

**ANALISIS DAN PERANCANGAN APLIKASI PEMILIHAN  
JENIS BEASISWA MENGGUNAKAN METODE  
ANALYTICAL HIERARCHY PROCESS  
(STUDI KASUS: BEASISWA UKRIDA)**

***ANALYSIS AND DESIGN APPLICATION OF SCHOLARSHIP  
TYPE USING AHP METHOD (STUDY CASE: UKRIDA  
SCHOLARSHIP)***

Heru Widjaya<sup>1</sup>, Cynthia Hayat<sup>2</sup>

Fakultas Teknik dan Ilmu Komputer Jurusan Sistem Informasi  
Universitas Kristen Krida Wacana

<sup>1</sup>heru\_widjaya21@yahoo.com, <sup>2</sup>cynthia.hayat@ukrida.ac.id

**Abstrak**

Banyaknya jenis beasiswa yang ditawarkan membuat mahasiswa harus dapat memilih dengan tepat sesuai kebutuhannya. Oleh karena itu, dibutuhkan sebuah metode yang dapat membantu pengambilan keputusan jenis beasiswa yang tepat. Metode AHP dipilih karena mempunyai kemampuan untuk mendapatkan hasil dengan berbagai alternatif. Dalam AHP, hasil penghitungan didapat dari hasil perkalian tiap nilai *eigen vector*. Penerapan metode AHP dalam penelitian ini berbasis aplikasi menggunakan *Visual Basic 2008*, dimulai dari tahap analisis sistem, perancangan *user interface*, dan tahap evaluasi pada aplikasi. Hasil dari aplikasi menjelaskan tentang nilai bobot terhadap alternatif yang ada untuk membantu mendapatkan keputusan jenis beasiswa. Nilai bobot alternatif yang dapat diambil berdasarkan nilai yang tertinggi dari satu nilai terhadap nilai yang lain.

**Kata kunci:** beasiswa, AHP, alternatif keputusan

**Abstract**

*The different types of scholarship program offered by the university require students to make the right decision before applying for the program. A method to help students with the decision making is therefore required. The AHP method is chosen because of its capacity to offer various alternatives. In AHP calculation result is obtained by multiplying the value of each eigenvector. The implementation of AHP in this study is based on an application using Visual Basic 2008, including the stages of system analysis, user interface design, and application evaluation. The results of the application demonstrate the weights of the alternatives that can help to choose the scholarship type. The decision can be made based on the highest value of the available alternatives.*

**Keywords:** scholarship, AHP, alternative

**Tanggal Terima Naskah : 18 September 2014**

**Tanggal Persetujuan Naskah : 24 September 2014**

## 1. PENDAHULUAN

### 1.1 Latar Belakang

Beasiswa merupakan pembiayaan yang tidak bersumber dari pendanaan sendiri, melainkan berasal dari swasta, pemerintah, individu maupun kelompok, dan universitas. Beasiswa tersebut diberikan kepada yang berhak menerima berdasarkan kriteria yang berlaku [1].

Demikian juga dengan Universitas Krida Wacana yang telah memiliki program pemberian beasiswa. Calon penerima beasiswa akan melalui tahap-tahap yang berlaku sebelum beasiswa diberikan kepada mahasiswa tersebut. Tahapan ini salah satunya berupa tahap rapat yang dilakukan oleh pengurus beasiswa dengan yayasan. Namun, sering kali terjadi perbedaan pendapat dalam penentuan penerima beasiswa, sehingga mengakibatkan ketidakpastian dalam pengambilan keputusan.

Untuk itu, diperlukan suatu metode yang dapat dijadikan sebagai *second opinion* dalam proses pengambilan keputusan. *Analytical Hierarchy Process* (AHP) digunakan karena *input* utamanya bersumber pada persepsi manusia, dalam hal ini adalah pihak yang ahli dalam masalah beasiswa atau pihak yang mengerti permasalahan beasiswa. Metode AHP juga mempunyai kemampuan untuk mendapatkan hasil dari masalah semi terstruktur dan tidak terstruktur, sehingga model ini merupakan cara yang komprehensif.

### 1.2 Rumusan Masalah

Bersadarkan latar belakang tersebut, maka permasalahan yang akan dibahas adalah “Bagaimana merancang aplikasi pemilihan jenis beasiswa dengan penerapan metode *Analytical Hierarchy Process* untuk membantu membuat keputusan?”.

### 1.3 Tujuan dan Manfaat

Adapun tujuan dari penelitian ini adalah membuat sebuah aplikasi yang hasilnya dapat dijadikan sebagai metode alternatif dalam pengambilan keputusan, namun hal tersebut hanya sebagai *second opinion* pengambilan keputusan pemilihan jenis beasiswa.

Adapun manfaat dari penelitian ini adalah untuk mempercepat dan mempermudah proses pengambilan keputusan penerimaan jenis beasiswa, karena AHP memiliki kelebihan yang dapat menghasilkan pengukuran yang dinilai dengan skala angka.

### 1.4 Ruang Lingkup

Ruang lingkup penelitian ini adalah sebagai berikut:

1. Kriteria yang diberikan diperoleh berdasarkan aturan yang berlaku, yakni semester, Indeks Prestasi Kumulatif, dan penghasilan orang tua.
2. Jenis beasiswa yang digunakan adalah Beasiswa Bantuan Belajar Mahasiswa dan Beasiswa Bank INA.
3. Aplikasi yang akan dibangun dibuat dengan menggunakan *software Visual Basic* 2008, karena versi 2008 masih banyak digunakan oleh masyarakat.

## 2. KONSEP DASAR

### 2.1 Beasiswa

Adapun beberapa jenis dan ketentuan beasiswa yang ada, diantaranya [2]:

Tabel 1. Jenis beasiswa, syarat dan ketentuan.

