



**SISTEM INFORMASI PENGKLASIFIKASIAN PENERIMAAN CALON  
NASABAH ASURANSI MENGGUNAKAN *CLASSIFICATION TREE C4.5*  
*ALGORITHM* DAN *COST SENSITIVE LEARNING***

**SKRIPSI**

Oleh

**Selvia Gusti Bastian**

**NIM 142410101005**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER**

**2018**



**SISTEM INFORMASI PENGKLASIFIKASIAN PENERIMAAN CALON  
NASABAH ASURANSI MENGGUNAKAN *CLASSIFICATION TREE C4.5*  
*ALGORITHM* DAN *COST SENSITIVE LEARNING***

**SKRIPSI**

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Pendidikan Sarjana (S1) Program Studi Sistem Informasi Universitas Jember dan menncapai gelar Sarjana Komputer

Oleh  
**Selvia Gusti Bastian**  
**NIM 142410101005**

**PROGRAM STUDI SISTEM INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER  
2018**

## PERSEMBAHAN

Skripsi ini saya persembahkan untuk:

1. Allah SWT yang senantiasa memberikan rahmat dan hidayah-Nya untuk mempermudah dan melancarkan dalam mengerjakan skripsi;
2. Ibunda Haryati Ningsih dan Ayahanda Basuni tersayang dan tercinta yang selalu memberikan do'a, kasih sayang, semangat, dan motivasi yang luar biasa;
3. Kakak Ayu Ella Bastian dan Heru Setiawan, adik Annur Meisya Bastian, serta nenek Sitti tercinta yang selalu memberi dukungan dan do'a;
4. Seluruh keluarga besar yang selalu menyayangiku;
5. Sahabat - sahabatku bersama dukungan dan doanya;
6. Guru - guruku sejak taman kanak-kanak sampai dengan perguruan tinggi;
7. Almamater Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

**MOTTO**

يُوقِنُونَ لَا َالَّذِينَ يَسْتَخِفُّونَكَ وَلَا ِحَقُّ َاللَّهِ وَعَدَدَ إِنَّا فَاصِحِرُ

“Dan bersabarlah kamu, sesungguhnya **janji** Allah adalah **benar**”

(Ar-Rum: 60)

وُسْعَهَا إِلَّا نَفْسًا َاللَّهُ يُكَلِّفُ لَا

“Allah tidak **membebani** seseorang melainkan sesuai dengan **kesanggupannya**”

(Al-Baqarah: 286)



**PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Selvia Gusti Bastian

NIM : 142410101005

menyatakan dengan sesungguhnya bahwa karya ilmiah yang berjudul “Sistem Informasi Pengklasifikasian Penerimaan Calon Nasabah Asuransi menggunakan *Classification Tree C4.5 Algorithm* dan *Cost Sensitive Learning*”, adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya, belum pernah diajukan pada institusi mana pun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa ada tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika di kemudian hari pernyataan ini tidak benar.

Jember, 20 Mei 2018

Yang menyatakan,

Selvia Gusti Bastian

NIM 142410101005

**SKRIPSI**

**SISTEM INFORMASI PENGKLASIFIKASIAN PENERIMAAN CALON  
NASABAH ASURANSI MENGGUNAKAN *CLASSIFICATION TREE C4.5*  
*ALGORITHM* DAN *COST SENSITIVE LEARNING***

Oleh

**Selvia Gusti Bastian**

**NIM 142410101005**

Pembimbing:

Dosen Pembimbing Utama : Muhammad Arief Hidayat., S.Kom., M.Kom.

Dosen Pembimbing Pendamping : Ifrina Nuritha., S.Kom., M.Kom.

### PENGESAHAN PEMBIMBING

Skripsi berjudul “Sistem Informasi Pengklasifikasian Penerimaan Calon Nasabah Asuransi Menggunakan *Classification Tree C4.5 Algorithm* dan *Cost Sensitive Learning*” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 31 Mei 2018

tempat : Fakultas Ilmu Komputer Universitas Jember.

Disetujui oleh:

Pembimbing I,

Pembimbing II,

M. Arief Hidayat, S.Kom., M.Kom.

Ifrina Nuritha, S.Kom., M.Kom.

NIP. 198101232010121003

NRP. 760016786

**PENGESAHAN PENGUJI**

Skripsi berjudul “Sistem Informasi Pengklasifikasian Penerimaan Calon Nasabah Asuransi Menggunakan *Classification Tree C4.5 Algorithm* dan *Cost Sensitive Learning*” telah diuji dan disahkan pada:

hari, tanggal : Kamis, 31 Mei 2018

tempat : Fakultas Ilmu Komputer Universitas Jember.

Tim Penguji,

Penguji I,

Penguji II,

Anang Andrianto, ST.,M.T

Tio Dharmawan S.Kom., M.Kom

NIP. 196906151997021002

NRP. 760016851

Mengesahkan

Penjabat Dekan Fakultas Ilmu Komputer,

Prof. Drs. Slamin, M.Comp.Sc., Ph.D

NIP. 196704201992011001



## RINGKASAN

**Sistem Informasi Pengklasifikasian Penerimaan Calon Nasabah Asuransi Menggunakan *Classification Tree C4.5 Algorithm* dan *Cost sensitive learning*;** Selvia Gusti Bastian, 142410101005; 2018: 330 halaman; Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Asuransi merupakan salah satu cara untuk memproteksi diri di masa depan. Kemungkinan yang terjadi merupakan ketidakpastian yang merugikan sebagai akibat suatu peristiwa tidak terduga, oleh karena itu proteksi asuransi akan merubah kerugian menjadi ganti rugi atau santunan klaim dengan syarat pembayaran premi. Saat ini asuransi dapat dianggap sebagai kebutuhan primer untuk sebagian orang. Hal ini dapat terlihat banyaknya orang yang memiliki asuransi baik dikalangan bawah maupun atas. Seiring dengan dianggapnya asuransi sebagai kebutuhan primer, setiap orang berusaha untuk memenuhi akan kebutuhan asuransi tersebut. Permasalahan yang sering timbul dalam perusahaan asuransi adalah banyaknya nasabah yang menunggak dalam membayar premi, oleh karena itu diperlukan sebuah sistem yang dapat mengklasifikasikan calon nasabah mana yang masuk ke dalam kelompok lancar dan nasabah mana yang masuk kedalam kelompok tidak lancar dalam membayar iuran premi, sehingga pihak asuransi bisa mengatasi sejak dini permasalahan tersebut. Proses pengelolaan penerimaan calon nasabah dapat dioptimalkan dan dapat membantu pihak perusahaan asuransi dalam memberikan keputusan pemberian polis asuransi sehingga meminimalisasi terjadinya premi tidak lancar, menggunakan konsep *data mining* menggunakan metode klasifikasi.

Pada penelitian ini dibangun sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*. Algoritma *classification tree C4.5* digunakan untuk menghasilkan

*rule* yang diperlukan untuk proses klasifikasi kelancaran calon nasabah dalam pembayaran premi dan terbentuk dari hasil konversi pembangunan pohon klasifikasi (*classification tree*), sedangkan *cost sensitive learning* digunakan untuk mengatasi masalah *misclassification cost* di setiap split sehingga berpengaruh pada kinerja pengklasifikasi (*C4.5 algorithm*), kemudian menentukan termasuk pada kelas / kelompok yang tepat. Penentuan calon nasabah yang tepat (kelancaran dalam pembayaran premi) dikelompokkan menjadi 2 bagian yaitu Kelompok Lancar dan Kelompok Tidak Lancar.

Tujuan dibangunnya sistem informasi pengklasifikasian penerimaan calon nasabah asuransi ini diharapkan dapat membantu pihak asuransi dalam mengelola penerimaan calon nasabah asuransi serta dapat membantu penentuan penerimaan calon nasabah asuransi sehingga dapat mengurangi kerugian perusahaan pada nasabah asuransi yang menunggak dalam pembayaran premi.

## PRAKATA

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan skripsi dengan judul “Sistem Informasi Pengklasifikasian Penerimaan Calon Nasabah Asuransi menggunakan *Classification Tree C4.5 Algorithm* dan *Cost Sensitive Learning*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Strata Satu (S1) pada Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada:

1. Prof. Drs. Slamin, M.Comp.Sc., Ph.D., selaku Dekan Fakultas Ilmu Komputer Universitas Jember;
2. Muhammad Arief Hidayat, S.Kom., M.Kom., selaku Dosen Pembimbing Utama dan Ifrina Nuritha, S.Kom., M.Kom., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran, dan perhatian dalam penulisan skripsi;
3. Dr Saiful Bukhori, S.T., M.Kom., sebagai dosen pembimbing akademik, yang telah mendampingi penulis sebagai mahasiswa;
4. Seluruh Bapak dan Ibu dosen beserta staf karyawan di Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember;
5. Ibunda Haryati Ningsih dan Ayahanda Basuni tersayang dan tercinta yang selalu memberikan do'a, kasih sayang, semangat, dan motivasi yang luar biasa;
6. Kakak Ayu Ella Bastian dan Heru Setiawan, adik Annur Meisya Bastian, serta nenek Sitti tercinta yang selalu memberi dukungan dan do'a;
7. Seluruh keluarga besar yang selalu menyayangiku;
8. Perusahaan Asuransi XYZ Cabang Jember sebagai lokasi penelitian yang telah meluangkan waktu yang membantu dalam melancarkan skripsi ini;
9. Keluarga besar BIDIKMISI Universitas Jember;

10. Direktorat Pembelajaran dan Kemahasiswaan Ditjen Dikti yang telah memberikan beasiswa bidikmisi kepada penulis sehingga dapat menyelesaikan pendidikan S1;
11. Sahabat seperantauan yaitu Dini, Choy, Ayu, Upin, Ika, Oyik, dan Widia yang selalu memberikan dukungan dan do'a terbaik;
12. Sahabat seperjuangan yaitu Riya, Nurul, Lutfi, Laily, Laras, Tika, Dian, Bazliah, Difari, dan Nugroho yang selalu bersama saling menguatkan untuk lulus;
13. Keluarga besar SENSATION angkatan 2014 yang telah menjadi keluarga selama menempuh pendidikan S1;
14. Teman-Teman Fakultas Ilmu Komputer di semua angkatan.
15. Sahabat KKN Setengah 8 serta keluarga Desa Ramban Wetan yang selalu memberikan dukungan dan do'a terbaik;
16. Semua pihak yang tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa laporan ini masih jauh dari sempurna, oleh sebab itu penulis mengharapkan adanya masukan yang bersifat membangun dari semua pihak. Penulis berharap skripsi ini dapat bermanfaat bagi semua pihak.

Jember, 20 Mei 2018

Penulis

## DAFTAR ISI

	Halaman
PERSEMBAHAN .....	ii
MOTTO .....	1
PERNYATAAN.....	2
SKRIPSI.....	3
PENGESAHAN PEMBIMBING.....	4
PENGESAHAN PENGUJI.....	5
RINGKASAN .....	6
PRAKATA.....	8
DAFTAR ISI.....	10
DAFTAR TABEL.....	14
DAFTAR GAMBAR .....	19
DAFTAR LAMPIRAN.....	<b>Error! Bookmark not defined.</b>
BAB 1. PENDAHULUAN .....	23
1.1 Latar Belakang .....	23
1.2 Rumusan Masalah .....	26
1.3 Tujuan dan Manfaat .....	27
1.3.2 Tujuan.....	27
1.3.2 Manfaat.....	27
1.4 Batasan Masalah.....	28
1.5 Sistematika Penulisan.....	29
BAB 2. TINJAUAN PUSTAKA .....	31
2.1 Penelitian Terdahulu .....	31
2.2 Asuransi.....	34
2.3 Sistem Informasi .....	35

2.4	<i>Data Mining</i> .....	35
2.5	Klasifikasi.....	36
2.6	<i>Decision Tree</i> .....	37
2.7	<i>C4.5 Algorithm</i> .....	38
2.8	<i>Cost Sensitive Learning</i> .....	43
2.9	<i>Confusion Matrix</i> .....	46
2.10	<i>System Development Life Cycle (SDLC)</i> .....	47
BAB 3.	METODOLOGI PENELITIAN .....	49
3.1	Jenis Penelitian .....	49
3.2	Tempat dan Waktu Penelitian .....	49
3.3	Tahapan Penelitian .....	49
3.3.1	Analisis Kebutuhan .....	51
3.3.2	Desain Sistem.....	54
3.3.3	Implementasi .....	55
3.3.4	Pengujian Sistem .....	55
3.3.5	Pemeliharaan .....	56
3.4	Gambaran Umum Sistem .....	56
BAB 4.	PENGEMBANGAN SISTEM.....	63
4.1	<i>C4.5 Algorithm</i> dan <i>Cost Sensitive Learning</i> untuk Klasifikasi Kelancaran Pembayaran Premi .....	63
4.2	Analisis Kebutuhan .....	65
4.2.1	Pengumpulan Data .....	65
4.2.2	Kebutuhan Fungsional.....	70
4.2.3	Kebutuhan Non-Fungsional .....	71
4.2.4	Fungsi Sistem .....	72
4.3	Desain Sistem .....	74
4.3.1	Elemen <i>Input</i> dan <i>Output</i> .....	74

4.3.2	<i>Use Case Diagram</i> .....	75
4.3.3	<i>Use Case Skenario</i> .....	79
4.3.4	<i>Activity Diagram</i> .....	102
4.3.5	<i>Sequence Diagram</i> .....	114
4.3.6	<i>Class Diagram</i> .....	130
4.3.7	<i>Entity Relationship Diagram (ERD)</i> .....	133
4.4	Penulisan Kode Program .....	134
4.5	Pengujian Sistem .....	138
4.6.1	<i>White Box Testing</i> .....	139
4.6.2	<i>Black Box Testing</i> .....	149
BAB 5.	HASIL DAN PEMBAHASAN .....	152
5.1	Perhitungan <i>C4.5 Algorithm</i> dan <i>Cost Sensitive Learning</i> .....	152
5.2	Implementasi <i>C4.5 Algorithm</i> dan <i>Cost Sensitive Learning</i> .....	181
5.3	Hasil Pembuatan Sistem Informasi Pengklasifikasian Penerimaan Calon Nasabah Asuransi.....	196
5.3.1	Fitur <i>Login</i> .....	196
5.3.2	Fitur Register .....	197
5.3.3	Halaman <i>Dashboard</i> .....	198
5.3.4	Fitur Mengelola <i>Data Set</i> .....	199
5.3.5	Fitur Melihat <i>Data Training</i> .....	201
5.3.6	Fitur Melihat <i>Classification tree</i> .....	201
5.3.7	Fitur Melihat Hasil <i>Classification tree</i> .....	202
5.3.8	Fitur Melihat <i>Rule</i> .....	203
5.3.9	Fitur Melihat <i>Data Testing</i> .....	204
5.3.10	Fitur Melihat Hasil Uji Klasifikasi.....	204

5.3.11	Fitur Mengubah Data Diri .....	205
5.3.12	Fitur Mendaftar <i>Agent</i> .....	205
5.3.13	Fitur Melihat Aktivitas Penerimaan .....	207
5.3.14	Fitur Melihat Aktivitas Saran .....	208
5.3.15	Fitur Mengelola Manfaat Asuransi .....	208
5.3.16	Fitur Mengelola Persyaratan Nasabah Asuransi .....	208
5.3.17	Fitur Melihat Manfaat Asuransi .....	208
5.3.18	Fitur Melihat Persyaratan Nasabah Asuransi .....	208
5.3.19	Fitur Mengelola Data <i>Agent</i> .....	209
5.3.20	Fitur Mengelola Data Calon Nasabah .....	209
5.3.21	Fitur Melihat dan Mencetak Data Nasabah .....	211
5.3.22	Fitur Melihat Hasil Klasifikasi .....	211
5.3.23	Fitur Melihat Riwayat Klasifikasi .....	213
5.3.24	Fitur Validasi Calon Nasabah .....	213
5.4	Hasil Uji Klasifikasi .....	213
5.4.1	Uji Klasifikasi menggunakan <i>Cost Sensitive Learning</i> .....	216
5.4.2	Uji Klasifikasi tanpa <i>Cost Sensitive Learning</i> .....	220
5.4.3	Pengaruh <i>Cost Sensitive Learning</i> pada <i>C4.5 Algorithm</i> .....	224
BAB 6.	PENUTUP .....	226
6.1.	Kesimpulan .....	226
6.2.	Saran .....	227
	DAFTAR PUSTAKA .....	228



