BAB III
PERANCANGAN SISTEM

3.1 Analisis Masalah

Proses memprediksi mahasiswa merupakan suatu proses untuk mengetahui masa studi mahasiswa dengan menggunakan atribut matakuliah. Data-data dari atribut tersebut pada kenyataanya masih banyak mengalami *missing value*. Hal ini dikarenakan banyak data yang hilang pada jurusan.

Cara mengatasi yaitu dengan menyeleksi data-data tersebut dari data yang mengalami *missing value* dengan cara menghilangkan data tersebut sehingga data yang diolah adalah data lengkap tanpa *missing value.*

Dikarenakan data yang diolah atau data yang menjadi data training adalah data lengkap tanpa *missing value* maka proses perhitungan algoritmanya berdasarkan perhitungan gain tanpa *missing value* yang telah dibahas pada bab 2.

3.2 Analisis Data

Dalam penulisan tugas akhir ini akan dicari nilai *Gain* dari atribut-atribut matakuliah. Hasil dari proses tersebut dapat membantu untuk mendapatkan in*form*asi berapa lama masa studi mahasiswa S1 di jurusan Teknik Komputer.

3.3 Sumber Data

Dari data yang ada, kolom yang diambil sebagai atribut atau variabel keputusan adalah kolom masa studi mahasiswa, yang digunakan sebagai variabel penentu dalam pembentukan pohon keputusan adalah:

1. Algoritma Pemrograman 1
2. Fisika Dasar 1
3. Fisika Dasar 2
4. Kalkulus 1

5. Kalkulus 2
6. Pengantar Sistem Komputer
7. Masa Studi

Dari Proses di atas dihasilkan tabel yang siap untuk proses Klasifikasi, dengan atribut dan nilai atribut yang ada pada tabel III-1.

Table III-1 Atribut

Algoritma Pemrograman 1	**Fisika Dasar 1**	**Fisika Dasar 2**	**Kalkulus 1**	**Kalkulus 2**	**Pengantar Sistem Komputer**	**Masa Studi**
A	A	A	A	A	A	<= 5 TAHUN
B	B	B	B	B	B	> 5 TAHUN
C	C	C	C	C	C	
D	D	D	D	D	D	
E	E	E	E	E	E	

3.4 Pembersihan Data

Proses pembersihan data pada tugas akhir di sini adalah pembersihan data yang bersifat manual. Dimana proses pembersihan data dilakukuan di luar aplikasi. Data dibersihkan dari data yang memilki *missing value*. Maksud dari *Missing value* di sini contohnya adalah data yang tidak lengkap. Maksud dari data tidak lengkap contohnya ada field yang kosong. Dan apabila ada field yang kosong maka data tersebut akan dihilangkan secara manual. Atau contoh lain dari *missing value* adalah data yang isi dari fieldnya jauh dari data yang seharusnya. Misalkan field mata kuliah Algoritma yang berisi nilai-nilai yaitu A, B, C, D, dan E tetapi ada data nilai F atau G maka data tersebut akan dihilangkan secara manual.

3.5 Analisis Algoritma *C4.5*

Pada tabel di bawah ini implementasi perhitungan kasus algoritma C4.5 untuk menentukan masa studi mahasiswa berdasarkan 3 nilai mata kuliah yaitu algoritma, fisika 1 dan fisika 2. Dengan nilai masing-masing mata kuliah adalah A, B, C, D, dan E. Mata kuliah ini dijadikan atribut awal dalam implementasi

algoritma C4.5. dan aribut tujuannya adalah masa studi dengan nilai dari atribut masa studi adalah kurang dan lebih. Kurang artinya lulus kurang dari lima tahun dan lebih artinya lulus lebih dari lima tahun.

Table III-2 Contoh Kasus Algoritma C4.5

no	algoritma	fisika1	fisika2	masa studi
1	A	A	B	Kurang
2	B	B	B	Kurang
3	C	C	B	Kurang
4	B	A	A	Lebih
5	D	C	C	Lebih
6	A	A	A	Kurang
7	E	C	B	Lebih
8	D	C	D	Lebih
9	D	C	C	Lebih
10	A	B	D	Lebih
11	E	D	C	Lebih
12	B	C	D	Lebih
13	E	C	D	Lebih
14	C	E	E	Lebih
15	C	D	A	Kurang

Table III-3 Perhitungan mencari root

NODE			**jlm_kasus**	**Kurang**	**lebih**	**info(x)**	**E(x)**	**Gain**
1	jml_semua		15	5	10	0,9183		
	Algoritma						0,551	0,3673
		A	3	2	1	0,9183		
		B	3	1	2	0,9183		
		C	3	2	1	0,9183		
		D	3	0	3	0		
		E	3	0	3	0		
	fisika1						0,7264	0,1919
		A	3	2	1	0,9183		
		B	2	1	1	1		
		C	7	1	6	0,59167		
		D	2	1	1	1		
		E	1	0	1	0		
	fisika2						0,4	0,5183

A	3	2	1	0,9183
B	4	3	1	0,81128
C	3	0	3	0
D	4	0	4	0
E	1	0	1	0

Dari tabel di atas terdapat gain untuk masing-masing mata kuliah. Dimana hasil gain dari masing-masing mata kuliah tersebut diantaranya adalah:

1. Algoritma = 0,3673
2. Fisika 1 = 0,1919
3. Fisika 2 = 0,5183

Sesuai dengan ketentuan dalam algoritma C4.5 dimana ketentuannya adalah setiap atribut yang memiliki gain tertinggi maka akan menjadi *root* atau *node*. Dan dari ketiga mata kuliah tersebut, matakuliah yang memiliki gain tertinggi adalah Fisika 2. Maka mata kuliah Fisika 2 menjadi *root.*

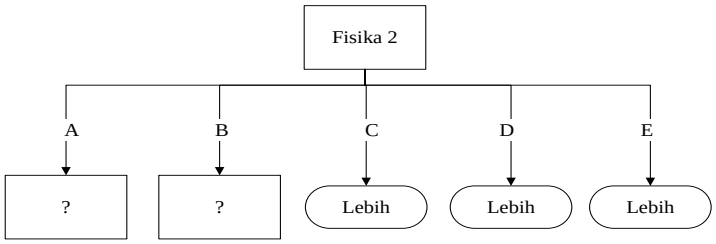
Setelah diperoleh *root* maka akan dilihat infoa(x) dari masing nilai atribut mata kuliah yang jadi *root* yaitu Fisika 2. Dari tabel di atas, nilai infoa(x) untuk nilai mata kuliah Fisika 2 diantaranya adalah:

1. Fisika 2 A : 0,9183
2. Fisika 2 B : 0,81128
3. Fisika 2 C : 0
4. Fisika 2 D : 0
5. Fisika 2 E : 0

Dari hasil infoa(x) tersebut yang masih memiliki nilai seperti Fisika 2 A, dan Fisika 2 B artinya nilai tersebut masih memiliki cabang dan harus dilakukan perhitungan kembali. Namun untuk nilai Fisika 2 C, Fisika 2 D, dan Fisika 2 E memiliki nilai infoa(x) = 0, yang artinya bahwa untuk nilai dari atribut tersebut sudah mencapai keputusan akhir atau *leaf*. Dan hasil keputusannya dapat dilihat dari jumlah yang masa studinya kurang dan lebih. Untuk nilai atribut Fisika C jumlah yang masa studinya kurang adalah 0 dan yang masa studinya lebih adalah 3, artinya keputusan akhir dari cabang tersebut adalah lebih karena nilai masa

studi lebih mempunyai nilai. Jadi cabang dari Fisika C adalah Lebih. Demikian juga untuk nilai atribut yang Fisika 2 D dan Fisika 2 E masa studi lebih memiliki nilai, sehingga leaf dua-duanya berisi lebih.

