BAB III

ANALISIS DAN PERANCANGAN

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari suatu sistem informasi yang utuh ke dalam bagian-bagian komponennya dengan maksud untuk mengidentifikasikan dan mengevaluasi permasalahan-permasalahan, kesempatan-kesempatan, hambatan-hambatan yang terjadi dan kebutuhan-kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan-perbaikannya.

3.1.1 Analisis Masalah

Pada tahun ajaran 2010/2011 jumlah calon mahasiswa baru UNIKOM yang dinyatakan lulus ujian saringan masuk adalah sebanyak 3813 calon mahasiswa. Namun, ternyata jumlah keseluruhan mahasiswa baru UNIKOM untuk tahun ajaran 2010/2011 adalah sebanyak 3341, itu berarti ada sekitar 472 calon mahasiswa mengundurkan diri dengan cara tidak melakukan registrasi. 12,5 % calon mahasiswa yang mungkin potensial, tidak mampu dipertahankan oleh UNIKOM Bandung.

Berdasarkan hal tersebut, maka jika kemungkinan pengunduran diri seorang calon mahasiswa baru dapat diketahui lebih dini mungkin pihak manajemen UNIKOM dapat melakukan tindakan-tindakan yang diperlukan untuk mempertahankan calon-calon mahasiswa tersebut. Namun, permasalahannya

hingga saat ini, UNIKOM belum memiliki standar baku yang dapat dijadikan sebagai alat untuk menganalisis kemungkinan pengunduran diri calon mahasiswa baru tersebut.

Kebijakan selama ini yang dilakukan oleh tim Manajemen Unikom untuk mengantisipasi pengunduran diri calon mahasiswa barunya adalah dengan cara menunggu calon mahasiswa melakukan registrasi hingga beberapa hari sebelum waktu registrasi ulang berakhir. Jika dalam waktu H-2 sebelum waktu registrasi berakhir calon mahasiswa tersebut belum melakukan registrasi, maka pihak manajemen akan mengirim surat pemberitahuan. Hal ini terasa kurang efektif karena harus menunggu dalam waktu yang lama untuk memastikan mahasiswa mana yang kemungkinan mengundurkan diri.

Untuk mengatasi permasalahan dalam melakukan pengambilan keputusan dalam proses penjaringan calon mahasiswa baru di Universitas Komputer Indonesia (UNIKOM), diperlukan suatu alat analisis bagi manajemen untuk mengetahui kemungkinan pengunduran diri calon mahasiswa baru. Analisis ini dapat dilakukan dengan memanfaatkan teori penalaran berbasis kasus, yaitu membandingkan kasus calon mahasiswa baru dengan kasus-kasus yang pernah terjadi di tahun-tahun sebelumnya. Alat analisis tersebut berupa sebuah aplikasi penunjang keputusan berbasis komputer yang mengimplementasikan algoritma klasifikasi *data mining.* Dengan ketersediaan data calon mahasiswa baru yang melimpah, *data mining* dapat menggali informasi yang terkubur dari kumpulan data calon mahasiswa baru tersebut. Aplikasi ini dapat digunakan untuk

menganalisis kemungkinan pengunduran diri calon mahasiswa baru berdasarkan data yang telah terkumpul sebelumnya.

3.1.2 Analisis Data

Data yang akan digunakan dalam membentuk pohon keputusan untuk menganalisis kemungkinan pengunduran diri calon mahasiswa baru Universitas Komputer Indonesia (UNIKOM) adalah data calon mahasiswa baru Tahun Ajaran 2010/2011 yang tersimpan pada tabel CalonMhsBaru. Data calon mahasiswa baru tersebut selanjutnya akan dilakukan pra-proses untuk menghasilkan data kasus yang siap untuk dibentuk menjadi sebuah pohon keputusan.

Data calon mahasiswa baru Universitas Komputer Indonesia tahun ajaran 2010/2011 dengan nama tabel MhsBaru memiliki format asli seperti tampak pada Tabel 3.1 berikut ini.

Tabel III.1 Format Asli Data CalonMhsBaru

NODAF
NORAF
NAMA
NILAI_UAN
JK
SEKOLAH
JUR_SEKOLAH
KODEPOS
TELEPON
ALAMAT
EMAIL
TGLDAFTAR
GELOMBANG
BEASISWA
KOMENTAR
CATATAN
NILAI_TEST
KET_LULUS
STATUS

AGAMA
NIM
JURUSAN
PROG_STUDI
DISPENSASI
THN_AKADEMIK
NO_IDENTITAS
NAMA_PROP
NAMA_KAB
REGISTRASI

Data yang tidak lengkap dan inkonsisten umumnya terjadi hampir pada setiap *database*, data yang tidak lengkap dapat disebabkan oleh berbagai macam sebab, seperti atribut dengan data yang salah atau dengan data yang kosong. Demikian pula dengan data calon mahasiswa baru Unikom, ada sebagian atribut yang tidak terlalu diperlukan sehingga proses *Data Preprocessing* perlu dilakukan sehingga *database* sesuai dengan ketentuan yang diperlukan oleh sistem.

Data preprocessing merupakan hal yang penting dalam proses *data mining*, hal-hal yang termasuk di dalamnya adalah:

a. Data Selection

Sebelum masuk ke proses *Data Preprocessing*, yang harus dilakukan lebih awal adalah pemilihan data (*data selection*). Data calon mahasiswa baru tersebut nantinya akan menjadi Data Kasus dalam proses operasional *data mining*. Dari data yang ada, kolom yang diambil sebagai atribut/variabel keputusan adalah kolom ***Registrasi***, sedangkan kolom yang diambil sebagai variabel penentu dalam pembentukan pohon keputusan adalah kolom:

1. NILAI_UAN
2. GELOMBANG
3. CATATAN

4. PROG_STUDI
5. NILAI_TEST
6. JURUSAN

Pemilihan variabel-variabel tersebut dengan pertimbangan bahwa jumlah nilai variabelnya tidak banyak sehingga diharapkan calon mahasiswa yang masuk dalam satu klasifikasi nilai variabel tersebut cukup banyak.

Berdasarkan variabel-variabel yang sudah terpilih, format data menjadi seperti tampak pada Tabel 3.2.

Tabel III.2 Format sampel data setelah pemilihan variabel

NILAI_UAN	GELOMBANG	CATATAN	NILAI_TEST	JURUSAN	PROG_STUDI	REGISTRASI
50,2	1	Test	C	MI	Diploma-3	Registrasi
51.6	1	Test	C	MI	Starata-1	Registrasi
43,7	2	Test	C	IF	Starata-1	Tidak Registrasi
48,4	1	Test	C	IF	Starata-1	Registrasi

b. Data Preprocessing/Data Cleaning

Data *cleaning* diterapkan untuk menambahkan isi atribut yang hilang atau kosong, dan merubah data yang tidak konsisten.

Dalam Tabel CaloMhsBaru yang ada di UNIKOM saat ini, terdapat beberapa atribut yang memiliki data kosong. Contoh atribut yang memiliki data kosong adalah atribut NILAI_TEST. Atribut NILAI_TEST berisi data nilai tes bagi calon mahasiswa yang mengikuti USM (Ujian Saringan Masuk) di UNIKOM. Sedangkan calon mahasiswa dengan jalur PMDK tidak memiliki nilai tes.

Tabel III.3 Contoh atribut yang tidak memiliki nilai

NILAI_UAN	GELOMBANG	CATATAN	NILAI_TEST	JURUSAN	PROG_STUDI	REGISTRASI
2	2	PMDK		Teknik Informatika	S1	Registrasi

Proses data *cleaning* yang dilakukan adalah dengan mengisi nilai dari atribut NILAI_TEST yang kosong dengan nilai T (Tidak ada nilai). Nilai T diberikan bagi calon mahasiswa baru yang tidak mengikuti ujian saringan masuk di UNIKOM yang melalui jalur PMDK.

Tabel III.4 Data atribut NILAI_TEST setelah di cleaning

NILAI_UAN	GELOMBANG	CATATAN	NILAI_TEST	JURUSAN	PROG_STUDI	REGISTRASI
2	2	PMDK	T	Teknik Informatika	S1	Registrasi

c. Data Transformation

Dalam proses ini, data ditransformasikan ke dalam bentuk yang sesuai untuk proses *data mining*.

1. Mengelompokkan nilai *nilai_uan*

Pengelompokkan nilai *nilai_uan* dilakukan dengan memasukkan *nilai_uan* calon mahasiswa dalam *range* seperti tampak pada Tabel 3.5 berikut ini.

Tabel III.5 Pengelompokkan nilai_uan

NILAI_UAN	Klasifikasi
30 – 45	1
45 – 60	2

2. Menerjemahkan nilai *jurusan*

   Penerjemahan nilai jurusan dilakukan denga mengganti nilai *jurusan* dengan nilai seperti Tampak pada Tabel 3.6.

Tabel III.6 Penerjemahan Jurusan

JURUSAN	Nilai Baru
MI	Manajemen Informatika
IF	Teknik Informatika

3. Menerjemahkan nilai PROG_STUDI

   Penerjemahan nilai PROG_STUDI dilakukan dengan mengganti nilai PROG_STUDI dengan nilai seperti Tampak pada Tabel 3.7.

Tabel III.7 Penerjemahan Prog_Studi

PROG_STUDI	Nilai Baru
Diploma-3	D3
Strata-1	S1

d. Data Reduction

Reduksi data dilakukan dengan menghilangkan atribut-atribut yang tidak diperlukan sehingga ukuran dari *database* menjadi kecil dan hanya menyertakan atribut-atribut yang diperlukan dalam proses *data mining*. Hal ini dikarenakan proses *data mining* akan lebih efisien terhadap data yang lebih kecil.

Format tabel yang diambil dari tabel CalonMhsBaru setelah dilakukan *Data Reduction* adalah sebagai berikut:

NILAI_UAN	GELOMBANG	CATATAN	NILAI_TEST	JURUSAN	PROG_STUDI	REGISTRASI

3.1.3 Analisis Algoritma C4.5

Dari proses *Data Preprocessing* yang telah dilakukan maka dihasilkan tabel yang siap untuk proses klasifikasi, seperti pada Tabel 3.8 dibawah ini.

Tabel III.8 Tabel Kasus

NILAI_UAN	GELOMBANG	CATATAN	NILAI_TEST	JURUSAN	PROG_STUDI	REGISTRASI
2	1	Test	C	Manajemen Informatika	D3	Registrasi
2	1	Test	C	Manajemen Informatika	S1	Registrasi
2	2	Test	B	Manajemen Informatika	S1	Registrasi
2	2	Test	D	Teknik Informatika	S1	Tidak Registrasi
1	2	Test	C	Teknik Komputer	S1	Tidak Registrasi
1	1	Test	C	Teknik Informatika	S1	Registrasi
1	1	Test	C	Teknik Komputer	D3	Registrasi
1	1	Test	C	Manajemen Informatika	S1	Tidak Registrasi
1	1	Test	C	Teknik Informatika	S1	Registrasi
2	1	PMDK	T	Teknik Komputer	S1	Registrasi
2	2	PMDK	T	Teknik Informatika	S1	Registrasi
2	2	PMDK	T	Teknik Informatika	S1	Tidak Registrasi
1	1	Test	B	Manajemen Informatika	S1	Registrasi
1	2	Test	C	Teknik Komputer	D3	Tidak Registrasi
2	2	Test	B	Teknik Komputer	S1	Registrasi
2	1	PMDK	T	Manajemen Informatika	S1	Registrasi
1	1	Test	B	Manajemen Informatika	D3	Tidak Registrasi
2	2	PMDK	T	Teknik Komputer	S1	Registrasi
1	1	Test	D	Teknik Informatika	S1	Tidak Registrasi
2	2	PMDK	T	Teknik Komputer	S1	Registrasi
2	2	Test	B	Teknik Komputer	S1	Tidak Registrasi
1	1	Test	C	Teknik Informatika	S1	Registrasi
2	1	Test	B	Manajemen Informatika	D3	Tidak Registrasi
1	1	PMDK	T	Teknik Informatika	S1	Registrasi
1	2	Test	C	Teknik Komputer	S1	Registrasi
2	1	Test	D	Teknik Informatika	S1	Registrasi
2	2	Test	B	Teknik Komputer	S1	Tidak Registrasi

2	1	Test	B	Teknik Informatika	S1	Tidak Registrasi
2	2	Test	C	Teknik Informatika	S1	Tidak Registrasi
1	1	Test	B	Teknik Komputer	S1	Registrasi
1	1	Test	C	Teknik Informatika	S1	Registrasi
1	1	Test	C	Teknik Informatika	S1	Registrasi
1	1	Test	D	Teknik Informatika	S1	Tidak Registrasi
1	2	Test	C	Manajemen Informatika	S1	Registrasi
2	2	Test	D	Teknik Informatika	S1	Tidak Registrasi

Tabel KASUS adalah kumpulan data yang akan diproses untuk membentuk sebuah pohon keputusan. Data diatas merupakan 35 data sampel yang diambil secara acak dari tabel MhsBaru. Atribut tujuannya yaitu **REGISTRASI**, dimana memiliki dua nilai atribut yang berbeda yaitu **Registrasi dan Tidak Registrasi.** Dari Data KASUS diatas akan dibuat pohon keputusan untuk menganalisis kemungkinan pengunduran diri calon mahasiswa baru berdasarkan nilai atribut tujuannya dengan melihat nilai atribut lainnya yaitu:

1. NILAI_UAN
2. GELOMBANG
3. CATATAN
4. NILAI_TEST
5. JURUSAN
6. PROG_STUDI

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut:

1. Pilih atribut sebagai akar.
2. Buat cabang untuk tiap-tiap nilai.

3. Bagi kasus dalam cabang.
4. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai *Gain* tertinggi dari atribut-atribut yang ada. Untuk menghitung *Gain* digunakan rumus seperti tertera dalam persamaan 1 berikut.

