* data admin. 51](#_Toc45514965)

[Gambar 3. 21 *Activity diagram* hapus data mahasiswa. 51](#_Toc45514966)

[Gambar 3. 22 *Activity diagram* edit data bobot. 52](#_Toc45514967)

[Gambar 3. 23 ER *diagram.* 53](#_Toc45514968)

[Gambar 3. 24 Desain halaman *login*. 55](#_Toc45514969)

[Gambar 3. 25 Desain halaman *dashboard*. 55](#_Toc45514970)

[Gambar 3. 26Desain halaman kelola mahasiswa. 56](#_Toc45514971)

[Gambar 3. 27Desain halaman kelola kriteia. 56](#_Toc45514972)

[Gambar 3. 28Desain halaman hasil seleksi. 57](#_Toc45514973)

[Gambar 3. 29 *Form input* data mahasiswa. 57](#_Toc45514974)

[Gambar 3. 30 *Form input* data kriteria. 58](#_Toc45514975)

[Gambar 4. 1 Implementasi *database*…………………………………………………...59](#_Toc45515042)

[Gambar 4. 2 Tabel admin 59](#_Toc45515043)

[Gambar 4. 3 Tabel kriteria 60](#_Toc45515044)

[Gambar 4. 4 Tabel mahasiswa 60](#_Toc45515045)

[Gambar 4. 5 *Class controller* pada sistem 61](#_Toc45515046)

[Gambar 4. 6 *Script* proses *login* sistem 62](#_Toc45515047)

[Gambar 4. 7 *Script* hitung bobot normalisasi 62](#_Toc45515048)

[Gambar 4. 8 *Script* metode WP 63](#_Toc45515049)

[Gambar 4. 9 *Class* model pada sistem 64](#_Toc45515050)

[Gambar 4. 10 *Script* validasi *user login* 64](#_Toc45515051)

[Gambar 4. 11 *Script* tambah admin 65](#_Toc45515052)

[Gambar 4. 12 *Class view* pada sistem 65](#_Toc45515053)

[Gambar 4. 13 Potongan *script* menampilkan hasil perangkingan 66](#_Toc45515054)

[Gambar 4. 14 Beranda untuk mahasiswa 67](#_Toc45515055)

[Gambar 4. 15 Tampilan hasil seleksi 67](#_Toc45515056)

[Gambar 4. 16 Sitemap tampilan untuk admin 68](#_Toc45515057)

[Gambar 4. 17 Tampilan halaman *login* 69](#_Toc45515058)

[Gambar 4. 18 Tampilah halaman *dashboard* 69](#_Toc45515059)

[Gambar 4. 19 Tampilan menu admin 70](#_Toc45515060)

[Gambar 4. 20 Tampilan profil 70](#_Toc45515061)

[Gambar 4. 21 *form* edit profil admin 70](#_Toc45515062)

[Gambar 4. 22 Halaman data admin 71](#_Toc45515063)

[Gambar 4. 23 Halaman tambah admin 71](#_Toc45515064)

[Gambar 4. 24 Halaman data mahasiswa 71](#_Toc45515065)

[Gambar 4. 25 Menampilkan detail informasi mahasiswa 72](#_Toc45515066)

[Gambar 4. 26 Menampilkan *form* tambah data mahasiswa 72](#_Toc45515067)

[Gambar 4. 27 Tampilan halaman kriteria dan bobot 72](#_Toc45515068)

[Gambar 4. 28 Tampilan data skor mahasiswa 73](#_Toc45515069)

[Gambar 4. 29 Tampilan hasil perangkingan 73](#_Toc45515070)

[Gambar 4. 30 *Basis* *path* proses menghitung bobot normalisasi 79](#_Toc45515071)

[Gambar 4. 31 *Basis path* proses menghitung vektor s dan vektor v 81](#_Toc45515072)

# **DAFTAR TABEL**

[Tabel 2. 1 Simbol *use case* diagram 22](#_Toc45514862)

[Tabel 2. 2 Simbol-simbol *diagram* aktivitas. 23](#_Toc45514863)

[Tabel 2. 3 Simbol *class diagram.* 24](#_Toc45514864)

[Tabel 2. 4 Simbol-simbol *sequence diagram.* 26](#_Toc45514865)

[Tabel 2. 5 Simbol ERD. 28](#_Toc45514866)

[Tabel 3. 1 Data mahasiswa……………………………………………………………..35](#_Toc45711172)

[Tabel 3. 2 Tabel Kriteria. 36](#_Toc45711173)

[Tabel 3. 3 Tingkat Bobot Kriteria. 36](#_Toc45711174)

[Tabel 3. 4 Tabel subkriteria. 37](#_Toc45711175)

[Tabel 3. 5 Tabel bobot dan kriteria setelah normalisasi. 38](#_Toc45711176)

[Tabel 3. 6 Tabel konversi nilai data mahasiswa. 39](#_Toc45711177)

[Tabel 3. 7 Hasil perangkingan WP. 41](#_Toc45711178)

[Tabel 3. 8 Tabel Mahasiswa 53](#_Toc45711179)

[Tabel 3. 9 Tabel admin 54](#_Toc45711180)

[Tabel 3. 10 Tabel Kriteria 54](#_Toc45711181)

[Tabel 4. 1 Hasil perhitungan sistem……………………………………………………74](#_Toc45514878)

[Tabel 4. 2 Tabel Hasil SPK 75](#_Toc45514879)

[Tabel 4. 3 Pengujian *black box testing* 78](#_Toc45514880)

[Tabel 4. 4 Pengujian proses perhitugan bobot 80](#_Toc45514881)

[Tabel 4. 5 *Path* proses menghitung vektor s dan vektor v 82](#_Toc45514882)

[Tabel 4. 6 Pengujian proses perhitugan vektor s dan vektor v 82](#_Toc45514883)

[Tabel 4. 7 Keterangan Penilaian Kusioner 84](#_Toc45514884)

[Tabel 4. 8 Hasil Perhitungan MOS Pegawai 84](#_Toc45514885)

[Tabel 4. 9 Hasil Perhitungan MOS Mahasiswa 85](#_Toc45514886)

# **ABSTRAK**

Pemberian beasiswa merupakan program kerja yang ada di setiap Universitas atau perguruan tinggi. Bidikmisi adalah program bantuan biaya pendidikan yang diberikan pemerintah melalui Direktorat Jenderal Pendidikan Tinggi. Dalam melakukan penetapan mahasiswa peserta beasiswa bidikmisi pihak Universitas masih menggunakan cara manual yaitu dengan cara melakukan pengecekan kembali data mahasiswa satu persatu di *Microsoft excel* oleh Subbagian Kesejahteraan Mahasiswadan hasil seleksi data diurutkan berdasarkan fakultas atau prodi masing-masing. Untuk membantu dalam proses seleksi maka perlu dibangun sistem pendukung keputusan, agar dapat menjadi pertimbangan dalam mengambil keputusan. Dalam sistem pendukung keputusan ini menggunakan 10 kriteria antara lain: penghasilan ayah, penghasilan ibu, jumlah tanggungan, kepemilikan rumah, luas tanah, luas bangunan, jarak pusat kota, mandi cuci kakus, sumber listrik dan sumber air. Didalam sistem pendukung keputusan ini perhitungannya menggunakan metode *weighted product.* Metode *weighted produc*t digunakan karena metode ini lebih bisa memberikan nilai bobot disetiap kriteria kemudian dilakukan perangkingan dari yang terbesar ke terkecil dari hasil perhitungan nilai akhir. Dalam pembangunan sistem ini, menggunakan bahasa pemograman PHP dan *database* yang digunakan *mysql* dengan menggunakan *framework Codeigniter.* Dalam sistem ini menggunakan 100 data mahasiswa yang diambil dari data file *Microsoft excel.* Seratus data tersebut diambil secara berurutan dari data nomor 1 sampai 100. Sistem dapat menghasilkan prioritas dalam membantu untuk melakukan keptusan, tetapi sebelum itu dilakukan survei lapangan terkait data-data yang ada.

**Kata Kunci** : *Weighted Product,* Beasiswa Bidikmisi, Sistem Pendukung Keputusan, PHP, MySQl, *Framework Codeigniter*

# **ABSTRACK**

*The awarding of scholarships is a work program that exists in every university or college. Bidikmisi is an education cost assistance program provided by the government through the Directorate General of Higher Education. In determining the students who take Bidikmisi scholarship, the University still uses the manual method by checking the student data one by one in Microsoft Excel by the Student Welfare Subdivision and the results of the data selection are sorted by their respective faculties or study programs. To assist in the selection process, a decision support system needs to be built, so that it can be considered in making decisions. In this decision support system using 10 criteria, among others: father's income, mother's income, number of dependents, home ownership, land area, building area, distance of the city center, bathing, washing, electricity and water sources. In this decision support system the calculation uses the weighted product method. The weighted productive method is used because this method is more able to provide weight values in each criterion, then ranking of the largest to the smallest from the calculation of the final value. In the development of this system, using the PHP programming language and database used mysql using the Codeigniter framework. In this system uses 100 student data taken from Microsoft Excel data files. One hundred of these data are taken sequentially from data numbers 1 to 100. The system can generate priority in helping to make decisions, but before that a field survey is carried out in relation to existing data.*

***Keywords****: Weighted Product, Bidikmisi Scholarship, Decision Support System, Codeigniter framework, mysql, PHP*

# **BAB I**

**PENDAHULUAN**

## **Latar Belakang**

Pendidikan merupakan suatu kebutuhan primer yang sejak dini hingga dewasa hendaknya dirasakan oleh seluruh masyarakat. Hal ini sesuai dengan amanat Undang-Undang Dasar (UUD) Negara Indonesia, anjuran agama, dan menjadi penentu kemajuan suatu bangsa. Pendidikan juga merupakan variabel vital untuk pembangunan suatu bangsa. Suatu bangsa bisa maju dengan cepat dibandingkan dengan negara lain karena penyebaran pengetahuan (*knowledge*) yang merata keseluruh lapisan masyarakatnya. Institusi yang paling bertanggung jawab untuk penyebaran pengetahuan adalah institusi Pendidikan. Untuk menyelenggarakan pendidikan yang bermutu diperlukan biaya yang cukup besar. Oleh karena itu setiap peserta didik pada satuan pendidikan berhak mendapatkan bantuan biaya pendidikan bagi mereka yang memiliki potensi akademik baik dan tidak mampu secara ekonomi serta berhak mendapatkan beasiswa bagi mereka yang berprestasi [1].

Beasiswa adalah pemberian berupa bantuan keuangan yang diberikan kepada perorangan, mahasiswa, atau pelajar yang digunakan demi keberlangsungan pendidikan yang ditempuh. Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan,yayasan atau instansi-instansi yang lain. Salah satunya beasiswa yang terdapat di perguruan tinggi yaitu bidikmisi. Program beasiswa bidikmisi adalah program bantuan biaya pendidikan bagi calon mahasiswa tidak mampu secara ekonomi dan memiliki potensi akademik baik untuk menempuh pendidikan di perguruan tinggi [2].

Untuk mendapatkan beasiswa bidikmisi calon peserta yaitu mahasiswa harus memenuhi persyaratan dan kriteria-kriteria yang telah ditetapkan oleh Ristekdikti selaku penyelenggara beasiswa bidikmisi. Hasil seleksi dari Ristekdikti akan dikirimkan ke Universitas masing-masing dan Universitas diberikan wewenang untuk menyeleksi kembali hasil keputusan dari Ristekdikti. Universitas Mataram merupakan salah satu Universitas yang menyelenggarakan program beasiswa bidikmisi. Dalam melakukan penetapan mahasiswa peserta beasiswa bidikmisi pihak Universitas masih menggunakan cara manual yaitu dengan cara melakukan pengecekan kembali data mahasiswa satu persatu di *Microsoft excel* oleh Subbag Kesejahteraan Mahasiswadan hasil seleksi yang dilakukan secara manual selanjutnya akan diputusakan dalam rapat yang diadakan diruangan wakil rektor III. Hasil akhir dari data penerima beasiswa bidikmisi diurutkan berdasarkan fakultas atau prodi dengan NIM masing-masing mahasiswa penerima beasiswa bidikmisi. Banyaknya kriteria yang menjadi acuan dalam menentukan mahasiswa penerima beasiswa bidikmisi antara lain penghasilan orang tua, jumlah tanggungan, kepemilikan rumah, luas tanah, luas bangunan, mandi cuci kakus, sumber air, sumber listrik dan jarak pusat kota. Sehingga dapat memakan waktu cukup lama. Oleh karena itu, perlu adanya suatu sistem yang mendukung proses penentuan penerima beasiswa, sehingga dapat mempersingkat waktu penyeleksian dan dapat meningkatkan kualitas keputusan dalam penentuan penerima beasiswa yaitu sistem pendukung keputusan.

Sistem Pendukung Keputusan (SPK) adalah bagian dari sistem informasi berbasis komputer termasuk sistem berbasis pengetahuan atau menajemen pengetahuan yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan[2]. Dalam sistem pendukung keputusan terdapat beberapa metode seperti SAW *(Simple Additive Weighting)*, WP *(Weighted Product)*, AHP *(Analytic Hierarchy Process)*, ELECTRE *(Elimination and Choice Expressing Reality)*, TOPSIS *(Technique for Order by Similarity to Ideal Solution)*, *Fuzzy* dan lain-lain [3]. Pada penelitian yang mengukur performa SAW dan *Weighted Product* yang dilakukan oleh Dewi Kusmawati pada tahun 2015 menyimpulkan bahwa penggunaan SAW dan WP sangat potensial untuk SPK pemberian beasiswa [4]. Metode yang digunakan dalam sistem pendukung keputusan ini adalah metode *weighted product*. Metode *weighted product* ini di pilih karena metode tersebut menentukan nilai bobot untuk setiap atribut, kemudian dilanjutkan dengan proses pemeringkatan yang akan menyeleksi alternatif terbaik dari sejumlah alternatif, dalam hal ini alternatif yang dimaksud adalah mahasiswa yang berhak menerima beasiswa berdasarkan kriteria-kriteria yang di tentukan [5]. Pada peneltian sistem seleksi pengangkatan karyawan tetap metode WP (*Weighted Product*) terbukti lebih efisien dengan rata-rata waktu yang dibutuhkan sebesar 1.6408 detik, jika dibandingkan dengan metode SAW (*Simple Additive Weighting*) yang membutuhkan waktu rata-rata 6.5822 detik. Metode WP (*Weighted Product*) terbukti lebih efisien dengan rata-rata selisih waktu sebesar 4.9414 detik dalam hal waktu yang dibutuhkan untuk melakukan perhitungan [6].

Dari permasalahan yang telah dipaparkan diatas peneliti membuat Sistem Pendukung Keputusan dengan judul penelitian “Implementasi metode *Weighted Product* pada Sistem Pendukung Keputusan Penerima Beasiswa Bidikmisi Universitas Mataram” yang diharapkan dapat membantu dalam penentuan mahasiswa penerima beasiswa bidikmisi secara cepat untuk menghasilkan *output* yang dapat digunakan sebagai sarana penunjang keputusan.

## **Rumusan Masalah**

Bagaimana merancang Sistem Pendukung Keputusan Penerima Beasiswa Bidikmisi Universitas Mataram dengan menggunakan metode *Weighted Product* sebagai pertimbangan dalam pengambilan keputusan.

## **Batasan Masalah**

Dari permasalahan yang telah dipaparkan terdapat batasan-batasan masalah dalam merancang dan membangun sistem ini sebagai berikut :

1. Sistem pendukung keputusan di bangun menggunakan *framework* *Codeigniter* dan MySQL sebagai *database*.
2. Sistem pendukung keputusan yang dibangun sebagai pertimbangan dalam pengambilan keputusan penerima beasiswa bidikmisi.
3. Menggunakan 10 kriteria sebagai acuan proses seleksi, antara lain penghasilan orang tua, jumlah tanggungan, kepemilikan rumah, luas tanah, luas bangunan, mandi cuci kakus, sumber air, sumber listrik dan jarak pusat kota.
4. Menggunakan data mahasiswa Universitas Mataram angkatan 2018.
5. Sistem yang dibangun akan digunakan oleh Subbagian Kesejahteraan Mahasiswa, Wakil Rektor III dan Mahasiswa.

## **Tujuan**

Merancang dan membangun sistem pendukung keputusan dengan menggunakan metode *Weighted Product* yang dapat memudahkan dalam mengolah data serta *output* yang dihasilkan berupa perangkingan sebagai pertimbangan dalam pengambilan keputusan.

## **Manfaat**

Adapun manfaat dari penelitian ini sebagai berikut:

1. Menunjang mahasiswa dalam menyelesaikan tugas akhir.
2. Bagi pihak rektorat dapat mempermudah dalam pengolahan data dan mempercepat menghasilkan *output* calon mahasiswa penerima beasiswa bidikmisi sebagai pertimbangan dalam pengambilan keputusan.

## **Sistematika**

Adapun sistematika laporan ini adalah sebagai berikut :

1. PENDAHULUAN

Bab ini berisi latar belakang penulisan tugas akhir, rumusan masalah, batasan masalah, tujuan , manfaat dan sistematika.

1. TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini menjelaskan mengenai penelitian yang dilakukan sebelumnya dan teori yang berkaitan dengan judul/masalah tugas akhir antara lain sistem pendukung keputusan, *website*, *unified modelling language*, *use case diagram, class**diagram, activity diagram, sequence diagram, framework codeigniter,* dan *entity relation diagram.*

1. METODE PENELITIAN

Bab ini berisi pembahasan mengenai metodologi penelitian, studi literatur, metode pengumpulan data, analisis sistem, peracangan sistem, pembuatan sistem dan pengujian sistem.

1. HASIL DAN PEMBAHASAN

Bab ini berisi hasil pembuatan sistem, pelaksanaan metode *weighted product* dan hasil perhitungan sistem.

1. KESIMPULAN DAN SARAN

Bab ini berisi beberapa kesimpulan yang didapat dan saran-saran berdasarkan hasil penelitian

# **BAB II**

**TINJAUAN PUSTAKA DAN DASAR TEORI**

## **Tinjauan Pustaka**

Beberapa penelitian yang terkait dengan sistem yang dibangun dengan menggunakan metode *Weighted Product* sebagai berikut:

Berdasarkan penelitian tentang “Penerapan Metode PROMETHEE Dalam Sistem Penunjang Keputusan Penentuan Penerima Beasiswa Bidik Misi Universitas Halu Oleo”. Dalam pembangunan sistem ini, peneliti menggunakan bahasa pemograman *java netbeans* dan *database* yang digunakan *mysql*. Dalam melakukan perhitungan menggunakan tujuh kriteria antara lain Pekerjaan ayah, pekerjaan ibu, penghasilan orangtua, tanggungan orang tua, kepemilikan rumah, luas tanah dan luas rumah. Dari hasil perbandingan antara hasil penentuan penerima beasiswa bidikmisi berdasarkan sistem menggunakan metode PROMETHEE dengan hasil penentuan berdasarkan keputusan UHO terhadap 24 data uji maka diperoleh 23 data calon penerima beasiswa Bidik Misi yang berhak menerima beasiswa tersebut [1].

Pada penelitian tentang “Implementasi Metode *Weighted Product* Menentukan Beasiswa Bidik Misi STMIK Pringsewu” dalam perhitungannya menggunakan lima kriteria yaitu indeks prestasi semester, penghasilan orang tua, jumlah tanggungan, pengabdian pkm, dan kelayakan rumah. Dalam penelitian ini menyimpulkan Sistem Pendukung Keputusan untuk menentukan penerimaan beasiswa Bidik Misi pada STMIK Pringsewu dapat membantu dan mempermudah Perguruan Tinggi dalam menentukan mahasiswa yang layak atau tidaknya mendapatkan beasiswa berdasarkan kriteria-kriteria yang telah ditentukan [7].

Penelitian tentang “Sistem Pendukung Keputusan Penentuan Penerimaan Mahasiswa Baru Jalur Bidikmisi Menggunakan Metode Topsis (Studi Kasus : Politeknik Negeri Malang) ”. Pengujian dilakukan dengan membandingkan data riil penerima yang telah dinyatakan diterima melalui jalur bidikmisi sebanyak 250 mahasiswa, kemudian dibandingkan dengan hasil perhitungan sistem sebanyak 250 mahasiswa. Hasil pengujian dari perhitungan sistem terdapat 108 mahasiswa yang seharusnya tidak diterima tapi diterima di data riil penerima bidikmisi tahun 2016. Dalam penelitian ini menyimpulkan sistem ini telah membantu tim seleksi dalam mengurangi kesalahan atau ketidak tepatan proses penilaian. Hasil pengujian dapat menyaring 56,8% dari data riil penerima bidikmisi atau 108 mahasiswa yang tidak diterima di bidikmisi [3].

