

Implementation TOPSIS on Admission Application for Karanganyar Islamic State High School

Widyaningdyah Hidayati

Informatika, Fakultas MIPA

Universitas Sebelas Maret

Jl. Ir. Sutami No. 36 A Surakarta

widyaningdyah@student.uns.ac.id

Rini Anggrainingsih

Informatika, Fakultas MIPA

Universitas Sebelas Maret

Jl. Ir. Sutami No. 36 A Surakarta

rini.anggrainingsih@staff.uns.ac.id

Sari Widya Sihwi

Informatika, Fakultas MIPA

Universitas Sebelas Maret

Jl. Ir. Sutami No. 36 A Surakarta

sariwidya@staff.uns.ac.id

Abstract – *This paper reports development an admission new students application using TOPSIS (Technique Order Preference by Similarity To Ideal Solution) method for support selection process in Karanganyar Islamic State High School (MAN Karanganyar). This method was chosen because alternative ranking focuses on the shortest distance to the positive ideal solution and longest distance to the negative ideal solution.*

Software development model for this application used iterative method through stages of analysis, design, implementation and testing phases. Iterative development in this application performed with two iterations.

Testing of this application used McCall's software quality factors from the perspective of product operations that include correctness, reliability, efficiency, integrity, and usability factor. Testing on correctness factor showed that this application qualifies user requirement specification of MAN Karanganyar. Testing on reliability factor performed using Apache JMeter software and obtained the result that as many as 20 users did not experience a failure accessing applications simultaneously, whereas the "input weights" and the "calculation" pages could only be accessed by single user. Testing on efficiency factor performed using GTMetrix website, obtained grade A in the second iteration. Testing on integrity factor performed encryption on user password in the second iteration. Testing on usability factor used questionnaire of System Usability Scale (SUS) showed that the average value of SUS in the second iteration is acceptable category. Based on the testing result obtained from five operation factors, this application can be used to help in the selection of the new students at MAN Karanganyar.

Keywords: Admission, Iterative, Software Quality, TOPSIS

1. PENDAHULUAN

Pendaftar peserta didik baru di MAN Karanganyar pada tahun pelajaran 2015/2016 sebanyak 1.063 orang dari MTs (Madrasah Tsanawiyah) / Pondok Pesantren / SMP (Sekolah Menengah Pertama) Negeri Swasta. Daya tampung MAN Karanganyar tahun ini sebanyak 504 siswa [1]. Proses seleksi calon peserta didik baru di MAN Karanganyar memerlukan waktu yang lama karena terbatasnya jumlah panitia dan banyaknya jumlah

pendaftar. Selain itu terdapat peluang kesalahan dalam pengambilan keputusan sehingga diperlukan aplikasi yang dapat membantu memberikan keputusan peserta didik yang paling sesuai dengan persyaratan sekolah.

Sistem Pendukung Keputusan atau dikenal dengan SPK merupakan bagian dari sistem informasi yang berbasis komputer. Terdapat beberapa tahapan dalam sistem pendukung keputusan yaitu mendefinisikan masalah, pengumpulan data yang relevan dan sesuai, dan pengolahan data menjadi informasi [2]. Salah satu metode pengambilan keputusan yang sering digunakan adalah TOPSIS (*Technique Order Preference by Similarity To Ideal Solution*). Metode TOPSIS memiliki beberapa kelebihan diantaranya: (1) logikanya bersifat sederhana dan mudah dipahami, (2) proses perhitungannya mudah, (3) alternatif terbaik yang terpilih merupakan model matematika sederhana, (4) penilaian terpenting berada pada prosedur yang diperbandingkan [3].

Penelitian yang menggunakan TOPSIS diantaranya adalah seleksi calon siswa baru di SMA Negeri 3 Garut [4], seleksi penerimaan beasiswa di SMA Negeri 1 Parililitan [5], serta seleksi penerimaan peserta didik program akselerasi di SMP Negeri 1 Wonogiri yang membandingkan metode AHP, TOPSIS dan AHP-TOPSIS [6]. Berdasarkan penelitian yang terkait, TOPSIS yang digunakan sebagai metode pengambilan keputusan untuk menentukan peserta didik baru di MAN Karanganyar.

Model pengembangan aplikasi yang digunakan adalah *Iterative*. Model ini dipilih karena aplikasi yang dibuat membutuhkan pengerjaan waktu yang singkat dan dapat dikerjakan oleh tim yang kecil [7]. Penelitian terkait yang menggunakan model *iterative* salah satunya pada aplikasi mHealth, adanya "Fase I" pada model *iterative* dapat langsung ditemukan kelemahan dan fitur baru dari aplikasi yang akan diperbaiki pada "Fase II" sehingga pengembangan aplikasi dapat dilakukan lebih cepat [8]. Pengembangan *Iterative* pada aplikasi ini melalui tahap analisis, perancangan, implementasi dan tahap pengujian dengan dua kali iterasi.

Pengujian aplikasi dilakukan berdasarkan faktor kualitas perangkat lunak McCall dari segi *operation* yang meliputi faktor *correctness*, *reliability*, *efficiency*, *integrity*, dan *usability*. Tujuan penelitian ini adalah menghasilkan aplikasi yang layak digunakan berdasarkan hasil pengujian tersebut untuk membantu proses seleksi peserta didik baru di MAN Karanganyar.

2. SISTEM PENDUKUNG KEPUTUSAN

Sistem Pendukung Keputusan (SPK) merupakan sistem berbasis komputer interaktif yang membantu pengambil keputusan memanfaatkan data dan model untuk menyelesaikan suatu masalah [2]. SPK bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik.

