



# PETIR

## JURNAL PENGKAJIAN DAN PENERAPAN TEKNIK INFORMATIKA

VOLUME 8 - NOMOR 2

SEPTEMBER 2015

ISSN 1978-9262

IMPLEMENTASI DAN PENGUJIAN KINERJA ORACLE 10g *REAL APLICATION CLUSTER* (RAC) PADA SISTEM OPERASI SUN SOLARIS 10

*Gatot Budi Santoso; Yanuar Indra Wirawan*

RANCANG BANGUN APLIKASI *MONITORING* PENCADANGAN DAYA LISTRIK DENGAN MEMANFAATKAN TENAGA KINCIR ANGIN

*Meilia Nur Indah Susanti*

APLIKASI PENGOLAHAN DATA PASIEN, STUDI KASUS RSUD SAWERIGADING PALOPO SULAWESI SELATAN

*Abdul Haris; Alan Burhan*

PENGUNAAN JARINGAN SYARAF TIRUAN DENGAN METODE BACKPROPAGATION DALAM MEMPREDIKSI INDEKS HARGA SAHAM GABUNGAN (IHSG)

*Wisnu Hendro Martono; Dian Hartanti*

APLIKASI KURSUS KOMPUTER *ONLINE* MENGGUNAKAN PHP PADA LEMBAGA KURSUS KOMPUTER YOGZ COURSE

*Harni Kusniyati; Yoga Hapsara Mursidigama*

MONITORING AKSES LOKER DOSEN MENGGUNAKAN *EMBEDDED SYSTEM* DENGAN ANTARMUKA ANDROID

*Riki Ruli A. Siregar; Jaka Mahardika*

TATA KELOLA TINGKAT LAYANAN SISTEM INFORMASI PEMESANAN TIKET MENGGUNAKAN KERANGKA KERJA COBIT 4.1 PADA ARNES SHUTTLE CABANG KOTA BANDUNG

*R.Fenny Syafariani; Gilang Nandapratama*

PERANCANGAN APLIKASI SISTEM PENDUKUNG KEPUTUSAN BERBASIS WEB UNTUK MENENTUKAN PENJURUSAN PADA SMA X DENGAN MENGGUNAKAN METODE AHP (*ANALYTICAL HIERARCHY PROCESS*)

*Yasni Djamain*

IMPLEMENTASI DEMPSTER SHAFER DALAM MENGHASILKAN KEPUTUSAN PENGAMBILAN TOPIK TUGAS AKHIR BAGI MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA UMB

*Desi Ramayanti*

SISTEM LAPORAN KEUANGAN DENGAN MENGGUNAKAN MOBILE PHONE, PHP DAN MYSQL

*Marliana Sari*

SISTEM MONITORING LABORATORIUM KOMPUTER PUSAT UNIVERSITAS MERCU BUANA DENGAN MENGGUNAKAN METODE SCREEN THIEF

*Sarwati Rahayu*

APLIKASI ANTRIAN SMS MENGGUNAKAN *MULTIPLE CHANNEL* DAN *MULTI PHASE SISTEM* DI PT IVM (INTITEK VIRTULINDO MANDIRI)

*Raka Yusuf; Harni Kusniyati; Yuyus Mohayus*

 ISSN 1978-9262 771978 926272	SEKOLAH TINGGI TEKNIK - PLN (STT-PLN)				
	PETIR	VOL. 8	NO. 2	HAL. 133 - 239	JAKARTA, SEPTEMBER 2015

# IMPLEMENTASI DEMPSTER SHAFER DALAM MENGHASILKAN KEPUTUSAN PENGAMBILAN TOPIK TUGAS AKHIR BAGI MAHASISWA PROGRAM STUDI TEKNIK INFORMATIKA UMB

Desi Ramayanti

Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Mercu Buana  
Jl. Raya Meruya Selatan, Kembangan, Jakarta 11650 (11 pt Italic)

E-mail : desi.ramayanti@mercubuana.ac.id

## Abstrak

Dalam pelaksanaan Tugas Akhir pada Program Studi Informatika Fasilkom UMB, banyak mahasiswa yang mengambil sebuah topik skripsi tidak sesuai dengan kemampuannya dalam bidang tersebut, hal tersebut terlihat dari mata kuliah pendukung untuk topik tersebut mendapatkan nilai yang tidak memuaskan. Sehingga akan berpengaruh kepada kesiapan mahasiswa tersebut dalam menyelesaikan tugas akhirnya. Maka penelitian ini merumuskan bagaimana mengimplementasikan teori Dempster Shafer dalam menghasilkan keputusan pengambilan topik tugas akhir bagi Mahasiswa Program Studi Teknik Informatika. Metodologi penelitian yang digunakan adalah waterfall, dan rekayasa perangkat lunak yang digunakan adalah menggunakan UML. Dengan implementasi teori Dempster Shafer ini, maka mahasiswa akan diberikan sebuah hasil yang terkait dengan topik tugas akhir yang akan diambil yaitu Pemrograman Aplikasi atau jaringan.