Penelitian tentang “ Sistem Pendukung Keputusan Penerimaan Beasiswa Bidik Misi di POLIBAN Dengan Metode SAW Berbasis Web” dalam perhitungannya menggunakan kriteria-kriteria yaitu penghasilan orang tua, jumlah tanggungan, nilai rata-rata raport semester empat dan lima, bukti rekening listrik dan bukti pembayaran PBB. Dalam penelitian ini dihasilkan kesimpulan sistem pendukung keputusan penerimaan beasiswa bidik misi ini dapat dengan baik melakukan perangkingan [2].

Dari penelitian-penelitian sebelumnya yang telah dipaparkan maka akan dilakukan penelitian tentang “ Implementasi metode *Weigthted Product* pada Sistem Pendukung Keputusan Penerima Beasiswa Bidikmisi Universitas Mataram Berbasis *Web* dengan *Framework Codeigniter”* dalam pengembangan sistemnya nantinya akan menggunakan metode *waterfall.* Sistem ini dibuat berbasis web dengan menggunakan *framework codeigniter* serta dalam perhitungan menggunakan 10 kriteria-kriteria yang sesuai dengan persyaratan beasiswa bidikmisi dan yang terdapat pada data yang diberikan, diharapkan sistem yang dibuat memiliki tingkat akurasi yang tinggi sesuai dengan perhitungan manual. Pada gambar 2.1 ringkasan dari tinjauan pustaka dan penelitian yang akan dilakukan dalam bentuk *fish bone.*

|  |
| --- |
|  |

Gambar 2. 1 *Fish bone* tinjauan pustaka.

## **Dasar Teori**

Beberapa dasar teori yang terkait dengan penelitian Implementasi metode *Weighted Product* pada sistem pendukung keputusan penerima beasiswa bidik misi Universitas Mataram berbasis *web* dengan *framework codeigniter* sebagai berikut:

### **Sistem Pendukung Keputusan**

Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS) adalah sebuah sistem yang mampu memberikan kemampuan pemecahan masalah maupun kemampuan pengkomunikasian untuk masalah dengan kondisi semi terstruktur dan tak terstruktur. Sistem ini digunakan untuk membantu pengambilan keputusan dalam situasi semi terstruktur dan situasi yang tidak terstruktur, dimana tak seorangpun tahu secara pasti bagaimana keputusan seharusnya dibuat. SPK bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik [8].

### **Metode *Weighted Product***

*Weighted Product* (WP) adalah keputusan analisis multi-kriteria dan merupakan metode pengambilan keputusan multi kriteria. Seperti semua metode-metode lainanya, WP adalah himpunan dari alternatif keputusan yang dijelaskan dalam istilah beberapa kriteria. Metode perkalian atau metode WP ini berbeda dengan metode SAW dalam perlakuan awal terhadap hasil penilaian atribut keputusan. Dalam metode WP tidak diperlukan manipulasi matriks karena metode ini mengalikan hasil penilaian setiap atribut. Hasil perkalian tersebut belum dibandingkan (dibagi) dengan nilai standar, dalam hal ini alternatif ideal sering digunakan sebagai nilai standar bobot, bobot untuk atribut manfaat berfungsi sebagai pangkat positif dalam proses perkalian antar atribut, sementara bobot berfungsi sebagai pangkat negatif [9]. Terdapat variabel *cost* dan *benefit* berguna untuk menentukan kriteria dan berpengaruh dalam pengambilan keputusan. Kriteria yang termasuk *benefit*  jika nilai terbesar adalah yang terbaik (bobot bernilai *negatif*), sedangkan *cost* jika nilai terkecil adalah yang terbaik (bobot bernilai *positif*).

Metode *Weighted Product* menggunakan perkalian untuk menghubungkan rating atribut, dimana rating setiap atribut harus dipangkatkan terlebih dahulu dengan bobot atribut yang bersangkutan. Proses ini sama halnya dengan proses normalisasi [9]. Pembobotan metode *Weighted Product* dihitung berdasarkan tingkat nilai preferensi.

Proses normalisasi bobot kriteria (W), ΣW = 1 adalah:

(2-1)

Keterangan:

Wj: Bobot kriteria

wj : w indeks ke-j

ΣWj: Penjumlahan bobot kriteria

Menghitung Vektor S

(2-2)

Keterangan:

Si: nilai dari setiap altenatif

Xij: menyatakan nilai/skor kriteria

Wj: bobot setiap kriteria

i: menyatakan alternatif

j: menyatakan kriteria

n: menyatakan banyaknya kriteria

Menghitung Vektor V:

(2-3)

Keteragan:

V: menyatakan alternatif yang dianalogikan sebagai vektor V

Xij: menyatakan nilai/skor kriteria

Wj: bobot setiap kriteria

i: menyatakan alternatif

j: menyatakan kriteria

n: menyatakan banyaknya kriteria.

### ***Website***

*Website* adalah kumpulan halaman-halaman yang digunakan untuk mempublikasikan informasi berupa teks, gambar dan program multimedia lainnya berupa animasi (gambar gerak, tulisan gerak), suara dan atau gabungan dari semuanya itu baik yang bersifat statis maupun dinamis yang membentuk satu rangkaian bangunan yang saling terkait antara satu *page* dengan *page* yang lain yang sering disebut sebagai *hyperlink*. *Website* biasanya terangkum dalam sebuah domain atau subdomain yang tempatnya berada di dalam *World Wide Web* (www) di internet. Sebuah *Web Page* adalah dokumen yang ditulis dalam format HTML yang hampir selalu bisa diakses melalui HTTP, yaitu *protocol* yang bisa menyampaikan informasi dari *server website* untuk ditampilkan kepada para pemakai melalui *web browser*. Semua publikasi dari *website* tersebut dapat membentuk sebuah jaringan informasi yang sangat besar [10].

### **UML**

*Unified Modelling Language* (UML) adalah sebuah bahasa yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem [11]. Dengan menggunakan UML dapat membuat model untuk semua jenis aplikasi piranti lunak, dimana aplikasi tersebut dapat berjalan pada piranti keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan *class* dan *operation* dalam konsep dasarnya, maka lebih cocok untuk penulisan piranti lunak dalam bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk *modeling* aplikasi prosedural dalam VB atau C.

#### **Use Case Diagram**

*Use case diagram* menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah *use* *case* merepresentasikan sebuah interaksi antara aktor dengan sistem. *Use case* merupakan sebuah pekerjaan tertentu, misalnya *login* ke sistem, membuat sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [11]. Simbol-simbol *usecase diagram* dapat dilihat pada Tabel 2.1.

Tabel 2. 1 Simbol *use case* diagram

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
| http://1.bp.blogspot.com/_zcEi4XhVozw/TSvfhOH72TI/AAAAAAAABkE/vEmVLt4uLFs/s1600/1.+actor.jpg | *Actor* | *Actor* tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem. |
|  | *Use Case* | Gambaran fungsionalitas dari suatu sistem, sehingga*customer* atau pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun. |
|  | *Association* | Menghubungkan *link* antar *element.* |
|  | *Include* | Kejadian yang harus terpenuhi agar sebuah *event*dapat terjadi, di mana pada kondisi ini sebuah *usecase* adalah bagian dari *usecase* lainnya. |
|  | *Extends* | Relasi *usecase* tambahan ke sebuah *usecase* dimana *usecase* yang ditambahkan dapat berdiri sendiri. |
|  | *Generalization* | Disebut juga*inheritance* (pewarisan). |

#### **Activity Diagram**

*Activity diagram* menggambarkan berbagai alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi. *Activity diagram* merupakan *state diagram* khusus, dimana sebagian besar *state* adalah aksi dan sebagian besar transisi dibangkitkan oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behavior* internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum [11]. Simbol-simbol pada diagram aktivitas dapat dilihat pada Tabel 2.2.

Tabel 2. 2 Simbol-simbol *diagram* aktivitas.

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Status awal | Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal. |
| Aktivitas | Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja. |
| Percabangan / *Decision* | Asosiasi percabangan di mana jika ada pilihan aktivitas lebih dari satu. |
| Penggabungan / *Join* | Asosiasi penggabungan di mana lebih dari satu aktivitas digabungkan menjadi satu. |
| Status Akhir | Status akhir yang dilakukan sistem, sebuah diagram aktivitas harus memiliki status akhir. |
| *Swimlane*  Atau | Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi. |

#### **Class Diagram**

*Class* adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). *Class diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti *containment*, pewarisan, asosiasi, dan lain-lain [11].

Pada *class diagram* terdapat beberapa simbol dan beberapa cara penulisan diantaranya dapat dilihat pada Tabel 2.3.

Tabel 2. 3 Simbol *class diagram.*

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Deskripsi** |
|  | *Class* | Kelas pada struktur sistem. |
|  | Asosiasi */ Association* | Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan *multiplicity*. |
|  | Asosiasi berarah / *Directed association* | Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan *multiplicity*. |
|  | Generalisasi | Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus) |
|  | Kebergantungan / *dependency* | Relasi antar kelas dengan makna kebergantungan antar kelas. |
|  | Agregasi / *Aggregation* | Relasi antar kelas dengan makna semua-bagian (*whole-part*) |
|  | Komposisi / *Composition* | Agregasi yang memiliki hubungan yang lebih kuat |

#### **Sequence Diagram**

*Sequence diagram* menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. *Sequence diagram* terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). *Sequence diagram* biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu. Diawali dari apa yang membangkitkan aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan [11]. Simbol-simbol yang terdapat pada *sequence* *diagram* dapat dilihat pada Tabel 2.4.

Tabel 2. 4 Simbol-simbol *sequence diagram.*

|  |  |
| --- | --- |
| **Simbol** | **Deskripsi** |
| Aktor | Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi belum tentu merupakan orang, biasanya dinyatakan menggunakan kata benda di bawa frase nama aktor. |
| *Lifeline* | Menyatakan kehidupan suatu objek. |
| Objek | Menyatakan objek yang berinteraksi pesan. |
| Waktu aktif | Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya |
| Pesan tipe *create* | Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat. |
| Pesan tipe *call* | Menyatakan suatu objek memanggil operasi/ metode yang ada pada objek ain atau dirinya sendiri, arah panah mengarah pada objek yang memiliki operasi/ metode, operasi/ metode yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi. |
| Pesan tipe *send* | Menyatakan bahwa suatu objek mengirimkan data/ masukan/ informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi. |
| Pesan tipe return | Menyatakan bahwa suatu objek yang telah menjalankan suatu operasi atau metode menghasilkan suatu kembalian ke objek tertentu, arah panah mengarah pada objek yang menerima kembalian. |

### ***Framework CodeIgniter***

*CodeIgniter* adalah sebuah *framework* yang digunakan untuk membuat sebuah aplikasi berbasis *web* yang disusun dengan menggunakan bahasa PHP. Di dalam CI terdapat beberapa macam kelas (*class*) yang berbentuk *library* dan *helper*. Keduanya berfungsi untuk membantu pemrogram (*programmer*) dalam mengembangkan aplikasinya. *CodeIgniter* (CI) adalah *framework* pengembangan aplikasi dengan menggunakan PHP, suatu kerangka untuk bekerja atau membuat program dengan menggunakan PHP yang lebih sistematis. MVC adalah konsep dasar yang harus diketahui sebelum mengenal *CodeIgniter*. MVC adalah singkatan dari *Model View Controller*. MVC sebenarnya adalah sebuah Teknik pemrograman yang memisahkan alur bisnis, penyimpanan data dan antarmuka aplikasi atau secara sederhana adalah memisahkan antara desain, data dan proses [12].Arsiterktur MVC dapat dilihat pada Gambar 2.2.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu *model, view dan controller*.

1. *Model*

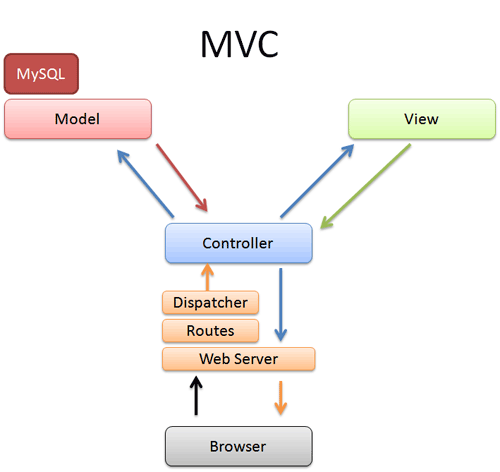
*Model* merupakan bagian penanganan yang berhubungan dengan pengolahan atau manipulasi *database*. seperti misalnya mengambil data dari *database*, meng-*input* dan pengolahan *database* lainnya. semua instruksi yang berhubung dengan pengolahan *database* di letakkan di dalam *model*.

1. *View*

*View* merupakan bagian yang menangani halaman *user interface* atau halaman yang muncul pada *user*. tampilan dari *user interface* di kumpulkan pada *view* untuk memisahkannya dengan *controller* dan *model* sehingga memudahkan *web designer* dalam melakukan pengembangan tampilan halaman *website*.

1. *Controller*

*Controller* merupakan kumpulan instruksi aksi yang menghubungkan model dan *view*, jadi *user* tidak akan berhubungan dengan *model* secara langsung, intinya dari *view* kemudian *controller* yang mengolah instruksi.



Gambar 2. 2 Arsitektur MVC.

### **ERD**

ERD (*Entity Relation Diagram)* merupakan sebuah diagram yang digunakan untuk merancang hubungan antar tabel-tabel dalam basis data.ERD (*Entity Relation Diagram)* berguna untuk menggambarkan gambaran dari dunia nyata yang akan diterapkan pada suatu *database* sebuah sistem. ERD melihat objek nyata dapat sebagai sebuah entitas - entitas yang memiliki relasi antara entitas yang satu ataupun yang lain [13]. Dengan ERD sendiri dapat membantu mengurangi kesalahan – kesalahan dalam melakukan perancangan *database* dari gambaran dunia nyata dan struktur *database* seperti redudansi data, hubungan – hubungan antara entitas dan lain sebagainya.

Pada ERD terdapat beberapa simbol dengan cara penulisan dapat dilihat pada Tabel 2.5.

Tabel 2. 5 Simbol ERD.

|  |  |  |
| --- | --- | --- |
| **Simbol** | **Nama** | **Keterangan** |
|  | Entitas / *Entitiy* | Merupakan data inti yang akan disimpan; bakal tabel pada basis data; biasanya mengacu pada benda yang terlibat dalam aplikasi yang akan dibuat. |
|  | Atribut | *Field* atau kolom data yang butuh disimpan dalam satu entitas. |
|  | Atribut kunci primer | *Field* atau kolom data yang butuh disimpan dalam suatu entitas sebagai kunci akses *record* yang diinginkan. |
|  | Relasi | Penghubung antar entitas; biasanya diawali dengan kata kerja. |
|  | Asosiasi / *Association* | Penghubung antara relasi dan entitas di mana di kedua ujungnya memiliki *multiplicity* kemungkinan jumlah pemakaian. |

### **MySQL**

SQL (*Structured Query Language*) adalah sebuah bahasa yang dipergunakan untuk mengakses data dalam basis data relasional. SQL juga dapat diartikan sebagai antar muka standar untuk sistem manajemen basis data relasional, termasuk sistem yang beroperasi pada komputer pribadi [13]. MySQL merupakan DBMS *open* *source* (kode programnya dapat di-*download* gratis) dan *freeware* (gratis dipakai), walau gratis MySQL tidak bisa dikatakan DBMS tidak bagus karena gratis, hal ini terbukti dengan adanya sebuah operator telpon selular yang menggunakan DBMS ini dan memiliki sistem yang andal dengan optomasi.

# **BAB III**

**METODE PENELITIAN**

## **Metodologi Penelitian**

Pada penelitian ini akan membahas tentang merancang dan membangun sistem yang berupa sistem pendukung keputusan penerima beasiswa bidikmisi. Sistem ini dapat membantu pihak terkait dalam menentukan mahasiswa yang layak mendapatkan beasiswa bidikmisi secara objektif di Universitas Mataram. Prosedur pelaksanaan penelitian dapat dilihat pada Gambar 3.1.

|  |
| --- |
|  |

Gambar 3. 1 Diagram alir penelitian.

## **Studi Literatur**

Studi literatur berfungsi untuk mengumpulkan dan mendalami teori teori yang berkaitan dengan perancangan dan pembuatan sistem pendukung keputusan penerima beasiswa bidikmisi. Adapun literatur yang digunakan berupa penelitian terdahulu dan dasar-dasar teori yang terkait.

## **Metode Pengumpulan Data**

Adapun metode pengumpulan data sebagai berikut:

1. Observasi

Penulis melakukan pengamatan langsung objek penelitian berupa data mahasiswa penerima beasiswa bidikmisi Universitas Mataram. Data yang didapat berupa data mentah dan data daftar penerima beasiswa bidikmisi Universitas Mataram tahun 2018.

1. Wawancara

Penulis melakukan pendekatan komunikasi yang berhubungan langsung dengan bapak Masri Rakasiwi di Kasubbag Kesejahteraan Mahasiswa Universitas Mataram selaku admin yang mengelola data mahasiswa peserta bidikmisi. Hasil wawancara yang didapat berupa alur kerja atau standar operasional proses seleksi calon penerima beasiswa bidikmisi Universitas Mataram.

## **Analisis Sistem**

Analisis sistem merupakan penjabaran dari suatu sistem informasi yang utuh ke dalam berbagai macam bagian komponennya dengan maksud agar kita dapat mengidentifikasi atau mengevaluasi berbagai macam masalah maupun hambatan yang akan timbul pada sistem sehingga nantinya dapat dilakukan penanggulangan, perbaikan atau juga pengembangan [14].

### **Analisis Sistem yang Berjalan**

Analisis sistem berjalan bertujuan untuk mengetahui alur kerja dari sistem yang telah ada. Dari alur kerja yang telah ada dan ditemukan masalah yang telah dijelaskan pada latar belakang, maka masalah tersebut dijadikan acuan untuk membuat perancangan sistem yang baru. Sistem yang digunakan dalam menentukan mahasiswa yang layak mendapat beasiswa bidikmisi masih bersifat manual dengan cara melakukan pengecekan data mahasiswa satu persatu pada *Microsoft excel* serta melakukan rapat atau diskusi yang bertempat di ruangan wakil rektor III Universitas Mataram.

### **Analisis Prosedur Sistem yang Berjalan**

Analisis prosedur yang berjalan bertujuan untuk mengetahui prosedur kerja yang terjadi di lapangan. Hasil dari analisis tersebut berupa kronologi kejadian secara langsung dari urutan kegiatan yang dilakukan khususnya berhubungan dengan mengolah data, sebagai berikut:

1. Mahasiswa pendaftar beasiswa bidikmisi mengumpulkan berkas pada saat daftar ulang dan telah mendaftarkan diri pada situs resmi Ristekdikti untuk melakukan pendaftaran *online*.
2. Setelah berkas masuk ke Kasubbag Kesejahteraan Mahasiswa Universitas Mataram, mahasiswa akan diseleksi dan dipilih secara manual sesuai dengan kriteria-krteria sebagai pertimbangan dalam proses seleksi.
3. Hasil seleksi Kasubbag Kesejahteraan Mahasiswa diteruskan ke wakil rektor III untuk diadakan rapat guna menentukan mahasiswa yang benar-benar layak mendapatkan beasiswa.
4. Kasubbag Kesejahteraan Mahasiswa membentuk tim untuk melakukan survey lapangan keseluruh mahasiswa yang terpilih dari hasil rapat yang diadakan diruangan wakil rektor III sebelumnya.
5. Dari hasil survey lapangan ditetapkan dan diumumkan mahasiswa yang memenuhi semua persyaratan beasiswa bidikmisi yang berlaku di Universitas Mataram.



Gambar 3. 2 *Flowmap* sistem yang sedang berjalan

### **Analisis Prosedur Sistem yang Akan Dibangun**

Analisa sistem yang akan di bangun bertujuan untuk mengetahui dan mendeskripsikan cara kerja dan prosedur kerja dari sistem.