**DAFTAR TABEL**

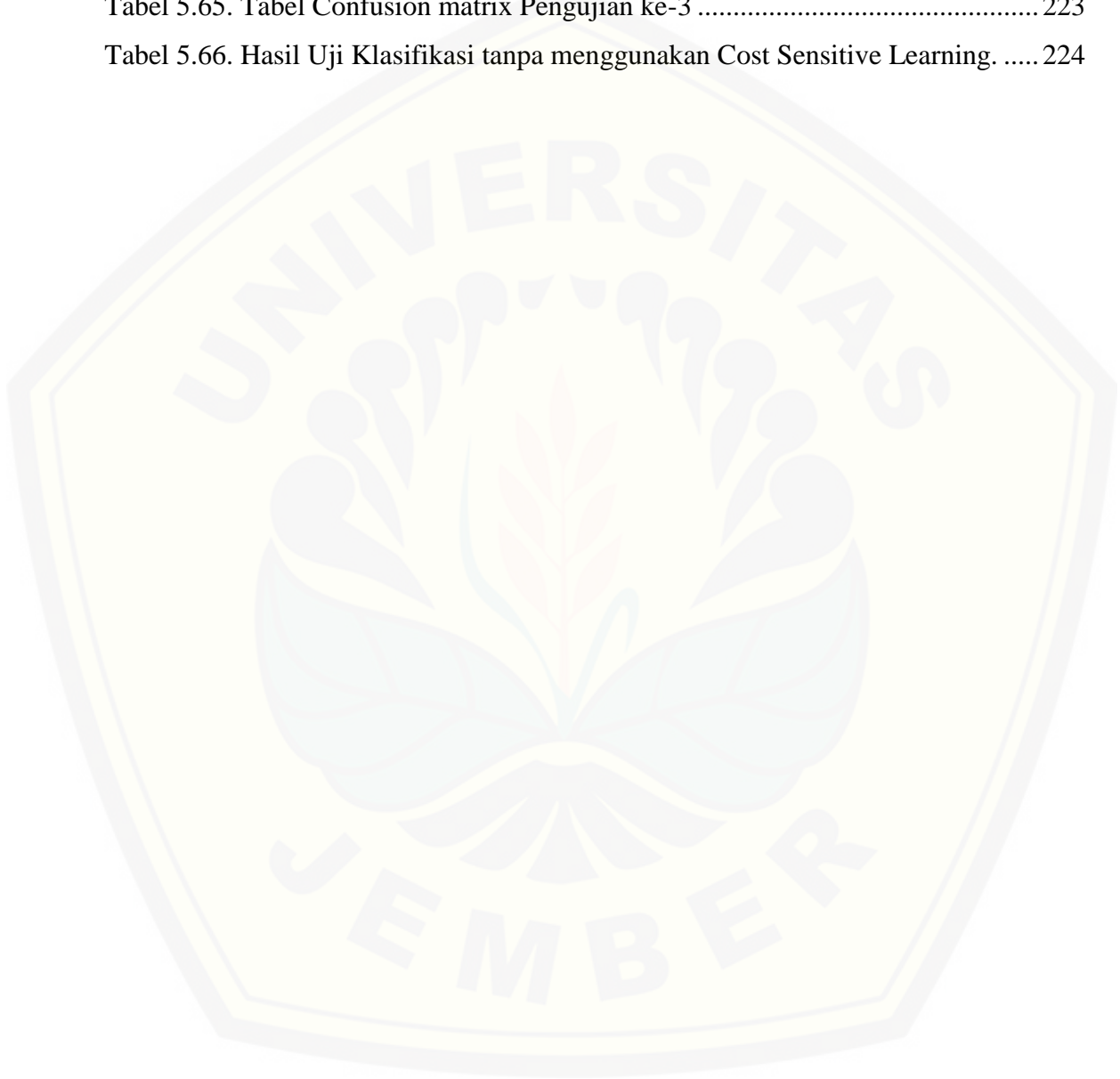
	Halaman
Tabel 2.1. Penelitian Terdahulu .....	31
Tabel 2.2. Data Training .....	40
Tabel 2.3. Perhitungan Entropy pada Class .....	41
Tabel 2.4. Perhitungan Atribut Jenis Kelamin .....	41
Tabel 2.5. Perhitungan Atribut Status Marital .....	41
Tabel 2.6. Perhitungan Atribut Premi .....	42
Tabel 2.7. Matriks Kontingensi Kelas Prediksi dan Aktual.....	46
Tabel 4.1. Atribut dan Nilai Atribut.....	69
Tabel 4.2. Data Agent .....	69
Tabel 4.3. Data Manfaat Asuransi .....	69
Tabel 4.4. Data Persyaratan Calon Nasabah .....	70
Tabel 4.5. Deskripsi Aktor .....	77
Tabel 4.6. Deskripsi Use Case .....	78
Tabel 4.7. Skenario Menambah Mendaftar Agent .....	80
Tabel 4.8. Skenario Mengelola Data Calon Nasabah .....	84
Tabel 4.9. Skenario Melihat Hasil Klasifikasi .....	89
Tabel 4.10. Skenario Mengelola Data Set.....	94
Tabel 4.11. Skenario Melihat Data Training.....	96
Tabel 4.12. Skenario Melihat Hasil Classification Tree dan Rule .....	97
Tabel 4.13. Skenario Melihat Data Testing .....	100
Tabel 4.14. Skenario Melihat Hasil Uji Klasifikasi .....	101
Tabel 4.15. Potongan Kode Program Perhitungan C4.5 Algorithm .....	134
Tabel 4.16. Potongan Kode Program Perhitungan Cost Sensitive Learning .....	136
Tabel 4.17. Potongan Kode Program Uji Klasifikasi.....	137
Tabel 4.18. Potongan Kode Program Confusion matrix .....	137

Tabel 4.19. Test Case Function Index()	148
Tabel 4.20. Test Case Function bagidataset()	148
Tabel 4.21. Test Case Function reset()	148
Tabel 4.22. Test Case Function insert_atribut()	148
Tabel 4.23. Black Box Testing Melihat Hasil Pembangunan Classification tree	149
Tabel 4.24. Black Box Testing Melihat Hasil Uji Klasifikasi	150
Tabel 5.1. Jumlah Nilai Tiap Class dan Entropy Total	153
Tabel 5.2. Jumlah Nilai, Entropy, dan Gain	153
Tabel 5.3. Jumlah Nilai Tiap Class dan Entropy Total Atribut Penghasilan Nilai 1	154
Tabel 5.4. Jumlah Nilai Tiap Class dan Entropy Total Atribut Penghasilan Nilai 2	154
Tabel 5.5. Jumlah Nilai, Entropy, dan Gain Atribut Penghasilan Nilai 1	155
Tabel 5.6. Jumlah Nilai, Entropy, dan Gain Atribut Penghasilan Nilai 2	155
Tabel 5.7. Jumlah Nilai Tiap Class dan Entropy Total Atribut Cara Bayar Nilai 1	157
Tabel 5.8. Jumlah Nilai Tiap Class dan Entropy Total Atribut Cara Bayar Nilai 2	157
Tabel 5.9. Jumlah Nilai Tiap Class dan Entropy Total Atribut Rencana Pembayaran Nilai 2	157
Tabel 5.10. Jumlah Nilai, Entropy, dan Gain Atribut Cara Bayar Nilai 1	158
Tabel 5.11. Jumlah Nilai, Entropy, dan Gain Atribut Cara Bayar Nilai 2	158
Tabel 5.12. Jumlah Nilai, Entropy, dan Gain Atribut Rencana Pembayaran Nilai 2	159
Tabel 5.13. Jumlah Nilai Tiap Class dan Entropy Total Atribut Premi Nilai 1	160
Tabel 5.14. Jumlah Nilai Tiap Class dan Entropy Total Atribut Status Marital Nilai 1	161
Tabel 5.15. Jumlah Nilai Tiap Class dan Entropy Total Atribut Status Marital Nilai 2	161
Tabel 5.16. Jumlah Nilai Tiap Class dan Entropy Total Atribut Usia Nilai 2	161
Tabel 5.17. Jumlah Nilai, Entropy, dan Gain Atribut Premi Nilai 1	161
Tabel 5.18. Jumlah Nilai, Entropy, dan Gain Atribut Status Marital Nilai 1	162
Tabel 5.19. Jumlah Nilai, Entropy, dan Gain Atribut Status Marital Nilai 2	162
Tabel 5.20. Jumlah Nilai, Entropy, dan Gain Atribut Usia Nilai 2	163

Tabel 5.21. Jumlah Nilai Tiap Class dan Entropy Total Atribut Rencana Pembayaran Nilai 1 .....	164
Tabel 5.22. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Rencana Pembayaran Nilai 2.....	165
Tabel 5.23. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Jenis Kelamin Nilai 1 .....	165
Tabel 5.24. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Jenis Kelamin Nilai 2 .....	165
Tabel 5.25. Jumlah Nilai Tiap Class dan Entropy Total Atribut Rencana Pembayaran Nilai 2.....	165
Tabel 5.26. Jumlah Nilai, Entropy, dan Gain Atribut Rencana Pembayaran Nilai 1	166
Tabel 5.27. Jumlah Nilai, Entropy, dan Gain Atribut Rencana Pembayaran Nilai 2	166
Tabel 5.28. Jumlah Nilai, Entropy, dan Gain Atribut Jenis Kelamin Nilai 1 .....	166
Tabel 5.29. Jumlah Nilai, Entropy, dan Gain Atribut Jenis Kelamin Nilai 2 .....	167
Tabel 5.30. Jumlah Nilai, Entropy, dan Gain Atribut Rencana Pembayaran Nilai 2	167
Tabel 5.31. Jumlah Nilai Tiap Class dan Entropy Total Atribut Status Marital Nilai 1 .....	169
Tabel 5.32. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Status Marital Nilai 2 .....	169
Tabel 5.33. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Status Marital Nilai 1 .....	169
Tabel 5.34. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Masa Pembebanan Biaya Nilai 1 .....	170
Tabel 5.35. Jumlah Nilai Tiap Class dan Entropy Total Atribut Masa Pembebanan Biaya Nilai 2 .....	170
Tabel 5.36. Jumlah Nilai Tiap Class dan Entropy Total Atribut Premi Nilai 2.....	170
Tabel 5.37. Jumlah Nilai, Entropy, dan Gain Atribut Status Marital Nilai 1 .....	170
Tabel 5.38. Jumlah Nilai, Entropy, dan Gain Atribut Status Marital Nilai 2 .....	171
Tabel 5.39. Jumlah Nilai, Entropy, dan Gain Atribut Status Marital Nilai 1 .....	171

Tabel 5.40. Jumlah Nilai, Entropy, dan Gain Atribut Masa Pembebanan Biaya Nilai 1 .....	171
Tabel 5.41. Jumlah Nilai, Entropy, dan Gain Atribut Masa Pembebanan Biaya Nilai 2 .....	172
Tabel 5.42. Jumlah Nilai, Entropy, dan Gain Atribut Premi Nilai 2 .....	172
Tabel 5.43. Jumlah Nilai Tiap Class dan Entropy Total Atribut Masa Pembebanan Biaya Nilai 1 .....	175
Tabel 5.44. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Masa Pembebanan Biaya Nilai 2 .....	175
Tabel 5.45. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Jenis Kelamin Nilai 2 .....	175
Tabel 5.46. Jumlah Nilai Tiap <i>Class</i> dan <i>Entropy</i> Total Atribut Premi Nilai 2.....	175
Tabel 5.47. Jumlah Nilai, Entropy, dan Gain Atribut Masa Pembebanan Biaya Nilai 1 .....	176
Tabel 5.48. Jumlah Nilai, Entropy, dan Gain Atribut Masa Pembebanan Biaya Nilai 2 .....	176
Tabel 5.49. Jumlah Nilai, Entropy, dan Gain Atribut Jenis Kelamin Nilai 2 .....	176
Tabel 5.50. Jumlah Nilai, Entropy, dan Gain Atribut Premi Nilai 2 .....	176
Tabel 5.51. Jumlah Nilai Atribut Usia per Class .....	178
Tabel 5.52. Jumlah Nilai Atribut Jenis Kelamin per Class .....	179
Tabel 5.53. Jumlah Nilai Atribut Usia per Class .....	179
Tabel 5.54. Jumlah Nilai Atribut Rencana Pembayaran per Class .....	179
Tabel 5.55. Pelabelan Leaf Node Atribut Usia .....	180
Tabel 5.56. Pelabelan Leaf Node Atribut Jenis Kelamin.....	180
Tabel 5.57. Pelabelan Leaf Node Atribut Usia .....	180
Tabel 5.58. Pelabelan Leaf Node Atribut Rencana Pembayaran .....	180
Tabel 5.59. Tabel Confusion matrix Pengujian ke-1 .....	216
Tabel 5.60. Tabel Confusion matrix Pengujian ke-2 .....	217
Tabel 5.61. Tabel Confusion matrix Pengujian ke-3 .....	218

Tabel 5.62. Hasil Uji Klasifikasi menggunakan Cost Sensitive Learning.....	220
Tabel 5.63. Tabel Confusion matrix Pengujian ke-1 .....	220
Tabel 5.64. Tabel Confusion matrix Pengujian ke-2 .....	221
Tabel 5.65. Tabel Confusion matrix Pengujian ke-3 .....	223
Tabel 5.66. Hasil Uji Klasifikasi tanpa menggunakan Cost Sensitive Learning. ....	224



## DAFTAR GAMBAR

	Halaman
Gambar 2.1. Node pada Classification Tree .....	38
Gambar 2.2. Cabang Masing-Masing Nilai .....	42
Gambar 2.3. Classification tree .....	42
Gambar 2.4. Pelabelan Leaf Node .....	45
Gambar 2.5. Pelabelan Leaf Node menggunakan Cost Sensitive Learning .....	46
Gambar 2.6. Model Waterfall .....	48
Gambar 3.1. Alur Penelitian Sesuai Model Waterfall.....	50
Gambar 3.2. Alur Sistem.....	57
Gambar 3.3. Flowchart Request Data .....	58
Gambar 3.4. Flowchart Proses Learning.....	59
Gambar 3.5. Flowchart Pembangunan Classification Tree.....	60
Gambar 3.6. Flowchart Pembuatan Node .....	61
Gambar 3.7. Pelabelan Leaf Node menggunakan Cost Sensitive Learning .....	62
Gambar 4.1. Bisnis Proses Penerimaan Calon Nasabah Asuransi .....	66
Gambar 4.2. Elemen input dan output Sibah Asuransi .....	75
Gambar 4.3. Use Case Diagram Sibah Asuransi.....	76
Gambar 4.5. Class Diagram Sibah Asuransi.....	132
Gambar 4.6. Entity Relationship Diagram (ERD) Sibah Asuransi.....	133
Gambar 4.7. Listing Program index() .....	140
Gambar 4.8. Listing Program bagidataset() .....	141
Gambar 4.9. Listing Program reset().....	141
Gambar 4.10. Listing Program insert_atribut().....	141
Gambar 4.11. Listing Program c45() .....	143
Gambar 4.12. Diagram Alir index().....	143
Gambar 4.13. Diagram Alir bagidataset().....	143

Gambar 4.14. Diagram Alir reset()	143
Gambar 4.15. Diagram Alir insert_atribut()	143
Gambar 4.16. Diagram Alir c45()	144
Gambar 5.1. Pohon Iterasi 1	154
Gambar 5.2. Pohon Iterasi 2	156
Gambar 5.3. Pohon Iterasi 3	160
Gambar 5.4. Pohon Iterasi 4	164
Gambar 5.5. Pohon Iterasi 5	168
Gambar 5.6. Iterasi Pohon 6	174
Gambar 5.7. Pohon Iterasi 7	178
Gambar 5.8. Hasil Pohon	181
Gambar 5.9. Tampilan Menu Data Training	182
Gambar 5.10. Tampilan Menu Data Testing	182
Gambar 5.11. Form Modal Bobot Cost Function	183
Gambar 5.12. Tampilan Proses Pembangunan Classification tree	184
Gambar 5.13. Pembagian Data Set	185
Gambar 5.14. Kode Program Proses Reset Data	185
Gambar 5.15. Kode Program Proses Menyiapkan Data Atribut	185
Gambar 5.16. Kode Program Proses Perhitungan Jumlah Tiap Nilai	186
Gambar 5.17. Kode Program Proses Perhitungan Entropy	186
Gambar 5.18. Kode Program Proses Perhitungan Gain	187
Gambar 5.19. Kode Program Proses Pengambilan Gain Tertinggi	187
Gambar 5.20. Kode Program Proses Penentuan Kondisi Atribut	187
Gambar 5.21. Kode Program Proses Memasukkan Atribut ke dalam Tabel Pohon	188
Gambar 5.22. Kode Program Proses Mengambil Atribut yang Belum Memiliki Keputusan	188
Gambar 5.23. Kode Program Proses Mengosongkan Tabel Perhitungan	189
Gambar 5.24. Kode Program Proses Menghapus Atribut dengan Gain Tertinggi	189
Gambar 5.25. Kode Program Proses Perhitungan Kembali C45	189

