



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODE DAN PERANCANGAN SISTEM

3.1 Metode Penelitian

Metode-metode yang digunakan dalam penelitian ini terbagi menjadi delapan tahap, delapan tahapan tersebut adalah sebagai berikut.

1. Studi Literatur

Studi literatur yaitu proses pengumpulan bahan-bahan referensi baik dari buku, jurnal, artikel, paper, maupun situs internet. Studi literatur memiliki tujuan untuk membantu menemukan teori-teori yang berkaitan dengan topik yang dibahas dalam penelitian ini seperti *data mining*, pohon keputusan, dan Algoritma C4.5.

2. Teknik Pengumpulan Data

Teknik yang digunakan dalam pengumpulan data perancangan sistem ini adalah sebagai berikut : investigasi *hard data*, pengamatan, dan wawancara.

3. Analisis Proses KPR

Analisis proses yang sudah berjalan berdasarkan data yang terkumpul untuk mengetahui kebutuhan sistem dan menentukan fitur yang akan dimasukkan ke dalam sistem yang selanjutnya dilakukan perancangan sistem.

4. Perancangan Sistem

Merancang sistem yang akan digunakan dalam pemrograman sistem berdasarkan analisis proses. Perancangan sistem berupa pembuatan flowchart, DFD, dan desain UI untuk setiap form.

5. Pemrograman Sistem

Membangun sistem yang telah dirancang menggunakan Visual Studio 2010 menggunakan bahasa C#. Pemrograman sistem adalah proses merealisasikan fitur-fitur yang telah ditentukan dengan rancangan yang telah dibuat, sehingga menghasilkan sistem yang sesuai dengan keinginan user.

6. *Testing* dan *Debugging* Aplikasi

Testing dan *debugging* aplikasi dilakukan selama tahapan pemrograman sistem dan setelah sistem telah berhasil dibangun yang bertujuan mencari *bug*.

Testing dan *debugging* bertujuan mengurangi kesalahan-kesalahan yang ada pada program agar mendapatkan performa aplikasi yang maksimal.

7. Data *Learning*, *Testing*, dan Evaluasi

Data *learning* dan data *testing* merupakan tahapan yang harus dilalui dalam Algoritma C4.5 agar dapat memprediksi data.

8. Penulisan Laporan

Penulisan laporan berguna untuk membuat dokumentasi dari suatu penelitian serta memberikan informasi terhadap peneliti selanjutnya dalam penelitian sejenis.

3.2 Analisis Data

Pada bagian ini akan dijelaskan tentang semua atribut dan data yang digunakan dalam sistem. Data yang dapat diolah oleh *user* adalah data konsumen yang meliputi data *testing*, dan data *learning*. Perbedaan antara keduanya adalah data *learning* memiliki hasil sedangkan data *testing* belum memiliki hasil prediksi. Data *learning* digunakan untuk membuat pohon keputusan, dan pohon keputusan digunakan untuk memprediksi hasil dari data *testing*. Atribut yang

digunakan pada sistem memiliki dua tipe data yang berbeda yaitu kategorikal dan numerik. Atribut kategorikal memiliki pilihan yang dapat dipilih *user* sedangkan atribut numerik merupakan atribut angka. Tipe data kategorikal dan numerik digunakan karena teknik klasifikasi hanya dapat memroses tipe data tersebut.

5C merupakan prinsip yang digunakan dalam pemilihan kreditur bank. DCL (*Document Checklist*) yang diberikan oleh bank kepada pihak PT Graha Samolo Indah. DCL merupakan daftar dokumen yang harus dilengkapi oleh konsumen sebagai syarat untuk mendapatkan peminjaman KPR. Berikut ini adalah dokumen yang harus dilengkapi oleh konsumen atau DCL.

Tabel 3.1 Document Checklist

JENIS DOKUMEN	
A.	DOKUMEN IDENTITAS
1	Pas Photo Suami dan Istri, 3 (tiga) Lembar 3x4
2	Photocopy KTP Suami dan Istri, 2 (dua) lembar
3	Photocopy Surat Nikah, 2 (dua) Lembar
4	Photocopy Kartu Keluarga (KK) 2 Lembar
5	Photokopi Kartu NPWP
6	Form BTN dan Form Lain yang Dipersyaratkan (di isi)
7	Form Wawancara
8	PPH 21 (Bagi yang berpenghasilan tidak di transfer)
B.	DOKUMEN KARYAWAN SWASTA DAN PEGAWAI NEGERI
1	Slip Gaji Terbaru 3 bulan terakhir (asli/stample asli)
2	Photocopy Rek Tabungan (Suami + Istri)
3	Rekening Koran 3 bulan terakhir (pemohon)
4	Photocopy Kartu Pegawai/Karyawan
5	Photocopy SK Awal dan SK Akhir (Suami + Istri)
6	Keterangan Masih Bekerja / Mengajar (Suami + Istri)
7	Surat Keterangan Belum Memiliki Rumah (Suami + Istri)
8	Photocopy Rek. Sertifikasi (Suami + Istri)
9	Photocopy Sertifikat Pendidik (Suami + Istri)
10	Surat Pernyataan Penghasilan (pemohon)
11	Surat Kuasa pemotongan Gaji
C.	DOKUMEN WIRASWASTA
1	Laporan Keuangan 6 Bulan Terakhir
2	Bukti Pembelian (Struk, Nota, Bon dll)

Tabel 3.1 Document Checklist (Lanjutan)

3	Photocopy Rek Tabungan Suami + Istri (Cover+Isi)
4	Rekening Koran 3 Bulan Terakhir
5	Photocopy Surat Keterangan Usaha (Pemohon)
6	Photocopy SIUP-Izin Usaha (Pemohon)
JENIS DOKUMEN	
C.	DOKUMEN Wiraswasta
7	Foto Tempat Usaha dan Denah Lokasi
8	Denah Lokasi Usaha
9	Surat Keterangan Belum Memiliki Rumah (Suami + Istri)
10	Surat Pernyataan Penghasilan (pemohon)

Tabel spesifikasi atribut merupakan daftar atribut yang digunakan sistem untuk memprediksi kemampuan konsumen berdasarkan variabel dalam prinsip 5C yang terdapat pada DCL.

Tabel 3.2 Spesifikasi Atribut

Atribut	Keterangan	Tipe Data
A1	Umur	Numerikal
A2	Status Pernikahan	Kategorikal
A3	Jumlah Tanggungan	Numerikal
A4	Pendapatan Perbulan	Numerikal
A5	Lama Bekerja	Numerikal
A6	Pekerjaan	Kategorikal
A7	Jenis Kelamin	Kategorikal
A8	Alamat KTP	Kategorikal
A9	Alamat Kantor	Kategorikal
A10	Hasil Prediksi	Diterima/Ditolak

3.3 Rancangan Aplikasi

Perancangan aplikasi meliputi pembuatan *flowchart*, *data flow diagram*, *entity relationship diagram*, dan antarmuka setiap form. Model SDLC (*System Development Life Cycle*) yang digunakan dalam pengembangan sistem ini adalah model iteratif. Model iteratif mengkombinasikan proses-proses pada model air terjun dan iteratif (pengulangan) pada model prototipe (Sukanto dan Shalahudin,

2015). Model iteratif akan menghasilkan versi perangkat lunak yang sudah mengalami penambahan fitur pada setiap pertambahannya (inkremen). Model iteratif dipilih untuk mengakomodasi perubahan yang terjadi pada saat pengembangan sistem dengan menghasilkan produk/aplikasi pada setiap inkremen.