Sistem Pendukung Keputusan memiliki empat komponen subsistem [2], yaitu:

1. Manajemen data, meliputi basis data yang berisi data-data yang relevan dengan keadaan dan dikelola oleh perangkat lunak yang disebut dengan Database Management System (DBMS).
2. Manajemen model, sebuah paket perangkat lunak yang berisi model-model finansial, statistik, management science, atau model kuantitatif, yang menyediakan kemampuan analisa dan perangkat lunak manajemen yang sesuai.
3. Antarmuka pengguna, merupakan subsistem yang dipakai oleh user untuk berkomunikasi dan memberi perintah.
4. Manajemen berbasis pengetahuan (knowledge-based), dapat mendukung subsistem lain atau sebagai komponen yang berdiri sendiri.

3. TECHNIQUE FOR ORDER PREFERENCE BY SIMILARITY TO IDEAL SOLUTION (TOPSIS)

Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) adalah metode pengambilan keputusan berdasarkan konsep bahwa alternatif terpilih harus memiliki jarak terpendek dari solusi ideal positif dan memiliki jarak terjauh dari solusi ideal negatif [9].

Prosedur dari metode TOPSIS mengikuti langkah-langkah sebagai berikut [9]:

1. Menentukan matriks keputusan yang ternormalisasi terbentuk dari persamaan 1 berikut:

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

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

2. Menghitung matriks keputusan yang ternormalisasi terbobot. Nilai bobot preferensi menunjukkan tingkat kepentingan relatif setiap kriteria atau subkriteria pada persamaan 2 berikut:

$$v_{ij} = w_j r_{ij} \quad (2)$$

Dengan w_j =bobot kriteria j ; $i=1,2,3,\dots,m$ dan $j=1,2,3,\dots,n$

3. Menentukan solusi ideal positif dan solusi ideal negatif. Solusi ideal positif dinotasikan dengan A^+ dan solusi ideal negatif dinotasikan dengan A^- yang ditentukan berdasarkan rating bobot ternormalisasi, seperti pada persamaan 3 dan persamaan 4 berikut:

$$a. \quad A^+ = \{(\max v_{ij} | j \in J), (\min v_{ij} | j \in J'), i = 1, 2, \dots, m\} = \{v_1^+, v_2^+, \dots, v_n^+\} \quad (3)$$

$$b. \quad A^- = \{(\min v_{ij} | j \in J), (\max v_{ij} | j \in J'), i = 1, 2, \dots, m\} = \{v_1^-, v_2^-, \dots, v_n^-\} \quad (4)$$

Keterangan:

$J = \{j = 1, 2, 3, \dots, n$ dan j merupakan himpunan kriteria keuntungan}

$J' = \{j = 1, 2, 3, \dots, n$ dan j merupakan himpunan kriteria biaya}

v_n^+ adalah elemen matriks solusi ideal positif

v_n^- adalah elemen matriks solusi ideal negatif

4. Menghitung Separasi. Separasi ini merupakan pengukuran jarak dari suatu alternatif ke solusi ideal positif dan solusi ideal negatif. Separasi untuk solusi ideal positif, seperti pada persamaan 5 berikut:

$$S_i^+ = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^+)^2} \quad (5)$$

Separasi untuk solusi ideal negatif, seperti pada persamaan (6) berikut:

$$S_i^- = \sqrt{\sum_{j=1}^n (v_{ij} - v_j^-)^2} \quad (6)$$

5. Menghitung kedekatan relatif dengan ideal positif seperti pada persamaan (7) berikut :

$$C_i^+ = \frac{S_i^-}{(S_i^+ + S_i^-)} \quad (7)$$

Dengan $0 \leq C_i^+ \leq 1$, dimana $C_i^+ = 0$ ketika $A_i = A^-$ dan $C_i^+ = 1$ ketika $A_i = A^+$.

6. Mengurutkan pilihan. Memilih alternatif dengan berdasarkan urutan C_i^+ dari nilai yang tertinggi. Maka dari itu, alternatif terbaik adalah salah satu yang berjarak terpendek terhadap solusi ideal dan berjarak terjauh dengan solusi ideal negatif.

4. SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

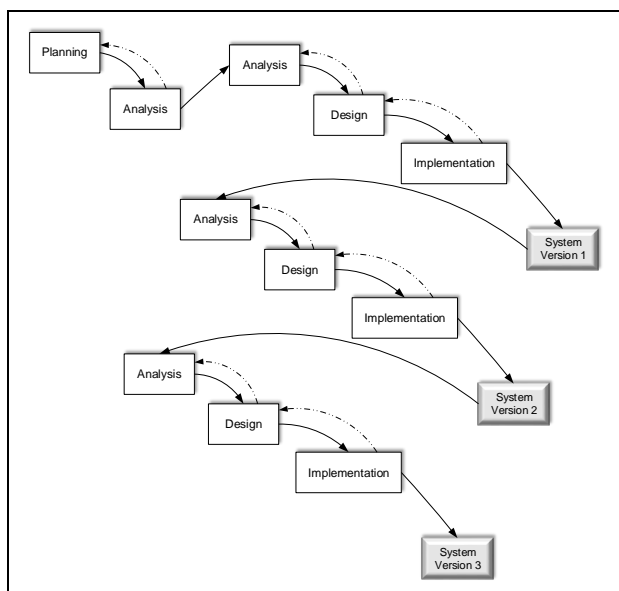
Software Development Life Cycle (SDLC) merupakan siklus pengembangan perangkat lunak yang terdiri dari empat fase fundamental [7], yaitu:

1. Perencanaan (*Planning*)
Fase perencanaan untuk mendefinisikan kenapa perangkat lunak harus dibuat dan menentukan bagaimana tim akan mengerjakan proyek tersebut.
2. Analisis (*Analysis*)
Fase analisis ini menjawab pertanyaan siapa yang akan menggunakan, apa yang akan dilakukan, dan kapan serta di mana perangkat lunak akan digunakan.
3. Perancangan (*Design*)
Di fase perancangan ini ditentukan bagaimana sistem akan beroperasi, berkenaan dengan perangkat lunak, perangkat keras, infrastruktur jaringan, antarmuka pengguna, *form, reports, database*, dll.
4. Implementasi (*Implementation*)
Di fase implementasi yaitu proses *delivery* kepada pengguna.

