BAB III

ANALISIS DAN PERANCANGAN SISTEM

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, kesempatan, hambatan yang terjadi dan kebutuhan yang diharapkan sehingga dapat diusulkan perbaikan. Tahap analisis sistem ini merupakan tahap yang sangat kritis dan sangat penting, karena kesalahan di dalam tahap ini akan menyebabkan juga kesalahan di tahap selanjutnya. Tugas utama analis sistem dalam tahap ini adalah menganalisis sistem yang sedang berjalan sehingga dapat diusulkan hasil dari analisis dan perancangan aplikasi sebagai solusi penyelesaian masalah yang lebih baik daripada sistem yang sebelumnya.

3.1.1. Analisis Masalah

Mengidentifikasi masalah merupakan langkah pertama yang dilakukan dalam tahap analisis sistem. Masalah dapat didefinisikan sebagai suatu pertanyaan yang diinginkan untuk dipecahkan. Masalah inilah yang menyebabkan sasaran dari sistem tidak dapat tercapai. Oleh karena itulah langkah pertama yang harus dilakukan pada tahap ini adalah mengidentifikasi terlebih dahulu masalah yang terjadi.

Berdasarkan hasil observasi dan wawancara dengan pihak Bagian Dana didapat kesimpulan permasalahan sebagai berikut,

1. Proses penentuan atau pemilihan target pemasaran oleh Bagian Dana masih mengklasifikasikan semua nasabah yang telah melunasi angsuran kreditnya ke dalam target pemasaran kredit.
2. Seringkali nasabah yang menjadi target pemasaran kredit ternyata tidak tepat sasaran, sehingga tidak efektif dan efisien.

3.1.2. Analisis Sistem yang Sedang Berjalan

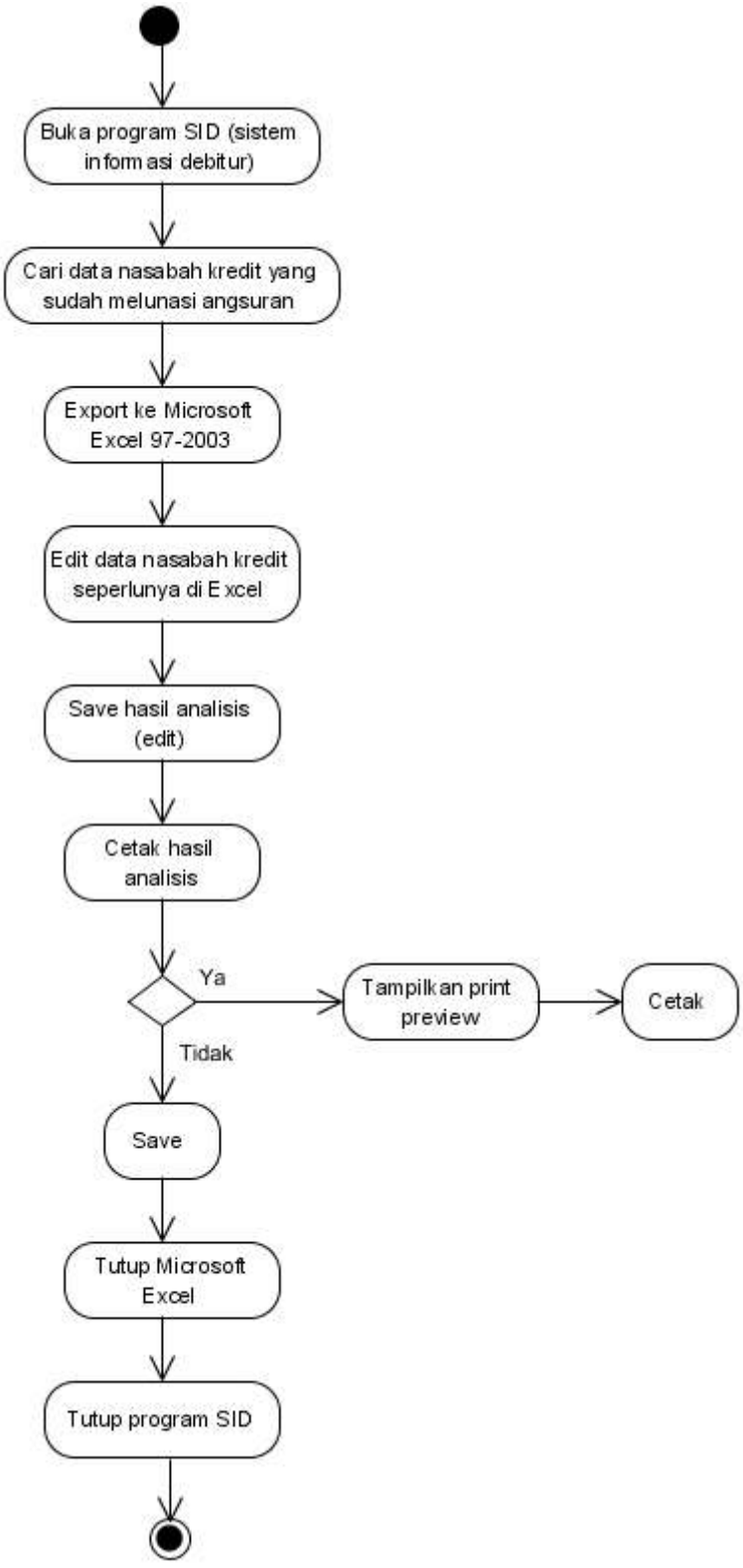
Analisis sistem yang sedang berjalan bertujuan untuk memberikan gambaran mengenai sistem yang sedang berjalan di tempat penelitian secara lebih jelas, sehingga dari hasil analisis tersebut dapat diketahui kelebihan dan kekurangan dari sistem yang dipakai.

3.1.2.1. Prosedur Pencarian Nasabah

Prosedur pencarian nasabah dilakukan oleh seorang pejabat perbankan yang menjabat sebagai Bagian Dana. Dalam melaksanakan tugasnya bagian ini dibantu oleh 6 orang *marketer* (pemasar) yang bertugas untuk menawarkan produk ataupun layanan jasa keuangan perbankan kepada para nasabahnya. Adapun prosedurnya sebagai berikut :

1. Bagian Dana telebih dahulu memeriksa di dalam *database* SID (Sistem informasi Debitur) nasabah-nasabah mana saja yang sudah melunasi angsuran kredit.

2. Setelah ditemukan maka dibuat daftar yang berisi nasabah-nasabah yang sudah melunasi angsuran kreditnya di dalam *microsoft excel*.
3. Bagian Dana kemudian menginstruksikan kepada bawahannya untuk menawarkan produk ataupun layanan jasa keuangan perbankan kepada para nasabah yang sudah melunasi angsuran kreditnya berdasarkan daftar nasabah yang telah dibuat.



Gambar 3.1 *Activity Diagram* Prosedur Pencarian Nasabah

3.1.3. Analisis Data

Berdasarkan hasil observasi yang telah dilakukan, data nasabah kredit PD BPR Kabupaten Bandung Cabang Batujajar tersimpan dalam *database microsoft excel* dengan nama MASTER_KREDIT hasil *import* dari Sistem Informasi Debitur (SID), dimana tabel tersebut terdiri dari 27 *fields.* Berikut struktur tabel MASTER_KREDIT pada table 3.1 dan *sample* isi data *record*-nya bisa dilihat pada tabel 3.2.

Tabel 3.1 Struktur tabel MASTER_KREDIT

Field
NASABAH_ID
NAMA_NASABAH
ALAMAT
JENIS_KELAMIN
TEMPATLAHIR
TGLLAHIR
UMUR
JENIS_ID
KODE_AGAMA
DESA
KECAMATAN
KOTA_KAB
PROPINSI
VERIFIKASI
TGL_REGISTER
KODEPOS
KODE_KANTOR
STATUS_MARITAL
BI_JENIS_PENGGUNAAN
BI_SEKTOR_EKONOMI
JML_PINJAMAN
NILAI_PINJAMAN
JML_ANGSURAN
KODE_INTEGRASI
STATUS
JENIS_AGUNAN
CEK

Tabel 3.2 *Sample Data Record*s tabel MASTER_KREDIT

NASABAH_ID	NAMA_NASABAH	ALAMAT	JENIS_KELAMIN	...
130.000.144	Maesaroh	Rongga 02/06 Cihampelas	P	
130.000.145	Titin Suhartini	Galanggang 01/02 Galanggang	P	
130.000.146	Ani Arniati	Sirah Situ 02/06 Cihampelas	P	

...	TEMPATLAHIR	TGLLAHIR	UMUR	JENIS_ID	KODE_AGAMA	...
	BANDUNG	17-SEP-60	50	KTP	ISLAM	
	BANDUNG	02-FEB-69	41	KTP	ISLAM	
	BANDUNG	22-MAR-76	34	KTP	ISLAM	

...	DESA	KECAMATAN	KOTA_KAB	PROPINSI	...
	CIHAMPELAS	CIHAMPELAS	BANDUNG	JAWABARAT	
	GALANGGANG	BATUJAJAR	BANDUNG	JAWABARAT	
	CIHAMPELAS	CIHAMPELAS	BANDUNG	JAWABARAT	

...	VERIFIKASI	TGL_REGISTER	KODEPOS	KODE_KANTOR	...
	1	30-JUN-09	40562	13	
	1	30-JUN-09	40561	13	
	1	30-JUN-09	40562	13	

...	STATUS_MARITAL	BI_JENIS_PENGGUNAAN	BI_SEKTOR_EKONOMI	...
	MENIKAH	10	6000	
	MENIKAH	10	6000	
	MENIKAH	10	6000	

...	JML_PINJAMAN	NILAI_PINJAMAN	JML_ANGSURAN	...
	1500000	SEDANG	4	
	1500000	SEDANG	3	
	1500000	SEDANG	3	

...	KODE_INTEGRASI	STATUS	JENIS_AGUNAN	CEK
	3	2	8	TIDAK-AKTIF
	1	2	1	TIDAK-AKTIF
	1	1	1	AKTIF

Di dalam teknik *classification* keluaran dari setiap data yang dijadikan parameter target atau *class* harus berupa bilangan bulat atau diskrit. Dari tabel 3.2, kolom atau *field* yang dijadikan parameter target atau variabel keputusannya (*class*) adalah kolom 'CEK' yang berisi nilai parameter AKTIF dan TIDAK-AKTIF. Nilai parameter AKTIF memiliki arti bahwa nasabah masih mempunyai angsuran kredit kepada bank, sedangkan nilai parameter TIDAK-AKTIF memiliki arti bahwa nasabah sudah tidak mempunyai angsuran kredit kepada bank lagi atau dengan kata lain sudah melunasi pembayaran kreditnya. Dengan melihat keluaran atau *records* dari data pada *field* 'CEK' tersebut, yakni berupa bilangan bulat (diskrit), AKTIF dan TIDAK-AKTIF, maka teknik *classification* bisa diterapkan untuk melakukan *data mining* pada data tersebut. Adapun kolom yang bisa diambil sebagai parameter-parameter pembentuk pohon keputusannya adalah :

1. JENIS_KELAMIN
2. UMUR
3. DESA
4. KECAMATAN
5. STATUS_MARITAL
6. NILAI_PINJAMAN
7. JML_ANGSURAN

8. KODE_INTEGRASI

Pemilihan parameter-parameter tersebut berdasarkan nilai parameter yang terkandung di dalamnya. Parameter 'NAMA_NASABAH', dan 'ALAMAT' tidak bisa dipilih dikarenakan jumlah nilai parameternya besar sedangkan parameter 'JENIS_KELAMIN' dan 'STATUS_MARITAL' bisa dipilih karena jumlah nilai parameternya terbatas misalnya parameter 'JENIS_KELAMIN' hanya mempunyai 2 nilai parameter saja yakni L atau P sedangkan parameter 'STATUS_MARITAL' sama hanya mempunyai 2 nilai parameter yakni SINGLE atau MENIKAH. Parameter 'JENIS_ID', 'PROPINSI' tidak bisa dipilih karena memiliki nilai parameter yang sama untuk setiap *record*-nya seperti parameter 'JENIS_ID' memiliki nilai parameter yang sama yakni KTP untuk setiap *record*-nya dan parameter 'PROPINSI' dengan nilai parameter JAWABARAT untuk setiap *record*-nya.

3.1.3.1. Contoh Pembentukan Pohon Keputusan

Berikut ini adalah contoh pembentukan pohon keputusan pada node pertama dengan menggunakan algoritma C4.5 terhadap 10 *sample data records* yang diambil dari *database* MASTER_KREDIT PD BPR Kabupaten Bandung Cabang Batujajar setelah melalui proses penyeleksian parameter. Model pohon keputusan yang akan dibangun diasumsikan menggunakan semua parameter dengan parameter 'CEK' sebagai parameter targetnya atau parameter *class*-nya. Parameter-parameter yang akan digunakan adalah parameter-parameter yang nilai parameternya tidak besar dan tidak sama

untuk setiap *record*-nya, parameter-parameter tersebut adalah JENIS_KELAMIN, UMUR, DESA, KECAMATAN, STATUS_MARITAL, NILAI_PINJAMAN, JML_ANGSURAN, dan KODE_INTEGRASI sedangkan 10 *sample data records* yang akan digunakan dalam contoh pembentukan node pertama pada pohon keputusan bisa dilihat pada tabel 3.3

Tabel 3.3 Sepuluh *Sample Data Record*s

JENIS_KELAMIN	UMUR	DESA	KECAMATAN	STATUS_MARITAL	NILAI_PINJAMAN	JML_ANGSURAN	KODE_INTEGRASI	CEK
P	25	CITAPEN	CIHAMPELAS	Single	SEDANG	4	3	AKTIF
L	25	CITAPEN	CIHAMPELAS	Single	KECIL	3	1	TIDAK-AKTIF
L	32	CITAPEN	CIHAMPELAS	Single	KECIL	4	3	AKTIF
L	32	CIPATIK	BATUJAJAR	Menikah	KECIL	3	1	TIDAK-AKTIF
P	36	CITAPEN	CIHAMPELAS	Menikah	SEDANG	3	3	AKTIF
L	36	BATUJAJAR BARAT	BATUJAJAR	Menikah	KECIL	4	1	TIDAK-AKTIF
L	36	CIPATIK	BATUJAJAR	Menikah	SEDANG	4	1	TIDAK-AKTIF
P	36	CIPATIK	BATUJAJAR	Menikah	SEDANG	3	3	TIDAK-AKTIF
L	52	BATUJAJAR BARAT	BATUJAJAR	Menikah	KECIL	4	3	AKTIF
P	52	BATUJAJAR BARAT	BATUJAJAR	Menikah	SEDANG	3	3	TIDAK-AKTIF

Untuk memilih parameter sebagai *root* maka dilakukan perhitungan nilai gain terlebih dahulu seperti yang tertera pada rumus 1

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

dengan,

S : Himpunan kasus

A : Atribut

n : Jumlah partisi A

$|S_i|$: Jumlah kasus pada partisi ke I

| S | : Jumlah kasus dalam S

Sedangkan untuk perhitungan nilai entropy dapat dilihat pada rumus 2

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

dengan,

S : Himpunan kasus

n : Jumlah partisi S

pi : Proporsi dari Si terhadap S

Setelah nilai gain didapat untuk setiap parameternya, maka parameter dengan nilai gain tertinggi akan dijadikan sebagai *root* pada model pohon keputusan. Hal pertama yang akan dilakukan adalah menghitung jumlah kasus secara keseluruhan, kemudian jumlah kasus untuk *class* AKTIF, jumlah kasus untuk *class* TIDAK-AKTIF, *entropy* dari semua kasus dan menghitung kasus

yang dibagi berdasarkan parameter yang akan digunakan. Setelah itu maka akan dilakukan perhitungan *gain* untuk setiap parameter yang masih tersisa. Berikut langkah-langkah pembentukan pohon keputusan dari 10 sample data records sampai membentuk penuh sebuah pohon keputusan :

1. Perhitungan untuk Node 1
   a. Menentukan *Entropy(S)*. Untuk mencari *entropy* himpunan kasus maka dilakukan perhitungan dengan menggunakan rumus 2 sebagai berikut,

$$Entropy(S) = (-\frac{6}{10} * \log_2(\frac{6}{10})) + (-\frac{4}{10} * \log_2(\frac{4}{10}))$$

$$Entropy(S) = 0.9710$$

   b. Menentukan *root*. Parameter yang dijadikan *root* pada pohon keputusan adalah parameter yang memiliki nilai gain tertinggi, berikut perhitungan gain untuk setiap parameternya dengan menggunakan rumus 1,
      1). Parameter Jenis_Kelamin

$$Gain(S, Jenis_Kela\min) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|Jenis_Kela\min_i|}{|S|} * Entropy(Jenis_Kela\min_i)$$
$$Gain(S, Jenis_Kela\min) = 0.9710 - ((\frac{6}{10} * 0{,}9183) + (\frac{4}{10} * 1))$$

$$Gain(S, Jenis_Kela\min) = \mathbf{0{,}02}$$

      2). Parameter Umur

$$Gain(S, Umur) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|Umur_i|}{|S|} * Entropy(Umur_i)$$
$$Gain(S, Umur) = 0.9710 - ((\frac{2}{10} * 1) + (\frac{2}{10} * 1) + (\frac{4}{10} * 0{,}8113) + (\frac{2}{10} * 1))$$

$$Gain(S, Umur) = \mathbf{0,0465}$$

3). Parameter Desa

$$Gain(S, Desa) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Desa_i\ |}{|\ S\ |} * Entropy(Desa_i)$$

$$Gain(S, Desa) = 0.9710 - ((\frac{3}{10} * 0{,}9183) + (\frac{3}{10} * 0) + (\frac{4}{10} * 0{,}8133))$$

$$Gain(S, Desa) = \mathbf{0,3710}$$

4). Parameter Kecamatan

$$Gain(S, Kecama\tan) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Kecama\tan_i\ |}{|\ S\ |} * Entropy(Kecama\tan_i)$$

$$Gain(S, Kecama\tan) = 0.9710 - ((\frac{6}{10} * 0{,}9183) + (\frac{4}{10} * 1))$$

$$Gain(S, Kecama\tan) = \mathbf{0,02}$$

5). Parameter Status_Marital

$$Gain(S, Status_Marital) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Status_Marital_i\ |}{|\ S\ |} * Entropy(Status_Marital_i)$$

$$Gain(S, Status_Marital) = 0.9710 - ((\frac{3}{10} * 0{,}9183) + (\frac{7}{10} * 0{,}8631))$$

$$Gain(S, Status_Marital) = \mathbf{0,0913}$$

6). Parameter Nilai_Pinjaman

$$Gain(S, Nilai_Pinjaman) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Nilai_Pinjaman_i\ |}{|\ S\ |} * Entropy(Nilai_Pinjaman_i)$$

$$Gain(S, Nilai_Pinjaman) = 0.9710 - ((\frac{5}{10} * 0{,}9710) + (\frac{5}{10} * 0{,}9710))$$

$$Gain(S, Nilai_Pinjaman) = \mathbf{0}$$

7). Parameter Jml_Angsuran

$$Gain(S, Jml_Angsuran) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Jml_Angsuran_i\ |}{|\ S\ |} * Entropy(Jml_Angsuran_i)$$

$$Gain(S, Jml_Angsuran) = 0.9710 - ((\frac{5}{10} * 0{,}7219) + (\frac{5}{10} * 0{,}9710))$$

$$Gain(S, Jml_Angsuran) = \mathbf{0{,}1246}$$

8). Parameter Kode_Integrasi

$$Gain(S, Kode_Integrasi) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Kode_Integrasi_i\ |}{|\ S\ |} * Entropy(Kode_Integrasi_i)$$

$$Gain(S, Kode_Integrasi) = 0.9710 - ((\frac{4}{10} * 0) + (\frac{6}{10} * 0{,}9183))$$

$$Gain(S, Kode_Integrasi) = \mathbf{0{,}42}$$

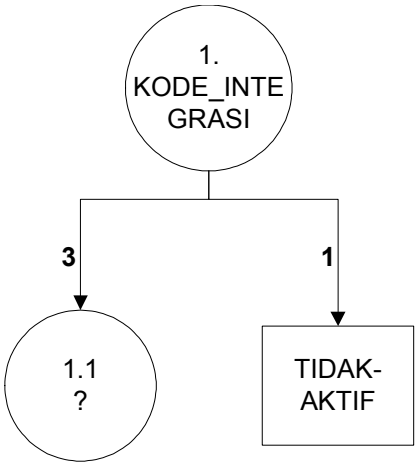
Dari hasil perhitungan dapat diketahui bahwa parameter KODE_INTEGRASI memliki nilai *gain* tertinggi yakni sebesar 0.42 maka parameter ini dapat menjadi *root* atau node pertama dari model pohon keputusan untuk permasalahan *sample data records* pada tabel 3.3. Hasil perhitungan lebih lengkapnya dapat dilihat pada tabel 3.4.