**Kata Kunci :** Dempster Shafer, Tugas Akhir.

## 1. PENDAHULUAN

Tugas Akhir adalah karya ilmiah yang disusun oleh mahasiswa setiap program studi berdasarkan hasil penelitian suatu masalah yang dilakukan secara seksama dengan bimbingan dosen pembimbing. Tugas akhir merupakan salah satu persyaratan kelulusan mahasiswa. Mahasiswa Teknik Informatika Fasilkom diwajibkan untuk menyelesaikan tugas akhir sebagai kulminasi rangkuman pendidikan selama mengikuti jenjang pendidikan tingkat S-1.

Dalam penyusunan Skripsi, mahasiswa Teknik Informatika dapat menentukan sendiri topik Skripsi, sesuai dengan minat mahasiswa dan ketersediaan pembimbing atau mahasiswa boleh berdiskusi dengan pembimbing untuk mengerjakan topik penelitian dari dosen pembimbing. Topik Skripsi mengacu pada bidang peminatan yang ada pada program studi Teknik Informatika, yaitu: Perancangan Infrastruktur & Pemrograman Aplikasi

Dalam pelaksanaannya, banyak mahasiswa yang mengambil sebuah topik skripsi tidak sesuai dengan kemampuannya dalam bidang tersebut, hal tersebut terlihat dari mata kuliah pendukung untuk topik tersebut mendapatkan nilai yang tidak memuaskan.

Sehingga akan berpengaruh kepada kesiapan mahasiswa tersebut dalam menyelesaikan tugas akhirnya.

Maka berdasarkan hal tersebut, penulis akan mencoba memberikan referensi kepada mahasiswa dengan membangun sebuah sistem pengambilan keputusan tugas akhir mahasiswa dengan menggunakan teori Dempster Shafer.

Teori Dempster-Shafer adalah representasi, kombinasi dan propogasi ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang secara instutitif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat. Dimana dalam penelitian ini, pengambilan keputusan untuk memilih topik yang akan diambil didasarkan kepada nilai nilai matakuliah yang telah diperoleh selama perkuliahan.

## 2. PERUMUSAN MASALAH

1. Bagaimana mengimplementasikan teori Dempster Shafer dalam menghasilkan keputusan pengambilan topik tugas akhir bagi Mahasiswa Program Studi Teknik Informatika.

2. Bagaimana membuat Sistem pengambilan keputusan Keputusan Topik Tugas Akhir : Studi Kasus Mahasiswa Prodi Teknik Informatika UMB

### 3. TINJAUAN PUSTAKA

#### Teori Dempster Shafer

Metode Dempster-Shafer pertama kali diperkenalkan oleh Dempster, yang melakukan percobaan model ketidakpastian dengan *range probabilities* dari pada sebagai probabilitas tunggal. Kemudian pada tahun 1976 Shafer mempublikasikan teori Dempster itu pada sebuah buku yang berjudul *Mathematical Theory Of Evident* [1].

Dempster-Shafer Theory Of Evidence, menunjukkan suatu cara untuk memberikan bobot keyakinan sesuai fakta yang dikumpulkan. Pada teori ini dapat membedakan ketidakpastian dan ketidaktahuan. Teori Dempster-Shafer adalah representasi, kombinasi dan propogasi ketidakpastian, dimana teori ini memiliki beberapa karakteristik yang secara institutif sesuai dengan cara berfikir seorang pakar, namun dasar matematika yang kuat.

Secara umum teori Dempster-Shafer ditulis dalam suatu interval: [Belief, Plausibility] [2]. *Belief* (Bel) adalah ukuran kekuatan evidence dalam mendukung suatu himpunan proposisi. Jika bernilai 0 maka mengindikasikan bahwa tidak ada evidence, dan jika bernilai 1 menunjukkan adanya kepastian.

*Plausibility* (Pls) akan mengurangi tingkat kepastian dari *evidence*. *Plausibility* bernilai 0 sampai 1. Jika yakin akan  $X'$ , maka dapat dikatakan bahwa  $Bel(X') = 1$ , sehingga rumus di atas nilai dari  $Pls(X) = 0$ .

