BAB III

PERANCANGAN SISTEM

3.1 Analisis Sistem

Tahapan analisis sistem ini, mempunyai tugas mendefinisikan masalah sistem, melakukan studi kelayakan, menganalisis kebutuhan sistem dan elemen-elemen yang terkait, seperti user, administrator, dan semua yang diperlukan dalam proses prediksi masa studi mahasiswa di jurusan teknik komputer UNIKOM.

3.1.1 Analisis Masalah

Setiap perguruan tinggi baik negeri maupun swasta tentunya memiliki kebijakan kurikulum yang berbeda. Di UNIKOM khususnya pada program studi S1 Teknik Komputer tingkat kelulusan masih terbilang rendah hal ini disebabkan oleh beberapa faktor, yang diantaranya: kurangnya strategi mahasiswa dalam belajar, sistem pengajaran yang kurang baik dan adanya perbedaan kemampuan setiap mahasiswa. Dari beberapa faktor tersebut menyebabkan banyak mahasiswa yang menempuh masa studi lebih dari lima tahun. Untuk mengatasi permasalahan tersebut maka penulis membuat solusi mengimplementasikan *data mining*, dengan memanfaatkan nilai akademik.

Dengan memanfaatkan sistem *data mining* khususnya algoritma ID3 dibuat sebuah perancangan aplikasi untuk memprediksi masa studi mahasiswa berdasarkan data nilai akademik, dengan dua kategori masa studi kurang dari lima tahun dan lebih dari lima tahun. Dengan adanya aplikasi ini diharapkan dapat dimanfaatkan dosen dan staff jurusan untuk memprediksi kelulusan mahasiswa dalam masa studinya.

3.1.2 Analisis Data

Data yang digunakan dalam penulisan tugas akhir ini adalah data akademik dengan menggunakan 6 matakuliah sebagai atribut yaitu fisika1, kalkulus1, pengantar sistem komputer, kalkulus2, fisika2 dan algoritma

pemrograman1. Sedangkan Atribut tujuan dari prediksi masa studi mahasiswa terdiri dari dua kategori yaitu masa studi <= 5 tahun dan > 5tahun. Diman 0 mewakili kelas yang <= 5 tahun dan 1 > mewakili kelas yang > 5 tahun.

3.1.2.1 Sumber Data

Data mahasiswa yang digunakan untuk uji data (*testing*) *dan* data latih *(training)* adalah data mahasiswa angkatan 2000-2006. Hal ini didasarkan pada kebutuhan data yang akan dihubungkan dengan sistem prediksi yang akan dirancang dari data kelulusan, data tersebut diperoleh dari sekretariat jurusan Teknik Komputer. Data yang diambil semester1 dan semester 2 dari data mahasiswa S1 yang telah lulus di Jurusan Teknik Komputer. Berikut adalah tabel atribut yang digunakan dalam aplikasi ini:

Tabel 1 Tabel 6 Atribut

NO	Nama Matakuliah	Bobot Nilai
1	Kalkulus1	A
		B
		C
		D
		E
2	Pengantar Sistem Komputer	A
		B
		C
		D
		E
3	Fisika 1	A
		B
		C
		D
		E
4	Kalkulus 2	A
		B
		C
		D
		E
5	Fisika 2	A
		B
		C

		D
		E
6	Pemrograman Algoritma1	A
		B
		C
		D
		E

3.1.3 Penggunaan Algoritma ID3

Perhitungan algoritma ID3 terlihat pada contoh kasus dan perhitungan node ke- pada tabel dibawah ini, dimana diambil 4 matakuliah sebagai atribut yaitu fisika1, kalkulus1, fisika2, dan kalkulus2. Dari 4 atribut tersebut, terdapat 5 nilai yang digunakan yaitu: A, B, C, D, dan E. Sedangkan ada 2 target atau atiribut tujuan yang dihasilkan yaitu masa studi kurang dari lima tahun diwakili dengan 0 dan lebih dari lima tahun diwakili 1. Pengambilan atribut-atribut tersebut diambil dari 2 semester yang terdapat di jurusan teknik komputer.

Tabel 2 Contoh Tabel Kasus

no	fisika1	kalkulus1	fisika2	kalkulus2	masa_studi
1	A	B	B	B	1
2	B	C	B	C	1
3	C	D	B	E	1
4	A	A	A	A	0
5	C	D	C	C	0
6	A	C	A	B	1
7	C	C	B	D	0
8	C	C	D	C	0
9	C	C	C	C	0
10	B	C	D	C	0

Tabel kasus diatas merupakan sampel kumpulan data yang akan diproses, yang terdiri dari 10 sampel. 4 sampel merupakan jumlah sampel lebih dari 5 tahun dengan simbol 1, dan 6 sampel merupakan jumlah sampel yang

kurang dari 5 tahun dengan simbol 0. Untuk melakukan proses klasifikasi maka dilakukan pengecekan isi dari atribut kelas data (matakuliah), karena isi atribut tersebut berbeda maka digunakan fungsi *entropy* untuk mendapatkan informasi tertinggi dengan perhitungan dari atribut matakuliah berikut:

$$\text{Entropy}([6+,4-]) = -\left(\frac{6}{10}\right)\log_2\left(\frac{6}{10}\right) - \left(\frac{4}{10}\right)\log_2\left(\frac{4}{10}\right)$$

$$= 0.97095$$

Dari tabel diatas misalnya atribut diterima = ”lulus kurang dari 5 tahun” merupakan sampel (+) dan atribut diterima = ”lebih dari 5 tahun” merupakan sampel (-). Untuk mencari atribut yang akan dijadikan root dapat dilakukan dengan menghitung gain ratio tertinggi, dengan menggunakan rumus (2), dapat dilihat dari tabel dibawah ini:

1. Perhitungan mencari atribut yang akan dijadikan *root*.

Tabel 3 Perhitungan Menentukan Node ke-1

node			**Jumlah_kasus**	**<=5thn**	**> 5 thn**	**Entropy**	**Gain**
1	Total		10	6	4	0,970951	
	fisika1						0,134498
		A	3	1	2	0,918296	
		B	2	1	1	1	
		C	5	4	1	0,721928	
		D	0	0	0	0	
		E	0	0	0	0	
	fisika2						0,446439
		A	2	1	1	1	
		B	4	1	3	0,811278	
		C	2	2	0	0	
		D	2	2	0	0	
		E	0	0	0	0	

kalkulus 1					0,219973
	A	1	1	0	0
	B	1	0	1	0
	C	6	4	2	0,918296
	D	2	1	1	1
	E	0	0	0	0
kalkulus 2					0,609987
	A	1	1	0	0
	B	2	0	2	0
	C	5	4	1	0,721928
	D	1	1	0	0
	E	1	0	1	0

Dari perhitungan diatas maka didapat Informasi Gain dari empat atribut matakuliah (Kalkulus1,Fisika1, Kalkulus 2, dan Fisika 2,)

IG Kalkulus1 = 0,219973

IG Fisika1 = 0, 134498

IG Kalkulus2 = 0,609987

IG Fisika2 = 0,446439

Dari keempat atribut nilai Informasi Gain (Skalkulus2) adalah nilai informasi gain terbesar sehingga atribut kalkulus 2 merupakan kelas yang terbaik dan harus diletakan sebagai root. Dengan perhitungan entropy dan gain untuk kalkulus2:

Pencarian Entropy untuk matakuliah kalkulus2:

Nilai (kalkulus2) = A, B, C, D, E

Entropy (S_A) = [1+, 0-] = $- (\frac{1}{1}) \log_2 (\frac{1}{1}) - (\frac{0}{0}) \log_2 (\frac{0}{1}) = 0$

Entropy (S_B) = [0+, 2-] = $- (\frac{0}{2}) \log_2 (\frac{0}{2}) - (\frac{2}{2}) \log_2 (\frac{2}{2}) = 0$

Entropy (S_C) = [4+, 1-] = $-\left(\frac{4}{5}\right)\log_2\left(\frac{4}{5}\right)-\left(\frac{1}{5}\right)\log_2\left(\frac{1}{5}\right)$ =0,721928

Entropy (S_D) = [1+, 0-] = $-\left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right)-\left(\frac{0}{1}\right)\log_2\left(\frac{0}{1}\right)$ = 0

Entropy (S_E) = [0+, 1-] = $-\left(\frac{0}{1}\right)\log_2\left(\frac{0}{1}\right)-\left(\frac{1}{1}\right)\log_2\left(\frac{1}{1}\right)$ = 0

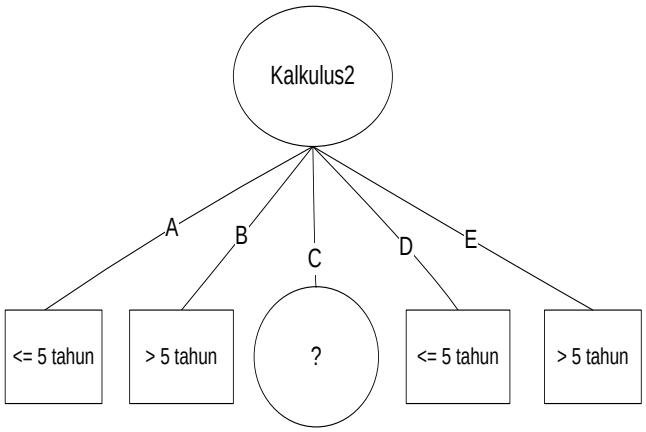
Pencarian Gain pada matakuliah kalkulus 2:

Gain (S,A) = Entropy(S) - $\sum\frac{|Sv|}{|S|}Entropy(S_V)$

$$Gain(112,6) = 0{,}970951-\left(\left(\frac{1}{10}.0\right)+\left(\frac{2}{10}.0\right)+\left(\frac{5}{10}.0{,}721928\right)+\left(\frac{1}{10}.0\right)+\left(\frac{1}{10}.0\right)\right)$$

= 0,609987

Dari perhitungan diatas didapat pohon keputusan sebagai berikut:



Gambar 1 Pohon Keputusan Hasil Perhitungan Node Ke-1

2. Perhitungan mencari atribut yang akan dijadikan leaf untuk Node ke-2 dari atribut kalkulus 2 dengan nilai C.

Tabel 4 Hasil Perhitungan Node Ke-2 dari Atribut kalkulus2

node			Jumlah_kasus	<=5thn	> 5 thn	Entropy	Gain
2	kalkulus2C		5	4	1	0,721928	
	fisika1						0,321928
		A	0	0	0	0	
		B	2	1	1	1	

	C	3	3	0	0	
	D	0	0	0	0	
	E	0	0	0	0	
fisika2						0,721928
	A	0	0	0	0	
	B	1	0	1	0	
	C	2	2	0	0	
	D	2	2	0	0	
	E	0	0	0	0	
kalkulus1						0,072906
	A	0	0	0	0	
	B	0	0	0	0	
	C	4	3	1	0,811278	
	D	1	1	0	0	
	E	0	0	0	0	

Tabel diatas merupakan perhitungan mencari atribut yang akan dijadikan leaf pada node ke-2 dari atribut kalkulus 2 dengan nilai C.

IG Kalkulus1 = 0,072906

IG Fisika1 = 0,321928

IG Fisika2 = 0,721926

Dari perhitungan tiga atribut diatas didapat nilai Informasi Gain fisika2 adalah nilai informasi gain terbesar, maka yang akan menjadi cabang node kalkulus2 selanjutnya adalah matakuliah fisika2 dengan perhitungan entropy dan gain sebagai berikut:

Perhitungan entropy untuk matakuliah fisika2:

Entropy (S_A) = [0+, 0-] = $- \left(\frac{0}{0}\right) \log_2 \left(\frac{0}{0}\right) - \left(\frac{0}{0}\right) \log_2 \left(\frac{0}{0}\right) = 0$

Entropy (S_B) = [0+, 1-] = $- \left(\frac{0}{1}\right) \log_2 \left(\frac{0}{1}\right) - \left(\frac{1}{1}\right) \log_2 \left(\frac{1}{1}\right) = 0$

Entropy (S_C) = [2+, 0-] = $- \left(\frac{2}{2}\right) \log_2 \left(\frac{2}{2}\right) - \left(\frac{0}{2}\right) \log_2 \left(\frac{0}{2}\right) = 0$

Entropy (S_D) = [2+, 0-] = $- \left(\frac{2}{2}\right) \log_2 \left(\frac{2}{2}\right) - \left(\frac{0}{2}\right) \log_2 \left(\frac{0}{2}\right) = 0$

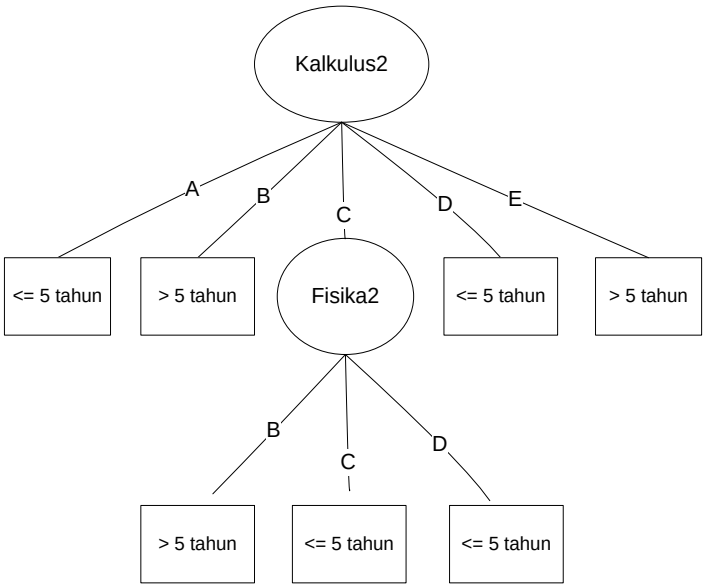
Entropy (S_E) = [0+, 0-] = - ($\frac{0}{0}$) $\log_2$ ($\frac{0}{0}$) –($\frac{0}{0}$) $\log_2$($\frac{0}{0}$) = 0

Perhitungan gain untuk matakuliah fisika2:

$$Gain(112,6) = 0{,}970951 - \left(\left(\frac{0}{5}.0\right) + \left(\frac{1}{5}.0\right) + \left(\frac{2}{5}.0\right) + \left(\frac{2}{5}.0\right) + \left(\frac{0}{5}.0\right)\right)$$

= 0,721928

Nilai A dan E pada fisika 2 jumlah kasusnya 0 artinya tidak ada data yang < = 5 tahun dan > 5 tahun, maka pohon keputusan yang didapat adalah:



Gambar 2 Pohon Keputusan Hasil Perhitungan Node Ke-2

Dari data kasus diatas maka dapat ditarik sebuah aturan bahwa:

a. JikaKalkulus2 = A maka masa studi < = 5 tahun.
b. JikaKalkulus2 = B maka masa studi > 5 tahun.
c. Jika Kalkulus2 = C dan Fisika2 = B maka masa studi > 5 tahun.
d. Jika Kalkulus2 = C dan Fisika2 = C maka masa studi < = 5 tahun.
e. Jika Kalkulus2 = C dan Fisika2 = D maka masa studi < = 5 tahun.
f. Jika Kalkulus2 = D maka masa studi < = 5 tahun.
g. Jika Kalkulus2 = E maka masa studi > 5 tahun.