Gambar 5.26. Kode Program Proses Pelabelan Leaf Node .....	190
Gambar 5.27. Kode Program Proses Pembentukan Pohon .....	190
Gambar 5.28. Tampilan Sistem Hasil Pembangunan Classification tree.....	191
Gambar 5.29. Hasil Pembangunan <i>Classification tree</i> .....	193
Gambar 5.30. Kode Program Proses Konversi Decision ke dalam Bentuk Rule .....	193
Gambar 5.31. Tampilan Sistem Rule .....	194
Gambar 5.32. Rule .....	196
Gambar 5.33. Fitur Login.....	197
Gambar 5.34. Fitur Register.....	197
Gambar 5.35. Dashboard Aktor Admin .....	198
Gambar 5.36. Dashboard Aktor Calon Nasabah.....	198
Gambar 5.37. Dashboard Aktor Agent .....	199
Gambar 5.38. Dashboard Aktor Leader .....	199
Gambar 5.39. Fitur mengelola Data Set.....	200
Gambar 5.40. Modal Form Tambah Data Set .....	200
Gambar 5.41. Fitur Melihat Data Training .....	201
Gambar 5.42. Fitur Melihat Classification tree.....	202
Gambar 5.43. Modal Form Bobot Cost Function .....	202
Gambar 5.44. Proses Pembangunan Classification tree.....	203
Gambar 5.45. Fitur Melihat Rule .....	203
Gambar 5.46. Fitur Melihat Data Testing .....	204
Gambar 5.47. Fitur Melihat Hasil Uji Klasifikasi - Perbandingan Class Aktual dan Klasifikasi .....	205
Gambar 5.48. Fitur Melihat Hasil Uji Klasifikasi - Hasil Confusion matrix .....	205
Gambar 5.49. Halaman Daftar Agent .....	206
Gambar 5.50. Modal Form Mendaftar Agent .....	207
Gambar 5.51. Halaman Data Calon Nasabah.....	209
Gambar 5.52. Modal Form Tambah Calon Nasabah .....	210
Gambar 5.53 Modal Form Ubah Calon Nasabah.....	211



Gambar 5.54. Halaman Detail Calon Nasabah .....	212
Gambar 5.55. Tampilan Hasil Klasifikasi.....	212
Gambar 5.56. Kode Program Klasifikasi Data Testing .....	214
Gambar 5.57. Kode Program Hitung Frekuensi Class .....	214
Gambar 5.58. Perbandingan Class Aktual dan Klasifikasi .....	215
Gambar 5.59. Kode Program Confusion Matrix.....	215
Gambar 5.60. Perhitungan <i>Confusion Matrix</i> Pengujian ke-1 .....	217
Gambar 5.61. Perhitungan <i>Confusion Matrix</i> Pengujian ke-2 .....	218
Gambar 5.62. Perhitungan <i>Confusion Matrix</i> Pengujian ke-3 .....	219
Gambar 5.63. Perhitungan <i>Confusion Matrix</i> Pengujian ke-1 .....	221
Gambar 5.64. Perhitungan <i>Confusion Matrix</i> Pengujian ke-2 .....	222
Gambar 5.65. Perhitungan <i>Confusion Matrix</i> Pengujian ke-3 .....	223

## BAB 1. PENDAHULUAN

Bab ini merupakan bab awal dari laporan tugas akhir. Pada bab ini akan dibahas tentang latar belakang, rumusan masalah, tujuan dan manfaat, batasan masalah, dan sistematika penulisan.

### 1.1 Latar Belakang

Kebutuhan selalu mengiringi kehidupan manusia. Salah satu kebutuhan manusia menurut tingkat kepentingannya dibagi menjadi tiga jenis, yaitu kebutuhan primer, sekunder, dan tersier. Kebutuhan primer yakni kebutuhan pokok yang harus terpenuhi pertama kali oleh manusia, yaitu pakaian, makanan, dan tempat tinggal. Kebutuhan sekunder merupakan kebutuhan tambahan yang melengkapi kebutuhan primer dimana kebutuhan sekunder muncul ketika kebutuhan primer sudah terpenuhi, sedangkan kebutuhan tersier merupakan kebutuhan yang disebut dengan kebutuhan akan sesuatu yang berhubungan dengan kemewahan. Jenis kebutuhan diatas bersifat relatif satu orang dan yang lainnya, yang artinya setiap orang memiliki pandangan berbeda pada suatu barang apakah termasuk kebutuhan primer, sekunder, atau tersier bagi dirinya. Sebagai contoh asuransi dahulu dianggap sebagai kebutuhan tersier karena dibutuhkan hanya jika terjadi sesuatu pada dirinya ataupun hanya digunakan untuk masa depan padahal masa sekarang masih banyak kebutuhan yang harus terpenuhi. Namun saat ini asuransi dapat dianggap sebagai kebutuhan primer untuk sebagian orang. Hal ini dapat terlihat banyaknya orang yang memiliki asuransi baik dikalangan bawah maupun atas. Seiring dengan dianggapnya asuransi sebagai kebutuhan primer, setiap orang berusaha untuk memenuhi akan kebutuhan asuransi tersebut.

Asuransi merupakan salah satu cara untuk memproteksi diri di masa depan. Kemungkinan yang terjadi merupakan ketidakpastian yang merugikan sebagai akibat suatu peristiwa tidak terduga, oleh karena itu proteksi asuransi akan merubah kerugian

menjadi ganti rugi atau santunan klaim dengan syarat pembayaran premi. Premi merupakan pendapatan bagi perusahaan asuransi yang jumlahnya ditentukan dalam suatu persentase atau tarif tertentu dari jumlah yang dipertanggungkan. Pendapatan premi untuk perusahaan asuransi didapat dari jumlah premi yang dibayar oleh nasabah.

Segala suatu aktivitas apapun selalu terdapat permasalahan, begitu pula dalam penggunaan jasa asuransi pasti terdapat adanya suatu resiko. Hal inilah yang menjadi bahan pertimbangan penyedia jasa asuransi dalam pemberian asuransi yang akan diberikan kepada calon nasabah. Permasalahan yang sering timbul dalam perusahaan asuransi adalah adanya nasabah yang menunggak dalam membayar premi sehingga mempengaruhi keuangan perusahaan dan tentunya merugikan perusahaan asuransi, oleh karena itu diperlukan sebuah sistem yang dapat memprediksi calon nasabah mana yang masuk ke dalam kelompok lancar dan nasabah mana yang masuk ke dalam kelompok tidak lancar dalam membayar iuran premi agar pihak asuransi dapat mengatasi sejak dini permasalahan tersebut.

Berdasarkan pemaparan permasalahan diatas, maka dibutuhkan suatu sistem yang dapat digunakan untuk membantu proses penerimaan calon nasabah yang tepat. Adanya jumlah data *history* transaksi yang dimiliki perusahaan asuransi, tentunya dapat dimanfaatkan untuk membantu proses penerimaan calon nasabah yang tepat. Dalam hal ini, peneliti ingin memberikan kontribusi terhadap keoptimalan dengan menghitung calon nasabah yang tepat tersebut menggunakan *data mining*.

Pada penelitian ini, *data mining* digunakan untuk proses klasifikasi. *Data mining* merupakan sebuah proses pencarian data secara otomatis pada *database* berukuran besar yang menghasilkan informasi. Ada beberapa algoritma dalam klasifikasi, diantaranya *naive bayes*, *k-nearest neighbour*, dan *classification tree*. Berdasarkan penelitian yang pernah dilakukan oleh (Sartika & Sensuse, 2017) mengenai perbandingan algoritma klasifikasi *naive bayes*, *k-nearest neighbour*, dan *decision tree* pada studi kasus pengambilan keputusan pemilihan pola pakaian, hasil perbandingan menunjukkan bahwa metode *decision tree* memiliki tingkat akurasi tertinggi dibandingkan algoritma *naive bayes* dan *k-nearest neighbour* yaitu mencapai 75.6%.

Penggunaan *data mining*, calon nasabah dapat dikategorikan ke dalam nasabah lancar atau tidak lancar dengan lebih selektif dan dalam waktu yang lebih singkat dibanding menggunakan cara pencarian manual. Hal tersebut tentunya akan sangat membantu pihak perusahaan asuransi dalam memberikan keputusan pemberian polis asuransi sehingga meminimalisir terjadinya premi tidak lancar. Metode *data mining* yang digunakan dalam penelitian ini yaitu metode klasifikasi *C4.5 algorithm* dan *cost sensitive learning*.

*Classification tree C4.5* merupakan algoritma yang digunakan untuk membentuk pohon klasifikasi (*classification tree*). Split atribut merupakan proses utama dalam pembentukan pohon klasifikasi (*classification tree*) di *classification tree C4.5 Algorithm*. Kelebihan dari algoritma ini adalah dapat menghasilkan pohon klasifikasi yang mudah diinterpretasikan, memiliki tingkat akurasi yang dapat diterima, efisien dalam menangani atribut bertipe diskret dan numerik (Han, Kamber, & Pei, 2001). Berdasarkan penelitian yang dilakukan oleh (Amin, Indwiarti, & Sibaroni, 2015) mengenai penggunaan *classification tree* dalam pengambilan keputusan permohonan kredit, diketahui nilai *precision* dicapai oleh algoritma C4.5 nilai sebesar 78,08 % dan nilai *recall* sebesar 96,4 %. Dari hasil data latih yang sama, ID3 menghasilkan *precision* sebesar 71,51% dan *recall* sebesar 92,09%, maka dapat disimpulkan bahwa pada kasus ini algoritma C4.5 memiliki tingkat akurasi yang tinggi dan lebih baik dari ID3. Tetapi, pada proses pemilihan split atribut di *classification tree C4.5 algorithm* belum dapat mengatasi *misclassification cost* di setiap split sehingga berpengaruh pada kinerja pengklasifikasi (Rahayu, Wahono, & Supriyanto, 2015), oleh karena itu *cost sensitive learning* digunakan untuk mengatasi masalah tersebut.

*Cost sensitive learning* merupakan salah satu teknik pembuatan *model* prediksi yang dalam prosesnya juga memperhitungkan *misclassification cost*. *Misclassification cost* merupakan kerugian (*cost*) yang timbul apabila suatu data salah diklasifikasikan (Tan, Steinbach, & Kumar, 2005), seperti pada pengklasifikasian calon nasabah asuransi. Pada hasil klasifikasi, calon nasabah diklasifikasikan ke dalam *class* tidak lancar, tetapi pada kenyataannya calon nasabah termasuk kategori lancar, sedangkan

nasabah yang diklasifikasikan kedalam *class* tidak lancar, pada kenyataannya termasuk kategori nasabah yang lancar dalam pembayaran premi. Penelitian mengenai penerapan *cost sensitive learning* pernah dilakukan oleh (Wicaksono, 2010) dengan judul pembangunan *model* prediksi *defect* pengembangan perangkat lunak menggunakan metode *ensemble classification tree* dan *cost sensitive learning*. Pada jurnal tersebut, untuk memperkecil kerugian yang ditimbulkan oleh kesalahan prediksi, maka pada proses pembuatan *classification tree* digunakan metode *cost sensitive learning*. Hasil uji coba menunjukkan bahwa, metode yang diajukan memiliki *accuracy* dan *recall* yang lebih baik dari *action-based defect prevention* (ABDP). *Accuracy* dan *recall* yang terbaik *dataset I* adalah 100% dan 100% sedangkan untuk *dataset II* adalah 98% dan 100%. Hasil dari penelitian sebelumnya diharapkan mampu mendukung dalam pengembangan sistem informasi pengklasifikasian penerimaan calon nasabah asuransi sehingga memberikan kemudahan dalam penerimaan calon nasabah yang tepat (lancar dan tidak lancar dalam pembayaran premi) pada perusahaan asuransi ini.

## 1.2 Rumusan Masalah

Berdasarkan uraian yang telah disampaikan dalam latar belakang, terdapat beberapa permasalahan yang harus diselesaikan dalam penulisan ini, yaitu:

1. Bagaimana memprediksi kelancaran pembayaran premi calon nasabah asuransi?
2. Bagaimana meminimalisir *misclassification cost* dalam klasifikasi penerimaan calon nasabah asuransi?
3. Bagaimana merancang dan membangun sistem informasi pengklasifikasian penerimaan calon nasabah asuransi?

### 1.3 Tujuan dan Manfaat

Berikut merupakan tujuan yang ingin dicapai dan manfaat yang ingin diperoleh dalam penelitian ini.

#### 1.3.2 Tujuan

Tujuan dari penulisan ini merupakan jawaban dari perumusan masalah yang telah disebutkan. Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Memprediksi kelancaran pembayaran premi calon nasabah asuransi menggunakan algoritma *classification tree C4.5* dan *cost sensitive learning*.
2. Meminimalisir *misclassification cost* dalam klasifikasi penerimaan calon nasabah asuransi menggunakan *cost sensitive learning*.
3. Merancang dan membangun sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* dengan metode pengembangan *SDLC waterfall*.

#### 1.3.2 Manfaat

Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Manfaat akademis

Hasil penelitian ini diharapkan dapat memberikan kontribusi dan masukan bagi siapa saja yang membutuhkan informasi yang berhubungan dengan judul penelitian ini. Selain itu, hasil penelitian ini merupakan suatu upaya untuk menambah varian judul penelitian yang ada di Program Studi Sistem Informasi Fakultas Ilmu Komputer Universitas Jember.

2. Manfaat bagi peneliti
  - 1) Mengetahui cara mengklasifikasi calon nasabah yang tepat menggunakan metode *classification tree C4.5 algorithm* dan *cost sensitive learning*.
  - 2) Melatih kemampuan dan menerapkan ilmu pengetahuan yang telah didapat di saat perkuliahan untuk membantu pemerintah, perusahaan atau masyarakat umum
3. Manfaat bagi objek penelitian  
Membantu instansi untuk melakukan penerimaan calon nasabah asuransi yang tepat secara optimal.

#### 1.4 Batasan Masalah

Agar tidak terjadi penyimpangan dalam proses penelitian dan pembangunan sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*, maka ditetapkan beberapa batasan permasalahan sebagai berikut:

- a. Algoritma yang digunakan dalam pengklasifikasian adalah *classification tree C4.5* dan *cost sensitive learning* tanpa membandingkannya dengan algoritma lain.
- b. Data sampel daftar nasabah asuransi terdahulu yang digunakan dalam penelitian ini adalah data sampel yang didapat dari Perusahaan Asuransi XYZ Cabang Jember.
- c. Klasifikasi hanya dilakukan untuk menentukan kelancaran calon nasabah dalam pembayaran premi.
- d. Atribut yang digunakan untuk menentukan calon nasabah asuransi yang tepat yaitu usia, status marital, jenis kelamin, penghasilan, premi, cara bayar, rencana pembayaran, dan masa pembebanan biaya.
- e. Tingkat ketepatan terdiri dari 2 yaitu lancar dan tidak lancar.

- f. *Cost sensitive learning* yang digunakan yaitu *cost function* sebagai kriteria untuk pelabelan *leaf node*.
- g. Pada penelitian ini tidak dilakukan pemeliharaan sistem.
- h. Sistem ini berbasis *website*.

### 1.5 Sistematika Penulisan

Adapun sistematika penulisan skripsi ini adalah sebagai berikut:

#### 1. Pendahuluan

Bab ini memuat uraian tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah, dan sistematika penulisan skripsi yang masing-masing diuraikan dalam subbab tersendiri.

#### 2. Tinjauan Pustaka

Bab ini memaparkan tinjauan terhadap hasil-hasil penelitian terdahulu berkaitan dengan masalah yang dibahas, landasan materi dan konsep penerimaan calon nasabah asuransi, dan kajian teori metode analisis data yang berkaitan dengan masalah dalam penelitian.

#### 3. Metodologi Penelitian

Bab ini menguraikan tentang jenis penelitian, tempat dan waktu penelitian, studi literatur, pengumpulan data, implementasi metode, pengujian algoritma, analisis hasil uji coba sistem, dan penarikan kesimpulan.

#### 4. Pengembangan Sistem

Bab ini berisi uraian tentang tentang proses pengembangan sistem. Pengembangan sistem dimulai dari analisis kebutuhan fungsional dan non-fungsional sistem, kemudian merancang *business process*, *usecase diagram*, *scenario*, *activity diagram*, *sequence diagram*, *class diagram* dan *entity relationship diagram* (ERD), kemudian penulisan kode program, lalu proses pengujian sistem.