Ada dua nilai yang dimiliki oleh parameter KODE_INTEGRASI yakni 1 dan 3. Dari kedua nilai parameter tersebut, nilai 1 sudah terklasifikasi kepada nilai parameter *class* TIDAK-AKTIF sehingga tidak perlu melakukan perhitungan lebih lanjut sedangkan nilai 3 masih perlu dilakukan perhitungan kembali.

Tabel 3.4 Perhitungan Node 1

Node	Parameter	Nilai Parameter	Jumlah Kasus (S)	TIDAK-AKTIF (S_1)	AKTIF (S_2)	Entropy	Gain
1	TOTAL		10	6	4	0,9710	
	JENIS_KELAMIN						**0,02**
		L	6	4	2	0,9183	
		P	4	2	2	1	
	UMUR						**0,0465**
		25	2	1	1	1	
		32	2	1	1	1	
		36	4	3	1	0,8113	
		52	2	1	1	1	
	DESA						**0,3710**
		BATUJAJAR BARAT	3	2	1	0,9183	
		CIPATIK	3	3	0	0	
		CITAPEN	4	1	3	0,8133	
	KECAMATAN						**0,02**
		BATUJAJAR	6	4	2	0,9183	
		CIHAMPELAS	4	2	2	1	
	STATUS_MARITAL						**0,0913**
		SINGLE	3	1	2	0,9183	
		MENIKAH	7	5	2	0,8631	
	NILAI_PINJAMAN						**0**
		KECIL	5	3	2	0,9710	
		SEDANG	5	3	2	0,9710	
	JML_ANGSURAN						**0,1246**
		3	5	4	1	0,7219	
		4	5	2	3	0,9710	
	KODE_INTEGRASI						**0,42**
		1	4	4	0	0	
		3	6	2	4	0,9183	

Hasil dari perhitungan node pertama ini dapat digambarkan sebagai pohon keputusan seperti pada gambar 3.2



Gambar 3.2 Pohon Keputusan Node 1

2. Perhitungan untuk Node 1.1
   a. Menentukan *Entropy(Kode_Integrasi=3).* Untuk mencari *entropy* himpunan kasus dengan nilai parameter Kode_Integrasi = 3 maka dilakukan perhitungan dengan menggunakan rumus 2 sebagai berikut,

$$Entropy(Kode_Integrasi = 3) = (-\frac{2}{6} * \log_2(\frac{2}{6})) + (-\frac{4}{6} * \log_2(\frac{4}{6}))$$

$$Entropy(Kode_Integrasi = 3) = 0{,}9183$$

   b. Menentukan *node* cabang untuk parameter Kode_Integrasi dengan nilai parameter 3. Parameter yang akan menjadi node cabang adalah parameter yang memiliki nilai gain tertinggi, berikut perhitungan gain untuk setiap parameternya dengan menggunakan rumus 1,

1). Parameter Jenis_Kelamin

$$Gain(S, Jenis_Kela\min) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Jenis_Kela\min_i\ |}{|\ S\ |} * Entropy(Jenis_Kela\min_i)$$
$$Gain(S, Jenis_Kela\min) = 0{,}9183 - ((\frac{2}{6} * 0) + (\frac{4}{6} * 1))$$

$$Gain(S, Jenis_Kela\min) = \mathbf{0{,}2516}$$

2). Parameter Umur

$$Gain(S, Umur) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Umur_i\ |}{|\ S\ |} * Entropy(Umur_i)$$
$$Gain(S, Umur) = 0.9183 - ((\frac{1}{6} * 0) + (\frac{1}{6} * 0) + (\frac{2}{6} * 1) + (\frac{2}{6} * 1))$$

$$Gain(S, Umur) = \mathbf{0{,}2516}$$

3). Parameter Desa

$$Gain(S, Desa) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Desa_i\ |}{|\ S\ |} * Entropy(Desa_i)$$
$$Gain(S, Desa) = 0{,}9183 - ((\frac{2}{6} * 1) + (\frac{1}{6} * 0) + (\frac{3}{6} * 0))$$

$$Gain(S, Desa) = \mathbf{0{,}5850}$$

4). Parameter Kecamatan

$$Gain(S, Kecama\tan) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Kecama\tan_i\ |}{|\ S\ |} * Entropy(Kecama\tan_i)$$
$$Gain(S, Kecama\tan) = 0{,}9183 - ((\frac{4}{6} * 1) + (\frac{2}{6} * 0))$$

$$Gain(S, Kecama\tan) = \mathbf{0{,}2516}$$

5). Parameter Status_Marital

$$Gain(S, Status_Marital) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Status_Marital_i\ |}{|\ S\ |} * Entropy(Status_Marital_i)$$
$$Gain(S, Status_Marital) = 0.9183 - ((\frac{2}{6} * 0) + (\frac{4}{6} * 1))$$

$$Gain(S, Status_Marital) = \mathbf{0,2516}$$

6). Parameter Nilai_Pinjaman

$$Gain(S, Nilai_Pinjaman) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Nilai_Pinjaman_i\ |}{|\ S\ |} * Entropy(Nilai_Pinjaman_i)$$
$$Gain(S, Nilai_Pinjaman) = 0.9183 - ((\frac{2}{6} * 0) + (\frac{4}{6} * 1))$$

$$Gain(S, Nilai_Pinjaman) = \mathbf{0,2516}$$

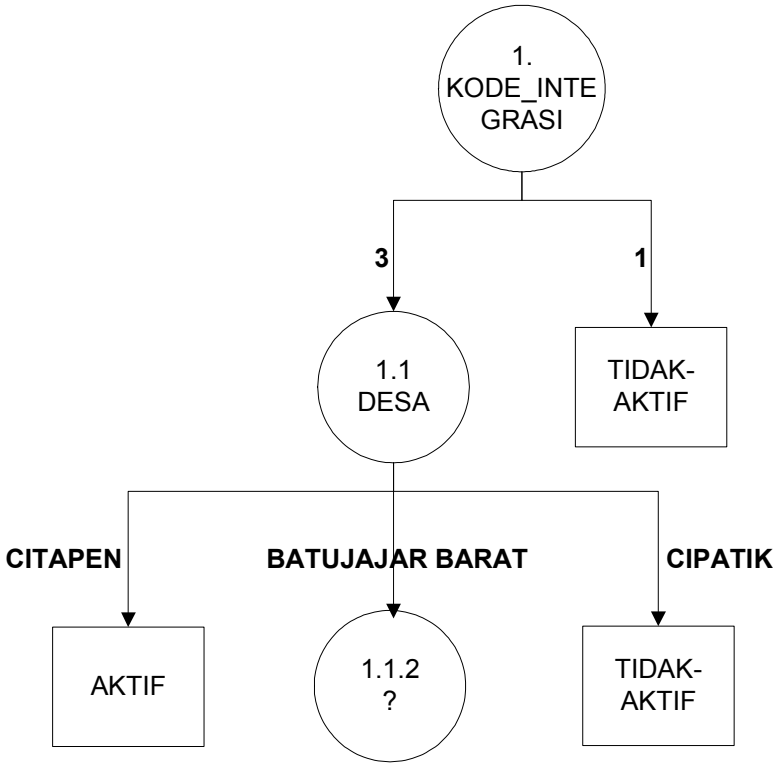
7). Parameter Jml_Angsuran

$$Gain(S, Jml_Angsuran) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Jml_Angsuran_i\ |}{|\ S\ |} * Entropy(Jml_Angsuran_i)$$
$$Gain(S, Jml_Angsuran) = 0{,}9183 - ((\frac{3}{6} * 0{,}9183) + (\frac{3}{6} * 0))$$

$$Gain(S, Jml_Angsuran) = \mathbf{0,4592}$$

Dari hasil perhitungan dapat diketahui bahwa parameter DESA memliki nilai *gain* tertinggi yakni sebesar 0.5850 maka parameter ini dapat menjadi *node* cabang dari parameter KODE_INTEGRASI dengan nilai parameter 3. Ada tiga nilai yang dimiliki oleh parameter DESA yakni BATUJAJAR BARAT, CIPATIK, dan CITAPEN. Dari ketiga nilai parameter tersebut, nilai parameter

CITAPEN sudah terklasifikasi kepada nilai parameter *class* AKTIF dan nilai parameter CIPATIK sudah terklasifikasi kepada nilai parameter *class* TIDAK-AKTIF sehingga tidak perlu melakukan perhitungan lebih lanjut sedangkan untuk nilai parameter BATUJAJAR BARAT masih perlu dilakukan perhitungan kembali dengan cara seperti ini. Pohon keputusan yang terbentuk sampai tahap ini dapat dilihat pada gambar 3.3



Gambar 3.3 Pohon Keputusan Node 1.1

Tabel 3.5 Perhitungan Node 1.1

Node	Parameter	Nilai Parameter	Jumlah Kasus (S)	TIDAK-AKTIF (S_1)	AKTIF (S_2)	Entropy	Gain
1.1	KODE_INTEGRASI = 3		6	2	4	0,9183	
	JENIS_KELAMIN						**0,2516**
		L	2	0	2	0	
		P	4	2	2	1	
	UMUR						**0,2516**
		25	1	0	1	0	
		32	1	0	1	0	
		36	2	1	1	1	
		52	2	1	1	1	
	DESA						**0,5850**
		BATUJAJAR BARAT	2	1	1	1	
		CIPATIK	1	1	0	0	
		CITAPEN	3	0	3	0	
	KECAMATAN						**0,2516**
		BATUJAJAR	4	2	2	1	
		CIHAMPELAS	2	0	2	0	
	STATUS_MARITAL						**0,2516**
		SINGLE	2	0	2	0	
		MENIKAH	4	2	2	1	
	NILAI_PINJAMAN						**0,2516**
		KECIL	2	0	2	0	
		SEDANG	4	2	2	1	
	JML_ANGSURAN						**0,4592**
		3	3	2	1	0,9183	
		4	3	0	3	0	

3. Perhitungan untuk Node 1.1.2

   a. Menentukan *Entropy((Kode_Integrasi=3) dan (Desa=Batujajar))*. Untuk mencari *entropy* himpunan kasus dengan nilai parameter Kode_Integrasi = 3 dan nilai parameter Desa=Batujajar maka dilakukan perhitungan dengan menggunakan rumus 2 sebagai berikut,

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

   b. Menentukan *node* cabang untuk parameter Kode_Integrasi dengan nilai parameter 3 dan parameter Desa dengan nilai parameter Batujajar. Parameter yang akan menjadi node cabang adalah parameter yang memiliki nilai gain tertinggi, berikut perhitungan gain untuk setiap parameternya dengan menggunakan rumus 1,

      1). Parameter Jenis_Kelamin

$$Gain(S, Jenis_Kela\min) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|Jenis_Kela\min_i|}{|S|} * Entropy(Jenis_Kela\min_i)$$
$$Gain(S, Jenis_Kela\min) = 1 - ((\frac{1}{2} * 0) + (\frac{1}{2} * 0))$$
$$Gain(S, Jenis_Kela\min) = \mathbf{1}$$

      2). Parameter Umur

$$Gain(S, Umur) =$$
$$Entropy(S) - \sum_{i-1}^{n} \frac{|Umur_i|}{|S|} * Entropy(Umur_i)$$
$$Gain(S, Umur) = 1 - ((\frac{0}{2} * 0) + (\frac{0}{2} * 0) + (\frac{0}{2} * 0) + (\frac{2}{2} * 1))$$
$$Gain(S, Umur) = \mathbf{0}$$

3). Parameter Kecamatan

$$Gain(S, Kecama\tan) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Kecama\tan_i\ |}{|\ S\ |} * Entropy(Kecama\tan_i)$$

$$Gain(S, Kecama\tan) = 1 - ((\frac{2}{2} * 1) + (\frac{0}{2} * 0))$$

$$Gain(S, Kecama\tan) = \mathbf{0}$$

4). Parameter Status_Marital

$$Gain(S, Status_Marital) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Status_Marital_i\ |}{|\ S\ |} * Entropy(Status_Marital_i)$$

$$Gain(S, Status_Marital) = 1 - ((\frac{0}{2} * 0) + (\frac{2}{2} * 1))$$

$$Gain(S, Status_Marital) = \mathbf{0}$$

5). Parameter Nilai_Pinjaman

$$Gain(S, Nilai_Pinjaman) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Nilai_Pinjaman_i\ |}{|\ S\ |} * Entropy(Nilai_Pinjaman_i)$$

$$Gain(S, Nilai_Pinjaman) = 1 - ((\frac{1}{2} * 0) + (\frac{1}{2} * 0))$$

$$Gain(S, Nilai_Pinjaman) = \mathbf{1}$$

6). Parameter Jml_Angsuran

$$Gain(S, Jml_Angsuran) =$$

$$Entropy(S) - \sum_{i-1}^{n} \frac{|\ Jml_Angsuran_i\ |}{|\ S\ |} * Entropy(Jml_Angsuran_i)$$

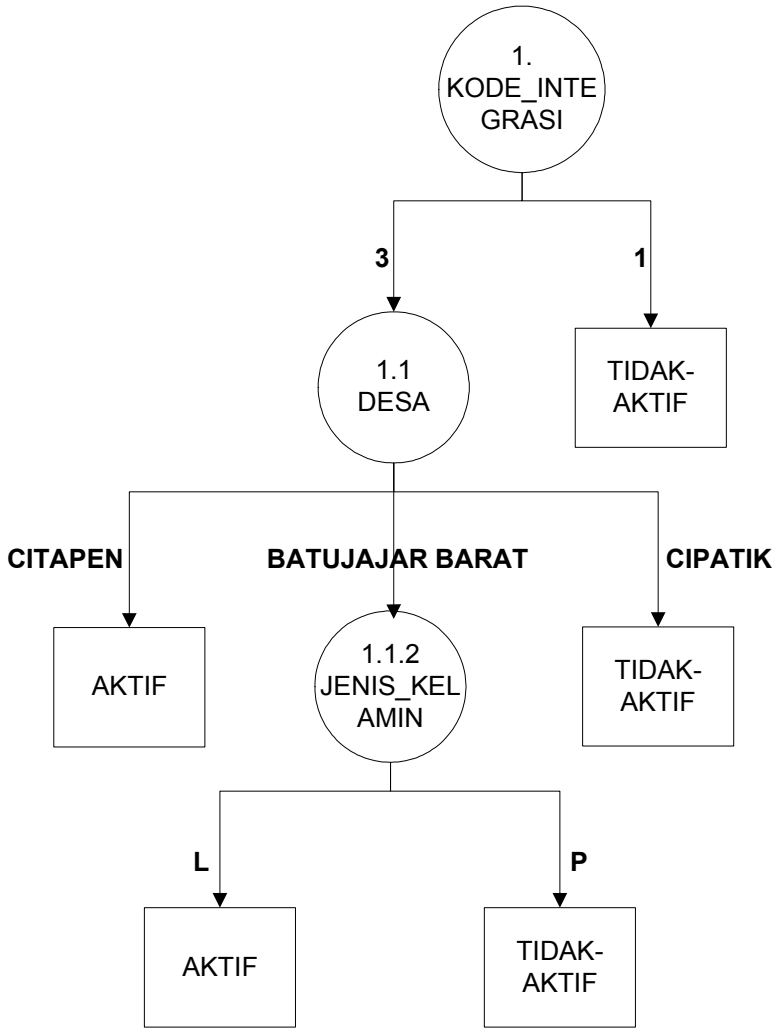
$$Gain(S, Jml_Angsuran) = 1 - ((\frac{1}{2} * 0) + (\frac{1}{2} * 0))$$

$$Gain(S, Jml_Angsuran) = \mathbf{1}$$

Dari hasil perhitungan dapat dilihat terdapat 3 parameter yang mempunyai nilai *gain* tertinggi yakni JENIS_KELAMIN, NILAI_PINJAMAN, dan JML_ANGSURAN dengan nilai *gain* masing-masing sebesar 1. Dikarenakan ada 3 parameter yang sama-sama memiliki nilai *gain* tertinggi maka parameter yang akan menjadi *node* cabang dari parameter DESA dengan nilai parameter BATUJAJAR BARAT adalah parameter yang pertama kali dilakukan perhitungan. Dengan demikian parameter yang akan menjadi node cabang berikutnya adalah parameter JENIS_KELAMIN. Ada dua nilai yang dimiliki oleh parameter JENIS_KELAMIN yakni L dan P. Dari kedua nilai parameter tersebut, nilai parameter L sudah terklasifikasi kepada nilai parameter *class* AKTIF dan nilai parameter P sudah terklasifikasi kepada nilai parameter *class* TIDAK-AKTIF sehingga tidak perlu melakukan perhitungan lebih lanjut. Maka sampai tahap ini proses pembentukan pohon keputusan telah selesai dan pohon keputusan telah terbentuk dengan sempurna. Pohon keputusan yang telah terbentuk sampai tahap ini dapat dilihat pada gambar 3.4

Tabel 3.6 Perhitungan Node 1.1.2

Node	Parameter	Nilai Parameter	Jumlah Kasus (S)	TIDAK-AKTIF (S_1)	AKTIF (S_2)	Entropy	Gain
1.1	KODE_INTEGRASI = 3 dan DESA = BATUJAJAR BARAT		2	1	1	1	
	JENIS_KELAMIN						**1**
		L	1	0	1	0	
		P	1	1	0	0	
	UMUR						**0**
		25	0	0	0	0	
		32	0	0	0	0	
		36	0	0	0	0	
		52	2	1	1	1	
	KECAMATAN						**0**
		BATUJAJAR	2	1	1	1	
		CIHAMPELAS	0	0	0	0	
	STATUS_MARITAL						**0**
		SINGLE	0	0	0	0	
		MENIKAH	2	1	1	1	
	NILAI_PINJAMAN						**1**
		KECIL	1	0	1	0	
		SEDANG	1	1	0	0	
	JML_ANGSURAN						**1**
		3	1	1	0	0	
		4	1	0	1	0	



Gambar 3.4 Pohon Keputusan Node 1.1.2 (utuh)

Setelah pohon keputusan terbentuk secara utuh maka kita bisa membuat aturan prediksi berdasarkan hasil pembentukan pohon keputusan tersebut. Adapun *rules* atau aturan prediksi yang bisa didapat dari pembentukan pohon keputusan pada contoh kasus yang telah dibuat sebelumnya adalah sebagai berikut :

1. Jika Pekerjaannya di bidang Perdagangan
   Dan DESA CITAPEN
   Diprediksikan Berpotensi
2. Jika Pekerjaannya di bidang Perdagangan
   Dan DESA BATUJAJAR BARAT
   Dan Jenis Kelaminnya Laki-Laki
   Diprediksikan Berpotensi
3. Jika Pekerjaannya di bidang Perdagangan
   Dan DESA CIPATIK
   Diprediksikan Tidak Berpotensi
4. Jika Pekerjaannya di bidang Perdagangan
   Dan DESA BATUJAJAR BARAT
   Dan Jenis Kelaminnya Perempuan
   Diprediksikan Tidak Berpotensi
5. Jika Pekerjaannya di bidang Pertanian
   Diprediksikan Tidak Berpotensi

3.1.4. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional dilakukan untuk menghasilkan spesifikasi kebutuhan non fungsional. Spesifikasi kebutuhan non fungsional adalah spesifikasi yang rinci tentang hal-hal yang akan dilakukan sistem ketika diimplementasikan. Analisis kebutuhan non fungsional ini diperlukan untuk menentukan keluaran yang akan dihasilkan sistem, masukan yang

diperlukan sistem, lingkup proses yang digunakan untuk mengolah masukan menjadi keluaran, volume data yang akan ditangani sistem, jumlah pemakai dan kategori pemakai, serta kontrol terhadap sistem.