Beberapa metode pengembangan software diantaranya yaitu *waterfall, iterative, system prototyping, throwaway prototyping*, dan *agile* [7].

5. ITERATIVE DEVELOPMENT

Pengembangan *iterative* membagi keseluruhan proyek menjadi serangkaian versi yang dikembangkan secara berurutan. Persyaratan yang paling penting dan mendasar terdapat di dalam versi pertama dari sistem. Versi yang dikembangkan dengan proses *mini-waterfall*, dan sekali diimplementasikan, pengguna dapat memberikan umpan balik untuk dimasukkan ke dalam versi berikutnya dari sistem seperti yang ditunjukkan pada Gambar 1 [7].



Gambar 1. Pengembangan Software dengan metode *Iterative* [7]

Pada tahapan analisis dilakukan identifikasi kebutuhan pengguna. Kemudian tahapan perancangan dilakukan perancangan program, data, dan antarmuka. Selanjutnya pada tahapan implementasi dilakukan pengkodean aplikasi dan *testing* untuk menguji kualitas perangkat lunak.

6. SOFTWARE QUALITY ASSURANCE

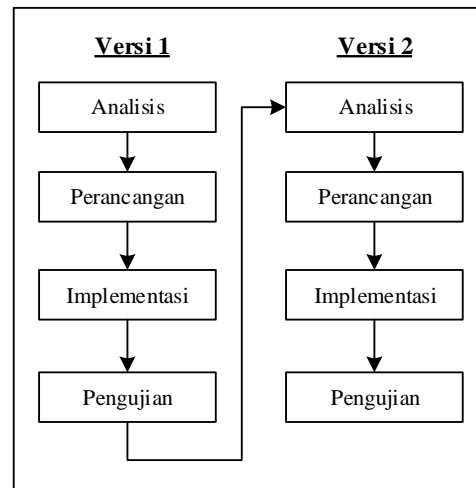
Kualitas perangkat lunak didapatkan melalui aktifitas pengujian perangkat lunak. Jika kualitas perangkat lunak buruk, maka sudah dipastikan *maintenance* sistem akan menjadi buruk atau tidak efektif. McCall mengemukakan faktor-faktor kualitas per menjadi tiga aspek penting, yaitu: (1) *Operations* terdiri dari faktor *correctness*, *reliability*, *efficiency*, *integrity*, *usability*; (2) *Revision* terdiri dari faktor *maintainability*, *flexibility*, *testability*; dan (3) *Transition* terdiri dari *portability*, *reusability* dan *interoperability* [10].

Pada aplikasi ini, pengujian kualitas dilakukan berdasarkan faktor kualitas perangkat lunak dari segi *operation*. Faktor kualitas perangkat lunak dari segi *operation* yaitu sebagai berikut:

1. *Correctness*
Correctness adalah sejauh mana suatu perangkat lunak memenuhi spesifikasi dan tujuan yang sudah ditetapkan sebelumnya.
2. *Reliability*
Reliability (Kehandalan) adalah kemampuan perangkat lunak untuk mempertahankan tingkat kinerja tertentu saat digunakan dalam kondisi tertentu.
3. *Efficiency*
Efficiency adalah kemampuan perangkat lunak dalam efisiensi penggunaan sumber daya dalam menjalankan fungsi-fungsinya.
4. *Integrity*
Integrity adalah berkenaan dengan keamanan sistem dari perangkat lunak, mencegah akses dari orang yang tidak berkepentingan.
5. *Usability*
Usability adalah kemudahan dalam penggunaan dan pemahaman oleh pengguna dari perangkat lunak tersebut.

7. METODOLOGI

Metode penelitian mengenai aplikasi seleksi penerimaan peserta didik baru di MAN Karanganyar dengan menggunakan metode TOPSIS dapat dibuat suatu alur kegiatan metode kerja *Iterative* seperti pada Gambar 2 sebagai berikut:



Gambar 2. Metodologi Iterative

7.1. Analisis

Tahapan ini dilakukan identifikasi kebutuhan pengguna sesuai dengan hasil wawancara dengan pihak sekolah dan algoritma perhitungan dengan menggunakan metode TOPSIS.

7.2. Perancangan

Pada tahapan perancangan sistem, terdapat *data design* dalam bentuk relasi database dan *UI Design* untuk menggambarkan rancangan *User Interface* dari aplikasi ini.

7.3. Implementasi

Tahap ini melakukan implementasi kebutuhan dari sistem ke dalam *code* dengan menggunakan bahasa pemrograman PHP.

7.4. Pengujian

Pengujian terhadap sistem yang telah dibangun berdasarkan faktor kualitas perangkat lunak oleh McCall dari segi *operation* yaitu *correctness*, *reliability*, *efficiency*, *integrity*, dan *usability*.

8. HASIL DAN PEMBAHASAN

8.1. Analisis

8.1.1. Analisis Kebutuhan Sistem

Pada tahap ini dilakukan analisis terhadap kebutuhan fungsional aplikasi seleksi penerimaan calon peserta didik baru. Kebutuhan fungsional pada aplikasi versi pertama yaitu sebagai berikut:

1. Panitia dapat login ke sistem
2. Panitia dapat melihat daftar nilai calon peserta didik yang akan diseleksi
3. Panitia dapat memasukkan detail calon peserta didik baru
4. Panitia dapat mengubah bobot yang digunakan untuk seleksi penerimaan peserta didik baru
5. Panitia dapat melihat hasil seleksi penerimaan peserta didik baru

Pada aplikasi versi kedua dilakukan penambahan yaitu sebagai berikut:

1. Menampilkan daftar calon peserta didik berdasarkan tahun ajaran (terdapat daftar calon peserta didik pada tahun sebelumnya)
2. Penambahan fungsi pencarian pada halaman hasil perangkaan berdasarkan pilihan kategori (nama, nomor pendaftaran, peringkat).