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^{n} \frac{|Si|}{|S|} * Entropy(Si) \quad (1)$$

Keterangan:

S : himpunan kasus

A : atribut

n : jumlah partisi atribut A

|Si| : jumlah kasus pada partisi ke-i

|S| : jumlah kasus dalam S

Sementara itu, penghitungan nilai entropi dapat dilihat pada persamaan berikut.

$$Entropy(S) = \sum_{i=1}^{n} - pi * log_2 pi \quad (2)$$

Keterangan

S : himpunan kasus

A : fitur

n : jumlah partisi S

pi : proporsi dari Si terhadap S

Berikut ini adalah penjelasan lebih terperinci mengenai tiap-tiap langkah dalam pembentukan pohon keputusan tabel KASUS diatas dengan menggunakan algoritma C4.5.

a. Menghitung jumlah kasus, jumlah kasus untuk keputusan Registrasi, jumlah kasus untuk keputusan Tidak Registrasi, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut ***NILAI_UAN, GELOMBANG, CATATAN, NILAI_TEST, JURUSAN*** dan ***PROG_STUDI.*** Proses perhitungan untuk mendapatkan nilai Entropy dan *Gain* setiap atribut adalah sebagai berikut.

Entropy(Total) = $(-\frac{14}{35} * \log_2(\frac{14}{35})) + (-\frac{21}{35} * \log_2(\frac{21}{35})) = 0.9710$

a. Atribut NILAI_UAN

   Entropy(1)= $(-\frac{9}{17} * \log_2(\frac{9}{17})) + (-\frac{8}{17} * \log_2(\frac{8}{17})) = 0.8655$

   Entropy(2)= $(-\frac{8}{18} * \log_2(\frac{8}{18})) + (-\frac{10}{18} * \log_2(\frac{10}{18})) = 0.9910$

b. Atribut GELOMBANG

   Entropy(1)= $(-\frac{14}{20} * \log_2(\frac{14}{20})) + (-\frac{6}{20} * \log_2(\frac{6}{20})) = 0.8819$

   Entropy(2)= $(-\frac{8}{15} * \log_2(\frac{8}{15})) + (-\frac{7}{15} * \log_2(\frac{7}{15})) = 0.9965$

c. Atribut CATATAN

   Entropy(Test)= $(-\frac{13}{28} * \log_2(\frac{13}{28})) + (-\frac{15}{28} * \log_2(\frac{15}{28})) = 0.9964$

   Entropy(PMDK)= $(-\frac{1}{7} * \log_2(\frac{1}{7})) + (-\frac{6}{7} * \log_2(\frac{6}{7})) = 0.5914$

d. Atribut NILAI_TEST

   Entropy(B)= $(-\frac{5}{9} * \log_2(\frac{5}{9})) + (-\frac{4}{9} * \log_2(\frac{4}{9})) = 0.9915$

Entrop(C)= $(-\frac{4}{14} * \log_2(\frac{4}{14})) + (-\frac{10}{14} * \log_2(\frac{10}{14})) = 0.8631$

Entropy(D)= $(-\frac{4}{5} * \log_2(\frac{4}{5})) + (-\frac{1}{5} * \log_2(\frac{1}{5})) = 0.7219$

Entropy(T)= $(-\frac{1}{7} * \log_2(\frac{1}{7})) + (-\frac{6}{7} * \log_2(\frac{6}{7})) = 0.5914$

e. Atribut JURUSAN

Entropy(TEKNIK INFORMATIKA)= $(-\frac{7}{15} * \log_2(\frac{7}{15})) + (-\frac{8}{15} * \log_2(\frac{8}{15})) =$ 0.9965

Entropy(MANAJEMEN INFORMATIKA)= $(-\frac{3}{9} * \log_2(\frac{3}{9})) + (-\frac{6}{9} * \log_2(\frac{6}{9})) =$ 0.9182

Entropy(TEKNIK KOMPUTER)= $(-\frac{4}{11} * \log_2(\frac{4}{11})) + (-\frac{7}{11} * \log_2(\frac{7}{11})) =$ 0.9450

f. Atribut PROG_STUDI

Entropy(S1)= $(-\frac{11}{30} * \log_2(\frac{11}{30})) + (-\frac{19}{30} * \log_2(\frac{19}{30})) = 0.9480$

Entropy(D3)= $(-\frac{3}{5} * \log_2(\frac{3}{5})) + (-\frac{2}{5} * \log_2(\frac{2}{5})) = 0.9789$

Menghitung nilai *Gain* dengan menggunakan persamaan 1.

1. Atribut NILAI_UAN

Gain(Total, NILAI_UAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{NILAI_UAN}|}{|Total|} * Entropy(\text{NILAI_UAN } i)$

Gain(Total, NILAI_UAN)

= 0.9710 – $((\frac{17}{35} * 0.8655) + (\frac{18}{35} * 0.9910))$ = 0.9710 – (0.4204 + 0.5096) = 0.041

2. Atribut GELOMBANG

Gain(Total, GELOMBANG)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{GELOMBANG}|}{|Total|} * Entropy(\text{GELOMBANG } i)$

= 0.9710 – (($\frac{20}{35} * 0.8819$) + ($\frac{15}{35}$ * 0.9965)) = 0.9710 – (0.5039 + 0.4271) = 0.04

3. Atribut CATATAN

Gain(Total, CATATAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{CATATAN}|}{|Total|} * Entropy(\text{CATATAN } i)$

= 0.9710 – (($\frac{28}{35} * 0.9964$) + ($\frac{7}{35}$ * 0.5914)) = 0.9710 – (0.7971 + 0.1183) = 0.056

4. Atribut NILAI_TEST

Gain(Total, NILAI_TEST)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{NILAI_TEST}|}{|Total|} * Entropy(\text{NILAI_TEST } i)$

= 0.9710 – (($\frac{9}{35} * 0.9915$) + ($\frac{14}{35}$ * 0.8631) + ($\frac{5}{35} * 0.7219$) + ($\frac{7}{35} * 0.5914$)) = 0.9710 – (0.2549 + 0.3452 + 0.1031 + 0.1183) = 0.1495

5. Atribut JURUSAN

Gain(Total, JURUSAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{JURUSAN}|}{|Total|} * Entropy(\text{JURUSAN } i)$

= 0.9710 – (($\frac{15}{35} * 0.9965$) + ($\frac{9}{35}$ * 0.9182) + ($\frac{11}{35} * 0.9450$)) = 0.9710 – (0.4271 + 0.2361 + 0.2970) = 0.0108

6. Atribut PROG_STUDI

Gain(Total, PROG_STUDI)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{PROG_STUDI}|}{|Total|} * Entropy(\text{PROG_STUDI } i)$

$= 0.9710 - ((\frac{30}{35} * 0.9480) + (\frac{5}{35} * 0.9789)) = 0.9710 - (0.7703 + 0.1398) = 0.061$

Hasil Perhitungan nilai Entropy dan nilai *Gain* setiap Atribut ditunjukkan oleh Tabel 3.9 berikut ini.

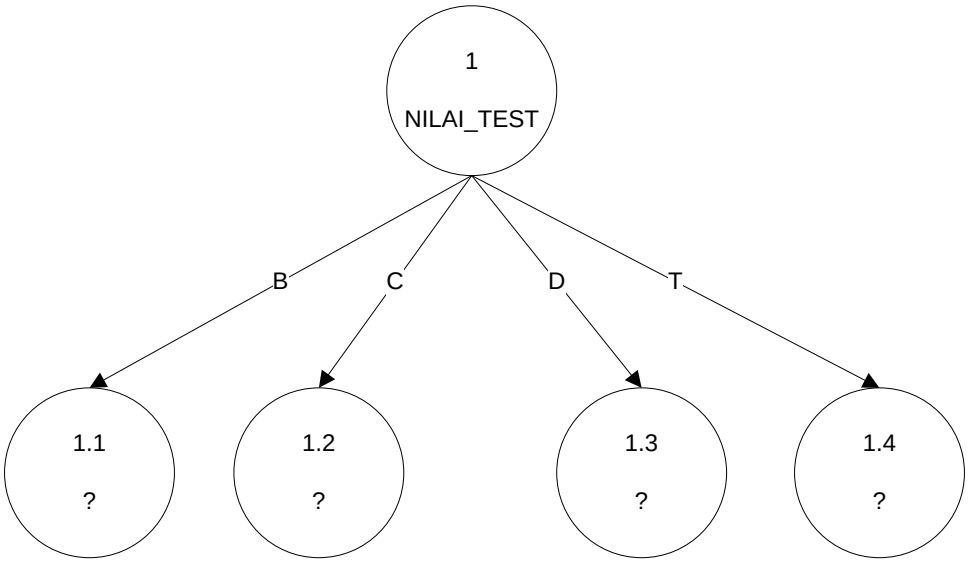
Tabel III.9 Perhitungan Node 1

Node			Jml kasus (S)	Tidak Registrasi (Si)	Registrasi (Si)	Entropy	Gain
1	TOTAL		35	14	21	0.9710	
	NILAI_UAN						0.041
		1	17	9	8	0.8655	
		2	18	8	10	0.9910	
	GELOMBANG						0.041
		1	20	14	6	0. 8819	
		2	15	8	7	0.9965	
	CATATAN						0.056
		TEST	28	13	15	0.9964	
		PMDK	7	1	6	0.5914	
	NILAI_TEST						0.1495
		B	9	5	4	0.9915	
		C	14	4	10	0.8631	
		D	5	4	1	0.7219	
		T	7	1	6	0.5914	
	JURUSAN						0.0108
		TEKNIK INFORMATIKA	15	7	8	0.9965	
		MANAJEMEN INFORMATIKA	9	3	6	0.9182	
		TEKNIK KOMPUTER	11	4	7	0.9450	
	PROG_STUDI						0.061
		S1	30	11	19	0.9480	
		D3	5	3	2	0.9789	

Dari hasil pada Tabel 3.9 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah ***NILAI_TEST***, yaitu sebesar 0.1495. Dengan demikian ***NILAI_TEST*** dapat menjadi *node* akar. Ada empat nilai atribut dari

NILAI_TEST, yaitu ***B, C, D,*** dan ***T***. Dari keempat nilai atribut tersebut masih perlu dilakukan perhitungan lagi.

Dari hasil perhitungan node 1 dapat digambarkan pohon keputusan sementaranya tampak seperti pada Gambar 3.1 berikut ini.



Gambar III.1 Pohon Keputusan Hasil Perhitungan Node 1

b. Menghitung jumlah kasus, jumlah kasus untuk keputusan ***Registrasi***, jumlah kasus untuk keputusan ***Tidak Registrasi***, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut ***NILAI_UAN, GELOMBANG, CATATAN, JURUSAN*** dan ***PROG_STUDI***. Pada pohon keputusan node 1 diatas diasumsikan yang akan menjadi node akar adalah atribut ***NILAI_TEST*** dengan nilai atribut ***B.***

Entropy(NILAI_TEST-B) = $(-\frac{5}{9} * \log_2(\frac{5}{9})) + (-\frac{4}{9} * \log_2(\frac{4}{9})) = 0.9910$

1. Atribut NILAI_UAN

   Entropy(1)= $(-\frac{1}{3} * \log_2(\frac{1}{3})) + (-\frac{2}{3} * \log_2(\frac{2}{3})) = 0.9182$

Entropy(2)= $(-\frac{4}{6} * \log_2(\frac{4}{6})) + (-\frac{2}{6} * \log_2(\frac{2}{6})) = 0.9182$

2. Atribut GELOMBANG

Entropy(1)= $(-\frac{3}{5} * \log_2(\frac{3}{5})) + (-\frac{2}{5} * \log_2(\frac{2}{5})) = 0.9710$

Entropy(2)= $(-\frac{2}{4} * \log_2(\frac{2}{4})) + (-\frac{2}{4} * \log_2(\frac{2}{4})) = 0$

3. Atribut CATATAN

Entropy(Test)= $(-\frac{5}{9} * \log_2(\frac{5}{9})) + (-\frac{4}{9} * \log_2(\frac{4}{9})) = 0.9910$

Entropy(PMDK)= 0

4. Atribut JURUSAN

Entropy(TEKNIK INFORMATIKA)= 0

Entropy(MANAJEMEN INFORMATIKA)= $(-\frac{2}{4} * \log_2(\frac{2}{4})) + (-\frac{2}{4} * \log_2(\frac{2}{4}))$ = 1

Entropy(TEKNIK KOMPUTER)= $(-\frac{2}{4} * \log_2(\frac{2}{4})) + (-\frac{2}{4} * \log_2(\frac{2}{4})) = 1$

5. Atribut PROG_STUDI

Entropy(S1)= $(-\frac{3}{7} * \log_2(\frac{3}{7})) + (-\frac{4}{7} * \log_2(\frac{4}{7})) = 0.9853$

Entropy(D3)= 0

Perhitungan *Gain.*

1. Atribut NILAI_UAN

Gain(Total, NILAI_UAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{NILAI_UAN}|}{|Total|} * Entropy(\text{NILAI_UAN } i)$

= $0.9910 - ((\frac{3}{9} * 0.9182) + (\frac{6}{9} * 0.9182)) = 0.9910 - (0.3061 + 0.6121) =$ 0.0728

2. Atribut GELOMBANG

Gain(Total, GELOMBANG)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{GELOMBANG}|}{|Total|} * Entropy(\text{GELOMBANG } i)$

= 0.9910 – $((\frac{5}{9} * \ 0.9710) + (\frac{4}{9} * 1))$ = 0.9910 – (0.5394 + 0.4444) = 0.0072

3. Atribut CATATAN

Gain(Total, CATATAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{CATATAN}|}{|Total|} * Entropy(\text{CATATAN } i)$

= 0.9910 – $((\frac{9}{9} * \ 0.9910) + (\frac{0}{9} * 0))$ = 0.9910 – (0.9910 + 0) = 0

4. Atribut JURUSAN

Gain(Total, JURUSAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{JURUSAN}|}{|Total|} * Entropy(\text{JURUSAN } i)$

= 0.9910 – $((\frac{1}{9} * \ 0) + (\frac{4}{9} * 1) + (\frac{4}{9} * \ 1))$ = 0.9910 – (0 + 0.4444 + 0.4444) = 0.1022

5. Atribut PROG_STUDI

Gain(Total, PROG_STUDI)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{PROG_STUDI}|}{|Total|} * Entropy(\text{PROG_STUDI } i)$

= 0.9910 – $((\frac{7}{9} * \ 0.9853) + (\frac{2}{9} * 0))$ = 0.9910 – (0.7663 + 0) = 0.2247