Aplikasi akan dikembangkan menggunakan bahasa pemrograman C#, dan Ms Visual Studio 2010 sebagai *development*, dan *runtime environment*. Bahasa pemrograman C# digunakan karena PT Graha Samolo Indah sudah menggunakan sistem informasi penjualan menggunakan bahasa pemrograman yang sama. Aplikasi akan diimplementasikan pada sistem informasi penjualan yang digunakan pada PT Graha Samolo Indah. Secara garis besar sistem informasi penjualan tersebut mempunyai fitur-fitur antara lain : mengelola konsumen, mengelola rumah, dan mengelola SPR. Aplikasi akan diimplementasikan pada fitur mengelola konsumen.

Fitur yang terdapat pada aplikasi sistem informasi KPR konsumen adalah sebagai berikut:

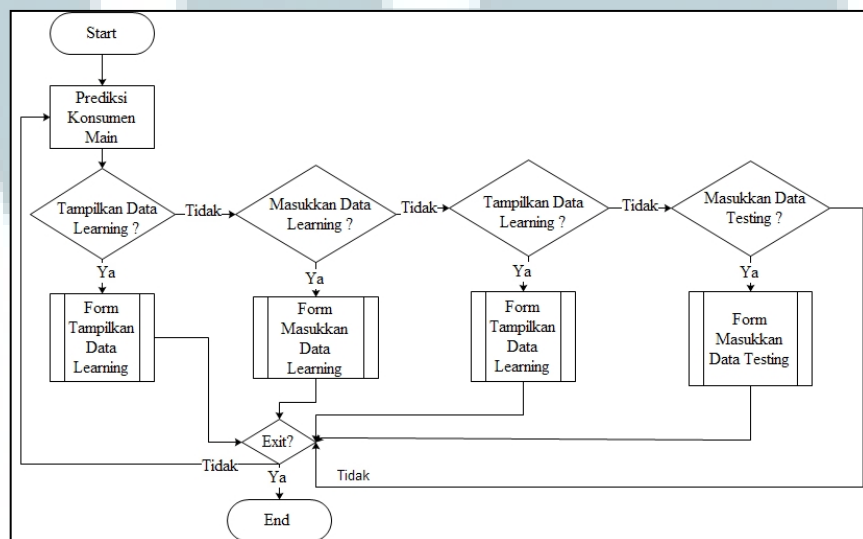
- Memasukkan data konsumen yang merupakan data *learning*, data *testing*.
- Merubah data konsumen yang merupakan data *learning*, data *testing*.
- Melihat data konsumen yang merupakan data *learning*, data *testing*, dan prediksi.

Aplikasi hanya dapat digunakan oleh *admin* atau *marketing support* perusahaan yang dapat menggunakan seluruh fitur yang terdapat dalam aplikasi. Aplikasi memiliki enam *form* yang terdiri dari *form* melihat data *learning*, *form* masukkan data *learning*, *form* merubah data *learning*, *form* melihat data *testing*,

form masukkan data *testing*, dan *form* merubah data *testing*. *Form* menghapus data konsumen tidak dibuat dikarenakan pada sistem informasi penjualan dimana aplikasi ini diimplementasikan sudah terdapat pilihan untuk menghapus konsumen. *Form* yang dibuat pada aplikasi disesuaikan dengan semua fitur yang terdapat pada aplikasi.

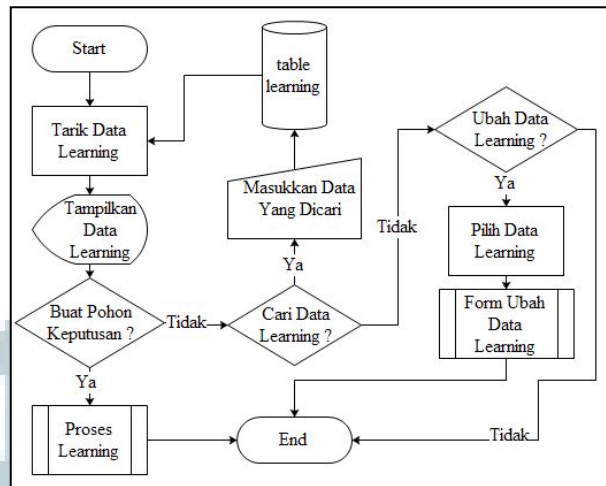
3.3.1 Flowchart

Flowchart digunakan untuk menggambarkan urutan proses yang terjadi secara detail dan hubungan suatu proses dengan proses yang lain dalam suatu sistem.



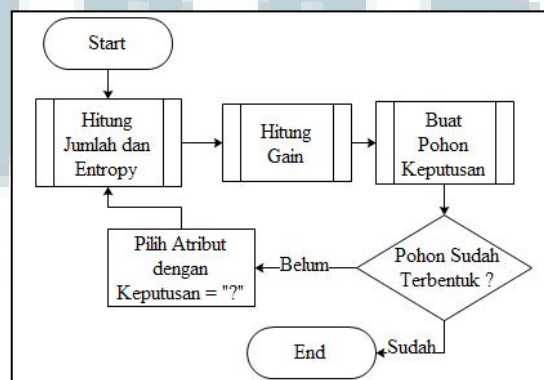
Gambar 3.1 Flowchart Aplikasi Sistem Informasi Prediksi Konsumen

Gambar 3.1 merupakan gambaran secara umum urutan proses yang terjadi pada saat aplikasi dijalankan untuk mendapatkan hasil prediksi. Pada saat aplikasi dijalankan *user* dapat memilih untuk tampilkan data *learning*, masukkan data *learning*, tampilkan data *testing*, atau masukkan data *testing*. Setelah *user* memilih untuk tampilkan data *learning*, tampilkan data *testing*, masukkan data *learning*, atau masukkan data *testing* *user* dapat memilih untuk keluar dari aplikasi atau tidak. Jika *user* memilih untuk keluar dari aplikasi maka *form* akan ditutup.



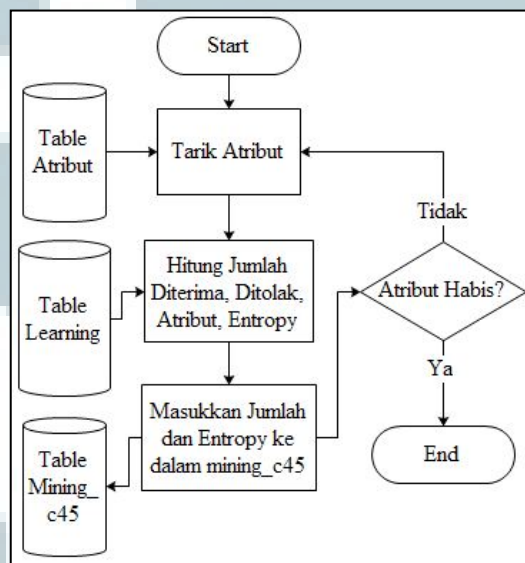
Gambar 3.2 Flowchart Subproses Tampilkan Data Learning

Subproses “Tampilkan Data Learning” dimulai dengan menarik data *learning* dari *database* yang kemudian ditampilkan pada *form* tampilkan data *learning*. *User* dapat memilih untuk membuat pohon keputusan, mencari data *learning*, atau mengubah data *learning*. Jika *user* memilih membuat pohon keputusan maka akan masuk ke subproses proses *learning*. Jika *user* ingin mencari data *learning* maka *user* harus memasukkan kata kunci dari data *learning* yang dicari kemudian aplikasi akan menarik data dari *database* yang selanjutnya akan ditampilkan. Jika *user* ingin mengubah data *learning* maka *user* harus memilih data *learning* yang ingin diubah dan selanjutnya akan masuk ke *form* ubah data *learning*.