Beasiswa	Syarat dan Ketentuan
1. <b>Beasiswa Bantuan Belajar Mahasiswa</b>	1. Mahasiswa yang paling tidak mampu. 2. Mahasiswa yang paling mendekati lulus. 3. Minimal IPK >2.50
2. <b>Beasiswa Bank INA</b>	1. Berkebangsaan Indonesia. 2. Mahasiswa tidak mampu. 3. Tidak menerima beasiswa dari pihak lain. 4. IPK minimal $\geq 3.00$

### 2.2 Analytical Hierarchy Process (AHP)

Metode *Analytic Hierarchy Process* (AHP) ini dikembangkan oleh Thomas L. Saaty berguna untuk membantu pengambil keputusan sehingga mendapatkan keputusan terbaik dengan membandingkan faktor-faktor yang berupa kriteria [3]. Dengan AHP, seseorang dapat mengatur pendapat dan intuisi dengan cara logika menggunakan hirarki dan memasukkan penilaian berdasarkan pengertian dan pengalaman. Perbedaan antara model AHP dengan model pengambilan keputusan lainnya terletak pada jenis *input*-nya. Model yang lain umumnya menggunakan *input* yang kuantitatif atau berasal dari data sekunder sehingga model tersebut hanya dapat mengolah hal-hal kuantitatif pula. Karena AHP menggunakan *input* yang kualitatif (persepsi manusia), maka model ini dapat juga mengolah hal-hal kualitatif di samping hal-hal yang kuantitatif.

### 2.3 Prinsip Dasar Analytic Hierarchy Process (AHP)

Dalam menyelesaikan persoalan dengan metode AHP terdapat beberapa prinsip dasar yang harus dipahami, diantaranya:

#### 1. *Decomposition*

Pengertian *decomposition* adalah memecahkan atau membagi persoalan yang utuh menjadi unsur – unsurnya ke bentuk hirarki proses pengambilan keputusan, dimana setiap unsur atau elemen saling berhubungan. Hirarki masalah disusun untuk membantu proses pengambilan keputusan dengan memperhatikan seluruh elemen keputusan yang terlibat dalam sistem. Sebagian besar masalah menjadi sulit untuk diselesaikan karena proses pemecahannya dilakukan tanpa memandang masalah sebagai suatu sistem dengan suatu struktur tertentu.

#### 2. *Comparative Judgement*

*Comparative judgement* dilakukan dengan penilaian tentang kepentingan relatif dua elemen pada suatu tingkat tertentu dalam kaitannya dengan tingkatan di atasnya. Penilaian ini merupakan inti dari AHP karena akan berpengaruh terhadap urutan prioritas dari elemen – elemennya. Hasil dari penilaian ini lebih mudah disajikan dalam bentuk matriks *pairwise comparisons*, yaitu matriks perbandingan berpasangan memuat tingkat preferensi beberapa alternatif untuk tiap kriteria. Skala preferensi yang digunakan, yaitu skala 1 yang menunjukkan tingkat yang paling rendah dan sampai dengan skala 9 yang menunjukkan tingkat yang paling tinggi.

#### 3. *Synthesis of Priority*

*Synthesis of priority* dilakukan dengan menggunakan *eigen vector method* untuk mendapatkan bobot relatif bagi unsur – unsur pengambilan keputusan.

4. *Logical Consistency*

*Logical consistency* merupakan karakteristik penting AHP. Hal ini dicapai dengan mengagresikan seluruh *eigen vector* yang diperoleh dari berbagai tingkatan hirarki dan selanjutnya diperoleh suatu vektor *composite* tertimbang yang menghasilkan urutan pengambilan keputusan.

Tahapan – tahapan pengambilan keputusan dalam metode AHP adalah sebagai berikut:

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan
2. Membuat struktur hirarki yang diawali dengan tujuan umum, dilanjutkan dengan kriteria-kriteria dan alternatif - alternatif pilihan yang ingin di *ranking*.
3. Membentuk matriks perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap masing-masing tujuan atau kriteria yang setingkat di atas. Perbandingan dilakukan berdasarkan pilihan atau *judgement* dari pembuat keputusan dengan menilai tingkat-tingkat kepentingan suatu elemen dibandingkan elemen lainnya.

Intensitas Kepentingan

1 = Kedua elemen sama pentingnya.

3 = Elemen yang satu sedikit lebih penting daripada elemen yang lainnya.

5 = Elemen lebih penting daripada yang lainnya.

7 = Satu elemen jauh lebih penting daripada elemen lainnya

9 = Satu elemen jelas lebih penting daripada elemen lainnya

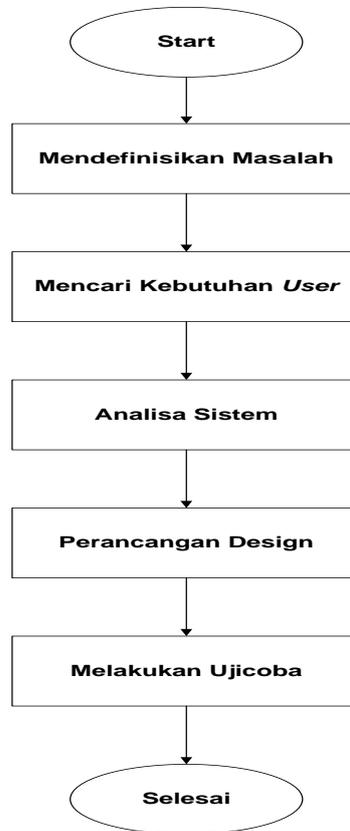
2,4,6,8 = Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan, nilai ini diberikan bila ada dua kompromi di antara 2 pilihan.

4. Menormalkan data, yaitu dengan membagi nilai dari setiap elemen di dalam matriks yang berpasangan dengan nilai total dari setiap kolom.
5. Menghitung nilai *eigen vector* dan menguji konsistensinya, jika tidak konsisten maka pengambilan data (preferensi) perlu diulangi. Nilai *eigen vector* yang dimaksud adalah nilai *eigen vector* maksimum yang diperoleh dengan menggunakan *microsoft excel* maupun dengan *manual*.
6. Mengulangi langkah, 3, 4, dan 5 untuk seluruh tingkat hirarki.
7. Menghitung *eigen vector* dari setiap matriks perbandingan berpasangan. Nilai *eigen vector* merupakan bobot setiap elemen. Langkah ini untuk mensintetis pilihan dalam penentuan prioritas elemen pada tingkat hirarki terendah sampai pencapaian tujuan.
8. Menguji konsistensi hirarki. Jika tidak memenuhi dengan  $CR < 0,100$  maka penilaian harus diulangi kembali.