8.1.2. Proses Perhitungan

Proses seleksi penerimaan peserta didik baru menggunakan metode TOPSIS. Aplikasi versi pertama dan versi kedua tidak adanya perubahan yaitu menggunakan metode TOPSIS, sebagai berikut:

1. Membangun matriks keputusan yang berasal dari data masing-masing kriteria. Matriks keputusan ditunjukkan pada Tabel 1.

Tabel 1. Matriks Keputusan

No. Pendaftaran	Laporan Hasil Belajar Siswa	Nilai Ujian Nasional	Tes Tertulis	Tes Wawancara
1	78,67	42,00	40	56,00
2	74,00	49,25	40	70,00
3	82,17	59,25	45	80,68
4	83,09	57,50	40	86,59
5	84,99	63,75	35	91,30

2. Membuat matriks keputusan yang ternormalisasi yang terbentuk dari persamaan 1, hasil dari perhitungan ditunjukkan pada Tabel 2.

Tabel 2. Matriks Keputusan Ternormalisasi

no_pendaftar	K1	K2	K3	K4
1	0,069683	0,053833	0,060929	0,053546
2	0,065547	0,063126	0,060929	0,066932
3	0,072779	0,075944	0,068545	0,077144
4	0,073594	0,073701	0,060929	0,082795
5	0,075281	0,081712	0,053313	0,087298

3. Membuat matriks keputusan normalisasi terbobot dimana untuk mendapatkan nilai dari matriks dilakukan perkalian antara nilai bobot pada Tabel 3 dengan nilai pada masing-masing elemen matriks ternormalisasi, didapatkan hasil seperti pada Tabel 4.

Tabel 3. Nilai Bobot Kriteria

LHBS	SKHU	TPA	WWCR
0.2	0.25	0.25	0.3

Tabel 4. Matriks Keputusan Ternormalisasi Terbobot

no_pendaftar	K1	K2	K3	K4
1	0,013937	0,013458	0,015232	0,016064
2	0,013109	0,015782	0,015232	0,02008
3	0,014556	0,018986	0,017136	0,023143
4	0,014719	0,018425	0,015232	0,024838
5	0,015056	0,020428	0,013328	0,02619

4. Menghitung nilai solusi ideal positif dan solusi ideal negatif, sebelumnya harus dilakukan pengelompokan terlebih dahulu terhadap masing-masing kriteria termasuk dalam kriteria *cost* atau *benefit*. Keempat kriteria (laporan hasil belajar siswa, nilai ujian nasional, tes tertulis, tes wawancara) termasuk dalam *benefit*. Kemudian didapatkan solusi ideal positif dan negatif dengan menggunakan nilai yang berasal dari matriks normalisasi terbobot seperti pada Tabel 5.

Tabel 5. Solusi ideal positif dan negatif

	K1	K2	K3	K4
Y ⁺	0,0942	0,1298	0,1254	0,1571
Y ⁻	0,0820	0,0855	0,0975	0,0964

5. Menghitung separasi, menghitung jarak alternatif dari solusi ideal dengan persamaan 5 dan persamaan 6, hasil perhitungan separasi ditunjukkan pada Tabel 6.

Tabel 6. Tabel Separasi

no_pendaftar	Si ⁺	Si ⁻
1	0,0768	0,0149
2	0,0506	0,0315
3	0,0207	0,0624
4	0,0206	0,0638
5	0,0279	0,0762

6. Menghitung kedekatan terhadap solusi ideal positif dengan menggunakan persamaan 7, hasil ditunjukkan pada Tabel 7.

Tabel 7. Kedekatan terhadap solusi ideal positif

no_pendaftar	Ci
1	0,162165
2	0,38382
3	0,751102
4	0,755364
5	0,732167

7. Selanjutnya, dari hasil Tabel 7 dilakukan proses perankingan dengan mengurutkan nilai dari Ci terbesar ke terkecil. Hasil perankingan ditunjukkan Tabel 8.

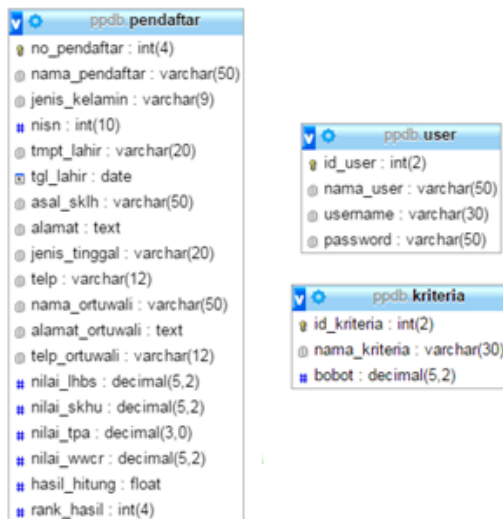
Tabel 8. Tabel Ranking

no_pendaftar	Ci	Ranking
4	0,755364	1
3	0,751102	2
5	0,732167	3
2	0,38382	4
1	0,162165	5

8.2. Perancangan

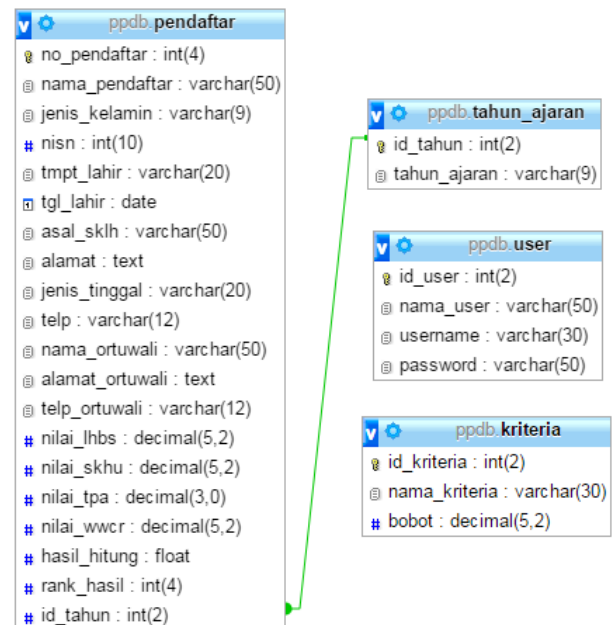
8.2.1. Data Design

Berdasarkan data hasil wawancara dilakukan desain relasi *database* pada aplikasi versi pertama yaitu pada Gambar 3 sebagai berikut:



Gambar 3. Relasi Database aplikasi versi pertama

Pada aplikasi versi kedua dilakukan penambahan kategori tahun ajaran pada aplikasi sehingga relasi *database* aplikasi versi kedua yaitu pada Gambar 4 sebagai berikut:



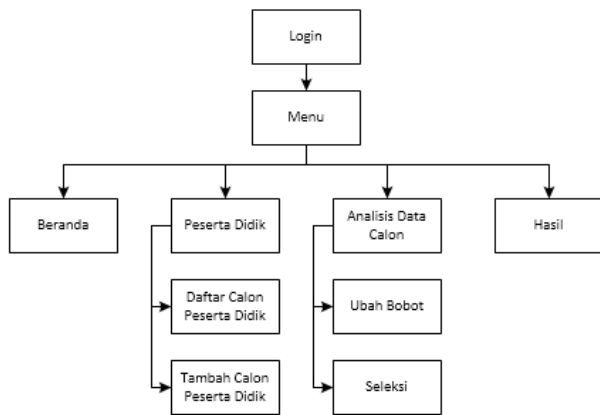
Gambar 4. Relasi Database aplikasi versi kedua

Pada aplikasi ini diasumsikan bahwa calon siswa hanya dapat mendaftar satu kali sehingga nilai-nilai setiap calon siswa hanya ada satu pada masing-masing nilai.

8.2.2. User Interface Design

User interface dirancang sesuai dengan struktur menu pada Gambar 5, yang terdiri dari:

1. Rancangan halaman *login*
Pada halaman *login* terdapat kotak *input username* dan *password* pengguna untuk masuk ke aplikasi ini.
2. Rancangan halaman menu beranda
Pada halaman menu beranda terdapat informasi seputar MAN Karanganyar.
3. Rancangan halaman menu peserta didik
Pada halaman menu peserta didik terdapat dua sub-menu, yaitu:
 - a. Halaman sub-menu daftar calon peserta didik
Pada halaman ini disajikan daftar calon peserta didik beserta nilai yang akan digunakan untuk penyeleksian.
 - b. Halaman sub-menu tambah calon peserta didik
Pada halaman ini merupakan halaman untuk menambahkan calon peserta didik beserta detailnya.
4. Rancangan halaman menu analisis data calon
Pada halaman menu menu analisis data calon terdapat dua sub-menu, yaitu:
 - a. Halaman ubah bobot
Pada halaman ini digunakan untuk mengubah bobot dari kriteria.
 - b. Halaman Seleksi / Hasil
Pada halaman ini digunakan untuk menjalankan perhitungan dan menampilkan hasil perankingan calon peserta didik dari hasil yang tertinggi.



Gambar 5. Struktur Menu Aplikasi Penyeleksian Calon Peserta Didik

Pada iterasi kedua, *user interface* pada iterasi pertama dilakukan penambahan sebagai berikut:

1. Pada halaman sub-menu daftar calon peserta didik ditambahkan pilihan daftar calon peserta didik berdasarkan tahun ajaran (terdapat daftar calon peserta didik pada tahun sebelumnya)
2. Pada halaman sub-menu daftar calon peserta didik ditambahkan fungsi pencarian pada halaman daftar calon peserta didik

8.3. Implementasi

Implementasi halaman seleksi calon peserta didik ditunjukkan pada Gambar 6 dan Gambar 7. Pada halaman ini ditampilkan daftar nama calon peserta didik berdasarkan urutan ranking terbesar ke terkecil dari hasil perhitungan TOPSIS.

Pada implementasi iterasi pertama, pengaksesan halaman lebih lama dibandingkan dengan implementasi iterasi kedua.

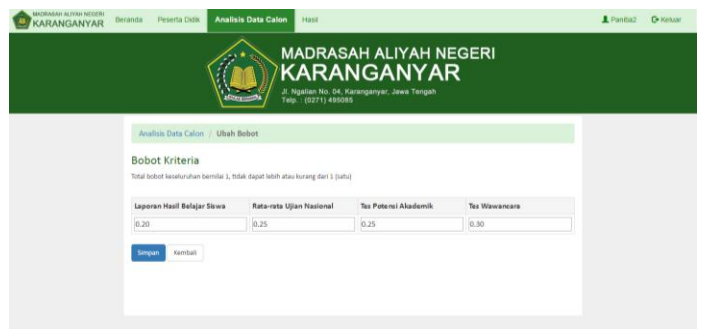


Gambar 7. Halaman Hasil Seleksi Aplikasi versi 2

Pada halaman Analisis Data Calon terdapat halaman untuk mengubah bobot kriteria untuk perhitungan TOPSIS. Halaman ubah bobot ditunjukkan pada Gambar 8 dan Gambar 9.