Gambar 3.3 Flowchart Subproses Proses Learning

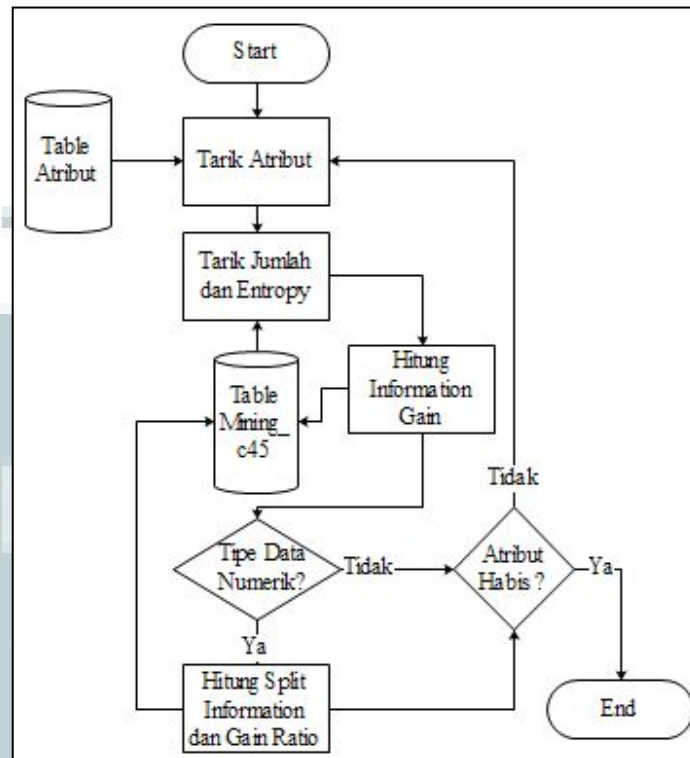
Gambar 3.3 merupakan subproses dari proses *learning* pada gambar 3.2. Proses *learning* dimulai dengan menghitung jumlah keseluruhan dan entropi untuk selanjutnya dihitung *gain* nya. Kemudian berdasarkan perhitungan *gain* akan dibuat pohon keputusan. Atribut dengan *gain* tertinggi akan menjadi cabang, dan disimpan ke dalam tabel aturan, setelah itu akan dicek apakah pohon keputusan sudah terbentuk. Jika belum maka akan dihitung kembali jumlah keseluruhan data berdasarkan nilai atribut yang menjadi cabang. Nilai atribut yang menjadi cabang adalah atribut yang memiliki keputusan “?” pada proses membuat pohon keputusan. Ketika pohon keputusan sudah terbentuk maka proses akan selesai.



Gambar 3.4 Flowchart Subproses Jumlah dan Entropy

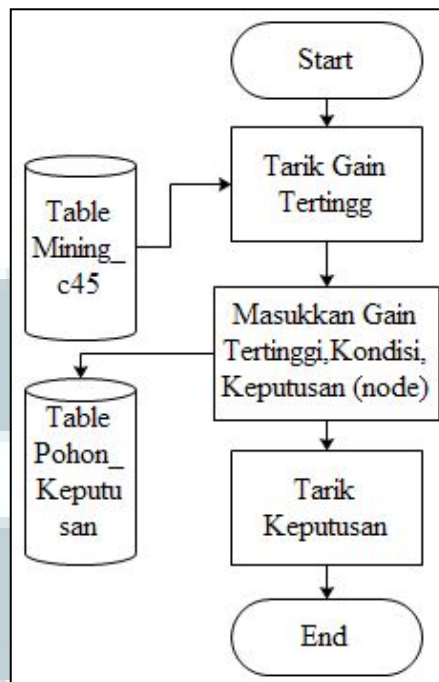
Subproses “Hitung Jumlah dan Entropy” dimulai dengan membaca semua atribut yang ada satu persatu yang ditarik dari *database*. Kemudian berdasarkan data *training* akan dihitung jumlah atribut, jumlah diterima, jumlah ditolak dan entropi. Setelah itu data akan disimpan ke dalam tabel. Setelah proses selesai akan

dicek apakah masih ada atribut yang harus dihitung. Jika masih ada maka proses akan mengulang sampai sudah tidak ada atribut yang harus dihitung.



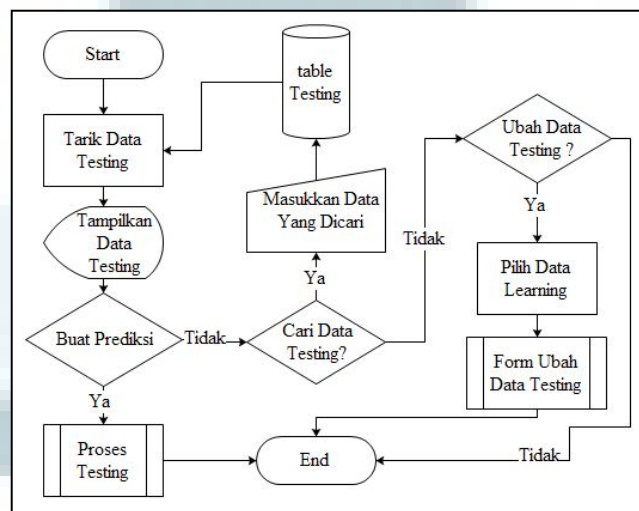
Gambar 3.5 Flowchart Subproses Hitung Gain

Gambar 3.5 menunjukkan aliran data pada subproses “Hitung Gain”. Aliran data dimulai dengan mengambil atribut kemudian mengambil jumlah data dan entropi dari tabel. Setelah itu akan dihitung *information gain* dan proses akan mengecek tipe data dari atribut. Jika tipe data numerik maka akan dihitung *split information* dan *gain ratio* nya. Lalu proses akan mengecek apakah atribut sudah habis. Jika belum maka proses akan terus mengulang sampai atribut habis.



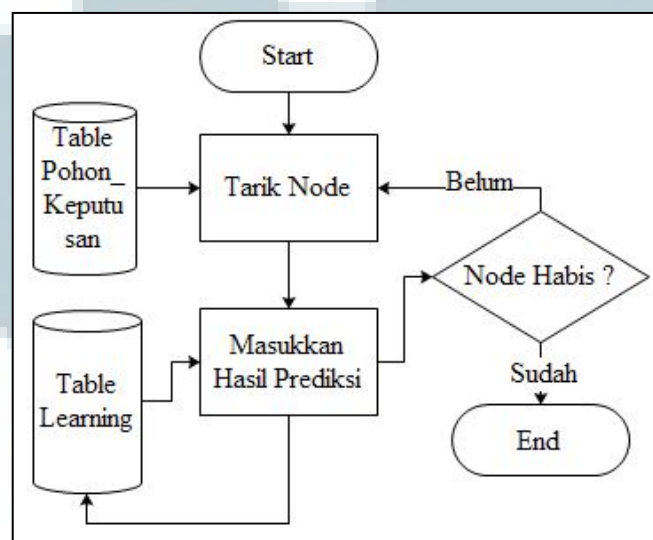
Gambar 3.6 Flowchart Subproses Buat Pohon Keputusan

Subproses “Hitung Buat Pohon Keputusan” dimulai dengan mengambil *gain* paling tinggi dari tabel. Setelah mendapatkan *gain* yang paling tinggi maka atribut tersebut akan dimasukkan ke dalam tabel untuk menjadi *node* (cabang/akar). Lalu proses akan menarik keputusan dan proses akan selesai.