Dan hasil *tree* dari perhitungan yang pertama adalah sebagai berikut:



Gambar III-1 Root Dan Cabangnya

Karena nilai untuk atribut Fisika 2 A dan Fisika 2 B masih memili nilai maka harus dilakukan perhitungan kembali untuk menentukan node cabang Fisika2 A dan Fisika2 B.

Pada tabel dibawah ini adalah perhitungan untuk cabang Fisika 2 A.

Table III-4 Perhitungan Cabang Fisika 2 A

NODE			**Jlm_kasus**	**Kurang**	**Lebih**	**infoa(x)**	**E(x)**	**Gain**
2	fisika2(A)		3	2	1	0,9183		
	Algoritma						0	0,9183
		A	1	1	0	0		
		B	1	0	1	0		
		C	1	1	0	0		
		D	0	0	0	0		
		E	0	0	0	0		
	fisika1						0,6667	0,2516
		A	2	1	1	1		
		B	0	0	0	0		
		C	0	0	0	0		
		D	1	1	0	0		
		E	0	0	0	0		

Dari tabel di atas terdapat gain untuk masing-masing mata kuliah. Dimana hasil gain dari masing-masing mata kuliah tersebut diantaranya adalah:

1. Algoritma = 0,9183
2. Fisika 1 = 0,2516

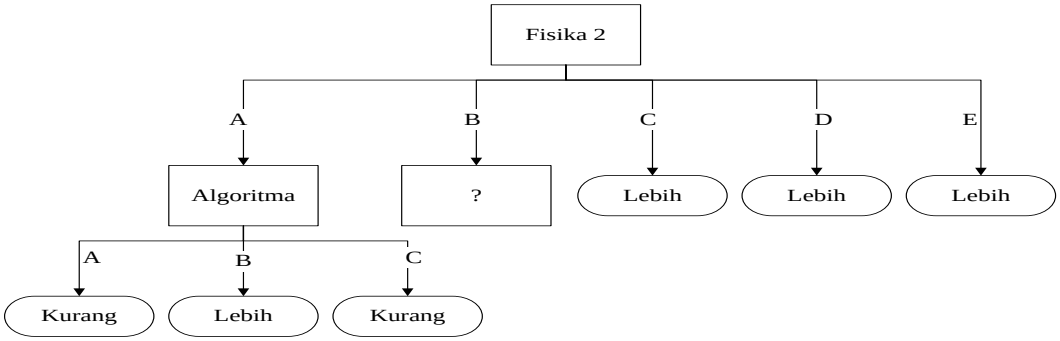
Dan dari kedua mata kuliah tersebut, matakuliah yang memiliki gain tertinggi adalah Algoritma, maka Algoritma akan menjadi *node* cabang dari Fisika 2 A.

Setelah diperoleh *node* cabang Fisika 2 A maka akan dilihat infoa(x) dari masing nilai atribut mata kuliah yang jadi *node* cabang Fisika 2 A yaitu Algoritma. Dari tabel di atas, nilai infoa(x) untuk nilai mata kuliah Algoritma diantaranya adalah:

1. Algoritma A : 0
2. Algoritma B : 0
3. Algoritma C : 0
4. Algoritma D : 0
5. Algoritma E : 0

Dari hasil infoa(x) tersebut semua bernilai 0, ini artinya semua cabang dari nilai mata kuliah Algoritma sudah mencapai keputusan akhir atau mencapai *leaf*. Dari data tabel di atas terlihat untuk nilai atribut Algoritma A, B, dan C memiliki jumlah data, namun untuk nilai atribut Algoritma yang D dan E bernilai 0 ini artinya dalam data training tidak ada data yang menunjukan Fisika dasarnya A dan Algoritmanya D dan E. Sehingga untuk cabang Algoritma D dan E tidak ada dalam *tree*. Sedangkan untuk cabang Algoritma yang A, B dan C sudah mencapai *leaf* atau keputusan akhir, dimana keputusan akhirnya yaitu untuk Algoritma yang A adalah kurang, Algoritma yang B adalah lebih dan Algoritma yang C adalah kurang.

Hasil dari perhitungan tabel di atas:



Gambar III-2 Cabang Fisiska 2 A

Pada tabel dibawah ini adalah perhitungan untuk cabang Fisika 2 B.

Table III-5 Perhitungan Cabang Fisika 2 B

NODE		**jlm_kasus**	**Kurang**	**lebih**	**infoa(x)**	**E(x)**	**Gain**
2	fisika2(B)	4	3	1	0,81128		
	Algoritma					0	0,8113
	A	1	1	0	0		
	B	1	1	0	0		
	C	**1**	1	0	0		
	D	0	0	0	0		
	E	1	0	1	0		
	fisika1					0,5	0,3113
	A	1	1	0	0		
	B	1	1	0	0		
	C	2	1	1	1		
	D	0	0	0	0		
	E	0	0	0	0		

Dari tabel di atas terdapat gain untuk masing-masing mata kuliah. Dimana hasil gain dari masing-masing mata kuliah tersebut diantaranya adalah:

1. Algoritma = 0,8113
2. Fisika 1 = 0,3113

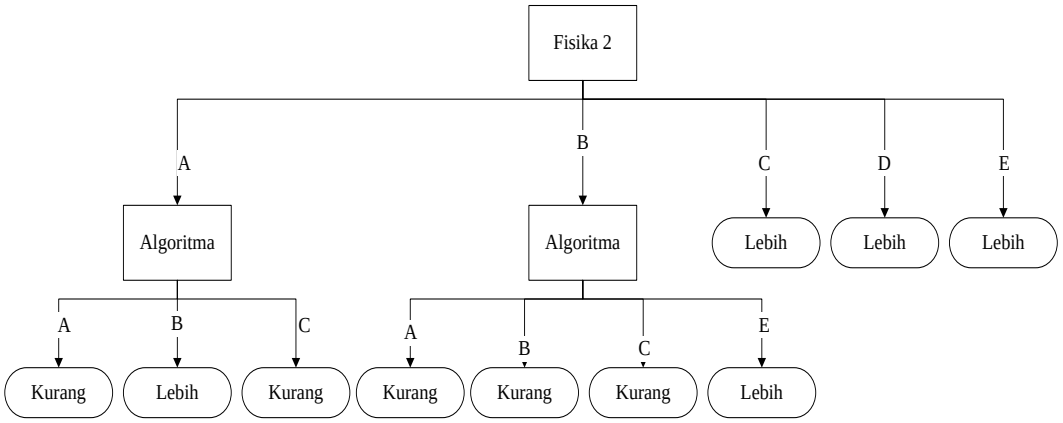
Dan dari kedua mata kuliah tersebut, matakuliah yang memiliki gain tertinggi adalah Algoritma, maka Algoritma akan menjadi *node* cabang dari Fisika 2 B.