Berikut alur kerja sistem yang akan dibuat :

1. Mahasiswa pendaftar beasiswa bidikmisi mengumpulkan berkas pada saat daftar ulang dan telah mendaftarkan diri pada situs resmi Ristekdikti untuk melakukan pendaftaran *online*.
2. Setelah berkas masuk ke Subbag kesejahteraan mahasiswa Universitas Mataram, admin memasukkan data mahasiswa ke dalam *database* sistem hingga data pendaftar terakhir dan data yang telah masuk akan dilakukan perhitungan dengan *Weighted Product* untuk menghasilkan *output* berupa perangkingan dari calon penerima beasiswa bidikmisi.
3. Dari hasil survei lapangan ditetapkan dan diumumkan mahasiswa yang memenuhi semua persyaratan beasiswa bidikmisi yang berlaku di Universitas Mataram.



Gambar 3. 3 Alur sistem yang akan dibangun

### **Analisis Kebutuhan Sistem**

Kebutuhan pembuatan sistem pendukung keputusan penerima baesiswa bidikmisi membutuhkan beberapa alat dan bahan yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*).

Perangkat keras yang dibutuhkan untuk membuat sistem berupa laptop yang akan digunakan sebagai media pembangunan sistem dan untuk uji coba sistem.

Perangkat lunak yang dibutuhkan untuk membuat sistem pendukung keputusan yaitu:

1. OS (*operating system)*, pada pembuatan aplikasi digunakan Windows 10 64-*bit* sebagai media untuk menjalankan *software* yang lain yang akan digunakan untuk membuat sistem.
2. *Visual Studio Code* merupakan *text editor* yang akan digunakan untuk membuat *code* program dari aplikasi yang akan dibuat pada tugas akhir ini.
3. Menggunakan PHP sebagai bahasa pemrogramannya
4. *CodeIgniter* digunakan sebagai *framework* pembuatan aplikasi berupa *website* yang akan mempermudah dalam pembangunan aplikasi.
5. XAMPP merupakan *software web server* apache yang didalamnya terdapat MySQL yang akan digunakan sebagai *server* dalam pembuatan aplikasi dan digunakan untuk mengelola *database* yang dibutuhkan aplikasi.
6. *Microsoft office* 2016 digunakan untuk penyusunan laporan.

Bahan yang dibutuhkan untuk membuat sistem berupa data mahasiswa penerima beasiswa bidikmisi tahun 2018*.*

## **Metode *Weighted Product***

*Weighted Product* (WP) adalah keputusan analisis multi-kriteria dan merupakan metode pengambilan keputusan multi kriteria. Seperti semua metode-metode lainnya, WP adalah himpunan dari alternatif keputusan yang dijelaskan dalam istilah beberapa kriteria. Berikut alur perhitungan dari metode *weighted product*.

|  |
| --- |
|  |

Gambar 3. 4 Alur perhitungan metode *weighted product.*

Pada studi kasus ini menggunakan 10 kriteria sesuai dengan data yang telah diberikan dan pada tabel 3.1 merupakan lima data mahasiswa yang dijadikan contoh dalam perhitungan.

Tabel 3. 1 Data mahasiswa.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nama mahasiswa | Kriteria-kriteria | | | | | | | | | |
| Penghasilan ayah (C1) | Penghasilan ibu (C2) | Jumlah tanggungan (C3) | Kepemilikan rumah (C4) | Luas tanah (C5) | Luas bangunan (C6) | Jarak pusat kota (C7) | Mandi cuci kakus (C8) | Sumber air (C9) | Sumber listrik (C10) |
| Fiqriyadi | Rp. 250.001 - Rp. 500.000 | < Rp. 250.000 | 2 | Sendiri | < 25 M2 | >200 M2 | 72 km | Kepemilikan sendiri diluar | Sumur | PLN |
| Sulistiawati | Rp.250.001 - Rp.500.000 | Rp.250.001 - Rp.500.000 | 5 | Sendiri | 100 - 200 M2 | 100 - 200 M2 | 25 km | Berbagi pakai | Sungai | PLN |
| Suryati | Rp. 250.001 - Rp. 500.000 | < Rp. 250.000 | 3 | Sendiri | 100 - 200 M2 | 25 - 50 M2 | 5 km | Kepemilikan sendiri didalam | Sumur | PLN |
| Aris Munandar | Rp. 500.001 –  Rp. 750.000 | Tidak Berpenghasilan | 5 | Sendiri | 100 - 200 M2 | 25 - 50 M2 | 25 km | Berbagi pakai | Sumur | PLN |
| Fikram Wadhaullah | Rp. 250.001 - Rp. 500.000 | < Rp. 250.000 | 3 | Sendiri | >200 M2 | >200 M2 | 25 km | Kepemilikan sendiri diluar | Sungai | PLN |

1. **Langkah-Langkah Perhitungan Metode *Weighted Product***
   * + 1. Menentukan kriteria dan bobot

Adapun Kriteria-kriteria dan bobot yang digunakan dapat dilihat pada tabel 3.2 dan pada tabel 3.3 digunakan sebagai acuan dalam pemberian bobot pada setiap kriteria. Bobot yang digunakan hasil dari diskusi dari Subbagian Kesejahteraan Mahasiswa dan wakil Rektor III.

Tabel 3. 2 Tabel Kriteria.

|  |  |  |
| --- | --- | --- |
| No | Kriteria | Bobot |
| 1 | Penghasilan Ayah | 5 |
| 2 | Penghasilan Ibu | 5 |
| 3 | Jumlah Tanggungan | 4 |
| 4 | Kepemilikan Rumah | 3 |
| 5 | Luas Tanah | 3 |
| 6 | Luas Bangunan | 3 |
| 7 | Jarak Pusat Kota | 3 |
| 8 | Mandi Cuci Kakus | 2 |
| 9 | Sumber Air | 1 |
| 10 | Sumber Listrik | 1 |

Tabel 3. 3 Tingkat Bobot Kriteria.

|  |  |
| --- | --- |
| Kriteria Penilaian | Nilai |
| Sangat rendah | 1 |
| Rendah | 2 |
| Sedang | 3 |
| Tinggi | 4 |
| Sangat tinggi | 5 |

Setelah pemberian bobot selesai pada masing-masing kriteria dilanjutkan dengan menentukan subkriteria dari masing-masing kriteria beserta nilainya, dapat dilihat pada tabel 3.4.

Tabel 3. 4 Tabel subkriteria.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Kriteria | Sub kriteria | Skor | Keterangan |
| 1 | Penghasilan Ayah | **>**Rp 2.000.000 | 1 | *Cost* |
| >Rp 1.500.000-2.000.000 | 2 |
| >Rp 1.000.000-1.500.000 | 3 |
| >Rp 500.000-1.000.000 | 4 |
| <=Rp 500.000 | 5 |
| 2 | Penghasilan Ibu | **>**Rp 2.000.000 | 1 | *Cost* |
| >Rp 1.500.000-2.000.000 | 2 |
| >Rp 1.000.000-1.500.000 | 3 |
| >Rp 500.000-1.000.000 | 4 |
| <=Rp 500.000 | 5 |
| 3 | Jumlah Tanggungan | 1-2 orang | 1 | *Benefit* |
| 3 orang | 2 |
| 4 orang | 3 |
| 5 orang | 4 |
| Lebih dari 5 orang | 5 |
| 4. | Kepemilikan rumah | Sendiri | 1 | *Benefit* |
| Menumpang | 5 |
| 5. | Luas tanah | > 200 m2 | 1 | *Cost* |
| 100 – 200 m2 | 2 |
| > 50 – 99 m2 | 3 |
| > 25 – 50 m2 | 4 |
| < 25m2 | 5 |
| 6 | Luas bangunan | > 200 m2 | 1 | *Cost* |
| 100 – 200 m2 | 2 |
| > 50 – 99 m2 | 3 |
| > 25 – 50 m2 | 4 |
| < 25m2 | 5 |
| 7 | Jarak pusat kota | <= 5 km | 1 | *Benefit* |
| > 5 - 10 km | 2 |
| > 10 – 15 km | 3 |
| > 15 – 20 km | 4 |
| > 20 km | 5 |
| 8 | Mandi cuci kakus | Ada dalam rumah | 1 | *Benefit* |
| Ada diluar | 3 |
| Umum/Berbagi Pakai | 5 |
| 9. | Sumber air | PDAM | 1 | *Benefit* |
| Sumur | 3 |
| Sungai/Mata air | 5 |
| 10 | Sumber listrik | PLN | 1 | *Benefit* |
| Genset/Mandiri | 3 |
| Tidak Ada | 5 |

Keterangan : *Benefit* = jika nilai terbesar adalah terbaik (bobot bernilai negatif).

*Cost*         = jika nilai terkecil adalah terbaik (bobot bernilai positif).

* + - 1. Menghitung perbaikan bobot

Proses normalisasi bobot kriteria (W), ΣW = 1 adalah:

(2-1)

Keterangan:

Wj: Bobot kriteria

wj : w indeks ke-j

ΣWj: Penjumlahan bobot kriteria

Proses Perhitungan normalisasi bobot dengan rumus diatas:

W1 = = = 0,167

W2 = = = 0,167

W3 = = = 0,133

W4 = = = 0,100

W5 = = = 0,100

W6 = = = 0,100

W7 = = = 0,100

W8 = = = 0,067

W9 = = = 0,033

W10 = = = 0,033

Setelah mendapatkan hasil perhitungan diatas dapat dilihat normalisasi bobot pada tabel

Tabel 3. 5 Tabel bobot dan kriteria setelah normalisasi.

|  |  |  |  |
| --- | --- | --- | --- |
| No | Kriteria | Bobot (Sebelum) | Bobot (Sesudah) |
| 1 | Penghasilan Ayah | 5 | 0,167 |
| 2 | Penghasilan Ibu | 5 | 0,167 |
| 3 | Jumlah Tanggugan | 4 | 0,133 |
| 4 | Kepemilikan Rumah | 3 | 0,100 |
| 5 | Luas Tanah | 3 | 0,100 |
| 6 | Luas Bangunan | 3 | 0,100 |
| 7 | Jarak Pusat Kota | 3 | 0,100 |
| 8 | Mandi Cuci Kakus | 2 | 0,067 |
| 9 | Sumber Air | 1 | 0,033 |
| 10 | Sumber Listrik | 1 | 0,033 |
| ΣW | | 30 | 1 |

* + - 1. Menghitung nilai skor

Sebelum menghitung nilai skor terlebih dahulu tabel 3.1 data mahasiswa dikonversi sesuai dengan nilai atau skor pada tabel 3.4. Hasil konversi dapat dilihat pada tabel

Tabel 3. 6 Tabel konversi nilai data mahasiswa.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Alternatif | Kriteria | | | | | | | | | |
| C1 | C2 | C3 | C4 | C5 | C6 | C7 | C8 | C9 | C10 |
| A1 | 5 | 5 | 1 | 1 | 5 | 1 | 5 | 3 | 3 | 1 |
| A2 | 5 | 5 | 4 | 1 | 2 | 2 | 5 | 5 | 5 | 1 |
| A3 | 5 | 5 | 2 | 1 | 4 | 2 | 1 | 1 | 3 | 1 |
| A4 | 4 | 5 | 4 | 1 | 4 | 2 | 5 | 5 | 3 | 1 |
| A5 | 5 | 5 | 2 | 1 | 1 | 1 | 5 | 3 | 5 | 1 |

Keterangan : A : alternatif (mahasiswa).

C : Kriteria.

Setelah melakukan konversi nilai barulah dihitung nilai skor dengan rumus sebagai berikut :

(2-2)

Keterangan:

Si: nilai dari setiap altenatif

Xij: menyatakan nilai/skor kriteria

Wj: bobot setiap kriteria

i: menyatakan alternatif

j: menyatakan kriteria

n: menyatakan banyaknya kriteria

Berikut proses perhitungan menggunakan rumus diatas :

S1=(50,167)(50,167)(1-0,133)(1-0,100)(50,100)(10,100)(5-0,100)(3-0,067)(3-0,033)(1-0,033)= 1,533712

S2=(50,167)(50,167)(4-0,133)(1-0,100)(20,100)(20,100)(5-0,100)(5-0,067)(5-0,033)(1-0,033)= 1,185206

S3=(50,167)(50,167)(2-0,133)(1-0,100)(40,100)(20,100)(1-0,100)(1-0,067)(3-0,033)(1-0,033)= 1,853459

S4=(40,167)(50,167)(4-0,133)(1-0,100)(40,100)(20,100)(5-0,100)(5-0,067)(3-0,033)(1-0,033)= 1,244612

S5=(50,167)(50,167)(2-0,133)(1-0,100)(10,100)(10,100)(5-0,100)(3-0,067)(3-0,033)(1-0,033)= 1,190721

* + - 1. Menghitung nilai vektor

Setelah menghitung nilai skor selanjutnya menghitung nilai vektor, dimana hasilnya digunakan sebagai hasil akhir. Berikut rumus yang digunkan untuk menghitung nilai vektor :

(2-3)

Keteragan:

V: menyatakan alternatif yang dianalogikan sebagai vektor V

Xij: menyatakan nilai/skor kriteria

Wj: bobot setiap kriteria

i: menyatakan alternatif

j: menyatakan kriteria

n: menyatakan banyaknya kriteria

Proses perhitungan nilai vector sebagai berikut :

V 1 = = = 0,218861

V 2 = = = 0.169129

V 3 = = = 0.264489

V 4 = = = 0.177606

V 5 = = = 0.169916

Dari perhitungan yang telah dilakukan didapat bahwa A3 atau mahasiswa nomor 3 memliki nilai V tertingi sehingga A3 sangat direkomendasikan mendapat beasiswa untuk lebih jelas dapat dilihat pada tabel 3.7.

Tabel 3. 7 Hasil perangkingan WP.

|  |  |  |
| --- | --- | --- |
| Rangking WP | Alternatif | Nilai V |
| 1 | A3 | 0.264489 |
| 2 | A1 | 0.218861 |
| 3 | A4 | 0.177606 |
| 4 | A5 | 0.169916 |
| 5 | A2 | 0.169129 |

## **Perancangan Sistem**

Setelah melakukan analisis terhadap sistem kemudian dirancang menggunakan UML. Hal yang pertama kali dilakukan adalah merancang *Usecase* *diagram* untuk interaksi tipikal antara para pengguna sistem dengan sistem itu sendiri. Kemudian merancang *class* *diagram* untuk menjelaskan struktur dari program yang akan dibuat, biasanya *class* *diagram* dirancang pada sistem yang menggunakan konsep OOP. Setelah itu buat perancangan untuk *sequence diagram* yang harus sesuai dengan *class diagram* yang ada pada sistem. Selanjutnya merancang *activity diagram* pada sebuah sistem yang akan dirancang agar alur dari kerja suatu sistem dari awal hingga akhir dapat dimengerti oleh perancang sistem*.* Perancangan dari segi *database* menggunakan penggambaran ERD juga perlu, untuk menggambarkan data-data yang ada dalam sistem.

### ***Usecase Diagram***

Pada rancangan *usecase* yang dibuat untuk sistem hanya terdapat dua aktor yaitu pegawai Subbagian Kesejahteraan Mahasiswa, wakil rektor IIIdan mahasiswa. Pegawai Subbagian Kesejahteraan Mahasiswa dan wakil rektor IIIdapat melakukan pengolahan data yaitu hapus data administrator, data mahasiswa, nilai botot, melakukan proses perhitungan, melihat hasil perhitungan dan mencetak laporan hasil seleksi. Sedangkan mahasiswa hanya dapat melihat hasil seleksi, bentuk *usecase* dapat dilihat pada Gambar 3.5.

|  |
| --- |
|  |

Gambar 3. 5 *Usecase diagram*

### ***Class Diagram***

Berikut rancangan class diagram dari sistem yang akan dibangun :

|  |
| --- |
|  |

Gambar 3. 6 *Class diagram*

Sistem yang dibangun menggunakan metode analisa dan perancangan berorientasi objek dan menggunakan pola arsitektur *Model* *View* *Controller* (MVC), sehingga dilakukan perancangan kelas – kelas yang ada pada sistem yang terdiri dari kelas *Controller* dan *Model.* Pada kelas *Controller* terdapat kelas kelola\_kriteria, kelola\_mahasiswa, kelola\_admin, perhitungan, *Dashboard*, spkbidimisi, Login dan profil. Sedangkan pada *Model* terdapat kelas Login\_m, admin\_m, mahasiswa\_m, dan kritria\_m.

### ***Sequence Diagram***

Berikut merupakan *sequence digram* dari sistem pendukung keputusan untuk menunjang keputusan penerima beasiswa:

* + - 1. *Login*

|  |
| --- |
|  |

Gambar 3. 7 *Sequence diagram login*.

Gambar 3.7 merupakan *sequence diagram* dari proses *login*. Proses ini diawali dari *form* *login*, pegawai atau WR III memasukkan *username* dan *password* lalu memanggil fungsi proses() pada *controller* Login. Kemudian memanggil fungsi cek($post) pada *model* Login dan mengecek *username* dan *password* pada *database*. Dari *controller* mengrimkan nilai *login* berhasil atau gagal apabila berhasil menampilkan halaman *dashboard,* jika gagal kembali ke *form* *login* .

* + - 1. Tambah Data Mahasiswa

|  |
| --- |
|  |

Gambar 3. 8 *Sequence diagram* tambah data mahasiswa.

Gambar 3.8 merupakan *sequence diagram* dari proses tambah data mahasiswa. Proses ini diawali dari *form input* data yang telah disediakan, pegawai memasukkan data secara lengkap lalu memanggil fungsi tambah() pada *controller* kelola\_mahasiswa dan melakukan validasi dengan *libarary form\_validation*. Setelah data divalidasi dan sudah sesuai function tambah() pada *model* mahasiswa\_m melakukan penyimpanan ke*database* dan data berhasil disimpan dan langsung dialihkan ke halaman data mahasiswa, sedangkan jika tidak berhasil akan kembali diarahkan ke *form input* data.

* + - 1. Hapus Data Mahasiswa

|  |
| --- |
|  |

Gambar 3. 9 *Sequence diagram* hapus data mahasiswa.

Gambar 3.9 merupakan *sequence diagram* dari proses hapus data mahasiswa. Proses ini diawali darimelihatdata mahasiswa, pegawai mengklik tombol hapus lalu memanggil fungsi hapus() pada *controller* kelola\_mahasiswa. Kemudian memanggil fungsi hapus($id) pada *model* mahasiswa\_m dan menghapus data di *database* berdasarkan *id*.

* + - 1. Edit Data Mahasiswa

|  |
| --- |
|  |

Gambar 3. 10 *Sequence diagram* edit data mahasiswa.

Gambar 3.10 merupakan *sequence diagram* dari proses edit data mahasiswa. Proses ini diawali darimelihatdata mahasiswa, pegawai atau mengklik tombol edit lalu memanggil fungsi edit($id) pada *controller* kelola\_mahasiswa. Kemudian memanggil fungsi edit($post) pada *model* mahasiswa\_m dan memperbarui data di *database* berdasarkan *id*.

* + - 1. Tambah Admin

|  |
| --- |
|  |

Gambar 3. 11 *Sequence diagram* tambah data admin.

Gambar 3.11 merupakan *sequence diagram* dari proses tambah data admin. Proses ini diawali dari *form input* data yang telah disediakan, pegawai memasukkan data secara lengkap lalu memanggil fungsi tambah() pada *controller* kelola\_admin dan melakukan validasi dengan *libarary form\_validation*. Setelah data divalidasi dan sudah sesuai *function* tambah() pada *model* admin\_m melakukan penyimpanan ke*database* dan data berhasil disimpan dan langsung dialihkan ke halaman data mahasiswa, sedangkan jika tidak berhasil akan kembali diarahkan ke *form input* data.

* + - 1. Hapus Admin

|  |
| --- |
|  |

Gambar 3. 12 *Sequence diagram* hapus data admin.

Gambar 3.12 merupakan *sequence diagram* dari proses hapus data admin. Proses ini diawali darimelihatdata admin, pegawai mengklik tombol hapus lalu memanggil fungsi hapus() pada *controller* kelola\_admin. Kemudian memanggil fungsi hapus($id) pada *model* admin\_m dan menghapus data di *database* berdasarkan *id*.

* + - 1. Edit Profil

|  |
| --- |
|  |

Gambar 3. 13 *Sequence diagram* edit data admin.

Gambar 3.13 merupakan *sequence diagram* dari proses edit data admin. Proses ini diawali dengan mengklik tombol profil, lalu pegawai atau WR III mengklik tombol edit lalu memanggil fungsi edit($id) pada *controller* kelola\_mahasiswa. Kemudian memanggil fungsi edit($post) pada *model* mahasiswa\_m dan memperbarui data di *database* berdasarkan *id*.