Gambar 8. Halaman ubah bobot kriteria pada aplikasi versi 1



Gambar 9. Halaman ubah bobot kriteria pada aplikasi versi 2

8.4. Pengujian

Pengujian aplikasi dilakukan berdasarkan faktor kualitas perangkat lunak dari segi *operation* yaitu sebagai berikut:

1. *Correctness*

Pengujian aspek *correctness* bertujuan untuk menguji pemenuhan spesifikasi dan tujuan yang ingin dicapai oleh pengguna. Pengujian dilakukan dengan membandingkan spesifikasi kebutuhan pengguna dengan aplikasi yang telah dibuat. Tiap kebutuhan pengguna dibandingkan dengan aplikasi telah memenuhi kebenaran sehingga menunjukkan bahwa aplikasi ini memenuhi pengujian *correctness* yang ditunjukkan pada Tabel 9. Pada aplikasi versi 1 dan versi 2, dihasilkan hasil yang sama.



Gambar 6. Halaman Hasil Seleksi Aplikasi versi 1

Tabel 9. Hasil Pengujian *Correctness*

Spesifikasi Kebutuhan Pengguna	Aplikasi Seleksi Peserta Didik	Output (Versi 1)	Output (Versi 2)
Dapat melihat daftar nilai calon peserta didik yang akan diseleksi	Menampilkan daftar nilai calon peserta didik yang akan diseleksi	Benar	Benar
Dapat memasukkan detail calon peserta didik baru	Mengisikan dan menambahkan detail calon peserta didik baru	Benar	Benar
Dapat merubah bobot yang digunakan untuk seleksi penerimaan peserta didik baru	Merubah bobot kriteria seleksi penerimaan peserta didik baru	Benar	Benar
Dapat melihat hasil seleksi penerimaan peserta didik baru	Menampilkan hasil seleksi penerimaan peserta didik baru	Benar	Benar

2. Reliability

Pengujian pada *reliability* bertujuan untuk mengetahui performa aplikasi dalam kondisi tertentu. Pengembangan aplikasi diperlukan pengujian performa dengan cara menjalankan aplikasi pada waktu yang bersamaan dan dilakukan oleh banyak pengguna. Pengujian dilakukan dengan menggunakan perangkat lunak *Apache JMeter*. Pada setiap halaman aplikasi diakses oleh 10 pengguna. Pada *Apache JMeter* akan menunjukkan persentase kegagalan akses dari pengguna yang ditunjukkan pada Tabel 10.

Tabel 10. Persentase kegagalan akses pengguna pada *Apache JMeter*

Halaman yang diakses	Kegagalan (Versi 1)	Kegagalan (Versi 2)
Halaman Login	0%	0%
Halaman Beranda	0%	0%
Halaman Calon peserta didik	0%	0%
Halaman Tambah calon peserta didik	0%	0%
Halaman Analisis peserta didik	0%	0%
Halaman Input Bobot	100%	100%
Halaman Perhitungan	100%	100%

Aplikasi versi 1 dan versi 2 didapatkan hasil yang sama. Pada halaman tambah calon peserta didik

dilakukan masukan dengan data yang berbeda tiap masukan pengguna. Pada halaman input bobot dan perhitungan terjadi kegagalan 100% karena oleh 20 pengguna secara bersamaan mengubah bobot. Sehingga hanya bisa dilakukan oleh seorang pengguna saja.

Menurut standar Telcordia mengenai aspek *reliability* bahwa apabila persentase sukses pada aplikasi mencapai lebih dari 95% maka dapat disimpulkan bahwa aplikasi yang dikembangkan telah memenuhi aspek *reliability* [11]. Semua halaman aplikasi ini, pengguna tidak mengalami kegagalan mengakses aplikasi sehingga didapatkan persentase sukses yaitu 100%.

3. Efficiency

Pengujian pada aspek *efficiency* bertujuan untuk mengukur efisiensi penggunaan sumber daya dalam menjalankan fungsi-fungsi aplikasi tersebut. Pengujian dilakukan dengan menghitung rata – rata *response time* yang digunakan untuk melakukan beberapa tugas pada aplikasi dengan menggunakan tools GTmetrix. *Response time* merupakan ukuran perkiraan waktu untuk melakukan tugas yang diberikan pengguna ke sistem. GTmetrix digunakan untuk menghitung *response time* dari pengaksesan aplikasi ini. Hasil GTmetrix dari aplikasi ini ditunjukkan pada Gambar 10 dan Gambar 11.

**Gambar 10.** Hasil pengujian dengan GTmetrix aplikasi versi 1**Gambar 11.** Hasil pengujian dengan GTmetrix aplikasi versi 2

Hasil pengujian pada aplikasi dengan menggunakan GTmetrix diketahui *response time* pada aplikasi versi 1 yaitu 1,5 detik dan pada aplikasi versi 2 yaitu 1,2 detik. Waktu yang dibutuhkan pengguna dalam menunggu respon aplikasi ditunjukkan pada Tabel 11 [12]. Aplikasi versi memiliki *response time* 1,2 detik sehingga berada pada rentang waktu respon 0-5 detik dengan *rating* “*High (Good)*”. Pada penilaian GTmetrix dari aspek PageSpeed aplikasi versi 1 adalah 54% dengan grade E dan aspek Yslow adalah 94% dengan grade A. Sedangkan pada aplikasi versi nilai aspek PageSpeed adalah 99% dengan grade A dan aspek Yslow adalah 97% dengan grade A. Sehingga terjadi peningkatan pada aplikasi versi 2 dibandingkan dengan versi 1.

Tabel 11. *Response Time* dan *Rating* dari Pengguna [12]

<i>Response Time</i>	<i>Rating</i>
0-5 sec	High (Good)
6 – 10 sec	Average
> 11 sec	Low (Poor)

4. Integrity

Pengujian pada aspek integrity bertujuan untuk menguji keamanan sistem dari perangkat lunak dan mencegah akses dari orang yang tidak berkepentingan. Pada aplikasi versi pertama, *password* pengguna hanya diamankan dengan kotak *input* bertipe *password*. Pada aplikasi versi kedua, aspek integrity dilakukan dengan penambahan enkripsi pada *password* pengguna. Enkripsi adalah proses menyandikan plaintexts menjadi ciphertexts (pesan yang tersandi) [13]. Enkripsi dilakukan dengan menggunakan MD5 dan dikalikan dengan kunci acak yang telah ditentukan. MD5 adalah fungsi *hash* satu arah yang menerima masukan berupa pesan dengan ukuran sembarang dan menghasilkan *message digest* yang panjangnya 128 bit [13]. Pada bahasa pemrograman PHP menyediakan perintah `md5()` untuk enkripsi dengan MD5. Enkripsi ini dilakukan agar akun pengguna lebih aman tidak dapat digunakan oleh orang yang tidak berkepentingan.