Menurut Giarratano dan Riley fungsi Belief dapat diformulasikan dan ditunjukkan pada persamaan (1):

$$Bel(X) = \sum_{Y \subseteq X} m(Y)$$

Dan *Plausibility* dinotasikan pada persamaan (2):

$$Pls(X) = 1 - Bel(X) = 1 - \sum_{Y \subseteq X} m(Y)$$

Dimana :

Bel (X) = Belief (X)

Pls (X) = Plausibility (X)

$m(X)$  = mass function dari (X)

$m(Y)$  = mass function dari (Y)

Teori Dempster-Shafer menyatakan adanya *frame of discrement* yang dinotasikan dengan simbol ( $\Theta$ ). *frame of discrement* merupakan semesta pembicaraan dari sekumpulan hipotesis sehingga sering disebut dengan *environment* yang ditunjukkan pada persamaan (3) :

$$\Theta = \{ \theta_1, \theta_2, \dots, \theta_n \}$$

Dimana :

$\Theta$  = *frame of discrement* atau *environment*

$\theta_1, \dots, \theta_n$  = *element/* unsur bagian dalam *environment*

*Environment* mengandung elemen-elemen yang menggambarkan kemungkinan sebagai jawaban, dan hanya ada satu yang akan sesuai dengan jawaban yang dibutuhkan. Kemungkinan ini dalam teori Dempster-Shafer disebut dengan *power set* dan dinotasikan dengan  $P(\Theta)$ , setiap elemen dalam *power set* ini memiliki nilai interval antara 0 sampai 1.  $m : P(\Theta) \rightarrow [0,1]$ .

Sehingga dapat dirumuskan pada persamaan (4) :

$$\sum_{X \in P(\Theta)} m(X) = 1$$

Dengan :

$P(\Theta)$  = *power set*

$m(X)$  = *mass function* (X)

*Mass function* (m) dalam teori Dempster-shafer adalah tingkat kepercayaan dari suatu *evidence* (gejala), sering disebut dengan *evidence measure* sehingga dinotasikan dengan (m). Tujuannya adalah mengaitkan ukuran kepercayaan elemen-elemen  $\theta$ . Tidak semua *evidence* secara langsung mendukung tiap-tiap elemen. Untuk itu perlu adanya probabilitas fungsi densitas (m). Nilai m tidak hanya mendefinisikan elemen-elemen  $\theta$  saja, namun juga semua subsetnya. Sehingga jika  $\theta$  berisi n elemen, maka subset  $\theta$  adalah  $2^n$ . Jumlah semua m dalam subset  $\theta$  sama dengan 1. Apabila tidak ada informasi apapun untuk memilih hipotesis, maka nilai :  $m\{\theta\} = 1,0$

Apabila diketahui X adalah subset dari  $\theta$ , dengan  $m_1$  sebagai fungsi densitasnya, dan Y juga merupakan subset dari  $\theta$  dengan  $m_2$  sebagai fungsi densitasnya, maka dapat dibentuk fungsi kombinasi  $m_1$  dan  $m_2$  sebagai  $m_3$ , yaitu ditunjukkan pada persamaan (5) :

$$m_3(Z) = \frac{\sum_{X \cap Y = Z} m_1(X) \cdot m_2(Y)}{1 - \sum_{X \cap Y = \emptyset} m_1(X) \cdot m_2(Y)}$$

Dimana :

$m_3(Z)$  = mass function dari evidence (Z)

$m_1(X)$  = mass function dari evidence (X), yang diperoleh dari nilai keyakinan suatu evidence dikalikan dengan nilai disbelief dari evidence tersebut.

$m_2(Y)$  = mass function dari evidence (Y), yang diperoleh dari nilai keyakinan suatu evidence dikalikan dengan nilai disbelief dari evidence tersebut.

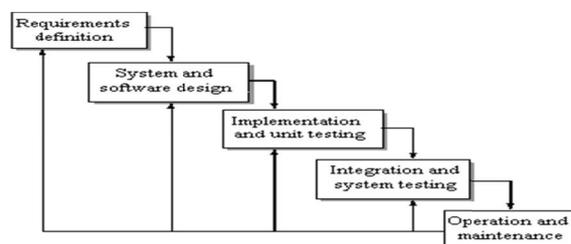
$\sum_{X \cap Y = Z} m_1(X) m_2(Y)$  = merupakan nilai kekuatan dari evidence Z yang diperoleh dari kombinasi nilai keyakinan sekumpulan evidence.