* + - 1. Edit Data Bobot

|  |
| --- |
|  |

Gambar 3. 14 *Sequence diagram* edit data bobot.

Gambar 3.14 merupakan *sequence diagram* dari proses edit data bobot. Proses ini diawali darimelihatdata kriteria, pegawai atau WR III mengganti nilai yang terdapat pada *text input* lalu mengklik tombol perbarui maka otomatis data akan berubah.

### ***Activity Diagram***

Berikut merupakan *activity diagram* dari sistem pendukung keputusan sebagai penunjang keputusan mahasiswa penerima beasiswa bidikmisi:

1. Proses *login*



Gambar 3. 15 *Activity diagram login.*

Pada gambar 3.15 merupakan proses *login* yang dimulai dari sistem menampilkan *form login*. Kemudian admin memasukkan *username* dan *password* setelah itu, memverifikasi data yang telah dimasukkan apabila data ada dan sudah valid maka sistem menampilkan halaman *dashboard* apabila data tidak ditemukan atau data yang diisi salah maka *user* harus memasukkan ulang data *username* dan *password* yang sesuai.

1. Proses tambah data mahasiswa



Gambar 3. 16 *Activity diagram* tambah data mahasiswa.

Pada gambar diatas merupakan proses tambah data mahasiswa. Proses ini berawal dari sistem menampilkan halaman tambah data mahasiswa admin memasukkan data mahasiswa sesuai yang ada pada *form*. Data divalidasi, apabila data sudah lengkap dan tidak sama dengan data yang sudah ada maka data disimpan ke dalam *database* jika data tidak lengkap atau data sama dengan sebelumnya *admin* harus mengecek kembali data yang disi pada *form*.

1. Proses *edit* data mahasiswa



Gambar 3. 17 *Activity diagram edit* data mahasiswa.

Pada gambar diatas merupakan proses *edit* data mahasiswa. Proses ini berawal dari sistem menampilkan *form edit* data mahasiswa kemudian admin mengisi data yang ingin di-*edit*.Data yang telah diperbarui akan disimpan ke *database*.

1. Proses hapus data mahasiswa



Gambar 3. 18 *Activity diagram* hapus data mahasiswa.

Pada gambar 3.18 merupakan proses hapus data mahasiswa. Proses hapus data mahasiswa berawal dari sistem menampilkan halaman kelola data mahasiswa kemudian *admin* mengklik ikon hapus data mahasiswa dan akan di hapus dari *database*.

1. Proses tambah admin



Gambar 3. 19 *Activity diagram* tambah admin.

Pada gambar diatas merupakan proses tambah admin. Proses ini berawal dari sistem menampilkan halaman *form inputan* data *admin*.. Data divalidasi, apabila data sudah lengkap dan tidak sama dengan data yang sudah ada maka data disimpan ke dalam *database* jika data tidak lengkap atau data sama dengan sebelumnya *admin* harus mengecek kembali data yang disi pada *form*.

1. Proses Edit Admin



Gambar 3. 20 *Activity diagram edit* data admin.

Pada gambar 3.20 merupakan proses *edit* data admin. Proses ini berawal dari membuka profil lalu menekan tombol edit, maka sistem menampilkan *form edit* data admin kemudian admin mengisi data yang ingin di-*edit*.Data yang telah diperbarui akan disimpan ke *database*.

1. Proses Hapus Admin



Gambar 3. 21 *Activity diagram* hapus data mahasiswa.

Pada gambar 3.21 merupakan proses hapus dataadmin. Proses hapus data mahasiswa berawal dari sistem menampilkan halaman kelola data admin kemudian *admin* mengklik ikon hapus data admin dan akan di hapus dari *database*.

1. Proses Edit Bobot



Gambar 3. 22 *Activity diagram* edit data bobot.

Pada gambar diatas merupakan proses edit data bobot. Proses edit data bobot berawal dari sistem menampilkan halaman kelola kriteria kemudian *admin* mengganti data di input text dan menekan tombol hitung maka data akan diperbarui dan mnghitung bobot ternomalisasi.

### ***Entinity Relationship Diagram* (ERD)**

Berikut merupakan rancangan *Entinity Relationship Diagram* (ERD) pada sistem yang akan dibangun:

|  |
| --- |
|  |

Gambar 3. 23 ER *diagram.*

Gambar 3.23 merupakan *entinity relationship diagram* yang terdiri dari tiga entitas yaitu Mahasiswa, Kriteria, dan Admin, serta terdapat tabel relasi yaitu menghitung. Detail atribut dari entitas telah disebutkan dapat dilihat pada Tabel 3.8 sampai dengan Tabel 3.10.

Tabel 3. 8 Tabel Mahasiswa

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| mhs\_id | Int (11) | No | None |
| ­mhs\_nim | Varchar (9) | No | None |
| mhs\_nama | Varchar (50) | No | None |
| mhs\_jrs | Varchar (50) | No | None |
| C1 | Int (1) | No | None |
| C2 | Int (1) | No | None |
| C3 | Int (1) | No | None |
| C4 | Int (1) | No | None |
| C5 | Int (1) | No | None |
| C6 | Int (1) | No | None |
| C7 | Int (1) | No | None |
| C8 | Int (1) | No | None |
| C9 | Int (1) | No | None |
| C10 | Int (1) | No | None |

Tabel 3. 9 Tabel admin

|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| Id | int (2) | No | None |
| Username | Varchar (20) | No | None |
| Nama\_lengkap | Varchar(50) | No | None |
| Password | Varchar(20) | No | None |
| Email | Varchar(30) | No | None |
| No.Hp | Varchar(13) | No | None |
| level | int (1) | No | None |
| Akun\_dibuat | timestamp | No | None |

Tabel 3. 10 Tabel Kriteria

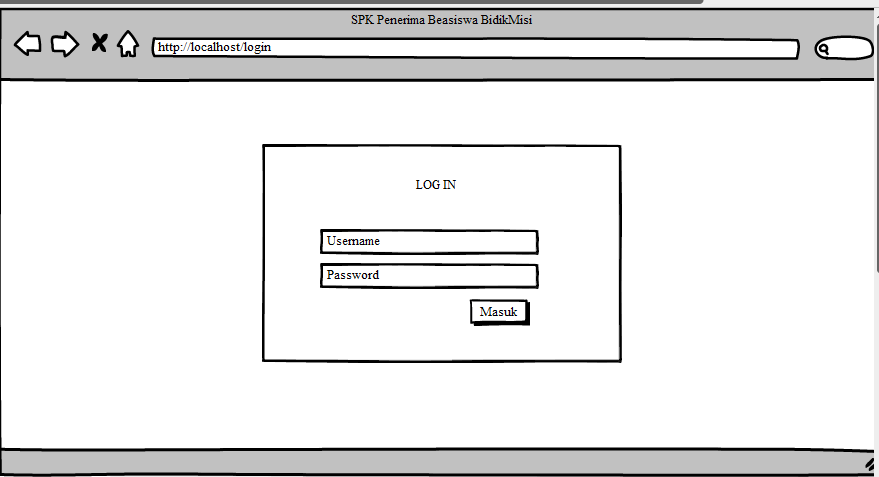
|  |  |  |  |
| --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Default** |
| krit\_id | Varchar (4) | No | None |
| krit\_nama | Varchar(20) | No | None |
| krit\_bobot | int | No | None |
| Krit\_bobot\_s | double | No | None |
| kategori | Varchar(7) | No | None |

### 

### **Desain Tampilan Sistem**

Berikut merupakan rancangan *interface* dari sistem yang akan dibangun:

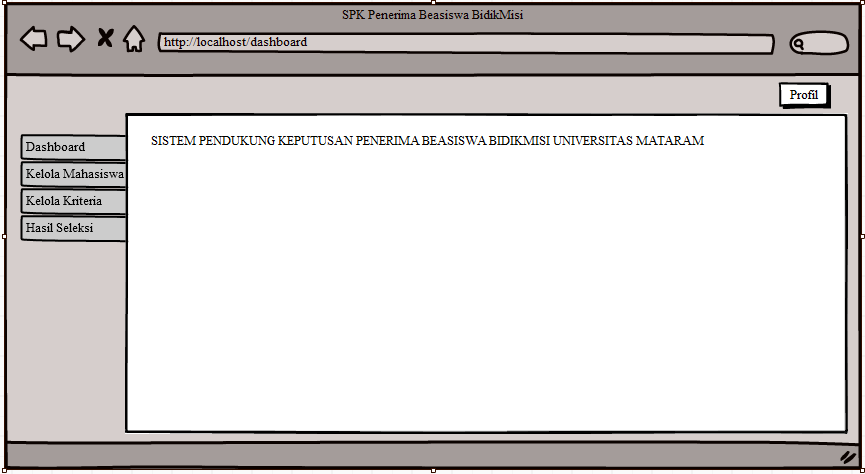
1. Halaman *login*



Gambar 3. 24 Desain halaman *login*.

Gambar 3.24 merupakan desain dari halaman *login*. Halaman *login* merupakan halaman yang digunakan untuk masuk ke dalam sistem.

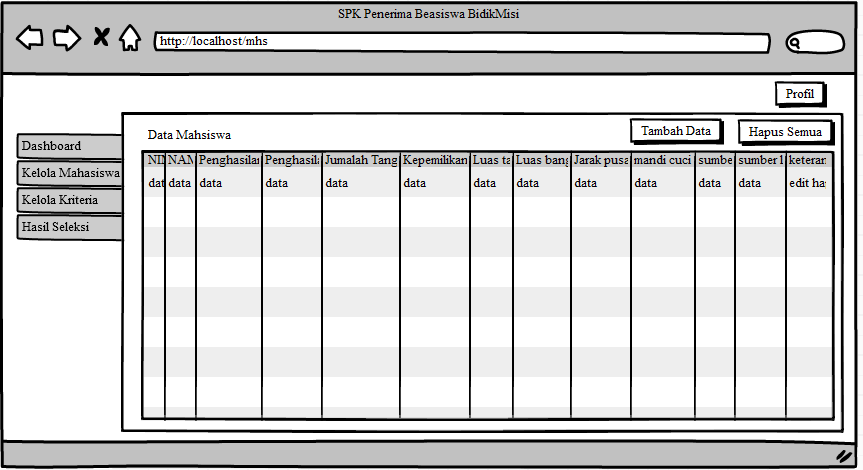
1. Halaman *dashboard*



Gambar 3. 25 Desain halaman *dashboard*.

Gambar 3.26 merupakan desain dari halaman awal sistem.Halaman ini merupakan halaman yang akan muncul ketika sudah masuk ke dalam sistem. Terdapat menu pilih dashboard, kelola data, kelola kriteria dan hasil seleksi.

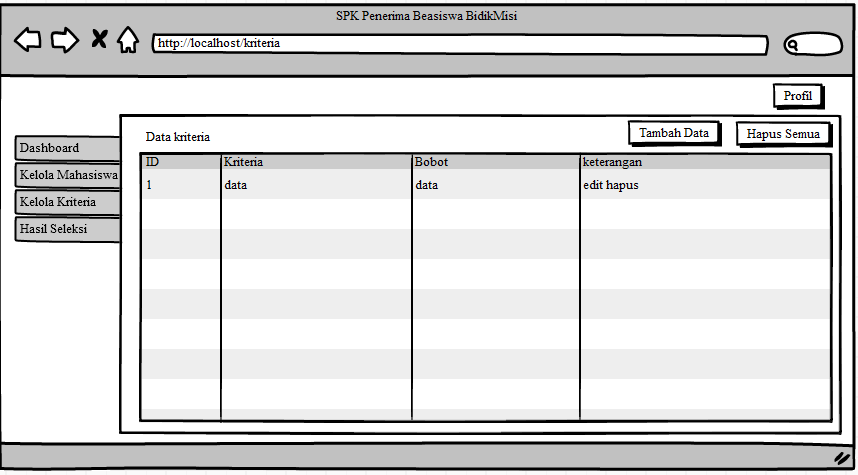
1. Halaman Kelola Data



Gambar 3. 26Desain halaman kelola mahasiswa.

Gambar 3.27 merupakan desain dari halaman Kelola mahasiswa.Halaman ini akan menampilkan data mahasiswa yang ada di*database* yang belum dilakukan perhitungan, data ditampilkan dalam bentuk tabel yang bisa dihapus dan diedit.

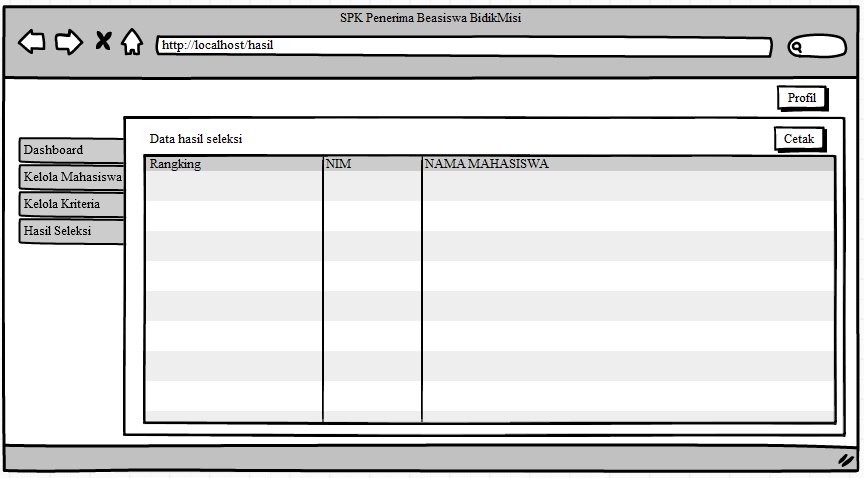
1. Halaman Kelola Kriteria



Gambar 3. 27Desain halaman kelola kriteia.

Gambar 3.28 merupakan desain dari halaman Kelola Krtiteria.Halaman ini akan menampilkan data Kriteria yang ada di*database*, data ditampilkan dalam bentuk tabel yang bisa dihapus dan diedit.

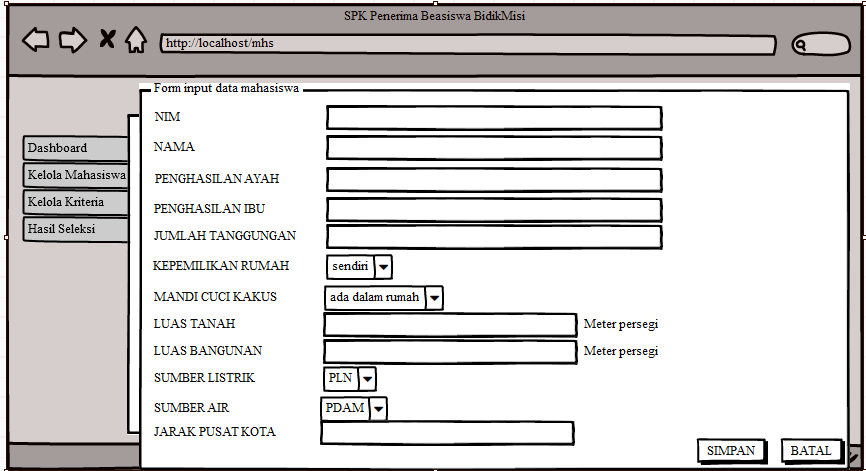
1. Halaman Hasil seleksi



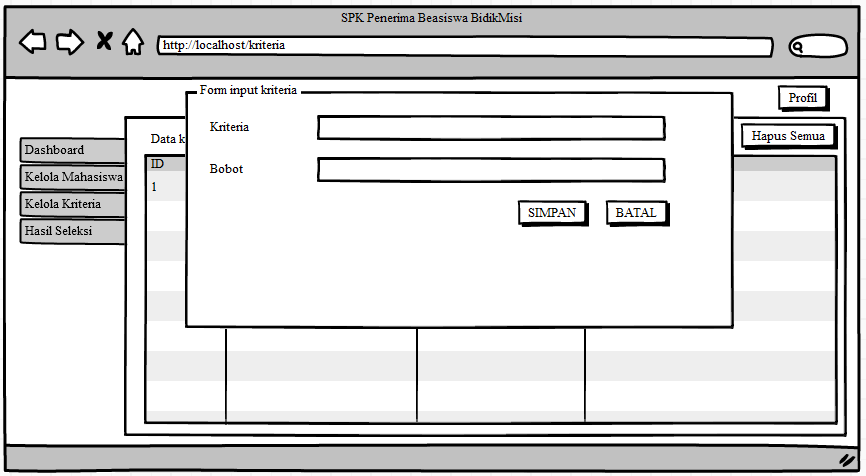
Gambar 3. 28Desain halaman hasil seleksi.

Gambar 3.29 merupakan desain dari halaman Hasil perhitungan dengan metode yang digunakan.Pada halaman ini ditampilkan perangkingan mahasiswa dari perhitungan yang telah dilakukan dan ada fitur cetak.

1. *Form Input* data mahasiswa dan data kriteria



Gambar 3. 29 *Form input* data mahasiswa.



Gambar 3. 30 *Form input* data kriteria.

## **Pembuatan Sistem**

Pada sistem pendukung keputusan yang akan dibangun menggunakan bahasa pemrograman PHP dan MySQL sebagai *database* serta menggunakan *framework codeigniter.* Sistem yang akan dibangun memiliki beberapa fitur seperti kelola data mahasiswa, kelola data kriteria dan menghitung menggunakan metode *weighted product* untuk menghasilkan *output* berupa perangkingan dengan prioritas tertinggi.

## **Pengujian Sistem**

Untuk melakukan pengujian pada sistem yang akan dibangun menggunakan dua metode pengujian sebagai berikut:

### **Metode *White Box Testing***

*White Box Testing* adalah salah satu cara untuk menguji suatu aplikasi atau *software* dengan cara melihat modul untuk dapat meneliti dan menganalisa kode dari program yang di buat ada yang salah atau tidak [15]. Pada sistem yang akan dibuat pengujian *white box* dilakukan untuk mengecek hasil perhitungan dari fungsi metode *weighted product*.

### **Metode *Black Box Testing***

Black Box Testing atau yang sering dikenal dengan sebutan pengujian fungsional merupakan metode pengujian Perangkat Lunak yang digunakan untuk menguji perangkat lunak tanpa mengetahui struktur internal kode atau program [16]. Pada sistem yang akan dibuat pengujian *black box* dilakukan untuk mengecek fitur-fitur yang berhubungan dengan pengolahan data.

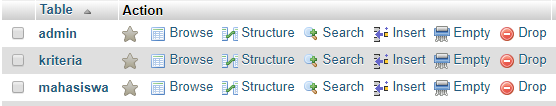
# **BAB IV**

**HASIL DAN PEMBAHASAN**

## **Implementasi Sistem**

Pada bab ini akan dibahas implementasi dan pengujian dari “Implementasi *Metode Weighted* *Product* Pada Sistem Pendukung Keputusan Penerima Beasiswa Bidikmisi Universitas Mataram” atas perancangan yang telah dilakukan pada bab sebelumnya. Implementasi yang dibahas yaitu implementasi *database*, implementasi perancangan *class diagram*, implementasi *interface* dan Hasil perngkingan dengan sistem. Sedangkan pada tahap pengujian, dilakukan 2 jenis pengujian yaitu pengujian *black box testing* dan *white box testing*.