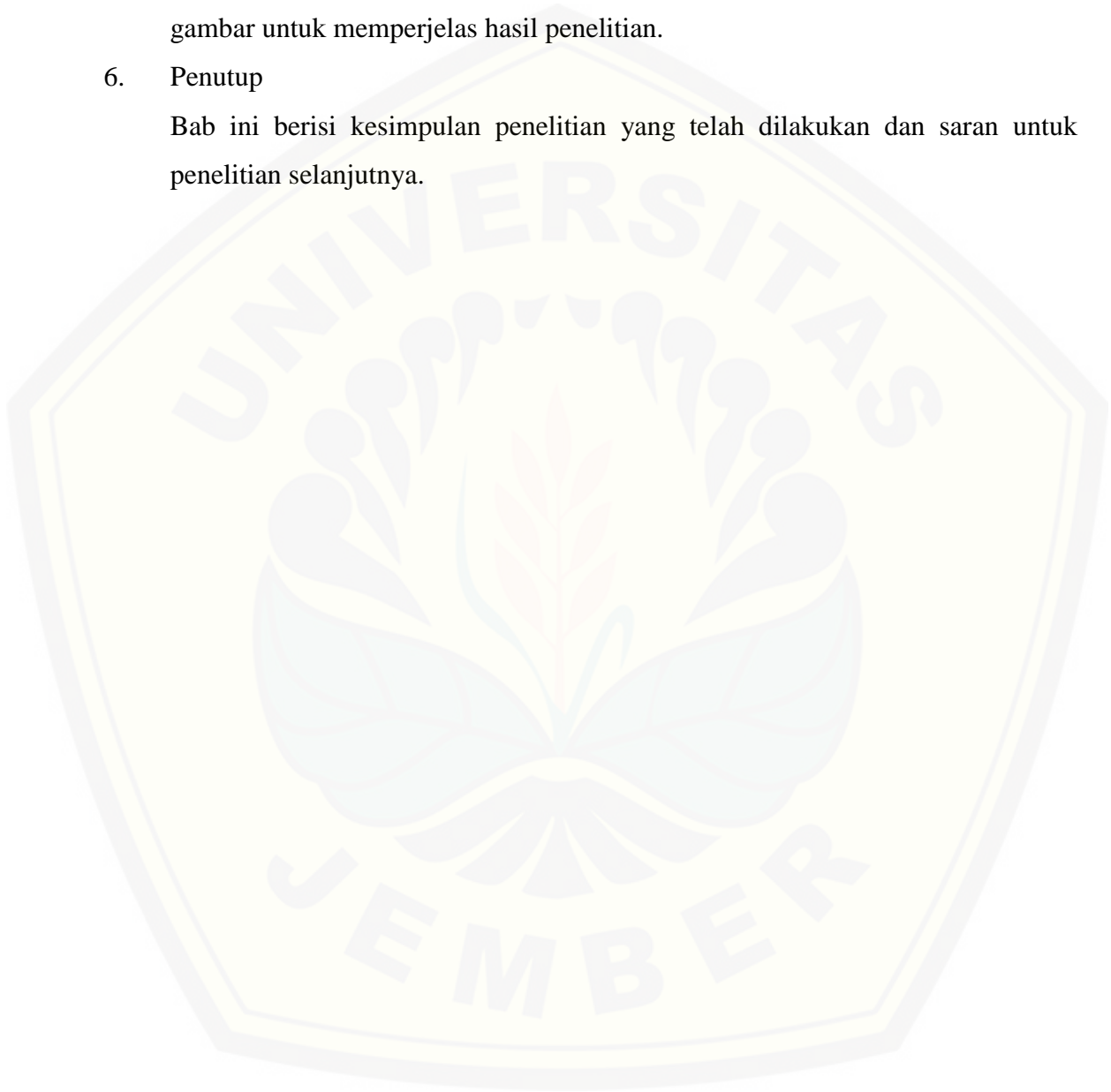


5. Hasil dan Pembahasan

Bab ini memaparkan secara rinci pemecahan masalah melalui analisis yang disajikan dalam bentuk deskripsi dibantu dengan ilustrasi berupa tabel dan gambar untuk memperjelas hasil penelitian.

6. Penutup

Bab ini berisi kesimpulan penelitian yang telah dilakukan dan saran untuk penelitian selanjutnya.



## BAB 2. TINJAUAN PUSTAKA

Pada bagian ini dipaparkan tinjauan yang berkaitan dengan masalah yang dibahas, kajian teori yang berkaitan dengan masalah, dan juga penelitian-penelitian terdahulu.

### 2.1 Penelitian Terdahulu

Peneliti memaparkan empat hasil penelitian terdahulu yang relevan dengan permasalahan yang akan diteliti. Penelitian tersebut diantaranya tentang: (i) Perbandingan algoritma klasifikasi *naive bayes*, *nearest neighbour*, dan *classification tree* pada studi kasus pengambilan keputusan pemilihan pola pakaian (Sartika & Sensuse, 2017); (ii) Implementasi klasifikasi *classification tree* dengan algoritma C4.5 dalam pengambilan keputusan permohonan kredit oleh debitur (Studi Kasus: Bank Pasar Daerah Istimewa Yogyakarta) (Amin, Indwiarti, & Sibaroni, 2015); (iii) Pembangunan *model* prediksi *defect* pengembangan perangkat lunak menggunakan metode *ensemble classification tree* dan *cost sensitive learning Invalid source specified.*; (iv) Penerapan algoritma C4.5 untuk penentuan jurusan mahasiswa (Swastina, 2013). Penjelasan detail mengenai keempat penelitian tersebut disajikan pada .

Tabel 2.1. Penelitian Terdahulu

1.	<b>Judul</b>	Perbandingan Algoritma Klasifikasi <i>Naive Bayes</i> , <i>Nearest Neighbour</i> , dan <i>Classification tree</i> pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian
	<b>Penulis</b>	Dewi Sartika dan Dana Indra Sensuse
	<b>Tahun</b>	2017
	<b>Analisis</b>	Penelitian ini mengkaji mengenai perbandingan algoritma antara algoritma <i>naive bayes</i> , <i>nearest neighbour</i> , dan <i>classification tree</i> pada studi kasus pengambilan keputusan pemilihan pola pakaian. Penulis menjelaskan bagaimana pendekatan yang dapat digunakan untuk menganalisis sekumpulan data dengan menggunakan klasifikasi. Penulis mencantumkan hasil penelitiannya berbentuk bab-bab berdasarkan algoritma-algoritma yang digunakan. Pada setiap algoritma, penulis

		mencantumkan bagaimana proses algoritmanya. Setelah menjelaskan tiap algoritma, penulis kemudian membandingkan ketiga algoritma yang digunakan menggunakan perangkat lunak WEKA ( <i>Waikato Environment for Knowledge Analysis</i> ) dengan menggunakan beberapa kali pengujian dengan data <i>training</i> sebesar 90%, 80%, 70%, dan 60% dari data set.
	<b>Hasil</b>	Hasil penelitian mengenai perbandingan algoritma klasifikasi <i>nearest neighbour</i> , <i>naive bayes</i> , dan <i>classification tree</i> yang digunakan pada studi kasus pengambilan keputusan pemilihan pola pakaian menunjukkan bahwa algoritma klasifikasi <i>classification tree</i> merupakan algoritma klasifikasi yang memiliki tingkat akurasi paling tinggi dibandingkan algoritma klasifikasi <i>naive bayes</i> dan <i>nearest neighbour</i> yaitu mencapai 75.6% pada pengujian yang dilakukan dengan menggunakan mode pengujian percentage split. Sedangkan pada algoritma <i>naive bayes</i> memiliki akurasi sebesar 41.5% dan <i>nearest neighbor</i> memiliki akurasi sebesar 22.5%
2.	<b>Judul</b>	Implementasi Klasifikasi <i>Classification tree</i> Dengan Algoritma C4.5 Dalam Pengambilan Keputusan Permohonan Kredit Oleh Debitur (Studi Kasus: Bank Pasar Daerah Istimewa Yogyakarta)
	<b>Penulis</b>	Rafik Khairul Amin, Dra.Indwiarti, M.Si., dan Yuliant Sibaroni, S.Si., M,T
	<b>Tahun</b>	2015
	<b>Analisis</b>	Penelitian ini bertujuan untuk tujuan menganalisis kinerja algoritma pohon klasifikasi C4.5 pada identifikasi kelayakan kredit oleh debitur. Pada penelitian ini, peneliti ingin meneliti kondisi calon debitur yang diperkirakan dapat mempengaruhi kemampuan mereka dalam memenuhi kewajiban kepada bank. Dalam hal ini, peneliti melakukan pengumpulan data-data tentang calon debitur baik yang kuantitatif seperti data keuangan, maupun kualitatif seperti penilaian terhadap pengelolaan perusahaan dan sebagainya. Kemudian data-data ini akan diolah dan diproses sesuai prosedur pada bank tersebut sebelum akhirnya diambil keputusan apakah layak untuk memperoleh pinjaman kredit dari bank. Berdasarkan beberapa penelitian sebelumnya, peneliti menggunakan algoritma C4.5 dalam kasus dengan data yang berisi atribut dan <i>record</i> yang berbeda yaitu persetujuan penerimaan kredit di Bank dengan atribut berupa karakteristik debitur yang mempunyai <i>record</i> sesuai dengan historis yang telah tersimpan pada data bank.
	<b>Hasil</b>	Penelitian ini menunjukkan bahwa nilai <i>precision</i> terbesar yang dicapai oleh algoritma C4.5 sebesar 78.08% dengan nilai <i>recall</i> sebesar 96.4%. Sedangkan dari data <i>training</i> yang sama namun

		menggunakan algoritma ID3 memiliki nilai <i>precision</i> sebesar 71.51% dengan nilai <i>recall</i> sebesar 92.09%. Hasil akhir dari penelitian ini membuktikan bahwa pada kasus ini algoritma C4.5 memiliki tingkat akurasi yang tinggi dan lebih baik dari ID3.
3.	<b>Judul</b>	Pembangunan <i>Model</i> Prediksi <i>Defect</i> Pengembangan Perangkat Lunak Menggunakan Metode <i>Ensemble Classification tree</i> dan <i>Cost Sensitive Learning</i>
	<b>Penulis</b>	Satrio Agung Wicaksono, Daniel Oranova S., dan Sarwosri
	<b>Tahun</b>	2010
	<b>Analisis</b>	Penelitian ini peneliti melakukan pengembangan perangkat lunak dengan melakukan tindakan pencegahan munculnya <i>high defect</i> . Terdapat beberapa pendekatan untuk mencegah munculnya <i>high defect</i> . Salah satu pendekatan adalah <i>action-based defect prevention</i> (ABDP). Namun <i>model</i> prediksi ABDP dapat melakukan kesalahan prediksi yang dapat dibagi menjadi dua tipe. Tipe 1 adalah memprediksi aksi yang sebenarnya menghasilkan <i>low defect</i> sebagai <i>high defect</i> . Tipe 2 adalah memprediksi aksi yang sebenarnya menghasilkan <i>high defect</i> sebagai <i>low defect</i> . Kesalahan tipe 2 menimbulkan <i>cost</i> (kerugian) lebih besar dari tipe 1. Jadi untuk meningkatkan akurasi prediksi dari <i>model</i> prediksi, peneliti membangun dengan <i>ensemble method</i> yang menggunakan banyak 2 <i>classification tree</i> . Untuk memperkecil kerugian yang ditimbulkan oleh kesalahan prediksi dan menekan munculnya kesalahan prediksi tipe 2, peneliti melakukan proses pembuatan <i>classification tree</i> menggunakan metode <i>cost sensitive learning</i> .
	<b>Hasil</b>	Penelitian ini menunjukkan bahwa metode yang diajukan memiliki <i>accuracy</i> dan <i>recall</i> yang lebih baik dari ABDP. <i>Accuracy</i> dan <i>recall</i> yang terbaik dataset I adalah 100% dan 100% sedangkan untuk dataset II adalah 98% dan 100%
4.	<b>Judul</b>	Penerapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa
	<b>Penulis</b>	Liliana Swastina
	<b>Tahun</b>	2013
	<b>Analisis</b>	Penelitian ini dilakukan untuk mengkaji mengenai penentuan jurusan dalam bidang studi yang akan diambil oleh mahasiswa menggunakan pendekatan algoritma klasifikasi <i>data mining</i> . Penelitian ini dilakukan agar mahasiswa tidak salah dalam memilih jurusan yang akan di tempuh selama belajar pada perguruan tinggi. Kecenderungan yang saat ini terjadi yaitu banyaknya siswa kelas XII yang tidak tahu minatnya dan bakatnya serta akan memilih jurusan apa ketika akan kuliah nanti akan membuat efek buruk yang terjadi seperti keengganan belajar dan menurunnya kualitas serta prestasi akademik karena

	<p>merasa salah dalam memilih jurusan Pemilihan jurusan yang tidak sesuai dengan kemampuan, kepribadian, minat dan bakat dapat mempengaruhi mahasiswa dalam mengikuti perkuliahan. Oleh karena itu, peneliti menggunakan algoritma C4.5 digunakan untuk menentukan jurusan yang akan diambil oleh mahasiswa sesuai dengan data siswa seperti latar belakang, minat, dan kemampuan. Parameter pemilihan jurusan adalah Indeks Prestasi Kumulatif Semester 1 dan 2.</p>
<b>Hasil</b>	<p>Penelitian ini menunjukkan bahwa algoritma <i>classification tree</i> C4.5 akurat diterapkan untuk penentuan kesesuaian jurusan mahasiswa dengan pembagian data <i>set</i> sebesar 90% data <i>training</i> dan 10% data <i>testing</i>, yang tingkat keakuratannya sebesar 93,31 % dan akurasi rekomendasi jurusan sebesar 82,64%. Dengan adanya penelitian ini, peneliti mengharapkan dapat memberikan solusi bagi mahasiswa dan dapat membantu mahasiswa dalam menentukan jurusan yang sesuai yang akan ditempuh oleh mahasiswa selama studi sehingga peluang untuk sukses dalam studi di perguruan tinggi.</p>

## 2.2 Asuransi

Asuransi adalah alat untuk mengurangi risiko dengan menggabungkan sejumlah unit yang menyebabkan kerugian guna mengumpulkan taksiran yang mungkin terjadi (Hartono, 1995). Bentuk-bentuk asuransi selalu akan muncul dan ditawarkan sebagai salah satu jasa produksi dari perusahaan asuransi. Jasa pokok yang ditawarkan oleh asuransi adalah rasa aman, rasa terlindungi karena sudah adanya janji dengan pihak penanggung kepada tertanggung, apabila ia menderita suatu kerugian akan mendapat ganti kerugian. Dalam hal ini asuransi dapat dikatakan merupakan benteng pertama untuk menghadapi risiko, atau dapat pula lebih tegas dikatakan bahwa asuransi merupakan satu-satunya alat menghadapi atau pelawan risiko.

Suatu perusahaan asuransi menggunakan beberapa variabel untuk mengklasifikasikan data calon nasabah asuransi untuk dikategorikan ke dalam kelompok lancar maupun tidak lancar dalam pembayaran premi yang nantinya dapat digunakan untuk membantu perusahaan dalam proses penerimaan nasabah asuransi.

### 2.3 Sistem Informasi

Sistem informasi merupakan suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan (Jogiyanto, 2005). Sedangkan definisi sistem informasi menurut Alter, 1992 adalah kombinasi antara prosedur kerja, informasi, orang dan teknologi informasi yang diorganisasikan untuk mencapai tujuan dalam sebuah organisasi (Kadir, 2014).

Menurut Robert A. Leith dan K. Roscoe Davis dalam (Jogiyanto, 2005) sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial dan kegiatan strategi dari suatu organisasi dan menyediakan pihak luar tertentu dengan laporan-laporan yang diperlukan.

Dari definisi-definisi tersebut dapat disimpulkan bahwa sistem informasi adalah suatu sistem yang digunakan untuk mengolah transaksi-transaksi bersifat manajerial yang membutuhkan kombinasi antara prosedur kerja, informasi, manusia dan teknologi dalam pembuatan laporan-laporan atau dokumentasi.