### Metode Waterfall

Metode perancangan software berdasarkan teori model waterfall menurut Sommerville adalah tahapan utama yang langsung mencerminkan dasar pembangunan kegiatan, berikut ini adalah tahapannya:

- *Requirements analysis and definition*, mengumpulkan apa yang dibutuhkan secara lengkap untuk kemudian dianalisis guna mendefinisikan kebutuhan yang harus dipenuhi oleh program yang akan dibangun. Fase ini harus dikerjakan dengan lengkap untuk menghasilkan desain yang lengkap.
- *System and software design*, setelah apa yang dibutuhkan telah selesai dikumpulkan dan sudah lengkap maka desain kemudian di kerjakan.
- *Implementation and unit testing*, desain program diterjemahkan dalam kode-kode dengan menggunakan bahasa pemrograman yang sudah di tentukan. Program yang dibangun langsung diuji secara unit, apakah sudah bekerja dengan baik.
- *Integration and system testing*, penyatuan unit-unit program untuk kemudian di uji secara keseluruhan (*system testing*).

- *Operating and maintenance*, mengoperasikan program dilingkungannya dan melakukan pemeliharaan, seperti penyesuaian atau perubahan untuk adaptasi dengan situasi yang sebenarnya.



Gambar 1. Metode Waterfall

### Unified Modeling Language (UML)

*Unified modeling language* ialah bahasa yang di gunakan untuk pengembangan sebuah sistem serta menentukan, memvisualisasikan, membangun dan mendokumentasikan suatu sistem informasi [3]. Pemodelan UML yang sering digunakan untuk menggambarkan suatu sistem terdiri dari :

#### Use Case Diagram

Menurut [4] *use case diagram* adalah penggambaran dari interaksi antara sistem dan aktor. Cara kerja *use case* dengan cara mendeskripsikan tipe interaktif antara pengunjung (*user*) sebuah sistem dengan sistem dipakai, serta memfasilitasi komunikasi diantara analis dan pengguna dan antara analis dan *client*. Notasi yang digunakan dalam *use case* adalah persegi panjang yang merupakan *system boundary*, oval yang merupakan suatu proses, dan gambar orang yang berinteraksi dalam proses tersebut.

Elemen *use case* terdiri dari :

- Diagram *use case*, disertai dengan narasi dan skenario.
- Aktor, mendefinisikan entitas diluar sistem yang memakai sistem.
- Asosiasi, mengidikasikan aktor mana yang berinteraksi dengan *use case* dalam suatu sistem.
- `<<include>>` dan `<<extend>>`. Merupakan indicator yang menggambarkan jenis relasi dan interaksi antar *use case*.
- Generalisasi, menggambarkan hubungan turunan antara *use case* atau antar aktor.

### **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, di mana sebagian besar *state* adalah *action* dan sebagian besar transisi di-*trigger* oleh selesainya *state* sebelumnya (*internal processing*). Oleh karena itu *activity diagram* tidak menggambarkan *behaviour internal* sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum. Menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. Struktur diagram ini mirip *flowchart* atau Data Flow Diagram pada perancangan terstruktur. Sangat bermanfaat apabila kita membuat diagram ini terlebih dahulu dalam memodelkan sebuah proses untuk membantu memahami proses secara keseluruhan [4].

## **4. TUJUAN PENELITIAN**

Membangun sebuah sistem yang dapat melakukan pengambilan keputusan untuk menentukan pilihan topik tugas akhir mahasiswa prodi teknik informatika.

## **5. METODE PENELITIAN**

Metode penelitian berisi metode-metode penelitian yang digunakan untuk melakukan penelitian. Metode penelitian yang dilakukan dalam penelitian ini adalah :

### **Metode Analisa**

Penulis memperoleh pemikiran teoritis dengan mencari dan mempelajari topik-topik yang berkaitan dengan teori sistem pakar, *dempster shafer*, bahasa pemrograman yang mendukung, serta dengan membaca buku-buku dan *literature* lain yang berkaitan dengan permasalahan yang menjadi pembahasan pada penelitian ini.

### **Metode perancangan perangkat lunak**

Metode perancangan aplikasi dalam tugas akhir ini menggunakan metode pemodelan perangkat lunak *classic life cycle*

yang lebih dikenal dengan nama *waterfall model*. Model ini memberikan pendekatan yang sistematis dan berurutan bagi pengembangan perangkat lunak. Pendekatan sistematis tersebut direpresentasikan kedalam tahapan-tahapan sebagai berikut:

## **1. Analisa System**

### **Analisa Masalah**

- Program Studi Teknik Informatika Fakultas ilmu komputer memiliki 2 jenis peminatan yaitu Perancangan Infrastruktur dan Pemrograman Aplikasi
- Setiap Mahasiswa akan menyelesaikan studinya, dengan mengambil satu mata kuliah yang bernama Tugas Akhir
- Pengambilan tema Tugas Akhir tidak berdasarkan jalur peminatan yang sudah dipilih oleh setiap mahasiswa, hal ini dikarenakan mahasiswa belum punya panduan siapa dosen dan apa saja tema yang ingin diambil berdasarkan pilihan peminatan.