3.1.4 Analisis Kebutuhan *Non Fungsional*

Analisis *non fungsional* adalah sebuah langkah dimana seorang pembangun perangkat lunak menganalisis sumber daya yang akan menggunakan perangkat lunak yang dibangun.

Analisis *non fungsional* tidak hanya menganalisis siapa saja yang akan menggunakan aplikasi tetapi juga menganalisis perangkat keras dan perangkat lunak yang dimiliki oleh pemesan, sehingga dapat ditentukan kompabilitas aplikasi yang dibangun terhadap sumber daya yang ada. Setelah melakukan analisis non fungsional, maka dilanjutkan kelangkah berikutnya yaitu menentukan kebutuhan non fungsional sistem yang akan dibangun untuk disesuaikan dengan fakta yang ada.

Apabila terjadi ketidakcocokan antara fakta dan kebutuhan maka perlu adanya penyesuaian fakta terhadap kebutuhan yang ada. Apabila kebutuhan tidak dipenuhi maka sistem yang dibangun tidak akan berjalan baik sesuai yang diharapkan.

Analisis *non fungsional* dan kebutuhan *non fungsional* yang dilakukan dibagi dalam tiga tahapan, yaitu:

1. Kebutuhan pengguna (*user*)
2. Kebutuhan perangkat keras (*hardware*)
3. Kebutuhan perangkat lunak (*software*)

3.1.4.1 Analisis Kebutuhan Pengguna (*user*)

Aplikasi sistem prediksi ini akan digunakan oleh dosen dan staff jurusan teknik komputer. Dengan ketentuan sebagai berikut:

1. Terbiasa menggunakan aplikasi yang ada di sistem operasi Windows.
2. Memiliki pengetahuan tentang *database*.
3. Mengetahui atribut yang dianggap kuat untuk dilibatkan dalam proses *Data mining*

3.1.4.2 Analisis Kebutuhan Perangkat keras (*Hardware*)

Berdasarkan analisis yang telah dilakukan, dapat diketahui spesifikasi perangkat keras yang digunakan jurusan Teknik Komputer adalah sebagai berikut:

a. Processor 1.8 GHz
b. Harddisk 80 GB
c. RAM 504 MB
d. Monitor 14 inch dengan kualitas warna 32 bit.

Kebutuhan *hardware* yang disarankan untuk menjalankan spesifikasi ini minimal memiliki spesifikasi sebagai berikut:

a. Processor 1,5 GHz
b. Harddisk 20 GB
c. RAM 512 MB
d. Monitor yang mendukung kualitas warna 16 bit dengan resolusi 800 x 600 *pixels*.

Berdasarkan spesifikasi yang telah ada, secara keseluruhan kebutuhan perangkat keras untuk aplikasi ini telah terpenuhi.

3.1.4.3 Analisis Perangkat Lunak (Software)

Perangkat Lunak yang dikembangkan dalam Aplikasi *Data Mining* ini berbasis *desktop* dengan *database* lokal. Perangkat lunak yang disarankan untuk menjalankan aplikasi ini adalah sebagai berikut:

a. Sistem Operasi

   Untuk sistem operasi windows disarankan, karena sistem operasi ini banyak dikenal oleh *user* awam dan lebih mudah dipelajari.

b. Microsoft Office

c. Database Management System

Database Management System (DBMS) adalah sistem untuk mengelola basisdata yang digunakan. Untuk aplikasi *Data mining* ini digunakan Microsoft Access sebagai DBMS.

3.1.5 Spesifikasi Kebutuhan Fungsional

Spesifikasi kebutuhan fungsional pada Aplikasi *Data Mining* ini merujuk pada kebutuhan akan perancangan *data mining*, seperti berikut ini :

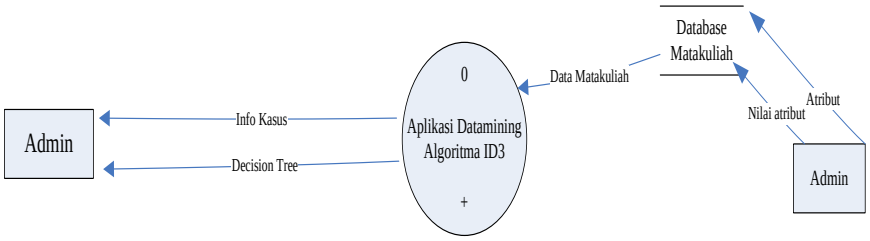
1. Dapat menggabungkan data yang akan diproses *mining* dari data matakuliah dan data mahasiswa.
2. Dapat menghapus data-data yang tidak relevan serta atribut yang tidak dipakai .
3. Dapat merubah data menjadi data yang siap diproses.
4. Dapat memproses data untuk di*mining.*
5. Dapat menampilkan hasil proses *mining* dengan nilai *support* dan *confidence.*

3.1.6 Pemodelan Fungsi

Pemodelan fungsi digambarkan dengan DCD *(Data Context Diagram)*, DFD *(Data Flow Diagram)* dan kamus data *(Data Dictionary).*

3.1.6.1 Diagram Konteks

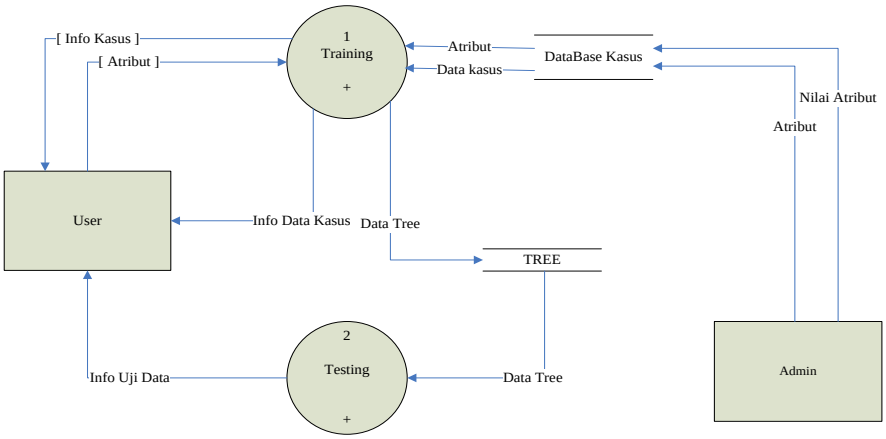
Diagram konteks ini menggambarkan suatu sistem secara garis besarnya, yang menjelaskan bagian-bagian umum yang terdapat dalam sistem.