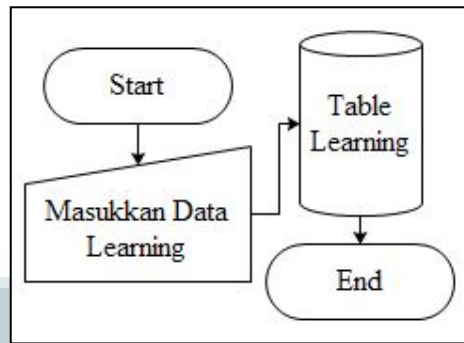
Gambar 3.7 Flowchart Tampilkan Data Testing

Subproses “Tampilkan Data Testing” dimulai dengan menarik data *testing* dari *database* yang kemudian ditampilkan pada *form* tampilan data *testing*. *User* dapat memilih untuk membuat prediksi, mencari data *testing*, atau mengubah data *testing*. Jika *user* memilih membuat prediksi maka akan masuk ke subproses proses *testing*. Jika *user* ingin mencari data *testing* maka *user* harus memasukkan kata kunci dari data *testing* yang dicari kemudian aplikasi akan menarik data dari *database* yang selanjutnya akan ditampilkan. Jika *user* ingin mengubah data *testing* maka *user* harus memilih data *testing* yang ingin diubah dan selanjutnya akan masuk ke *form* ubah data *testing*.



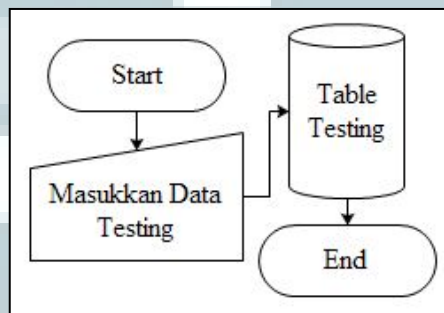
Gambar 3.8 Flowchart Subproses Proses Testing

Gambar 3.8 merupakan subproses dari proses *testing* pada Gambar 3.7. Proses *testing* dimulai dengan membaca *node* (cabang/akar) dari tabel. Setelah itu hasil prediksi akan diubah berdasarkan keputusan. Proses akan diulang sampai semua *node* pada pohon sudah digunakan.



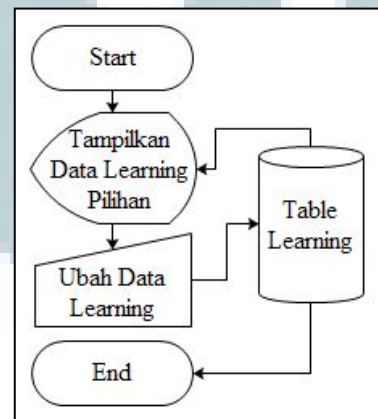
Gambar 3.9 Flowchart Subproses Masukkan Data Learning

Subproses “Masukkan Data Learning” dimulai dengan *user* memasukkan data learning yang selanjutnya akan disimpan pada tabel *learning*. Setelah data sudah dimasukkan kedalam tabel *learning* maka proses akan selesai.



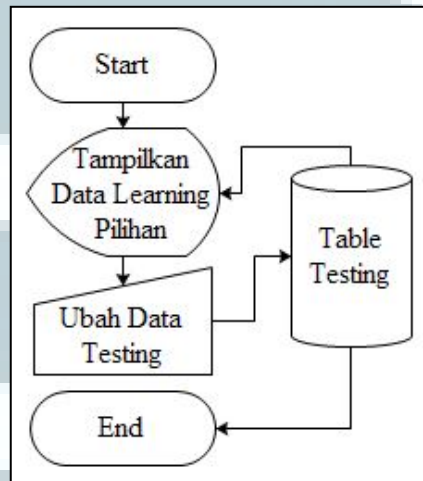
Gambar 3.10 List Flowchart Subproses Masukkan Data Testing

Subproses “Masukkan Data Testing” dimulai dengan *user* memasukkan data testing yang selanjutnya akan disimpan pada tabel *testing*. Setelah data sudah dimasukkan kedalam tabel *testing* maka proses akan selesai.



Gambar 3.11 List Flowchart Subproses Ubah Data Learning

Subproses “Ubah Data Learning” dimulai dengan menarik data dari tabel *learning* untuk ditampilkan kepada *user*. Data yang telah ditampilkan lalu dapat diubah oleh *user*. Data yang telah diubah lalu disimpan ke tabel *learning* dan proses akan selesai.

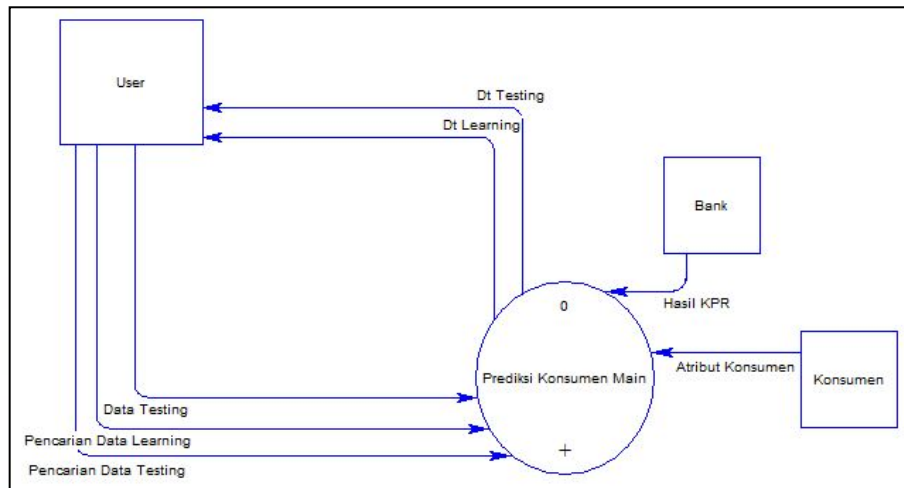


Gambar 3.12 List Flowchart Subproses Ubah Data Testing

Subproses “Ubah Data Testing” dimulai dengan menarik data dari tabel *testing* untuk ditampilkan kepada *user*. Data yang telah ditampilkan lalu dapat diubah oleh *user*. Data yang telah diubah lalu disimpan ke tabel *testing* dan proses akan selesai.

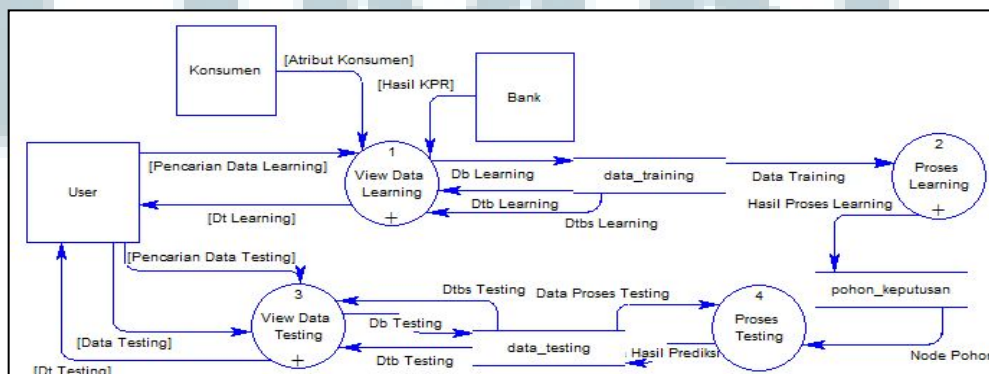
3.3.2 Data Flow Diagram

Data Flow Diagram menurut Rosa dan Salahudin adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (input) dan keluaran (output) (Sukanto dan Shalahudin, 2015). Aplikasi yang akan dirancang memiliki diagram *context* dan 2 level abstraksi yaitu DFD *level 0*, dan *level 1*.