Setelah diperoleh *node* cabang Fisika 2 B maka akan dilihat infoa(x) dari masing nilai atribut mata kuliah yang jadi *node* cabang Fisika 2 B yaitu Algoritma. Dari tabel di atas, nilai infoa(x) untuk nilai mata kuliah Algoritma diantaranya adalah:

1. Algoritma A : 0
2. Algoritma B : 0
3. Algoritma C : 0
4. Algoritma D : 0
5. Algoritma E : 0

Dari hasil infoa(x) tersebut semua bernilai 0, ini artinya semua cabang dari nilai mata kuliah Algoritma sudah mencapai keputusan akhir atau mencapai *leaf*. Dari data tabel di atas terlihat untuk nilai atribut Algoritma A, B, C dan E memiliki jumlah data, namun untuk nilai atribut Algoritma yang D bernilai 0 ini artinya dalam data training tidak ada data yang menunjukan Fisika dasarnya A dan Algoritmanya D. Sehingga untuk cabang Algoritma D tidak ada dalam *tree*. Sedangkan untuk cabang Algoritma yang A, B, C dan D sudah mencapai *leaf* atau keputusan akhir, dimana keputusan akhirnya yaitu untuk Algoritma yang A adalah kurang, Algoritma yang B adalah kurang, Algoritma yang C adalah kurang dan Algoritma yang E adalah lebih.

Hasil dari perhitungan tabel di atas:



Gambar III-3 Cabang Fisika 2 B

Gambar di atas merupakan hasil akhir dari proses pembentukan *tree*. Setelah selesai pembentukan *tree* adalah proses pembuatan aturan berdasarkan *tree* yang terbentuk. Dan aturan dari *tree* di atas adalah sebagai berikut:

1. Jika Fisika 2 A dan Algoritma A maka masa studinya adalah kurang
2. Jika Fisika 2 A dan Algoritma B maka masa studinya adalah lebih
3. Jika Fisika 2 A dan Algoritma C maka masa studinya adalah kurang
4. Jika Fisika 2 B dan Algoritma A maka masa studinya adalah kurang
5. Jika Fisika 2 B dan Algoritma B maka masa studinya adalah kurang
6. Jika Fisika 2 B dan Algoritma C maka masa studinya adalah kurang
7. Jika Fisika 2 B dan Algoritma E maka masa studinya adalah lebih
8. Jika Fisika 2 C maka masa studinya adalah lebih
9. Jika Fisika 2 D maka masa studinya adalah lebih
10. Jika Fisika 2 E maka masa studinya adalah lebih

3.6 Analisis Kebutuhan Non Fungsional

Analisis merupakan penguraian dari suatu masalah atau objek yang akhirnya menghasilkan suatu kesimpulan. Hal ini dimaksud untuk mengidentifikasi dan mengevaluasi masalah atau objek.

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 ke langkah 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 system 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*)

3.6.1 Analisis Kebutuhan Pengguna (User)

Aplikasi Klasifikasi masa studi mahasiswa di jurusan Teknik Komputer akan digunakan oleh dosen dan sekretariatan jurusan, dimana keterlibatannya dalam menggunakan aplikasi ini harus mengacu pada spesifikasi pengguna 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*

3.6.2 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 spesifikassi yang telah ada, secara keseluruhan kebutuhan perangkat keras untuk aplikasi ini telah terpenuhi.

3.6.3 Analisis Kebutuhan Perangkat Lunak (Software)

Perangkat lunak merupakan sarana pendukung lainnya bagi pembangunan aplikasi Klasifikasi masa studi mahasiswa di jurusan Teknik Komputer ini. 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.

3.7 Analisis Kebutuhan Fungsional

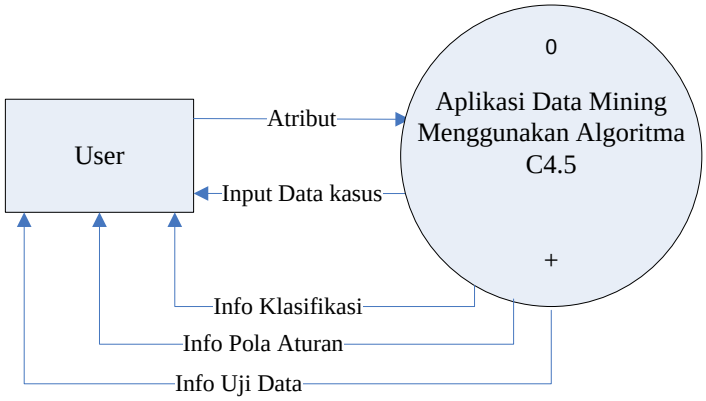
Setelah melakukan kebutuhan non fungsional maka dilanjutkan ke langkah berikutnya yaitu analisis kebutuhan fungsional. Dalam langkah ini dilakukan

penentuan entitas-entitas baik entitas internal maupun entitas eksternal, data yang mengalir, serta prosedur-prosedur yang bisa dilakukan oleh masing-masing entitas.

3.7.1 Diagram Konteks

Diagram konteks merupakan salah satu alat bantu dalam melakukan analisis terstruktur. Diagram konteks ini menggambarkan suatu system secara garis besarnya atau keseluruhannya saja. Dalam diagram konteks juga digambarkan entitas eksternal yang merupakan *brainware* yang menghasilkan data yang akan diolah oleh sistem maupun tujuan dari in*form*asi yang dihasilkan oleh sistem.