### 3. METODE PENELITIAN

#### 3.1 Langkah-langkah Penelitian

Berikut adalah *flowchart* penelitian:



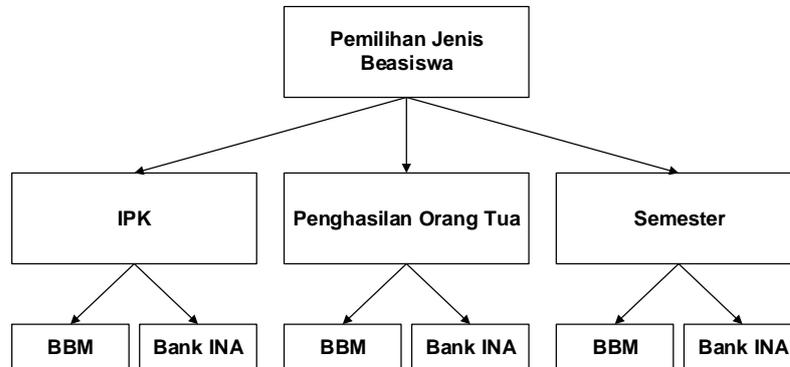
Gambar 1. Langkah-langkah penelitian

Keterangan:

1. Mendefinisikan Masalah: Pada tahap ini akan didefinisikan masalah yang ada dan mencari data atau informasi yang diperlukan.
2. Mencari Kebutuhan *User*: Pada tahap ini akan dilakukan pemahaman sistem, baik dari hasil studi pendahulu maupun dari kebutuhan informasi *user*, serta mengidentifikasi masalah untuk segera mendapatkan data dan menganalisisnya.
3. Analisis Sistem: Pada tahap ini akan dilakukan evaluasi permasalahan atau hambatan yang terjadi pada sistem atau proses yang telah ada dengan membuat hirarki menu, *state transition diagram*, UML (*use case*, *use case description*, *activity diagram*).
4. Perancangan: Pada tahap ini akan dibuat gambaran umum mengenai perancangan sistem, seperti *design user interface*, *design* hirarki tujuan, kriteria dan alternatif pada aplikasi yang akan dibuat.
5. Melakukan Ujicoba: Pada tahap ini akan dilakukan implementasi atau ujicoba terhadap sistem atau aplikasi yang telah dibangun, untuk mengetahui apakah sistem atau aplikasi berjalan dengan baik sesuai dengan hasil yang diharapkan.

### 3.2 Hirarki Prosedur AHP

Berikut adalah hirarki prosedur AHP yang digunakan.



Gambar 2. Hirarki prosedur AHP

Keterangan:

1. Pemilihan Jenis Beasiswa: Merupakan tujuan utama dari hasil *input* dan penghitungan yang dilakukan.
2. IPK: Indeks Prestasi Kumulatif, merupakan salah satu kriteria yang dibutuhkan untuk mendapatkan sebuah alternatif keputusan.
3. Penghasilan Orang Tua: Merupakan salah satu kriteria yang dibutuhkan untuk mendapatkan sebuah alternatif keputusan.
4. Semester: Merupakan salah satu kriteria yang dibutuhkan untuk mendapatkan sebuah alternatif keputusan.
5. Bank INA: Merupakan salah satu jenis beasiswa yang digunakan dan juga sebagai hasil alternatif keputusan.
6. BBM: Beasiswa Bantuan Belajar Mahasiswa, merupakan salah satu jenis beasiswa yang digunakan dan juga sebagai hasil alternatif keputusan.

### 3.3 Prosedur AHP Berdasarkan Kasus

Berikut ini adalah prosedur AHP yang dilakukan berdasarkan kasus yang digunakan:

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, kemudian menyusun hirarki dari permasalahan yang dihadapi.
2. Menentukan prioritas kriteria, pada langkah ini terdapat tiga kriteria yang digunakan, yaitu IPK, Penghasilan Orang Tua, dan Semester. Berikut ini adalah prioritas yang digunakan pada masing-masing kriteria.

Tabel 2. Prioritas IPK

IPK	Prioritas
2.50 – 2.59	1
2.60 – 2.79	2
2.80 – 2.99	3
3.00 – 3.19	4
3.20 – 3.39	5
3.40 – 3.59	6
3.60 – 3.79	7
3.80 – 3.99	8
4.00	9

Tabel 3. Prioritas penghasilan orang tua

Penghasilan Orang Tua	Prioritas
5.000.000 – 5.499.999	1
4.500.000 – 4.999.999	2
4.000.000 – 4.499.999	3
3.500.000 – 3.999.999	4
3.000.000 – 3.499.999	5
2.500.000 – 2.999.999	6
2.000.000 – 2.499.999	7
1.500.000 – 1.999.999	8
1.000.000 – 1.499.999	9

Tabel 4. Prioritas semester

Semester	Prioritas
1	1
2	3
3	5
4	7
5	8
6	9

3. Menghitung Normalisasi Kriteria, setelah melakukan *input* kriteria maka aplikasi akan menghitung normalisasi yang didapat dari:
  - a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
  - b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
  - c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.
4. Mencari nilai *eigen vector* 1 (Kriteria), untuk mendapatkan nilai *eigen vector* ini, hasil jumlah dari normalisasi kriteria tiap elemen dibagi dengan jumlah kriteria yang digunakan. Pada penulisan ini digunakan tiga kriteria (IPK, Penghasilan Orang Tua, dan Semester). Hal yang sama terjadi dalam mencari *eigen vector* 2 (Alternatif), namun alternatif yang digunakan adalah dua, oleh karena itu jumlah dari normalisasi alternatif dibagi dengan dua.
5. Mencari Konsistensi, cara yang digunakan untuk mendapatkannya ialah hasil dari *eigen vector* 2 (alternatif) dikalikan dengan *eigen vector* 1 (kriteria). Setelah itu akan muncul bobot prioritas dari alternatif yang dapat digunakan sebagai pengambilan keputusan jenis beasiswa yang sesuai.
6. Mencari  $\lambda$  Maks, didapat dari hasil prioritas yang dimasukkan pada data mahasiswa dikalikan dengan nilai *eigen vector* 1 (kriteria) tiap elemen.
7. Mencari *Consistency Index* (CI) dengan rumus:

$$CI = \frac{(\lambda_{maks} - n)}{n}$$

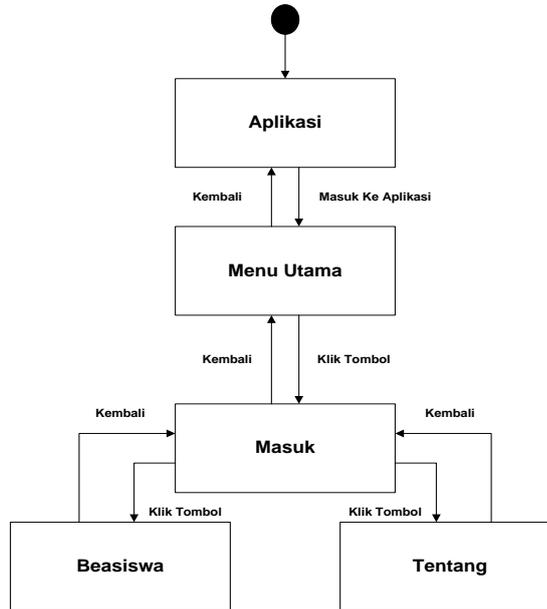
8. Menghitung Rasio Konsistensi atau *Consistency Ratio* (CR) dengan rumus:

$$CR = \frac{CI}{RI}$$

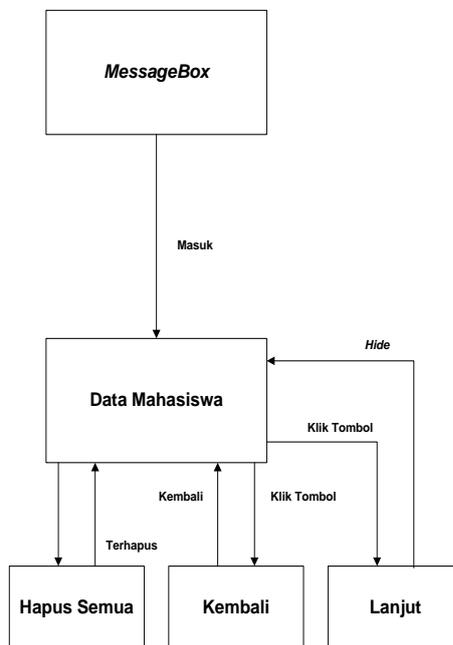
9. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10%, maka penilaian data *judgment* harus diperbaiki. Namun jika Rasio Konsistensi  $\frac{CI}{RI}$  kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar.

### 3.4 State Transition Diagram

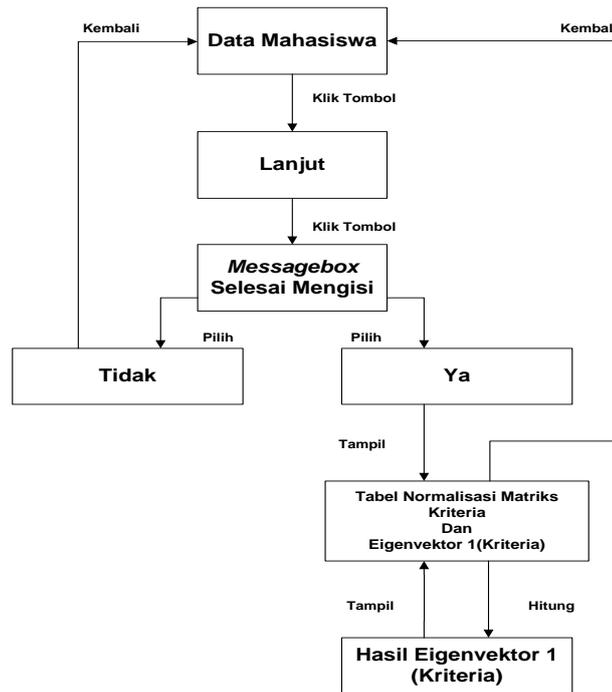
Berikut ini adalah *State Transition Diagram* yang digunakan.



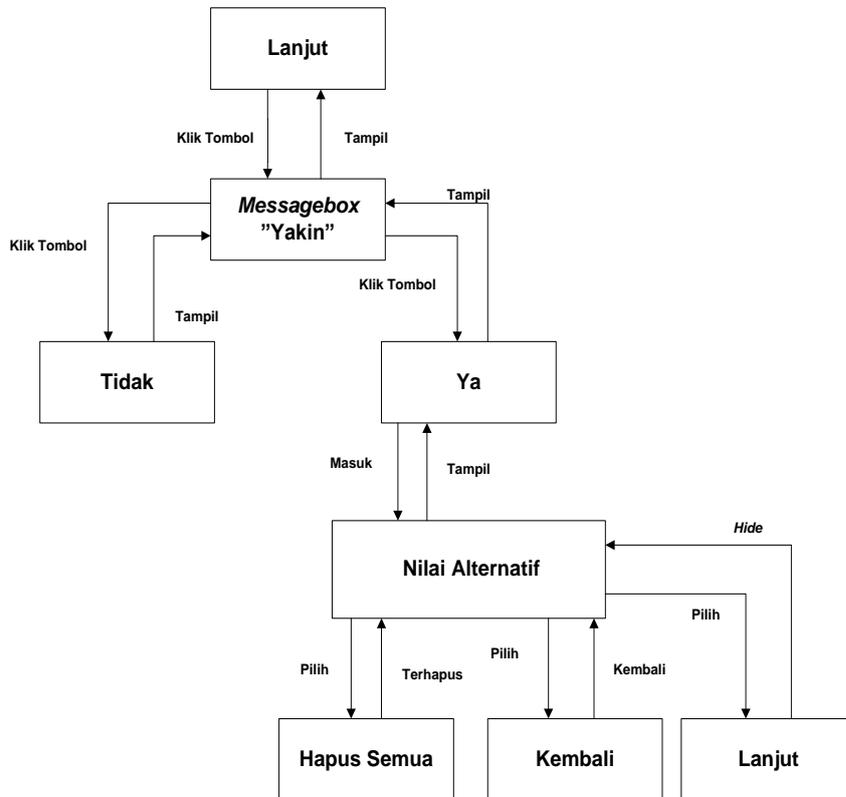
Gambar 3. *State transition diagram* menu utama