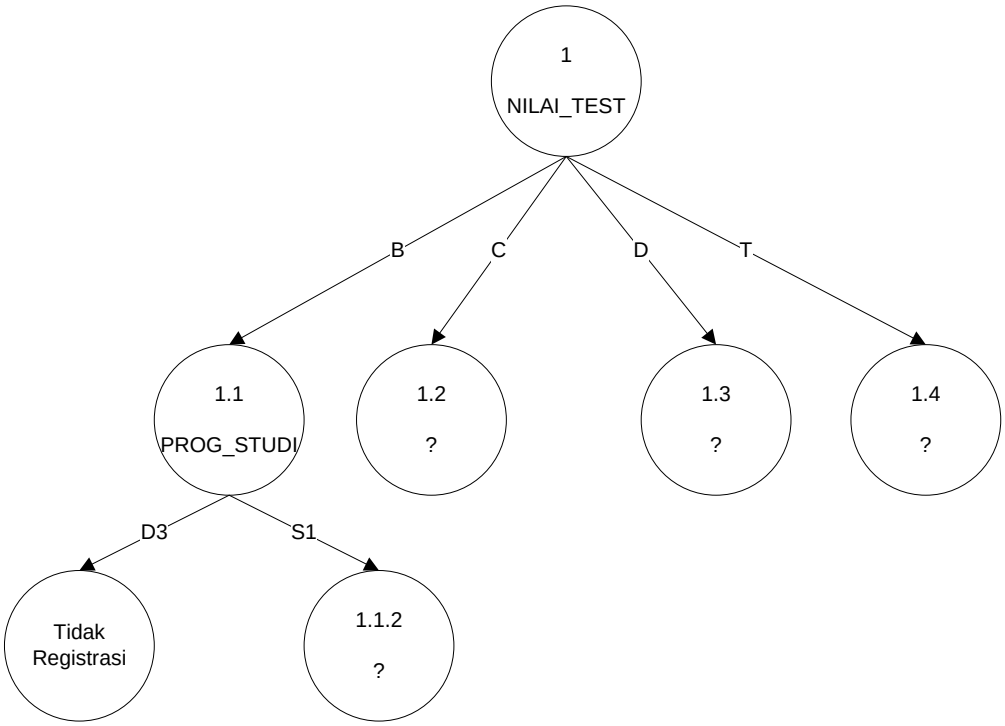
Hasil Perhitungan nilai Entropy dan nilai *Gain* setiap Atribut ditunjukkan oleh Tabel 3.10 berikut ini.

Tabel III.10 Perhitungan Node 1.1

Node			Jml kasus (S)	Tidak Registrasi (Si)	Registrasi (Si)	Entropy	Gain
1.1	NILAI_TEST-B		9	5	4	0.9910	
	NILAI_UAN						0.0728
		1	3	1	2	0.9182	
		2	6	4	2	0.9182	
	GELOMBANG						0.0072
		1	5	3	2	0.9710	
		2	4	2	2	1	
	CATATAN						0
		TEST	9	5	4	0.9910	
		PMDK	0	0	0	0	
	JURUSAN						0.1022
		TEKNIK INFORMATIKA	1	1	0	0	
		MANAJEMEN INFORMATIKA	4	2	2	1	
		TEKNIK KOMPUTER	4	2	2	1	
	PROG_STUDI						0.2247
		S1	7	3	4	0.9853	
		D3	2	2	0	0	

Dari hasil pada Tabel 3.10 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah ***PROG_STUDI***, yaitu sebesar 0.2247. Dengan demikian ***PROG_STUDI*** dapat menjadi *node* akar. Ada dua nilai atribut dari ***PROG_STUDI***, yaitu ***S1*** dan ***D3***. Dari kedua nilai atribut tersebut masih perlu dilakukan perhitungan lagi.

Dari hasil perhitungan node 1.1 dapat digambarkan pohon keputusan sementaranya tampak seperti pada Gambar 3.2 berikut ini.



Gambar III.2 Pohon Keputusan Hasil Perhitungan Node 1.1

c. Menghitung jumlah kasus, jumlah kasus untuk keputusan ***Registrasi***, jumlah kasus untuk keputusan ***Tidak Registrasi***, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut ***NILAI_UAN, GELOMBANG, CATATAN,*** dan ***JURUSAN***.

Entropy(NILAI_TEST-B dan PROG_STUDI-S1) = $(-\frac{3}{7} * \log_2(\frac{3}{7}))$ + $(-\frac{4}{7} * \log_2(\frac{4}{7}))$ = 0.9853

1. Atribut NILAI_UAN

   Entropy(1)= $(-\frac{0}{2} * \log_2(\frac{0}{2}))$ + $(-\frac{2}{2} * \log_2(\frac{2}{2}))$ = 0

   Entropy(2)= $(-\frac{3}{5} * \log_2(\frac{3}{5}))$ + $(-\frac{2}{5} * \log_2(\frac{2}{5}))$ = 0.9710

2. Atribut GELOMBANG

   Entropy(1)= $(-\frac{1}{3} * \log_2(\frac{1}{3}))$ + $(-\frac{2}{3} * \log_2(\frac{2}{3}))$ = 0.9182

Entropy(2)= $(-\frac{2}{4} * \log_2(\frac{2}{4}))$ + $(-\frac{2}{4} * \log_2(\frac{2}{4}))$ = 0

3. Atribut CATATAN

Entropy(Test)= $(-\frac{3}{7} * \log_2(\frac{3}{7}))$ + $(-\frac{4}{7} * \log_2(\frac{4}{7}))$ = 0.9853

Entropy(PMDK)= 0

4. Atribut JURUSAN

Entropy(TEKNIK INFORMATIKA)= 0

Entropy(MANAJEMEN INFORMATIKA)= 0

Entropy(TEKNIK KOMPUTER)= $(-\frac{2}{4} * \log_2(\frac{2}{4}))$ + $(-\frac{2}{4} * \log_2(\frac{2}{4}))$ = 1

Perhitungan *Gain*

1. Atribut NILAI_UAN

Gain(Total, NILAI_UAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{NILAI_UAN}|}{|Total|} * Entropy(\text{NILAI_UAN } i)$

= 0.9853 – $((\frac{2}{7} * 0) + (\frac{5}{7} * 0.9710))$ = 0.9853 – (0 + 0.6936) = 0.2917

2. Atribut GELOMBANG

Gain(Total, GELOMBANG)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{GELOMBANG}|}{|Total|} * Entropy(\text{GELOMBANG } i)$

= 0.9853 – $((\frac{3}{7} * 0.9182) + (\frac{4}{7} * 1))$ = 0.9853 – (0.3935 + 0.5714) = 0.0204

3. Atribut CATATAN

Gain(Total, CATATAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{CATATAN}|}{|Total|} * Entropy(\text{CATATAN } i)$

= 0.9853 – $((\frac{7}{7} * 0.9853) + (\frac{0}{7} * 0))$ = 0.9853 – (0.9853 + 0) = 0

4. Atribut JURUSAN

Gain(Total, JURUSAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{JURUSAN}|}{|Total|} * Entropy(\text{JURUSAN } i)$

= 0.9853 – $((\frac{1}{7} * 0) + (\frac{2}{7} * 0) + (\frac{4}{7} * 1))$ = 0.9853 – (0 + 0 + 0.5714) = 0.4139

Hasil Perhitungan nilai Entropy dan nilai *Gain* setiap Atribut ditunjukkan oleh Tabel 3.11 berikut ini.

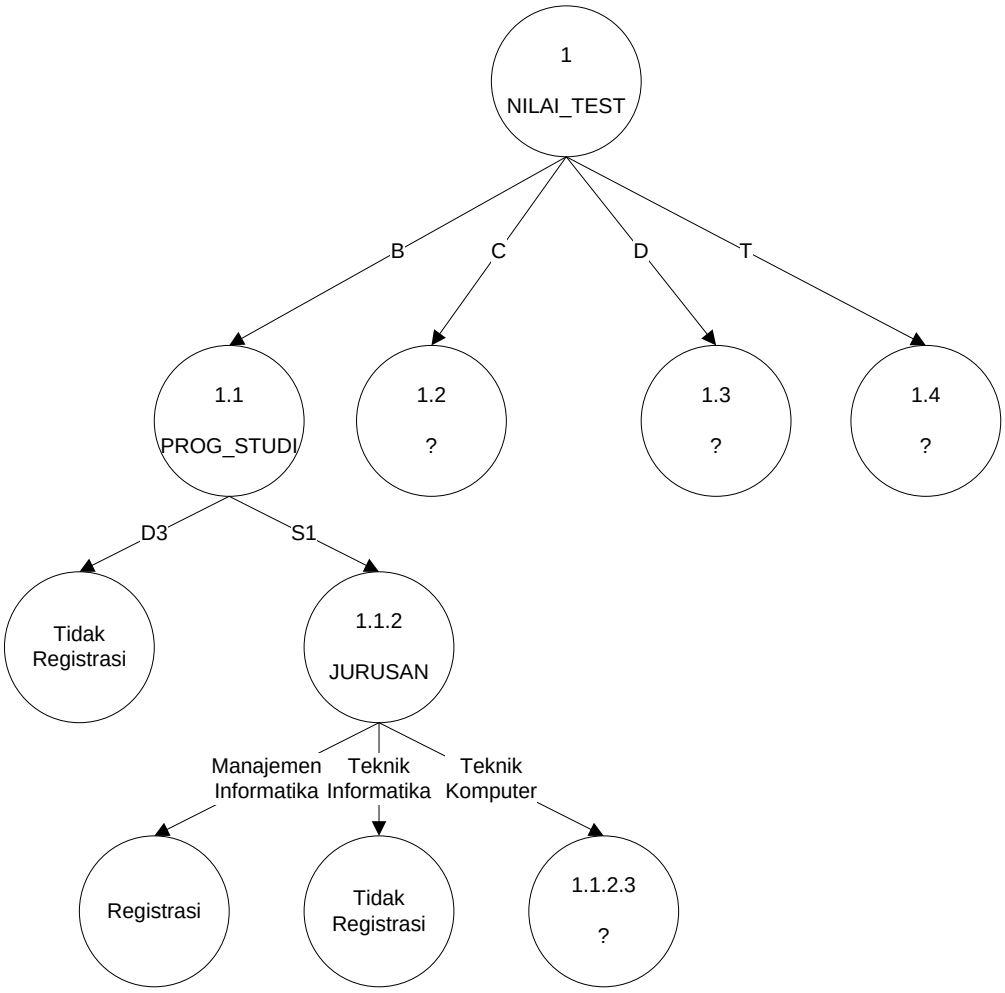
Tabel III.11 Hasil Perhitungan Node 1.1.2

Node			Jml kasus (S)	Tidak Registrasi (Si)	Registrasi (Si)	Entropy	Gain
1.1.2	NILAI_TEST-B dan PROG_STUDI-S1		7	3	4	0.9853	
	NILAI_UAN						0.2917
		1	2	0	2	0	
		2	5	3	2	0.9710	
	GELOMBANG						0.0204
		1	3	1	2	0.9182	
		2	4	2	2	1	
	CATATAN						0
		TEST	7	3	4	0.9853	
		PMDK	0	0	0	0	
	JURUSAN						0.4139
		TEKNIK INFORMATIKA	1	1	0	0	
		MANAJEMEN INFORMATIKA	2	0	2	0	
		TEKNIK KOMPUTER	4	2	2	1	

Dari hasil pada Tabel 3.11 dapat diketahui bahwa atribut dengan *Gain* tertinggi adalah ***JURUSAN***, yaitu sebesar 0.4139. Dengan demikian ***JURUSAN*** dapat menjadi *node* akar. Ada tiga nilai atribut dari ***JURUSAN***, yaitu ***TEKNIK*** INFORMATIKA**,** ***MANAJEMEN*** INFORMATIKA**,** dan ***TEKNIK***

KOMPUTER. Dari ketiga nilai atribut tersebut masih perlu dilakukan perhitungan lagi.

Dari hasil perhitungan node 1.1.2 dapat digambarkan pohon keputusan sementaranya tampak seperti pada Gambar 3.3 berikut ini.



Gambar III.3 Pohon Keputusan Hasil Perhitungan Node 1.1.2

d. Menghitung jumlah kasus, jumlah kasus untuk keputusan ***Registrasi***, jumlah kasus untuk keputusan ***Tidak Registrasi***, dan Entropy dari semua kasus dan kasus yang dibagi berdasarkan atribut ***NILAI_UAN, GELOMBANG, CATATAN***.