### 2.4 Data Mining

*Data Mining* adalah suatu istilah yang digunakan untuk menguraikan penemuan pengetahuan di dalam *database*. *Data mining* adalah proses yang menggunakan teknik statistik, matematika, kecerdasan buatan dan *machine learning* untuk mengekstraksi dan mengidentifikasi informasi yang bermanfaat dan pengetahuan yang terakit dari berbagai *database* besar (Turban, Aronson, & Liang, 2005). Menurut Gartner Group dalam (Larose, 2014) *data mining* adalah suatu proses menemukan hubungan yang berarti, pola, dan kecenderungan dengan memeriksa dalam sekumpulan besar data yang tersimpan dalam penyimpanan dengan menggunakan teknik pengenalan pola seperti teknik statistik dan matematika. Dari beberapa definisi di atas dapat ditarik

kesimpulan bahwa *data mining* merupakan proses ataupun kegiatan untuk mengumpulkan data yang berukuran besar kemudian mengekstraksi data tersebut menjadi informasi – informasi yang nantinya dapat digunakan.

Beberapa teknik dan sifat dalam mengolah data di dalam *data mining* adalah sebagai berikut:

1. *Classification* (Klasifikasi)

Menentukan sebuah *record* data baru ke salah satu dari beberapa kategori yang telah didefinisikan sebelumnya.

2. *Clustering* (Klustering)

Membagi *data-set* menjadi beberapa *sub-set* atau kelompok sedemikian rupa sehingga elemen-elemen dari suatu kelompok tertentu memiliki *set* property yang dishare bersama, dengan tingkat similaritas yang tinggi dalam suatu kelompok dan tingkat similaritas antar kelompok yang rendah.

3. *Regression* (Regresi)

Memprediksi nilai dari suatu *variable* kontinu yang diberikan berdasarkan nilai dari *variable* yang lain, dengan mengasumsikan sebuah *model* ketergantungan linier atau non-linier.

4. *Association Rule Discovery* (Kaidah Asosiasi)

Mendeteksi kumpulan atribut-atribut yang muncul bersamaan dalam frekuensi yang sering, dan membentuk sejumlah kaidah dari kumpulan-kumpulan tersebut.

5. *Sequential Pattern Discovery* (Pencarian Pola Sekuensial)

Mencari sejumlah *event* yang secara umum terjadi bersama-sama. Pola-pola sekuensial pertama, pada dasarnya dibentuk dengan cara mencari semua kemungkinan pola yang ada.

## 2.5 Klasifikasi

Klasifikasi menurut (Bertalya, 2009) adalah suatu proses untuk menyatakan suatu objek ke salah satu kategori yang sudah didefinisikan sebelumnya. Proses

pembelajaran fungsi target (model klasifikasi) yang memetakan setiap sekumpulan atribut  $x$  (input) ke salah satu kelas  $y$  yang didefinisikan sebelumnya. Klasifikasi adalah proses untuk menemukan *model* atau fungsi yang menjelaskan atau membedakan konsep atau kelas data, dengan tujuan untuk dapat memperkirakan kelas dari suatu objek yang labelnya tidak diketahui.

Pada awalnya, kegiatan klasifikasi dokumen dilakukan secara manual oleh seorang ahli. Untuk melakukan klasifikasi, seorang ahli harus menyediakan waktu untuk membaca isi setiap dokumen baru dan membandingkan isinya dengan skema kategori yang tersedia. Dengan demikian klasifikasi ini merupakan kegiatan yang menghabiskan waktu dan biaya. Selain alasan yang telah sebelumnya ditambah dengan peningkatan jumlah dokumen telah mendorong berkembangnya metode pengklasifikasian secara otomatis yang dapat melakukan klasifikasi dengan cara belajar dari sekumpulan contoh dokumen yang telah diklasifikasikan sebelumnya. Adanya metode tersebut, keuntungan yang diperoleh adalah penghematan tenaga kerja dan efektivitas.

## 2.6 *Decision Tree*

*Decision tree* merupakan metode yang ada pada teknik klasifikasi dalam *data mining*. *Decision tree* merupakan sebuah teknik klasifikasi yang menggunakan *tree* untuk membangun model prediksinya (Tan, Steinbach, & Kumar, 2005). Metode ini digunakan untuk memprediksi nilai diskrit dari fungsi target, yang mana fungsi pembelajaran dipresentasikan oleh sebuah *classification tree*.

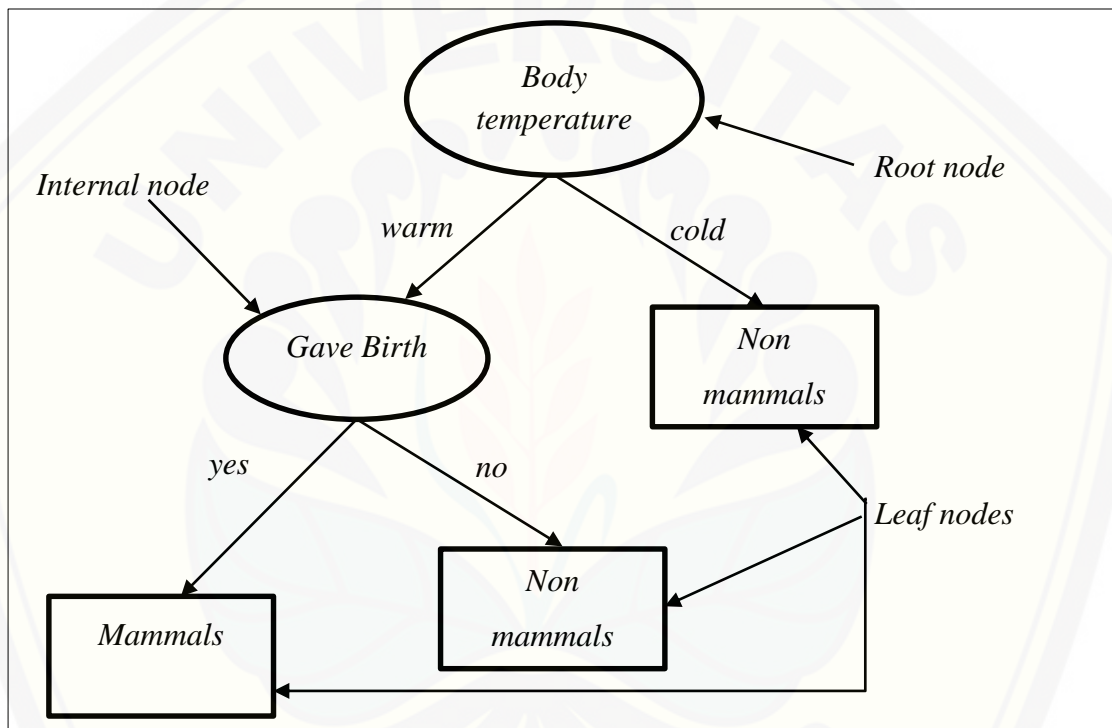
Terdapat 3 jenis *node* yang terdapat pada *classification tree*, yaitu (Tan, Steinbach, & Kumar, 2005):

- a. *Root node*, merupakan *node* paling atas, pada *node* ini tidak ada *input* dan bisa tidak mempunyai *output* atau mempunyai *output* lebih dari satu.



- b. *Internal node*, merupakan *node* percabangan. Pada *node* ini terdapat percabangan. Pada *node* ini terdapat satu *input* dan memiliki *output* minimal dua.
- c. *Leaf node* atau *terminal node*, merupakan *node* akhir. Pada *node* ini terdapat satu *input* dan tidak mempunyai *output*.

3 jenis *node* yang terdapat pada *classification tree* tersebut dapat dilihat pada Gambar 2.1.



Gambar 2.1. Node pada Classification Tree

## 2.7 C4.5 Algorithm

C4.5 algorithm merupakan algoritma yang digunakan untuk membentuk pohon klasifikasi. Pohon klasifikasi merupakan metode klasifikasi dan prediksi yang sangat kuat dan terkenal. Metode pohon klasifikasi mengubah fakta yang sangat besar menjadi pohon klasifikasi yang merepresentasikan aturan. Aturan dapat dengan mudah dipahami dengan bahasa alami dan mereka juga dapat diekspresikan dalam bentuk

bahasa basis data seperti *Structured Query Language* untuk mencari *record* pada kategori tertentu. Pohon klasifikasi juga berguna untuk mengeksplorasi data, menemukan hubungan tersembunyi antara sejumlah calon variabel input dengan sebuah variabel target (Kusrini & Luthfi, 2009).

Untuk membangun pohon klasifikasi pada algoritma C4.5 hal yang harus kita lakukan adalah:

- a. Hitung frekuensi kemunculan masing-masing *class* dan nilai atribut pada data *training*.
- b. Hitung nilai *Entropy* dari total *class* masing-masing nilai atribut  
*Entropy* digunakan untuk menentukan seberapa informatif sebuah masukan atribut untuk menghasilkan sebuah atribut. Rumus dasar dari *Entropy* adalah sebagai berikut (Jefri, 2013):

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (\text{Persamaan 1})$$

Keterangan:

$S$  : Himpunan Kasus

$n$  : Jumlah partisi  $S$

$p_i$  : Proporsi dari  $S_i$  terhadap  $S$

- c. Hitung nilai *Gain*

Untuk menghitung *gain* digunakan rumus seperti yang tertera berikut (Jefri, 2013):

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{S_i}{S} * Entropy(S_i) \quad (\text{Persamaan 2})$$

Keterangan:

$S$  : Himpunan kasus

$A$  : Atribut

$n$  : Jumlah partisi atribut  $A$

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

$|S|$  : Jumlah kasus dalam  $S$

- d. Ambil nilai *gain* terbesar dan jadikan *root node*.
- e. Hilangkan atribut yang dipilih sebelumnya dan ulangi perhitungan nilai *Entropy* dan *Gain* dengan memilih *Gain* terbesar dan dijadikan *internal node* pohon.
- f. Ulangi perhitungan tersebut hingga semua atribut pohon memiliki kelas.
- g. Proses partisi pohon klasifikasi akan berhenti saat semua cabang dalam *node* mendapat kelas yang sama dan atau tidak ada atribut yang dipartisi lagi dan atau tidak ada cabang yang kosong.

Untuk memperjelas konsep di atas, dapat digunakan contoh sebagai berikut mengenai pembentukan pohon klasifikasi dengan menggunakan algoritma C4.5 untuk menyelesaikan permasalahan penerimaan calon nasabah asuransi yang tepat. Contoh data *training* yang dapat dilihat pada Tabel 2.2.

Tabel 2.2. Data *Training*

Jenis kelamin	Status marital	Premi	Class
Pria	tidak kawin	Tinggi	lancar
Wanita	Kawin	Rendah	lancar
Pria	tidak kawin	Tinggi	lancar
Pria	tidak kawin	Rendah	lancar
Wanita	tidak kawin	Tinggi	lancar
Wanita	tidak kawin	Rendah	lancar
Wanita	Kawin	Tinggi	tidak lancar
Wanita	Kawin	Rendah	lancar
Wanita	Kawin	Rendah	tidak lancar
Pria	Kawin	Tinggi	tidak lancar
Pria	Kawin	Rendah	lancar
Pria	Kawin	Tinggi	tidak lancar

- a. Pemilihan atribut sebagai akar

Pemilihan akar ditentukan dengan perhitungan *entropy* dan *gain*. Rumus dapat dilihat pada persamaan 1 dan 2 pada C4.5 *algorithm* pada bab Tinjauan Pustaka. *Gain* dengan nilai tertinggi dijadikan akar.

- 1) Menghitung *entropy* pada *class*.

Menghitung jumlah kasus, jumlah kasus untuk keputusan Lancar, jumlah kasus untuk keputusan Tidak Lancar, dan *Entropy* dari semua kasus dan kasus yang dibagi berdasarkan atribut pekerjaan, premi, dan cara bayar. Perhitungan *entropy* pada *class* dapat dilihat pada Tabel 2.3.

Tabel 2.3. Perhitungan *Entropy* pada *Class*

lancar	8
tidak lancar	4
TOTAL	12
<b>ENTROPY</b>	0.918295834

- 2) Menghitung *entropy* dan *gain* tiap atribut

Perhitungan *entropy* dan *gain* tiap atribut dapat dilihat pada Tabel 2.4, Tabel 2.5, dan Tabel 2.6.

Tabel 2.4. Perhitungan Atribut Jenis Kelamin

	Pria	Wanita
Lancar	4	4
tidak lancar	2	2
TOTAL	6	6
<b>ENTROPY</b>	0.918295834	0.918295834
<b>GAIN</b>	0	

Tabel 2.5. Perhitungan Atribut Status Marital

	Kawin	Tidak kawin
Lancar	3	5
tidak lancar	4	0
TOTAL	7	5
<b>ENTROPY</b>	0.985228136	0
<b>GAIN</b>	0.343579421	

Tabel 2.6. Perhitungan Atribut Premi

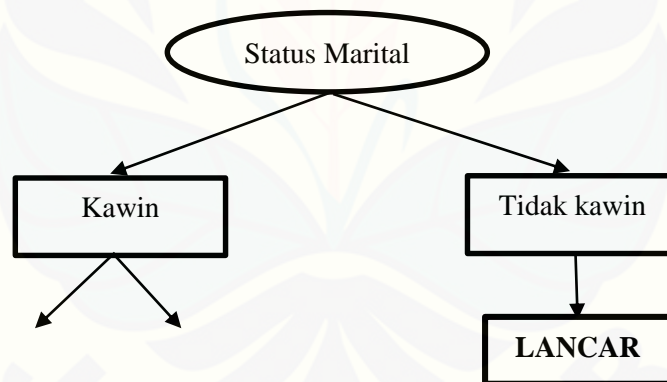
	Rendah	Tinggi
Lancar	6	2
tidak lancar	1	3
TOTAL	7	5
<b>ENTROPY</b>	0.591672779	0.970950594
<b>GAIN</b>	0.168590632	

## 3) Memilih akar

Atribut status marital memiliki nilai *gain* tertinggi, oleh karena itu atribut status marital dipilih sebagai akar.

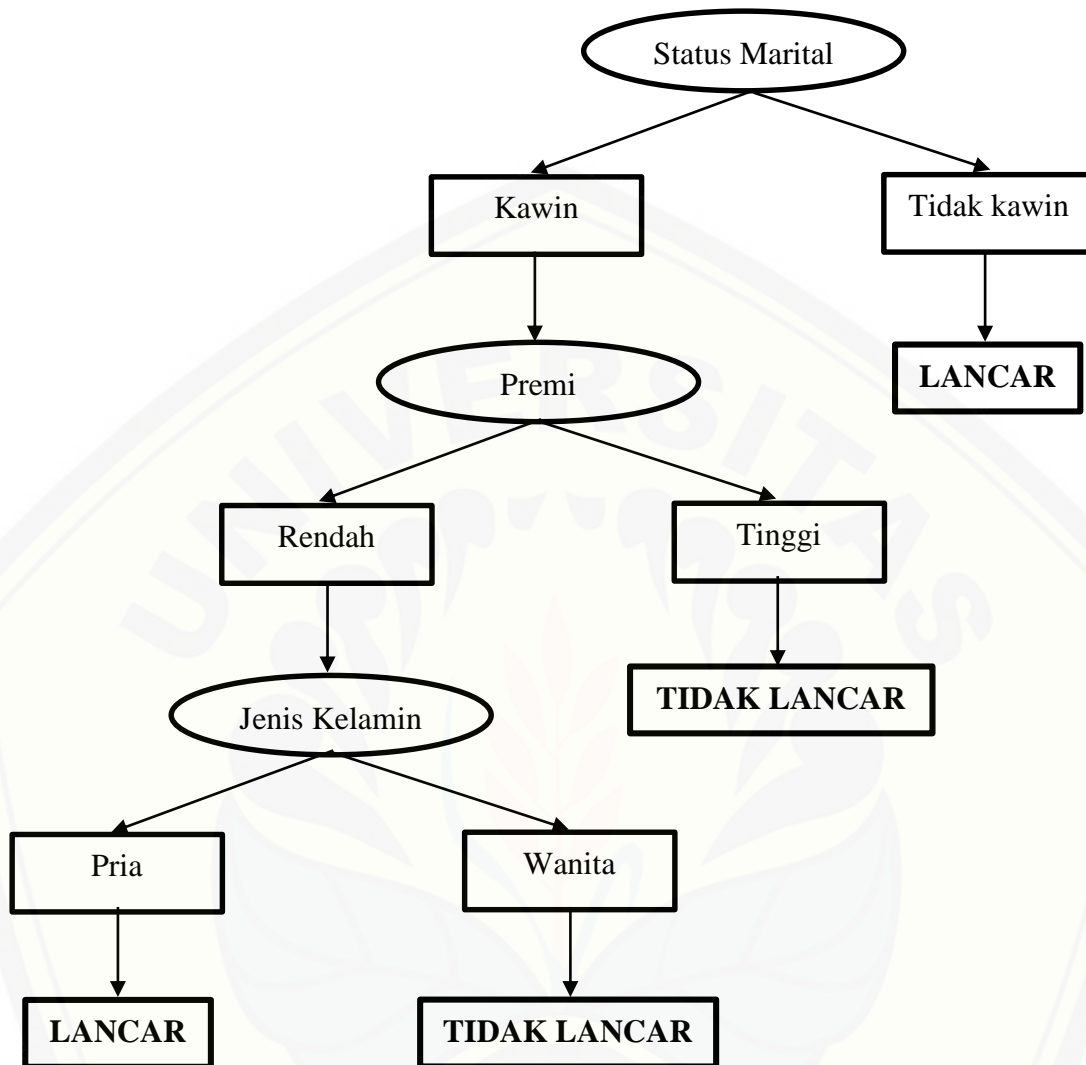
## b. Buat cabang untuk masing- masing nilai dan bagi kasus dalam cabang

Atribut yang sudah ditetapkan sebagai akar, kemudian displit. *Node* yang sudah memiliki nilai dominan atau mutlak ke salah satu *class*, maka *class* dapat ditentukan. *Node* yang masih memiliki nilai di masing-masing *class*, displit lagi. Berikut contoh hasil dari pembuatan cabang dapat dilihat di Gambar 2.2.