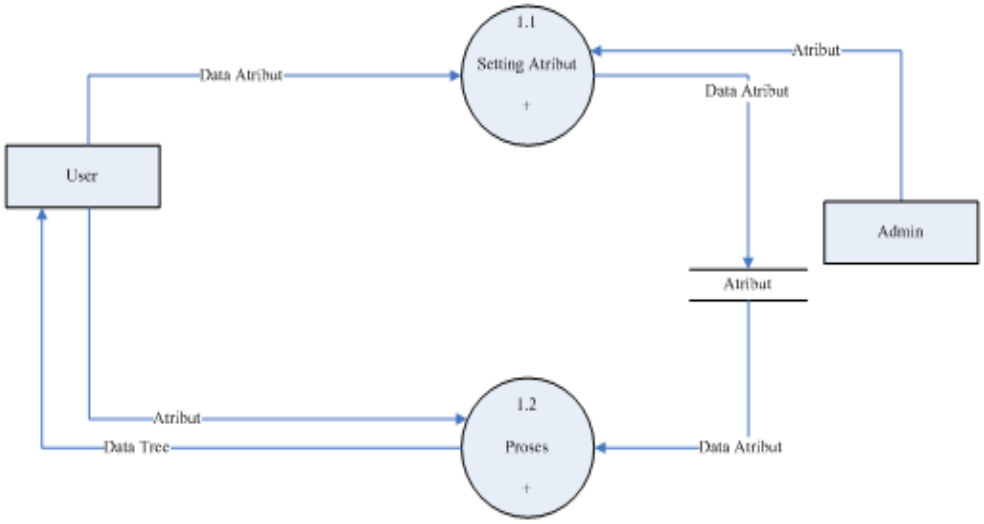
Gambar 3 Diagram Konteks Aplikasi Data Mining

3.1.6.2 Diagram Alir Data Level 1

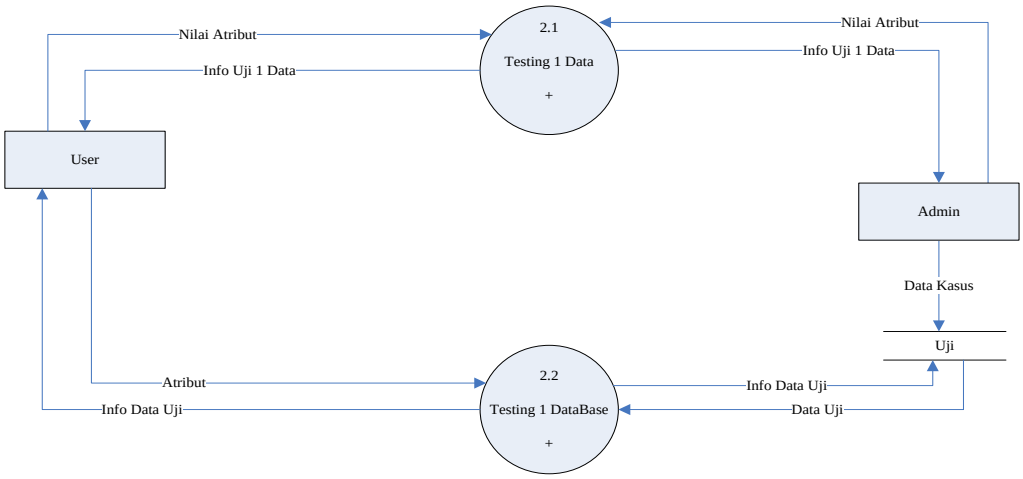
Diagram ini menggambarkan bagian-bagian yang saling berhubungan, yang terdapat dalam sistem. Dapat dilihat diagram alir data level1 sebagai berikut:



Gambar 4 DFD Level 1 Proses Data Mining



Gambar 5 DFD Level 1 Proses Data Mining



Gambar 6 DFD Level 2 Proses Data Mining

3.1.7 Spesifikasi Proses

Pada spesifikasi proses ini menjelaskan proses-proses yang terdapat dalam diagram aliran data. Dapat dilihat dalam tabel sebagai berikut:

Tabel 5 Tabel Proses

No.	Proses	Keterangan
1.	No. Proses	1
	Nama Proses	*Proses Training*
	Sumber	*User*, Tabel DataBase Kasus
	Input	*Atribut*, Data Kasus
	Output	Info Data Kasus, Data Tree
	Deskripsi	Melakukan proses *Training* untuk menghasilkan pohon keputusan
	Logika Proses	Melakukan proses *training* data kasus
2.	No. Proses	2

	Nama Proses	*Proses Testing*
	Sumber	*User,* Tabel TREE
	Input	Data *Tree*
	Output	Info Uji Data
	Deskripsi	Melakukan proses *testing* (pengujian) data kasus
	Logika Proses	*User* memilih menu testing
3.	No. Proses	1.1
	Nama Proses	Seting Atribut
	Sumber	*User,* Tabel KASUS
	Input	Atribut
	Output	Data Atribut
	Deskripsi	Melakukan pemilihan atribut yang akan digunakan untuk menentukan atribut tujuan
	Logika Proses	Adodc3.Recordset.Requery h = Adodc3.Recordset.RecordCount For q = 1 To h Adodc3.Recordset.Requery Adodc3.Recordset.Delete Adodc3.Recordset.Requery Next Adodc1.Recordset.Requery total = Adodc1.Recordset.RecordCount Adodc1.Recordset.Filter = "masa_studi='0'" Adodc1.Recordset.Requery TotalKurang = Adodc1.Recordset.RecordCount Adodc1.Recordset.Filter = "masa_studi='1'" Adodc1.Recordset.Requery TotalLebih = Adodc1.Recordset.RecordCount Entropy = ((-1 * (TotalKurang / total)) *

		(Log(TotalKurang / total) / Log(2))) + ((-1 * (TotalLebih / total)) * (Log(TotalLebih / total) / Log(2)))
		gainfisika1 = Entropy
		gainkalkulus1 = Entropy
		gainpskom = Entropy
		gainfisika2 = Entropy
		gainkalkulus2 = Entropy
		gainalgo1 = Entropy
		Adodc1.Recordset.Filter = "masa_studi<>'hjhj'"
		Adodc1.Recordset.Requery
		p = 1
		Do Until lup = "stop"
		Adodc1.Recordset.Requery
		total = Adodc1.Recordset.RecordCount
		Adodc1.Recordset.MoveNext
		Next
		If total <> 0 Then
		If a1.Value = Checked Then
		If (hil1 <> "F1") And (hil2 <> "F1") And (hil6 <> "F1") And (hil3 <> "F1") And (hil4 <> "F1") And (hil5 <> "F1") Then
		tambah = "yes"
		hasil1 = ""
		hasil2 = ""
		hasil3 = ""
		hasil4 = ""
		hasil5 = ""
		If Fisika1AKurang = 0 Then
		EntropiFisika1A = 0
		hasil1 = "Lebih"
		ElseIf Fisika1ALebih = 0 Then
		EntropiFisika1A = 0
		hasil1 = "Kurang"
		Else
		EntropiFisika1A = ((-1 * (Fisika1AKurang /