5. Usability

Pengujian pada aspek usability bertujuan untuk mengetahui bagaimana kemudahan dalam penggunaan dan pemahaman oleh pengguna dari aplikasi ini. Aspek *usability* dilakukan dengan menggunakan metode kuesioner kepada pengguna. Kuesioner dibagikan kepada panitia seleksi penerimaan peserta didik baru di MAN Karanganyar. Kuesioner yang digunakan mengacu pada kuesioner *System Usability Scale* (SUS) [14]. Skala penilaian pada kuesioner SUS ditunjukkan pada Tabel 12.

Tabel 12. Skala penilaian kuesioner SUS [14]

Jawaban	Skala Penilaian
Sangat tidak setuju	1
Tidak setuju	2
Ragu-ragu	3
Setuju	4
Sangat setuju	5

Untuk menghitung nilai pada kuesioner, SUS mempunyai aturan tersendiri. Daftar pernyataan kuesioner ditunjukkan pada Tabel 13. Untuk pernyataan nomor ganjil, nilai yang dijawab pada kuesioner dikurangi dengan satu. Sedangkan untuk pernyataan dengan nomor genap, angka lima dikurangi dengan nilai skor yang dijawab. Kemudian semua skor dijumlahkan selanjutnya dikalikan dengan angka 2,5. Nilai SUS memiliki range nilai 0 – 100.

Tabel 13. Kuesioner untuk pengujian *usability* [14]

No.	Pernyataan	Kode
1.	Saya akan sering menggunakan aplikasi ini	P1
2.	Aplikasi ini kompleks	P2
3.	Aplikasi ini mudah digunakan	P3
4.	Saya butuh bantuan untuk menggunakan aplikasi	P4
5.	Fungsi-fungsi dalam aplikasi ini dapat berjalan dengan baik	P5
6.	Banyak yang tidak konsisten dalam aplikasi ini	P6
7.	Aplikasi ini mudah dimengerti	P7
8.	Aplikasi ini tidak praktis	P8
9.	Saya sangat yakin dapat menggunakan aplikasi ini	P9
10.	Saya perlu belajar sebelum menggunakan aplikasi ini	P10

Kuesioner SUS yang telah didapatkan, nilai pada pernyataan 1, 3, 5, 7, dan 9 dikurangi dengan 1. Sedangkan nilai pada pernyataan 2, 4, 6, 8, dan 10 yaitu dengan 5 dikurangi dengan nilai dari pernyataan tersebut. Nilai pada tiap responden dijumlahkan kemudian dikalikan dengan 2,5. Pada pengujian ini terdapat 5 responden yaitu R1, R2, R3, R4, dan R5. Hasil kuisisioner ditunjukkan pada Tabel 14 untuk aplikasi versi 1 dan Tabel 15 untuk aplikasi versi 2.

Tabel 14. Hasil kuesioner SUS (versi 1)

Responden Pernyataan	Responden				
	R1	R2	R3	R4	R5
P1	3	3	2	3	1
P2	1	1	2	1	2
P3	3	3	3	3	2
P4	1	1	1	1	1
P5	2	2	2	2	2
P6	2	3	2	2	1
P7	2	3	3	3	2
P8	3	2	3	2	3
P9	2	2	3	3	2
P10	2	1	1	3	1
Nilai SUS	52,5	52,5	55	57,5	42,5

Nilai SUS yang diperoleh dari seluruh responden kemudian dihitung nilai rata-ratanya yaitu sebagai berikut :

$$\begin{aligned}
 \text{Rata - rata nilai SUS} &= \frac{\text{Jumlah nilai SUS}}{\text{Jumlah responden}} \\
 &= \frac{52,5 + 52,5 + 55 + 57,5 + 42,5}{5} \\
 &= \frac{260}{5} = 52
 \end{aligned}$$

Tabel 15. Hasil kuesioner SUS (versi 2)

Responden Pernyataan	R1	R2	R3	R4	R5
P1	4	4	3	3	4
P2	4	3	3	4	3
P3	3	4	3	3	4
P4	3	3	3	3	3
P5	4	3	3	4	4
P6	3	3	3	3	3
P7	4	4	4	3	3
P8	4	3	3	4	4
P9	3	3	3	4	3
P10	3	3	4	3	3
Nilai SUS	87,5	82,5	80	85	85

Nilai SUS yang diperoleh dari seluruh responden kemudian dihitung nilai rata-ratanya yaitu sebagai berikut :

$$\begin{aligned}
 \text{Rata - rata nilai SUS} &= \frac{\text{Jumlah nilai SUS}}{\text{Jumlah responden}} \\
 &= \frac{87,5 + 82,5 + 80 + 85 + 85}{5} \\
 &= \frac{420}{5} \\
 &= 84
 \end{aligned}$$

Rata-rata nilai SUS dibandingkan dengan *range* nilai yang ditunjukkan pada Tabel 16. *Range* nilai digunakan untuk membantu dalam menentukan apakah Nilai SUS yang diperoleh menunjukkan suatu aplikasi dapat diterima baik atau tidak dari segi *usability* [15].