Entropy(NILAI_TEST-B dan PROG_STUDI-S1 dan JURUSAN-Teknik Komputer) = $(-\frac{2}{4} * \log_2(\frac{2}{4})) + (-\frac{2}{4} * \log_2(\frac{2}{4})) = 1$

1. Atribut NILAI_UAN

   Entropy(1)= $(-\frac{1}{1} * \log_2(\frac{1}{1})) + (-\frac{0}{1} * \log_2(\frac{0}{1})) = 0$

   Entropy(2)= $(-\frac{2}{3} * \log_2(\frac{2}{3})) + (-\frac{1}{3} * \log_2(\frac{1}{3})) = 0.9182$

2. Atribut GELOMBANG

   Entropy(1)= $(-\frac{0}{1} * \log_2(\frac{0}{1})) + (-\frac{1}{1} * \log_2(\frac{1}{1})) = 0$

   Entropy(2)= $(-\frac{2}{3} * \log_2(\frac{2}{3})) + (-\frac{1}{3} * \log_2(\frac{1}{3})) = 0.9182$

3. Atribut CATATAN

   Entropy(Test)= $(-\frac{2}{4} * \log_2(\frac{2}{4})) + (-\frac{2}{4} * \log_2(\frac{2}{4})) = 1$

   Entropy(PMDK)= 0

   Perhitungan *Gain*

1. Atribut NILAI_UAN

Gain(Total, NILAI_UAN)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{NILAI_UAN}|}{|Total|} * Entropy(\text{NILAI_UAN } i)$

= 1 – $((\frac{1}{4} * 0) + (\frac{3}{4} * 0.9182))$ = 1 – (0 + 0.6886) = 0.3114

2. Atribut GELOMBANG

Gain(Total, GELOMBANG)

= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{GELOMBANG}|}{|Total|} * Entropy(\text{GELOMBANG } i)$

= 1 – $((\frac{1}{4} * 0) + (\frac{3}{4} * 0.9182))$ = 1 – (0 + 0.6886) = 0.3114

3. Atribut CATATAN

Gain(Total, CATATAN)

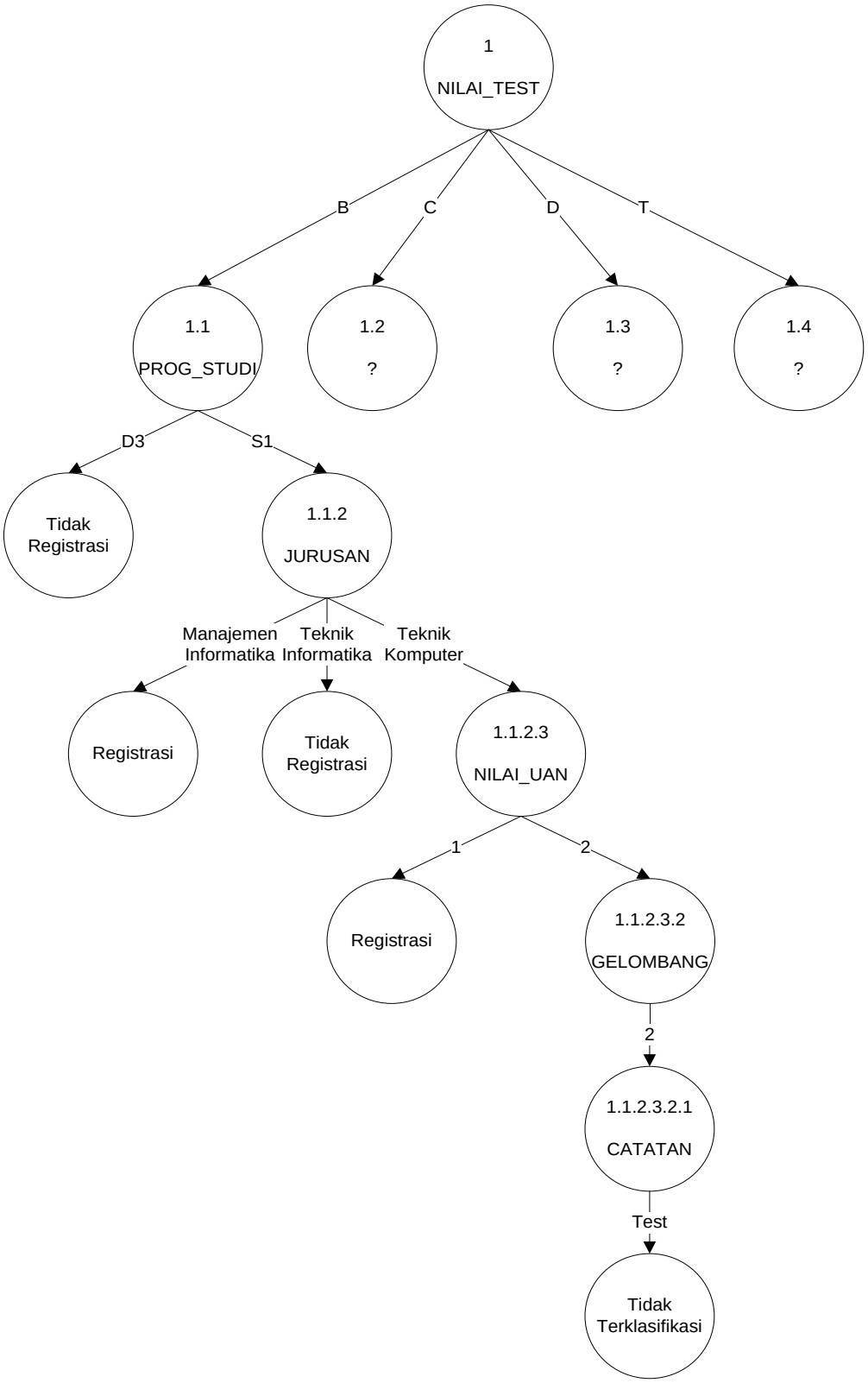
= *Entropy(Total)* – $\sum_{i=1}^{n} \frac{|\text{CATATAN}|}{|Total|} * Entropy(\text{CATATAN } i)$

= 1 – (($\frac{4}{4}$ * 1) + ($\frac{0}{7}$ * 0)) = 1 – (1 + 0) = 0

Hasil Perhitungan nilai Entropy dan nilai *Gain* setiap Atribut ditunjukkan oleh Tabel 3.12 berikut ini.

Tabel III.12 Hasil Perhitungan Node 1.1.2

Node			Jml kasus (S)	Tidak Registrasi (Si)	Registrasi (Si)	Entropy	Gain
1.1.2.3	NILAI_TEST-B dan PROG_STUDI-S1 dan JURUSAN Teknik Komputer		4	2	2	1	
	NILAI_UAN						0.3114
		1	1	0	1	0	
		2	3	2	1		
	GELOMBANG						0.3114
		1	1	0	1	0	
		2	3	2	1		
	CATATAN						0
		TEST	4	2	2	1	
		PMDK	0	0	0		



Gambar III.4 Pohon Keputusan Hasil Perhitungan Terakhir

Dengan memperhatikan pohon keputusan pada gambar 3.4, diketahui bahwa semua kasus sudah masuk dalam kelas. Dengan demikian, pohon keputusan pada Gambar 3.4 merupakan pohon keputusan terakhir yang dibentuk untuk atribut NILAI_TEST dengan nilai atribut B.

3.1.3.1 Daftar Aturan

Berdasarkan pohon keputusan pada gambar 3.4, maka langkah terakhir dalam menganalisis adalah menyajikan daftar aturan yang terbentuk berdasarkan pohon keputusan.

Node akar dari pohon keputusan diatas adalah atribut NILAI_TEST dengan nilai atribut terdiri dari B, C, D, dan T. Calon Mahasiswa dengan nilai tes B diklasifikasikan lagi berdasarkan atribut PROG_STUDI. Atribut PROG_STUDI memiliki nilai atribut yaitu D3 dan S1. Calon mahasiswa dengan program studi D3 diklasifikasikan tidak registrasi. Calon mahasiswa dengn program studi S1 diklasifikasikan lagi berdasarkan atribut JURUSAN. Atribut JURUSAN memiliki nilai atribut yaitu MANAJEMEN INFORMATIKA, TEKNIK INFORMATIKA, dan TEKNIK KOMPUTER. Calon mahasiswa yang memilih jurusan Teknik Komputer diklasifikasikan lagi berdasarkan atribut NILAI_UAN. Atribut NILAI_UAN memiliki nilai atribut yaitu 1 dan 2. Calon mahasiswa dengan nilai uan 2 diklasifikasikan lagi berdasarkan atribut GELOMBANG. Atribut GELOMBANG memiliki nilai atribut yaitu 1 dan 2. Calon mahasiswa yang masuk pada gelombang 2 diklasifikasikan lagi berdasarkan atribut CATATAN. Atribut CATATAN menjadi node akhir dalam pohon keputusan yang terbentuk.

Berikut ini merupakan daftar aturan selengkapnya yang terbentuk berdasarkan pohon keputusan menggunakan Algoritma C4.5.

1 JIKA NILAI_TEST = B
DAN PROG_STUDI = D3
MAKA Tidak Registrasi

2 JIKA NILAI_TEST = B
DAN PROG_STUDI = S1
DAN JURUSAN = Manajemen Informatika
MAKA Registrasi

3 JIKA NILAI_TEST = B
DAN PROG_STUDI = S1
DAN JURUSAN = Teknik Informatika
MAKA Tidak Registrasi

4 JIKA NILAI_TEST = B
DAN PROG_STUDI = S1
DAN JURUSAN = Teknik Komputer
DAN NILAI_UAN = 1
MAKA Registrasi

5 JIKA NILAI_TEST = B
DAN PROG_STUDI = S1
DAN JURUSAN = Teknik Komputer
DAN NILAI_UAN = 2
DAN GELOMBANG = 2
DAN CATATAN = Test
MAKA tidak terklasifikasi

6 JIKA NILAI_TEST = C
DAN GELOMBANG = 1
DAN JURUSAN = Manajemen Informatika
DAN NILAI_UAN = 1
MAKA Tidak Registrasi

7 JIKA NILAI_TEST = C
DAN GELOMBANG = 1
DAN JURUSAN = Manajemen Informatika
DAN NILAI_UAN = 2
MAKA Registrasi

8 JIKA NILAI_TEST = C
DAN GELOMBANG = 1
DAN JURUSAN = Teknik Informatika

MAKA Registrasi

9 JIKA NILAI_TEST = C
DAN GELOMBANG = 1
DAN JURUSAN = Teknik Komputer
MAKA Registrasi

10 JIKA NILAI_TEST = C
DAN GELOMBANG = 2
DAN JURUSAN = Manajemen Informatika
MAKA Registrasi

11 JIKA NILAI_TEST = C
DAN GELOMBANG = 2
DAN JURUSAN = Teknik Informatika
MAKA Tidak Registrasi

12 JIKA NILAI_TEST = C
DAN GELOMBANG = 2
DAN JURUSAN = Teknik Komputer
DAN PROG_STUDI = D3
MAKA Tidak Registrasi

13 JIKA NILAI_TEST = C
DAN GELOMBANG = 2
DAN JURUSAN = Teknik Komputer
DAN PROG_STUDI = S1
DAN NILAI_UAN = 1
DAN CATATAN = Test
MAKA tidak terklasifikasi

14 JIKA NILAI_TEST = D
DAN NILAI_UAN = 1
MAKA Tidak Registrasi

15 JIKA NILAI_TEST = D
DAN NILAI_UAN = 2
DAN GELOMBANG = 1
MAKA Registrasi

16 JIKA NILAI_TEST = D
DAN NILAI_UAN = 2
DAN GELOMBANG = 2
MAKA Tidak Registrasi

17 JIKA NILAI_TEST = T
DAN JURUSAN = Manajemen Informatika
MAKA Registrasi

18 JIKA NILAI_TEST = T
DAN JURUSAN = Teknik Informatika
DAN NILAI_UAN = 1
MAKA Registrasi

19 JIKA NILAI_TEST = T
DAN JURUSAN = Teknik Informatika
DAN NILAI_UAN =2
DAN GELOMBANG = 2
DAN PROG_STUDI = S1
DAN CATATAN = PMDK
MAKA tidak terklasifikasi

20 JIKA NILAI_TEST = T
DAN JURUSAN = Teknik Komputer
MAKA Registrasi

3.1.3.2 Kategori Aturan

Dari aturan-aturan yang terbentuk diatas, ada dua kategori aturan, yaitu aturan yang menghasilkan klasifikasi tertentu (registrasi dan tidak registrasi) dan aturan yang kesimpulannya adalah tidak terklasifikasi. Contoh aturan kategori kedua adalah aturan 5 berikut.

5 JIKA NILAI_TEST = B
DAN PROG_STUDI = S1
DAN JURUSAN = Teknik Komputer
DAN NILAI_UAN = 2
DAN GELOMBANG = 2
DAN CATATAN = Test
MAKA tidak terklasifikasi

Aturan – aturan yang hasilnya adalah tidak terklasifikasi disebabkan oleh semua variabel yang terdefinisikan (NILAI_TEST, PROG_STUDI, JURUSAN, NILAI_UAN, GELOMBANG, CATATAN) sudah muncul dalam *tree* untuk

suatu cabang, tetapi kasus belum bisa mengelompokkan dalam satu klasifikasi yaitu Registrasi atau Tidak Registrasi.

3.1.4 Analisis Kebutuhan Non Fungsioanl

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 pengguna, 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 sistem yang dibangun tidak akan berjalan baik sesuai yang diharapkan.

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

1. Analisis Kebutuhan Pengguna (*User*)
2. Analisis Kebutuhan Perangkat Keras (*Hardware*)
3. Analisis Kebutuhan Perangkat Lunak (*Software*)

3.1.4.1 Analisis Kebutuhan Pengguna (*User*)

Seorang analis yang berada di lingkaran tim manajemen Unikom adalah orang yang biasa menganalisis data-data mahasiswa, salah satunya data calon mahasiswa yang tersimpan pada *database* perusahaan. Hal tersebut berhubungan dengan aplikasi yang akan dibangun, bahwa pengguna (*user*) aplikasi *data mining* menggunakan Algoritma C4.5 ini adalah Seorang Analis dalam tim manajemen Unikom, dimana keterlibatannya dalam menggunakan aplikasi ini harus mengacu pada spesifikasi pengguna sebagai berikut:

1. Terbiasa menggunakan aplikasi yang ada di system operasi Windows.
2. Memiliki pengetahuan mengenai *databases.*
3. Mengetahui atribut yang dianggap kuat untuk dilibatkan dalam proses *data mining.*

Spesifikasi pengguna aplikasi dapat pula dijelaskan pada Tabel 3.11 berikut ini.

Tabel III.13 Spesifikasi pengguna aplikasi

Pengguna	Hak Akses	Pendidikan	Tingkat Keterampilan	Pengalaman
Analis	Melakukan analisis data	Minimal Strata - 1	Bisa mengikuti petunjuk yang ada di sistem	Pelatihan pengguna

Dengan demikian, Analis di Tim Manajemen Unikom sudah termasuk pada kualifikasi kebutuhan pengguna aplikasi Algoritma C4.5 ini.