		fisika1A)) * (Log(Fisika1AKurang / fisika1A) / Log(2))) + ((-1 * (Fisika1ALebih / fisika1A)) * (Log(Fisika1ALebih / fisika1A) / Log(2)))
		End If
		gainfisika1 = gainfisika1 - ((fisika1A / total) * EntropiFisika1A
		If a3.Value = Checked Then
		If (hil1 <> "Or") And (hil2 <> "Or") And (hil6 <> "Or") And (hil3 <> "Or") And (hil4 <> "Or") And (hil5 <> "Or") Then
		tambah = "yes"
		hasil1 = ""
		hasil2 = ""
		hasil3 = ""
		hasil4 = ""
		hasil5 = ""
		If pskomAKurang = 0 Then
		EntropipskomA = 0
		hasil1 = "Lebih"
		ElseIf pskomALebih = 0 Then
		EntropipskomA = 0
		hasil1 = "Kurang"
		Else
		EntropipskomA = ((-1 * (pskomAKurang / pskomA)) * (Log(pskomAKurang / pskomA) / Log(2))) + ((-1 * (pskomALebih / pskomA)) * (Log(pskomALebih / pskomA) / Log(2)))
		End If
		gainpskom = gainpskom - ((pskomA / total) * EntropipskomA)
		If pskomBKurang = 0 Then
		EntropipskomB = 0
		hasil2 = "Lebih"
		ElseIf pskomBLebih = 0 Then
		EntropipskomB = 0
		hasil2 = "Kurang"
		Else

		EntropipskomB = ((-1 * (pskomBKurang / pskomB)) * (Log(pskomBKurang / pskomB) / Log(2))) + ((-1 * (pskomBLebih / pskomB)) * (Log(pskomBLebih / pskomB) / Log(2)))
		End If
		gainpskom = gainpskom - ((pskomB / total) * EntropipskomB)
		If pskomCKurang = 0 Then
		EntropipskomC = 0
		hasil3 = "Lebih"
		ElseIf pskomCLebih = 0 Then
		EntropipskomC = 0
		hasil3 = "Kurang"
		Else
		EntropipskomC = ((-1 * (pskomCKurang / pskomC)) * (Log(pskomCKurang / pskomC) / Log(2))) + ((-1 * (pskomCLebih / pskomC)) * (Log(pskomCLebih / pskomC) / Log(2)))
		End If
		gainpskom = gainpskom - ((pskomC / total) * EntropipskomC)
		If pskomDKurang = 0 Then
		EntropipskomD = 0
		hasil4 = "Lebih"
		ElseIf pskomDLebih = 0 Then
		EntropipskomD = 0
		hasil4 = "Kurang"
		Else
		EntropipskomD = ((-1 * (pskomDKurang / pskomD)) * (Log(pskomDKurang / pskomD) / Log(2))) + ((-1 * (pskomDLebih / pskomD)) * (Log(pskomDLebih / pskomD) / Log(2)))
		End If
		gainpskom = gainpskom - ((pskomD / total) * EntropipskomD)
		gainalgo1 = gainalgo1 - ((algo1C / total) * EntropiAlgo1C)
		If algo1DKurang = 0 Then

		EntropiAlgo1D = 0
		hasil4 = "Lebih"
		ElseIf algo1DLebih = 0 Then
		EntropiAlgo1D = 0
		hasil4 = "Kurang"
		Else
		EntropiAlgo1D = ((-1 * (algo1DKurang / algo1D)) * (Log(algo1DKurang / algo1D) / Log(2))) + ((-1 * (algo1DLebih / algo1D)) * (Log(algo1DLebih / algo1D) / Log(2)))
		End If
		gainalgo1 = gainalgo1 - ((algo1D / total) * EntropiAlgo1D)
		If algo1EKurang = 0 Then
		EntropiAlgo1E = 0
		hasil5 = "Lebih"
		ElseIf algo1ELebih = 0 Then
		EntropiAlgo1E = 0
		hasil5 = "Kurang"
		Else
		EntropiAlgo1E = ((-1 * (algo1EKurang / algo1E)) * (Log(algo1EKurang / algo1E) / Log(2))) + ((-1 * (algo1ELebih / algo1E)) * (Log(algo1ELebih / algo1E) / Log(2)))
		End If
		gainalgo1 = gainalgo1 - ((algo1E / total) * EntropiAlgo1E)
		End If
		End If
		End If

4.	No. Proses	1.2
	Nama Proses	Proses
	Sumber	*Tabel Atribut*

	Input	*Data Atribut*
	Output	Info Data Tree

	Logika Proses	Algorithm : Generete_decision_tree.
		Narrative : Generate a decision tree from the given training data.
		Input : The training samples, samples, represented by discrete-valued attribute, the set of candidate attributes, attribute-list.
		Output : A dicision tree.
		
		Method
		1) Craete a node *N*;
		2) If *samples* are all of the same class, C then
		3) Return N as a leaf node labeled with class C
		4) If attribute-list is empty then
		5) Return N as leaf node labeled with the most common class in samples; // majority voting
		6) Selsct test-attribute, the attribute among attribute-list with the highest gain ratio;
		7) Label node N with last attribute;
		8) For each known value a, of test-attribute
		9) Grow a branch from node N for the condition test-attribute = a_i
		10) Let S_i be the set of samples in samples for wichh test-attribute = a_i // a partition
		11) If S_i is empty then
		12) attach a leaf labeled with the most common class in *sample*;
		Else attach the node returned by
		Generate_decision_tree (S_i, attribute-list-list-attribute);

5.	No.Proses	2.1
	Nama Proses	Testing 1 Data
	Source	Nilai Atribut

	Input	Nilai Atribut
	Output	Info Uji Data
	No.Proses	2.2
	Nama Proses	Testing 1 Database
	Sumber	Tabel Uji
	Input	Data Uji Atribut
	Output	Info Data Uji 1 Database

3.2 Perancangan Sistem

3.2.1 Perancangan Data

Perancangan data atau perancangan *database* pada tugas akhir ini menggunakan *microsoft access* 2007, dengan nama databasenya adalah ”tugas”. Berikut ini nama tabel yang digunakan beserta field-field yang terdapat pada tabel berikut:

1. Tabel Akhir, tabel akhir ini adalah tabel *training* yang berfungsi untuk menyimpan data matakuliah beserta masa studi mahasiswa yang telah lulus.

Tabel 6 Tabel Akhir untuk Data Training

No	Field Name	Data Type	Field Size	keterangan
1	No	AutoNumber	Long Integer	Primary Key
2	Nama Matakuliah	Teks	50	-
3	Masa Studi	Teks	50	-

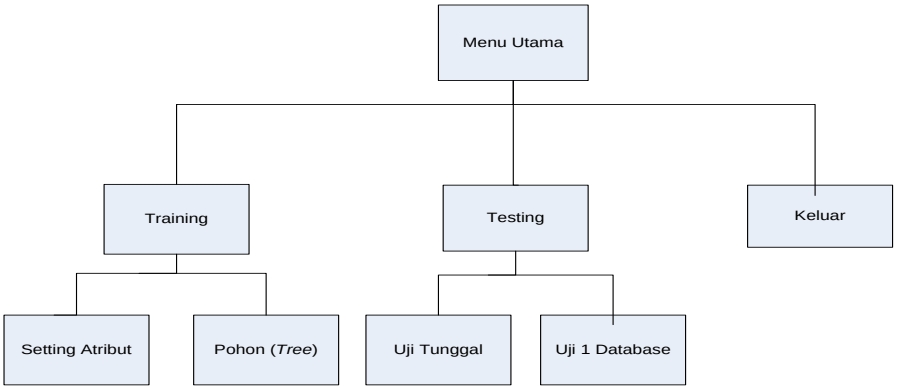
2. Tabel Uji, tabel ini adalah tabel *testing* yang berfungsi untuk menguji data yang terdapat dalam data base dengan algoritma yang digunakan.