3.1.4.1. Analisis Perangkat Keras

Berdasarkan hasil tinjauan di lapangan, spesifikasi perangkat keras komputer yang digunakan oleh Bagian Dana di PD BPR Batujajar adalah seperti yang tertera pada tabel 3.7. Spesifikasi perangkat keras komputer yang digunakan oleh Bagian Dana tersebut pada dasarnya sudah mencukupi kebutuhan untuk menjalankan aplikasi *data mining* yang dibangun, namun diperkirakan waktu yang dibutuhkan untuk menganalisis atau melakukan *mining* data akan membutuhkan waktu yang cukup lama. Oleh karena itu direkomendasikan spesifikasi perangkat keras komputer dengan spesifikasi minimal seperti yang tertera pada tabel 3.8

Tabel 3.7 Spesifikasi Perangkat Keras

Perangkat Keras	Spesifikasi
Processor	*Intel Pentium IV* 1.7 Ghz
Memory	256 MB
Harddisk	40 GB
ROM	*CD-ROM*
Keyboard	*Standard*
Mouse	*Optical Mouse*
Monitor	17”

Tabel 3.8 Spesifikasi Perangkat Keras Rekomendasi

Perangkat Keras	Spesifikasi
Processor	*Intel dual core* 1.86 GHz
Memory	1 GB
Harddisk	40 GB
ROM	*CD-ROM*
Keyboard	*Standard*
Mouse	*Optical Mouse*
Monitor	17”

Tabel 3.9 Spesifikasi Perangkat Lunak

Perangkat Lunak	Spesifikasi
Sistem Operasi	*Windows XP*
Development Tool	*Borland Delphi 7*
Database	*MySQL 5*
UML Modeler	*Visual Paradigm for UML 7*

3.1.4.2. Analisis Perangkat Lunak

Kebutuhan perangkat lunak merupakan faktor yang harus dipenuhi untuk merancang sebuah perangkat lunak sehingga perangkat lunak yang telah dibangun akan sesuai dengan maksud dan tujuan perancangan. Adapun kebutuhan spesifikasi perangkat lunak yang dibutuhkan untuk perancangan dan implementasi dapat dilihat pada tabel 3.9

3.1.4.3. Analisis Pengguna

Analisis pengguna dimaksudkan untuk mengetahui siapa saja dan seperti apa karakteristik yang dimiliki oleh pengguna atau *user* yang dapat menggunakan aplikasi yang akan dibuat. Aplikasi yang dibangun akan digunakan oleh seorang *user* atau pengguna yang memegang jabatan sebagai

Bagian Dana di PD. BPR Kabupaten Bandung Cabang Batujajar, adapun karakteristiknya sebagai berikut,

Pendidikan : S1 Ekonomi Manajemen

Umur : 42 Tahun

Kemampuan : Dapat mengoperasikan sistem operasi *microsoft windows*, dapat bekerja dengan *microsoft office*.

Dari pemaparan yang telah disampaikan maka karakteristik pengguna atau *user* sudah memenuhi kriteria untuk dapat menggunakan program aplikasi yang akan dibangun. Hanya saja dibutuhkan pengenalan dan pelatihan kepada pengguna baru untuk dapat mengoperasikan aplikasi ini secara maksimal dan optimal, karena saat ini pengguna sudah terbiasa menggunakan aplikasi *microsoft office excel* sebagai *tools* untuk menganalisis data.

3.1.5. Analisis Kebutuhan Fungsional

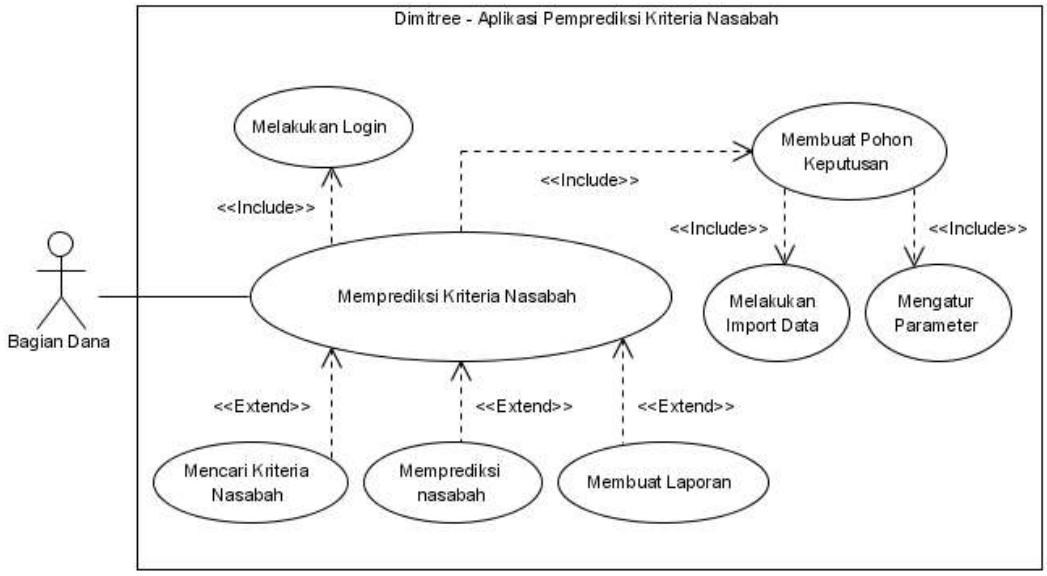
Analisis kebutuhan fungsional dilakukan untuk memberikan gambaran aliran data yang ada pada program aplikasi yang akan dibangun.

3.1.5.1. Model Use Case

Pada bagian ini akan dijelaskan mengenai interaksi antara pengguna dengan aplikasi dengan menggunakan pemodelan use case.

3.1.5.1.1. Diagram Use Case

Berikut diagram *use case* aplikasi data mining yang akan dibangun, dapat dilihat pada gambar 3.5



Gambar 3.5 *Use Case Diagram* Aplikasi *Data Mining*

3.1.5.1.2. Definisi Aktor

Aktor yang terkait dengan aplikasi data mining adalah Bagian Dana. Aktor ini mempunyai wewenang sepenuhnya untuk menangani aplikasi seperti melakukan *login*, meng-*import* data, mengatur parameter, memproses pohon keputusan, melakukan prediksi, menguji kriteria nasabah, membuat laporan, dan mengubah *password* untuk autentifikasi *login* aplikasi.

3.1.5.1.3. Definisi *Use Case*

Berikut akan dijelaskan definisi mengenai masing-masing *use case* yang terdapat pada aplikasi *data mining*, dapat dilihat pada tabel 3.10

Tabel 3.10 Definisi *Use Case* Aplikasi *Data Mining*

No.	*Use Case*	Deskripsi
UC-01	Melakukan *Login*	Melakukan autentifikasi pengguna dengan mengisikan *password login.*
UC-02	Mengimpor Data	Memasukan *database* yang akan dianalisis ke dalam aplikasi dengan format *Microsoft Excel* (*.xls) yang berisikan data-data nasabah kredit PD BPR Batujajar.
UC-03	Mengatur Parameter	Memilih parameter-parameter yang akan dijadikan parameter pembentuk pohon keputusan.
UC-04	Membuat Pohon Keputusan	Membentuk pohon keputusan berdasarkan parameter-parameter yang telah dipilih.
UC-05	Memprediksi Kriteria Nasabah	Melakukan analisis data berdasarkan pohon keputusan yang telah terbentuk sebelumnya dengan menelusuri hasil dari pohon keputusan tersebut kemudian membuat pola data (*data pattern*) berupa aturan prediksi sebagai model untuk memprediksi kriteria nasabah.
UC-06	Mencari Kriteria Nasabah	Melakukan pencarian terhadap kriteria nasabah tertentu berdasarkan pola data (*data pattern*) yang telah terbentuk, apakah tergolong nasabah yang berpotensi untuk melakukan peminjaman kredit atau tergolong nasabah yang tidak berpotensi untuk melakukan peminjaman kredit
UC-07	Memprediksi Nasabah	Melakukan pemprediksian terhadap nasabah – nasabah berdasarkan hasil aturan prediksi yang telah terbentuk.
UC-08	Membuat Laporan	Membuat laporan daftar prediksi kriteria nasabah maupun daftar nasabah yang berpotensi dan yang tidak berpotensi.

3.1.5.1.4. Skenario *Use Case*

Berikut akan dijelaskan secara rinci mengenai skenario *use case-use case* yang terdapat pada aplikasi data mining, penjelasannya dapat dilihat pada tabel 3.11 hingga tabel 3.17

Tabel 3.11 Skenario *Use Case* Melakukan *Login*

Identifikasi		
Nomor	UC-01	
Nama	Melakukan *Login*	
Tujuan	Memastikan hak akses pengguna	
Identifikasi		
Deskripsi	Sistem menerima data identifikasi dari pengguna berupa *password login.* Berdasarkan data tersebut maka sistem akan memeriksa dan memutuskan apakah pengguna memiliki hak akses untuk menggunakan program ini atau tidak.	
Aktor	Bagian Dana	
Skenario Utama		
Kondisi awal	Aplikasi dalam kondisi baru dijalankan (eksekusi).	
Aksi Aktor		**Reaksi Sistem**
		1. Menampilkan halaman *login* yang meminta pengguna untuk memasukan *password login*
2. Memasukan *password login*		
3. Mengklik tombol "*login"*		
		4. Melakukan verifikasi *password login*
		5. Jika sesuai maka tampilkan halaman menu utama aplikasi *data mining*.
Skenario Alternatif – Login gagal		
Aksi Aktor		**Reaksi Sistem**
		1. Menampilkan pesan kesalahan *login*
2. Memasukan *password login* kembali		
3. Mengklik tombol *"login"*		
		4. Melakukan verifikasi ulang.
Kondisi akhir	Hak akses pengguna sudah dipastikan	

Tabel 3.12 Skenario *Use Case* Melakukan *import* Data

Identifikasi	
Nomor	UC-02
Nama	Melakukan *import* Data
Tujuan	Mengimpor data yang akan diolah atau dianalisis
Identifikasi	
Deskripsi	Sistem membaca lokasi alamat data yang diinputkan pengguna, melakukan proses *import* data setelah aktor mengklik tombol "*import"*, kemudian menampilkan hasil data yang telah selesai di impor ke dalam aplikasi.
Aktor	Bagian Dana

Bersambung,

Tabel 3.12 Skenario *Use Case* Melakukan *import* Data (Lanjutan)

Skenario Utama	
Kondisi awal	Pengguna sudah terautentifikasi dan berada pada halaman utama aplikasi *data mining*
Aksi Aktor	**Reaksi Sistem**
1. Mengklik tombol menu "*Import Data"*	
	2. Menampilkan halaman *Import Data*
3. Mengklik tombol "*Browse"*	
	4. Menampilkan halaman *Open Dialog*
5. Mencari lokasi *database* nasabah dalam format *Microsoft Excel* (*.xls)	
6. Setelah ditemukan kemudian klik tombol "*Open*"	
	7. Menampilkan lokasi alamat *database*
8. Klik tombol "*Import Data*" untuk memulai proses *import data*	
	9. Mengimpor data
	10. Menampilkan hasil data yang telah diimpor ke dalam sistem.
Skenario Alternatif – Lokasi data belum ditentukan	
Aksi Aktor	**Reaksi Sistem**
1. Mengklik tombol "*Import Data"*	
	2. Tombol "*Import Data*" tidak aktif atau *disabled.*
2. Mencari lokasi *database* nasabah dalam format *Microsoft Excel* (*.xls).	
Kondisi akhir	Data berhasil diimpor

Tabel 3.13 Skenario *Use Case* Mengatur Parameter

Identifikasi	
Nomor	UC-03
Nama	Mengatur Parameter
Tujuan	Memilih parameter-parameter yang akan digunakan sebagai parameter pembentuk model pohon keputusan
Deskripsi	Sistem menerima pengaturan parameter dari pengguna dan menyimpannya
Aktor	Bagian Dana
Skenario Utama	
Kondisi awal	Pengguna sudah terautentifikasi, telah melakukan *import data*, dan pengguna berada pada halaman utama aplikasi *data mining*
Aksi Aktor	**Reaksi Sistem**
1. Mengklik tombol menu "Parameter*"*	
	2. Menampilkan halaman Parameter

Bersambung,

Tabel 3.13 Skenario *Use Case* Mengatur Parameter (Lanjutan)

Aksi Aktor	Reaksi Sistem
3. Memilih parameter dengan cara mencentangnya.	
4. Mengklik tombol "Simpan"	
	5. Menampilkan konfirmasi simpan
6. Mengklik tombol "Yes"	
	7. Menyimpan hasil pengaturan parameter
	8. Kembali ke halaman utama *data mining*
Skenario Alternatif – Parameter yang dipilih kurang dari 3 buah	
Aksi Aktor	**Reaksi Sistem**
	1. Tombol "Simpan" dalam keadaan *disabled*
2. Memilih 3 buah parameter dengan cara mencentangnya	
Kondisi akhir	Parameter telah dipilih dan pengaturan telah disimpan

Tabel 3.14 Skenario *Use Case* Membuat Pohon Keputusan

Identifikasi	
Nomor	UC-04
Nama	Membuat Pohon Keputusan
Tujuan	Membentuk model pohon keputusan menggunakan algoritma C4.5 berdasarkan parameter-parameter yang telah dipilih oleh pengguna dengan parameter "Cek" sebagai *default* parameter targetnya (*class atribut*)
Deskripsi	Sistem membentuk pohon keputusan dan menampilkan hasil pembentukannya.
Aktor	Bagian Dana
Skenario Utama	
Kondisi awal	Pengguna sudah terautentifikasi, telah melakukan *import data*, telah melakukan pengaturan parameter, dan pengguna berada pada halaman utama aplikasi *data mining*
Aksi Aktor	**Reaksi Sistem**
1. Mengklik tombol menu "Pohon Keputusan*"*	
	2. Menampilkan halaman Pohon Keputusan
3. Mengklik tombol "Proses"	
	4. Membentuk pohon keputusan
	5. Menampilkan hasil pembentukan pohon keputusannya
Kondisi akhir	Pohon keputusan telah terbentuk

Tabel 3.15 Skenario *Use Case* Memprediksi Kriteria Nasabah

Identifikasi		
Nomor	UC-05	
Nama	Memprediksi Kriteria Nasabah	
Tujuan	Membentuk pola data (*data pattern*) sebagai model prediksi.	
Deskripsi	Sistem menelusuri hasil pembentukan model pohon keputusan sehingga terbentuk sebuah pola data (*data pattern*) yang unik yang ditampilkan kedalam bentuk daftar aturan prediksi.	
Aktor	Bagian Dana	
Skenario Utama		
Kondisi awal	Pengguna sudah terautentifikasi, *import data* berhasil, parameter telah dipilih dan disimpan, pohon keputusan telah dibuat, dan pengguna berada pada halaman utama aplikasi *data mining*	
Aksi Aktor		**Reaksi Sistem**
1. Mengklik tombol menu "Prediksi"		
		2. Menampilkan halaman prediksi
3. Mengklik tombol "Proses"		
		4. Menelusuri pohon keputusan
		5. Membuat pola data (*data pattern*)
Kondisi akhir	Pola data (*data pattern*) atau *rule* telah terbentuk	

Tabel 3.16 Skenario *Use Case* Mencari Kriteria Nasabah

Identifikasi		
Nomor	UC-06	
Nama	Mencari Kriteria Nasabah	
Tujuan	Mengetahui hasil prediksi sebuah kriteria nasabah	
Deskripsi	Sistem menggunakan pola data (*data pattern*) yang telah terbentuk sebelumnya sebagai model untuk memprediksi sebuah kriteria tertentu dari nasabah, apakah berpotensi untuk melakukan peminjaman kredit atau tidak berpotensi untuk melakukan peminjaman kredit.	
Aktor	Bagian Dana	
Skenario Utama		
Kondisi Awal	Pengguna sudah terautentifikasi, *import data* berhasil, parameter telah dipilih dan disimpan, pohon keputusan telah dibuat, pola data (*data pattern*) telah terbentuk, dan pengguna berada pada halaman utama aplikasi *data mining*	
Aksi Aktor		**Reaksi Sistem**
1. Mengklik tombol menu "Prediksi Kriteria"		
		2. Menampilkan halaman Prediksi Kriteria

Bersambung,

Tabel 3.16 Skenario *Use Case* Mencari Kriteria Nasabah (Lanjutan)

Aksi Aktor	Reaksi Sistem
3. Memilih nilai parameter	
	4. Menampilkan hasil prediksi berdasarkan kriteria tertentu dari seorang nasabah
Kondisi akhir	Hasil prediksi sebuah kriteria diketahui

Tabel 3.17 Skenario *Use Case* Memprediksi Nasabah

Identifikasi	
Nomor	UC-07
Nama	Memprediksi Nasabah
Tujuan	Mengetahui hasil prediksi nasabah
Deskripsi	Sistem menggunakan pola data (*data pattern*) yang telah terbentuk sebelumnya sebagai model untuk memprediksi nasabah-nasabah, apakah berpotensi untuk melakukan peminjaman kredit atau tidak berpotensi untuk melakukan peminjaman kredit. Data nasabah yang diujikan adalah data yang telah diimpor sebelumnya.
Aktor	Bagian Dana
Skenario Utama	
Kondisi awal	Pengguna sudah terautentifikasi, *import data* berhasil, parameter telah dipilih dan disimpan, pohon keputusan telah dibuat, pola data (*data pattern*) telah terbentuk, dan pengguna berada pada halaman utama aplikasi *data mining*
Aksi Aktor	**Reaksi Sistem**
1. Mengklik tombol menu "Prediksi Nasabah*"*	
	2. Menampilkan halaman Prediksi Nasabah
3. Mengklik tombol "Nasabah Berpotensi" atau "Nasabah Tidak Berpotensi"	
	4. Menampilkan hasil prediksi nasabah-nasabah yang berpotensi atau tidak berpotensi.
Kondisi akhir	Hasil prediksi nasabah diketahui

Tabel 3.18 Skenario *Use Case* Membuat Laporan

Identifikasi		
Nomor	UC-08	
Nama	Membuat Laporan	
Tujuan	Membuat laporan hasil analisis data	
Deskripsi	Sistem membuat laporan berdasarkan hasil analisis data yang telah dilakukan sebelumnya dan menampilkannya ke dalam sebuah bentuk laporan.	
Aktor	Bagian Dana	
Skenario Utama		
Kondisi awal	Pengguna sudah terautentifikasi, *import data* berhasil, parameter telah dipilih dan disimpan, pohon keputusan telah dibuat, pola data (*data pattern*) telah terbentuk, dan pengguna berada pada halaman utama aplikasi data mining	
Aksi Aktor		**Reaksi Sistem**
1. Mengklik tombol menu "Laporan"		
		2. Menampilkan halaman Laporan
3. Memilih salah satu jenis pelaporan		
		4. Menampilkan laporan sesuai dengan pilihan yang dipilih oleh pengguna ketika membuat laporan.
Kondisi akhir	Laporan sudah dibuat	