Gambar 2.2. Cabang Masing-Masing Nilai

## c. Ulangi perhitungan tersebut hingga semua atribut pohon memiliki kelas. Sehingga pohon dapat terbentuk seperti pada Gambar 2.3.



## 2.8 Cost Sensitive Learning

*Cost sensitive learning* merupakan salah satu teknik pembuatan *model* prediksi yang dalam prosesnya juga memperhitungkan *misclassification cost*. *Misclassification cost* merupakan kerugian yang timbul ketika ada data yang salah diklasifikasikan ke kelas tertentu jika mengklasifikasikan sebuah data yang sebenarnya termasuk suatu *class* menjadi *class* yang lain. *Misclassification cost* merupakan hal yang sangat penting karena data yang salah diklasifikasikan dapat menimbulkan *cost* yang berbeda-beda.  $C(i, j)$  menotasikan *cost function* yang berarti kerugian jika mengklasifikasikan

sebuah data yang sebenarnya berasal dari *class*  $i$  menjadi *class*  $j$  (Tan, Steinbach, & Kumar, 2005).

Dari sekumpulan data *training*, *cost* total dari sebuah *model* prediksi  $C(M)$  yang dibangun dari data *training* tersebut adalah (Tan, Steinbach, & Kumar, 2005):

$$C(M) = YY \times C(+, +) + NY \times C(-, +) + YN \times C(+, -) + NN \times C(-, -)$$

(Persamaan 3)

Keterangan:

- YY : Banyaknya data positif yang diklasifikasi positif
- NY : Banyaknya data negatif yang diklasifikasi positif
- YN : Banyaknya data positif yang diklasifikasi negatif
- NN : Banyaknya data negatif yang diklasifikasi negatif.

Pada umumnya, nilai  $C(+, +)$  dan  $C(-, -)$  adalah 0 karena tidak ada penalti jika *model* prediksi melakukan prediksi dengan benar. Dengan demikian, persamaan di atas dapat disederhanakan menjadi (Tan, Steinbach, & Kumar, 2005):

$$C(M) = NY \times C(-, +) + YN \times C(+, -) \quad \text{(Persamaan 4)}$$

Pada *classification tree*, *cost sensitive learning* dapat diintegrasikan dalam beberapa cara (Tan, Steinbach, & Kumar, 2005):

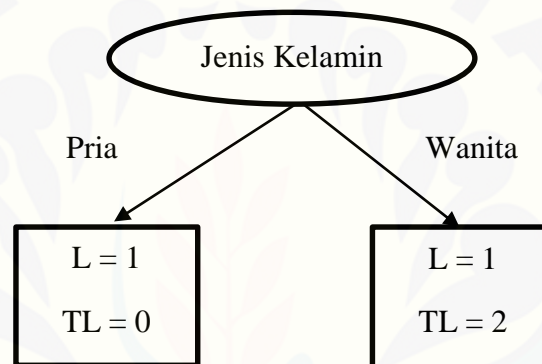
- 1) *Cost function* digunakan sebagai kriteria untuk pemilihan atribut yang akan displit
- 2) *Cost function* digunakan untuk memilih bagian *tree* yang akan dipangkas
- 3) *Cost function* digunakan memanipulasi bobot dari data *training* sehingga menghasilkan *tree* dengan *cost* minimal
- 4) *Cost function* digunakan sebagai kriteria untuk pelabelan *leaf node*

Penggunaan *cost sensitive learning* pada algoritma C4.5 salah satunya dapat diintegrasikan dengan cara penggunaan *cost function* sebagai kriteria untuk pelabelan *leaf node*. Proses pembuatan *classification tree* dilakukan seperti biasa, tetapi pada

proses penentuan *class* label dari *leaf node* tidak dilakukan dengan menggunakan *class* mayoritas isi *leaf node*. Metode penentuan *class* label dari *leaf node* dilakukan dengan memberikan *class* label pada *leaf node* yang memberikan *cost* paling kecil.

Untuk memperjelas konsep di atas, dapat digunakan contoh sebagai berikut.

Proses pelabelan kelas *leaf node* menggunakan *cost function*, dengan nilai  $C(L, L) = C(TL, TL) = 0$  dan  $C(L, TL) = 70$ ,  $C(TL, L) = 30$ . Penentuan *cost* tiap peleburan L ke L, TL ke TL, L ke TL dan TL ke L berdasarkan seorang *expert* yang ahli dalam objek penelitian yang sedang di teliti. *Classification tree* yang akan dilakukan pelabelan *leaf node* ditunjukkan oleh Gambar 2.4.



Gambar 2.4. Pelabelan *Leaf Node*

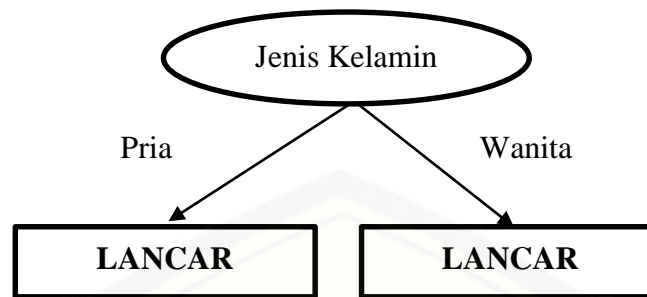
Jika pada *classification tree* biasa, *leaf node* jenis kelamin wanita adalah kelas TL berdasarkan kelas mayoritas dimana jumlah kelas TL lebih banyak dari pada jumlah kelas L, jika proses pelabelan memasukkan *cost function* maka akan diperoleh:

$$L \text{ ke TL} = 1 \times 70 = 70,$$

$$TL \text{ ke L} = 2 \times 30 = 60,$$

sehingga *leaf node* jenis kelamin wanita adalah kelas L karena *cost* mengklasifikasikan *leaf node* sebagai kelas L kurang dari *cost* mengklasifikasikan *leaf node* sebagai TL. Sehingga hasil pelabelan *leaf node classification tree* seperti yang ditunjukkan pada Gambar 2.5.





Gambar 2.5. Pelabelan *Leaf Node* menggunakan *Cost Sensitive Learning*

## 2.9 Confusion Matrix

*Confusion matrix* merupakan sebuah tabel yang terdiri dari banyaknya baris data uji yang diprediksi benar dan tidak benar oleh *model* klasifikasi. Tabel confusion matrix diperlukan untuk menentukan kinerja suatu model klasifikasi (Tan, Steinbach, & Kumar, 2005). Hasil dari pengukuran metrik ini berupa persentase dimana semakin tinggi persentase yang dihasilkan oleh *accuracy*, *precision*, dan *recall* berarti kontribusi yang diajukan memberikan hasil yang baik namun jika semakin rendah persentase yang dihasilkan oleh *accuracy*, *precision*, dan *recall* berarti kontribusi yang diajukan memberikan hasil yang kurang baik. Untuk mempermudah pemahaman mengenai uji perfromansi dapat dilihat pada Tabel 2.7.

Tabel 2.7. Matriks Kontingensi Kelas Prediksi dan Aktual

		Kelas Prediksi	
		<i>Yes</i>	<i>No</i>
Kelas Aktual	<i>Yes</i>	YY	NY
	<i>No</i>	YN	NN

Sumber: (Swastina, 2013)

Keterangan:

- YY adalah diprediksi *yes*, fakta *yes*
- YN adalah diprediksi *yes*, fakta *no*
- NY adalah diprediksi *no*, fakta *yes*
- NN adalah prediksi *no*, fakta *no*

*Accuracy* adalah rasio jumlah ketepatan prediksi suatu kelas dengan data *testing* secara keseluruhan dalam kelas tersebut. Rumusnya:

$$Accuracy = \frac{YY+NN}{YY+YN+NN+NY} \quad (\text{Persamaan 5})$$

*Precision* adalah rasio jumlah ketepatan prediksi suatu kelas dengan jumlah total prediksi yang diklasifikasikan dalam kelas tersebut. Rumusnya:

$$Precision (P) = \frac{YY}{YY+YN} \quad (\text{Persamaan 6})$$

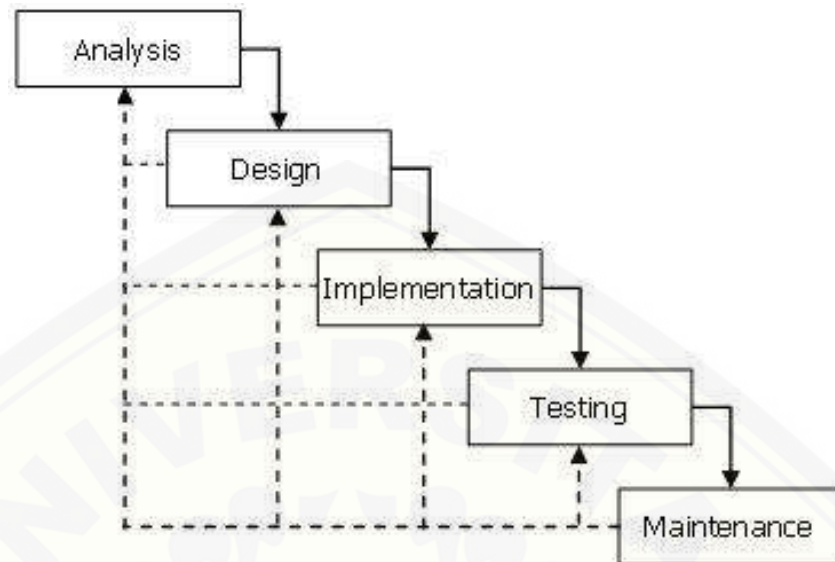
*Recall* adalah rasio jumlah ketepatan prediksi suatu kelas dengan jumlah total fakta yang diklasifikasikan dalam kelas tersebut. Rumusnya:

$$Recall (P) = \frac{YY}{YY+NY} \quad (\text{Persamaan 7})$$

Penggunaan *recall* digunakan untuk mengetahui tingkat akurasi dengan penggunaan *cost sensitive learning*, karena menurut Wicaksono (2010) penggunaan *cost sensitive learning* bertujuan untuk meningkatkan nilai *recall model* prediksi.

## 2.10 System Development Life Cycle (SDLC)

*System Development Life Cycle* (SDLC) merupakan model untuk merancang, membangun dan memelihara sistem. Ada banyak model SDLC, salah satunya model *waterfall*. Kelebihan model *waterfall* adalah mudah dimengerti, sudah banyak digunakan dan dikenal, menjadi model linier dan sangat mudah diimplemtasikan, setiap proses dilakukan satu persatu secara berurutan. Kelemahan model *waterfall* adalah tidak *fleksible*, bukan model yang digunakan untuk projek yang kompleks, dokumentasi hanya bisa dibuat sampai tahap terakhir telah selesai (Alshamrani & Bahttab, 2015). SDLC model *waterfall* merupakan model yang memiliki tahapan pengembangan sistem secara terurut. Ada lima tahap pada model *waterfall* yaitu *analysis, design, implementation, testing* dan *maintenance* (Bassil Y. , 2012). Tahapan dari model *waterfall* dapat dilihat pada gambar 2.6.



Gambar 2.6. Model *Waterfall*  
(Sumber : (Bassil Y. , 2012))

Berdasarkan penjelasan mengenai model *waterfall* diatas, pada penelitian ini menggunakan *System Development Life Cycle (SDLC) model waterfall*. Model *waterfall* ini dipilih karena mempunyai tahapan pengembangan sistem yang dilakukan secara terurut, mulai dari analisis kebutuhan, desain sistem, implementasi sistem dan pengujian sistem. Sistem informasi pengklasifikasian penerimaan calon nasabah asuransi ini tergolong sistem yang tidak terlalu kompleks dan pengerjaannya dilakukan oleh perseorangan sehingga model *waterfall* sangat sesuai jika diterapkan pada penelitian ini.

### BAB 3. METODOLOGI PENELITIAN

Bab ini menjelaskan tentang metode yang digunakan dalam merancang dan membangun sistem yaitu metode pengembangan.

#### 3.1 Jenis Penelitian

Jenis penelitian yang digunakan pada penelitian ini merupakan jenis penelitian kualitatif dan kuantitatif. Penelitian kualitatif digunakan dalam proses analisa studi literatur yang berhubungan dengan sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*. Penelitian kuantitatif dilakukan untuk pengumpulan data dan analisis data serta mengkaji teori dan penelitian yang telah dilakukan sebelumnya.

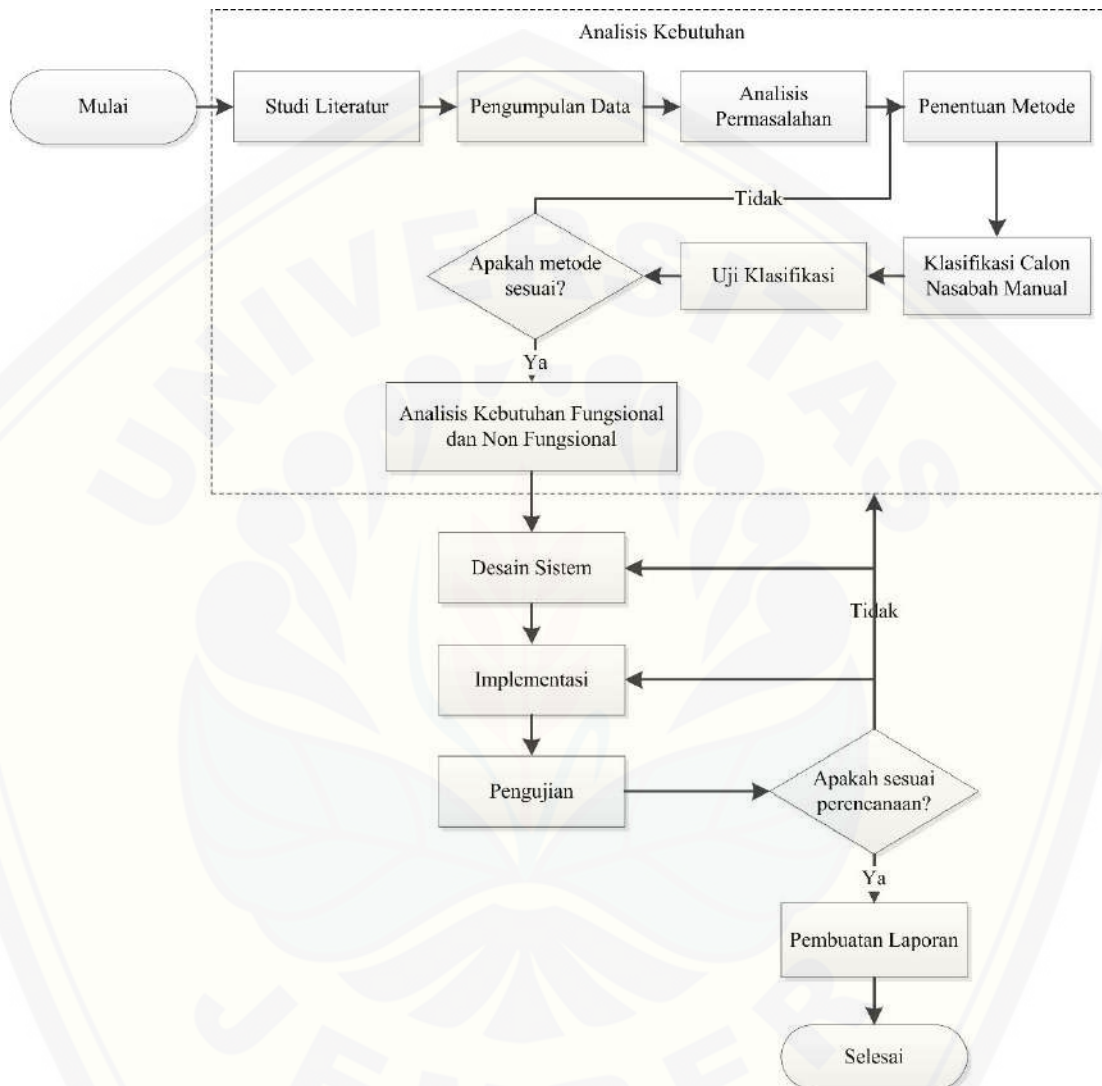
#### 3.2 Tempat dan Waktu Penelitian

Tempat Penelitian yang dilaksanakan untuk penelitian ini adalah Perusahaan Asuransi XYZ Indonesia Cabang Jember. Penelitian dilakukan selama 6 bulan yaitu dari bulan Oktober 2017 – Maret 2018.