Tabel 7 Tabel Uji untuk Data Testing

No	Field Name	Data Type	Field Size	keterangan
1	No	AutoNumber	Long Integer	Primary Key
2	Nama Matakuliah	Teks	50	-
3	Masa Studi	Teks	50	-

3.2.2 Arsitektur Menu

Berikut ini adalah perancangan menu yang terdapat dalam sistem aplikasi prediksi masa studi mahasiswa:



Gambar 7 Arsitektur Menu

3.2.3 Perancangan Antar Muka

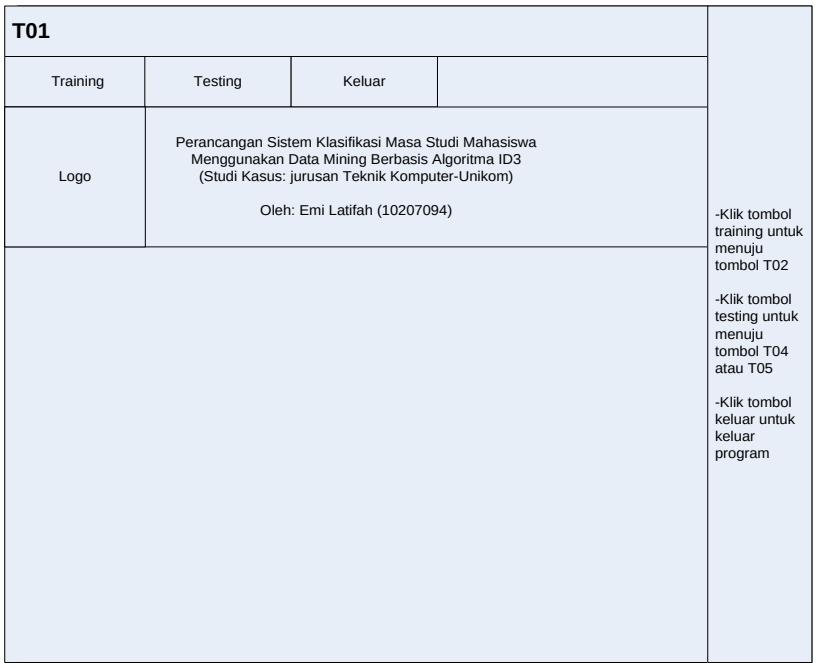
Perancangan antar muka merupakan suatu bentuk tampilan dari program yang akan dibuat untuk kebutuhan *interface* dengan *user*. Spesifikasi antarmuka terdiri dari perancangan tampillan menu, tampilan form, tampilan pesan, dan keluaran

3.2.4 Perancangan Form

Dalam perancangan form di sini dilakukan perancangan baik terhadap form input dan form output. Adapun form-form yang terdapat dalam aplikasi data mining ini adalah sebagai berikut.

1. Form Utama

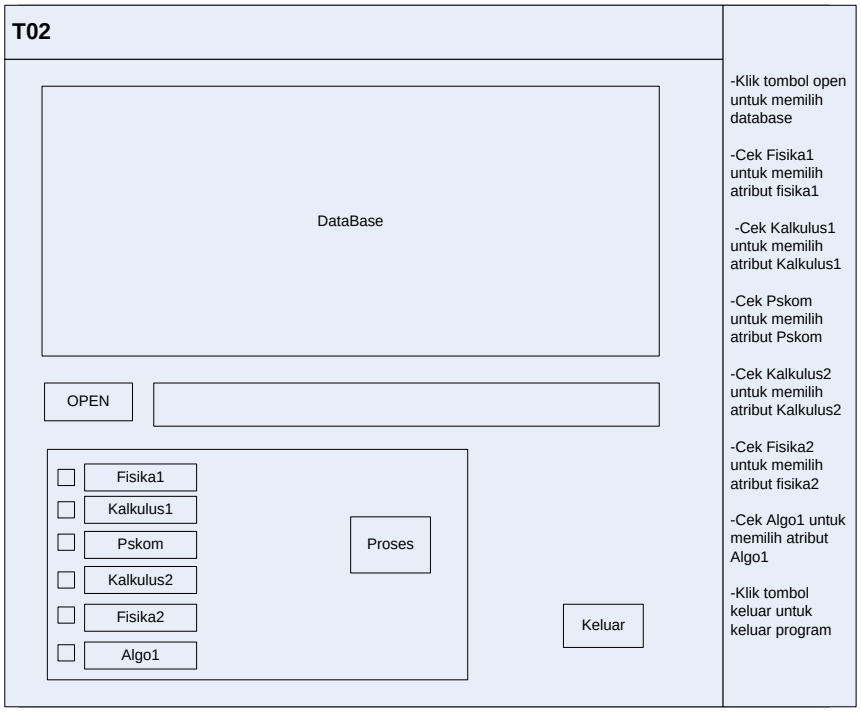
Tampilan form utama merupakan tampilan utama ketika sistem aplikasi data mining dijalankan, yang terdiri dari beberapa form-form dapat dilihat sebagai berikut:



Gambar 8 Gambar Form Utama pada Aplikasi Data Mining

2. Form Pemilihan Atribut dan Proses

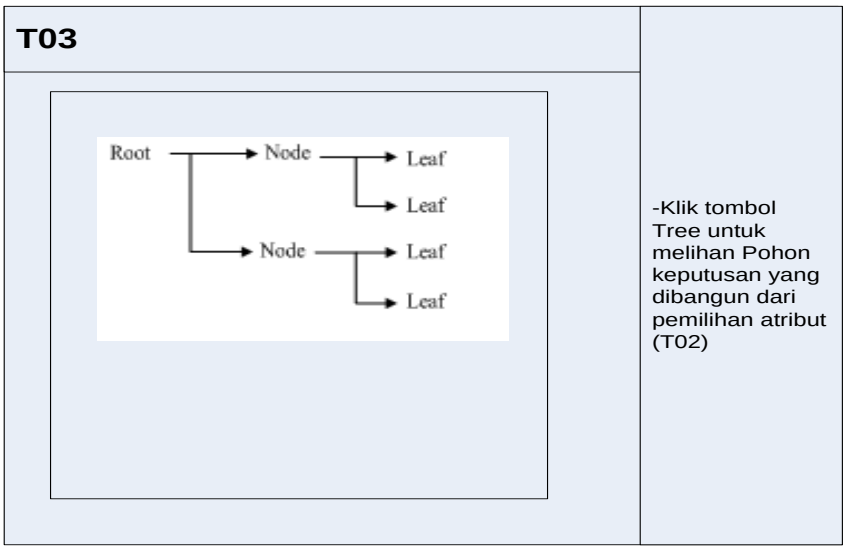
   Pada form ini menampilkan 6 atribut yang harus dipilih oleh *user*, setelah beberapa atribut dipilih kemudian diproses sesuai dengan perhitungan algoritma ID3 yang digunakan yang nantinya akan menghasilkan suatu pohon keputusan. Berikut tampilannya:



Gambar 9 Form Pemilihan Atribut dan Proses

3. Form *Tree*

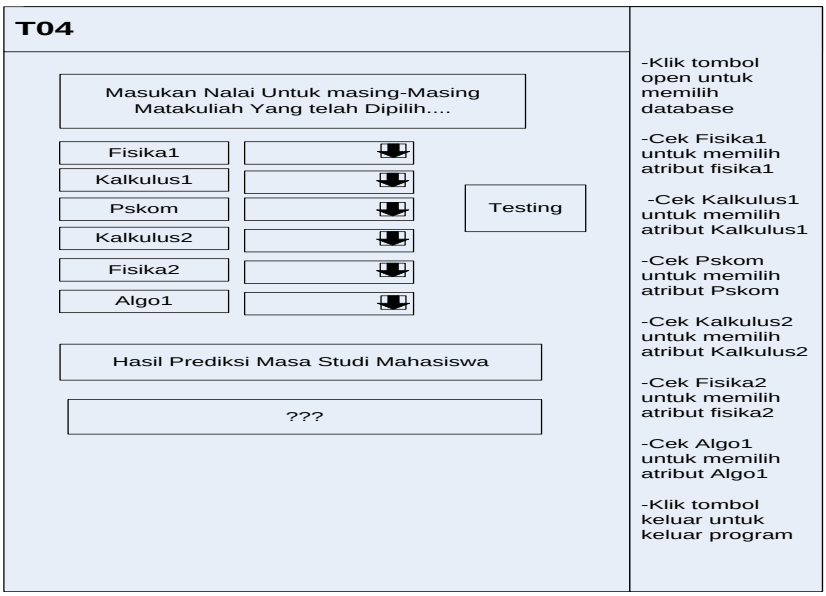
Pada form ini menampilkan pohon keputusan mulai dari akar sampai kecabang-cabangnya hingga menemukan suatu keputusan masa studi mahasiswa yang lebih dari 5 tahun dan kurang dari 5 tahun.



Gambar 10 Form Decision Tree pada Aplikasi Data Mining

4. Form Pengujian Tunggal

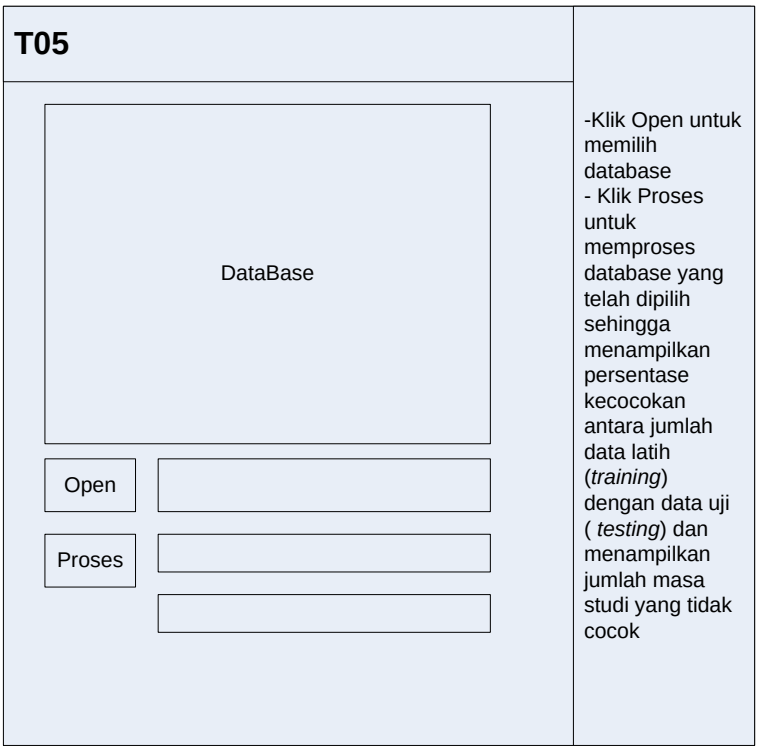
Pada form ini terdapat pengujian tunggal untuk beberapa atribut dimana hasil dari pengujian tersebut sesuai dengan *tree* yang telah dibuat.



Gambar 11 Form Pengujian Tunggal pada Aplikasi DataMining

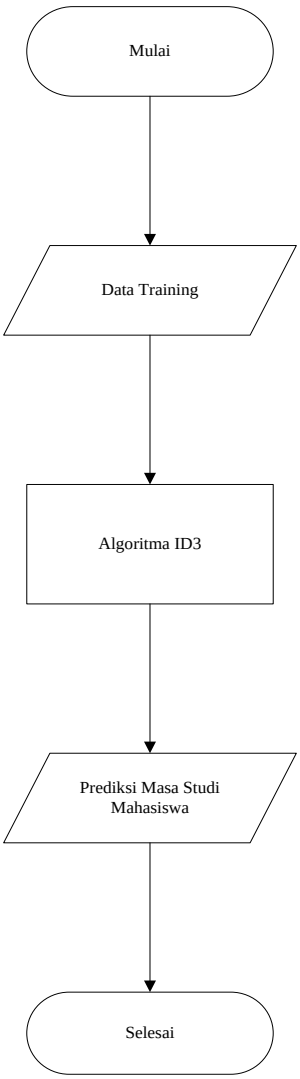
5. Form Pengujian DataBase

Pada form ini terdapat pengujian dengan *database* dimana pengujian tersebut dapat menguji dengan cara membuka *database* baru kemudian diproses hasil perhitungannya dalam bentuk persentase (perbandingan) antara data base dengan perhitungan algoritma *ID3* dengan *database* asli.