Gambar 3.13 Context Diagram Prediksi Konsumen

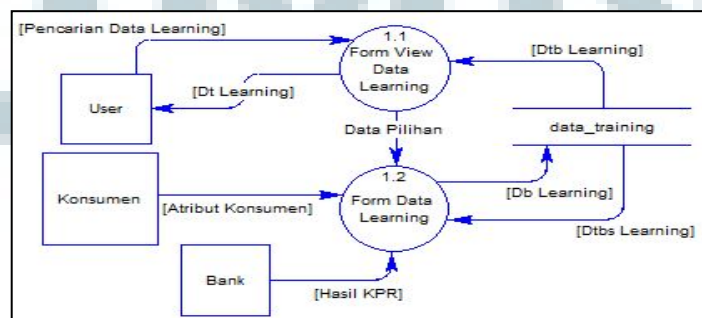
Pada Gambar 3.13 yaitu *context diagram* terdapat tiga entitas, satu proses, dan delapan aliran data. Entitas yang terdapat pada diagram konteks bernama *User* yang merupakan *admin* atau *marketing support* perusahaan, *Bank* yang merupakan pemberi kredit, *Konsumen* yang memberikan atribut. Proses *Prediksi Konsumen Main* memiliki turunan dan menerima tiga masukan dari *User*, satu masukan dari *Bank*, satu masukan dari *Konsumen*, serta memberikan dua keluaran kepada *User* yang direpresentasikan dengan aliran data. Masukan pada proses *Prediksi Konsumen Main* yaitu *Atribut Konsumen*, *Hasil KPR*, *Data Testing*, *Pencarian Data Learning*, dan *Pencarian Data Testing*. Keluaran pada proses *Prediksi Konsumen Main* yaitu *Dt Testing*, *Dt Learning*, dan *Hasil Prediksi*.



Gambar 3.14 DFD Level 0 Prediksi Konsumen Main

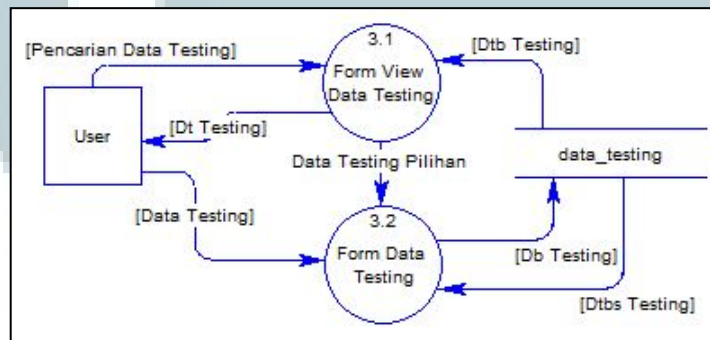
Gambar 3.14 yaitu DFD *level 0* menjelaskan turunan dari Prediksi Konsumen *Main* yang memiliki tiga entitas, empat proses, tiga tabel, delapan belas aliran data. Empat proses ini meliputi *View Data Testing*, *Proses Testing*, *View Data Testing*, dan *Proses Testing*. Tabel yang terdapat pada *level 0* ini adalah *data_testing*, *data_testing*, dan *pohon_keputusan*. Proses *View Data Testing* menerima aliran data *Pencarian Data Testing* dari entitas *User*, aliran data *Atribut Konsumen* dari entitas *Konsumen*, aliran data *Hasil KPR* dari entitas *Bank*, dan aliran data *Dtb Testing*, *Dtbs Testing* dari tabel *data_testing*. Proses *View Data Testing* memberikan aliran data *Dt Testing* ke entitas *User*, dan aliran data *Db Testing* ke tabel *data_testing*. Proses *Testing* menerima aliran data *Data Training* dari tabel *data_testing*, dan memberikan aliran data *Hasil Proses Testing* ke tabel *pohon_keputusan*.

Proses *Testing* menerima aliran data *Dt Aturan* dari tabel *pohon_keputusan*, aliran data *Data Proses Testing* dari tabel *data_testing*, dan memberikan aliran data *Hasil Proses Testing* ke tabel *data_testing*. Proses *View Data Testing* menerima aliran data *Pencarian Data Testing*, aliran data *Data Testing* dari entitas *User*, dan aliran data *Dtb Testing*, *Dtbs Testing* dari tabel *data_testing*. Proses *View Data Testing* memberikan aliran data *Dt Testing* ke entitas *User*, dan aliran data *Db Testing* ke tabel *data_testing*.



Gambar 3.15 DFD Level 1-1 View Data Learning

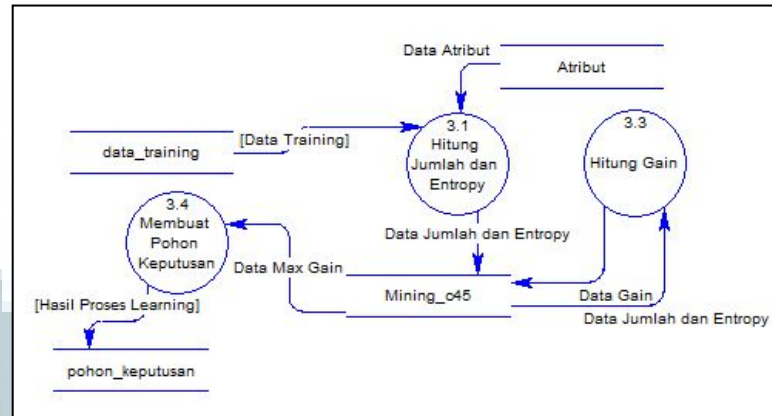
DFD level 1-1 *View Data Learning* memiliki tiga entitas *User*, *Konsumen*, *Bank*, dua proses *Form View Data Learning*, *Form Data Learning*, satu tabel *data_learning*, dan delapan aliran data. Proses *Form View Data Learning* memiliki dua masukan aliran data *Dtb Learning* dari tabel *data_learning*, aliran data *Pencarian Data Learning* dari entitas *User*, dan dua keluaran aliran data *Dt Learning* ke entitas *User*, aliran data *Data Pilihan* ke proses *Form Data Learning*. Proses *Form Data Learning* memiliki tiga masukan aliran data *Data Pilihan* dari proses *Form View Data Learning*, aliran data *Dtbs Learning* dari tabel *data_learning*, aliran data *Atribut Konsumen* dari entitas *Konsumen*, aliran data *Hasil KPR* dari entitas *Bank*, dan keluaran aliran data *Db Learning* ke tabel *data_learning*.



Gambar 3.16 DFD Level 1-3 View Data Testing

DFD level 1-1 *View Data Testing* memiliki satu entitas *User*, dua proses *Form View Data Testing*, *Form Data Testing*, satu tabel *data_testing*, dan delapan aliran data. Proses *Form View Data Testing* memiliki dua masukan aliran data *Dtb Testing* dari tabel *data_testing*, aliran data *Pencarian Data Testing* dari entitas *User*, dan dua keluaran aliran data *Dt Testing* ke entitas *User*, aliran data *Data Pilihan* ke proses *Form Data Testing*. Proses *Form Data Testing* memiliki satu masukan aliran data *Data Pilihan* dari proses *Form View Data Testing*, aliran data

Dtbs *Testing* dari tabel *data_testing*, dan keluaran aliran data Db *Testing* ke tabel *data_testing*.



Gambar 3.17 DFD Level 1-2 Proses Learning

DFD *level* 1-2 Proses *Learning* memiliki tiga proses Hitung Jumlah dan Entropy, Hitung Gain, Membuat Pohon Keputusan, empat tabel *Atribut*, *Mining_c45*, *pohon_keputusan*, *data_training*, dan tujuh aliran data. Proses Hitung Jumlah dan Entropy memiliki dua masukan aliran data Data Atribut dari tabel *Atribut*, aliran data Data Training dari tabel *data_training*, dan satu keluaran aliran data Data Jumlah dan Entropy ke tabel *mining_c45*. Proses Hitung Gain memiliki satu masukan aliran data Data Jumlah dan Entropy dari tabel *mining_c45*, dan keluaran aliran data Data Gain ke tabel *mining_c45*. Proses Membuat Pohon Keputusan memiliki satu masukan aliran data Data Max Gain dari tabel *mining_c45*, dan keluaran aliran data Hasil Proses *Testing* ke tabel *pohon_keputusan*.