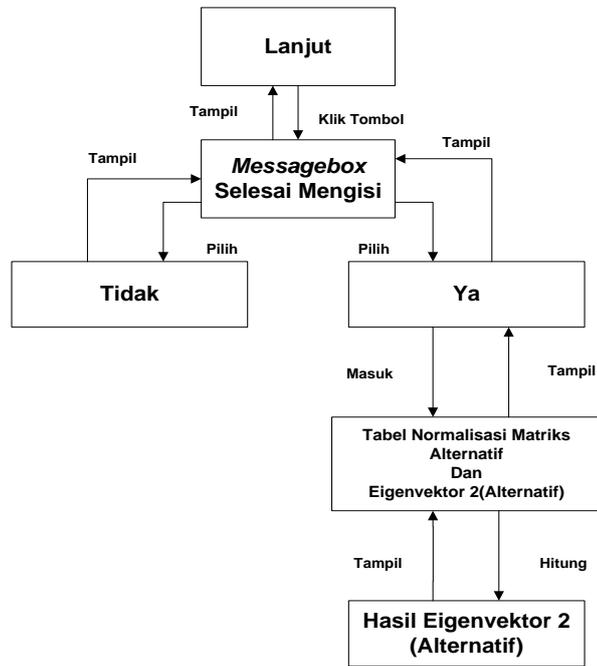
Gambar 4. *State transition diagram* data mahasiswa



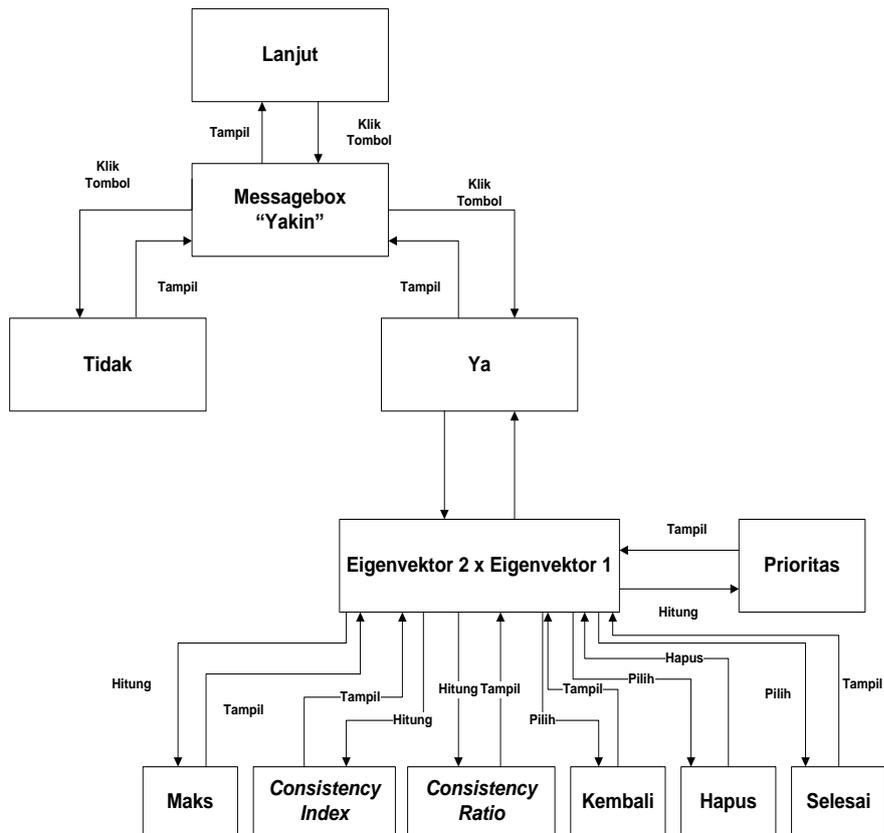
Gambar 5. State transition diagram tabel normalisasi matriks kriteria dan eigen vector 1 (kriteria)



Gambar 6. State transition diagram nilai alternatif



Gambar 7. State transition diagram tabel normalisasi matrik alternatif dan eigen vector 2 (alternatif)

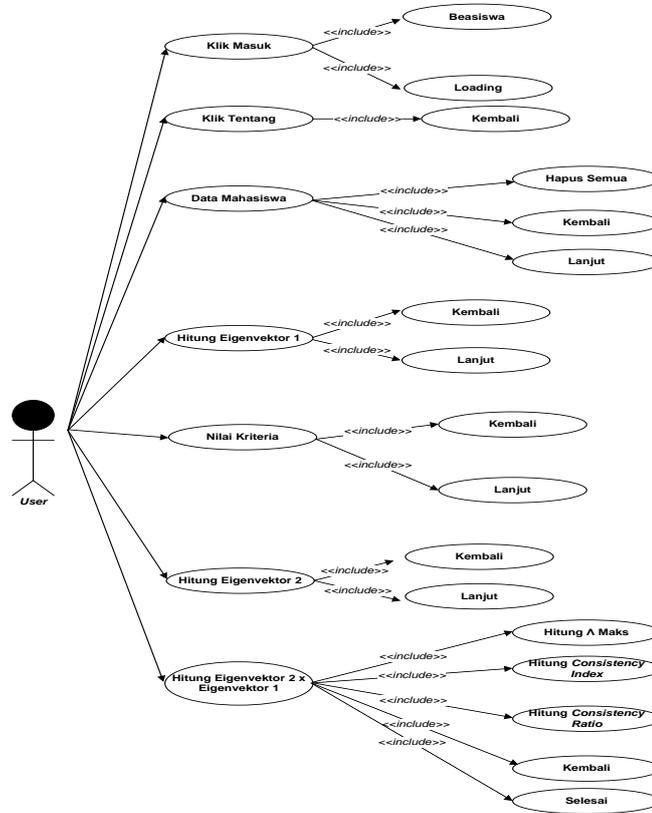


Gambar 8. State transition diagram eigen vector 2 x eigen vector 1

#### 4. ANALISIS DAN PEMBAHASAN

##### 4.1 Use Case

Berikut ini adalah *Use Case* yang digunakan.