### **Analisa Kebutuhan**

- Diperlukan suatu system yang dapat menjadi panduan bagi mahasiswa untuk mengambil tema Tugas Akhir yang sesuai dengan pilihan peminatan
- System yang akan dibangun merupakan system pengambilan keputusan menggunakan teori Dempster Shafer
- Mahasiswa bisa mendapatkan tema Tugas Akhir secara otomatis berdasarkan jalur peminatan, dan berdasarkan nilai dari mata kuliah yang pernah diambil.
- Mata kuliah yang akan dijadikan alat pengambil keputusan adalah matakuliah Fakultas dan matakuliah program studi saja, sehingga tidak melibatkan mata kuliah universitas (MKCU), dalam hal ini mata kuliah dari semester 1 sampai semester 4.
- System bisa diakses secara *online*, dalam hal ini berbasis Web
- Diharapkan bisa terkoneksi dengan SIA, sehingga tema yang akan diambil bisa diketahui pada saat mahasiswa mengklik KRS untuk matakuliah Tugas Akhir.

### **Analisa Data**

Mata kuliah yang akan dijadikan acuan adalah pada Tabel 1 dibawah.

Tabel 1. Pembagian Mata kuliah

Pemrograman Aplikasi	Jaringan
Dasar Pemrograman	Dasar Pemrograman
Algoritma Struktur Data	
P. Algoritma & Struktur Data	
Pemrograman Web (Lab)	
Analisa Berorientasi Object	Analisa Berorientasi Object
P. Basis Data (ORACLE)	
Mobile Programming	
PBO	
Basis Data	Basis Data
	Arsitektur Komputer
	Teknologi Pusat Data
	Sistem Operasi
	Sistem Jaringan

Perhitungan Dempster Shafer

- Nilai probabilitas Densitas

Berdasarkan persamaan (4), yaitu

$$\sum_{X \in P(\theta)} m(X) = 1$$

Tabel 2. Nilai densitas mata kuliah

MATA KULIAH	PROBABILITAS DENSITAS	m{0}	
Dasar Pemrograman	m1 {P,J}	0.5	0.5
Algoritma Struktur Data	m2 {P}	0.5	0.5
P. Algoritma & Struktur Data	m3 {P}	0.5	0.5
Pemrograman Web (Lab)	m4 {P}	0.8	0.2
Analisa Berorientasi Object	m5 {P,J}	0.5	0.5
P. Basis Data (ORACLE)	m6 {P}	0.8	0.2
Mobile Programming	m7 {P}	0.75	0.25
PBO	m8 {P}	0.75	0.25
Basis Data	m9 {P,J}	0.5	0.5
Arsitektur Komputer	m10 {J}	0.75	0.25
Teknologi Pusat Data	m11 {J}	0.5	0.5
Sistem Operasi	m12 {J}	0.5	0.5
Sistem Jaringan	m13 {J}	0.8	0.2

Normalisasi Nilai untuk setiap mata kuliah dapat disajikan sebagai berikut pada Tabel 3.

Tabel 3. Normalisasi Nilai

Nilai Huruf	Nilai Angka	Normalisasi Nilai
A	4	1
B+	3.5	0.875
B	3	0.75
C+	2.5	0.625
C	2	0.5
D	1	0.25
E	0	0

- Hasil penggabungan Tabel 2 dengan Tabel 3 dengan menambahkan nilai dari setiap matakuliah merupakan normalisasi nilai densitas yang akan digunakan untuk proses selanjutnya, dapat dilihat pada Tabel 4.