3.3.3 Rancangan Antar Muka Aplikasi

Rancangan antarmuka aplikasi terdiri dari enam *form* dan dua *loading form*. Enam *form* terdiri dari *form* memasukkan data *testing*, dan *testing, form* melihat data *testing*, dan *testing, form* melihat pohon keputusan, dan *form* melihat

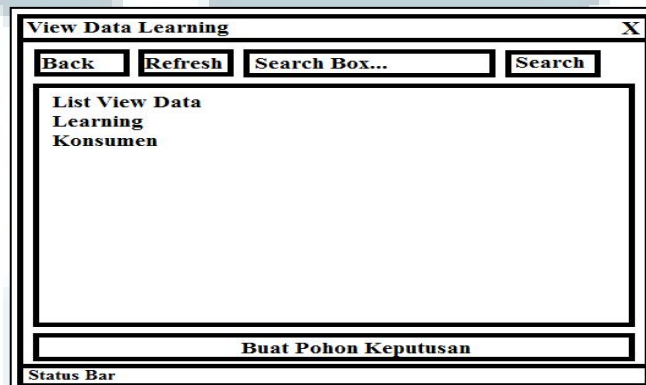
hasil prediksi. Dua *loading form* yaitu *loading form* untuk proses testing dan proses testing. Rancangan antarmuka *form* adalah sebagai berikut :



Gambar 3.18 Rancang Interface Prediksi Konsumen Main

1. *Form* Prediksi Konsumen *Main*

Form Prediksi Konsumen *Main* merupakan *form* utama dari aplikasi ini dan *user* dapat memilih untuk melihat data *testing*, melihat data *testing*, masukkan data *testing*, memasukkan data *testing*, *help*, atau keluar dari aplikasi.



Gambar 3.19 Rancang Interface View Data Testing

2. *Form* Melihat Data *Learning* (View Data Learning)

Form melihat data *learning* (View Data Learning) merupakan *form* untuk melihat semua data *learning* yang tersedia. Pada *form* ini *user* dapat memilih konsumen untuk diubah data *learning* nya. Mengubah data pada *form* ini

dilakukan dengan mengklik dua kali konsumen yang ingin diubah data *learning* nya dan akan masuk ke form masukkan data *learning*. *Text box search* dapat diisi nama konsumen yang dicari dan akan dieksekusi pencariannya jika *user* mengklik *buton search*. *Buton refresh* diklik untuk memuat ulang data *learning*. *Buton back* diklik untuk keluar dari *form*. Jika *user* ingin melakukan pembuatan aturan maka *user* dapat mengklik *buton* buat pohon keputusan dan akan masuk ke *loading form*.

Gambar 3.20 Rancangan Form Masukkan Data Learning

3. Form Memasukkan Data *Learning* (Insert Data Learning)

Form memasukkan data *learning* berfungsi untuk memasukkan atau mengubah data *learning*. Halaman ini muncul pada saat pengguna memilih konsumen pada *form* melihat data *learning*. Pengguna dapat mengisi data *learning*. Setelah mengisi data *learning* pengguna dapat menyimpan data dengan menekan tombol simpan. Atau pengguna dapat kembali/keluar dengan menekan tombol *back*.

Gambar 3.21 Rancangan Form View Data Testing

4. *Form* Melihat Data *Testing* (View Data Testing)

Form melihat data *testing* (View Data Testing) merupakan *form* untuk melihat semua data *testing* yang tersedia. Pada *form* ini *user* dapat memilih konsumen untuk dimasukkan atau diubah data *testing* nya. Mengubah data pada *form* ini dilakukan dengan mengklik dua kali konsumen yang ingin diubah data *testing* nya dan akan masuk ke form masukkan data *testing*. *Text box search* dapat diisikan nama konsumen yang dicari dan akan dieksekusi pencariannya jika *user* mengklik *buton search*. *Button refresh* diklik untuk memuat ulang data *testing*. *Button back* diklik untuk keluar dari *form*. Jika *user* ingin melakukan prediksi data maka *user* dapat mengklik *button* prediksi data dan akan masuk ke *loading form*.

Insert Data Testing		X	
Data Umum			
ID Customer	<input type="text"/>		
Nama	<input type="text"/>		
Data KPR			
Umur	<input type="text"/>	Lama Bekerja	<input type="text"/>
Status Pernikahan	<input type="text"/>	Pekerjaan	<input type="text"/>
Jumlah Tanggungan	<input type="text"/>	Jenis Kelamin	<input type="text"/>
Pendapatan	<input type="text"/>	Alamat Kantor	<input type="text"/>
Alamat KTP	<input type="text"/>	Hasil Prediksi	-
<input type="button" value="Back"/>		<input type="button" value="Simpan"/>	

Gambar 3.22 Rancangan Form Masukkan Data Testing

5. Form Memasukkan Data *Testing* (Insert Data Testing)

Form memasukkan data *testing* berfungsi untuk memasukkan atau mengubah data *testing*. Halaman ini muncul pada saat pengguna memilih konsumen pada *form* melihat data *testing*. Pengguna dapat mengisi data *testing*. Setelah mengisi data *testing* pengguna dapat menyimpan data dengan menekan tombol simpan. Atau pengguna dapat kembali/keluar dengan menekan tombol *back*.

3.3.4 Struktur Tabel

Berikut merupakan penjelasan mengenai semua tabel yang terdapat dalam *database* yang digunakan saat membangun aplikasi sistem informasi KPR konsumen.

- Nama Tabel : data_training

Fungsi : menyimpan data *learning* yang dimasukkan atau diubah untuk dihitung jumlah dan entropinya.

Primarykey : -

Foreign key : id_customer

Tabel 3.3 Struktur Tabel data_training

Nama Tabel	Tipe Data	Keterangan
id_customer	Int(4)	<i>Foreign key</i> tabel
Nama_konsumen	Varchar(100)	Nama Konsumen
umur	Varchar(100)	Umur Konsumen
status_pernikahan	Varchar(100)	Status Pernikahan Konsumen
jumlah_tanggungan	Varchar(100)	Anak/Tanggungan Yang Dimiliki Konsumen
Pendapatan	Varchar(100)	Pendapat Perbulan Konsumen
lama_bekerja	Varchar(100)	Lama Bekerja Konsumen
Pekerjaan	Varchar(100)	Pekerjaan Konsumen
jenis_kelamin	Varchar(100)	Jenis Kelamin Konsumen
Alamat_ktp	Varchar(100)	Alamat Konsumen
Alamat_kantor	Varchar(100)	Alamat Kantor
hasil_prediksi	Varchar(100)	Hasil Dari Prediksi Konsumen

2. Nama Tabel : data_testing

Fungsi : menyimpan data *testing* yang dimasukkan atau diubah untuk dilakukan pengujian pada pohon keputusan.

Primarykey : -

Foreign key : id_customer

Tabel 3.4 Struktur Tabel data_testing

Nama Tabel	Tipe Data	Keterangan
id_customer	Int(4)	<i>Foreign key</i> tabel
Nama_konsumen	Varchar(100)	Nama Konsumen
umur	Varchar(100)	Umur Konsumen

Tabel 3.4 Struktur Tabel data_testing (Lanjutan)

Nama Tabel	Tipe Data	Keterangan
status_pernikahan	Varchar(100)	Status Pernikahan Konsumen
jumlah_tanggungannya	Varchar(100)	Anak/Tanggungan Yang Dimiliki Konsumen
Pendapatan	Varchar(100)	Pendapat Perbulan Konsumen
lama_bekerja	Varchar(100)	Lama Bekerja Konsumen
Pekerjaan	Varchar(100)	Pekerjaan Konsumen
jenis_kelamin	Varchar(100)	Jenis Kelamin Konsumen
Alamat_ktp	Varchar(100)	Alamat Konsumen
Alamat_kantor	Varchar(100)	Alamat Kantor
hasil_prediksi	Varchar(100)	Hasil Dari Prediksi Konsumen

3. Nama Tabel : mining_c45

Fungsi : Menyimpan jumlah, entropi, dan gain setelah proses *testing* dijalankan untuk dicari *gain* terbesar.