Gambar 9. *Use case diagram user*

##### 4.2 Use Case Description

Merupakan salah satu *Use Case Description* yang digunakan.

Tabel 5. *Use case description data mahasiswa*

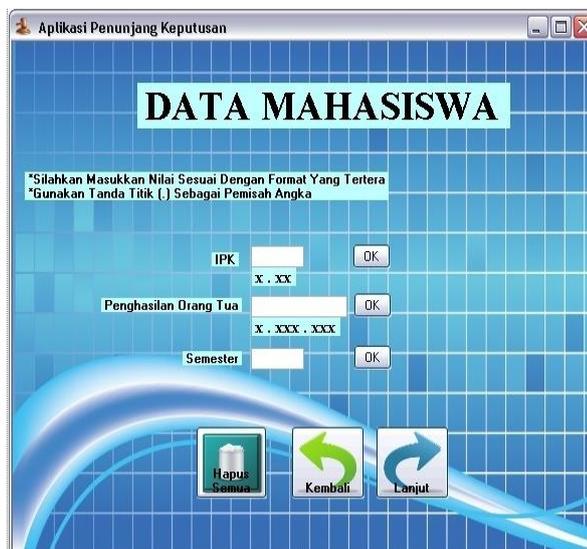
Nama Use Case	Data Mahasiswa
Aktor	User
Deskripsi	User melakukan <i>input</i> data mahasiswa berupa nilai IPK, Penghasilan Orang Tua, dan Semester.
Normal Course	<ol style="list-style-type: none"> <li>Muncul tampilan menu Data Mahasiswa.</li> <li>User mengisi nilai tiap kolom pada Data Mahasiswa.</li> </ol>
Alternate Course	<ol style="list-style-type: none"> <li>User melakukan hapus semua.</li> <li>User memilih untuk kembali ke menu sebelumnya.</li> </ol>
Pre-Condition	<ol style="list-style-type: none"> <li>Muncul tampilan <i>loading</i>.</li> <li>User memilih OK pada <i>messagebox</i>.</li> </ol>
Post-Condition	<ol style="list-style-type: none"> <li>Muncul tampilan Data Mahasiswa.</li> <li>User telah mengisi tiap kolom Data Mahasiswa.</li> </ol>
Assumption :	-

### 4.3 Implementasi

Berikut ini adalah beberapa *screenshot* aplikasi yang telah dibuat.



Gambar 10. Menu masuk



Gambar 11. Data mahasiswa



Gambar 12. Normalisasi matriks kriteria dan *eigen vector* 1 (kriteria)

## 5. PENGUJIAN

Menggunakan metode *Black Box Testing*, dimana pengujian hanya dilakukan dengan mengamati hasil eksekusi melalui data uji dan memeriksa fungsional dari perangkat lunak [4]. Berikut ini merupakan pengujian menggunakan metode *black box testing*.

Tabel 6. Uji coba *Form1* menu utama

Keterangan	Status	
	Ya	Tidak
<b>Form1</b>		
1. Saat masuk ke dalam menu utama, <i>background</i> aplikasi muncul.	√	
2. Saat masuk ke dalam aplikasi, layar aplikasi menampilkan menu Masuk dan Tentang.	√	
3. Saat masuk ke dalam aplikasi, muncul teks Selamat Datang.	√	
<b>Menu Masuk</b>		
4. Ketika dipilih muncul sub-menu Beasiswa.	√	
<b>Menu Tentang</b>		
5. Ketika dipilih muncul teks baru pada layar aplikasi.	√	
6. Ketika dipilih muncul tombol Kembali.	√	
7. Saat tombol Kembali dipilih, tampilan teks berubah kembali menjadi teks Selamat Datang.	√	
8. Pada saat sub-menu Beasiswa dipilih, muncul tampilan <i>loading</i> dan <i>Form1</i> menutup tampilan.	√	

Tabel 7. Uji coba *Form2* menu *loading*

Keterangan	Status	
	Ya	Tidak
<b>Form2</b>		
1. Pada saat memasuki <i>Form2</i> , <i>background</i> pada <i>Form2</i> muncul.	√	
2. Pada saat memasuki <i>Form2</i> , teks “Mohon Tunggu Sebentar” muncul.	√	
3. <i>ProgressBar</i> akan langsung berjalan hingga penuh, dan muncul <i>messagebox</i> “Terima Kasih Telah Menunggu”	√	
4. Pada saat memilih “OK” pada <i>messagebox</i> , <i>Form3</i> muncul dan <i>Form2</i> menutup tampilan.	√	

Tabel 8. Uji coba *Form3* data mahasiswa

Keterangan	Status	
	Ya	Tidak
<b>Form3</b>		
1. Pada saat memasuki <i>Form3</i> , <i>background</i> pada <i>Form3</i> muncul.	√	
<b>TextBox</b>		
2. Setiap kolom <i>TextBox</i> pada <i>Form3</i> dapat dimasukkan dengan angka.	√	
3. Angka pada <i>textbox</i> dapat dimasukkan sesuai dengan format yang tertera.	√	

Tabel 8. Uji coba *Form3* data mahasiswa (lanjutan)

Keterangan	Status	
	Ya	Tidak
<b>TextBox</b>		
4. Pada <i>messagebox</i> “Masukkan angka dari 1-9” terdapat tombol Ok, saat dipilih <i>messagebox</i> akan menutup tampilan.	√	
5. Hasil jumlah akan langsung muncul ketika melakukan <i>input</i> nilai pada tiap kolom.	√	
<b>Tombol Hapus Semua</b>		
6. Pada saat tombol “Hapus Semua” dipilih, maka akan langsung menghapus nilai kolom secara keseluruhan.	√	
<b>Tombol Kembali</b>		
7. Pada saat memilih tombol “Kembali”, maka akan muncul <i>messagebox</i> “Ingin Kembali”.	√	
8. Pada saat memilih “Yes” pada <i>messagebox</i> “Ingin Kembali”, maka <i>user</i> akan kembali ke <i>Form1</i> dan <i>Form3</i> menutup tampilan.	√	
9. Pada saat memilih “No” pada <i>messagebox</i> “Ingin Kembali”, maka akan muncul <i>messagebox</i> “Anda membatalkan perintah”.	√	
10. Pada saat memilih Ok pada <i>messagebox</i> “Anda membatalkan perintah”, maka <i>user</i> akan kembali ke <i>Form3</i> .	√	
<b>Tombol Lanjut</b>		
11. Pada saat memilih tombol “Lanjut”, namun pada <i>TextBox</i> terdapat kolom yang belum diisi, maka akan muncul <i>messagebox</i> “Data Belum Lengkap”.	√	
12. Pada saat memilih tombol “Lanjut”, maka akan muncul <i>messagebox</i> “Selesai Mengisi”.	√	
13. Pada saat memilih “Yes” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>Form5</i> akan muncul dan <i>Form3</i> menutup tampilan.	√	
14. Pada saat memilih “No” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>user</i> akan kembali ke <i>Form3</i> .	√	