3.1.5.2. Activity Diagram

Pada bagian ini akan dijelaskan mengenai interaksi antara pengguna dengan aplikasi dengan menggunakan *activity diagram,*

3.1.5.2.1. *Activity Diagram* Melakukan *Login*

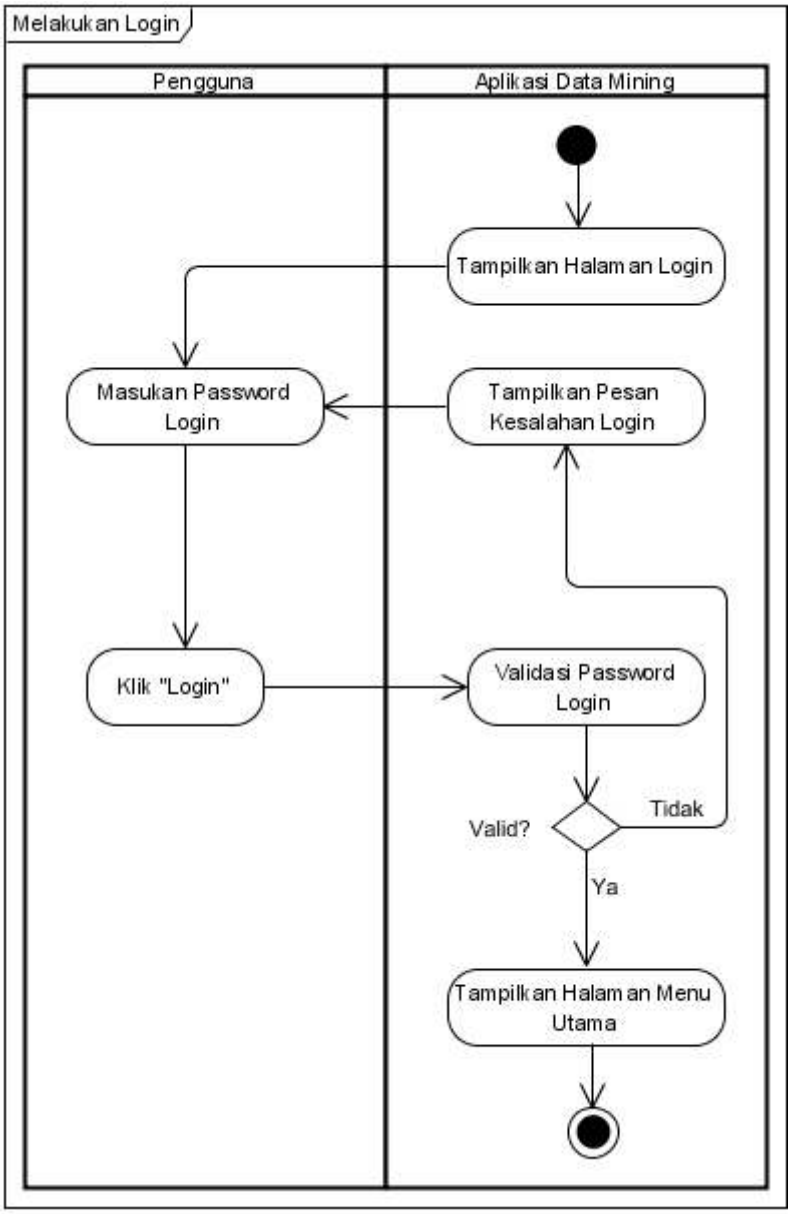
Diagram ini menggambarkan alur interaksi antara pengguna dengan aplikasi dalam melakukan proses *login*. Proses ini merupakan proses yang pertama kali dilakukan ketika menjalankan aplikasi *data mining*. Adapun gambar diagramnya bisa dilihat pada gambar 3.6

3.1.5.2.2. *Activity Diagram* Mengimpor Data

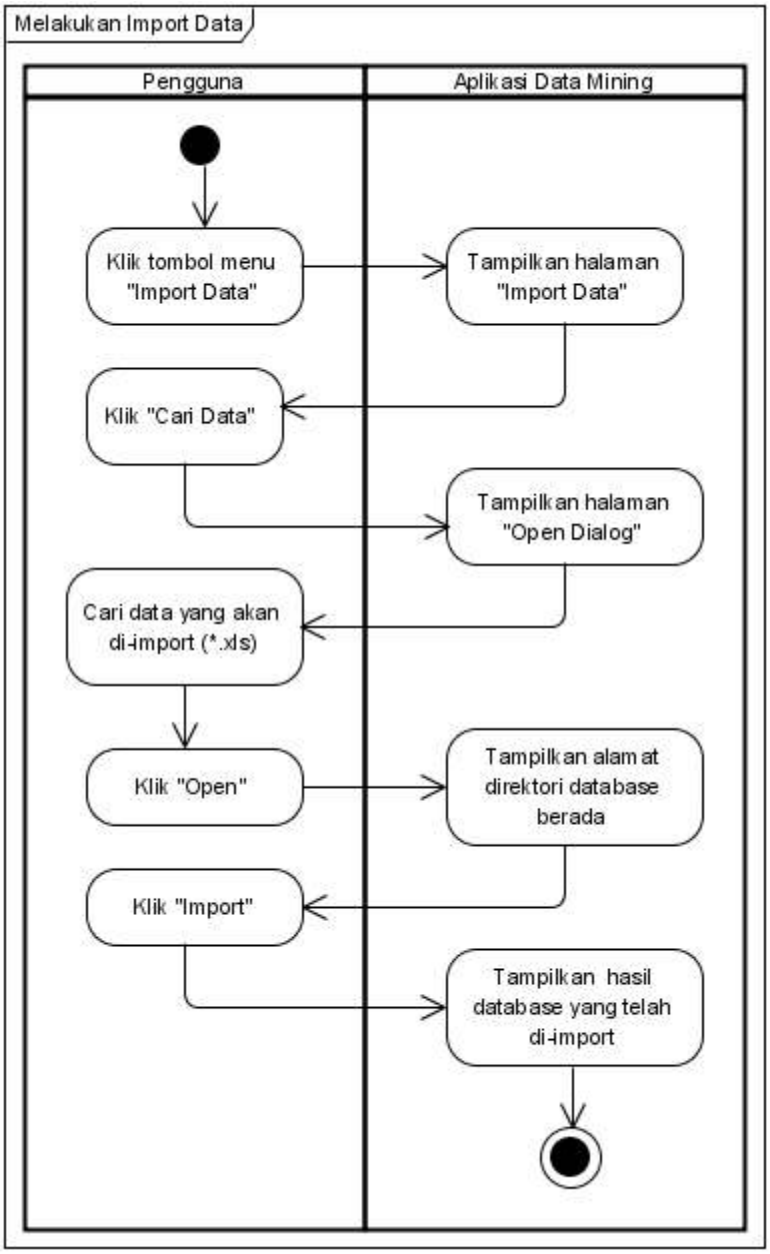
Proses ini menggambarkan rangkaian urutan aktifitas cara melakukan *import* data yang akan dianalisis atau di-*mining* ke dalam aplikasi, bisa dilihat pada gambar 3.7

3.1.5.2.3. *Activity Diagram* Mengatur Parameter

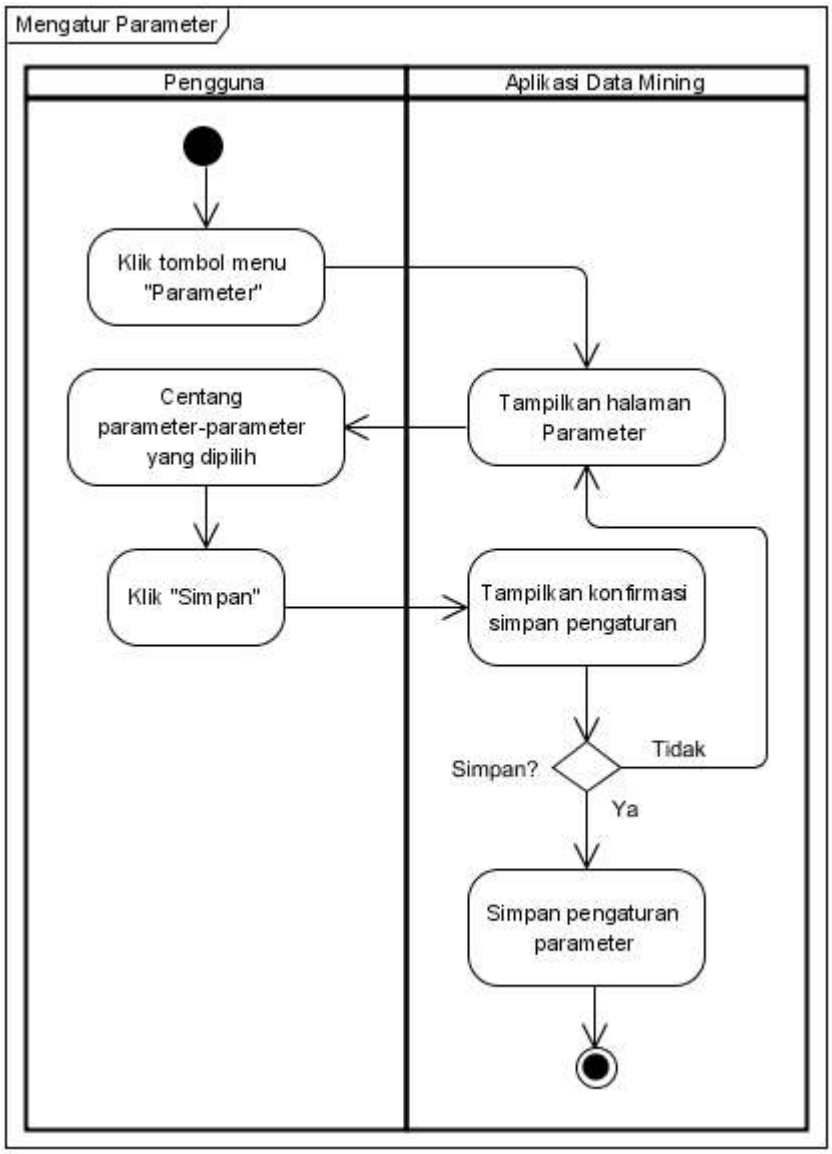
Tahap ini merupakan tahap yang penting karena kunci dari keberhasilan dan keakuratan dalam membentuk model pohon keputusan tergantung kepada parameter yang dipilih pengguna. Hasil dari pengaturan parameter akan berpengaruh juga terhadap hasil prediksi akhir. Gambar diagramnya digambarkan seperti yang terdapat pada gambar 3.8



Gambar 3.6 *Activity Diagram* Melakukan *Login*



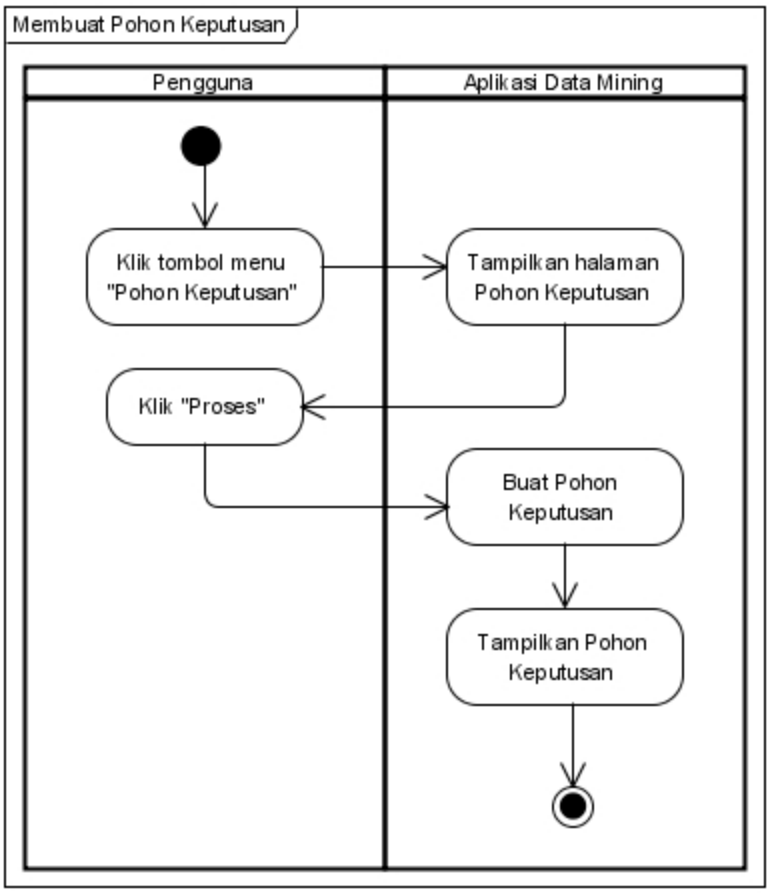
Gambar 3.7 *Activity Diagram* Mengimpor Data



Gambar 3.8 *Activity Diagram* Mengatur Parameter

3.1.5.2.4. *Activity Diagram* Membuat Pohon Keputusan

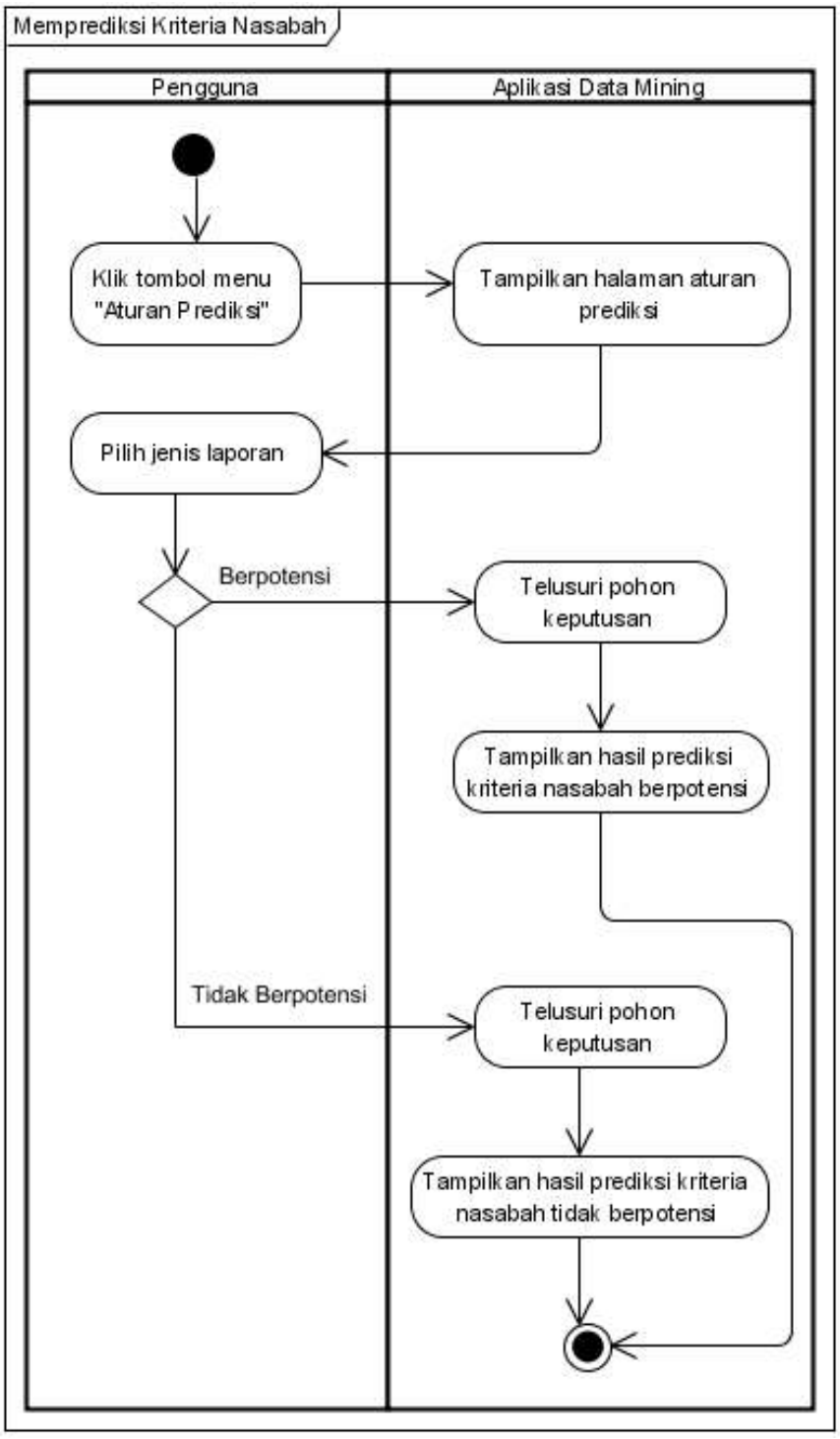
Pada tahap ini pohon keputusan akan dibuat oleh aplikasi berdasarkan parameter-parameter yang telah dipilih oleh pengguna dengan parameter 'CEK' sebagai *default* parameter targetnya atau *class* atributnya, gambar *activity diagram*-nya dapat dilihat pada gambar 3.9.



Gambar 3.9 *Activity Diagram* Membuat Pohon Keputusan

3.1.5.2.5. *Activity Diagram* Memprediksi Kriteria Nasabah

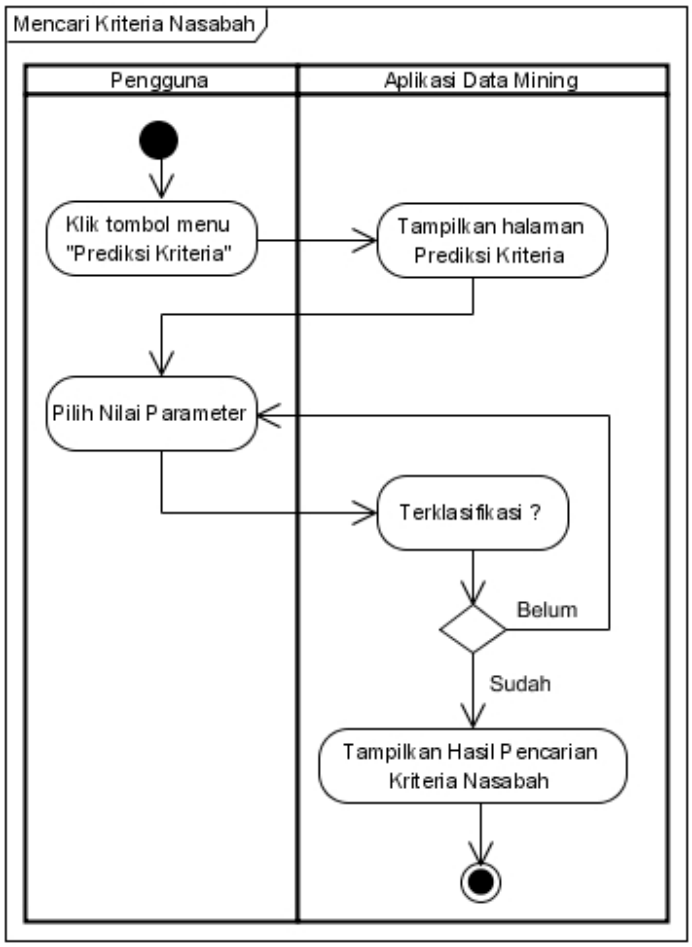
Tahap ini berfungsi untuk memprediksi kriteria nasabah kredit beserta hasil prediksinya, apakah berpotensi atau tidak berpotensi. Tahap ini menelusuri pohon keputusan yang telah terbentuk sebelumnya. Hasil penelusuran tersebut berbentuk aturan prediksi (*rules*) atau pola data (*data pattern*) yang beraturan, adapun *activity diagram* memprediksi kriteria nasabah dapat dilihat pada gambar 3.10.