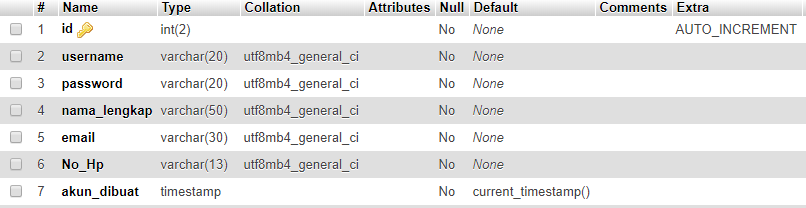
### **Implementasi *database***



Gambar 4. 1 Implementasi *database*

Pada sistem pendukung keputusan menggunakan MySQL sebagai penyimpanan data. *Database* yang dibuat diberi nama dengan spk\_bm terdapat 3 tabel antara lain tabel admin, mahasiswa dan kriteria. Dari 3 tabel tersebut terdapat 2 tabel yang saling berelasi untuk digunakan dalam melakukan perhitungan dengan metode *weighted product*. Untuk lebih jelasnya terakait dengan masing-masing tabel, akan ditampilkan sebagai berikut:

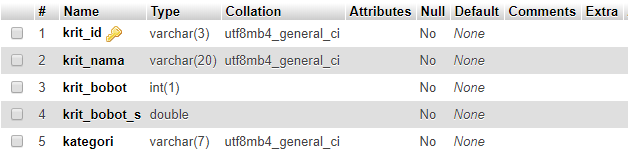
Tabel Admin



Gambar 4. 2 Tabel admin

Pada Gambar 4.2 merupakan struktur dari tabel admin yang terdiri dari : *id* sebagai *primary key* dengan tipe data *integer*(2), *username* dengan tipe data *varchar*(20) untuk menyimpan *username* dari *user* yang digunakan dalam validasi saat melakukan *login*, *password* dengan tipe data *varchar*(20) untuk menyimpan *password* dari *user* digunakan sebagai validasi saat *login*, nama\_lengkap dengan tipe data *varchar*(50) untuk menyimpan nama *user*, email dengan tipe data *varchar*(30) untuk menyimpan email *user* dengan maksimal karakter, No\_Hp dengan tipe data *varchar*(13) untuk meyimpan nomor telepon *user* dan akun\_dibuat dengan tipe data *timestamp* menyimpan waktu *user* ditambahkan.

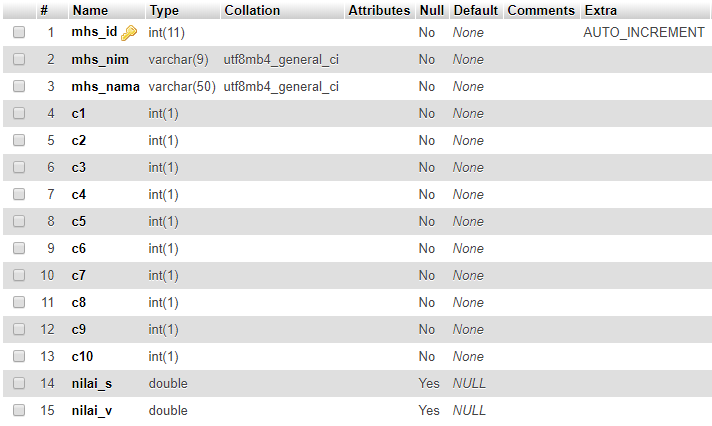
Tabel Kriteria



Gambar 4. 3 Tabel kriteria

Pada Gambar 4.3 merupakan struktur dari tabel kriteria yang terdiri dari: krit\_id sebagai *primary key* dengan tipe data *varchar*(3), krit\_nama dengan tipe data *varchar*(20) untuk menyimpan nama kriteria dengan maksimal 20 karakter, krit\_bobot dengan tipe data *int*(1) untuk menyimpan bobot awal dari keriteria, krit\_bobot\_s dengan tipe data *double* untuk menyimpan bobot setelah normalisasi dan kategori dengan tipe data *varchar*(7) untuk menyimpan kategori dari setiap kriteria, apakah kriteria itu *cost* atau *benefit*.

Tabel Mahasiswa



Gambar 4. 4 Tabel mahasiswa

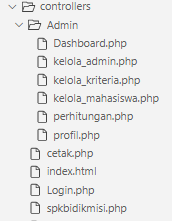
Pada Gambar 4.4 merupakan struktur dari tabel mahasiswa yang terdiri dari : mhs\_id sebagai *primary key* dengan tipe data *int*(11), mhs\_nim dengan tipe data *varchar*(9) untuk menyimpan NIM mahasiswa, mhs\_nama dengan tipe data *varchar*(50) untuk menyimpan nama mahasiswa dengan maksimal 50 karakter, c1 sampai c10 dengan tipe data *int*(1) untuk menyimpan skor mahasiswa di masing-masing kriteria, nilai\_s dengan tipe data *double* untuk menyimpan hasil perhitungan vektor s dan nilai\_v dengan tipe data *double* untuk menyimpan nilai vektor v, dimana vektor v merupakan nilai akhir yang menjadi rujukan dalam perangkingan.

### **Implementasi *class***

Pada bagian ini akan dibahas *class-class* yang digunakan dalam pembuatan sistem. Pembuatan sistem menggunakan *framework codeigniter*, dimana struktur *class* dalam *framework* *codeIgniter* yakni menggunakan struktur MVC (Model *View Controller*). Berikut akan diuraikan sebagian class yang dianggap penting dalam pembuatan sistem :

1. *Class controller*

Pada *class* ini berfungsi untuk menghubungkan antara *class* model dan *class view* dan yang bertugas dalam untuk mengarahkan halaman-halaman yang dituju. Pada Gambar 4.5 ditunjukkan *class-class controller* yang digunakan dalam pembuatan sistem.



Gambar 4. 5 *Class controller* pada sistem

Pada gambar diatas dapat dilihat terdapat sembilan *controller* yang digunakan dalam pembuatan sistem. Diantara sembilan *controller*, ada beberapa *controller* yang akan dijelaskan secara singkat. Berikut akan dijelaskan secara singkat beberapa *controller* antara lain :

1. *Script* proses *login* sistem

|  |
| --- |
|  |

Gambar 4. 6 *Script* proses *login* sistem

*Script* diatas merupakan *function* proses() yang terdapat pada *class controller login*. *Script* ini berfungsi untuk memvalidasi *user* sebelum dapat memasuki sistem. Pertama akan me*load* model Login\_m, lalu *function* cek pada Login\_m akan melakukan pengecekan keseluruh data, jika *username* dan *password* yang dimasukkan ada maka *login* berhasil dan diarahkan kehalaman *dashboard* jika tidak kembali ke halaman *login*.

1. *Script* menghitung bobot normalisasi

|  |
| --- |
|  |

Gambar 4. 7 *Script* hitung bobot normalisasi

Pada Gambar 4.7 merupakan *script function* hitungBobot() yang terdapat pada *controller* kelola\_krtiteria. Script ini berfungsi untuk menghitung bobot yang akan dinormalisasi. Pertama akan me*load* model ktiteria\_m danm menjumahkan semua bobot yang di*iput*kan. Jumlah bobot akan jadi pembagi untuk tiap-tiap nilai bobot menggunakan perulangan.

1. *Script* menghitung nilai vektor s dan vektor v

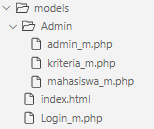
|  |
| --- |
|  |

Gambar 4. 8 *Script* metode WP

Pada Gambar 4.8 merupakan *function* process pada *controller* perhitungan *function* ini juga merupakan perhitungan dengan metode *weighted product* untuk mencari nilai dari vektor s. Pertama mengambil nilai bobot yang telah dinormalisasi sesauai dengan id kriteria, lalu masing-masing kriteria akan dicek lagi kategorinya dengan kondisi if. Jika kondisinya *cost* nilai bobot normalisasi bernilai positif sedangkan jika kondisinya *benefit* nilai bobot normalisasi bernilai negatif. Kemudian dilakukan pemangkatan skor dengan bobot normalisasi untuk setiap kriteria. Setelah hasil pemangkatan tiap-tiap kriteria didapat lalu mengalikan semua hasil perpangkatan untuk mengahasilkan nilai vektor s. Kemudian menghitung nilai vektor v dengan cara membagi tiap-tiap nilai vektor s dibagi dengan hasil total penjumlahan keseluruhan nilai vektor s.

1. *Class* Model

Pada *class* ini berfungsi untuk mengolah data seperti menyimpan data, mengedit data , mengambil data dan menghapus data. Pada Gambar 4.9 ditunjukkan *class-class* model yang digunakan dalam pembuatan sistem.



Gambar 4. 9 *Class* model pada sistem

Pada gambar diatas terdapat empat model yang digunakan dalam pembuatan model. Model-model tersebut sebagian memiliki fungsi yang sama seperti model admin\_m, kriteria\_m dan mahasiswa\_m berfungsi untuk mengolah data seperti hapus, tambah, mengambil dan edit data, sedangkan login\_m yaitu untuk mengecek *username* dan *password* dari yang diinputkan *user* pada saat *login*. Berikut *script* dari model Login\_m dan admin\_m :

*Script* validasi *user* yang *login*

|  |
| --- |
|  |

Gambar 4. 10 *Script* validasi *user login*

Pada Gambar 4.10 merupakan *function* cek dengan *parameter post* yang terdapat pada *class* model Login\_m yang berfungsi untuk mengecek *username* dan *password* yang di*input user* pada tabel admin ketika melakukan  *login*.

* 1. *Script* tambah data admin

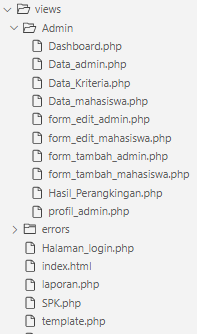
|  |
| --- |
|  |

Gambar 4. 11 *Script* tambah admin

Pada Gambar 4.11 merupakan *function* tambah dengan *parameter post* pada model admin\_m yang berfungsi untuk menambah data admin. Begitu pula untuk menambah data mahasiswa konsepnya sama yaitu dengan membaca *inputan* yang dimasukkan oleh *user* makanya *inputan* akan di*insert* di*database*.

1. *Class view*

Pada *class* ini berfungsi untuk menampilkan *interface* pada sistem untuk melakukan interaksi dengan *user*. Pada Gambar 4.12 ditunjukkan *class-class view* yang digunakan dalam pembuatan sistem.



Gambar 4. 12 *Class view* pada sistem

Pada Gambar 4.12 merupakan *class-class* yang ada *class view* yang digunakan untuk menampilkan *userinterface* pada sistem untuk memudahkan *user* dalam menggunakan sistem. Berikut merupakan potongan *script* untuk menampilkan hasil perhitungan.

|  |
| --- |
|  |

Gambar 4. 13 Potongan *script* menampilkan hasil perangkingan

### **Implementasi *interface* sistem**

Pada tahap ini akan dijelaskan hasil implementasi dari rancangan tampilan yang telah dibuat sebelumnya. Implementasi *class* dan tampilan dibagi menjadi 2 bagian yaitu implementasi tampilan untuk admin dan implementasi tampilan untuk mahasiswa.

#### **Interface sistem pada mahasiswa**

|  |
| --- |
|  |

Gambar 4. 14 Beranda untuk mahasiswa

|  |
| --- |
|  |

Gambar 4. 15 Tampilan hasil seleksi

Pada Gambar 4.14 merupakan tampilan beranda untuk mahasiswa dan diberanda terdapat tombol *login* untuk masuk kehalaman *dashboard*. Sedangkan pada Gambar 4.15 merupakan tabel hasil seleksi yang sudah dilakukan oleh metode *weighted product*.

#### **Interface sistem pada admin**

Pada tampilan untuk admin terdapat tampilan *login*, *dashboard*, kelola data yang terdri dari admin, kriteria, mahasiswa dan hasil perangkingan. Untuk lebih jelasnya dapat dilihat pada *sitemap* berikut :

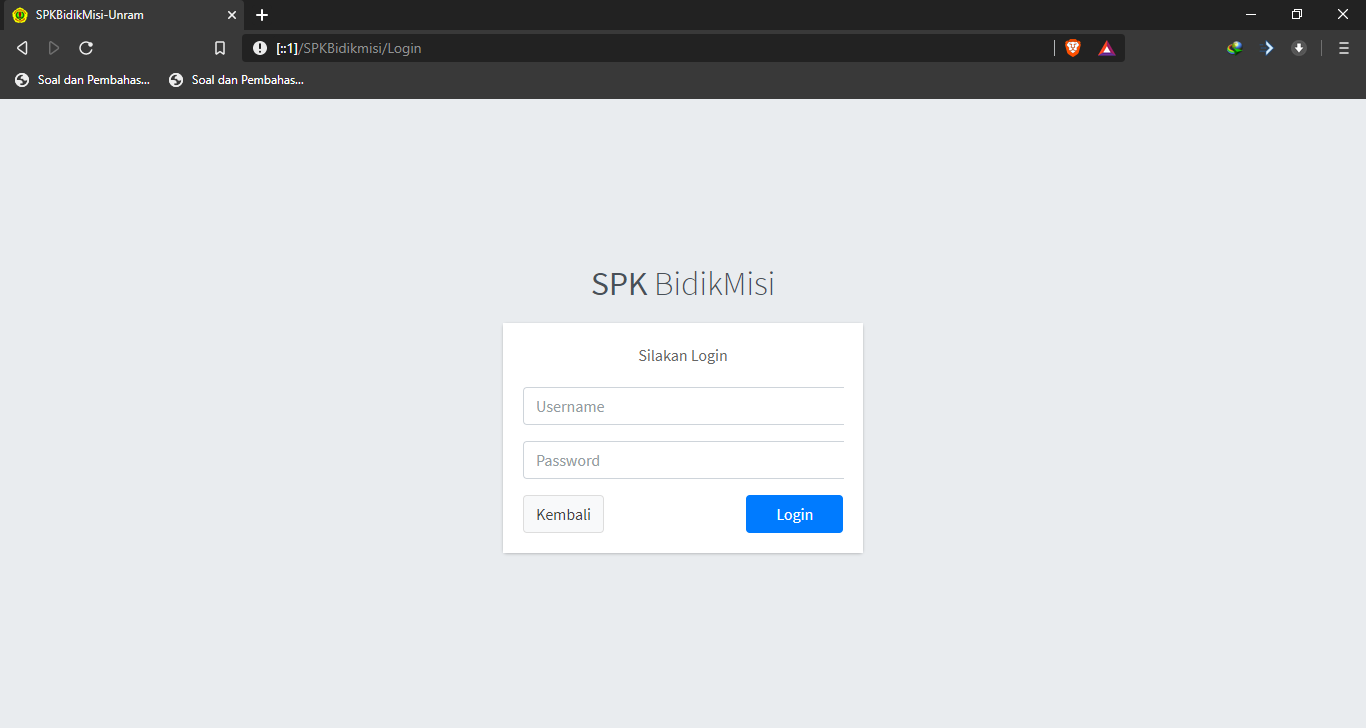
|  |
| --- |
|  |

Gambar 4. 16 *Sitemap* tampilan untuk admin

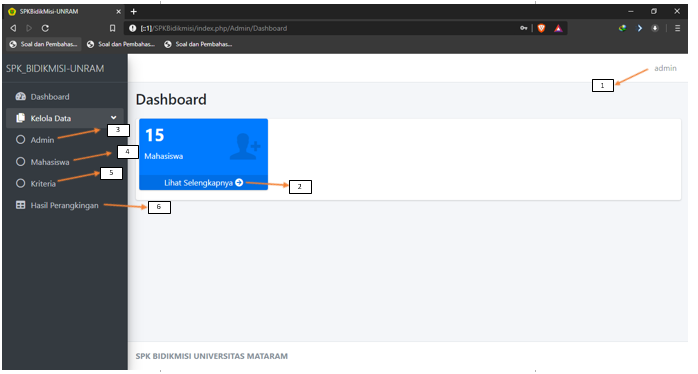
Pada *class login* terdapat *form login* dapat untuk masuk ke sistem dapat dilihat pada Gambar 4.16. Admin harus memasukkan *username* dan *password* yang sesuai dengan yang ada di *database* agar dapat masuk ke halaman *dashoard* dari sistem dapat dilihat pada Gambar 4.17. Pada Gambar 4.18(1) merupakan nama *user* yang sedang *login* dan terdapat menu profil dan *logout* bagi *user* yang sedang *login,*dapat dilihat pada Gambar 4.19. Apabila yang dipilih profil maka akan diarahkan ke halaman profil admin dapat dillihat pada Gambar 4.20 dan dihalaman ini terdapat tombol edit untuk mengedit data admin lalu diarahkan ke halaman edit data admin dapat dilihat pada Gambar 4.21. Pada Gambar 4.18(2) menampilkan jumlah data mahasiswa.

Pada Gambar 4.18(3) merupakan menu admin apabila di klik akan menampilkan data admin yang ada di *database* dalam bentuk tabel dan dapat dilihat pada Gambar 4.22 dan pada halaman itu juga terdapat fitur hapus admin dan tambah data admin, *form* tambah data admin dapat dilihat pada Gambar 4.23. Pada Gambar 4.18(3) merupakan menu mahasiswa, jika menu mahasiswa di klik akan diarahkan kehalaman tampil mahasiswa dapat dilihat pada Gambar 2.24, pada halaman ini terdapat fitur tambah, edit, detail mahasiswa dan hapus data. Jika fitur detail di klik akan menampilkan data mahasiswa secara keseluruhan dapat dilihat pada Gambar 4.25 dan jika fitur tambah di klik akan diarahkan ke halaman *form* tambah data dapat dilihat pada Gambar 4.26.

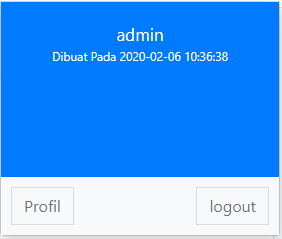
Pada Gambar 4.18(4) merupakan menu kriteria yang ada pada sistem apabila menu kriteria di klik akan diarahkan ke halaman kriteria, pada halaman kriteria terdapat data kriteria dan bobot kriteria yang bisa di edit seperti yang terlihat pada Gambar 4.27. Sedangkan pada Gambar 4.18(5) merupakan menu hasil perangkingan dapat di lihat pada Gambar 4.28 dan Gambar 4.29. Pada Gambar 4.28 merupakan data tabel mahasiswa dengan skor masing-masing kriteria sebanyak 10 kriteria dengan kode c1 sampai c10, sedangkan pada Gambar 4.29 merupakan hasil perhitungan dengan menggunakan metode *weighted product* setelah menekan tombol hitung sehingga menghasilkan data peangkingan dan data diurutkan dari yang terbesar sesuai dengan nilai akhir yang diperoleh.