Tabel 9. Uji coba *Form4* menu tabel normalisasi matriks kriteria dan *eigen vector* 1 (kriteria)

Keterangan	Status	
	Ya	Tidak
<b>Form4</b>		
1. Pada saat memasuki <i>Form4</i> , <i>background</i> pada <i>Form4</i> muncul.	√	
2. Saat memasuki <i>Form4</i> , nilai normalisasi tabel akan langsung muncul beserta dengan nilai jumlah dari tiap baris.	√	

Tabel 9. Uji coba *Form4* menu tabel normalisasi matriks kriteria dan *eigen vector* 1 (kriteria) (lanjutan)

Keterangan	Status	
	Ya	Tidak
<b>Tombol Hitung</b>		
3. Saat tombol “Hitung” dipilih, maka akan muncul hasil penghitungan <i>eigen vektor</i> 1 (Kriteria).	√	
4. Nilai jumlah dari <i>eigen vector</i> 1 (Kriteria) adalah 1 (satu) atau mendekati 1 (satu).	√	
<b>Tombol Kembali</b>		
5. Pada saat memilih tombol “Kembali”, maka akan muncul <i>messagebox</i> “Ingin Kembali”.	√	
6. Pada saat memilih “Yes” pada <i>messagebox</i> “Ingin Kembali”, maka <i>user</i> akan kembali ke <i>Form3</i> dan <i>Form4</i> menutup tampilan.	√	
7. Pada saat memilih “No” pada <i>messagebox</i> “Ingin Kembali”, maka akan muncul <i>messagebox</i> “Anda membatalkan perintah”.	√	
8. Pada saat memilih Ok pada <i>messagebox</i> “Anda membatalkan perintah”, maka <i>user</i> akan kembali ke <i>Form4</i>	√	
<b>Tombol Lanjut</b>		
9. Pada saat memilih tombol “Lanjut”, namun nilai <i>eigen vector</i> belum dihitung, maka akan muncul <i>messagebox</i> “ <i>Eigen vector</i> Belum Dihitung”.	√	
10. Pada saat memilih tombol “Lanjut”, maka akan muncul <i>messagebox</i> “Selesai Mengisi”.	√	
11. Pada saat memilih “Yes” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>Form5</i> akan muncul dan <i>Form4</i> menutup tampilan.	√	
12. Pada saat memilih “No” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>user</i> akan kembali ke <i>Form4</i> .	√	

Tabel 10. Uji coba *Form5* menu nilai alternatif

Keterangan	Status	
	Ya	Tidak
<b>Form5</b>		
1. Pada saat memasuki <i>Form5</i> , <i>background</i> pada <i>Form5</i> muncul.	√	
<b>TextBox</b>		
2. Setiap kolom <i>TextBox</i> pada <i>Form5</i> dapat memunculkan angka.	√	
3. Hasil jumlah akan langsung muncul pada tiap kolom.	√	

Tabel 10. Uji coba *Form5* menu nilai alternatif (lanjutan)

Keterangan	Status	
	Ya	Tidak
<b>Tombol Hapus Semua</b>		
1. Pada saat tombol “Hapus Semua” dipilih, maka akan langsung menghapus nilai kolom secara keseluruhan.	√	
<b>Tombol Kembali</b>		
2. Pada saat memilih tombol “Kembali”, maka akan muncul <i>messagebox</i> “Ingin Kembali”.	√	
3. Pada saat memilih “Yes” pada <i>messagebox</i> “Ingin Kembali”, maka <i>user</i> akan kembali ke <i>Form4</i> dan <i>Form5</i> menutup tampilan.	√	
4. Pada saat memilih “No” pada <i>messagebox</i> “Ingin Kembali”, maka akan muncul <i>messagebox</i> “Anda membatalkan perintah”.	√	
5. Pada saat memilih Ok pada <i>messagebox</i> “Anda membatalkan perintah”, maka <i>user</i> akan kembali ke <i>Form4</i> .	√	
<b>Tombol Lanjut</b>		
6. Pada saat memilih tombol “Lanjut”, namun pada <i>TextBox</i> terdapat kolom yang belum diisi, maka akan muncul <i>messagebox</i> “Data Belum Lengkap”.	√	
7. Pada saat memilih tombol “Lanjut”, maka akan muncul <i>messagebox</i> “Selesai Mengisi”.	√	
8. Pada saat memilih “Yes” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>Form6</i> akan muncul dan <i>Form5</i> menutup tampilan.	√	
9. Pada saat memilih “No” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>user</i> akan kembali ke <i>Form5</i> .	√	

Tabel 11. Uji coba *Form6* menu normalisasi matriks alternatif dan *eigen vector 2* (alternatif)

Keterangan	Status	
	Ya	Tidak
<b>Form6</b>		
1. Pada saat memasuki <i>Form6</i> , <i>background</i> pada <i>Form6</i> muncul.	√	
2. Saat memasuki <i>Form6</i> , nilai normalisasi tabel akan langsung muncul beserta dengan nilai jumlah dari tiap baris.	√	
<b>Tombol Hitung</b>		
3. Saat tombol “Hitung” dipilih, maka akan muncul hasil penghitungan <i>eigen vector 2</i> (Alternatif).	√	
4. Nilai jumlah dari <i>eigen vector 2</i> (Alternatif) adalah 1 (satu) atau mendekati 1 (satu).	√	
<b>Tombol Kembali</b>		
5. Pada saat memilih tombol “Kembali”, maka akan muncul <i>messagebox</i> “Ingin Kembali”.	√	