3.1.4.2 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Berdasarkan pengamatan yang telah dilakukan, dapat diketahui spesifikasi perangkat keras yang digunakan saat ini adalah sebagai berikut:

a. *Processor* 2.10 GHz
b. *Harddisk 320 GB*
c. *RAM 2 GB*
d. Monitor 17 inch dengan resolusi minimal 1024 x 768 pixels

Kebutuhan *hardware* yang disarankan untuk menjalankan aplikasi ini adalah sebagai berikut:

a. Processor 1.5 GHz
b. Harddisk 20 GB
c. RAM 512 MB
d. Monitor 14 inch dengan resolusi minimal 1024 x 768 pixels

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

3.1.4.3 Analisis Kebutuhan Perangkat Lunak (*Software*)

Perangkat lunak merupakan sarana pendukung lainnya bagi pembangunan aplikasi *data mining* ini. Adapun perangkat lunak yang terpasang di komputer Unikom Center saat ini secara umum adalah sebagai berikut:

a. Sistem Operasi Windows XP
b. Microsoft Office
c. Borland Delphi 7
d. Database Borland Interbase

Sedangkan 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 untuk dipelajari.

b. Microsoft Office

c. *Database Management System*

   Database Management System (DBMS) adalah sistem untuk mengelola basis data yang digunakan. Untuk aplikasi *data mining* ini digunakan Borland Interbase sebagai DBMS. Dengan demikian, pengguna disarankan menginstal DBMS yang sesuai dengan kebutuhan untuk menjalankan aplikasi *data mining* ini.

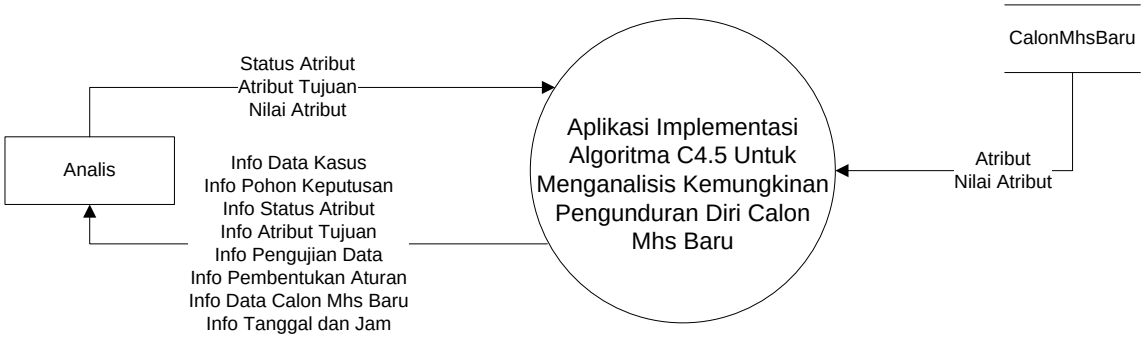
3.1.5 Analisis Kebutuhan Fungsional

Setelah melakukan analisis 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.

3.1.5.1 Diagram Konteks

Diagram konteks merupakan salah satu alat bantu dalam melakukan analisis terstruktur. Diagram konteks ini menggambarkan suatu sistem secara garis besarnya atau keseluruhannya saja. Dalam Diagram konteks juga digambarkan

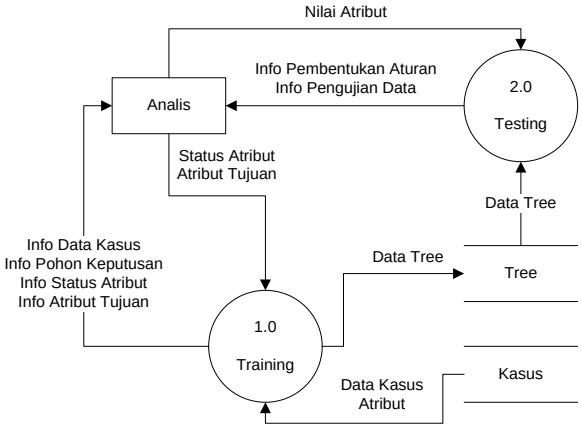
entitas eksternal yang merupakan *brainware* yang menghasilkan data yang akan diolah oleh sistem maupun tujuan dari informasi yang dihasilkan oleh sistem. Adapun diagram konteks yang diusulkan ditunjukan oleh Gambar 3.5 berikut ini.



Gambar III.5 Diagram Konteks

3.1.5.2 Diagram Alir Data Level 1

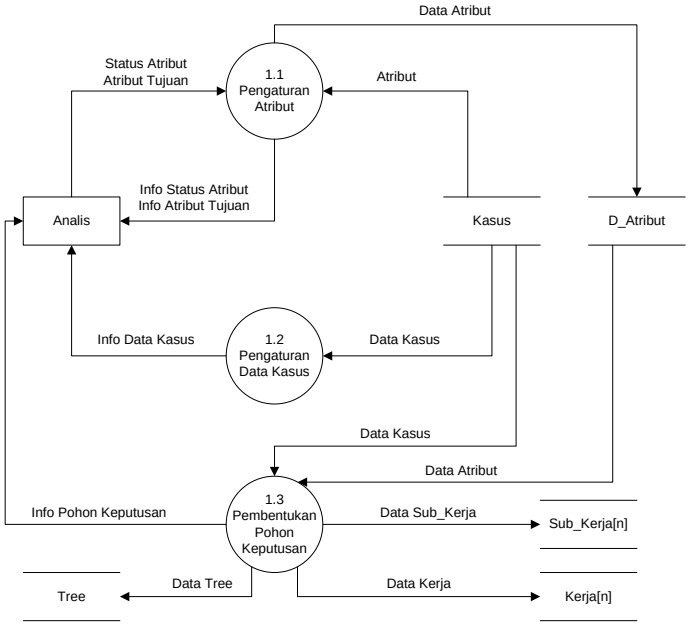
Diagram alir data merupakan sebuah representasi dari suatu sistem yang menggambarkan bagian-bagian dari sistem tersebut beserta keterkaitan antara bagian-bagian yang ada. Dari diagram alir data ini seseorang bisa mengetahui sumber dari informasi di dalam sistem maupun tujuan dari masukan yang berasal dari entitas eksternal. Adapun diagram alir data level 1 dari aplikasi *Data Mining* ditunjukan oleh Gambar 3.6 berikut ini.



Gambar III.6 DFD level 1

3.1.5.3 Diagram Alir Data Level 2 Proses Training

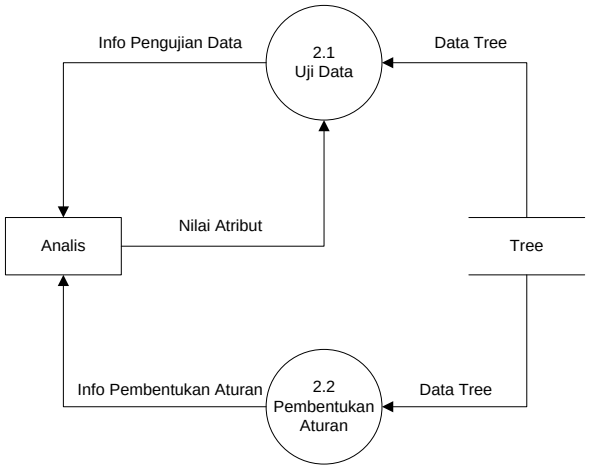
Berikut ini adalah diagram alir data Proses Training ditunjukkan pada Gambar 3.7 dibawah ini.



Gambar III.7 DFD Level 2 Proses Training

3.1.5.4 Diagram Alir Data Level 2 Proses Testing

Berikut ini adalah diagram alir data Proses Training ditunjukkan pada Gambar 3.8 dibawah ini.



Gambar III.8 DFD Level 2 Proses Testing

3.1.5.5 Spesifikasi Proses

Proses-proses yang terdapat pada Diagram Alir Data akan dijelaskan lebih terperinci pada spesifikasi proses, sebagaimana diperlihatkan pada Tabel 3.14 berikut ini.

Tabel III.14 Spesifikasi Proses

No.	Proses	Keterangan
1	No. Proses	1.0
	Nama Proses	Proses Training
	Sumber	Analis, Data Kasus
	Input	Status Atribut, Atribut Tujuan, Atribut, Data Kasus
	Output	Info Data Kasus, Info Pohon Keputusan, Info Status Atribut, Info Atribut Tujuan, Data *Tree*
	Deskripsi	Melakukan pengaturan Atribut dan Atribut Tujuan untuk mendapatkan pohon keputusan
	Logika Proses	User memilih Menu Data Trainig User memilih submenu
2	No. Proses	2
	Nama Proses	Proses Testing
	Sumber	Analis, Data Tree
	Input	Nilai Atribut, Data Tree

<table>
<tr><td rowspan="3"></td><td>Output</td><td>Info Pembentukan Aturan, Info Pengujian Data</td></tr>
<tr><td>Deskripsi</td><td>Melakukan proses pengujian terhadap data terhadap pol aturan yang sudah terbentuk</td></tr>
<tr><td>Logika Proses</td><td>User memilih menu Testing
User memilih submenu</td></tr>
<tr><td rowspan="7">3</td><td>No. Proses</td><td>1.1</td></tr>
<tr><td>Nama Proses</td><td>Pengaturan Atribut</td></tr>
<tr><td>Sumber</td><td>Analis, Data Kasus</td></tr>
<tr><td>Input</td><td>Status Atribut, Atribut Tujuan, Atribut</td></tr>
<tr><td>Output</td><td>Info Status Atribut, Info Atribut Tujuan, Data Atribut</td></tr>
<tr><td>Deskripsi</td><td>Melakukan pengaturan terhadap atribut yang akan digunakan sebagai atribut penentu dan atribut tujuan</td></tr>
<tr><td>Logika Proses</td><td><pre>
procedure
TFrmSettingAtribut.BtnSimpanClick(Sender:
TObject);
var i : integer;
is_salah : boolean;
begin
if EdtAtribut.Text = '' then
begin
MessageDlg('Pilih nama atributnya!',
mtInformation, [mbOK],0);
CbAktif.SetFocus;
Exit;
end;
with DM.Q1 do
begin
SQL.Text :=
' update d_atribut set '+
' is_aktif = :is_aktif '+
' where nama_atribut = :namalama ';
ParamByName('namalama').AsString :=
LVAtribut.Selected.Caption;
if CbAktif.Checked = true then
ParamByName('is_aktif').AsString := 'Y'
else ParamByName('is_aktif').AsString := 'T';
try
ExecSQL;
except
end;
try
IF DM.IBT.Active = False THEN
DM.IBT.Active := True;
DM.IBT.Commit;
except
end;
end;
//buat tabel kasus
tampil;
end;
</pre></td></tr>
<tr><td rowspan="5">4</td><td>No. Proses</td><td>1.2</td></tr>
<tr><td>Nama Proses</td><td>Pengaturan Data Kasus</td></tr>
<tr><td>Sumber</td><td>Data Kasus</td></tr>
<tr><td>Input</td><td>Data Kasus</td></tr>
<tr><td>Output</td><td>Info Data Kasus</td></tr>
</table>

<table>
<tr><td></td><td>Deskripsi</td><td>Menampilkan item data dari data kasus yang akan diproses data mining sesuai atribut yang telah dikonfigurasi</td></tr>
<tr><td></td><td>Logika Proses</td><td><pre>
procedure TFrmKasus.FormShow(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].FieldName :=
Fields[0].AsString;
DBGKasus.Columns[i].Width := 120;
Next;
inc(i);
end;
end;
end;
</pre></td></tr>
<tr><td rowspan="7">5</td><td>No. Proses</td><td>1.3</td></tr>
<tr><td>Nama Proses</td><td>Pembentukan Pohon Keputusan</td></tr>
<tr><td>Sumber</td><td>Data Kasus, Data Daftar_Atribut</td></tr>
<tr><td>Input</td><td>Data Kasus, Data Atribut</td></tr>
<tr><td>Output</td><td>Info Pohon Keputusan, Data Kerja, Data Sub_Kerja, Data Tree</td></tr>
<tr><td>Deskripsi</td><td>Melakukan proses data mining dengan membentuk pohon keputusan menggunakan algoritma C4.5</td></tr>
<tr><td>Logika Proses</td><td><pre>
begin
  try
  with TIBQuery.Create(self) do
  begin
    Name := 'IQ'+ IntToStr(Level);
    Database := DM.DB;
    Transaction := DM.IBT;
  end;
  except
  end;

  with FindComponent('IQ'+ IntToStr(Level)) as
TIBQuery do
  begin
    //cari entropy
    Entropy := 0;
</pre></td></tr>
</table>

		`//cari jumlah kasus`
		`Close;`
		`SQL.Text := 'select count (*) from kasus where 1=1 ';`
		`for j := 0 to level-1 do`
		`SQL.Add(' and '+ Atribut_terpakai[j] + ' =:p' + IntToStr(j));`
		`for j := 0 to level-1 do`
		`ParamByName('p'+IntToStr(j)).AsString := Nilai_Atribut_terpakai[j];`
		`Open;`
		
		`v_jml_kasus := Fields[0].AsInteger;`
		
		`for i := 0 to Jml_Hasil-1 do`
		`begin`
		`Close;`
		`SQL.Text := 'select count(*) from kasus where '+`
		`atribut_hasil + ' = :hasil';`
		`for j := 0 to level - 1 do`
		`SQL.Add(' and '+ Atribut_Terpakai[j] + ' =:p' + IntToStr(j));`
		`for j := 0 to level - 1 do`
		`ParamByName('p' + IntToStr(j)).AsString := Nilai_Atribut_terpakai[j];`
		
		`ParamByName('hasil').AsString := tampil[i];`
		`Open;`
		`if (Fields[0].AsInteger <> 0) and (v_jml_kasus <>0 )then`
		`Entropy := Entropy +`
		`(- Fields[0].AsInteger/v_jml_kasus *`
		`log2 (Fields[0].AsInteger/v_jml_kasus));`
		`end;`

6	No. Proses	2.1
	Nama Proses	Proses Uji Data
	Sumber	Analis, Data Tree
	Input	Nilai Atribut, Data Tree
	Output	Info Pengujian Data
	Deskripsi	Melakukan pengujian data terhadap pola aturan yang sudah terbentuk