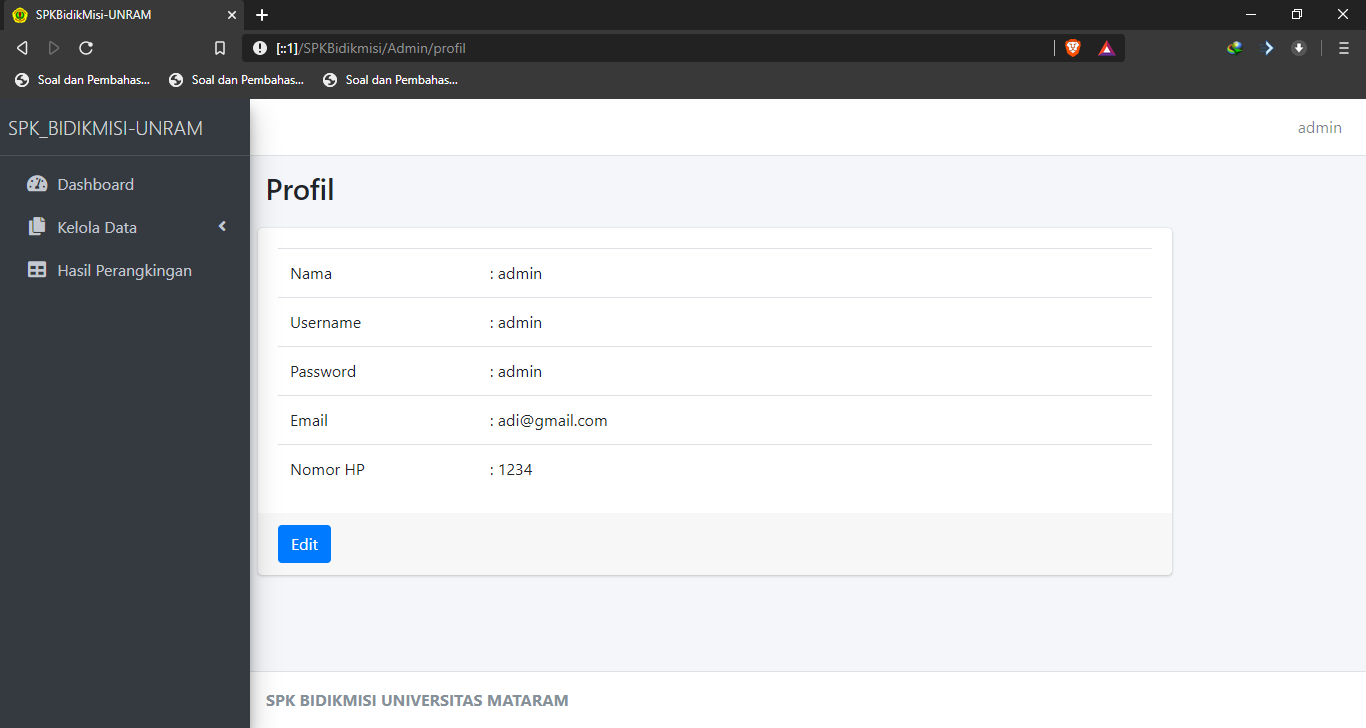
Gambar 4. 17 Tampilan halaman *login*



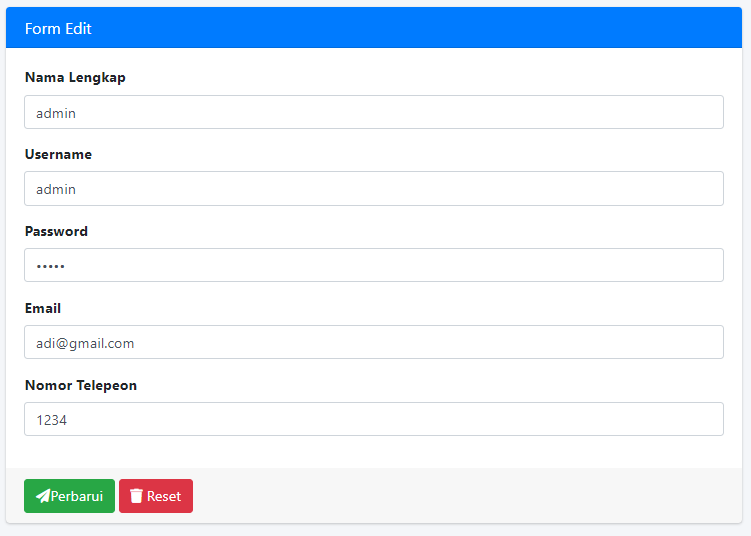
Gambar 4. 18 Tampilah halaman *dashboard*