Gambar 12 Form Pengujian 1 Data Base

3.2.5 Flowchart



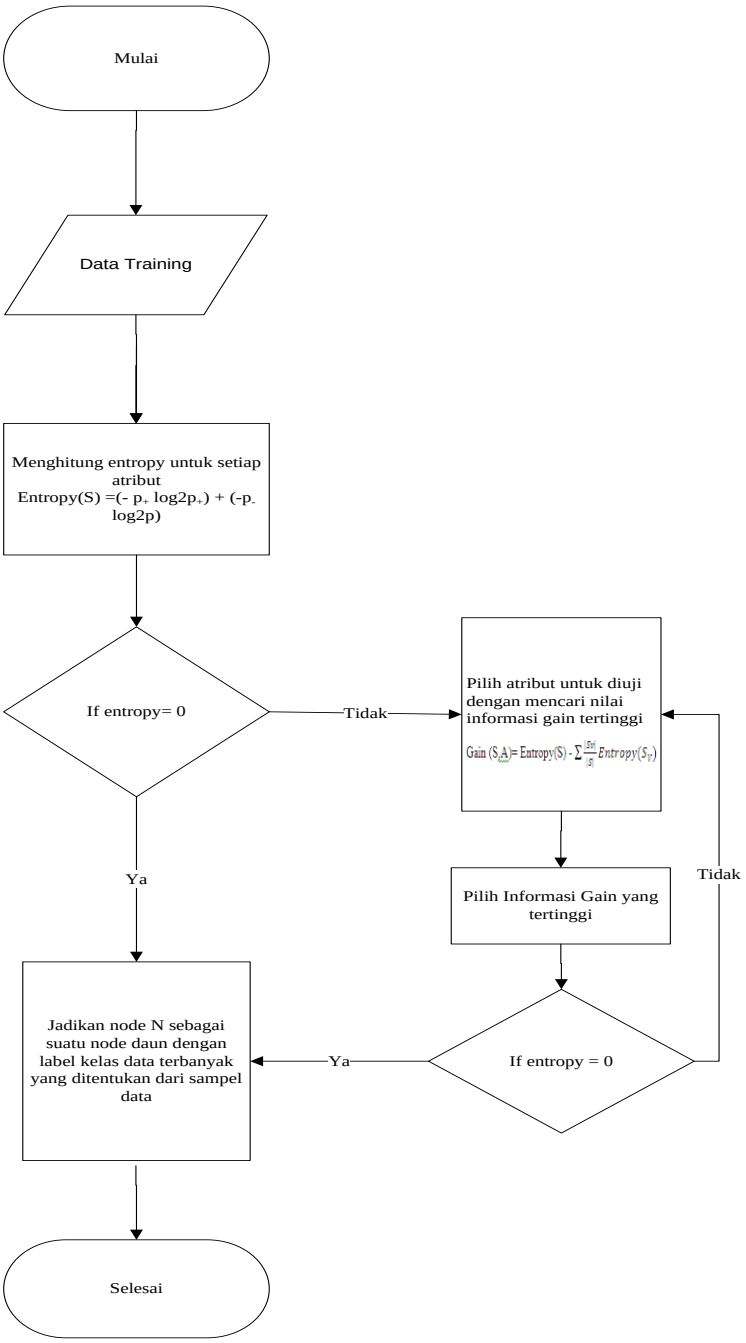
Gambar 13 Flowchart Aplikasi Sistem

Secara garis besar proses aplikasi sistem diatas adalah sebagai berikut:

1. Input data *training*.
2. Masuk ke proses algoritma untuk dihitung berdasarkan langkah-langkah algoritma, hasilnya adalah *tree*.
3. Setelah *tree* terbentuk akan dibuat aturan berdasarkan *tree*, dengan ketentuan *if-then*.

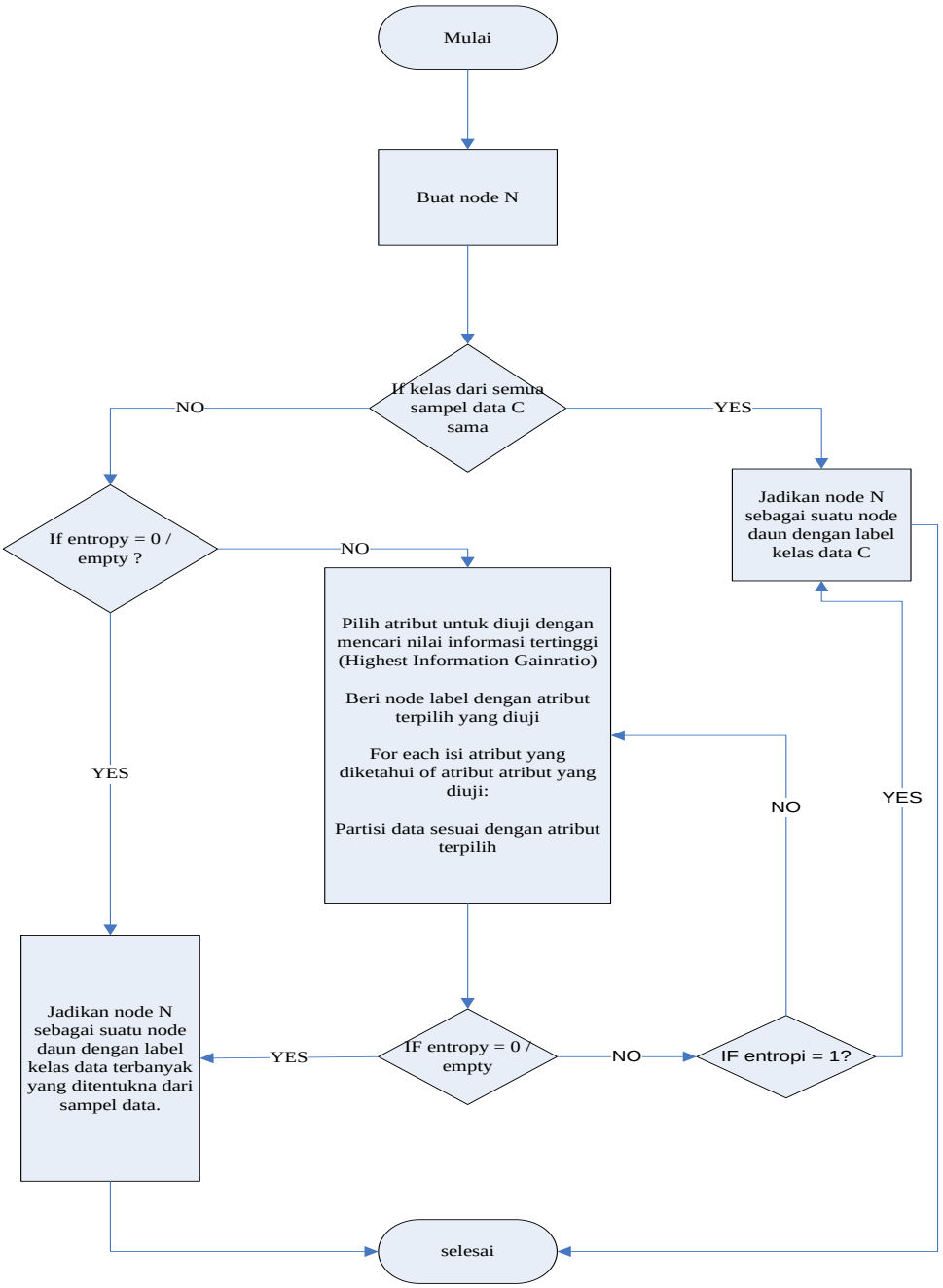
4. Dari aturan yang ada akan dilakukan proses pengujian dengan data testing berdasarkan pohon keputusan yang telah terbentuk.
5. Hasil dari prediksi prohon keputusan tersebut berupa hasil prediksi.

3.2.6 Flowchart Pencarian *Entropy* dan *Gain*



Gambar 14 Flowchart Pencarian Entropy dan Gain

3.2.7 Flowchart Pembuatan Pohon Keputusan



Gambar 15 Flowchart Algoritma ID3

Secara garis besar langkah-langkah yang dilakukan algoritma ID3 untuk membentuk sebuah pohon keputusan adalah sebagai berikut:

1. Pada saat awal pembentukan pohon akan dimulai dengan membuat suatu *node* yang melambangkan *training* sampel.
2. Jika sampel-sampel tersebut memiliki kelas yang sama, maka node tersebut dijadikan *leaf node* dengan label kelas tersebut.
3. Jika sampel-sampel tersebut tidak memiliki kelas yang sama maka algoritma ini akan mencari *gain* tertinggi dari atribut yang tersedia, sebagai cara untuk memilih atribut yang berpengaruh pada *training* sampel yang tersedia.
4. Cabang untuk setiap node akan dibentuk berdasarkan nilai-nilai yang diketahui dari atribut pengujian.
5. Algoritma ini akan terus melakukan proses yang sama untuk membentuk suatu pohon keputusan.