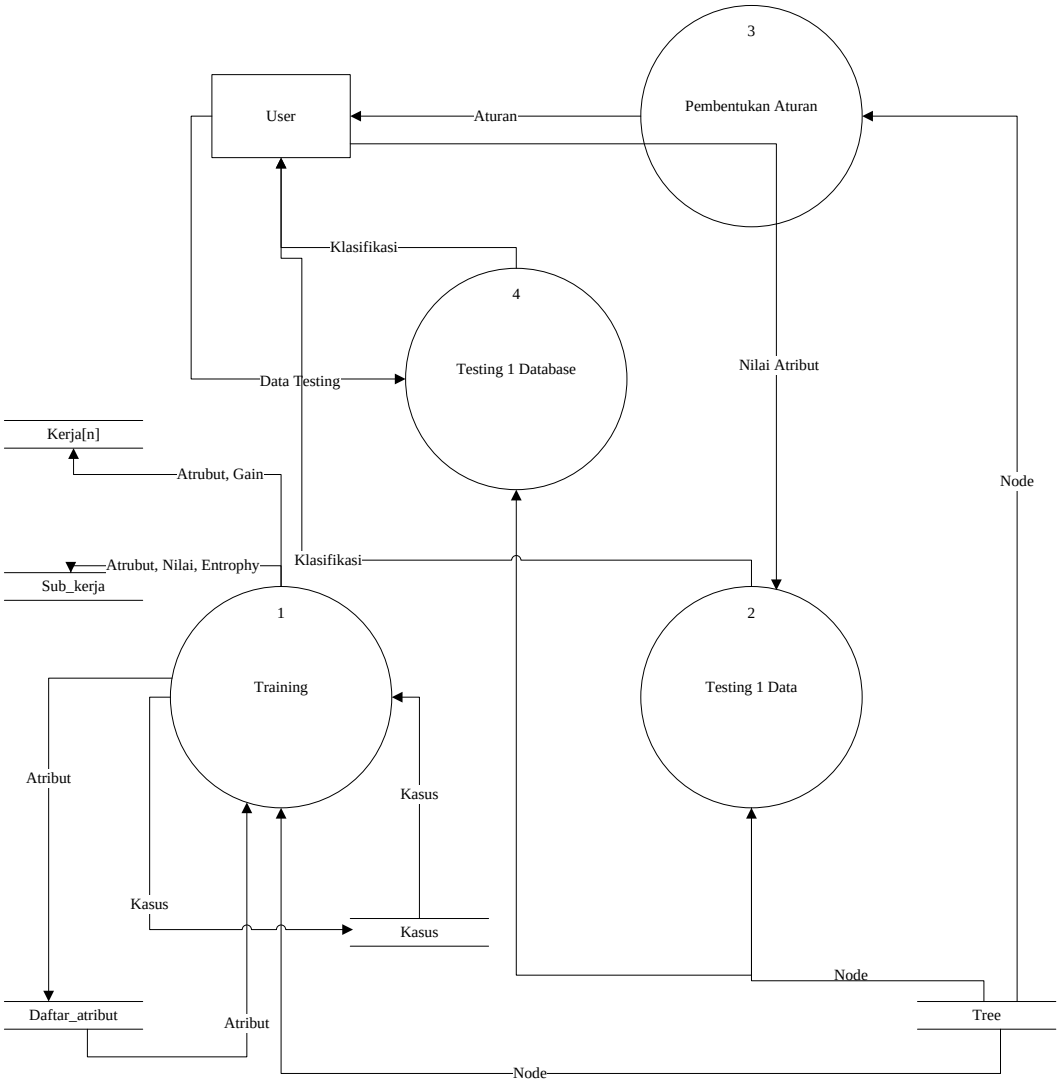
Gambar di bawah merupakan Diagram konteks pada Aplikasi Klasifikasi masa studi mahasiswa menggunakan algoritma *C4.5.*



Gambar III-4 Diagram Konteks Aplikasi *Data mining*

3.7.2 Diagram Alir Data Level 1

Diagram alir data merupan sebuah representasi dari suatu sistem yang menggambarkan bagian-bagian dari sistem tersbut beserta keterkaitan antara bagian-bagian yang ada. Dari diagraam alir data ini seseorang bisa mengetahui sumber dari in*form*asi di dalam sistem maupun tujuan dari masukan yang berasal dari entitas eksternal. Adapun diagram alir data level 1 dari aplikkasi *Data mining* ditunjukan oleh gambar



Gambar III-5 DFD Level 1 Aplikasi Data mining

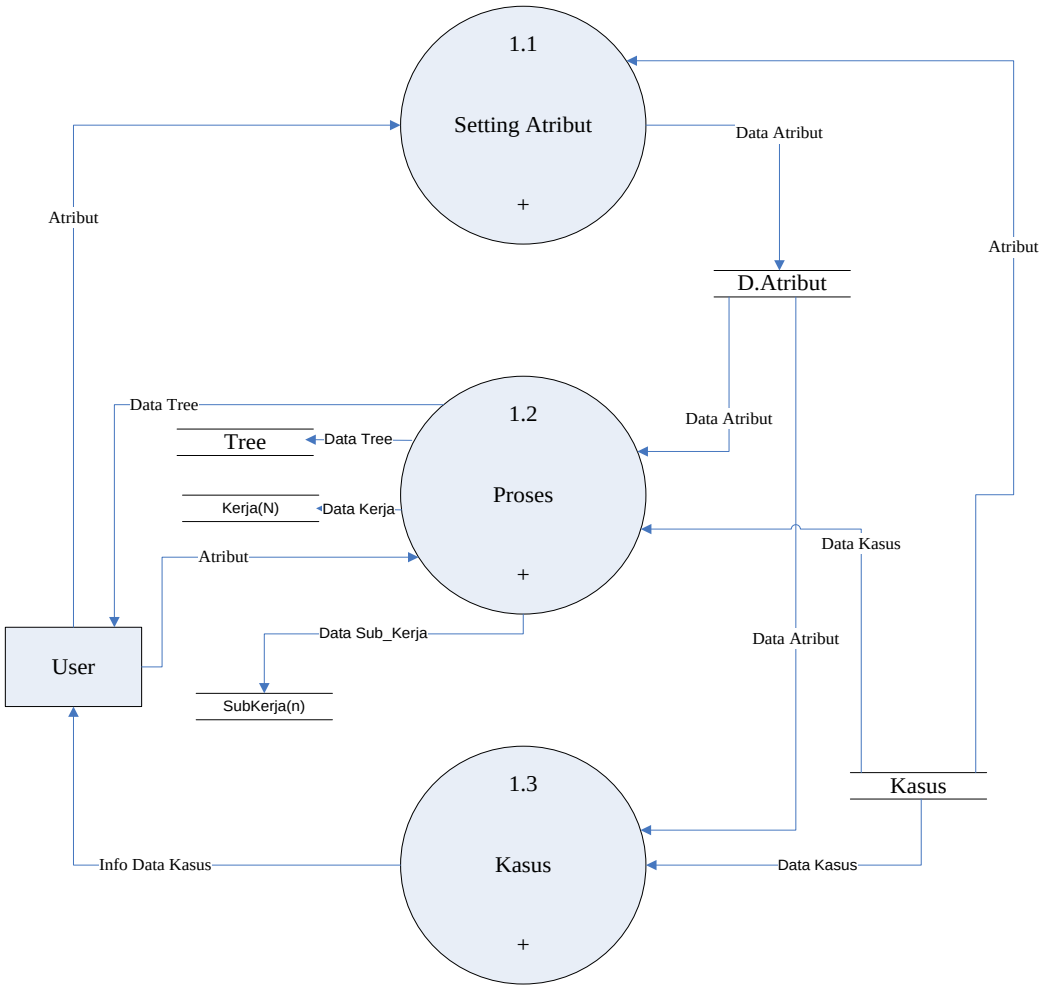
Table III-6 Spesifikasi Proses DFD Level 1 Aplikasi Data mining