Primarykey : -

Foreign key : id_atribut

Tabel 3.5 Struktur Tabel mining_c45

Nama Tabel	Tipe Data	Keterangan
Id_Atribut	Int(4)	<i>Foreign key</i> tabel
Jml_total	Varchar(5)	Jumlah total atribut
Jml_diterima	Varchar(5)	Jumlah diterima

Tabel 3.5 Struktur Tabel mining_c45 (Lanjutan)

Nama Tabel	Tipe Data	Keterangan
Jml_ditolak	Varchar(5)	Jumlah ditolak
Entropy	Varchar(10)	Nilai entropi
Inf_gain	Varchar(10)	Nilai Information Gain
Split_info	Varchar(10)	Nilai Split Information
Gain_ratio	Varchar(10)	Nilai Gain Ratio

4. Nama Tabel : pohon_keputusan
 Fungsi : Menyimpan pohon keputusan untuk digunakan dalam memprediksi hasil.
 Primarykey : -
 Foreign key : id_atribut

Tabel 3.6 Struktur Tabel pohon_keputusan

Nama Tabel	Tipe Data	Keterangan
Id_Atribut	Int(4)	<i>Foreign key</i> tabel
Id_parent	Int(11)	Menyimpan Id Parent
Jml_diterima	Varchar(5)	Jumlah diterima
Jml_ditolak	Varchar(5)	Jumlah ditolak
Keputusan	Varchar(100)	Keputusan jika data memiliki atribut tersebut
Diproses	Varchar(10)	Penanda apakah atribut <i>node</i> sudah diproses atau belum
Kondisi_atribut	Varchar(255)	Sebagai pointer saat menghitung jumlah data
Looping_kondisi	Varchar(15)	Penanda apakah atribut <i>node</i> sudah looping kondisi atau belum

5. Nama Tabel : atribut
- Fungsi : Menyimpan atribut yang digunakan dalam *learning*.
- Primarykey* : id_atribut
- Foreign key* : -

Tabel 3.7 Struktur Tabel Atribut

Nama Tabel	Tipe Data	Keterangan
Id_Atribut	Int(4)	<i>Primary key</i> tabel
Atribut	Varchar(100)	Nama atribut
Nilai_atribut	Varchar(100)	Nilai atribut
Tipe_data	Varchar(100)	Tipe data atribut
Keterangan	Varchar(10)	Keterangan atribut

3.4 Analisis Model

Perhitungan manual Algoritma C4.5 akan dijelaskan pada sub bab ini, guna memperjelas Algoritma yang dipergunakan dalam pengembangan sistem. Perhitungan manual akan mepergunakan lima belas data konsumen sampel, memiliki empat atribut, dan dua kategori. Empat atribut yang digunakan yaitu jumlah tanggungan, pendapatan, pekerjaan, dan Umur. Data akan berada dalam dua kelas yaitu diterima, dan ditolak. Jumlah tanggungan merupakan atribut kategorikal yang terdiri dari nol sampai empat. Pendapatan merupakan atribut numerikal. Pekerjaan merupakan atribut kategorikal yang terdiri dari pegawai negeri, karyawan, dan wiraswasta. Umur merupakan atribut numerik yang terdiri dari <21, 21-50, dan >50 S1,D3,SMA. Berikut ini adalah sampel data yang digunakan untuk analisis model.

Tabel 3.8 Tabel sampel data perhitungan node pertama

Konsumen	Jumlah Tanggungan	Pendapatan	Pekerjaan	Pendidikan Terakhir	Kategori
K1	0	2,800,000.00	Pegawai Negeri	<21	Baik
K2	1	1,500,000.00	Wiraswasta	>50	Jelek
K3	0	1,900,000.00	Karyawan	>50	Jelek
K4	0	2,700,000.00	Karyawan	21-50	Jelek
K5	2	3,200,000.00	Wiraswasta	21-50	Baik
K6	1	3,000,000.00	Pegawai Negeri	21-50	Baik
K7	2	3,800,000.00	Karyawan	>50	Baik
K8	0	5,200,000.00	Wiraswasta	<21	Jelek
K9	1	4,000,000.00	Pegawai Negeri	<21	Baik
K10	2	2,800,000.00	Karyawan	21-50	Jelek
K11	2	3,500,000.00	Karyawan	<21	Baik
K12	0	1,500,000.00	Pegawai Negeri	21-50	Jelek
K13	0	3,800,000.00	Wiraswasta	<50	Baik
K14	0	2,800,000.00	Karyawan	<21	Baik

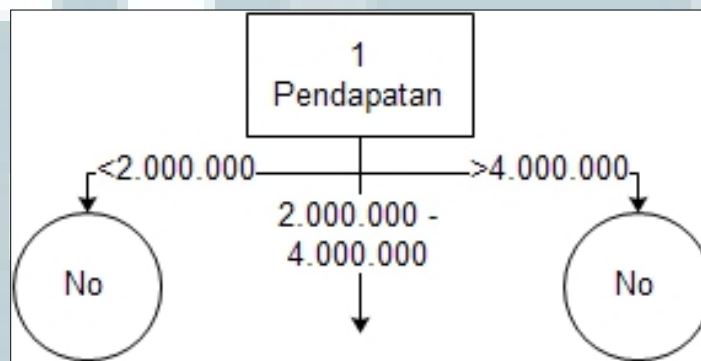
Pohon keputusan terbentuk dari *node* dan cabang-cabang yang dihitung satu per satu secara *top-down* dimulai dari *node* yang akan menjadi akar. *Node* terpilih berdasarkan perhitungan nilai *gain* paling besar diantara semua atribut. Atribut kategorikal akan menggunakan perhitungan *information gain* seperti pada Rumus 2.1, sedangkan atribut numerikal akan menggunakan perhitungan *gain ratio* seperti pada Rumus 2.3. Untuk menghitung *gain* diperlukan perhitungan entropi total dan entropi detail seperti pada Rumus 2.2. Untuk atribut numerikal sebelum dapat menghitung *gain ratio* diperlukan perhitungan *split information* seperti pada rumus 2.4. Untuk menentukan cabang dari pohon keputusan yang akan terbentuk terlebih dahulu perlu dihitung entropi dan *gain*. Setelah mengikuti semua aturan di atas maka akan memperoleh hasil seperti Tabel 3.9.

Tabel 3.9 Hasil perhitungan *gain* dan entropi untuk node pertama

Node	Atribut		Total Sampel	Diterima	Ditolak	Entropi	Perhitungan		
1	Total		14	8	6	0.9852 28136			
	Pekerjaan	Pegawai Negeri	4	3	1	0.8112 78124			
		Karyawan	6	3	3	1			
		Wiraswasta	4	2	2	1			
							IG :	0.039 149	
	Umur	<21	4	2	2	1			
		21-50	5	2	3	0.9709 50594			
		>50	5	4	1	0.7219 28095			
								IG :	0.094 914
							SI :	1.577 406	
							GR :	0.060 171	
	Jumlah Tanggungan	0	7	3	4	0.9852 28136			
		1	3	2	1	0.9182 95834			
		2	4	3	1	0.8112 78124			
								IG :	0.064 043
	Pendapatan	<2000000	3	0	3	0			
		2000000 - 4000000	10	8	2	0.7219 28095			
		>4000000	1	0	1	0			
								IG:	0.469 565
								SI :	1.094 914
						GR :	0.428 86		

IG pada Tabel 3.9 merupakan *information gain*, SI merupakan *split information*, GR merupakan *gain ratio*. Berdasarkan Tabel 3.9 diketahui *gain* paling tinggi ada pada atribut pendapatan oleh karena itu pendapatan akan dipilih

sebagai *node* pertama atau *node* akar. Cabang yang akan terbentuk adalah nilai yang terdapat pada atribut tersebut. Dalam sampel ini cabang yang terbentuk adalah lebih kecil dari dua juta, dua juta sampai empat juta, dan lebih besar dari empat juta. Cabang lebih kecil dari dua juta dan lebih besar dari empat juta tidak memiliki sampel pada kolom diterima oleh karena itu *node* selanjutnya pada cabang ini memiliki klasifikasi ditolak. Seperti pada gambar 3.23.