#### 3.3 Tahapan Penelitian

Tahapan penelitian yang digunakan pada penelitian ini dilakukan dengan menggunakan model SDLC (*System Development Life Cycle*) *waterfall*. Model *waterfall* ini dipilih karena mempunyai tahapan pengembangan sistem yang dilakukan secara terurut, mulai dari analisis kebutuhan, desain sistem, implementasi sistem dan pengujian sistem. Sistem informasi pengklasifikasian penerimaan calon nasabah asuransi ini tergolong sistem yang tidak terlalu kompleks dan pengerjaannya dilakukan oleh perseorangan sehingga model *waterfall* sangat sesuai jika diterapkan pada

penelitian ini. Berikut alur tahapan penelitian sesuai model *waterfall* yang dapat dilihat pada Gambar 3.1.



Gambar 3.1. Alur Penelitian Sesuai Model *Waterfall*  
(Sumber: (Hasil Analisis))

Pengembangan sistem pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi dapat dilakukan dengan *model waterfall* melalui tahapan berikut:

### 3.3.1 Analisis Kebutuhan

Tahap awal dari penelitian ini adalah analisis kebutuhan. Tahap ini terdiri dari studi literatur, pengumpulan data, analisis permasalahan, penentuan metode, klasifikasi calon nasabah manual dan uji klasifikasi.

#### 1) Studi Literatur

Studi literatur yang digunakan pada penelitian ini berguna untuk mendapatkan gambaran yang menyeluruh tentang apa yang sudah dikerjakan peneliti lain dan bagaimana peneliti mengerjakannya, kemudian mengidentifikasi celah penelitian sebagai dasar penelitian yang akan dilakukan. Penelitian-penelitian terdahulu berfungsi sebagai pendukung bagi peneliti untuk melakukan penelitian baru dalam mengidentifikasi permasalahan pada penerimaan calon nasabah asuransi dan metode-metode yang digunakan dalam *data mining*. Penggalan informasi dari literatur dilakukan sebagai upaya memperjelas permasalahan yang memiliki kaitan dengan penelitian ini, sekaligus untuk membedakan antara penelitian ini dengan penelitian sebelumnya. Literatur yang digunakan sebagai acuan dalam penelitian ini merupakan hasil penelitian dari kalangan akademis yang disajikan dalam bentuk paper dalam jurnal.

#### 2) Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan untuk memperoleh informasi yang dibutuhkan dalam rangka mencapai tujuan penelitian (Gulo, 2000). Pengumpulan data diawali dengan mengumpulkan informasi yang berhubungan dengan masalah penelitian. Informasi tersebut diperoleh dengan melakukan survey untuk menentukan lokasi penelitian, mengidentifikasi kompetensi calon *user* terhadap sistem berbasis teknologi informasi dan mengidentifikasi permasalahan calon *user* yang akan menggunakan sistem klasifikasi penerimaan calon nasabah asuransi. Teknik pengumpulan data yang akan dilakukan pada penelitian ini adalah wawancara.

Wawancara merupakan cara pengumpulan data yang berhubungan langsung dengan narasumber dengan menyajikan pertanyaan kepada narasumber. Wawancara

terhadap narasumber bertujuan untuk memperoleh data yang dibutuhkan dalam penyelesaian penelitian. Wawancara ini ditujukan *admin*, *leader*, dan *agent* yang mengetahui permasalahan di bidang asuransi. Dalam penelitian ini dilakukan wawancara mengenai bisnis proses penerimaan calon nasabah asuransi dan mendapatkan data *set* yaitu nasabah asuransi yang akan digunakan sebagai data *training* dan data *testing* dalam penerapan *classification tree C4.5 algorithm* dan *cost sensitive learning* untuk menentukan penerimaan calon nasabah asuransi.

### 3) Analisis Permasalahan

Tahap analisis permasalahan dimulai dengan menelaah secara keseluruhan yang telah diperoleh dari studi literatur dan pengumpulan data. Langkah selanjutnya yaitu menganalisis permasalahan yang terjadi pada Perusahaan Asuransi XYZ Cabang Jember. Permasalahan yang terjadi pada Perusahaan Asuransi XYZ Cabang Jember ialah adanya nasabah yang menunggak dalam membayar premi sehingga mempengaruhi keuangan perusahaan dan tentunya merugikan perusahaan asuransi. Setelah mengetahui permasalahan yang ada, pada penelitian ini solusi yang digunakan untuk mengatasi permasalahan tersebut yaitu dengan pembangunan sistem informasi pengklasifikasian penerimaan calon nasabah asuransi yang di fokuskan pada klasifikasi penerimaan calon nasabah asuransi yang tepat oleh *leader*.

### 4) Penentuan Metode

Pada tahap ini data-data yang sudah dikumpulkan, kemudian dianalisis untuk mengetahui data tersebut memenuhi syarat untuk digunakan pada proses klasifikasi yang selanjutnya dilakukan perubahan data asli menjadi data *set* yang digunakan dalam proses pembangunan *classification tree* (pohon klasifikasi). Penentuan metode klasifikasi ini dilakukan untuk mempermudah perhitungan klasifikasi calon nasabah. Penentuan metode dilakukan dengan cara menyesuaikan data yang diperoleh dari hasil analisis data yang terkumpul dengan tipe data dan akurasi yang dihasilkan. Berdasarkan tipe data set dan akurasi yang dihasilkan, *classification tree C4.5*

*algorithm* dan *cost sensitive learning* dipilih dan digunakan dalam memprediksi kelancaran pembayaran premi calon nasabah asuransi.

#### 5) Klasifikasi Calon Nasabah

Pada tahap ini dilakukan proses pembangunan *classification tree* (pohon klasifikasi) yang dilakukan untuk menghasilkan *rule*. *Rule* dihasilkan dari hasil konversi *classification tree* (pohon klasifikasi) yang dibuat sebelumnya. Pembangunan *classification tree* ini disesuaikan dengan bisnis proses yang ada pada Perusahaan Asuransi XYZ Cabang Jember. Berdasarkan bisnis proses tersebut, *rule* yang dihasilkan dari konversi *classification tree* tersebut akan digunakan untuk proses klasifikasi kelancaran calon nasabah dalam pembayaran premi sehingga dapat ditentukan calon nasabah mana yang dikategorikan Lancar, dan mana yang Tidak Lancar.

#### 6) Uji Klasifikasi

Uji klasifikasi yang digunakan untuk mengukur performa hasil klasifikasi adalah *confusion matrix*. Pada tahap ini dilakukan uji klasifikasi hasil pembangunan *classification tree* untuk mengetahui kinerja suatu model klasifikasi berdasarkan nilai *accuracy*, *precision*, dan *recall*. Hasil dari uji klasifikasi ini berupa persentase dimana semakin tinggi persentase yang dihasilkan oleh *accuracy*, *precision*, dan *recall* berarti metode yang digunakan memberikan hasil yang baik namun jika semakin rendah persentase yang dihasilkan oleh *accuracy*, *precision*, dan *recall* berarti metode yang digunakan memberikan hasil yang kurang baik.

#### 7) Analisis Kebutuhan Fungsional dan Non – Fungsional Sistem

Pada tahap ini dilakukan dengan menentukan kebutuhan fungsional dan non fungsional dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi. Kebutuhan fungsional ini berisikan proses-proses yang akan dilakukan oleh sistem, salah satunya mampu mengklasifikasi kelancaran calon nasabah dalam pembayaran premi. Setelah itu, dilakukan penentuan kebutuhan non-fungsional yang meliputi



kebutuhan diluar sistem seperti perangkat lunak atau perangkat keras yang mendukung pembangunan sistem informasi pengklasifikasian penerimaan calon nasabah asuransi.

### 3.3.2 Desain Sistem

Pada penelitian ini, pembuatan desain sistem menggunakan *Unified Modeling Language* (UML) yang dirancang dengan konsep *Object Oriented Design*. Pemodelan UML yang digunakan adalah sebagai berikut:

- 1) Elemen *input* dan *output* sistem (Menggambarkan elemen yang terdiri dari elemen *input* atau masukkan ke sistem, elemen *output* atau keluaran dari sistem, *goal* atau tujuan dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi dan *uses* atau media yang digunakan untuk membuat sistem. Penentuan elemen *input* dan *output* ini berdasarkan kebutuhan fungsional yang telah dijelaskan pada tahap analisis kebutuhan di atas)
- 2) *Use Case Diagram* (Menggambarkan apa saja aktifitas yang dilakukan oleh tiap *user* dengan hak akses yang berbeda-beda pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi)
- 3) *Scenario Diagram* (Menjelaskan alur sistem dari fitur yang ada di *job specification* dan *job description* yang ada pada *diagram use case* sistem informasi pengklasifikasian penerimaan calon nasabah asuransi)
- 4) *Activity Diagram* (Menggambarkan berbagai alir aktivitas dalam sistem informasi pengklasifikasian penerimaan calon nasabah asuransi yang sedang dirancang, bagaimana masing-masing aktivitas berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas berakhir)
- 5) *Sequence Diagram* (Rangkaian pesan yang dikirim antar objek dan interaksi antar objek pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi)
- 6) *Class Diagram* (Menggambarkan struktur dan deskripsi *class* serta hubungan antar *class*, sehingga memudahkan dalam proses pengkodean pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi)

- 7) *Entity Relationship Diagram* (ERD) (menjelaskan hubungan antar data dalam basis data berdasarkan data yang mempunyai hubungan antar relasi pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi)

### 3.3.3 Implementasi

Pada tahap ini desain yang telah dibuatkan akan diimplementasikan dalam kode program. Beberapa hal yang dilakukan dalam tahap implementasi antara lain:

- 1) Implementasi kode program (*coding*) menggunakan bahasa *Page Hypertext Preprocessor* (PHP) dan menggunakan sublime sebagai IDE-nya.
- 2) Menggunakan *framework CodeIgniter*.
- 3) Manajemen basis data menggunakan *Database Management System* (DBMS) MySQL.
- 4) Menggunakan aplikasi XAMPP untuk menjalankan program *localhost*.

### 3.3.4 Pengujian Sistem

Pada penelitian ini akan dinilai apakah sistem yang dirancang telah sesuai dengan apa yang diharapkan, serta untuk mengevaluasi efektivitas, keunggulan dan kelemahan terhadap kualitas sistem informasi pengklasifikasian penerimaan calon nasabah asuransi ini.

Uji coba dan evaluasi dilakukan dua metode pengujian yaitu:

- 1) *White Box Testing*

Merupakan cara pengujian dengan mencari kesalahan ada yang ada pada kode program. Pengujian ini dilakukan oleh *developer*.

- 2) *Black Box Testing*

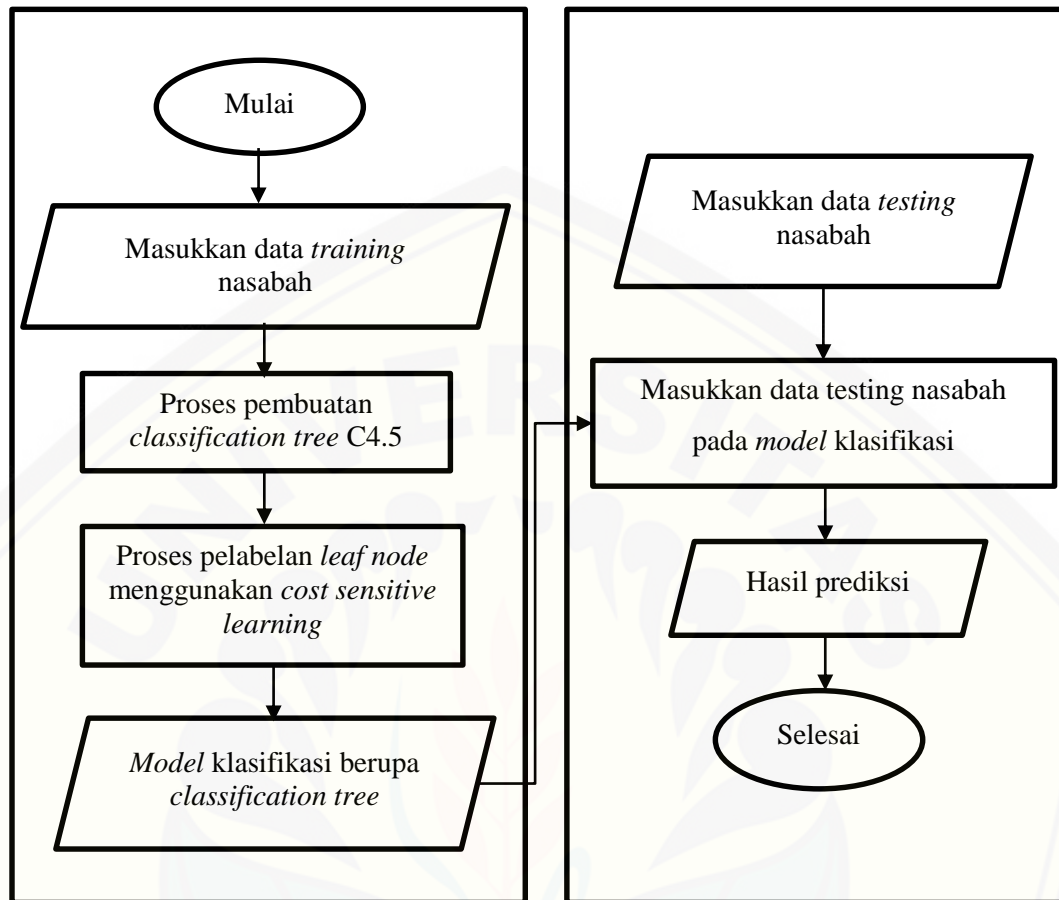
Merupakan cara pengujian dengan memperhatikan fungsionalitas yang berkaitan dengan input/output apakah sudah sesuai dengan sistem yang dijalankan. Pengujian ini dilakukan oleh pengguna/*user*.

### 3.3.5 Pemeliharaan

Pemeliharaan dilakukan untuk mengatasi masalah ketika sistem informasi pengklasifikasian penerimaan calon nasabah asuransi sudah digunakan oleh pengguna/*user*. Ketika *user* menemukan bug pada sistem tersebut, maka *user* akan konfirmasi ke *developer* untuk mendapat penanganan dari pengembang sistem. *Update* sistem dilakukan dengan memulainya kembali ke tahap analisis kebutuhan dan seterusnya hingga sistem sesuai dengan keinginan *user*.

## 3.4 Gambaran Umum Sistem

Sistem yang akan dibangun pada penelitian ini adalah sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*. Sistem ini nantinya digunakan oleh *leader* untuk mengklasifikasi apakah calon nasabah asuransi yang mendaftar diklasifikasikan lancar atau tidak lancar dalam pembayaran premi dengan memanfaatkan data nasabah yang sudah ada, kemudian perusahaan akan melakukan evaluasi dan validasi kepada calon nasabah dengan memberikan izin untuk melakukan registrasi pendaftaran asuransi yang berkategori lancar, sedangkan untuk nasabah yang berkategori tidak lancar akan mendapat *feedback* berupa saran untuk pengajuan asuransi. Pada sistem ini juga terdapat data nasabah yang ada pada Perusahaan Asuransi XYZ Cabang Jember dan beberapa informasi mengenai asuransi untuk calon nasabah. Alur sistem dapat dilihat pada Gambar 3.2.