Gambar 3.10 *Activity Diagram* Memprediksi Kriteria Nasabah

3.1.5.2.6. *Activity Diagram* Mencari Kriteria Nasabah

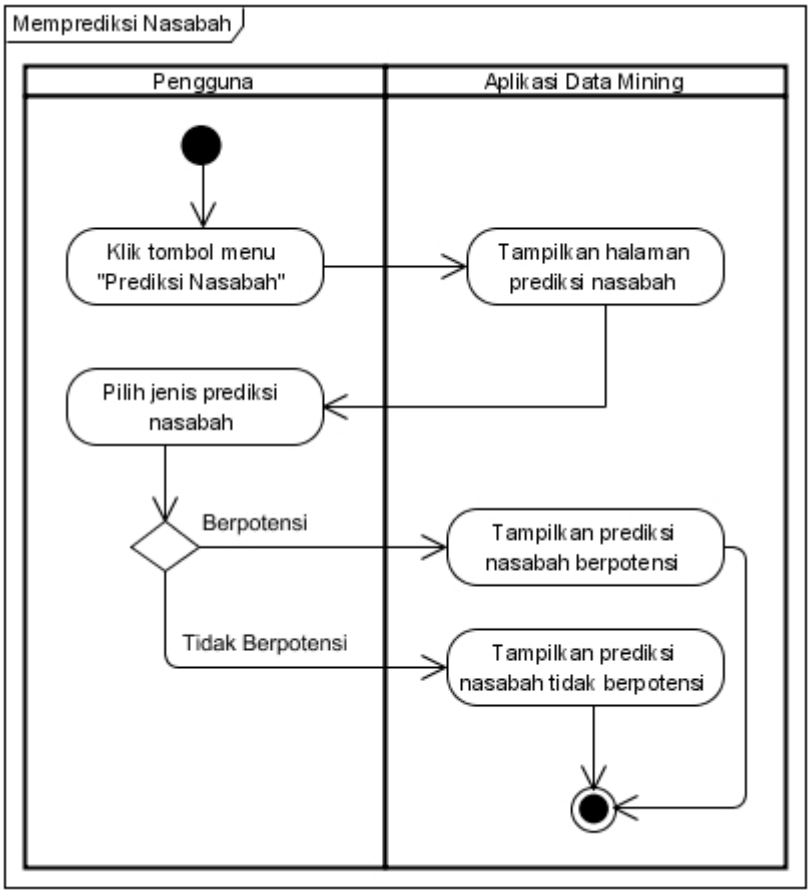
Tahap ini merupakan tahap peramalan atau pemprediksian kriteria tertentu dari seorang nasabah. Pada tahap ini pengguna bisa melakukan pencarian untuk menguji apakah kriteria nasabah tertentu tersebut berpotensi untuk melakukan peminjaman kredit atau tidak berpotensi untuk melakukan peminjaman kredit. Pemprediksian kriteria nasabah ini berdasarkan pola data yang telah terbentuk sebelumnya. Adapun diagramnya bisa dilihat pada gambar 3.11



Gambar 3.11 *Activity Diagram* Mencari Kriteria Nasabah

3.1.5.2.7. *Activity Diagram* Memprediksi Nasabah

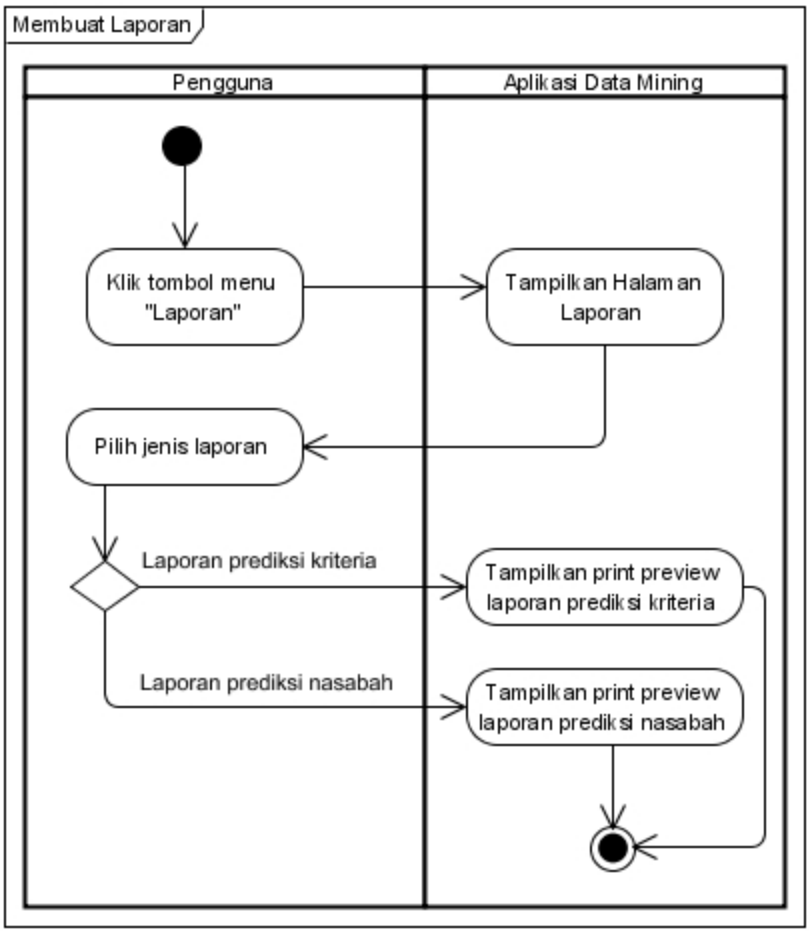
Pada tahap ini data nasabah yang telah diimpor sebelumnya akan diujicobakan dengan pola data atau *rule* yang telah terbentuk, sehingga bisa diketahui nasabah-nasabah mana saja yang diprediksikan berpotensi atau tidak. Adapun *activity digram* pada tahap ini bisa dilihat pada gambar 3.12



Gambar 3.12 *Activity Diagram* Memprediksi Nasabah

3.1.5.2.8. *Acivity Diagram* Membuat Laporan

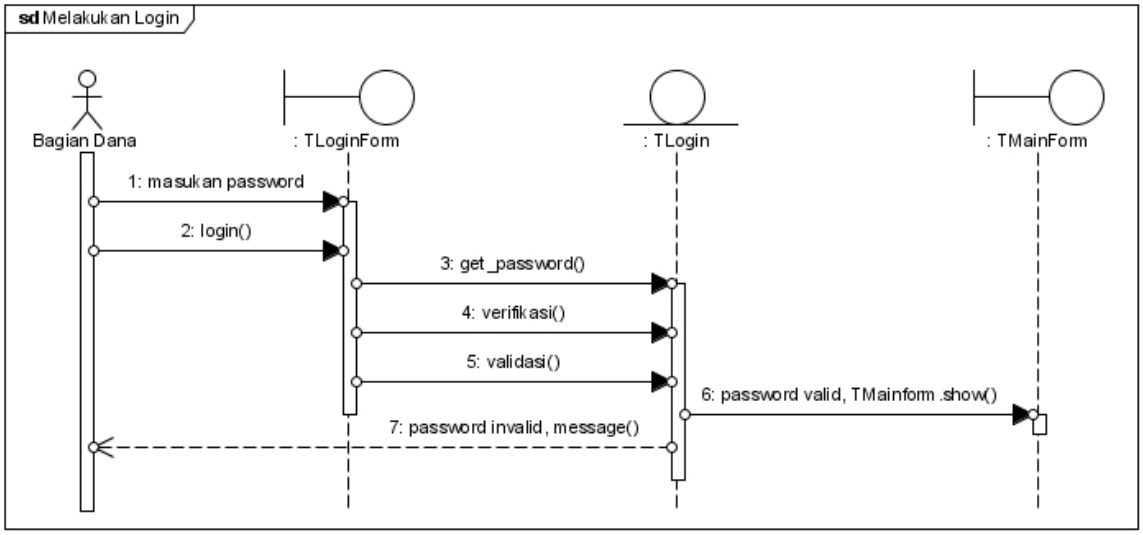
Diagram yang terdapat pada gambar 3.13 menggambarkan alur interaksi antara pengguna dengan aplikasi dalam membuat laporan hasil analisis prediksi yang telah diproses berupa pola data atau *trend* tentang prediksi kriteria nasabah. Pengguna bisa memilih dua jenis laporan dalam tahap ini yakni laporan yang berisi pola data prediksi kriteria nasabah dan laporan yang berisi prediksi nasabah itu sendiri.



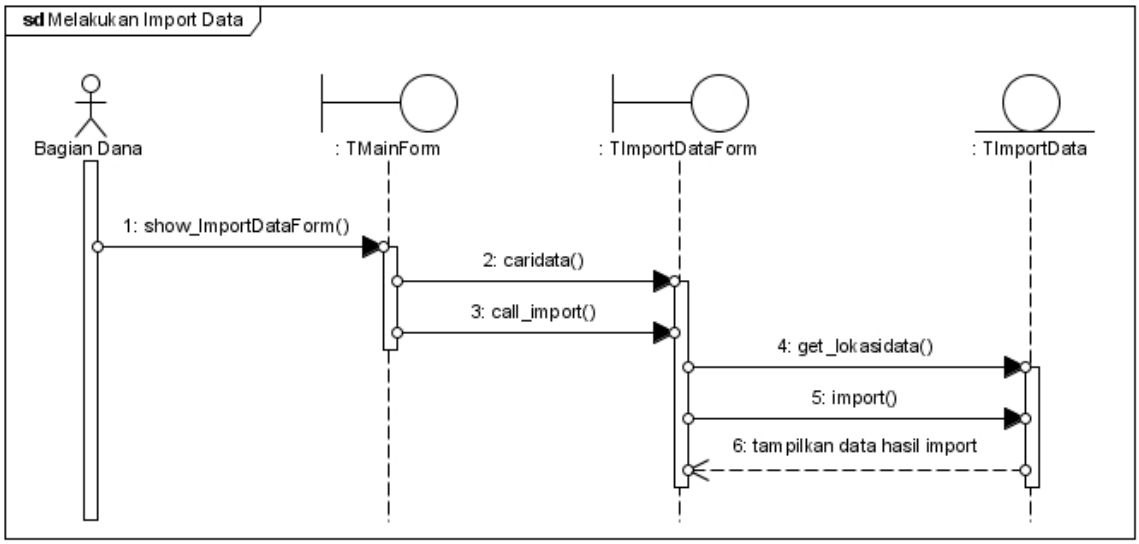
Gambar 3.13 *Activity Diagram* Membuat Laporan

3.1.5.3. *Sequence Diagram*

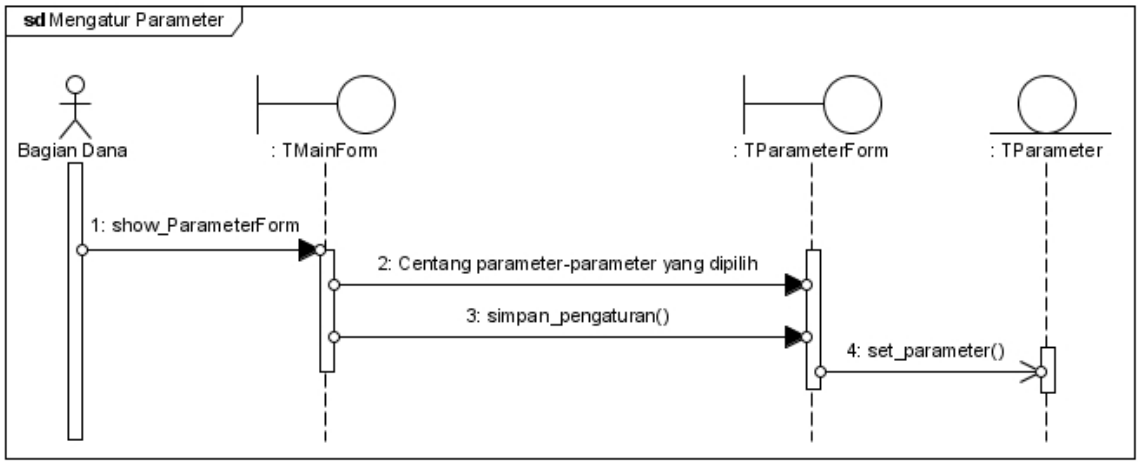
Dari skenario *use case* yang terdapat pada subbab 3.2.2.1.4 akan dibuat diagram *sequence* untuk kasus normal dari setiap skenario *use case.* Berikut diagram *sequence* aplikasi data mining yang digambarkan pada gambar 3.14 sampai dengan gambar 3.21



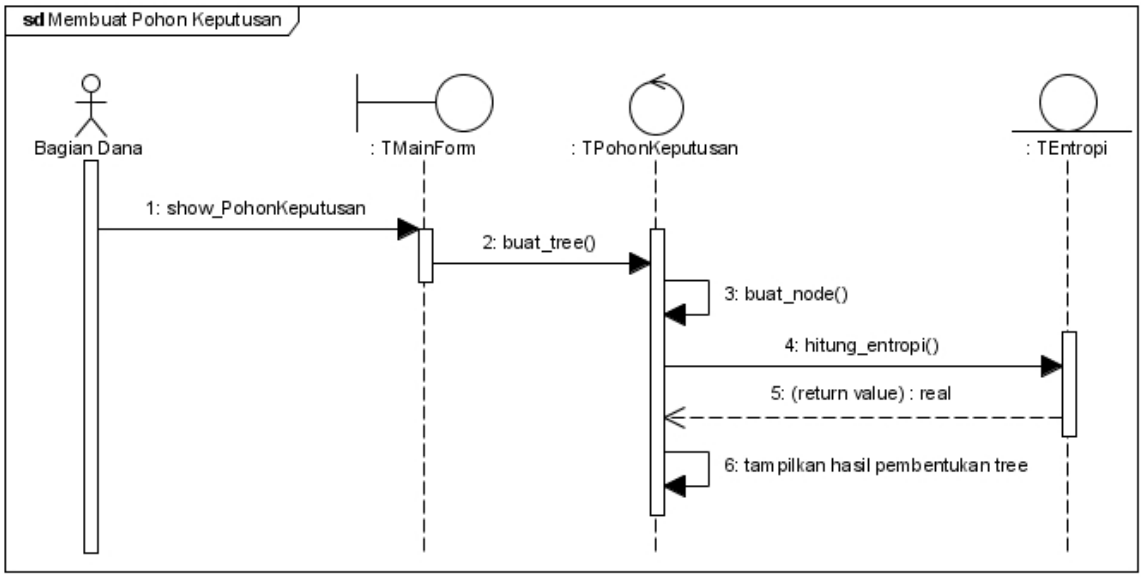
Gambar 3.14 *Sequence Diagram* Melakukan *Login*



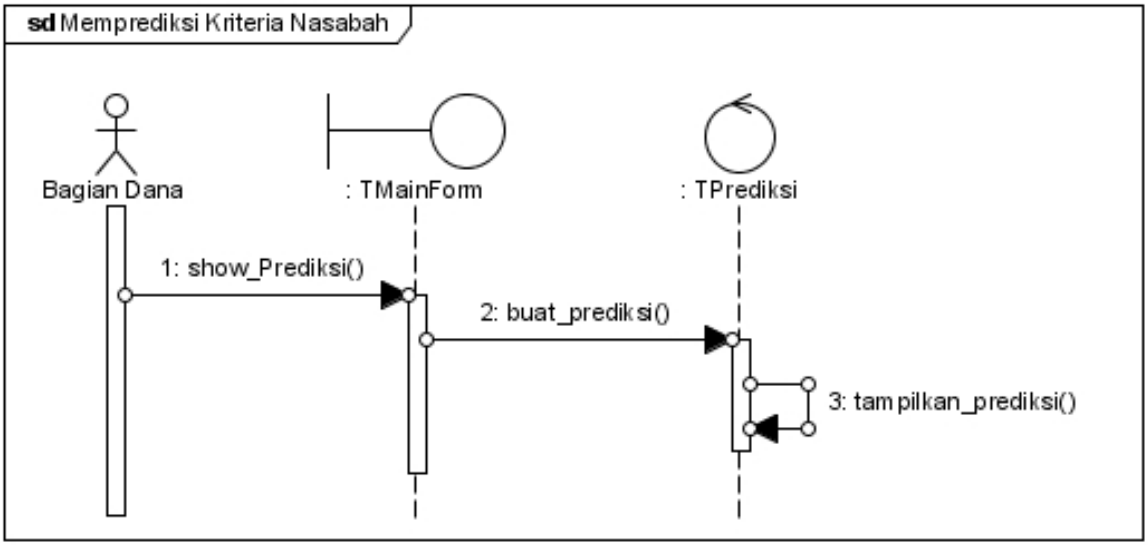
Gambar 3.15 *Sequence Diagram* Melakukan *Import* Data



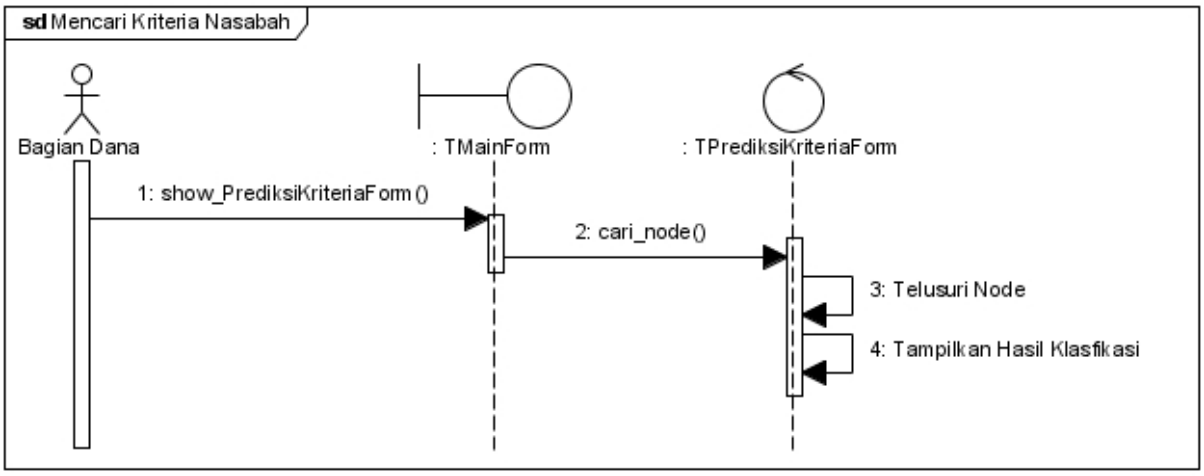
Gambar 3.16 *Sequence Diagram* Mengatur Parameter