Tabel 4. Normalisasi Densitas

MATA KULIAH	PROBABILITAS DENSITAS	m{0}	Nilai	Nilai Angka	Nilai Huruf	Normalisasi Nilai	Prob Hasil	m{0} hasil	
Dasar Pemrograman	m1 {P,J}	0.5	0.5	87	4	A	1	0.5	0.5
Algoritma Struktur Data	m2 {P}	0.5	0.5	66	2.5	C-	0.625	0.3125	0.6875
P. Algoritma & Struktur Data	m3 {P}	0.5	0.5	76	3.5	B-	0.875	0.4375	0.5625
Pemrograman Web (Lab)	m4 {P}	0.8	0.2	80	4	A	1	0.8	0.2
Analisa Berorientasi Object	m5 {P,J}	0.5	0.5	45	1	D	0.25	0.125	0.875
P. Basis Data (ORACLE)	m6 {P}	0.8	0.2	56	2	C	0.5	0.4	0.6
Mobile Programming	m7 {P}	0.7	0.25	66	2.5	C-	0.625	0.46875	0.53125
PBO	m8 {P}	0.7	0.25	69	2.5	C-	0.625	0.46875	0.53125
Basis Data	m9 {P,J}	0.5	0.5	71	3	B	0.75	0.375	0.625
Arsitektur Komputer	m10 {J}	0.7	0.25	74	3.5	B-	0.875	0.65625	0.34375
Teknologi Pusat Data	m11 {J}	0.5	0.5	80	4	A	1	0.5	0.5
Sistem Operasi	m12 {J}	0.5	0.5	77	3.5	B-	0.875	0.4375	0.5625
Sistem Jaringan	m13 {J}	0.8	0.2	73	3.5	B-	0.875	0.7	0.3

Untuk probabilitas hasil diperoleh dengan persamaan : Probabilitas densitas (mn) dikali dengan normalisasi nilai.

Untuk m{0} hasil diperoleh dari normalisasi nilai dikurangi dengan probabilitas hasil.

- Berdasarkan nilai densitas pada Tabel 4 (m1 – m13) diatas, maka akan diperoleh nilai densitas selanjutnya yaitu m14, dengan menggunakan persamaan(5)

$$m3(Z) = \frac{\sum_{X \cap Y = Z} m1(X) \cdot m2(Y)}{1 - \sum_{X \cap Y = \emptyset} m1(X) \cdot m2(Y)}$$

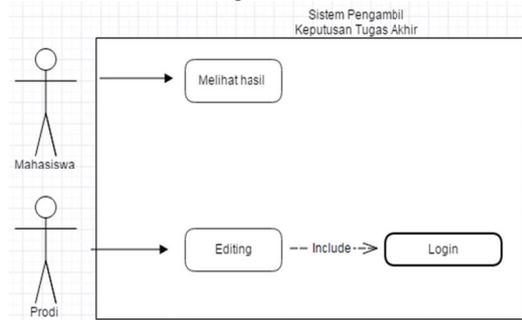
Hasil perhitungan m14, yaitu terlihat pada Tabel 5 dibawah ini.

Tabel 5. Perhitungan M14

M14 {P}	0.1563	+	0.1563	+	0	+	0	/	1	-	0	=	0.3125
M14 {P,J}	0.3438	+	0	+	0	+	0	/	1	-	0	=	0.3438
M14 {0}	0.3438	+	0	+	0	+	0	/	1	-	0	=	0.3438

## 2. Perancangan Sistem

- Use case Diagram



Gambar 2. Use case Diagram Sistem

### Keterangan

#### Aktor

1. Mahasiswa  
Aktor mahasiswa hanya bisa mengakses use case **Melihat Hasil**
2. Prodi  
Aktor Prodi bisa mengakses semua use case yaitu **Melihat Hasil**, dan **Editing**. Dan untuk melakukan **Editing**, maka diperlukan **Login** terlebih dahulu.

#### Use case

##### 1. Melihat Hasil

Pada usecase ini semua aktor bisa melihat hasil dari pengambilan keputusan yang berkaitan dengan Program Studi / Jurusan yang akan diambil.

Pada usecase ini, user hanya memasukkan salah satu dari dua keyword yaitu NIM>Nama mahasiswa yang ingin dicari. Jika pencarian sukses, maka outputnya adalah Pilihan Program Studi atau Jurusan yang akan diambil yaitu Program Aplikasi atau Jaringan Infrastruktur.

##### 2. Editing

Pada usecase ini, aktor yang terlibat adalah Prodi. Prodi bisa melakukan editing terhadap data mahasiswa, seperti *add*, *delete* dan *edit*. Dan untuk bisa masuk kedalam usecase ini, maka prodi wajib melakukan proses *login*, agar memperoleh hak akses.