No	Proses	Keterangan
1	No.Proses	1
	Nama Proses	*Training*
	Sumber	Kasus
	Input	Atribut, Data Kasus
	Output	Data *Tree*, Data Kasus
	Deskripsi	Proses utama dari algoritma *C4.5*
	Logika Proses	Setting Atribut. Keluar data kasus sesuai atribut yang dipiih. Tampil *Tree* hasil perhitungan berdasarkan atribut yang dipilih dan data pada tabel KASUS.
2	No.Proses	2

	Nama Proses	Aturan
	Sumber	*Tree*
	Input	Data *Tree*
	Output	Data Aturan
	Logika Proses	`Klik proses` `Keluar Aturan`
3	No.Proses	3
	Nama Proses	*Testing* 1 Data
	Sumber	*Tree*
	Input	Atribut
	Output	Prediksi Masa Studi
	Logika Proses	`Input Nilai Atribut` `Keluar Hasil Klasifikasi`
4	No.Proses	4
	Nama Proses	*Testing* 1 Database
	Sumber	*Tree*
	Input	Database
	Output	Persentase Kecocokan Prediksi program dengan data asli

	Logika Proses	`Open File DataBase`
		`Pilih Table`
		`Klik Proses`
		`Keluar Hasil Klasifikasi`

3.7.3 Diagram Alir Data Level 2 Proses 1 Proses Training



Gambar III-6 DFD Level 2 Proses 1 Proses Training

Table III-7 Spesifikasi Proses DFD Level 2 Proses 1 Proses Training

No	Proses	Keterangan
1	No.Proses	1.1
	Nama Proses	*Setting* Atribut
	Sumber	D_Atribut
	Input	Atribut
	Output	Data Atribut
	Deskripsi	Melakukan pengaturan terhadap atribut yang akan digunakan sebagai atribut penentu dan atribut tujuan

	Logika Proses	`Klik atribut`
		`Ceklis`
		`simpan`

2	No.Proses	1.2
	Nama Proses	Proses
	Sumber	D_atribut, Kasus, *Tree*, Kerja(N), Sub_kerja(N)
	Input	Data Kasus, Data Atribut
	Output	Data *Tree*, Data Sub_kerja, data Kerja

	Deskripsi	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);

3	No.Proses	1.3
	Nama Proses	Kasus
	Sumber	Kasus, D_atribut

	Input	Data Atribut, Data Kasus
	Output	Data Kasus

	Deskripsi	procedure TFrmKasus.*Form*Show(Sender: TObject);
		  var i : integer;
		begin
		  DBGKasus.Columns.Clear;
		  DM.IBTable1.TableName := 'Kasus';
		  DM.IBTable1.Active := false;
		  DM.IBTable1.Active := true;
		  DM.DS.DataSet := DM.IBTable1;
		  DBGKasus.DataSource := DM.DS;
		//ambil jumlah atribut aktif
		  with DM.Q1 do
		  begin
		  Close;
		  SQL.Text := 'SELECT NAMA_ATRIBUT FROM D_ATRIBUT WHERE IS_AKTIF = ''Y''';
		  Open;
		  i := 0;
		  if IsEmpty then
		  begin
		  Close;
		  SQL.Text := 'DELETE FROM kasus';
		  ExecSQL;
		  end;
		  while not Eof do
		  begin
		  DBGKasus.Columns.Add;
		  DBGKasus.Columns[i].*Field*Name := *Field*s[0].AsString;
		  DBGKasus.Columns[i].Width := 120;
		  Next;
		  inc(i);
		  end;
		  end;
		end;

3.8 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang dimaksudkan.

3.9 Perancangan Data

Perancangan data atau lebih dikenal dengan perancangan basis data yaitu menciptakan atau merancangan data yang terhubung dan disimpan secara bersama-sama. Untuk menggambarkannya digunakanlah struktur file.

3.9.1 Struktur Tabel

Pada proses operasi yang dilakukan aplikasi *Data mining* ini, dibutuhkan suatu database untuk penyimpanan data hasil proses. Adapun struktur tabel digunakan untuk menggambarkan tabel-tabel beserta *field-field* yang digunakan secara lebih detail, dimana tipe data dan ukuran suatu *field*.

Table III-8 KASUS

Nama *Field*	Tipe	Panjang
Algoritma_1	Varchar	1
Sitem_digital	Varchar	1
Organisasi_komputer	Varchar	1
Bahasa_rakitan	Varchar	1
masa_studi	Varchar	20

Table III-9 SUB_KERJA

Nama *Field*	Tipe	Panjang
Nama_atribut	Varchar	30
Nilai	Varchar	255
info(x)	Float	15,2
info_a(x)	Float	15,2
split_info(x)	float	15,2
Float	Float	15,2
result1	Varchar	30
result2	Varchar	30
jml_kasus	Integer	

Table III-10 KERJA

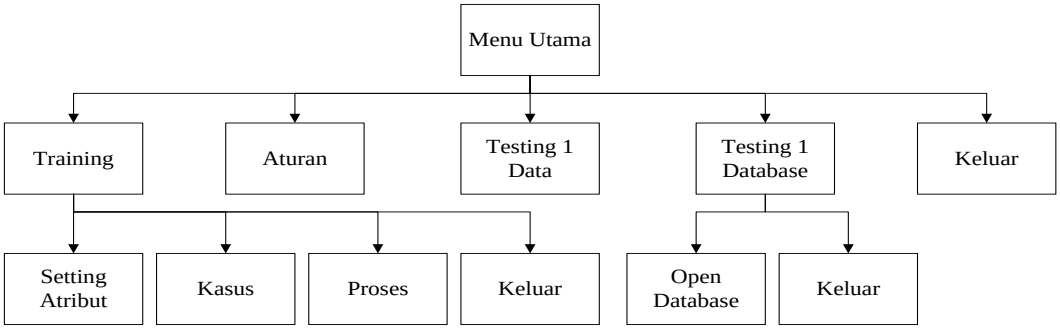
Nama *Field*	Tipe	Panjang
nama_atribut	Varchar	30
Gain	Float	15,2

Table III-11 TREE

Nama *Field*	Tipe	Panjang
id_node	Integer	30
Node	Varchar	30
Nilai	Varchar	30
Induk	Varchar	30
is_atribut	Char	1

3.10 Arsitektur menu

Berikut ini adalah perancangan struktur menu yang akan diterapkan pada program dan dirancang seefisien mungkin untuk mengurangi kompleksitas sistem.