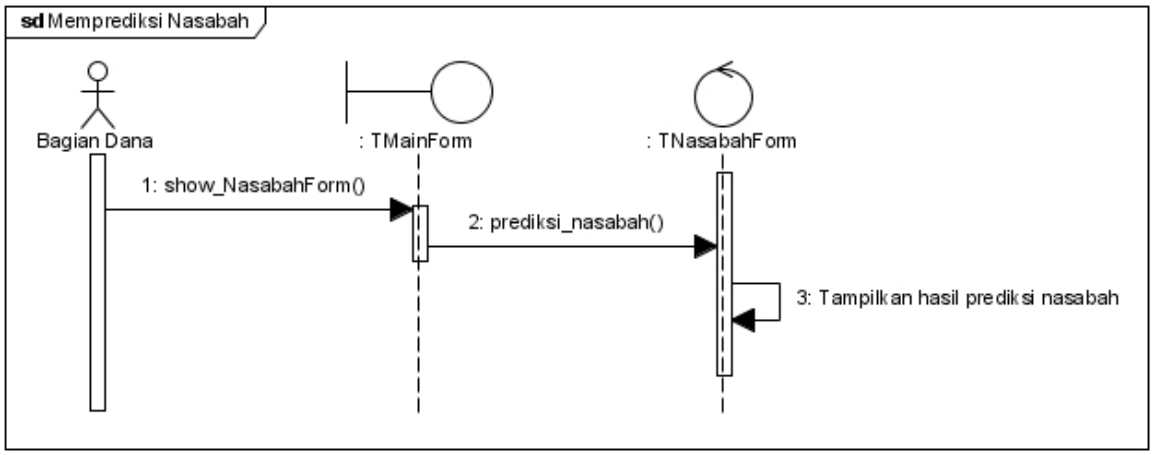
Gambar 3.17 *Sequence Diagram* Membuat Pohon Keputusan



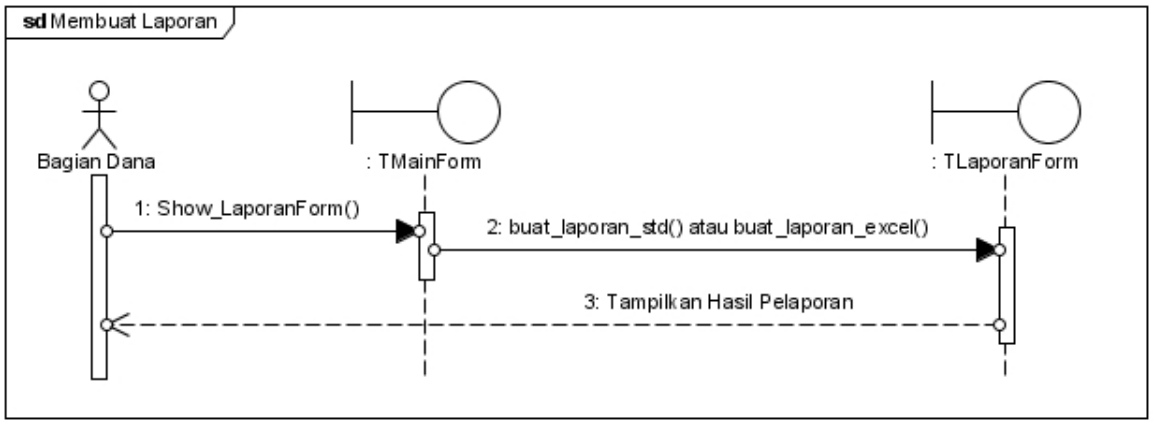
Gambar 3.18 *Sequence Diagram* Memprediksi Kriteria Nasabah



Gambar 3.19 *Sequence Diagram* Mencari Kriteria Nasabah



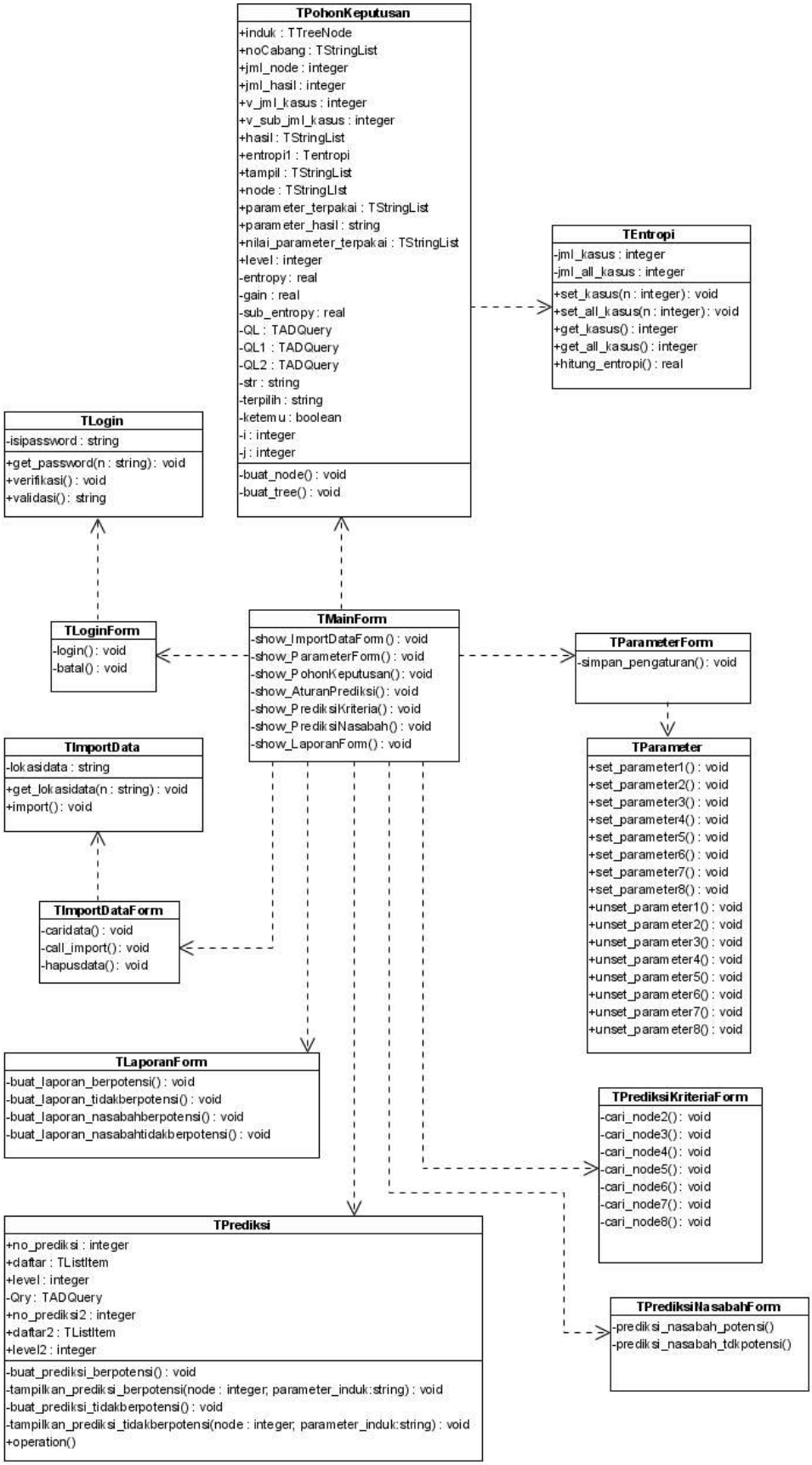
Gambar 3.20 *Sequence Diagram* Memprediksi Nasabah



Gambar 3.21 *Sequence Diagram* Membuat Laporan

3.1.5.4. Class Diagram

Class diagram merupakan realisasi dari *use case diagram* pada tahap analisis. Selanjutnya *class diagram* yang telah terdefinisikan nantinya akan dipergunakan pada tahap implementasi. *Class Diagram* aplikasi data mining yang akan dibangun dapat dilihat pada gambar 3.22



Gambar 3.22 *Class Diagram* Aplikasi Data Mining

3.1.5.4.1. Identifikasi Kelas

Identifikasi kelas bertujuan untuk menemukan jenis kelas berdasarkan *use case* dan skenario yang telah disusun pada subbab 3.2.2.1.1 dan 3.2.2.1.4.

Kelas-kelas yang terindentifikasi pada tahap analisis dan perancangan aplikasi data mining ini dapat dilihat pada tabel 3.19

Tabel 3.19 Identifikasi Kelas Analisis

No	Kelas	Jenis Kelas	Keterangan
1	TLoginForm	Boundary	Form login aplikasi
2	TLogin	Entity	Kelas yang digunakan untuk menyimpan data login pengguna
3	TMainForm	Boundary	Form utama aplikasi data mining yang berisi menu dan konfigurasi
4	TImportDataForm	Boundary	Form antarmuka untuk melakukan *import* data ke dalam aplikasi
5	TImportData	Entity	Kelas yang digunakan untuk menyimpan hasil *import* data
6	TParameterForm	Boundary	Form yang digunakan untuk menerima masukan atau pengaturan dari *user* berupa pemilihan parameter-parameter
7	TParameter	Entity	Kelas yang digunakan untuk menyimpan hasil pengaturan parameter oleh *user*
8	TPohonKeputusan	Control	Form yang digunakan untuk membentuk pohon keputusan.
9	TEntropi	Entity	Kelas yang digunakan oleh TPohonKeputusan dalam perhitungan nilai entropinya
10	TPrediksi	Control	Kelas yang digunakan untuk membuat pola data (*data pattern*) sebagai model prediksi dengan menelusuri hasil pohon keputusan dan menampilkannya ke dalam bentuk sebuah daftar aturan prediksi.

Bersambung,

Tabel 3.19 Identifikasi Kelas Analisis (Lanjutan)

No	Kelas	Jenis Kelas	Keterangan
11	TPrediksiKriteriaForm	Control	Kelas yang digunakan untuk memprediksi sebuah kriteria nasabah tertentu.
12	TPrediksiNasabahForm	Control	Kelas yang digunakan untuk memprediksi potensi nasabah.
13	TLaporanForm	Boundary	Form yang digunakan sebagai antarmuka dalam membuat laporan.

3.2 Perancangan Sistem

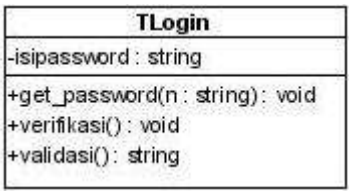
Perancangan sistem adalah tahap untuk memperbaiki, karena sangat penting dalam menentukan baik atau tidaknya hasil perancangan sistem yang diperoleh. Tahap perancangan sistem dapat digambarkan sebagai perancangan untuk membangun suatu sistem dan mengkonfirmasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang lebih baik. Tujuan dari perancangan sistem ini adalah memberikan gambaran secara umum kepada pemakai tentang sistem yang baru. Perancangan dapat didefinisikan sebagai komponen-komponen sistem informasi yang akan didesain secara rinci atas apa yang diusulkan.

3.2.1. Perancangan Data

Perancangan data bertujuan untuk mendeskripsikan data yang akan diimplementasikan berdasarkan kelas analisis yang bertipe *entity*.

3.2.1.1. Kelas TLogin

Kelas TLogin digunakan untuk melakukan autentikasi pengguna.



Gambar 3.23 Diagram Kelas TLogin

3.2.1.1.1. Deskripsi Atribut

Adapun deskripsi atribut dari kelas TLogin dapat dilihat pada tabel 3.19

Tabel 3.20 Deskripsi Atribut Kelas TLogin

Nama	Tipe	Panjang	Range	Nilai Tetap
isipassword	String	50	'a'..'Z','0'..'9',-,/,spasi	-

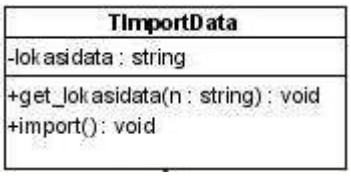
3.2.1.1.2. Deskripsi Layanan

Operasi-operasi yang ada dalam kelas ini adalah :

1. *Method* get_password(n:string):void, digunakan untuk melakukan pembacaan *password* yang telah diinputkan pengguna.
2. *Method* verifikasi():void, digunakan untuk mengecek data *password* yang diinputkan dengan data *password* di *database*.
3. *Method* validasi():string, digunakan untuk memvalidasi hasil verifikasi data *password* yang diinputkan pengguna.

3.2.1.2. Kelas TImportData

Kelas TImportData digunakan untuk meng-*impor* data ke dalam database aplikasi.



Gambar 3.24 Diagram Kelas TImportData

3.2.1.2.1. Deskripsi Atribut

Adapun deskripsi atribut dari kelas TLogin dapat dilihat pada tabel 3.20

Tabel 3.21 Deskripsi Atribut Kelas TImportData

Nama	Tipe	Panjang	Range	Nilai Tetap
lokasidata	String	50	’a’..’Z’,’0’..’9’,-,/,spasi	-

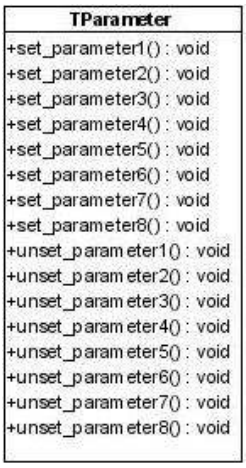
3.2.1.2.2. Deskripsi Layanan

Operasi-operasi yang ada dalam kelas ini adalah :

1. *Method* get_lokasi(n:string):void, digunakan untuk melakukan pembacaan lokasi yang telah diinputkan pengguna.
2. *Method* import():void, digunakan untuk memulai proses *import* data ke dalam *database* aplikasi.

3.2.1.3. Kelas TParameter

Kelas TParameter digunakan untuk menyimpan hasil pemilihan parameter oleh pengguna yang akan digunakan sebagai parameter pembentuk pohon keputusan.



Gambar 3.25 Diagram Kelas TParameter

3.2.1.3.1. Deskripsi Atribut

Tidak ada atribut pada kelas ini.

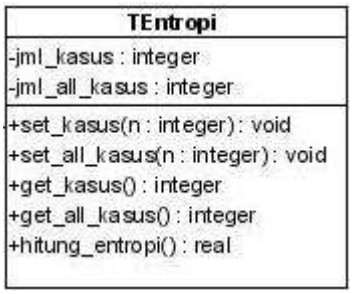
3.2.1.3.2. Deskripsi Layanan

Operasi-operasi yang terdapat dalam kelas ini bisa dikelompokan menjadi 2 buah operasi yakni :

1. *Method* set_parameter1():void, digunakan apabila parameter tertentu dipilih.
2. *Method* unset_parameter():void, digunakan apabila parameter tertentu tidak dipilih.

3.2.1.4. Kelas TEntropi

Kelas TEntropi digunakan untuk menghitung nilai entropi sebuah parameter dengan nilai-nilai parameternya.



Gambar 3.26 Diagram Kelas TEntropi

3.2.1.4.1. Deskripsi Atribut

Deskripsi atribut dari kelas TEntropi dapat dilihat pada tabel 3.21

Tabel 3.22 Deskripsi Atribut Kelas TEntropi

Nama	Tipe	Panjang	Range	Nilai Tetap
Jml_kasus	integer	15	0-9	-
Jml_all_kasus	integer	15	0-9	-

3.2.1.4.2. Deskripsi Layanan

Operasi-operasi yang ada dalam kelas ini adalah :

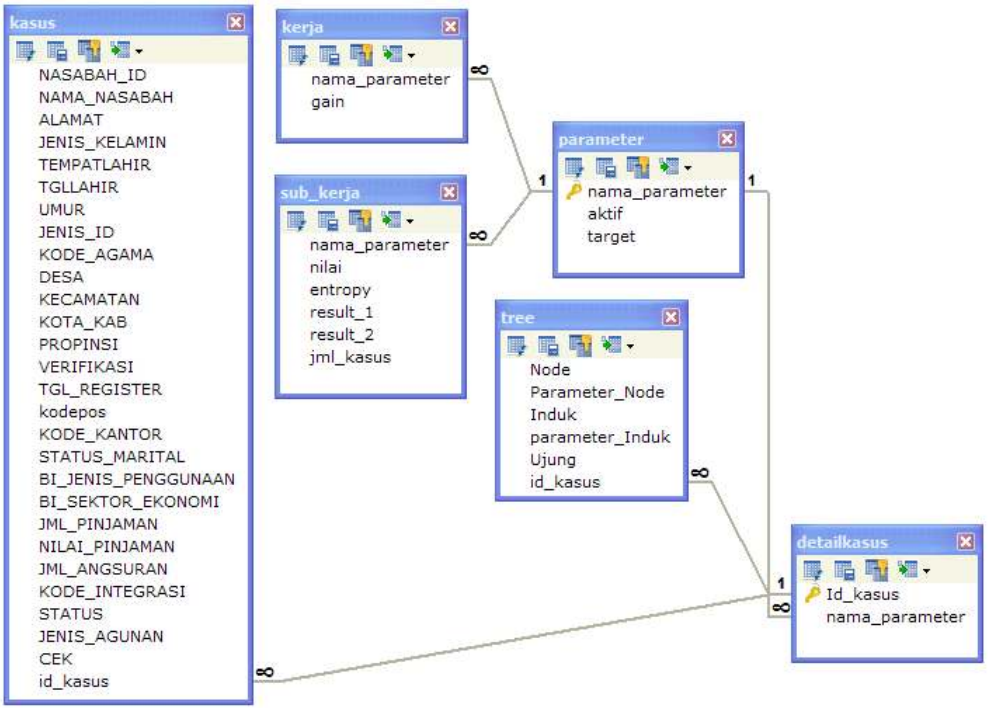
1. *Method* set_kasus(n:integer):void, digunakan untuk menghitung jumlah kasus yang dibagi berdasarkan parameter tertentu.
2. *Method* set_all_kasus(n:integer):void, digunakan untuk menghitung jumlah kasus secara keseluruhan.
3. *Method* get_kasus() :integer, digunakan untuk mengambil nilai jumlah kasus yang dibagi berdasarkan parameter tertentu berdasarkan hasil perhitungan yang telah dilakukan oleh *method* set_kasus(n:integer):void

4. *Method* get_all_kasus() :integer, digunakan untuk mengambil nilai jumlah kasus secara keseluruhan berdasarkan hasil perhitungan yang telah dilakukan oleh *method* set_all_kasus(n:integer):void

5. *Method* hitung_entropi():real, digunakan untuk melakukan perhitungan entropi.

3.2.1.5. Skema Relasi

Skema relasi merupakan hubungan dua tabel atau lebih pada sistem. Proses relasi antar *file* merupakan gabungan antar *file* yang memiliki *primary key* yang sama, sehingga *file-file* tersebut menjadi satu kesatuan yang dihubungkan oleh field atau atribut kunci. Gambar dibawah ini menjelaskan skema relasi yang akan dibangun pada aplikasi *data mining*.