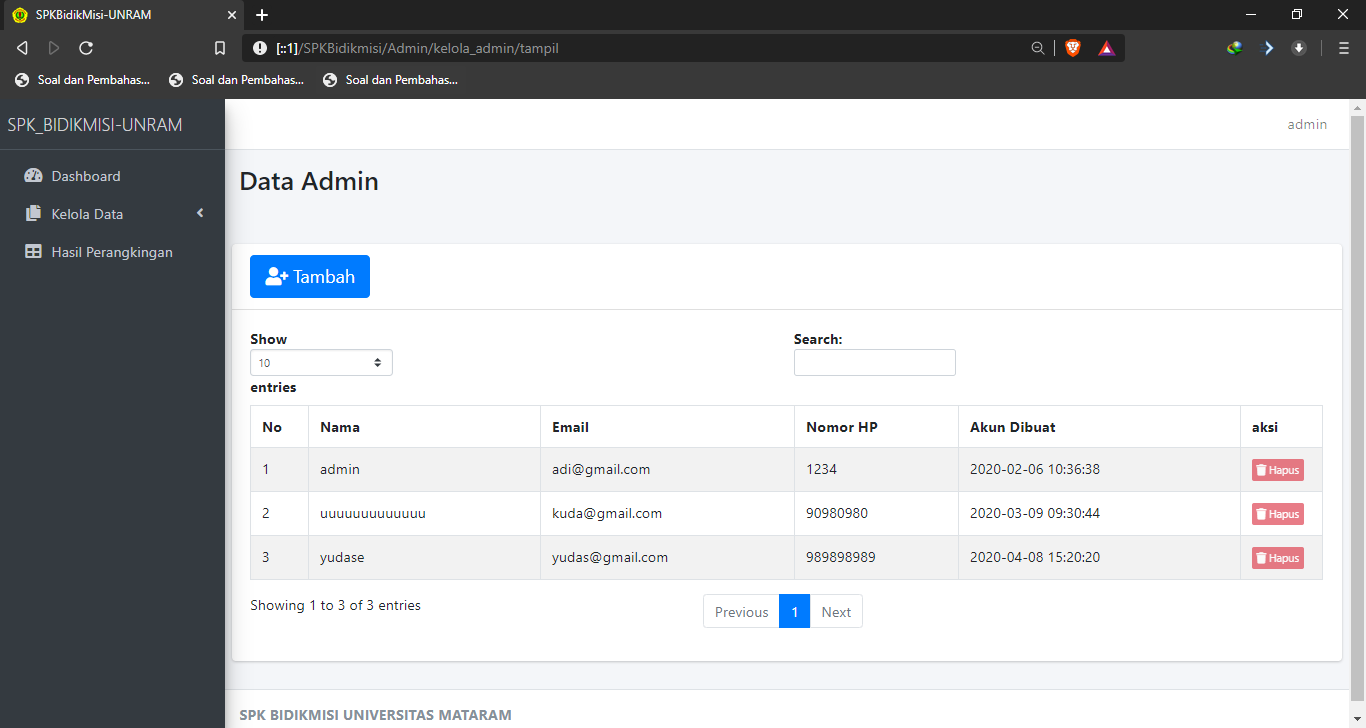
Gambar 4. 19 Tampilan menu admin



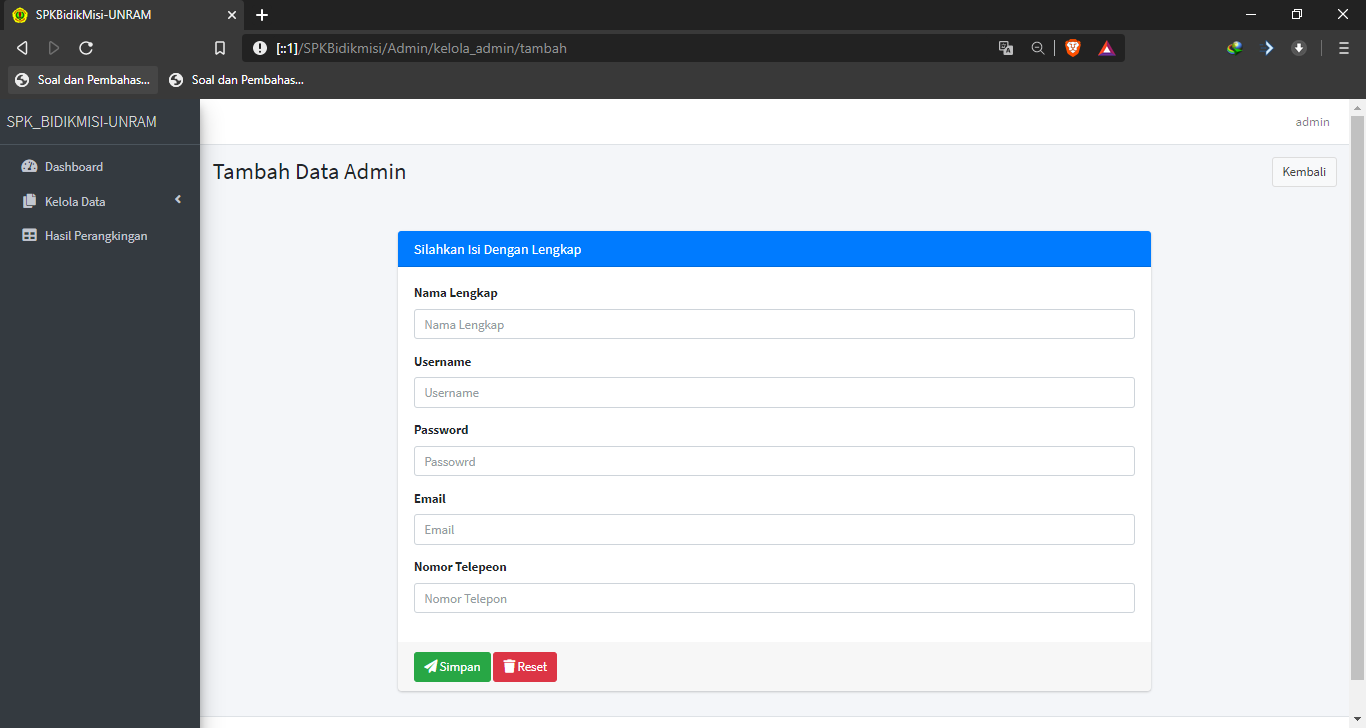
Gambar 4. 20 Tampilan profil



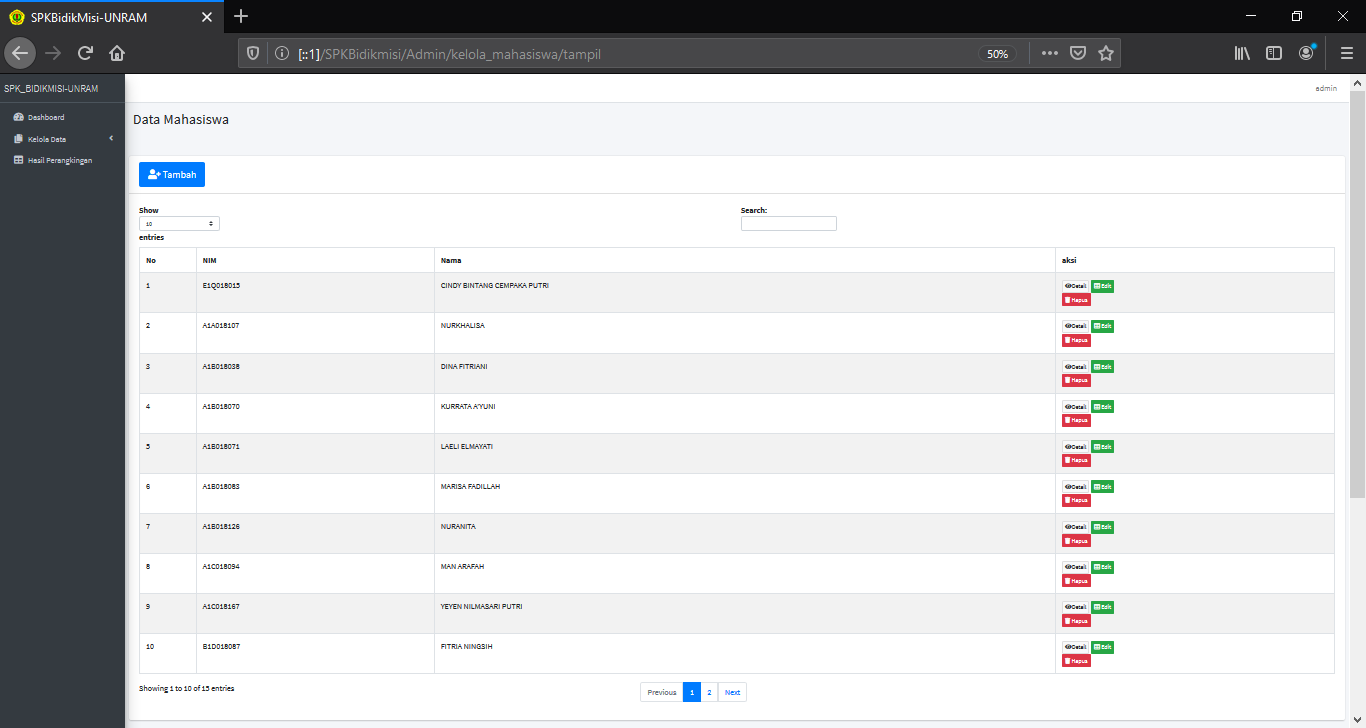
Gambar 4. 21 *form* edit profil admin



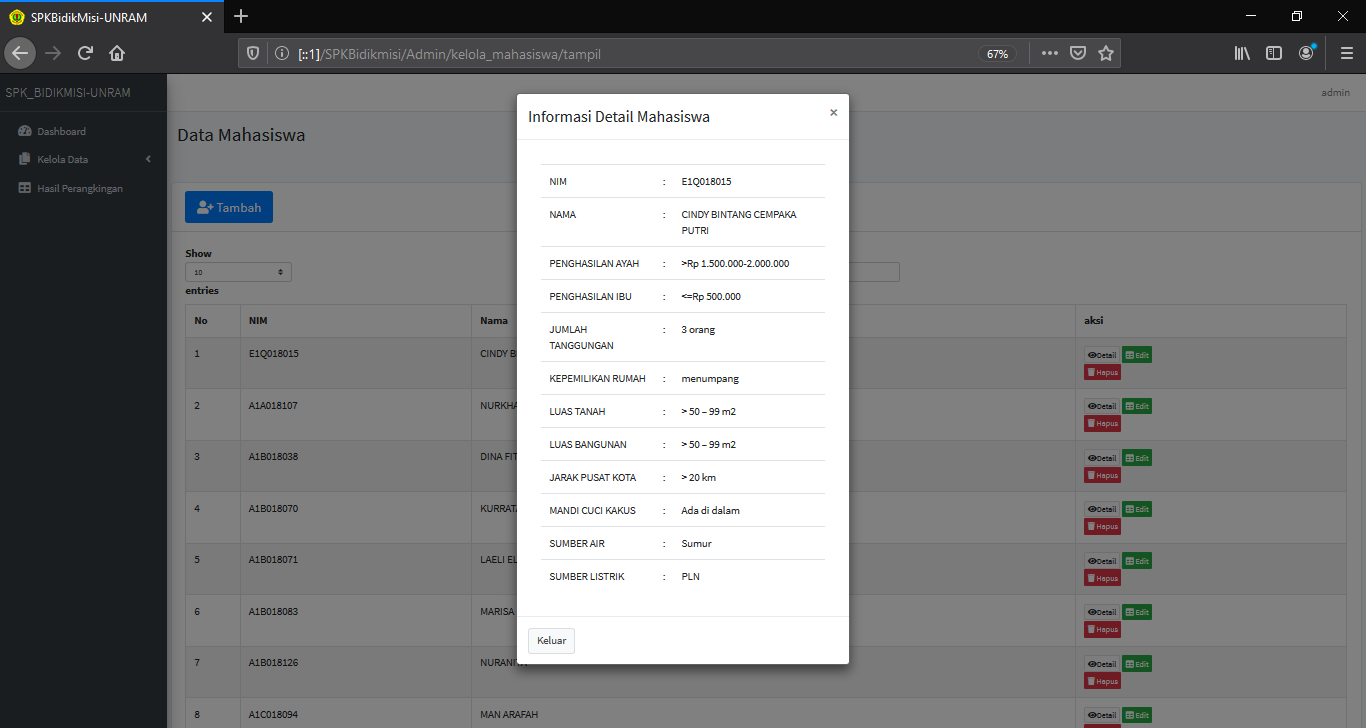
Gambar 4. 22 Halaman data admin



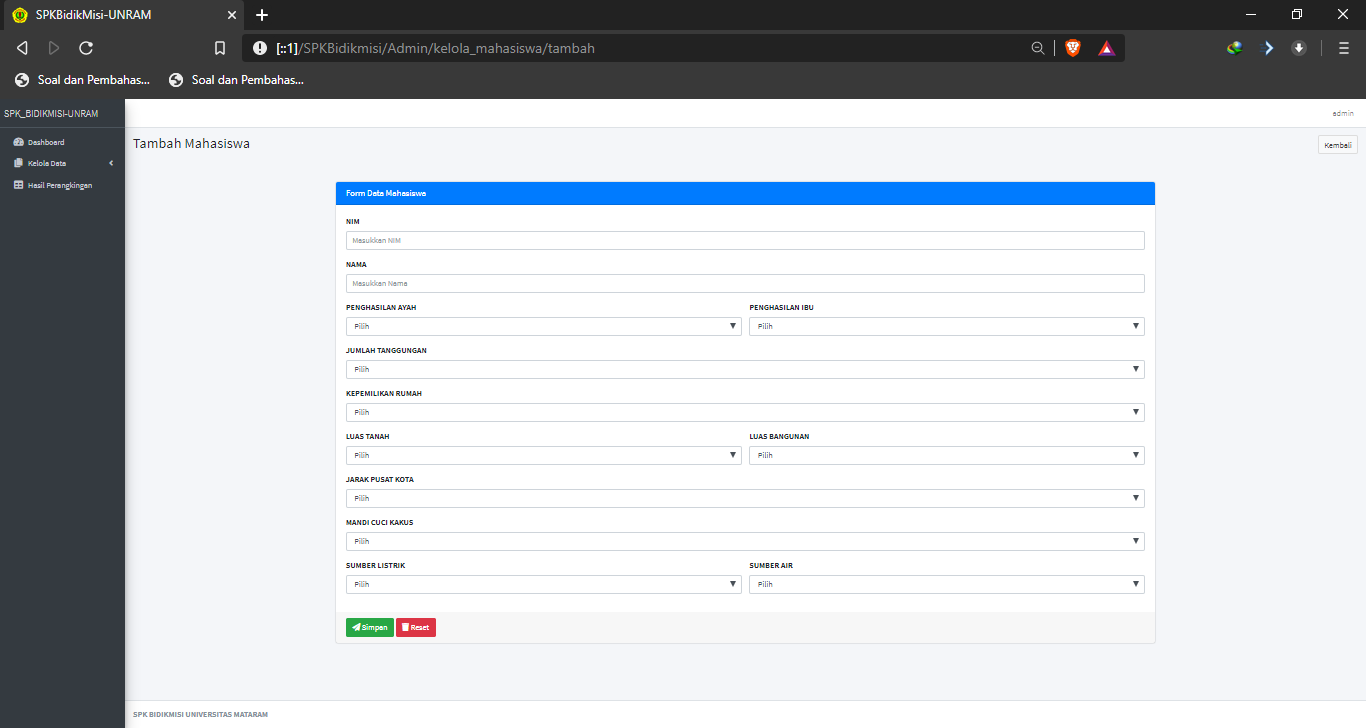
Gambar 4. 23 Halaman tambah admin



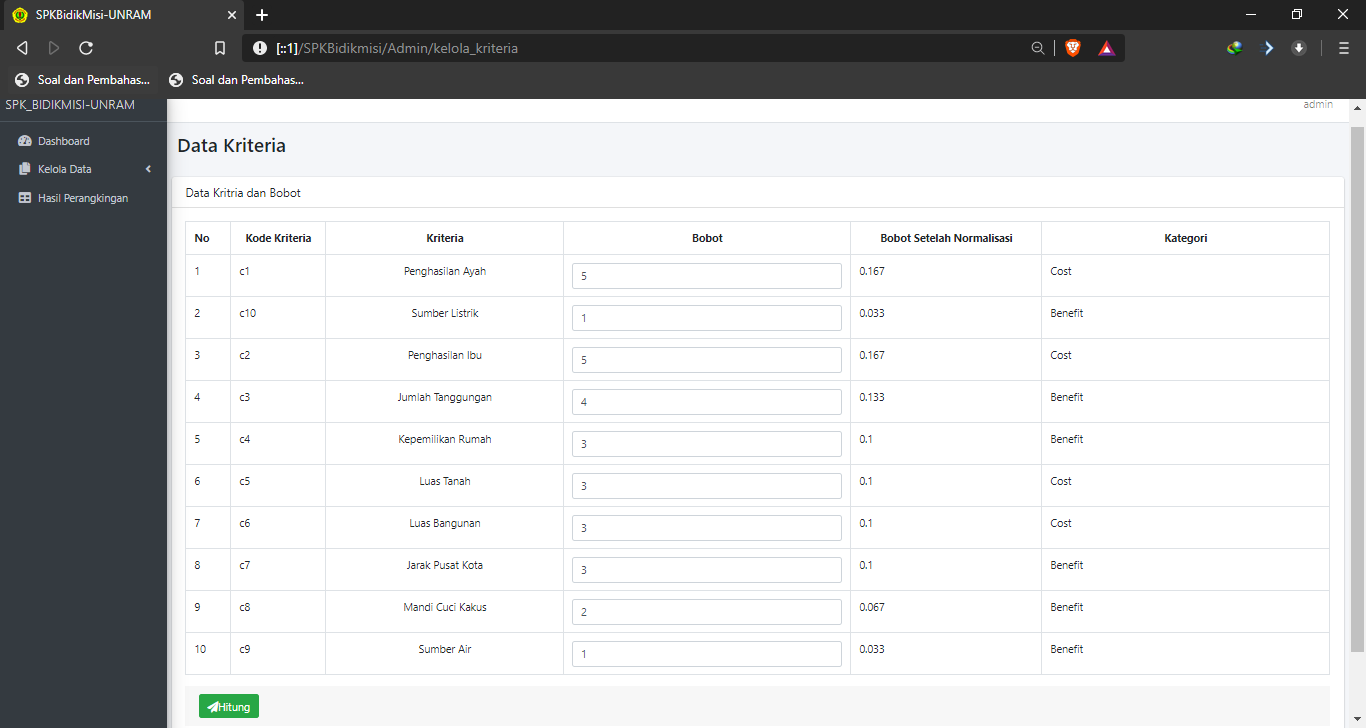
Gambar 4. 24 Halaman data mahasiswa



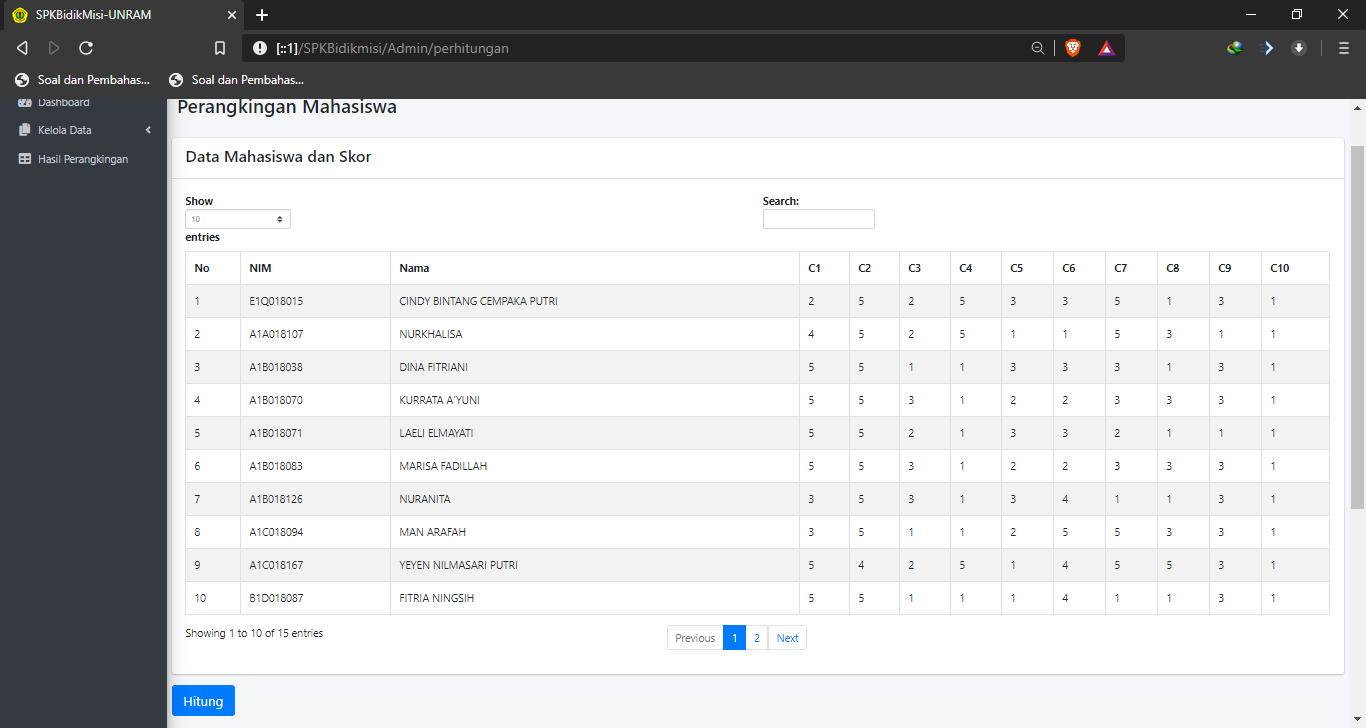
Gambar 4. 25 Menampilkan detail informasi mahasiswa



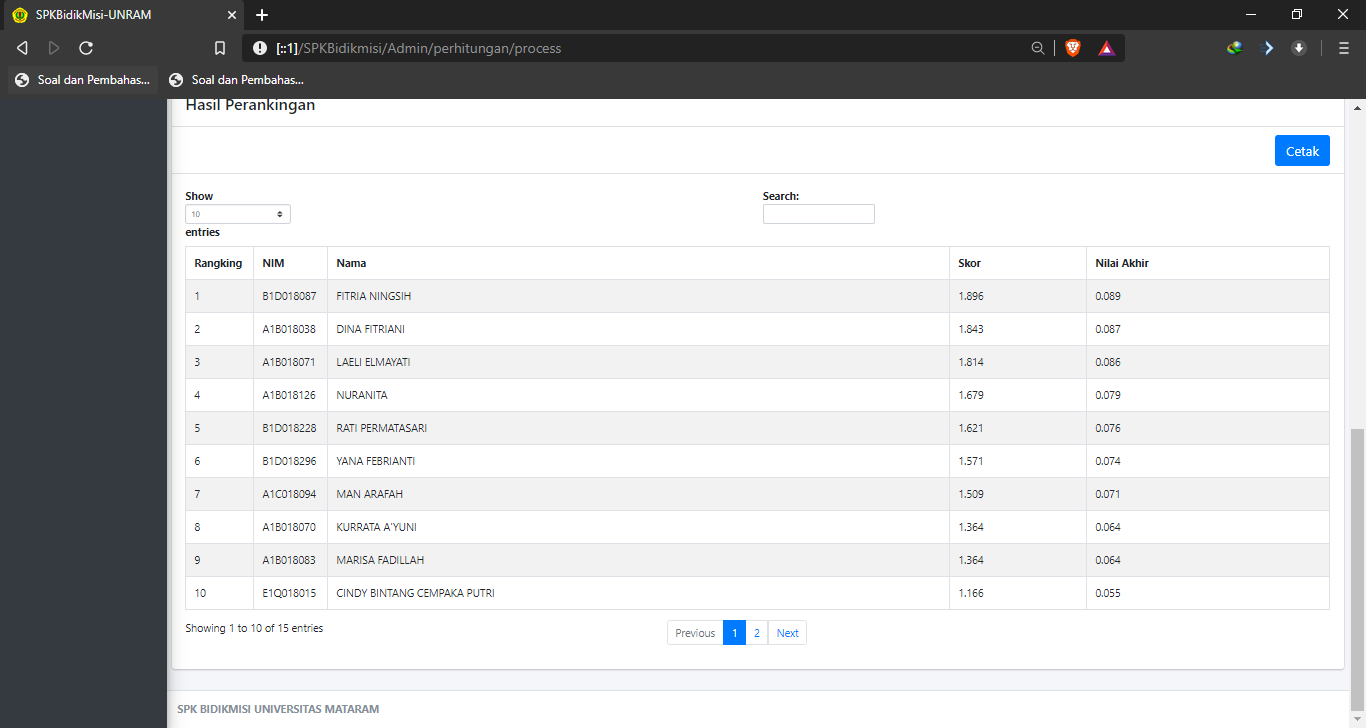
Gambar 4. 26 Menampilkan *form* tambah data mahasiswa



Gambar 4. 27 Tampilan halaman kriteria dan bobot



Gambar 4. 28 Tampilan data skor mahasiswa



Gambar 4. 29 Tampilan hasil perangkingan

### **Implementasi Hasil Perangkingan**

Pada tahap ini akan ditampikan hasil perhitungan menggunakan *microsoft excel* dan hasil perhitungan yang didapat sistem dengan metode yang sama dengan jumlah data yang sama yaitu menggunakan 15 sampel data mahasiswa. Berikut hasil masing-masing perhitungan yang didapat.

Tabel 4. 1 Hasil perhitungan sistem

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ranking | NIM | Nama | Nilai Akhir (Sistem) | Nilai Akhir (*Excel*) | Keterangan |
| 1 | B1D018087 | FITRIA NINGSIH | 0,089 | 0,089 | Sama |
| 2 | A1B018038 | DINA FITRIANI | 0,087 | 0,087 | Sama |
| 3 | A1B018071 | LAELI ELMAYANTI | 0,086 | 0,086 | Sama |
| 4 | A1B018126 | NURANITA | 0,079 | 0,079 | Sama |
| 5 | B1D018228 | RATI PERMATASARI | 0,076 | 0,076 | Sama |
| 6 | B1D018296 | YANA FEBRIANTI | 0,074 | 0,074 | Sama |
| 7 | A1C018094 | MAN ARAFAH | 0,071 | 0,071 | Sama |
| 8 | A1B018070 | KURRATA A’YUNI | 0,064 | 0,064 | Sama |
| 9 | A1B018083 | MARISA FADILLAH | 0,064 | 0,064 | Sama |
| 10 | E1Q018015 | CINDI BINTANG CEMPAKA PUTRI | 0,055 | 0,055 | Sama |
| 11 | B1D018146 | LIANA WULANDARI | 0,053 | 0,053 | Sama |
| 12 | B1D018312 | ZULFIKARULLAH | 0,052 | 0,052 | Sama |
| 13 | A1C018167 | YEYEN NILAMSARI PUTRI | 0,051 | 0,051 | Sama |
| 14 | B1D018166 | MARSINA | 0,050 | 0,050 | Sama |
| 15 | A1A018107 | NURKHALISA | 0,048 | 0,048 | Sama |

Pada tabel 4.1 merupakan data mahasiswa beserta hasil perhitungan menggunakan aplikasi *Microsoft excel* dan perhitungan sistem yang dihitung menggunakan rumus metode *weighted product*. Dari hasil nilai akhir yang ditampilkan baik dengan *excel* maupun sistem dengan jumlah dan data yang sama menghasilkan nilai akhir sama atau tingkat akurasi sebesar 100%. Sedangkan pada tabel 4.2 merupakan perbandingan urutan pada SPK menggunakan metode WP dengan data yang ada pada *Microsoft excel*.

Tabel 4. 2 Tabel Hasil SPK

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Urutan Sistem** | **NIM** | **Nama** | **Nilai Akhir** | **Urutan Di*excel*** |
| 1 | D1A018290 | YUDA PRSETYO | 0.015 | 93 |
| 2 | C1B018039 | RENI JULIANI | 0.013 | 81 |
| 3 | B1D018087 | FITRIA NINGSIH | 0.013 | 10 |
| 4 | C1M018040 | ELSA DAMAYANTI | 0.013 | 88 |
| 5 | C1L018063 | SURYATI | 0.013 | 23 |
| 6 | D1A018276 | URIT SANTIKA | 0.013 | 91 |
| 7 | D1A018277 | USWATUN HASANAH | 0.013 | 92 |
| 8 | A1B018038 | DINA FITRIANI | 0.013 | 3 |
| 9 | A1B018071 | LAELI ELMAYATI | 0.012 | 5 |
| 10 | F1A018007 | AKMALUDIN | 0.012 | 51 |
| 11 | J1B018024 | EVI WAHYUNI | 0.012 | 63 |
| 12 | C1K018059 | WULAN RAHMANIA | 0.012 | 20 |
| 13 | L1B018096 | RATU FADILLAH | 0.012 | 71 |
| 14 | J1B018083 | RIZKA NURHALIFAH | 0.012 | 67 |
| 15 | E1F018027 | FITRIANI | 0.012 | 43 |
| 16 | C1M018062 | IRA AISYAH | 0.012 | 25 |
| 17 | J1B018035 | ISKANDAR JULKARNAIN | 0.012 | 65 |
| 18 | E1C018078 | NURHASANAH | 0.012 | 38 |
| 19 | C1G018130 | NURFADILAH | 0.012 | 17 |
| 20 | C1M018098 | NURKHOFIFFAH ATMA NEGARA | 0.012 | 89 |
| 21 | C1G018089 | LUH ANDANI DWI AKARI | 0.012 | 85 |
| 22 | A1B018126 | NURANITA | 0.012 | 7 |
| 23 | E1A018085 | SAFIKA UMURAHMA ABD | 0.011 | 94 |
| 24 | J1B018067 | NURFIKHIYAH AULIYA | 0.011 | 66 |
| 25 | G1B018048 | NURFITRIANINGSIH | 0.011 | 60 |
| 26 | C1M08099 | NURWAFIQ AZIZAH | 0.011 | 90 |
| 27 | G1B018042 | MUHAMMAD HUMAM SAIFUL JIHADI | 0.011 | 59 |
| 28 | B1D018228 | RATI PERMATASARI | 0.011 | 13 |
| 29 | E1E018117 | QURATUL UYYUN | 0.011 | 41 |
| 30 | C1B018051 | WAHIDIN | 0.011 | 82 |
| 31 | F1C018011 | AYATULLAH | 0.011 | 99 |
| 32 | B1D018256 | SISKA ROSALINDA | 0.011 | 79 |
| 33 | E1C018005 | ANDI AKBAR | 0.011 | 95 |
| 34 | B1D018296 | YANA FEBRIANTI | 0.011 | 14 |
| 35 | E1B018071 | NURUL HIDAYATULAH | 0.011 | 36 |
| 36 | L1C018010 | ANITA KARDIYANTI | 0.011 | 73 |
| 37 | F1A018079 | NURUL FAUZIAH | 0.011 | 53 |
| 38 | C1L018021 | FIQRIYADI | 0.011 | 21 |
| 39 | D1A018116 | HAERUNISAH | 0.011 | 30 |
| 40 | A1C018094 | MAN ARAFAH | 0.010 | 8 |
| 41 | F1D018057 | ULAN RAFIKA | 0.010 | 55 |
| 42 | C1B018024 | LARASSATY OCTAPRAMA | 0.010 | 80 |
| 43 | L1B018018 | ARYA HIDAYAT | 0.010 | 70 |
| 44 | E1D018040 | FITRIANINGSIH | 0.010 | 96 |
| 45 | E1F018066 | NUR FANI | 0.010 | 44 |
| 46 | J1B018095 | SRI WARDAN | 0.010 | 65 |
| 47 | A1C018118 | NENGAH SUDARSANA | 0.010 | 77 |
| 48 | E1Q018022 | ELZA ULFANI | 0.010 | 48 |
| 49 | L1C018059 | MIFTAHUL JANNAH | 0.010 | 74 |
| 50 | B1D018165 | MARPATI | 0.010 | 78 |
| 51 | L1C018063 | MUH NADIRIN | 0.010 | 75 |
| 52 | A1A018147 | WIRANTO | 0.010 | 76 |
| 53 | G1C018001 | AAN ARISKA | 0.010 | 62 |
| 54 | E1A018075 | NURSEHA | 0.010 | 33 |
| 55 | E1E018015 | ANJELI SAFITRI | 0.010 | 39 |
| 56 | G1B018033 | MARDATILLAH | 0.010 | 58 |
| 57 | J1A018061 | KURATUL AINI | 0.010 | 100 |
| 58 | F1A018093 | SANTRI ISLAMIATI | 0.010 | 54 |
| 59 | C1M018113 | ROSFANI | 0.009 | 26 |
| 60 | F1A018063 | MAYA MASITA | 0.009 | 52 |
| 61 | G1B018050 | NURFADHILA | 0.009 | 61 |
| 62 | E1B018061 | MULYADIN | 0.009 | 35 |
| 63 | E1R018084 | UMAYRAH | 0.009 | 97 |
| 64 | A1B018070 | KURRATA A'YUNI | 0.009 | 4 |
| 65 | A1B018083 | MARISA FADILLAH | 0.009 | 6 |
| 66 | J1B018087 | SALSABILA PUTRI | 0.009 | 68 |
| 67 | E1S018062 | RASMIATURRAHMANIA | 0.009 | 98 |
| 68 | C1G018113 | MUHAMMAD SAIDIN | 0.009 | 16 |
| 69 | C1G018157 | ROSA PUTRI AMALIA | 0.009 | 86 |
| 70 | E1F018075 | NURZAKIAH | 0.009 | 45 |
| 71 | E1E018082 | LISHA LUSYANA | 0.009 | 40 |
| 72 | E1Q018062 | ROSTINA | 0.009 | 49 |
| 73 | E1B018007 | ASNI PUTRI AGUSTIN | 0.009 | 34 |
| 74 | C1G018158 | ROSDIANA | 0.009 | 18 |
| 75 | C1G018042 | FANI PUTRIANI | 0.009 | 83 |
| 76 | E1M018002 | ALFIDA FITRI WARDANI | 0.009 | 46 |
| 77 | E1M018057 | NUR ASIA | 0.009 | 47 |
| 78 | C1G018186 | YENY WANTINY | 0.009 | 87 |
| 79 | C1G018088 | LISA AULIA KARTINI | 0.009 | 84 |
| 80 | D1A018009 | AGUSTINA | 0.009 | 28 |
| 81 | L1C018006 | AJIBURRAHMAN | 0.009 | 72 |
| 82 | C1M018014 | ARIS MUNANDAR | 0.009 | 24 |
| 83 | E1F018002 | AINUN JARIAH | 0.008 | 42 |
| 84 | G1A018053 | RAMDONI PUJI ASTUTI | 0.008 | 56 |
| 85 | D1A018123 | HENI | 0.008 | 31 |
| 86 | C1L018062 | SULISTIAWATI | 0.008 | 22 |
| 87 | D1A018220 | NUR INTAN | 0.008 | 32 |
| 88 | D1A018103 | FIKRAM WADDHAULLAH | 0.008 | 29 |
| 89 | E1Q018015 | CINDY BINTANG CEMPAKA PUTRI | 0.008 | 1 |
| 90 | C1K018021 | FARAH JULIANTI | 0.008 | 19 |
| 91 | E1S018090 | UMRATULMUNTAHA | 0.008 | 50 |
| 92 | B1D018146 | LIANA WULANDARI | 0.008 | 11 |
| 93 | J1B018026 | FUJIATIN | 0.008 | 64 |
| 94 | B1D018312 | ZULFIKARULLAH | 0.008 | 15 |
| 95 | A1C018167 | YEYEN NILMASARI PUTRI | 0.007 | 9 |
| 96 | E1C018030 | FUJIATUN | 0.007 | 37 |
| 97 | G1A018077 | WULANDARI | 0.007 | 57 |
| 98 | B1D018166 | MARSINA | 0.007 | 12 |
| 99 | A1A018107 | NURKHALISA | 0.007 | 2 |
| 100 | C1M018118 | SINTA NURYAH | 0.006 | 27 |

Pada tabel diatas merupakan hasil dari perangkingan yang dilakukan oleh SPK dengan menggunakan metode WP yang menggunakan 10 kriteria sebagai acuan dalam proses perhitungan. Pada kolom rank sistem merupakan hasil pengurutan atau prioritas dalam mendapatkan beasiswa bidikmisi yang bisa digunakan sebagai acuan dalam mengambil keputusan ,kemudian dilakukan survei lapangan terkait data-data yang ada sebelum mengambil keputusan secara *final*. Sedangkan kolom urutan *excel* merupakan nomor urut mahasiswa pada file *Microsoft excel* yang diurutkan berdasarkan NIM atau Prodi masing-masing mahasiswa.

## **Pengujian Sistem**

Pada sistem pendukung keputusan yang dibuat dilakukan dua metode pengujian antara lain metode *black* *box testing* dan metode *white box* *testing*. Pada pengujian *black* *box testing* hal-hal yang diuji yaitu fungsionalitas pada sistem sedangkan metode *white box testing* menguji kebenaran dari metode yang digunakan. Berikut diterapkan masing – masing pengujian terhadap sistem.

### **Pengujian *black box***

Pada pengujian ini dilakukan pengujian fitur-fitur yang terdapat pada sistem yang dilakukan oleh *developer* sendiri dan pegawai sub bagian kesejahteraan mahasiswa, untuk lebih jelasnya dapat dilihat pada tabel 4.3.