Tabel 16. Rentang penilaian SUS [15]

Nilai SUS	Interpretasi
< 50	<i>Not Acceptable</i>
50 – 70	<i>Marginal</i>
> 70	<i>Acceptable</i>

Rata-rata nilai SUS pada aplikasi versi 1 adalah 52 dan termasuk pada kategori *marginal*. Sedangkan rata-rata nilai SUS pada aplikasi versi 2 adalah 84 dan termasuk pada kategori *acceptable*, hal ini menunjukkan bahwa aplikasi versi 2 dapat diterima oleh pengguna dengan baik ditinjau dari aspek *usability* dibandingkan dengan aplikasi versi 1.

9. KESIMPULAN

Pada Penelitian ini telah dibangun aplikasi penerimaan peserta didik baru di MAN Karanganyar dengan metode TOPSIS dan model pengembangan sistem menggunakan *Iterative* dengan melalui tahap analisis, perancangan, implementasi dan tahap pengujian. Pengembangan *Iterative* pada aplikasi ini dilakukan dengan dua kali iterasi.

Pengujian aplikasi dilakukan berdasarkan faktor kualitas perangkat lunak menurut McCall dari segi *operation*. Pengujian pada faktor *correctness* diperoleh hasil bahwa aplikasi telah memenuhi spesifikasi kebutuhan pengguna di MAN Karanganyar. Pengujian pada faktor *reliability* dilakukan dengan menggunakan perangkat lunak Apache Jmeter dan diperoleh hasil bahwa sebanyak 20 pengguna tidak mengalami kegagalan mengakses aplikasi secara bersamaan, sedangkan khusus untuk halaman input bobot hanya dapat diakses oleh satu pengguna untuk mengubah bobot kriteria. Pengujian pada faktor *efficiency* yang dilakukan dengan mengevaluasi performa menggunakan GTmetrix, memperoleh grade E pada iterasi pertama dan mengalami peningkatan pada iterasi kedua menjadi grade A. Pengujian pada faktor *integrity* dilakukan penambahan enkripsi pada password pengguna di iterasi kedua karena pada iterasi pertama password pengguna hanya diamankan dengan kotak *input* bertipe *password*. Pengujian pada faktor *usability* dilakukan dengan menggunakan kuisisioner *System Usability Scale* (SUS) yang menunjukkan bahwa rata-rata nilai SUS pada iterasi pertama termasuk dalam katagori *marginal* dan iterasi kedua termasuk dalam kategori *acceptable*. Berdasarkan hasil pengujian lima faktor *operation* yang diperoleh, diketahui bahwa aplikasi ini layak digunakan untuk membantu proses seleksi peserta didik baru di MAN Karanganyar.

10. DAFTAR PUSTAKA

- [1] Yono, "PPDB MAN Karanganyar Membludak," 15 Juni 2015. [Online]. Available: <http://jateng.kemendiknas.go.id/index.php?a=berita&id=268225>. [Diakses 3 Agustus 2015].
- [2] E. Turban, R. Sharda dan D. Delen, *Decision Support and Business Intelligence Systems 9th edition*, New Jersey: Pearson Education Inc., 2011.
- [3] R. P. Perdana dan E. Yuliaty, "Integrasi Metode FMEA dan TOPSIS Untuk Menganalisis Risiko Kecelakaan Pada Proses Frame And Fork Welding," *Spektrum Industri Vol. 12 No. 1*, pp. 43-52, 2014.
- [4] A. H. Rustiawan, D. Destiani dan A. Ikhwana, "Sistem Pendukung Keputusan Penyeleksian Calon Siswa Baru di SMA N 3 Garut," *Jurnal Algoritma Sekolah Tinggi Teknologi Garut ISSN : 2302-7339 Vol. 09 No. 20*, 2012.
- [5] F. Sihotang, "Sistem Pendukung Keputusan Penerima Beasiswa Dengan Metode TOPSIS (Studi Kasus: SMA Negeri 1 Parlilitan)," *Pelita*

Informatika Budi Utama, vol. 5, no. 3, pp. 6-11, 2013.

Journal of Behavioral Nutrition and Physical Activity, 2012.

- [6] E. N. S. Purnomo, "Analisis Perbandingan Menggunakan Metode AHP, TOPSIS, dan AHP-TOPSIS dalam Studi Kasus Sistem Pendukung Keputusan Penerimaan Siswa Akselerasi," *Skripsi pada FMIPA UNS Surakarta*, 2013.
- [7] A. Dennis, B. H. Wixom dan R. M. Roth, *System Analysis and Design* 5th edition, New Jersey: John Wiley & Sons, Inc, 2012.
- [8] M. A. Jacobs dan A. L. Graham, "Iterative development and evaluation methods of mHealth behavior change interventions," *Current Opinion in Psychology*, vol. 9, p. 33–37, 2016.
- [9] J. Lu, G. Z. D. Ruan dan F. Wu, *Multi-Objective Group Decision Making - Series in Electrical and Computer Engineering Vol. 6*, Singapore: Imperial College Press, 2007.
- [10] D. Galin, *Software Quality Assurance: From theory to Implementation*, London: Pearson Education, 2004.
- [11] A. Asthana dan J. Olivieri, "Quantifying Software Reliability and Readiness," *IEEE International Workshop Technical Committee on Communications Quality and Reliability*, pp. 1-6, 2009.
- [12] A. Bouch, A. Kuchinsky dan N. Bhatti, "Quality is in the Eye of the Beholder: Meeting Users' Requirements for Internet Quality of Service," *Internet Systems and Applications of Hewlett Packard (HP)*, pp. 297-304, 2000.
- [13] R. Munir, *Kriptografi*, Bandung: Penerbit Informatika, 2006.
- [14] J. Brooke, "A Quick and Dirty Usability Scale," *International Journal of Human Computer Interaction*, pp. 4-5, 1996.
- [15] A. Bangor, P. Kortum dan J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective," *Journal of Usability Studies*, pp. 114-123, 2009.
- [16] B. S. Fjeldsoe, Y. D. Miller, J. L. O'Brien dan A. L. Marshall, "Iterative development of MobileMums: a physical activity intervention for women with young children," *International*