Gambar III-7 Arsitektur Menu

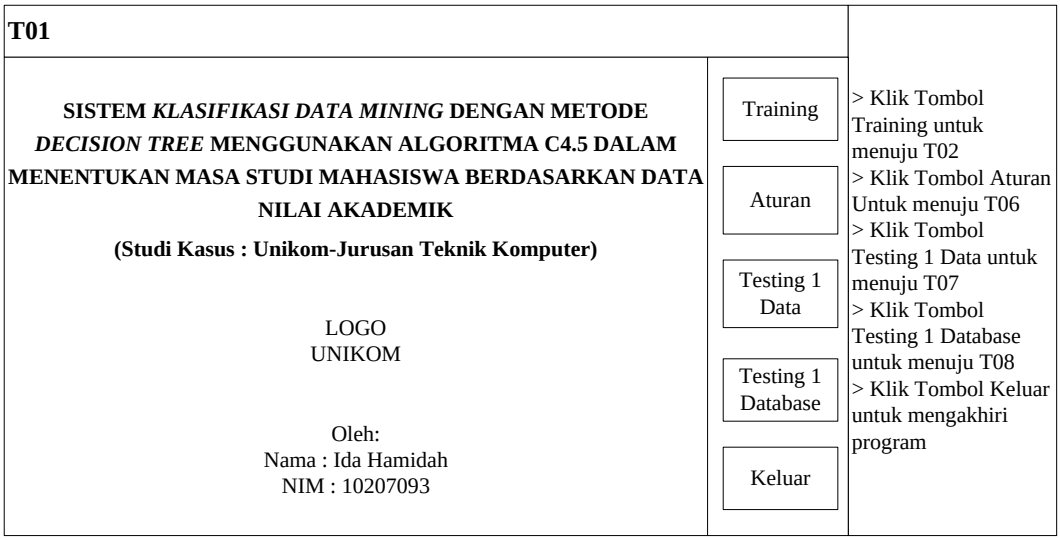
3.11 Perancangan Antarmuka

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

3.11.1 Perancangan Form

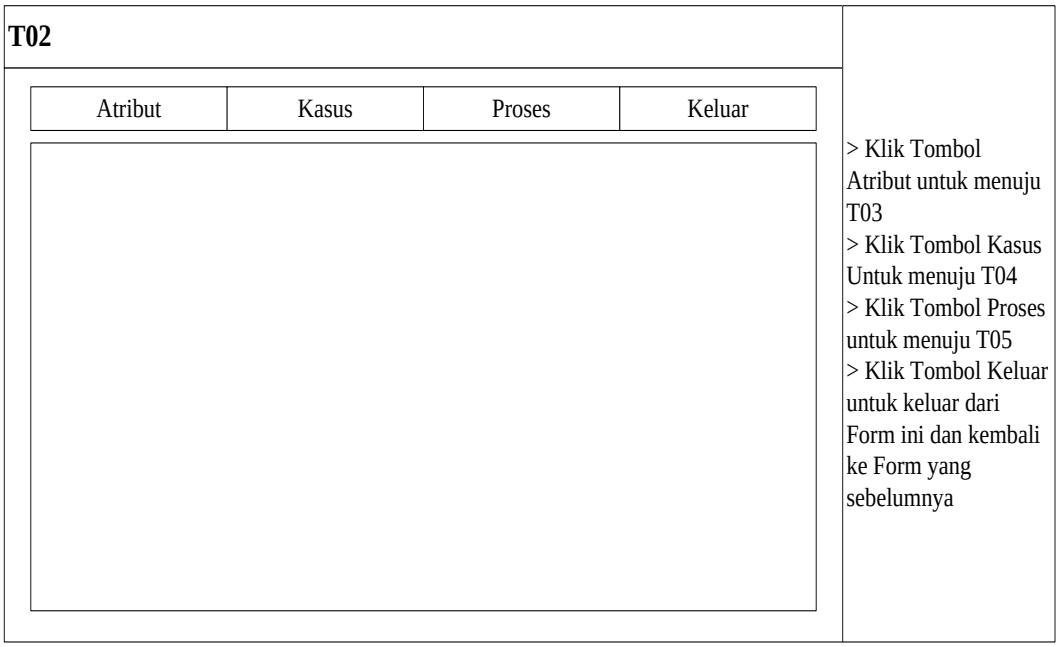
Dalam perancangan *form* di sini dilakukan perancangan baik terhadap *form* input dan *form* output. Adapun *form-form* yang terdapat dalam Aplikasi Klasifikasi untuk menentukan masa studi mahasiswa adalah sebagai berikut:

1. *Form* Utama



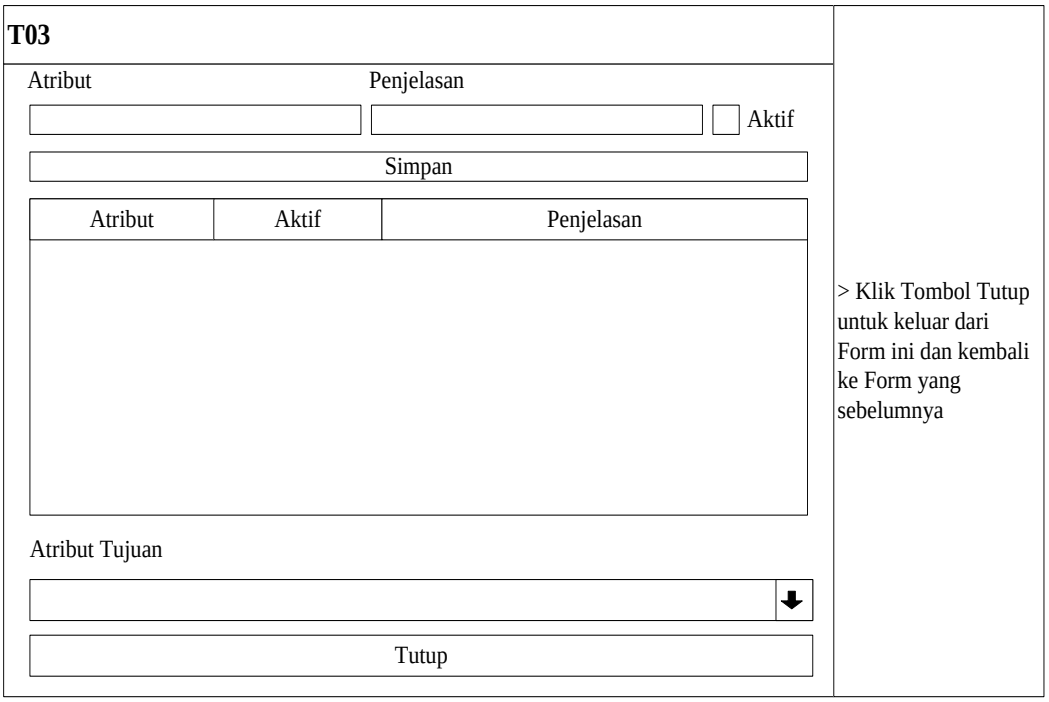
Gambar III-8 Gambar Form Utama pada Aplikasi Data mining