Tabel 11. Uji coba *Form6* menu normalisasi matriks alternatif dan *eigen vector 2* (alternatif) (lanjutan)

Keterangan	Status	
	Ya	Tidak
<b>Tombol Kembali</b>		
6. Pada saat memilih “Yes” pada <i>messagebox</i> “Ingin Kembali”, maka <i>user</i> akan kembali ke <i>Form5</i> dan <i>Form6</i> menutup tampilan.	√	
7. Pada saat memilih “No” pada <i>messagebox</i> “Ingin Kembali”, maka akan muncul <i>messagebox</i> “Anda membatalkan perintah”.	√	
8. Pada saat memilih Ok pada <i>messagebox</i> “Anda membatalkan perintah”, maka <i>user</i> akan kembali ke <i>Form6</i> .	√	
<b>Tombol Lanjut</b>		
9. Pada saat memilih tombol “Lanjut”, namun nilai <i>eigen vector</i> belum dihitung, maka akan muncul <i>messagebox</i> “Eigen vector Belum Dihitung”.	√	
10. Pada saat memilih tombol “Lanjut”, maka akan muncul <i>messagebox</i> “Selesai Mengisi”.	√	
11. Pada saat memilih “Yes” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>Form7</i> akan muncul dan <i>Form6</i> menutup tampilan.	√	
12. Pada saat memilih “No” pada <i>messagebox</i> “Selesai Mengisi”, maka <i>user</i> akan kembali ke <i>Form6</i> .	√	

Tabel 12. *Form7* menu *eigen vector 2 x eigen vector 1*

Keterangan	Status	
	Ya	Tidak
<b>Form7</b>		
1. Pada saat memasuki <i>Form7</i> , <i>background</i> pada <i>Form7</i> muncul.	√	
2. Nilai pada <i>eigen vector 1</i> dan <i>eigen vector 2</i> pada <i>Form</i> sebelumnya langsung muncul pada <i>Form7</i> .	√	
<b>Tombol Hitung</b>		
3. Pada saat tombol “Hitung” dipilih, maka akan muncul hasil dari perkalian <i>eigen vector 2 x eigen vector 1</i> , beserta dengan bobot prioritasnya.	√	
4. Jika, nilai hasil perkalian berjumlah sama maka bobot prioritas akan menampilkan angka 1 (satu).	√	
5. Nilai bobot prioritas akan memunculkan angka 1 (satu) untuk nilai yang lebih besar.	√	
6. Nilai bobot prioritas akan memunculkan angka 2 (dua) untuk nilai yang lebih kecil.	√	
<b>Tombol <math>\lambda</math> Maks</b>		
7. Ketika dipilih, akan langsung memunculkan hasil dari perhitungan $\lambda$ Maks.	√	

Tabel 12. *Form7* menu *eigen vector 2 x eigen vector 1* (lanjutan)

Keterangan	Status	
	Ya	Tidak
<b>Tombol Consistency Index</b>		
8. Ketika dipilih, akan langsung menampilkan hasil dari perhitungan <i>Consistency Index</i> .	√	
<b>Tombol Consistency Ratio</b>		
9. Saat dipilih, akan langsung menampilkan hasil perhitungan dari <i>Consistency Ratio</i>	√	
10. Hasil dari <i>Consistency Ratio</i> bernilai lebih kecil dari 0,1, atau sama dengan 0 (nol).	√	
<b>TextBox</b>		
11. Dapat dimasukkan angka.	√	
<b>Tombol Random Index</b>		
12. Akan menampilkan <i>Form9</i> tanpa menutup tampilan <i>Form8</i> .	√	
<b>Tombol Kembali</b>		
13. Pada saat memilih tombol “Kembali”, maka akan muncul <i>messagebox</i> “Ingin Kembali”.	√	
14. Pada saat memilih “Yes” pada <i>messagebox</i> “Ingin Kembali”, maka <i>user</i> akan kembali ke <i>Form6</i> dan <i>Form7</i> menutup tampilan.	√	
15. Pada saat memilih “No” pada <i>messagebox</i> “Ingin Kembali”, maka akan muncul <i>messagebox</i> “Anda membatalkan perintah”.	√	
16. Pada saat memilih Ok pada <i>messagebox</i> “Anda membatalkan perintah”, maka <i>user</i> akan kembali ke <i>Form7</i> .	√	
<b>Tombol Selesai</b>		
17. Saat dipilih, maka akan muncul <i>messagebox</i> “Keluar Dari Aplikasi”.	√	
18. Saat menekan tombol “Yes” pada <i>messagebox</i> “Keluar Dari Aplikasi”, maka aplikasi akan tertutup.	√	
19. Saat memilih tombol “No”, maka muncul <i>messagebox</i> “Batal”.	√	
20. Saat memilih “Ok” pada <i>messagebox</i> “Batal”, maka <i>user</i> akan kembali ke <i>Form7</i> .	√	

## 6. KESIMPULAN

Kesimpulan dari hasil perancangan dan pembuatan aplikasi yang telah dilakukan, yaitu aplikasi ini dapat dijadikan *second opinion* untuk pemilihan jenis beasiswa yang diperlukan oleh universitas. Selain itu, berdasarkan hasil pengujian aplikasi menggunakan *black box testing* aplikasi berjalan dengan lancar dan tanpa ada kesalahan pada *syntax* program maupun *bug* pada program aplikasi tersebut.

## REFERENSI

- [1]. Wikipedia. “Beasiswa”. <http://id.wikipedia.org/wiki/Beasiswa> (diakses 14 maret 2014).
- [2]. Infoakademika. “Jenis-jenis beasiswa”. <http://www.infoakademika.com/jenis-jenis-beasiswa> (diakses 17 Maret 2014).

- [3]. Darmawan, Agus. “Metode *Analitycal Hierarchy Process*”. <http://agusdar.wordpress.com/2013/05/13/metode-analitycal-hierarchy-process> (diakses 14 Maret 2014).
- [4]. Ruthyanti, Grace. “*Black Box Testing*”. <http://teknologi.kompasiana.com/gadget/2010/12/13/blackbox-testing-324503.html> (diakses 8 Mei 2014).