##### 3. Login

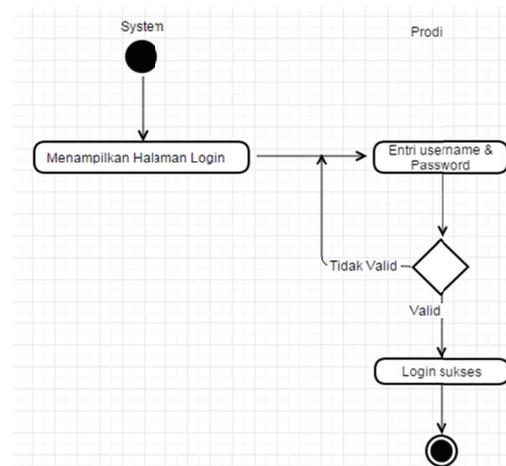
Pada usecase ini, aktor yang terlibat adalah Prodi.

- Activity Diagram

- Login

Untuk usecase Login pada Gambar 3 diatas, dapat diuraikan dalam bentuk activity diagram sebagai berikut:

1. Sistem akan menampilkan halaman Login, dimana pada halaman ini user akan diminta untuk mengisi *Username* dan *Password*
2. User mengisi *username* dan *Password* yang sudah didaftarkan sebelumnya dan tersimpan dalam sebuah table pada database yang digunakan.
3. System akan melakukan verifikasi untuk memastikan *username* dan *password* yang diisi valid atau tidak, dengan mencek kedalam database yang ada.
4. Jika *username* dan *password* yang dimasukkan ada dan tervalidasi, maka user diberikan hak akses, jika tidak maka user akan gagal masuk kedalam system.

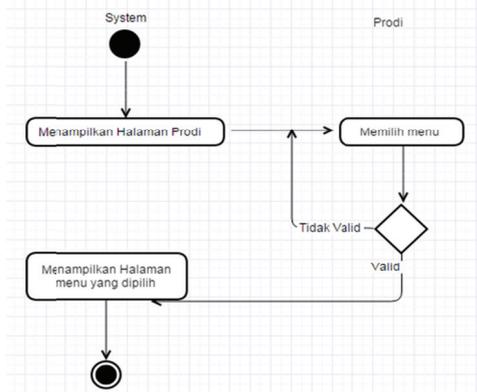


Gambar 3. Activity Diagram Login

- Editing

Untuk usecase *Editing* pada Gambar 4. diatas, dapat diuraikan dalam bentuk activity diagram sebagai berikut:

1. Sebelum masuk pada aktifitas *Editing* ini, Prodi diharuskan untuk melakukan Login, hal ini bertujuan untuk menjaga keamanan system sendiri.
2. Setelah sukses login, maka prodi bisa mengakses beberapa menu yaitu Add, View, Edit dan Delete Data,
3. Pada Menu Add, Prodi bisa menambahkan data mahasiswa berupa nilai nilai mahasiswa. Untuk menu Add ini direncanakan bisa diambil dari data mahasiswa pada SIA.



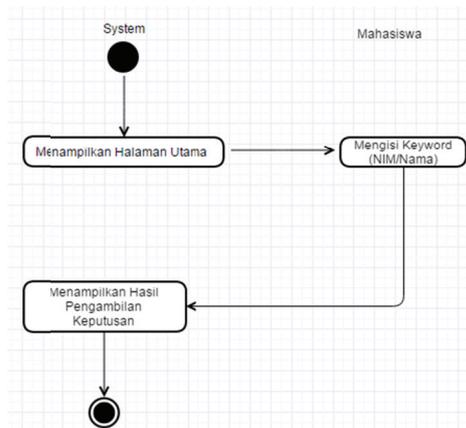
Gambar 4. Activity Diagram Editing

4. Pada Menu View, Prodi bisa melihat data dari mahasiswa terkait.
5. Pada Menu Edit, Prodi bisa melakukan koreksi terhadap data mahasiswa, dan jika sudah terkoneksi dengan SIA, maka menu ini tidak bisa difungsikan lagi.
6. Pada Menu Delete, Prodi bisa menghapus data mahasiswa, dan jika sudah terkoneksi dengan SIA, maka menu ini tidak bisa difungsikan lagi.