2. *Form Training*



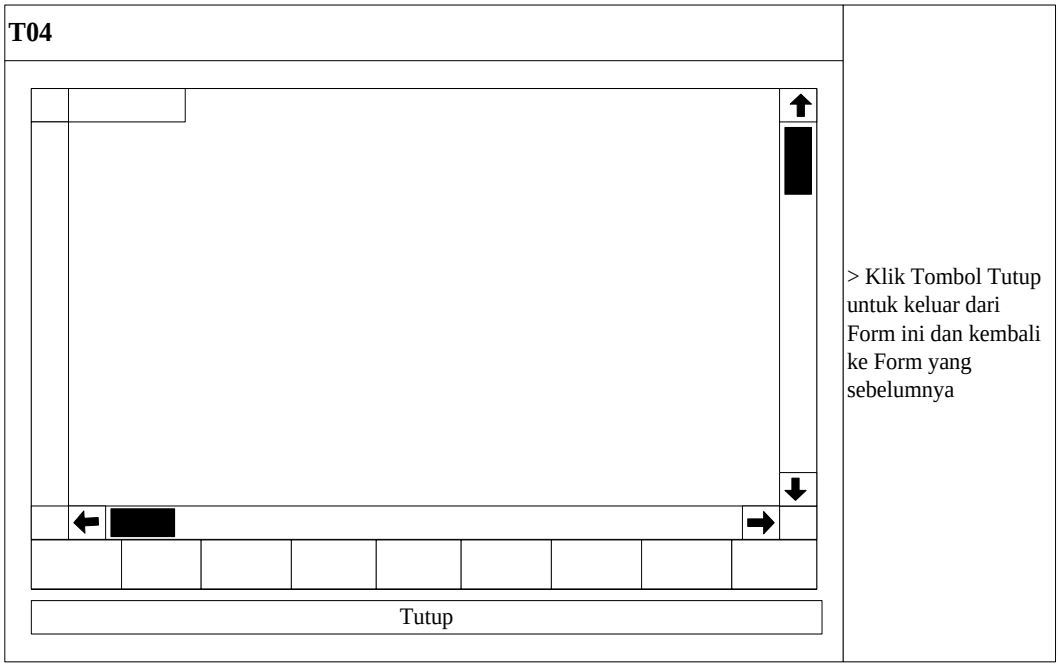
Gambar III-9 Form Training

3. *Form Setting* Atribut



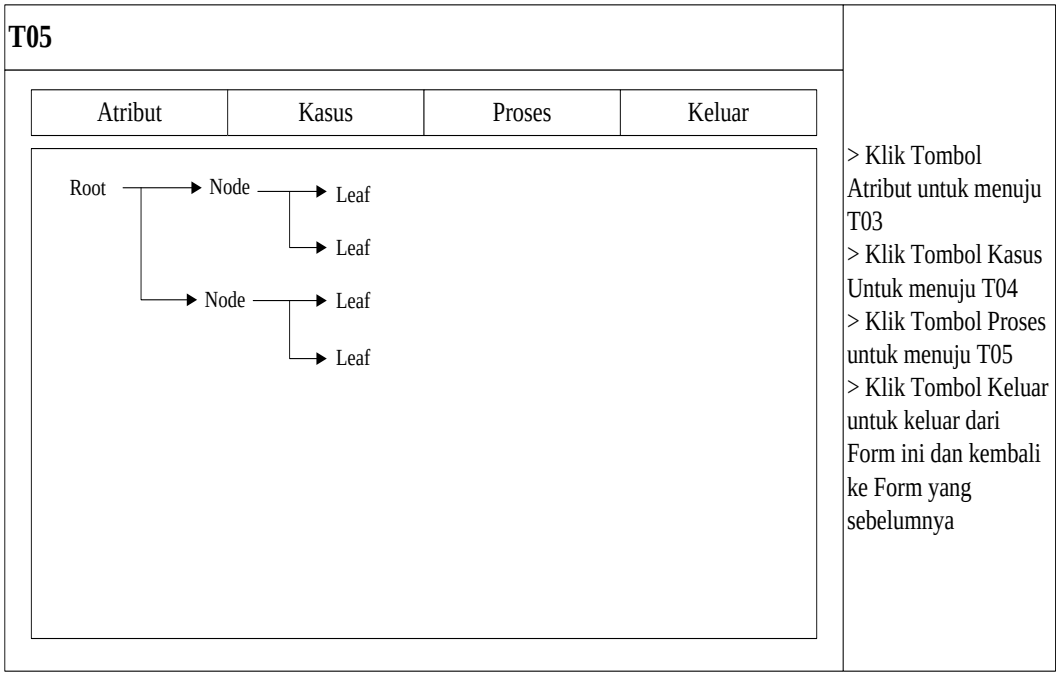
Gambar III-10 Form Setting Atribut

4. *Form* Kasus



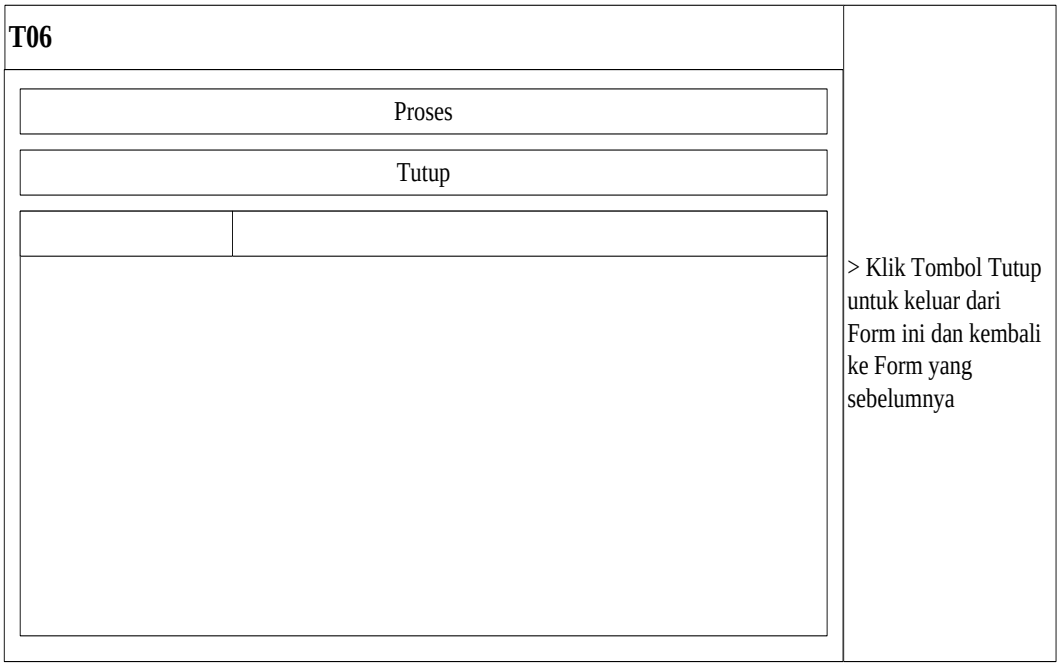
Gambar III-11 Form Kasus

5. *Form* Proses (*Tree*)