	Logika Proses	`begin`
		`if BtnNext.Caption = '&Mulai' then`
		`begin`
		`LVAtribut.Items.Clear;`
		`with DM.Q1 do`
		`begin`
		`//cari induk`
		`Close;`
		`SQL.Text := 'select id_node, node from tree where induk is null';`
		`Open;`
		`EdtAtribut.Text := Fields[1].AsString;`
		`id_node := Fields[0].AsInteger;`
		`Cari_Nilai(id_node);`
		`BtnNext.Enabled := true;`
		`LblHasil.Caption := '';`
		`BtnNext.Caption := '&Selanjutnya'`
		`end;`
		`end else`
		`begin`
		`with DM.Q1, daftar do`
		`begin`

		daftar := LVAtribut.Items.Add;
		Caption := EdtAtribut.Text;
		SubItems.Add(CmbNilai.Items[CmbNilai.Itemindex]);
		close;
		SQL.Text := 'select id_node, node,
		is_atribut '+
		' from tree '+
		' where induk = :induk '+
		' and nilai = :nilai '+
		' ORDER BY ID_NODE ';
		ParamByName('induk').AsInteger := id_node;
		ParamByName('nilai').AsString :=
		Nilai_Combo[CmbNilai.itemindex];
		Open;
		if Fields[2].AsString = 'Y' then
		begin
		EdtAtribut.Text := Fields[1].AsString;
		id_node := Fields[0].AsInteger;
		Cari_Nilai(id_node);
		end else
		begin
		EdtAtribut.Text := '';
		CmbNilai.Items.Clear;
		CmbNilai.Text := '';
		LblHasil.Caption := Fields[1].AsString;
		BtnNext.Caption := '&Mulai';
		end;
		end;
		end;

7	No. Proses	2.2
	Nama Proses	Proses Pembentukan Aturan
	Sumber	Data Tree
	Input	Data Tree
	Output	Info Pembentukan Aturan
	Deskripsi	Menampilkan pola Aturan yang sudah terklasifikasi

	Logika Proses	`var noaturan, level : integer;`
		`daftar : TListItem;`
		`QTemp : TIBQuery;`
		`procedure Tampil (id_node : integer; nilai : string);`
		`begin`
		`inc (level);`
		`try`
		`with TIBQuery.Create(self) do`
		`begin`
		`Name := 'TIQ'+ IntToStr(Level);`
		`Database := DM.DB;`
		`Transaction := DM.IBT;`
		`end;`
		`except`
		`end;`
		`QTemp := FindComponent('TIQ'+ IntToStr(Level)) as TIBQuery;`
		`with daftar do`
		`begin`
		`QTemp.Close;`
		`QTemp.SQL.Text := 'SELECT node, induk, nilai '+`
		`' FROM TREE '+`
		`' WHERE ID_NODE = :id '+`
		`' ORDER BY ID_NODE';`
		`QTemp.ParamByName('id').AsInteger := id_node;`
		`QTemp.Open;`
		`Tampil(QTemp.Fields[1].AsInteger, QTemp.Fields[2].AsString);`
		`dec (level);`
		`QTemp := FindComponent('TIQ'+ IntToStr(Level)) as TIBQuery;`
		`daftar := LVAturan.Items.Add;`
		`Caption := '';`
		`(FindComponent('TIQ'+ IntToStr(Level)) as TIBQuery).Free;`
		`end;`
		`end;`
		`begin`
		`LVAturan.Items.Clear;`
		`noaturan := 1;`
		`//cari induk`
		`with DM.Q1, daftar do`
		`begin`
		`Close;`
		`SQL.Text := 'SELECT ID_NODE, NODE, INDUK, nilai '+`
		`' FROM TREE '+`
		`' WHERE IS_ATRIBUT = ''T''' +`
		`' ORDER BY ID_NODE';`
		`Open;`
		`while not Eof do`
		`begin`
		`//tampilkan dalam listview`
		`daftar := LVAturan.Items.Add;`
		`Caption:= '';`

		SubItems.Add('');
		level := 0;
		if Fields[2].AsString <> '' then
		Tampil(Fields[2].AsInteger,
		Fields[3].AsString);
		daftar := LVAturan.Items.Add;
		Caption := '';
		SubItems.Add('MAKA ' + Fields[1].AsString);
		Application.ProcessMessages;
		Inc(noaturan);
		Next;
		end;
		end;

3.1.5.6 Kamus Data

Kamus data dapat mendefinisikan dengan lengkap data yang mengalir diantara proses, menyimpan data, dan entitas. Data yang mengalir tersebut dapat berupa masukan untuk sistem atau hasil dari proses sistem. Kamus data dibuat berdasarkan arus data yang mengalir pada Konteks Diagram dan Data Alir Diagram, sebagaimana diperlihatkan pada Tabel 3.15 di bawah ini.

Tabel III.15 Kamus Data

Nama Aliran Data	Data Atribut
Where used/how used	Pergi Ke proses 2.1, 2.3
Deskripsi	Berisi data berupa nama_atribut yang diperoleh dari data kasus
Struktur data	nama_atribut + is_aktif + is_hasil + ket
nama_atribut	[A-Z|a-z|0-9]
is_aktif	[A-Z|a-z|0-9]
is_hasil	[A-Z|a-z|0-9]
Ket	[A-Z|a-z|0-9]
Nama Aliran Data	Data Kasus
Where used/how used	Pergi Ke proses 1.0, 2.0, 2.1, 2.2, 2.3
Deskripsi	Berisi data hasil *preprocessing* dari data MhsBaru
Struktur data	Nilai_uan + gelombang + niai_test + program_studi + jurusan + catatan + registrasi
Nilai_uan	[A-Z|a-z|0-9]
gelombang	[A-Z|a-z|0-9]
Nilai_test	[A-Z|a-z|0-9]
Program_studi	[A-Z|a-z|0-9]
	[A-Z|a-z|0-9]
Nama Aliran Data	Data Sub_Kerja
Where used/how used	Pergi Ke proses 2.3
Deskripsi	Berisi data sub_kerja yang merupakan tempat penyimpanan hasil perhitungan *entropy*

Struktur data	nama_atribut+nilai+entropy+result1+result2+jml_kasus
nama_atribut	[A-Z|a-z|0-9]
nilai	[A-Z|a-z|0-9]
entropy	[0-9]
result1	[A-Z|a-z|0-9]
result2	[A-Z|a-z|0-9]
jml_kasus	[0-9]
Nama Aliran Data	Data Kerja
Where used/how used	Pergi Ke proses 2.3
Deskripsi	Berisi data kerja yang merupakan tempat penyimpanan hasil perhitungan *Gain*
Struktur data	nama_atribut+*Gain*
nama_atribut	[A-Z|a-z|0-9]
Gain	[0-9]
Nama Aliran Data	Data Tree
Where used/how used	Pergi Ke proses 2.0, 2.3, 3.0, 3.1, 3.2
Deskripsi	Berisi data kerja yang merupakan tempat penyimpanan hasil perhitungan *Gain*
Struktur data	id_node+node+nilai+induk+is_atribut
id_node	[0-9]
node	[A-Z|a-z|0-9]
nilai	[A-Z|a-z|0-9]
induk	[A-Z|a-z|0-9]
is_atribut	[A-Z|a-z|0-9]

3.2 Perancangan Sistem

Perancangan merupakan bagian dari metodologi pembangunan suatu perangkat lunak yang harus dilakukan setelah melalui tahapan analisis. Pada bagian ini akan dijelaskan perancangan sistem yang dimaksudkan untuk menggambarkan perbedaan antara sistem yang sedang berjalan dengan sistem yang diusulkan. Perancangan sistem ini menggunakan pendekatan sistem terkomputerisasi.

3.2.1 Perancangan Data

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

3.2.1.1 Struktur Tabel

Berikut adalah tabel-tabel yang terdapat dalam basis data yang digunakan dalam aplikasi yang akan dibangun :

Tabel III.16 Tabel D_ATRIBUT

Nama_Field	Tipe	Panjang	Kunci
Nama_atribut	varchar	30	*Primary Key*
is_aktif	char	1	-
is_hasil	char	1	-
Ket	varchar	255	-

Tabel III.17 Tabel KASUS

Nama_Field	Tipe	Panjang	Kunci
Nilai_uan	Varchar	6	-
gelombang	Varchar	255	-
niai_test	Varchar	255	-
program_studi	Varchar	255	-
jurusan	Varchar	255	-
catatan	Varchar	255	-
registrasi	Varchar	255	-

Tabel III.18 Tabel SUB_KERJA

Nama_Field	Tipe	Panjang	Kunci
Nama_atribut	varchar	30	-
nilai	varchar	255	-
entropy	numeric	15, 2	-
result1	varchar	30	-
result2	varchar	30	-
jml_kasus	integer	-	-

Tabel III.19 Tabel KERJA

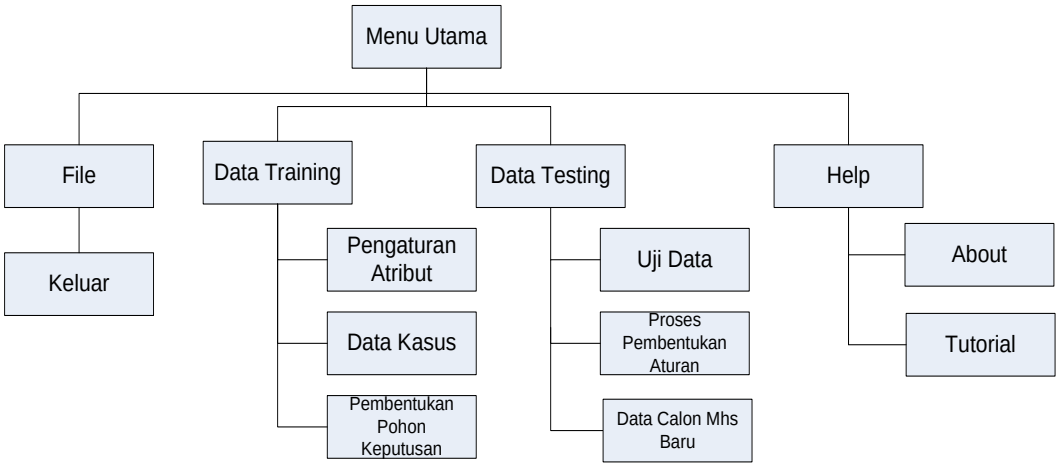
Nama_Field	Tipe	Panjang	Kunci
nama_atribut	varchar	30	-
Gain	float	15,2	-

Tabel III.20 Tabel TREE

Nama_Field	Tipe	Panjang	Kunci
id_node	integer	-	-
node	varchar	30	-
nilai	varchar	30	-
induk	varchar	30	-
is_atribut	char	1	-

3.2.2 Perancangan Menu

Berikut ini adalah perancangan struktur menu yang akan diterapkan pada program dan dirancang seefisien mungkin untuk mengurangi kompleksitas sistem. Struktur menu yang digunakan dapat dilihat pada Gambar 3.9 berikut ini.



Gambar III.9 Struktur Menu

3.2.3 Perancangan Antarmuka

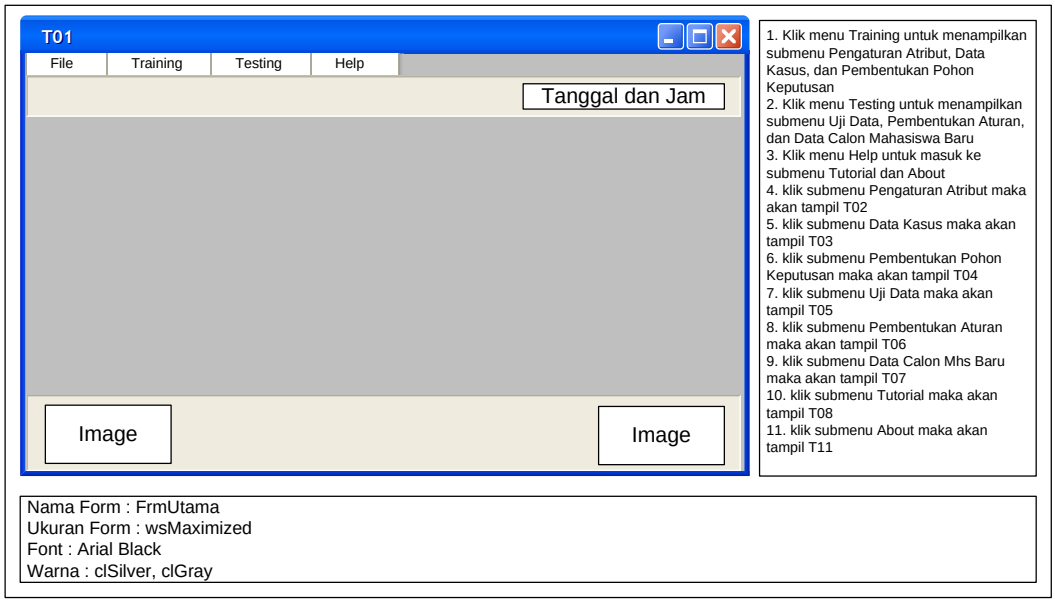
Perancangan antar muka merupakan suatu bentuk tampilan dari program yang akan dibuat untuk kebutuhan *interface* dengan *user*. Spesifikasi antarmuka terdiri dari perancangan tampilan menu, tampilan form, tampilan pesan, keluaran dan jaringan semantik.

3.2.3.1 Perancangan Form

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

1. Perancangan Form Utama

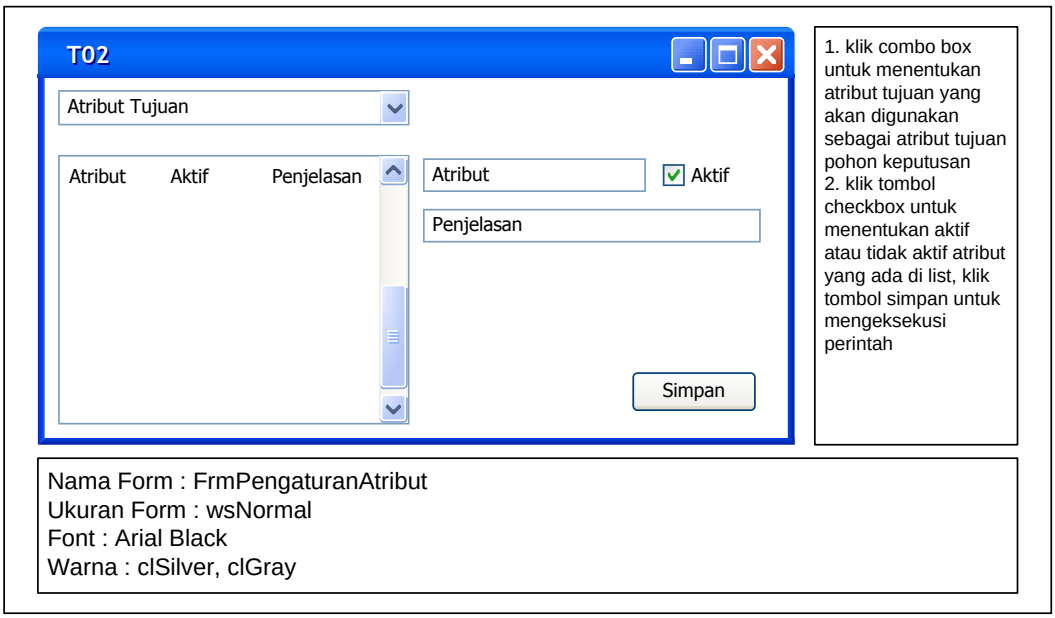
Perancangan form halaman utama aplikasi dapat dilihat pada Gambar 3. 10.