- Melihat Hasil

Untuk usecase Melihat Hasil pada Gambar 5. diatas, dapat diuraikan dalam bentuk activity diagram sebagai berikut:

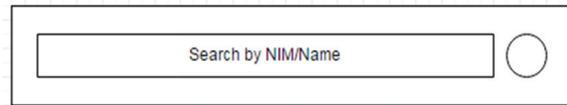
1. Pada halaman utama akan menampilkan sebuah form pencarian. Dimana mahasiswa akan melakukan pencarian dengan keyword adalah NIM dan nama mahasiswa.
2. Jika NIM atau nama mahasiswa yang dicari ada pada database maka system akan memberikan output hasil yaitu Program Aplikasi atau Jaringan



Gambar 5. Activity Diagram Melihat Hasil

● Perancangan Interface

1. Interface halaman pencarian

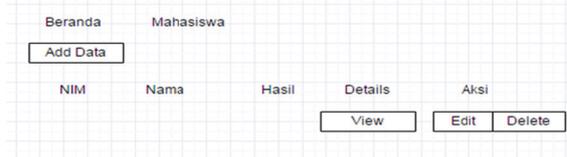


Gambar 6. Interface Halaman Pencarian

Keterangan:

- *Inputan* → Pada halaman ini, mahasiswa akan melakukan proses pencarian dengan mengetikkan NIM atau Nama
- *Tombol Search* → berfungsi untuk melakukan pencarian data kedalam database

2. Interface halaman prodi



Gambar 7. Interface Halaman Prodi

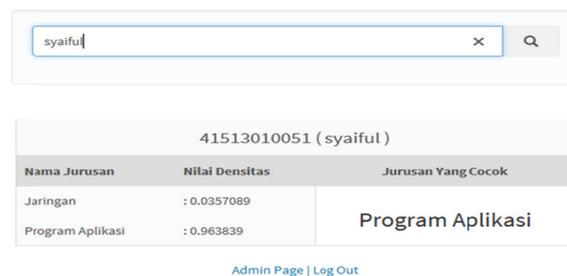
Keterangan:

- *Beranda* → Link ke halaman pencarian
- *Mahasiswa* → Halaman Prodi
- *Add Data* → Fungsi untuk menambah data
- *View* → Fungsi untuk melihat data mahasiswa
- *Edit* → Fungsi untuk melakukan perubahan data mahasiswa
- *Delete* → Fungsi untuk melakukan penghapusan data mahasiswa

3. Implementasi Sistem

● Implementasi

1. Halaman Pencarian



Gambar 8. Implementasi Halaman Pencarian

Keterangan:

- *Inputan* → Pada halaman ini, mahasiswa akan melakukan proses pencarian dengan mengetikkan NIM atau Nama
- *Tombol Search* → berfungsi untuk melakukan pencarian data kedalam database
- Jika data ditemukan, maka outputnya adalah hasil dari Proses pengambilan Keputusan.

## 2. Halaman Prodi

Data Mahasiswa

Baranda Mahasiswa

+ ADD DATA

Entries per page: 5

#	NIM	Nama	Hasil Jurusan	Detail	Aksi
1	4151301001	Ai maulodya	Sistem Jaringan	VIEW	EDIT DELETE
2	4151301002	Aling latomana	Program Aplikasi	VIEW	EDIT DELETE
3	4151301003	syafiq	Program Aplikasi	VIEW	EDIT DELETE
4	4151301007	Septian dwi purnomo	Sistem Jaringan	VIEW	EDIT DELETE
5	4151301008	Edo tessel	Program Aplikasi	VIEW	EDIT DELETE

1 2

Gambar 9. Implementasi Halaman Prodi

### Keterangan:

Pada halaman ini prodi dapat melihat semua data mahasiswa dan hasil pilihan topik tugas akhirnya. Selain itu prodi bisa melakukan penambahan data, view, edit serta delete data mahasiswa.

## 6. KESIMPULAN DAN SARAN

### Kesimpulan

1. Teori Dempster Shafer telah dapat diimplementasikan dalam menghasilkan keputusan pengambilan topik tugas akhir bagi Mahasiswa Program Studi Teknik Informatika
2. Aplikasi pengambilan keputusan Keputusan Topik Tugas Akhir Studi Kasus Mahasiswa Prodi Teknik Informatika UMB, telah dapat dibuat sesuai dengan rancangan.

### Saran

1. Aplikasi ini dapat terhubung dengan SIA (Sistem Informasi Akademik) sehingga data tidak perlu diinput lagi oleh Prodi, tetapi terhubung dengan database yang ada.
2. Bisa dikembangkan untuk seluruh Program Studi yang ada di Universitas Mercubuana.

### DAFTAR PUSTAKA

- [1] J. and R. G. Giarratano, *Expert Systems ; Principles and Programming*. PWS Publishing Company, Boston., 2005.
- [2] S. Kusumadewi, *Artificial Intelligence (Teknik dan Aplikasinya)*. 2003.
- [3] P. P. W. Herlawati, *Menggunakan UML (Unified Modelling Language)*. Informatika, 2011.
- [4] J. Hermawan, *Analisa Desain & Pemrograman Berorientasi Obyek dengan UML*. Yogyakarta: Andi, 2008.