Gambar 3.27 Skema Relasi Aplikasi *Data Mining*

3.2.1.6. Struktur File

Struktur file merupakan isi data yang berada dalam satu record pada satu tabel. Berikut struktur file yang terdapat dalam basis data aplikasi *data mining,* bisa dilihat pada tabel sampai dengan tabel

Tabel 3.23 Struktur file Parameter

No	***Field Name***	***Type***	***Length***	**Keterangan**
1	Nama_Parameter	*Varchar*	30	*Primary Key*
2	Gain	*Char*	1	-

Tabel 3.24 Struktur file Kerja

No	***Field Name***	***Type***	***Length***	**Keterangan**
1	Nama_Parameter	*Varchar*	30	*Foreign Key* (tabel parameter)
2	*Gain*	*Decimal*	(15,2)	-

Tabel 3.25 Struktur file Subkerja

No	***Field Name***	***Type***	***Length***	**Keterangan**
1	Nama_Parameter	*Varchar*	30	*Foreign Key* (tabel parameter)
2	Nilai	*Varchar*	255	-
3	*Entropy*	*Decimal*	(15,2)	-
4	Result_1	*Varchar*	30	-
5	Result_2	*Varchar*	30	-
6	Jml_kasus	*Integer*	11	-

Tabel 3.26 Struktur file Kasus

No	***Field Name***	***Type***	***Length***	**Keterangan**
1	NASABAH_ID	*Integer*	9	-
2	NAMA_NASABAH	*Varchar*	20	-
3	ALAMAT	*Varchar*	50	-
4	JENIS_KELAMIN	*Char*	1	-
5	TEMPATLAHIR	*Varchar*	9	-
6	TGLLAHIR	*Date*	-	-
7	UMUR	*Tinyint*	2	-

8	JENIS_ID	*Char*	3	-
9	KODE_AGAMA	*Char*	5	-
10	DESA	*Varchar*	20	-
11	KECAMATAN	*Varchar*	14	-
12	KOTA_KAB	*Char*	7	-
13	PROPINSI	*Char*	9	-
14	VERIFIKASI	*Tinyint*	1	-
15	TGL_REGISTER	*Date*	-	-
16	KODEPOS	*Smallint*	5	-
17	KODE_KANTOR	*Tinyint*	2	-
18	STATUS_MARITAL	*Char*	7	-
19	BI_JENIS_PENGGUNAAN	*Tinyint*	2	-
20	BI_SEKTOR_EKONOMI	*Smallint*	4	-
21	JML_PINJAMAN	*Mediumint*	7	-
22	NILAI_PINJAMAN	*Char*	6	-
23	JML_ANGSURAN	*Tinyint*	2	-
24	KODE_INTEGRASI	*Tinyint*	2	-
25	STATUS	*Tinyint*	1	-
26	JENIS_AGUNAN	*Tinyint*	1	-
27	CEK	*Varchar*	11	-
28	Id_kasus	*Char*	3	*Foreign Key* (tabel detailkasus)

Tabel 3.27 Struktur file *Tree*

No	***Field Name***	***Type***	***Length***	**Keterangan**
1	Node	*Integer*	11	-
2	Parameter_Node	*Varchar*	30	-
3	Induk	*Varchar*	30	-
4	Parameter_Induk	*Varchar*	30	-
5	Ujung	*Char*	1	-
6	Id_kasus	*Char*	3	*Foreign Key* (tabel detailkasus)

Tabel 3.28 Struktur file tabel detailkasus

No	***Field Name***	***Type***	***Length***	**Keterangan**
1	Id_kasus	*Char*	3	*Primary Key*
2	Nama_Parameter	*Varchar*	30	*Foreign Key* (tabel parameter)

Tabel Tambahan

Tabel 3.29 Struktur file *User*

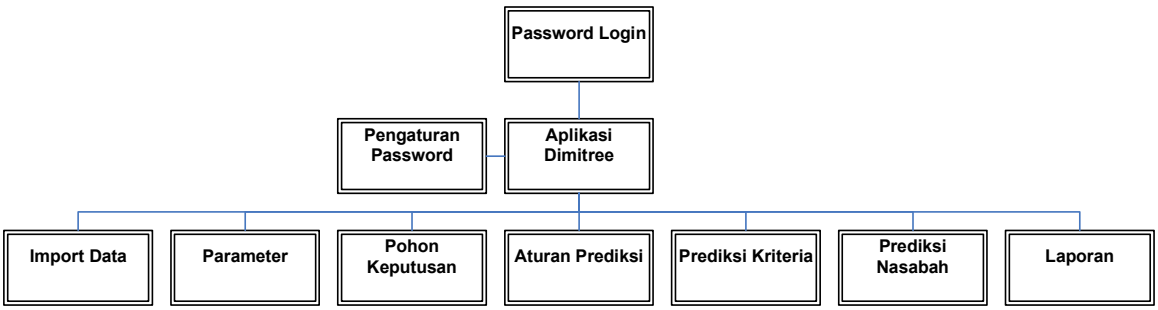
No	*Field Name*	*Type*	*Length*	Keterangan
1	Password	*Varchar*	255	-

3.2.2. Perancangan Arsitektur

Perancangan arsitektur merupakan perancangan yang dibuat sebelum program aplikasi dibuat. Dengan perancangan arsitektur akan mempermudah dalam proses pembangunan aplikasi *data mining* ini. Perancangan perangkat lunak ini mengikuti konsep yang telah dilakukan pada tahap analisis sistem. Perancangan arsitektur ini terdiri dari perancangan struktur menu, perancangan antar muka, dan jaringan semantik.

3.2.2.1. Perancangan Struktur Menu

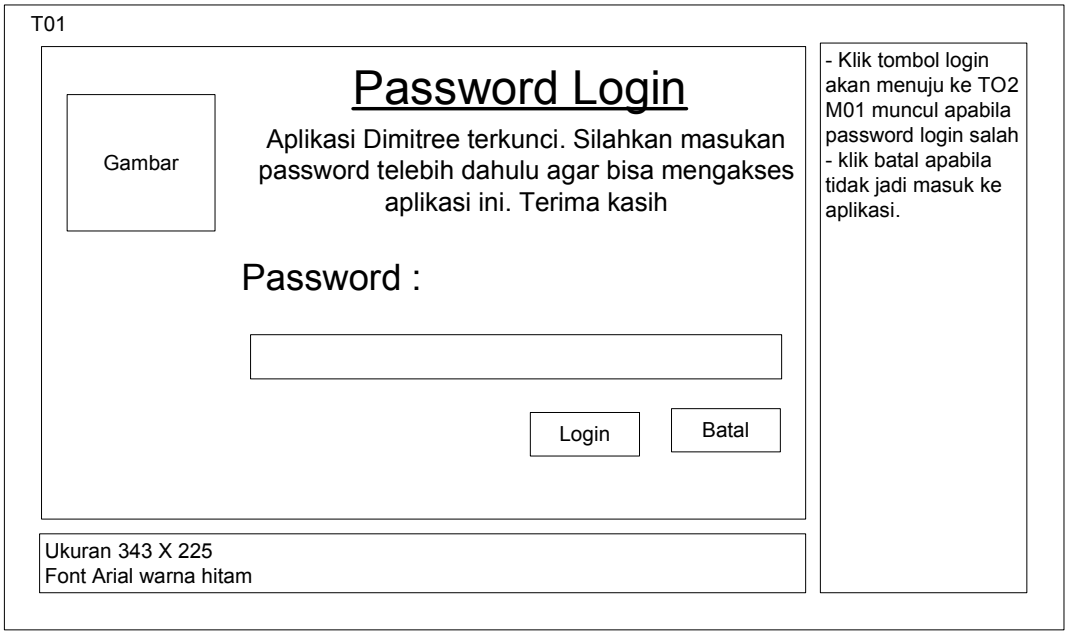
Dalam perancangan sebuah aplikasi dibutuhkan struktur menu yang berisikan menu yang berfungsi untuk memudahkan *user* dalam menggunakan aplikasi tersebut. Berikut ini digambarkan mengenai stuktur menu dalam aplikasi *data mining* yang akan dibangun,



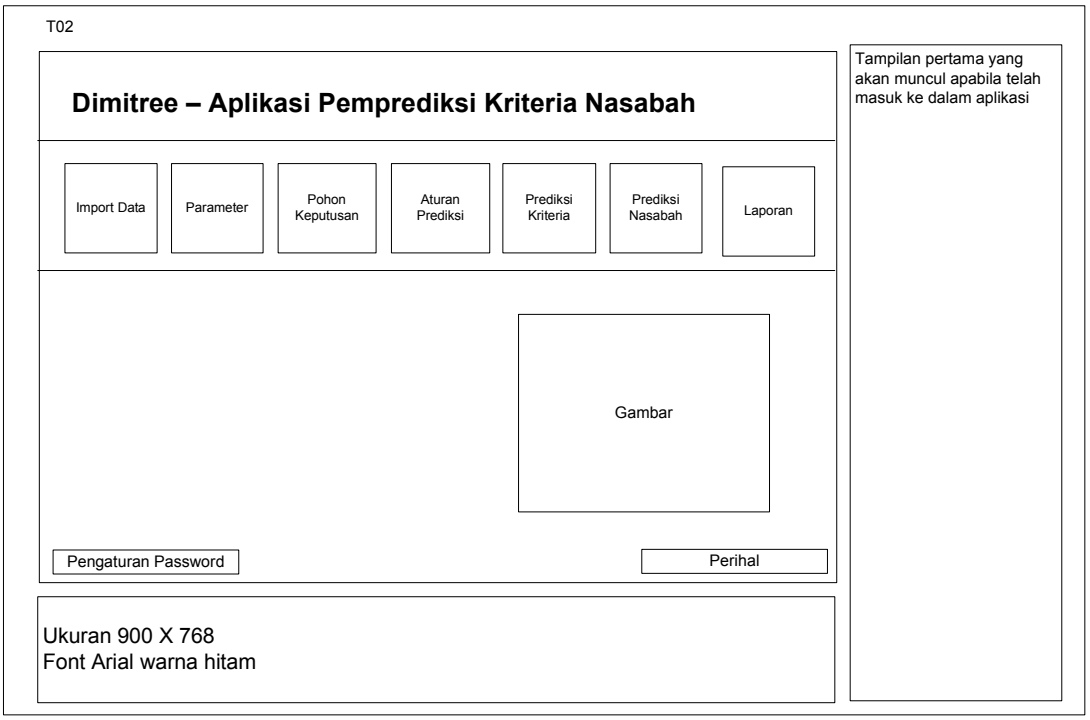
Gambar 3.28 Perancangan Struktur Menu Aplikasi *Data Mining*

3.2.2.2. Perancangan Antarmuka

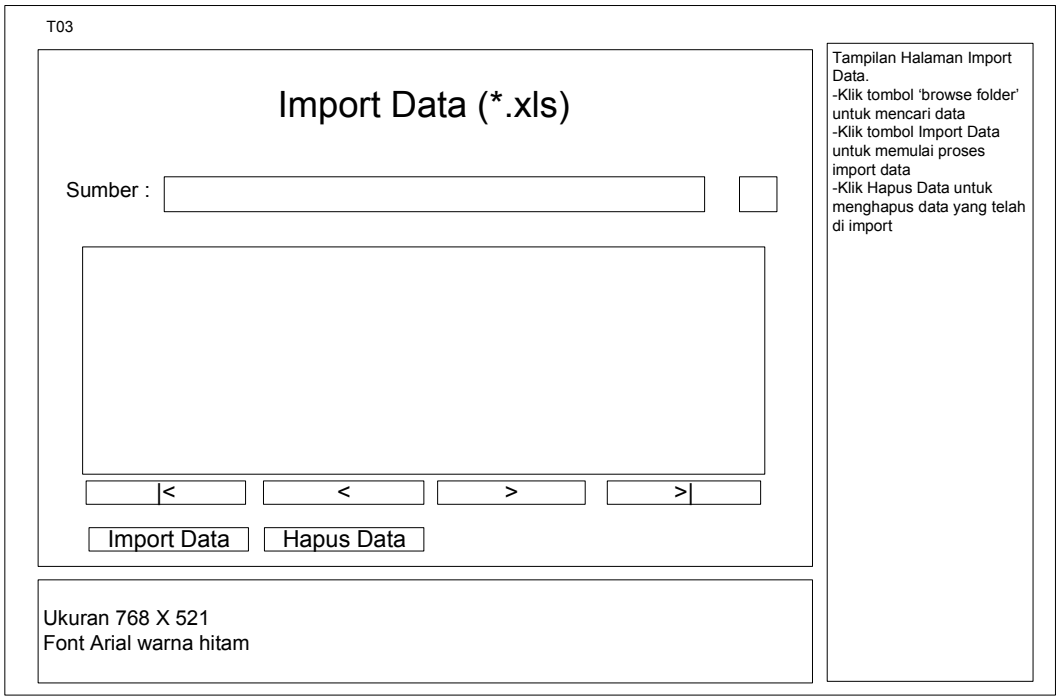
Dalam membuat aplikasi dibutuhkan tampilan yang sangat menarik, dimana dalam perancangan antar muka ini akan dituangkan beberapa imajinasi dalam merancang tampilan program dan tampilan pesan yang nantinya akan di pakai untuk membuat aplikasi sesungguhnya. Perancangan antar muka ini terdiri dari perancangan tampilan program dapat dilihat dari gambar 3.22 sampai dengan gambar 3.29, dan perancangan tampilan pesan dapat dilihat pada gambar 3.30



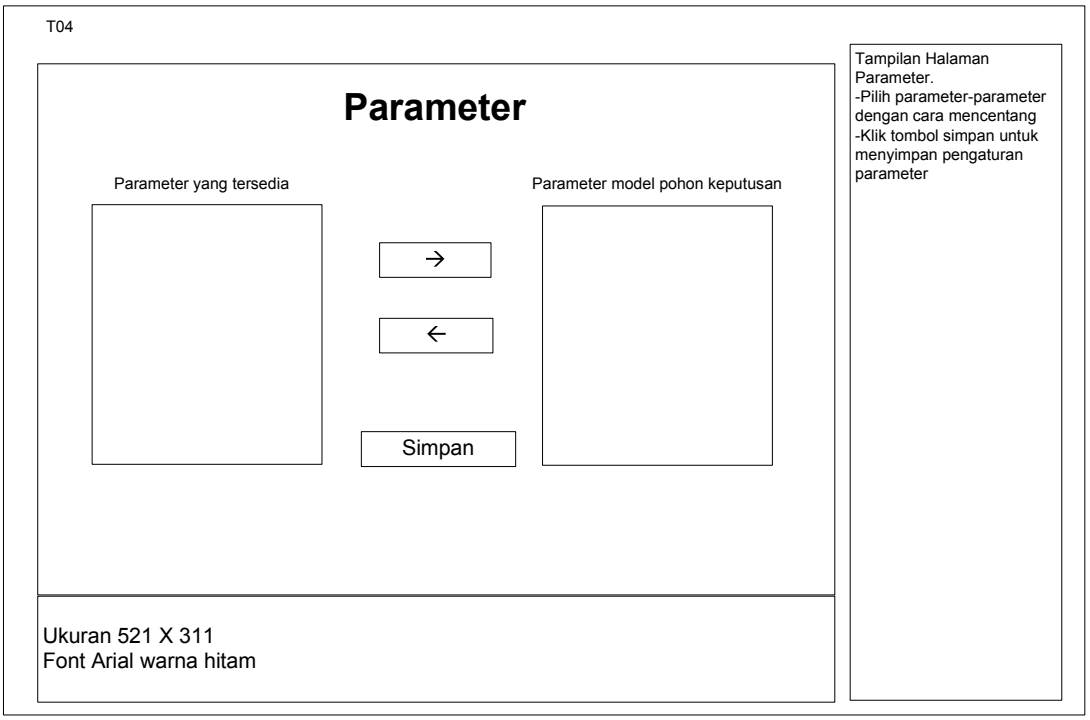
Gambar 3.29 Tampilan *Password Login*



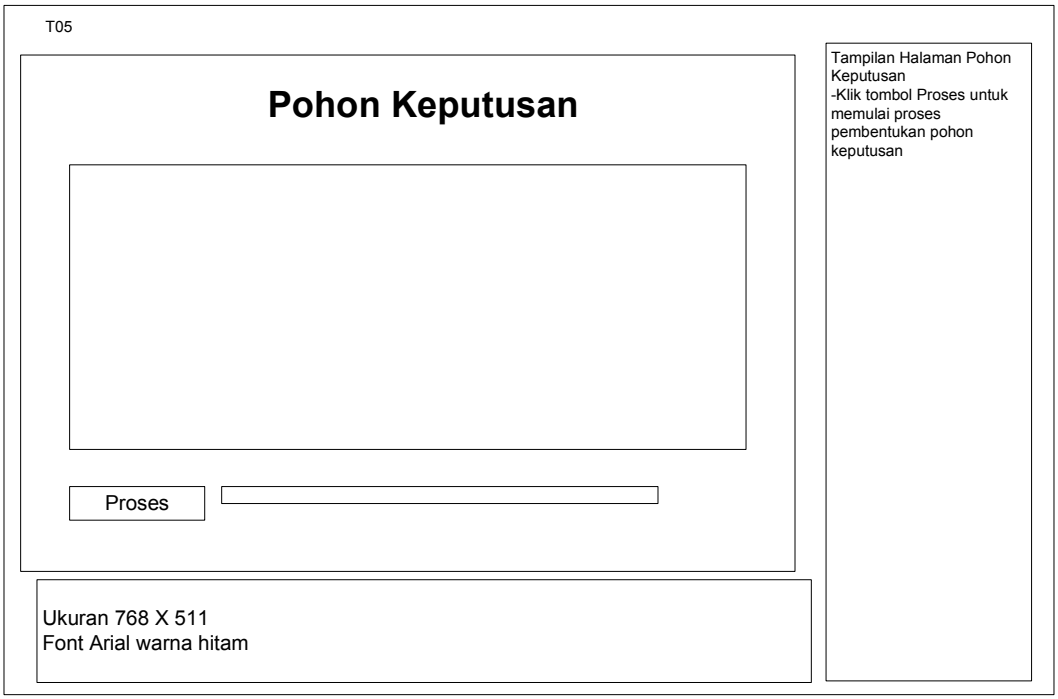
Gambar 3.30 Tampilan Halaman Utama Aplikasi *Data Mining*



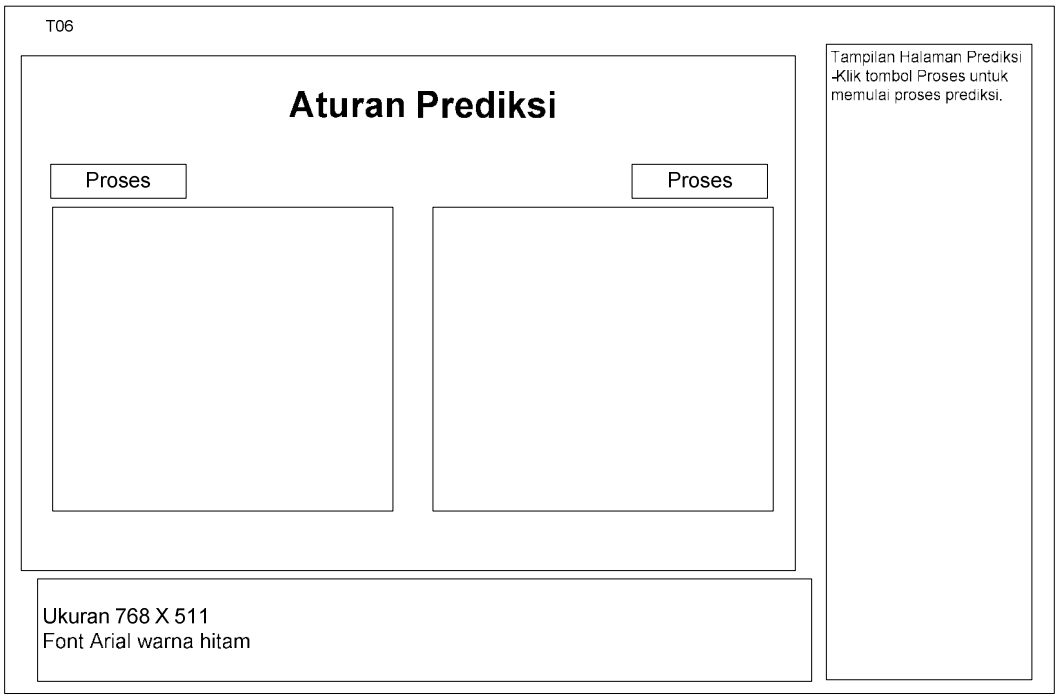
Gambar 3.31 Tampilan Halaman Import Data