Gambar III.10 Form utama aplikasi

2. Perancangan Form Pengaturan Atribut

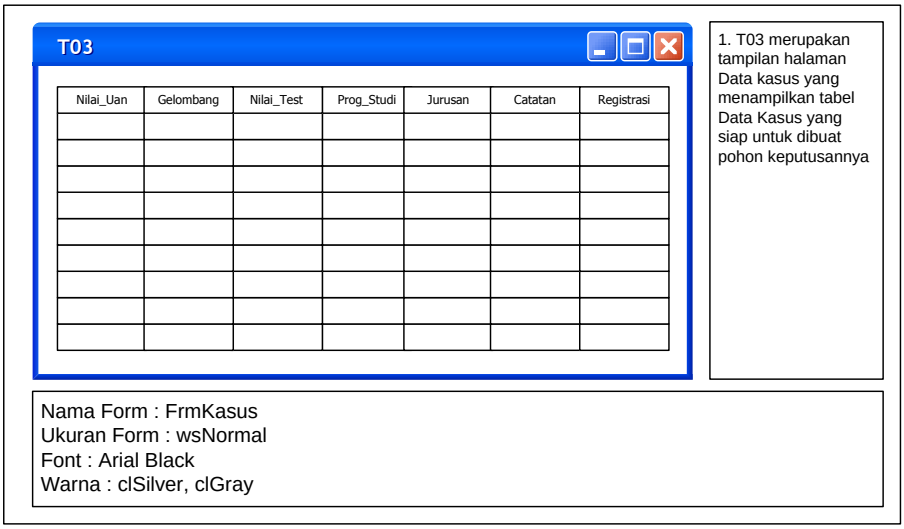
Perancangan form pengaturan atribut dapat dilihat pada Gambar 3.11.



Gambar III.11 Form Pengaturan Atribut

3. Perancangan Form Data Kasus

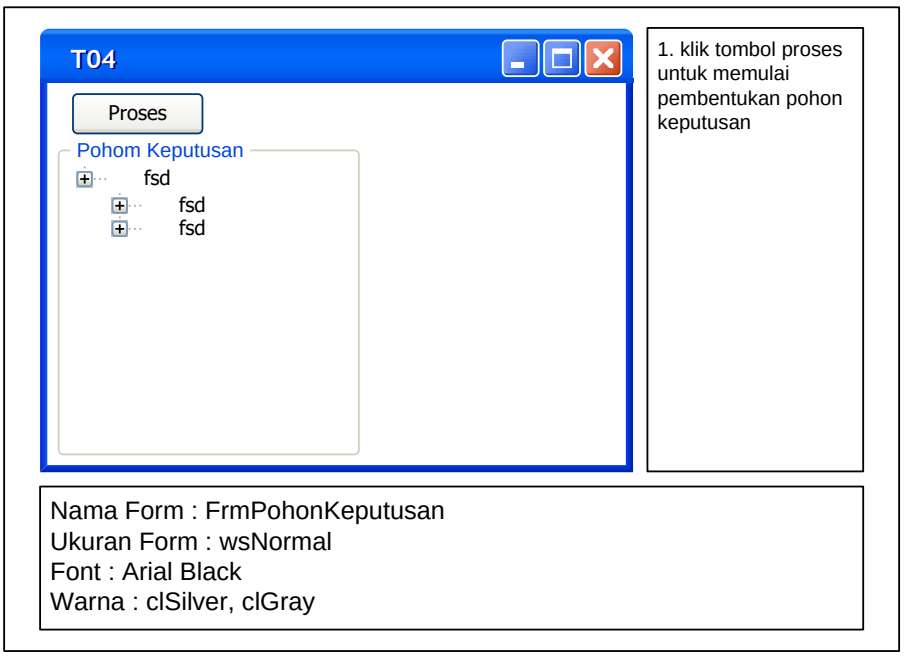
Perancangan tampilan form Data Kasus dapat dilihat pada Gambar 3.12.



Gambar III.12 Perancangan form Data Kasus

4. Perancangan Form Pembentukan Pohon Keputusan

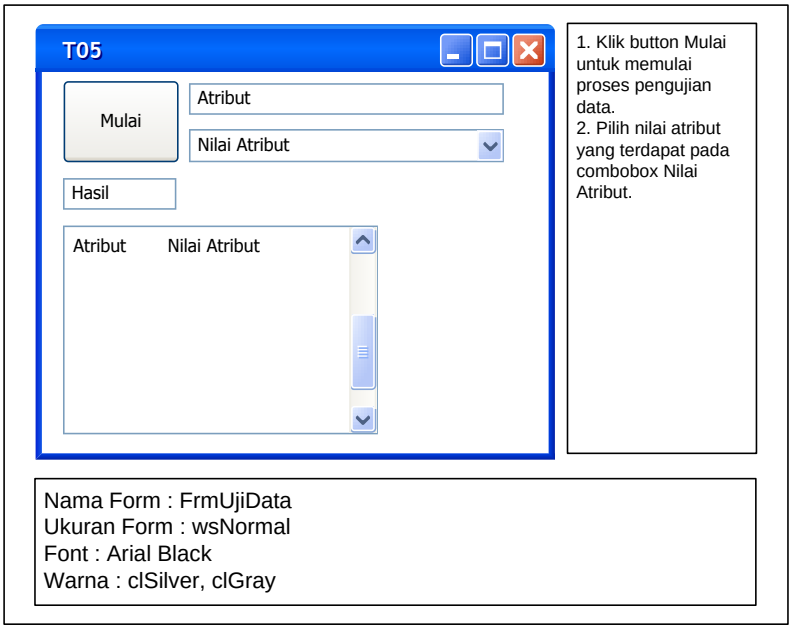
Perancangan tampilan form Pembentukan Pohon Keputusan dapat dilihat pada Gambar 3.13.



Gambar III.13 Perancangan form Pembentukan Pohon Keputusan

5. Perancangan Form Pengujian Data

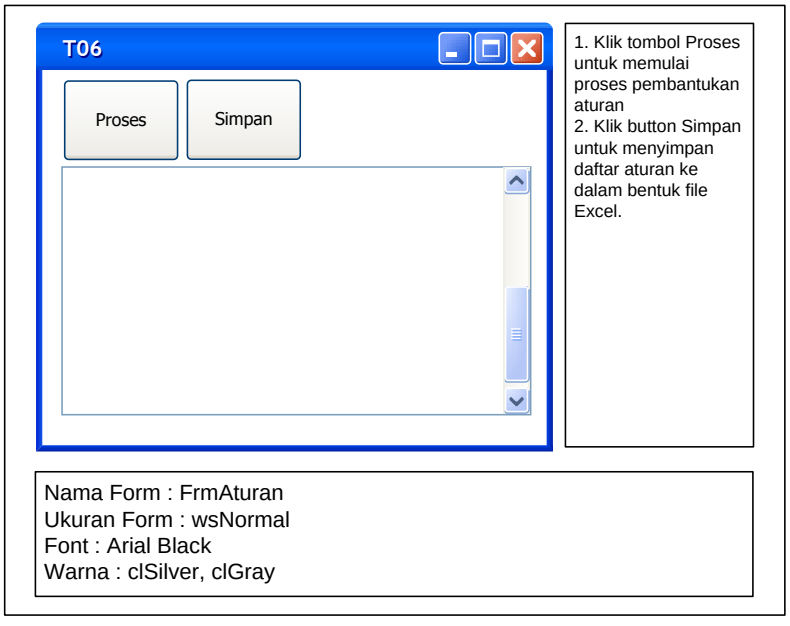
Perancangan tampilan form Pengujian Data dapat dilihat pada Gambar 3.14.



Gambar III.14 Perancangan form Pengujian Data

6. Perancangan Form Pembentukan Aturan

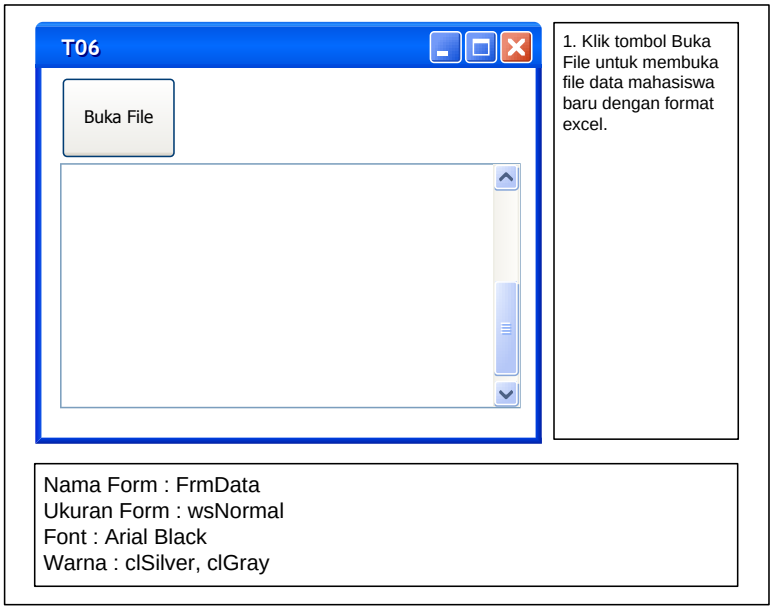
Perancangan form Pembentukan Aturan dapat dilihat pada Gambar 3.15.



Gambar III.15 Perancangan form Pembentukan Aturan

7. Perancangan Form Data Mahasiswa Baru

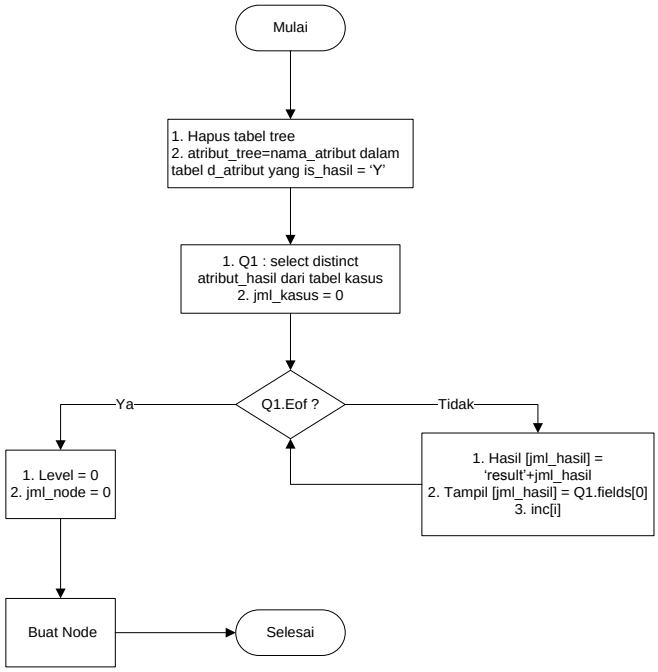
Perancanngan form Data Mahasiswa Baru dapat dilihat pada Gambar 3.16.



Gambar III.16 Perancangan form Data

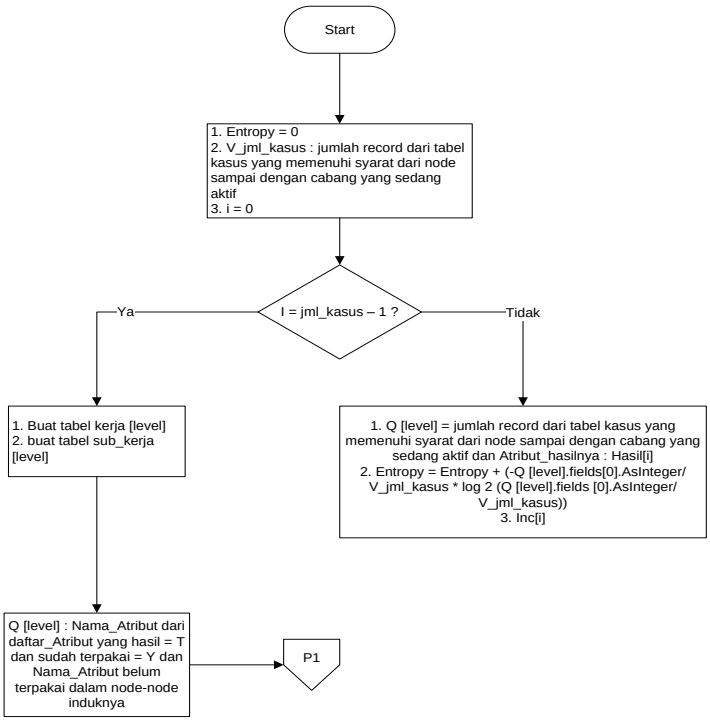
3.2.4 Perancangan Prosedural

Sebagai langkah terakhir dalam perancangan yaitu merancang prosedural yang akan diimplementasikan ke dalam sistem. Prosedural ini akan digunakan sebagai algoritma dasar dalam mengkodekan prosedur yang ada. Dalam aplikasi Algoritma C4.5 ini, pembentukan tree dilakukan secara rekursif. Inisialisasi awal pembentukan tree ditunjukkan oleh Gambar 3.17.