Gambar 3.23 Pohon keputusan cabang pertama

Untuk cabang dua juta sampai empat juta *node* selanjutnya belum memiliki kelas, dan dapat diketahui dengan cara yang sama seperti yang sudah dijelaskan yaitu dengan mencari gain paling tinggi berdasarkan perhitungan entropi. Tetapi atribut pendapatan tidak dimasukkan dalam perhitungan karena sudah menjadi *node* pertama, dan data yang digunakan hanya data konsumen yang memiliki pendapatan dua juta sampai empat juta. Berikut ini adalah data yang digunakan untuk mencari *node* kedua.

Tabel 3.10 Tabel sampel data perhitungan node kedua

Konsumen	Jumlah Tanggungan	Pekerjaan	Umur	Kelas
K4	0	Karyawan	21-50	Jelek
K10	2	Karyawan	21-50	Jelek
K6	1	Pegawai Negeri	21-50	Baik
K5	2	Wiraswasta	21-50	Baik
K14	0	Karyawan	<21	Baik

Tabel 3.10 Tabel sampel data perhitungan node kedua (lanjutan)

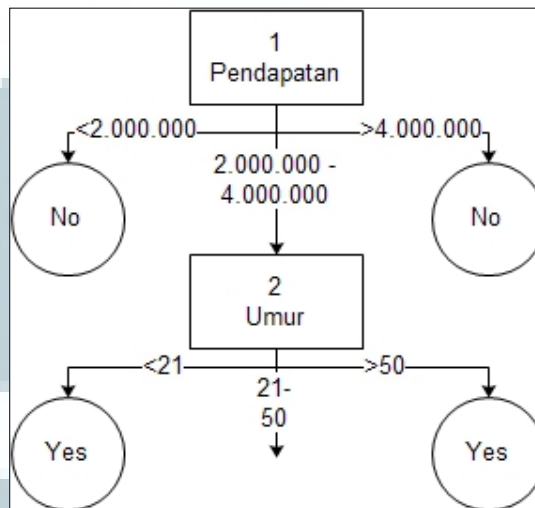
K1	0	Pegawai Negeri	<21	Baik
K11	2	Karyawan	<21	Baik
K9	1	Pegawai Negeri	<21	Baik
K13	0	Wiraswasta	>50	Baik
K7	2	Karyawan	>50	Baik

Berdasarkan data pada tabel 3.10 akan dihitung kembali *gain* dan entropi untuk mengetahui *node* kedua. Berikut ini adalah hasil dari perhitungan tersebut.

Tabel 3.11 Hasil perhitungan *gain* dan entropi untuk node kedua

Node	Atribut		Total Sampe 1	Sampe 1 Baik	Sampe 1 Jelek	Entropi	Perhitu ngan	
2	Total		10	8	2	0.72192 8095		
	Pekerjaan	Pegawai Negeri	3	3	0	0		
		Karyawan	5	3	2	0.97095 0594		
		Wiraswasta	2	2	0	0		
							IG :	0.2364 52798
	Umur	>50	2	2	0	0		
		21-50	4	2	2	1		
		<21	4	4	0	0		
							IG :	0.3219 28095
							SI :	1.4338 25
						GR :	0.2445 24	
	Jumlah Tanggungan	0	4	3	1	0.81127 8124		
		1	2	2	0	0		
		2	4	3	1	0.81127 8124		
							Informa tion gain :	0.0729 05595

Atribut yang memiliki *gain* paling tinggi adalah umur oleh karena itu umur akan menjadi *node* kedua. Berikut ini adalah gambar pohon keputusan yang sudah terbentuk.



Gambar 3.24 Pohon keputusan cabang kedua

Node ketiga dicari dengan cara yang sama seperti *node* pertama dan kedua yaitu dengan mencari *gain* paling tinggi. Umur dan pendapatan tidak dimasukkan dalam perhitungan *node* ketiga. Data yang digunakan hanya data yang memiliki nilai dua satu sampai lima puluh pada atributnya. Berikut ini adalah data yang digunakan untuk mencari *node* ketiga

Tabel 3.12 Hasil Tabel sampel data perhitungan *node* ketiga

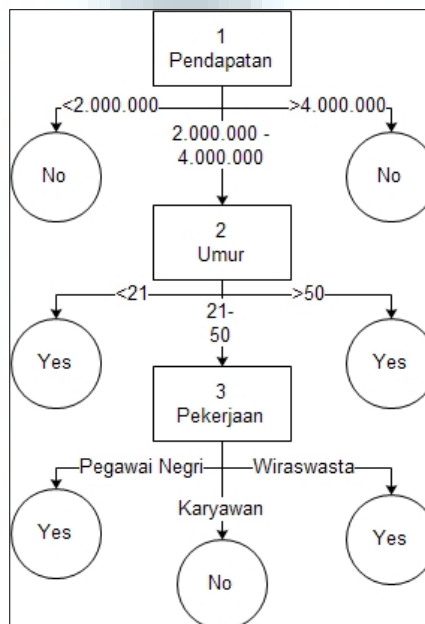
Konsumen	Jumlah Tanggungan	Pekerjaan	Kelas
K4	0	Karyawan	Jelek
K10	2	Karyawan	Jelek
K6	1	Pegawai Negeri	Baik
K5	2	Wiraswasta	Baik

Berdasarkan data pada tabel 3.12 akan dihitung kembali *gain* dan entropi untuk mengetahui *node* ketiga. Berikut ini adalah hasil dari perhitungan tersebut.

Tabel 3.13 Hasil perhitungan *gain* dan entropi untuk node ketiga

Node	Atribut	Total Sampe l	Sampe l Baik	Sampe l Jelek	Entropi	Perhi tunga n	
3	Total	4	2	2		1	
	Pekerjaan	Karyawan	2	0	2		0
		Pegawai Negeri	1	1	0		0
		Wiraswasta	1	1	0		0
						Informa tion gain :	
	Jumlah Tanggung an	0	1	0	1	0	
		1	1	1	0	0	
		2	2	1	1	1	
							Informa tion gain :

Atribut yang memiliki *gain* paling tinggi adalah pendidikan terakhir oleh karena itu pendidikan terakhir akan menjadi *node* kedua. Berikut ini adalah gambar pohon keputusan yang sudah terbentuk.



Gambar 3.25 Pohon keputusan cabang ketiga

Aturan yang terbentuk dari pohon keputusan ini adalah sebagai berikut :

- Jika pendapatan lebih kecil dari dua juta atau lebih besar dari empat juta maka tidak lolos.
- Jika pendapatan dua juta sampai empat juta dan umur lebih kecil dari dua satu atau lebih besar dari lima puluh maka akan lolos.
- Jika pendapatan dua juta sampai empat juta dan umur dia satu sampai lima puluh dan pekerjaan pegawai negeri atau wiraswasta maka akan lolos.
- Jika pendapatan dua juta sampai empat juta dan umur dia satu sampai lima puluh dan pekerjaan karyawan maka tidak lolos.

UMMN