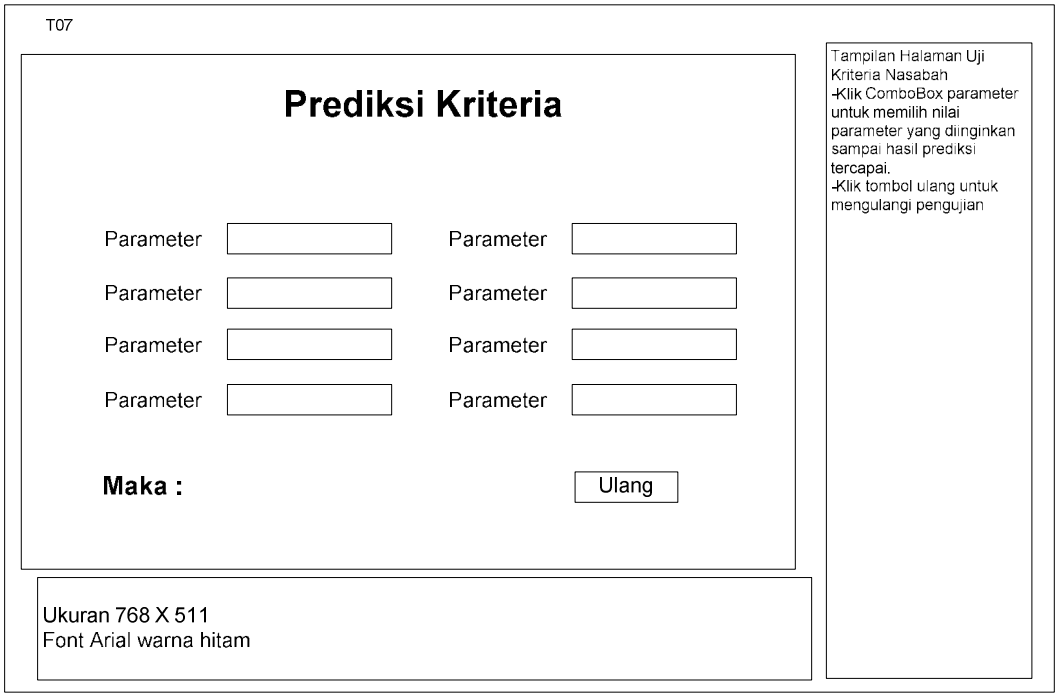
Gambar 3.32 Tampilan Halaman Parameter



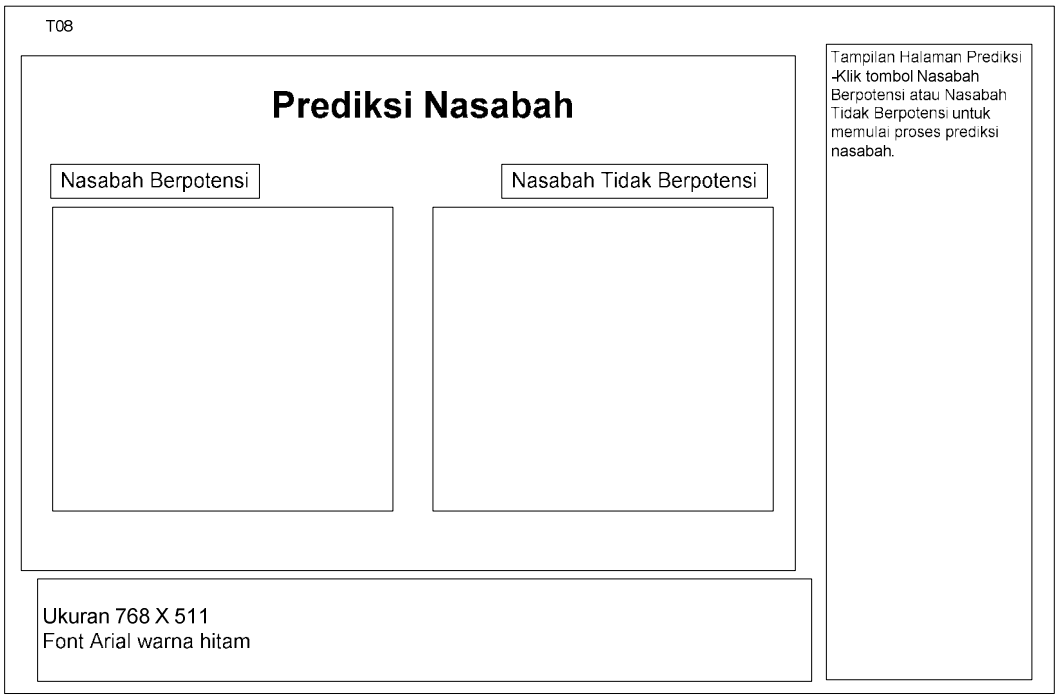
Gambar 3.33 Tampilan Halaman Pohon Keputusan



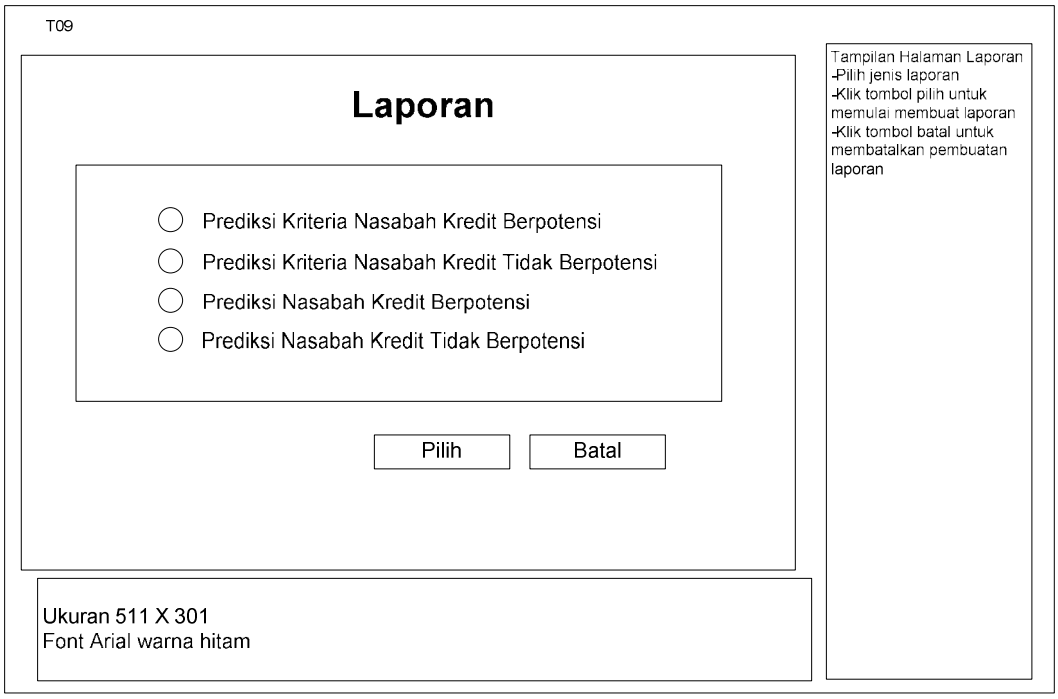
Gambar 3.34 Tampilan Halaman Aturan Prediksi



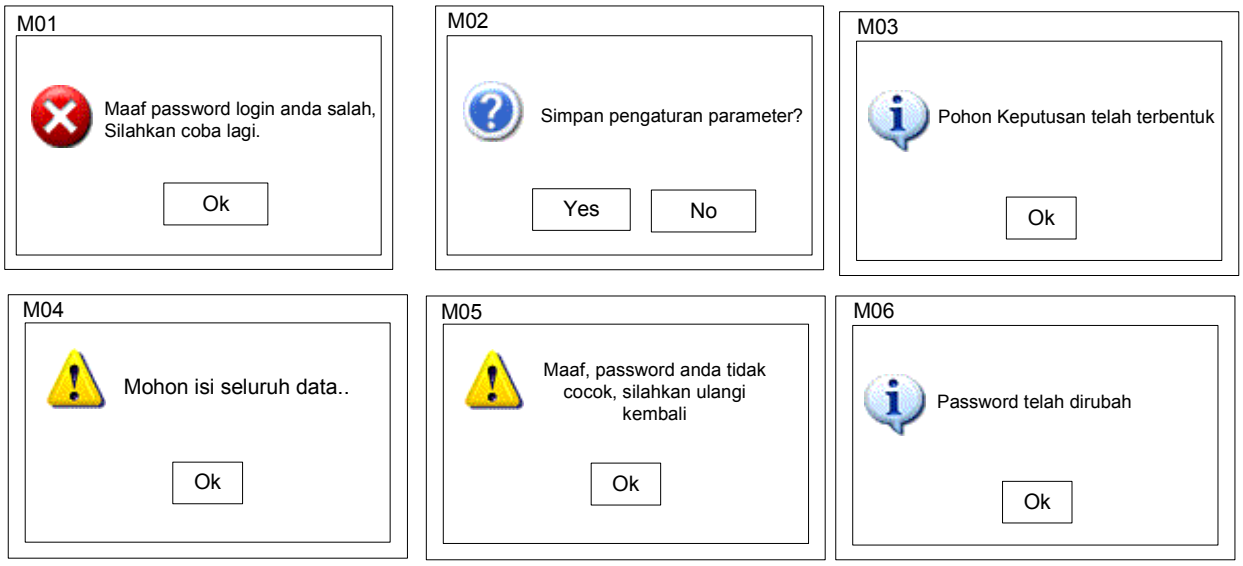
Gambar 3.35 Tampilan Halaman Prediksi (mencari) Kriteria



Gambar 3.36 Tampilan Halaman Prediksi Nasabah



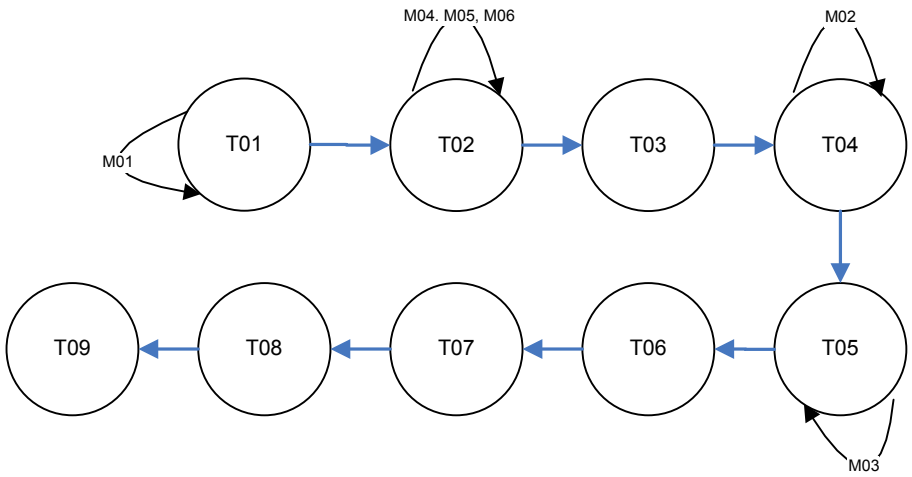
Gambar 3.37 Tampilan Halaman Laporan



Gambar 3.38 Tampilan Perancangan Pesan

3.2.2.3. Jaringan Semantik

Berikut adalah jaringan semantik dari aplikasi yang akan dibangun untuk menunjukan hubungan antar objek perancangan dapat dilihat pada gambar 3.31.

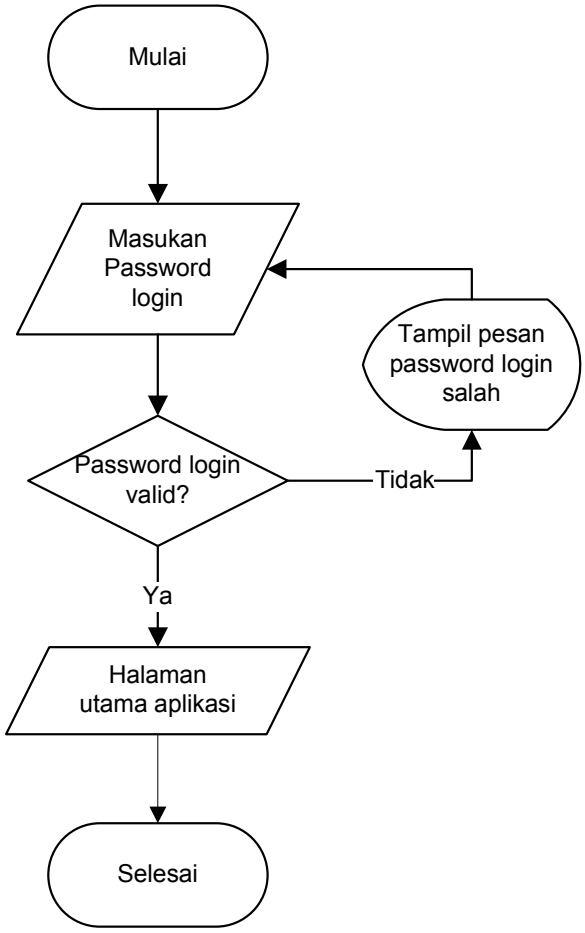


Gambar 3.39 Jaringan Semantik Aplikasi *Data Mining*

3.2.3 Perancangan Prosedural

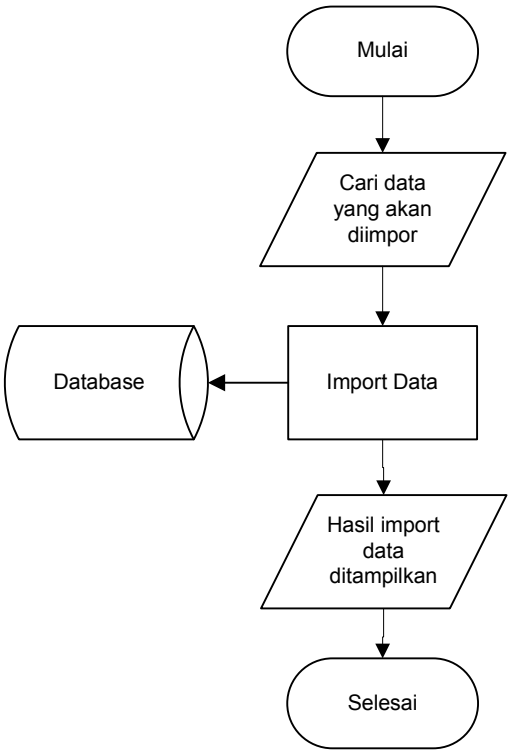
Perancangan prosedural menjelaskan tentang bagaimana program aplikasi berjalan. Perancangan prosedural ini digambarkan dalam bentuk *flow chart*. Adapun perancangan prosedural untuk aplikasi *data mining* yang akan di bangun sebagai berikut :

1. Prosedur *login*, merupakan prosedur yang terjadi ketika *user* akan mengakses aplikasi *data mining.*



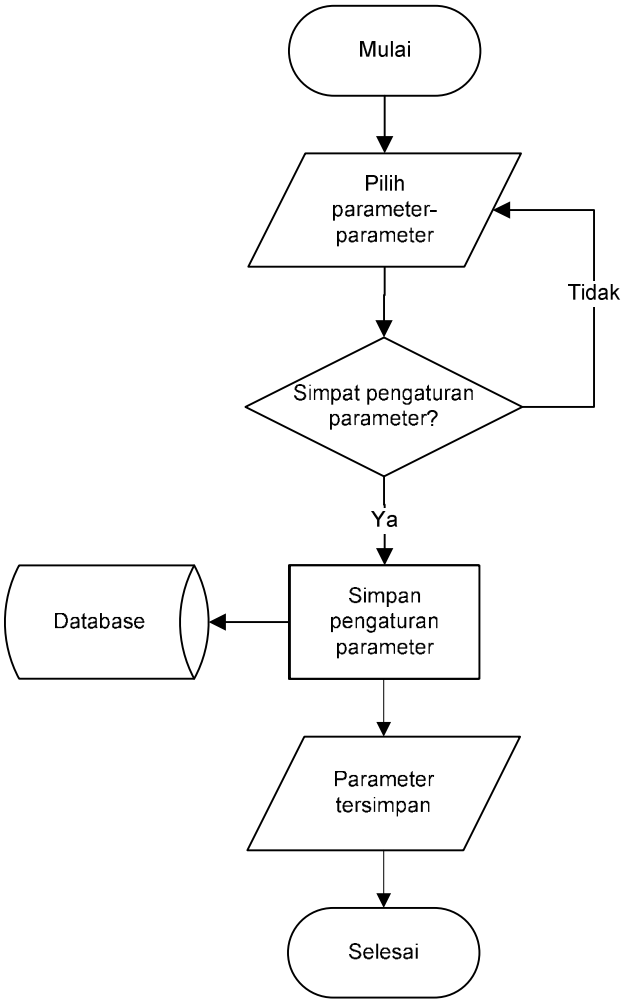
Gambar 3.40 Perancangan prosedural login

2. Prosedur *Import Data,* merupakan prosedur yang dilakukan ketika *user* akan mengimpor data ke dalam aplikasi



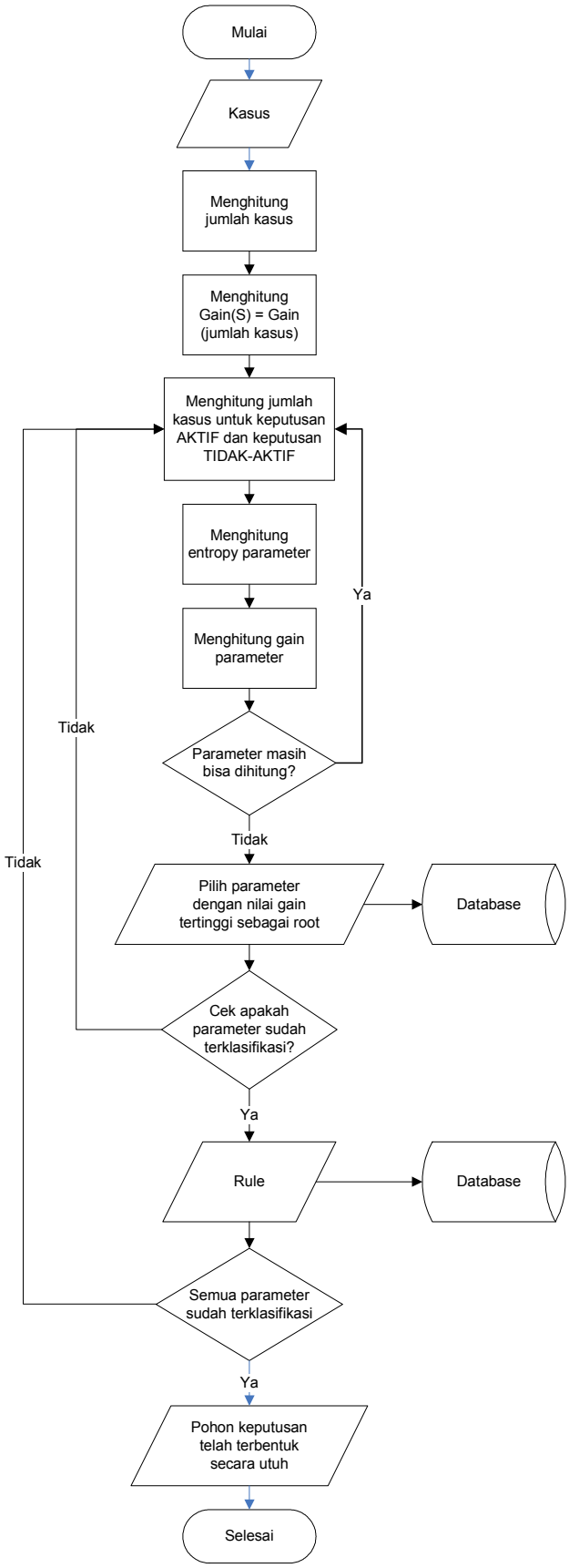
Gambar 3.41 Perancangan prosedural *Import Data*

3. Prosedur Parameter, merupakan prosedur yang dilakukan untuk mengatur parameter yang akan digunakan sebagai parameter pembentuk pohon keputusan.



Gambar 3.42 Perancangan prosedural parameter

4. Prosedur pohon keputusan (Algoritma C4.5) merupakan prosedur yang terjadi ketika user membuat pohon keputusan berdasarkan parameter-parameter yang telah dipilih sebelumnya oleh *user.*



Gambar 3.43 Perancangan prosedural pohon keputusan (Algoritma C4.5)