Tabel 4. 3 Pengujian *black box testing*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| No | Fungsi | Hasil yang diharapkan | Hasil pengujian | |
| Sesuai | Tidak |
| 1 | Login sebagai admin | Fungsi login sebagai admin sudah benar | **✔** |  |
| 2 | Tambah data admin | Fungsi tambah data admin sudah benar | **✔** |  |
| 3 | Edit data admin | Fungsi edit data admin sudah benar | **✔** |  |
| 4 | Hapus data admin | Fungsi hapus data admin sudah benar | **✔** |  |
| 5 | Tambah data mahasiswa | Fungsi tambah data mahasiswa sudah benar | **✔** |  |
| 6 | Edit data mahasiswa | Fungsi edit data mahasiswa sudah benar | **✔** |  |
| 7 | Hapus data mahasiswa | Fungsi hapus data mahasiswa sudah benar | **✔** |  |
| 8 | Lihat detail data mahasiswa | Fungsi lihat detail data mahasiswa sudah benar | **✔** |  |
| 9 | Edit bobot kriteria | Fungsi edit bobot kriteria sudah benar | **✔** |  |
| 10 | Cetak data hasil perankingan | Fungsi cetak data hasil perangkingan sudah benar | **✔** |  |

### **Pengujian *white box***

Pada pengujian ini dilakukan pengecekan langkah-langkah metode yang digunakan pada sistem yaitu metode *weighted product* dalam melakukan perhitungan. Dalam pengujian ada dua *script* yaitu perhitungan bobot dan nilai akhir. Berikut pengujian yang dilakukan dengan metode *white box*:

* + - 1. Pengujian *white box* pada perhitungan bobot

Dalam melakukan pengujian ini terlebih dahulu dilakukan pengubahan *code* algoritma proses menghitung bobot normalisasi pada metode *weighted product*.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18 | public function hitungBobot()      {          cek\_belum\_login();          $post=$this->input->post(null, TRUE);          $this->load->model('Admin/kriteria\_m');          $TOTAL\_BOBOT = $post['c1'] + $post['c2'] +    $post['c3'] + $post['c4'] + $post['c5'] +          $post['c6'] + $post['c7'] + $post['c8'] +    $post['c9'] + $post['c10'];          for($i = 1; $i <= 10; $i++) {              $index = 'c'.$i;              $data['krit\_bobot\_s'] =  number\_format  ( $post[$index] / $TOTAL\_BOBOT , 3);              $data['krit\_bobot'] = $post[$index];              $data['id'] = $index;              $this->kriteria\_m->edit($data); }          $this->index();  } |

Berdasarkan *code* algoritma proses menghitung bobot normalisasi pada metode *weighted product* di atas akan di ubah kedalam bentuk *basis path* sebagaimana dapat dilihat pada Gambar 4.30.

|  |
| --- |
|  |

Gambar 4. 30 *Basis* *path* proses menghitung bobot normalisasi

Keterangan :

**=** Menggambarkan kondisi

= Menggambarkan aksi

Dari Gambar 4.30 dapat dihitung *cylomatic complexity* sebagai berikut :

V(G) = E – N+2

V(G) = 9 – 8 + 2

V(G) = 1

Dimana :

E = Jumlah *edge* pada *basis path*

N = Jumlah *node* pada *basis path*

Jadi, berdasarkan hasil perhitungan *cylomatic complexity* diatas, maka didapatkan *path* atau jalur metode *weighted product* dalam proses menghitung bobot normalisasi sebagai berikut : Path 1 : 1, 2, 3, 4, 5, 6-9, 10-16, 17, 18

Selanjutnya dilakukan pengujian path algoritma proses menghitung bobot normalisasi dapat dilihat pada Tabel 4.4

Tabel 4. 4 Pengujian proses perhitugan bobot

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Path | Kasus Uji | Hasil Yang Diharapkan | Hasil Uji | Keterangan |
| 1 | Menginputkan bobot setiap kriteria | * Data inputan masuk ke database * Berhasil menghitung bobot normalisasi | * Data berhasil tersimpan * Mendapat hasil perhitungan | [✔] Alur terlewati  [ ] Alur tidak terlewati |

* + - 1. Pengujian perhitungan vektor s dan vektor v

Pada tahap ini dilakukan pengujian perhitungan nilai vektor s dan vektor v yang merupakan bagian dari metode *weighted product* setelah menentukan nilai bobot normalisasi. Terlebih dahulu dilakukan pengubahan *code* algoritma perhitungan vektor s dan vektor v.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52 | public function process()      {          $data\_mahasiswa['row'] = $this->mahasiswa\_m->tampil();  $data\_kriteria['row'] = $this->kriteria\_m->tampil();          $kriteria = $data\_kriteria['row']->result\_array();          $data = $data\_mahasiswa['row']->result\_array();          for($i = 0; $i < count($data); $i++) {               for($j = 1; $j <= 10; $j++) {                  $index = 'c' . $j;                  $bobot\_normalisasi;                  $kategori;                  for($k = 0; $k < count($kriteria); $k++) {                      if ($kriteria[$k]['krit\_id'] == $index) {                          $bobot\_normalisasi = $kriteria[$k]  ['krit\_bobot\_s'];                          $kategori = $kriteria[$k]['kategori'];                      }                  }                  if ($kategori == 'Benefit') {                      $data[$i][$index . '\_hasil'] = pow($data[$i]  [$index], ($bobot\_normalisasi \* -1));                  } else {                      $data[$i][$index . '\_hasil'] = pow($data[$i]  [$index], $bobot\_normalisasi);                  }              }              $data[$i]['vektor\_s'] = $data[$i]['c1\_hasil'] \* $data  [$i]['c2\_hasil'] \* $data[$i]['c3\_hasil'] \*                         $data[$i]['c4\_hasil'] \* $data[$i]  ['c5\_hasil'] \* $data[$i]['c6\_hasil'] \*                          $data[$i]['c7\_hasil'] \* $data[$i]  ['c8\_hasil'] \* $data[$i]['c9\_hasil'] \*                          $data[$i]['c10\_hasil'];          }          $total\_vektor\_s = 0;          for($i = 0; $i < count($data); $i++) {              $total\_vektor\_s = $total\_vektor\_s + $data[$i]['vektor\_s'];          }           for($i = 0; $i < count($data); $i++) {              $data[$i]['vektor\_v'] = $data[$i]['vektor\_s'] /  $total\_vektor\_s;          }    $data\_mahasiswa['row']=$this->mahasiswa\_m->tampil()->result\_array();          usort($data, function($a, $b) {              return $b['vektor\_v'] <=> $a['vektor\_v'];          });          $data\_mahasiswa['perhitungan'] = $data;          $this->template->load('template', 'Admin/Hasil\_Perangkingan', $data\_mahasiswa);    } |

Berdasarkan *code* algoritma proses menghitung nilai vektor s dan vektor v pada metode *weighted product* diatas akan diubah kedalam bentuk *basis path* sebagaimana dapat dilihat pada Gambar 4.31



Gambar 4. 31 *Basis path* proses menghitung vektor s dan vektor v

Keterangan :

**=** Menggambarkan kondisi

= Menggambarkan aksi

Dari Gambar 4.31 dapat dihitung *cylomatic complexity* sebagai berikut :

V(G) = E – N+2

V(G) = 27 – 26 + 2

V(G) = 3

Dimana :

E = Jumlah *edge* pada *basis path*

N = Jumlah *node* pada *basis path*

Jadi, berdasarkan hasil perhitungan *cylomatic complexity* diatas, maka didapatkan 3 path atau jalur metode *weighted product* dalam proses menghitung vektor s dan vektor v sebagaimana dapat dilihat pada Tabel 4. 5

Tabel 4. 5 *Path* proses menghitung vektor s dan vektor v

|  |  |
| --- | --- |
| Path 1 | 1, 2, 3, 4, 5, 6, 7-11, 12, 13, 14-16, 17, 18, 19, 20-24, 25, 26, 27-33, 34, 35-39, 40-43, 44-45, 46-48, 49, 50, 51, 52-53, 54 |
| Path 2 | 1, 2, 3, 4, 5, 6, 7-11, 12, 13, 17, 18, 19, 20-24, 25, 26, 27-33, 34, 35-39, 40-43, 44-45, 46-48, 49, 50, 51, 52-53, 54 |
| Path 3 | 1, 2, 3, 4, 5, 6, 7-11, 12, 13, 14-16, 17, 18, 19, 25, 26, 27-33, 34, 35-39, 40-43, 44-45, 46-48, 49, 50, 51, 52-53, 54 |

Selanjutnya dilakukan pengujian *path* algoritma proses perhitugan vektor s dan vektor v dapat dilihat pada Tabel 4.6

Tabel 4. 6 Pengujian proses perhitugan vektor s dan vektor v

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Path | Kasus Uji | Hasil Yang Diharapkan | Hasil Uji | Keterangan |
| 1 | Menghitung nilai vektor s dan vektor v | Mendapatkan hasil perhitungan vektor s dan vektor v | Berhasil menghitung vektor s dan vektor v | [✔] Alur terlewati  [ ] Alur tidak terlewati |
| 2 | Mencari nilai bobot normalisasi sesuai dengn id kriteria | Mendapat nilai bobot normalisasi sesuai dengn id kriteria | Berhasil mendapat nilai bobot normalisasi sesuai dengn id kriteria | [✔] Alur terlewati  [ ] Alur tidak terlewati |
| 3 | Menentukan kategori kriteria untuk mendapat nilai bobot normalisasi untuk mendapat nilai bobot bernilai positif atau negatif | Menadapat nilai bobot yang bernilai positif atau negatif sesauai dengan kategori | Berhasil menadapat nilai bobot yang bernilai positif atau negatif sesauai dengan kategori | [✔] Alur terlewati  [ ] Alur tidak terlewati |

### **Pengujian dengan kuisioner**

Pengujian dengan kuisioner dilakukan untuk mengetahui kinerja dan kualitas sistem dari sisi pengguna. Sebelum dapat mengisi kuisioner pengguna terlebih dahulu menjalankan sistem yang telah dibuat. Kemudian untuk mengetahui kinerja sistem pengguna diberikan kuisioner untuk diisi. Hasil kuisioner kemudian dikalkulasi untuk mengetahui kualitas sistem. Pengujian pertama dilakukan oleh lima responden yang merupakan pegawai Subbagian Kesejahteraan Mahasiswa Universitas Mataram. Pengujian dilakukan cara memberikan penjelesan tentang kegunaan dari sistem yang telah dibuat dan alur kerja dari sistem. Kemudian responden mencoba menggunakan sistem dengan mencoba mengelola data mahasiswa lalu mengisi kuisioner. Pengujian kedua dilakukan dengan kusioner online dengan menampilkan gambar hasil *screenshoot* dari halaman beranda dan hasil seleksi mahasiswa penerima beasiswa bidikmisi, terdapat 30 responden yang diambil secara acak dari mahasiswa universitas mataram.

Hasil dari jawaban responden nantinya akan dihitung dan ditarik kesimpulan mengenai hasil pengujian sistem. MOS ( *Mean Opinion Score* ) merupakan metode evaluasi yang menggunakan pendapat responden sebagai dasar penilaian. Perhitungan skor dari metode MOS dilakukan berdasarkan skala standar penilaian yang dimulai dari angka 1 hingga 5 [17]. Dapat dilihat pada Tabel 4.7 berikut.

|  |  |  |
| --- | --- | --- |
| No | Penilaian | Keterangan |
| 1 | 1 | Sangat Setuju |
| 2 | 2 | Setuju |
| 3 | 3 | Cukup |
| 4 | 4 | Tidak Setuju |
| 5 | 5 | Sangat Tidak Setuju |

Tabel 4. 7 Keterangan Penilaian Kusioner

Berikut daftar pertanyaan yang diajukan kepada responden pegawai Subbagian Kesejahteraan Mahasiswa Universitas Mataram:

1. Apakah tampilan sistem mudah digunakan
2. Apakah menu menu yang ada pada sistem mudah dipahami
3. Apakah alur dari proses perhitungan dan perangkingan pada sistem mudah dipahami
4. Apakah sistem sudah bisa memberikan informasi terkait hasil seleksi mahasiswa peserta beasiswa bidikmisi
5. Apakah sistem mempermudah dalam pelaporan dan sudah sesuai dengan hasil

Berikut daftar pertanyaan yang diajukan kepada responden mahasiswa Universitas Mataram :

Apakah tampilan sistem mudah digunakan

Apakah menu menu yang ada pada sistem mudah dipahami

Apakah sistem sudah bisa memberikan informasi terkait hasil seleksi mahasiswa peserta beasiswa bidikmisi

Pengujian dengan metode *Mean Opinion Score* Sistem Pendukung Keputusan Penerima Beasiswa Bidikmisi Universitas Mataram. Berikut ini perhitungan yang sudah dilakukan pada lima responden yaitu pegawai Subbagian Kesejahteraan Mahasiswa Universitas Mataram dapat dilihat pada Tabel 4.8.

Tabel 4. 8 Hasil Perhitungan MOS Pegawai

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Pertnyaan | Penilaian | | | | | Mean |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Pertanyaan 1 |  |  |  | 3 | 2 | 4,4 |
| 2 | Pertanyaan 2 |  |  |  | 1 | 4 | 4,8 |
| 3 | Pertanyaan 3 |  |  |  | 4 | 1 | 4,2 |
| 4 | Pertanyaan 4 |  |  |  | 3 | 2 | 4,4 |
| 5 | Pertanyaan 5 |  |  |  | 2 | 3 | 4,6 |
| Sub Total | |  |  |  | 13 | 13 | 22,4 |
| **MOS *(Mean Opinion Score)*** | | | | | | | 4,48 |

Berdasarkan hasil pengujian MOS yang dilakukan oleh lima pegawai Subbagian Kesejahteraan Mahasiswa Universitas Mataram dan penilian dengan *range* 1-5 mengacu pada Tabel 4.6, para responden telah memberikan penilaian pada setiap pertanyaan. Rincian penilaian MOS dapat dilihat pada Tabel 4.7. Pengujian yang dilakukan pada pegawai Subbagian Kesejahteraan Mahasiswa Universitas Mataram dengan MOS = 4.48 atau kategori  *Very Good*(Sangat Bagus) [18].

Berikut ini perhitungan yang sudah dilakukan pada 30 responden yaitu mahasiswa Universitas Mataram yang diambil secara acak dengan kusioner *online* dapat dilihat pada Tabel 4.8.

Tabel 4. 9 Hasil Perhitungan MOS Mahasiswa

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| No | Pertnyaan | Penilaian | | | | | Mean |
| 1 | 2 | 3 | 4 | 5 |
| 1 | Pertanyaan 1 |  |  | 3 | 19 | 8 | 4,17 |
| 2 | Pertanyaan 2 |  |  | 3 | 17 | 10 | 4,23 |
| 3 | Pertanyaan 3 |  |  | 4 | 18 | 8 | 4,13 |
| Sub Total | |  |  | 10 | 54 | 26 | 12,53 |
| **MOS *(Mean Opinion Score)*** | | | | | | | 4,17 |

Berdasarkan hasil pengujian MOS yang dilakukan oleh 30 mahasiswa Universitas Mataram yang diambil secara acak dan penilian dengan *range* 1-5 mengacu pada Tabel 4.6, para responden telah memberikan penilaian pada setiap pertanyaan. Rincian penilaian MOS dapat dilihat pada Tabel 4.8. Pengujian yang dilakukan pada pegawai mahasiswa Universitas Mataram dengan MOS = 4.17 atau kategori  *Good* (Bagus) [18].

# **BAB V**

**PENUTUP**

1. **Kesimpulan**

Berikut kesimpulan yang didapat dari sistem pendukung keputusan yang telah dibuat dengan metode *weighted product* :

1. Sistem pendukung keputusan yang telah dibuat sudah dapat menghasilkan prioritas mahasiswa yang menerima beasiswa bidikmisi tahun 2018 sebelum dilakukan survei lapangan.
2. Hasil yang didapat sistem dengan hasil perhitungan manual dengan *Microsoft excel* sudah sesuai dan menghasilkan nilai akhir yang sama dengan tingkat akurasi 100% yang dilakukan terhadap 15 data mahasiswa yang dijadikan sampel.
3. Hasil kusioner dari sisi admin atau pegawai mendapat nilai MOS sebesar 4,48 atau kategori *Very Good* (Sangat Bagus ) sedangkan dari sisi mahasiswa mendapat nilai MOS sebesar 4,17 atau kategori *Good* ( Bagus ).
4. **Saran**

Berikut saran untuk sistem pendukung keputusan yang telah dibuat agar bisa dikembangkan pada peneltian selanjutnya :

1. Diharapkan sistem dapat menambah kriteria dan bobot secara dinamis.
2. Menggunakan metode sistem pendukung keputusan yang lain untuk bisa dilakukan perbandingan hasil.

# **DAFTAR PUSTAKA**

[1] R. R. Ayu Septiana Sari, Jumadil Nangi, “Penerapan Metode Promethee Dalam Sistem Penunjang Keputusan Penentuan Penerima Beasiswa Bidik Misi Universitas Halu Oleo,” *Bianglala Inform.*, vol. 3, no. 1, pp. 229–236, 2015.

[2] R. Fauzan, Y. Indrasary, and N. Muthia, “Sistem Pendukung Keputusan Penerimaan Beasiswa Bidik Misi di POLIBAN Dengan Metode SAW Berbasis Web,” *JOIN (Jurnal Online Inform.*, vol. 2, no. 2, pp. 79–83, 2017, doi: 10.15575/join.v2i2.101.

[3] D. Puspitasari, M. Mentari, and F. A. Gunawan, “Sistem Pendukung Keputusan Penentuan Penerimaan Mahasiswa Baru Jalur Bidikmisi Menggunakan Metode Topsis (Studi Kasus : Politeknik Negeri Malang),” *J. Infromatika Polinema*, vol. 4, pp. 63–70, 2017.

[4] D. Kusumawati, “Penentuan Penerima Beasiswa Menggunakan Simple Additive Weighting Dan Weighted Product,” *J. Elektron. Sistim Inf. Dan Komput.*, vol. 1, no. 1, pp. 1–4, 2015.

[5] P. Oktavia, “Sistem Pendukung Keputusan Seleksi Penerima Beasiswa dengan Metode Weighted Product pada SMP Negeri 1 Parung Berbasis Web,” *J. Inform. Univ. Pamulang*, vol. 3, no. 2, p. 80, 2018, doi: 10.32493/informatika.v3i2.1432.

[6] Nurjaya, “Perbandingan Metode SAW Dengan Metode WP Pada Sistem Seleksi Karyawan Tetap,” 2017.

[7] M. Muslihudin, R. F. Andriyanti, and S. Mukodimah, “Implementasi Metode Weighted Product Menentukan Beasiswa Bidik Misi Stmik Pringsewu,” *Jatisi*, vol. 4, no. 2, pp. 156–163, 2018.

[8] M. Riadi, “Pengertian Sistem Pendukung Keputusan,” *Kajianpustaka.com*, 2013. [Online]. Available: https://www.kajianpustaka.com/2013/09/sistem-pendukung-keputusan-spk.html. [Accessed: 02-Oct-2019].

[9] H. K.S, H. S, and R. Warodyo, *Fuzzy Multi-Attribute Decision Making(Fuzzy MADM)*. Yogyakarta: Graha Ilmu, 2006.

[10] H. H, *Cara Instan Menguasi Pemrograman Website Secara Otodidak*. jakarta Barat: Agogos Publishing, 2011.

[11] D. Widodo, *Menggunakan UML*. Bandung: Informatika, 2011.

[12] I. Ketut Suharsana, I. Wirarama Wedashwara Wirawan, and N. L. A. K. Y. S, “Implementasi Model View Controller Dengan Framework Codeigniter Pada E-Commerce Penjualan Kerajinan Bali,” *J. Sist. dan Inform.*, vol. 11, no. 1, pp. 19–28, 2016.

[13] B. R, *Belajar Otodidak Membuat Database Menggunakan MySQL*. Bandung: Informatika, 2011.

[14] Pengajarku, “Analisis Sistem : Pengertian, Tujuan, Fungsi, Tahapan dan Contohnya,” *pengajar.co.id*, 2019. [Online]. Available: https://pengajar.co.id/analisis-sistem-pengertian-tujuan-fungsi-tahapan-dan-contohnya/. [Accessed: 08-Oct-2019].

[15] Dosenpendidikan, “Pegujian dan Pengertian White Box Testing,” *dosenpendidikan.co.id*, 2019. [Online]. Available: https://www.dosenpendidikan.co.id/pegujian-dan-pengertian-white-box-testing/. [Accessed: 17-Oct-2019].

[16] P. Kurniawati, “Pengujian Sistem,” *medium.com*, 2018. [Online]. Available: https://medium.com/skyshidigital/pengujian-sistem-52940ee98c77. [Accessed: 17-Oct-2019].

[17] U. Amri, I. G. P. S. Wijaya, and F. Bimantoro, “Steganografi Menggunakan Metode Pencocokan LSB dan Karakter Non-Breaking Space Sebagai Penanda Pesan,” *J. Comput. Sci. Informatics Eng.*, vol. 1, no. 1, p. 23, 2018, doi: 10.29303/jcosine.v1i1.18.

[18] R. Fitriyanti, Lindawati, and A. Aryanti, “Studi Literatur Mean Opinion Score Menggunakan Moving Picture Quality Metrics (MPQM) Di Jaringan LTE,” *Semin. Nas. Inov. dan Apl. Teknol. di Ind. 2018 Tema A - Penelit.*, vol. 4, no. 2, pp. 10–14, 2018.

LAMPIRAN