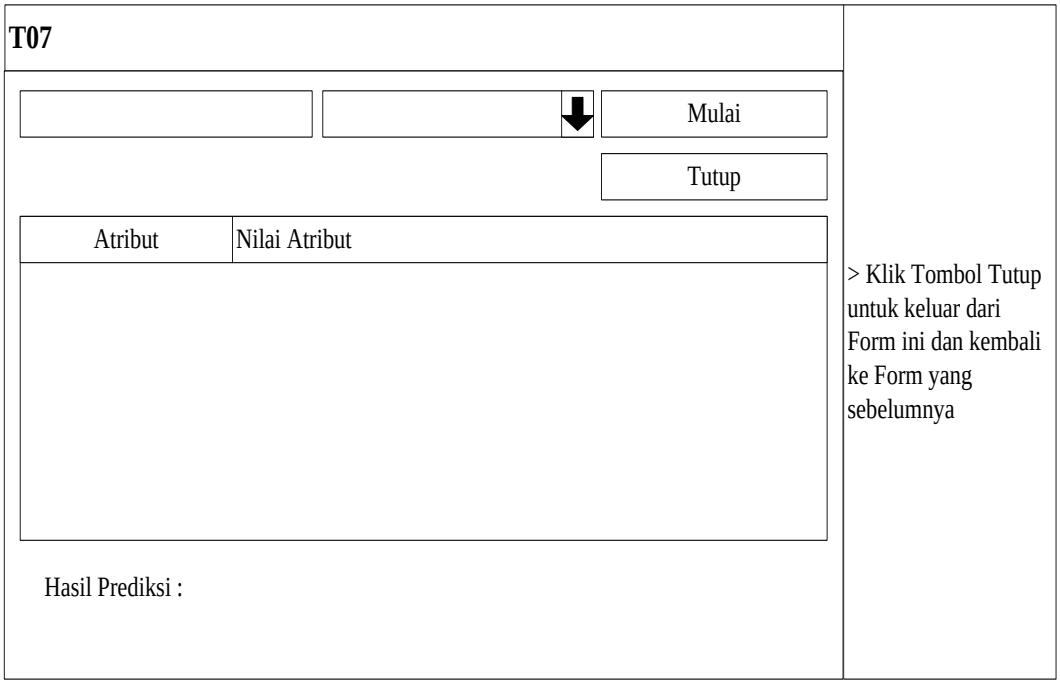
Gambar III-12 Form Proses (Tree)

6. *Form* Aturan



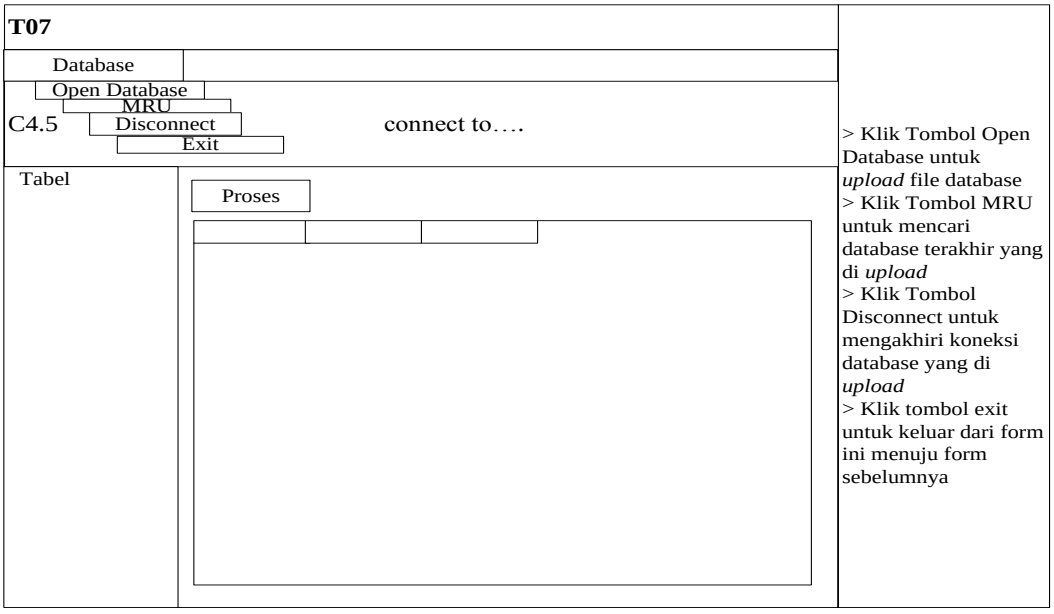
Gambar III-13 Form Aturan

7. *Form Testing* 1 Data



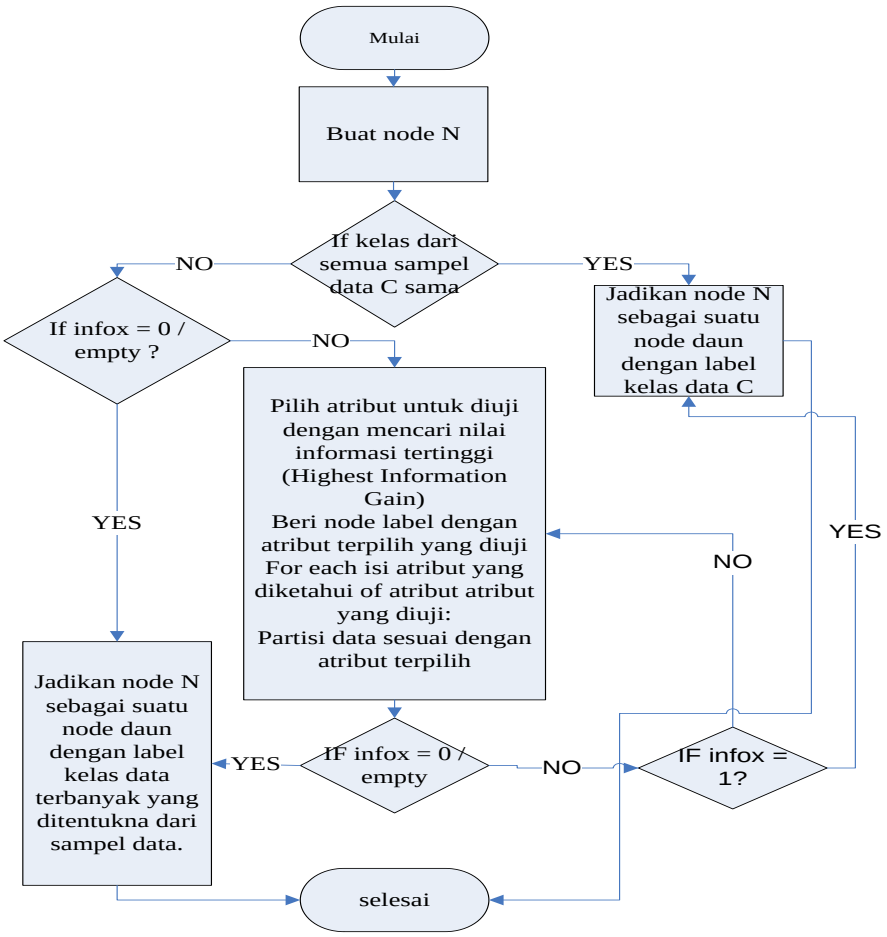
Gambar III-14 Form Testing 1 Data

8. *Form Testing* 1 Database



Gambar III-15 Form Testing 1 Database

3.11.2 Flowchart



Gambar III-16 Flowchart C4.5