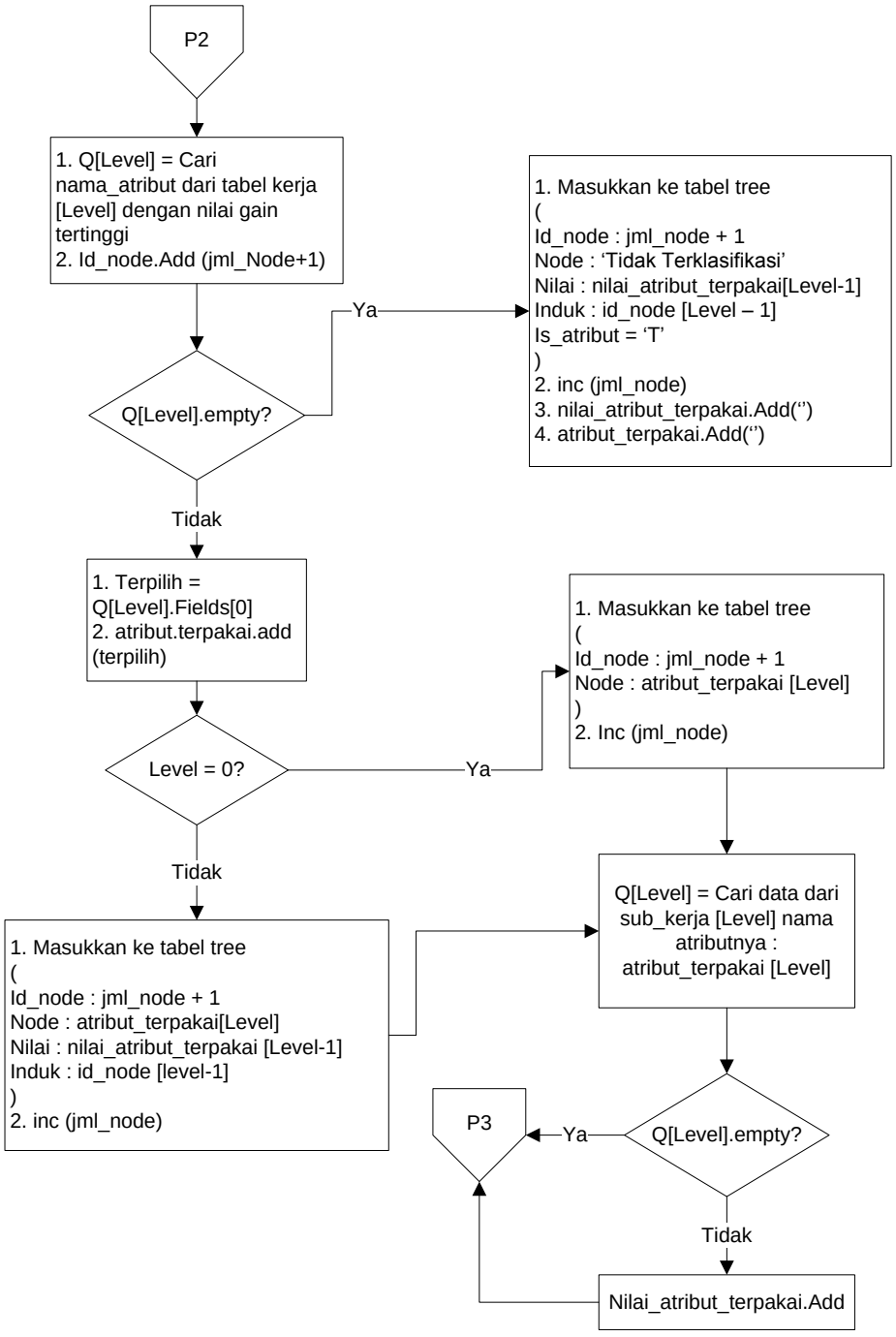


Gambar III.17 Algoritma inisialisasi pembentukan node

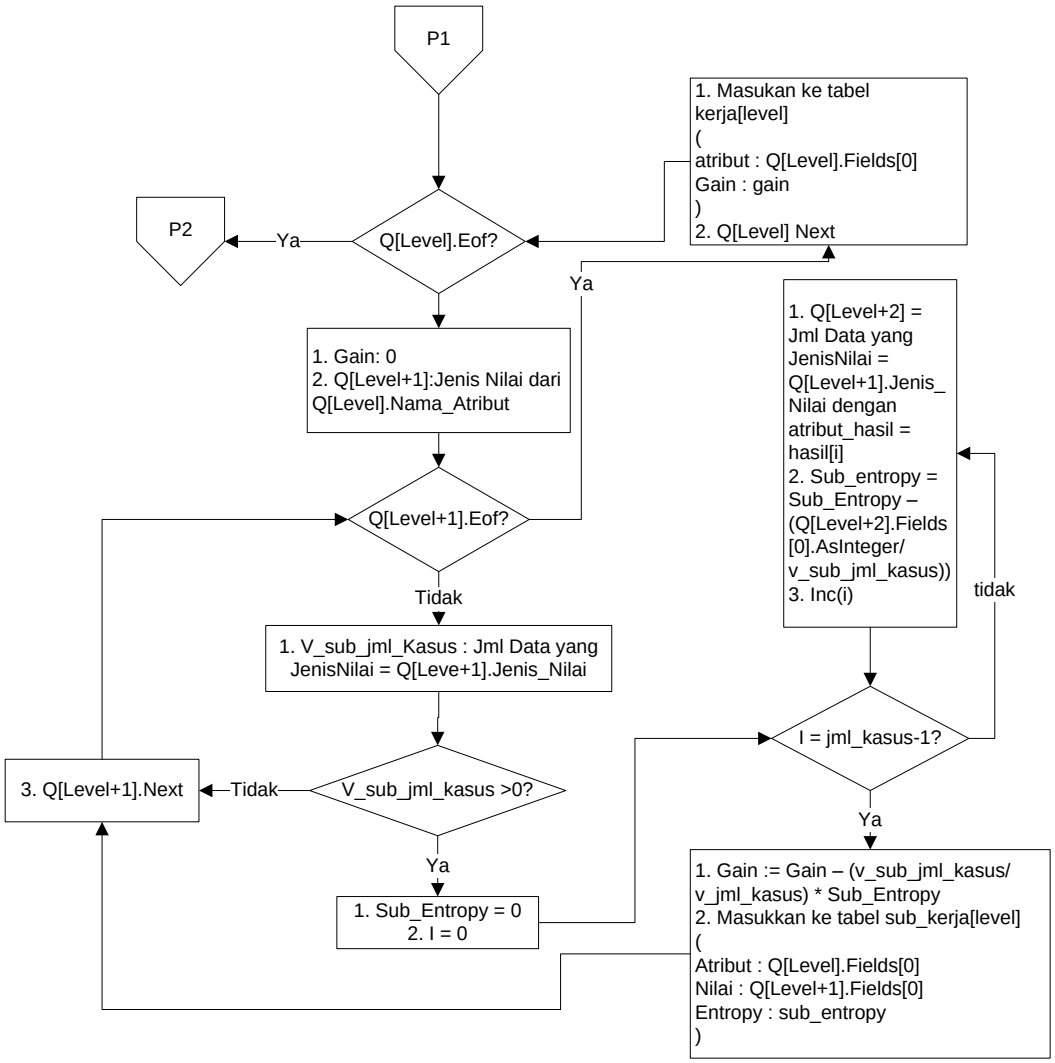
Salah satu langkah pada algoritma inisialisasi pembentukan node adalah buat_node. Algoritma prosedur buat_node ditunjukkan pada Gambar 3.18.



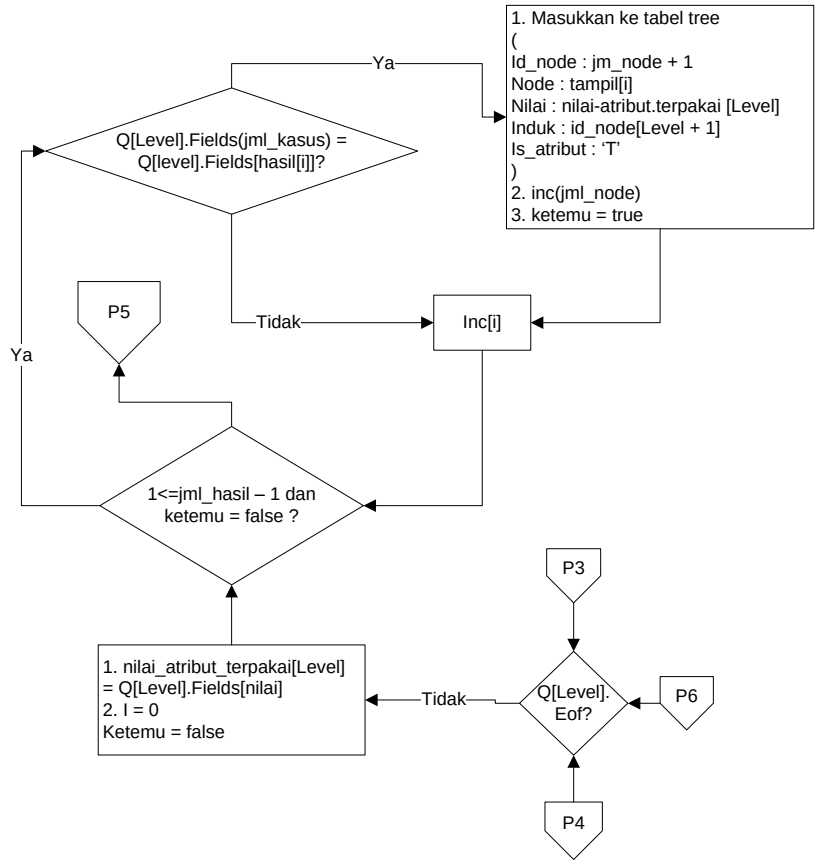
Gambar III.18 Algoritma pembentukan node 1



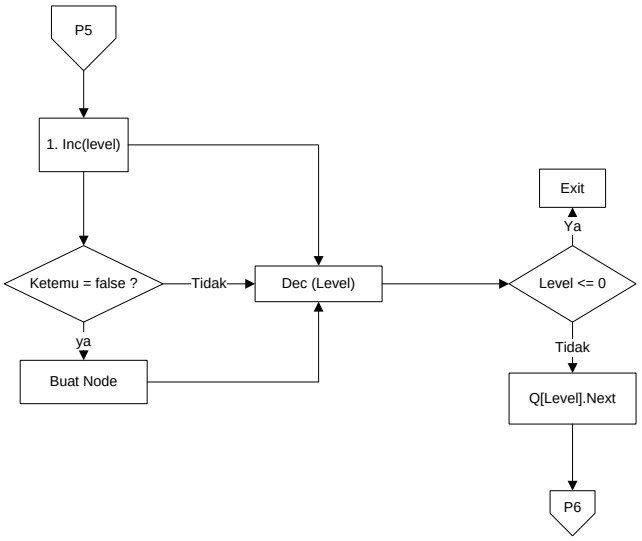
Gambar III.19 Algoritma pembentukan node 2



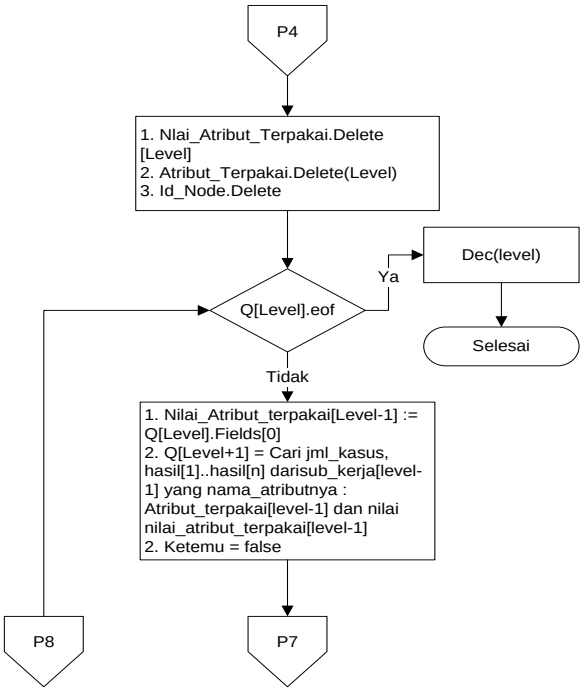
Gambar III.20 Algoritma pembentukan node 3



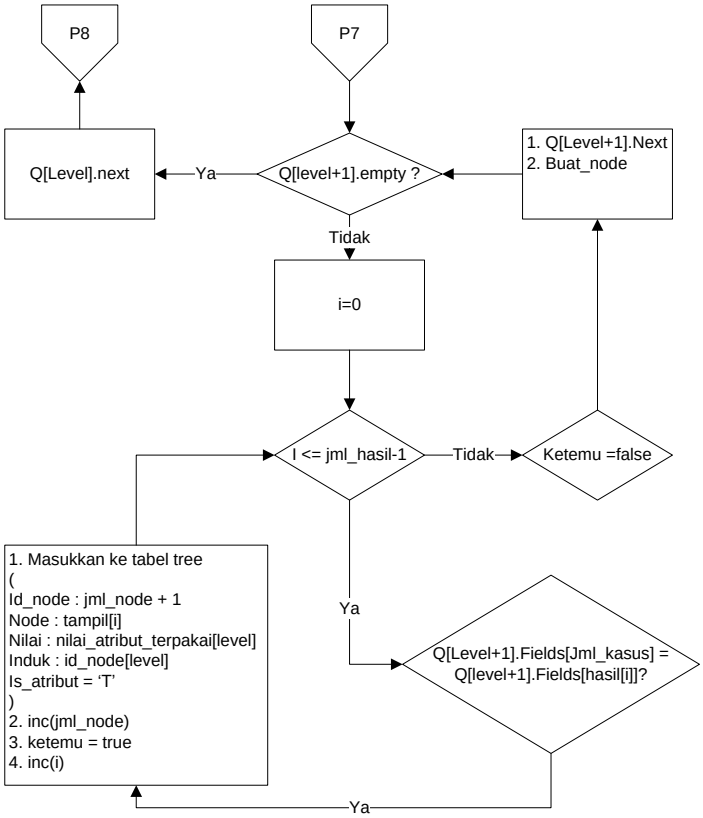
Gambar III.21 Algoritma pembentukan node 4



Gambar III.22 Algoritma pembentukan node 5



Gambar III.23 Algoritma pembentukan node 6



Gambar III.24 Algoritma pembentukan node 7