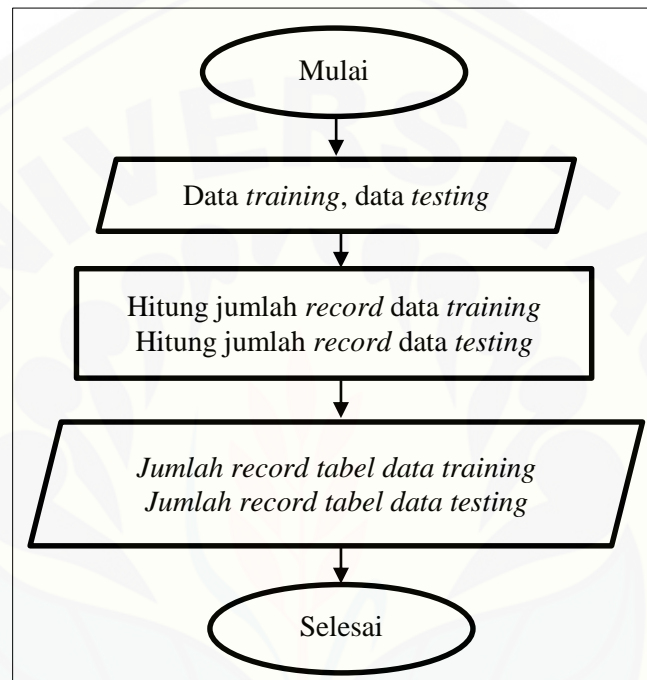


Gambar 3.2. Alur Sistem  
(Sumber: (Hasil Analisis))

Sistem ini mengklasifikasi status kelancaran dari calon nasabah berdasarkan data nasabah asuransi. Klasifikasi tersebut menggunakan *rule* yang dihasilkan dari proses *data mining* dengan menggunakan algoritma *classification tree* C4.5 dan *cost sensitive learning*.

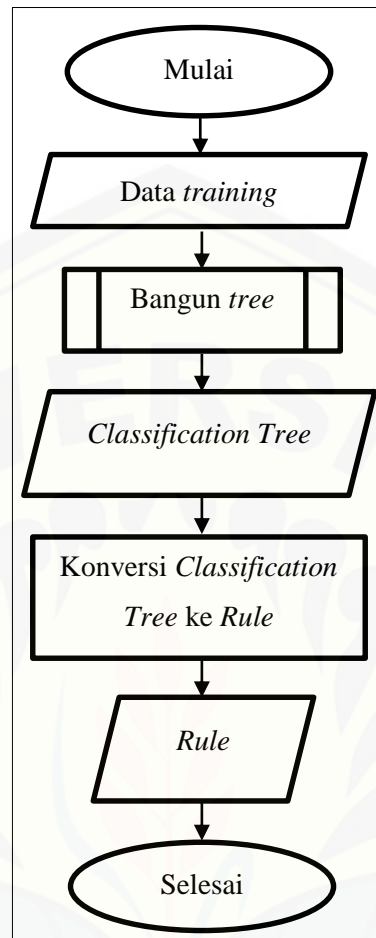
Proses klasifikasi *class* didasarkan pada *rule* yang telah dikonversi sebelumnya dari hasil membangun *classification tree*. Pada proses pembangunan *classification tree* menggunakan C4.5 dan *cost sensitive learning*, peneliti membuat sistem yaitu sistem informasi pengklasifikasian penerimaan calon nasabah asuransi. Pada proses pembangunan *classification tree*, terdapat beberapa proses utama, yaitu: proses *request data*, proses pembangunan *model* (proses *learning*), dan proses klasifikasi data (proses *testing*) (Aini, 2015).

Proses yang dilakukan yaitu proses *request* data, dimana sistem mendapatkan data untuk diolah, yaitu data *training* dan data *testing* beserta informasinya. Informasi yang didapat dari data *training* yaitu *record* tabel data *training*, sedangkan informasi yang didapat dari data *testing* yaitu *record* tabel data *testing*. Proses *request* data dapat dilihat pada Gambar 3.3.



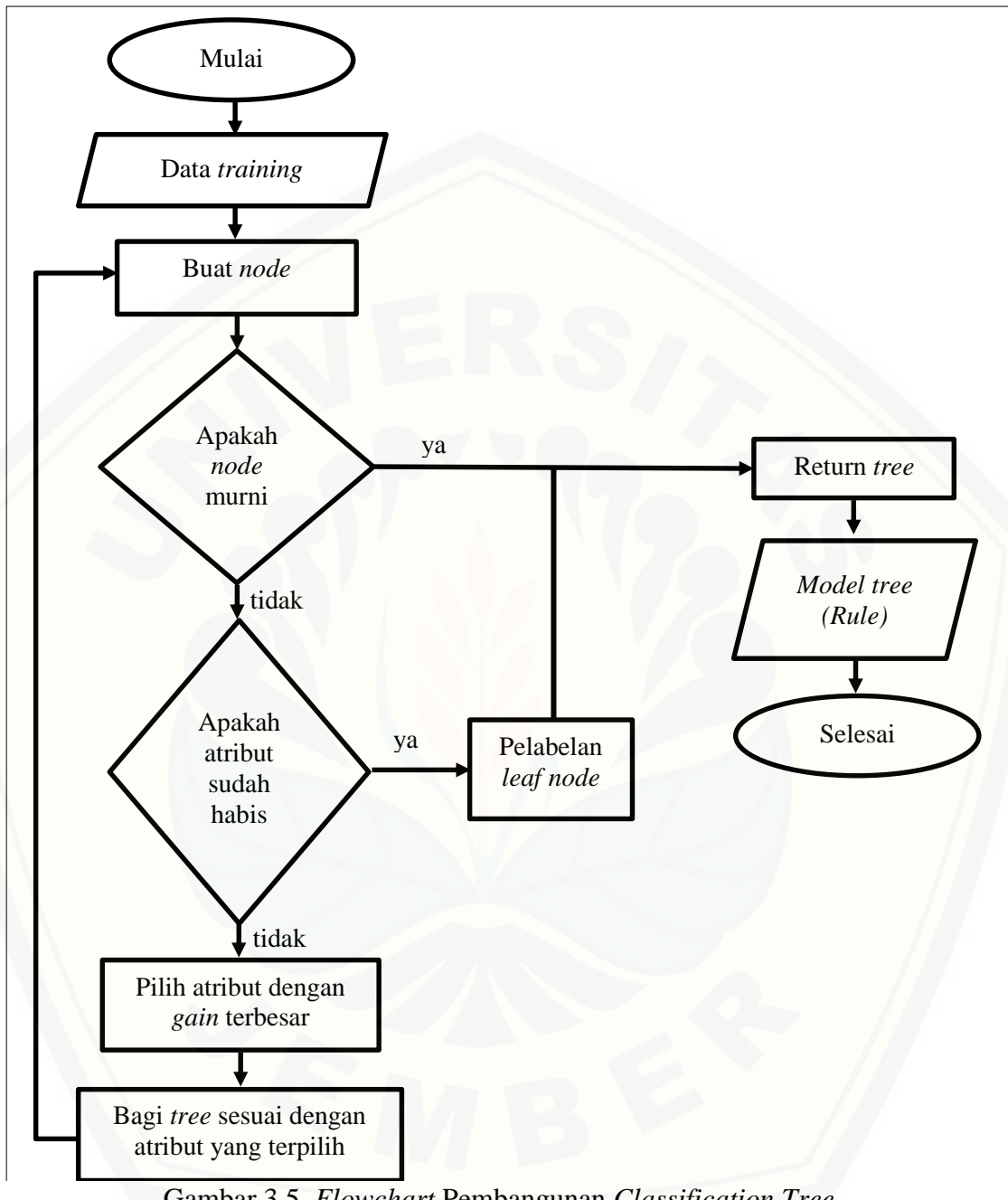
Gambar 3.3. Flowchart *Request* Data  
(Sumber: (Hasil Analisis))

Penjelasan Gambar 3.3 adalah langkah – langkah proses *request* data yang diawali dengan mendapatkan masukan yaitu tabel data *training* dan data *testing* dan kemudian menghitung jumlah *record* data *training* dan data *testing* lalu disimpan. Proses selanjutnya yaitu proses *learning*. Proses tersebut dapat dilihat pada Gambar 3.4.



Gambar 3.4. Flowchart Proses *Learning*  
(Sumber: (Hasil Analisis))

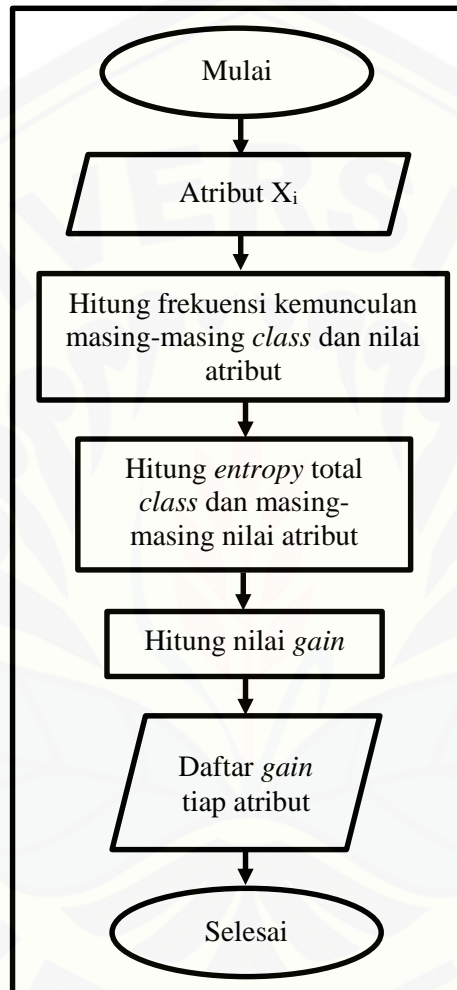
Penjelasan Gambar 3.4 adalah langkah-langkah proses *learning* yang diawali dengan mendapat data *training* kemudian proses pembangunan *tree* dan akan menghasilkan *classification tree*. Setelah itu *classification tree* akan dikonversi ke dalam bentuk *rule*. Proses pembangunan *tree* dapat dilihat pada Gambar 3.5.



Gambar 3.5. Flowchart Pembangunan Classification Tree  
(Sumber: (Hasil Analisis))

Penjelasan Gambar 3.5 adalah langkah – langkah untuk membangun *tree*. Jika *node* murni maka akan kembali ke *tree*, sedangkan jika *node* tidak murni maka dilakukan perulangan dan menghitung kembali *entropy* dan *gain*. Kemudian pilih

atribut dengan *gain* terbesar dan buat kembali *tree* atau pohon klasifikasi. Proses iterasi akan berakhir jika *node* telah murni dan tidak dapat dibagi lagi. *Node* dikatakan murni jika semua data didalamnya sudah memiliki kelas. Pembuatan *node* dapat dilihat pada Gambar 3.6.



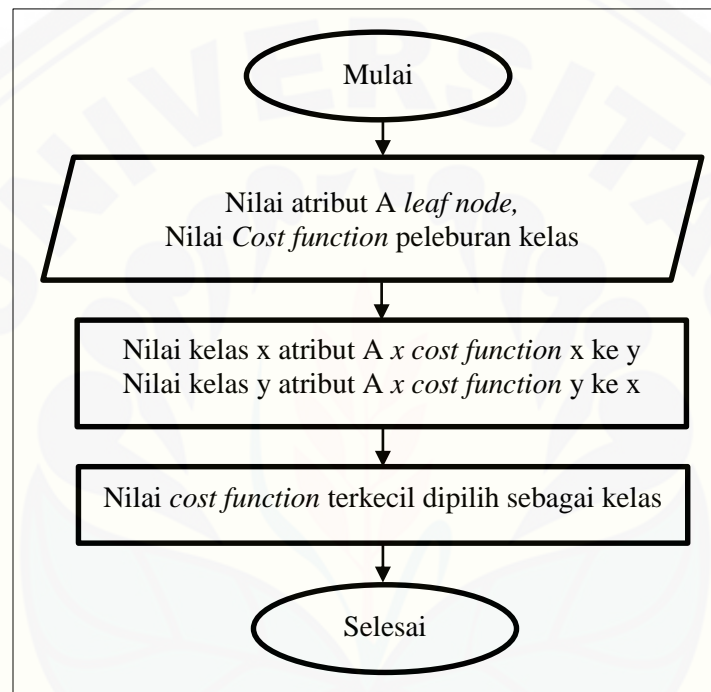
Gambar 3.6. Flowchart Pembuatan *Node*  
(Sumber: (Hasil Analisis))

Penjelasan Gambar 3.6 adalah langkah – langkah untuk pembuatan *node* yang diawali dengan mendapat data dengan atribut  $X_i$  lalu menghitung frekuensi kemunculan masing-masing *class* dan nilai atribut. Setelah itu menghitung *entropy* total *class* dan masing-masing nilai atribut menggunakan persamaan (1). Selanjutnya



menghitung nilai *gain* tiap atribut menggunakan persamaan (2). Setelah semuanya dihitung, didapatkan daftar *gain* tiap atribut.

Proses pelabelan *leaf node* pada algoritma C4.5, dipenelitian ini menggunakan *cost sensitive learning*. Seperti yang telah dijelaskan pada subbab 2.8, penggunaan *cost sensitive learning* digunakan untuk mengatasi *misclassification cost*. Proses tersebut ditunjukkan pada Gambar 3.7.



Gambar 3.7. Pelabelan Leaf Node menggunakan Cost Sensitive Learning  
(Sumber: (Hasil Analisis, 2017))

Penjelasan Gambar 3.7 merupakan langkah-langkah untuk pelabelan *leaf node* menggunakan *cost sensitive learning*, dengan A sebagai atribut *leaf node*, x dan y sebagai kelas, serta  $x$  merupakan perkalian.

## BAB 4. PENGEMBANGAN SISTEM

Bab ini menguraikan mengenai analisis kebutuhan, desain, implementasi, dan pengujian sistem yang digunakan dalam proses pengembangan sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*. Tahapan analisis hingga pengujian dilakukan sesuai dengan metode pengembangan *waterfall*.

### 4.1 C4.5 Algorithm dan Cost Sensitive Learning untuk Klasifikasi Kelancaran Pembayaran Premi

Bagian ini akan menguraikan mengenai bagaimana *classification tree C4.5 algorithm* dan *cost sensitive learning* digunakan untuk mengklasifikasi kelancaran calon nasabah dalam pembayaran premi. Proses pembangunan *classification tree* menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* dilakukan sebelum proses klasifikasi untuk menghasilkan *rule* yang akan digunakan pada proses klasifikasi. Oleh karena itu, pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi terdapat fitur untuk proses pembangunan *classification tree*. Proses utama dari pembangunan *classification tree* terdapat tiga, yaitu: proses *request data*, proses *learning*, dan proses *testing*.

#### 4.1.1 Request Data

*Request data* merupakan suatu proses yang pertama dilakukan dalam proses pembangunan *classification tree* yaitu mendapatkan data data untuk diolah, yaitu berupa data *training* dan data *testing*. Jumlah data *training* yang digunakan sebesar 90% dari jumlah data *set* yaitu 153 data (118 data *class* Lancar dan 35 data *class* Tidak Lancar), sedangkan data *testing* yang digunakan sebesar 10% dari jumlah data *set* yaitu 17 data (13 data *class* Lancar dan 4 data *class* Tidak Lancar). Untuk melihat data

*training* dan data *testing*, dalam sistem terdapat fitur menu data *training* dan data *testing*.

#### 4.1.2 Learning

Proses setelah mendapatkan data *training*, yaitu proses *learning*. Berikut langkah – langkah proses *learning* pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*:

1. Sistem mendapatkan data *training*
2. Menggunakan data *training* yang didapat, sistem melakukan proses pembangunan *classification tree* dengan menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*.
3. Sistem menghasilkan *classification tree*.
4. Sistem mengkonversi *classification tree* ke dalam bentuk *rule*.

Sistem melakukan proses pembangunan *classification tree* sesuai dengan langkah – langkah dan perhitungan yang telah dijelaskan pada subbab 2.7 untuk *classification tree C4.5 algorithm* dan subbab 2.8 untuk *cost sensitive learning*. Pada sistem informasi pengklasifikasian calon nasabah asuransi, proses pembangunan *classification tree* terdapat pada menu data *training* yaitu pada tombol *classification tree*.

#### 4.1.3 Testing

Proses terakhir pada pembangunan *classification tree* yaitu proses *testing* (uji klasifikasi). Data *testing* akan digunakan untuk melakukan proses *testing* yang akan menghasilkan *class* baru dari tiap data *testing*. Kemudian *class* baru tersebut akan dibandingkan dengan *class* lama untuk mendapatkan *accuracy*, *precision*, dan *recall* dari *rule* yang terbentuk. Proses *testing* telah dijelaskan pada subbab 2.9.

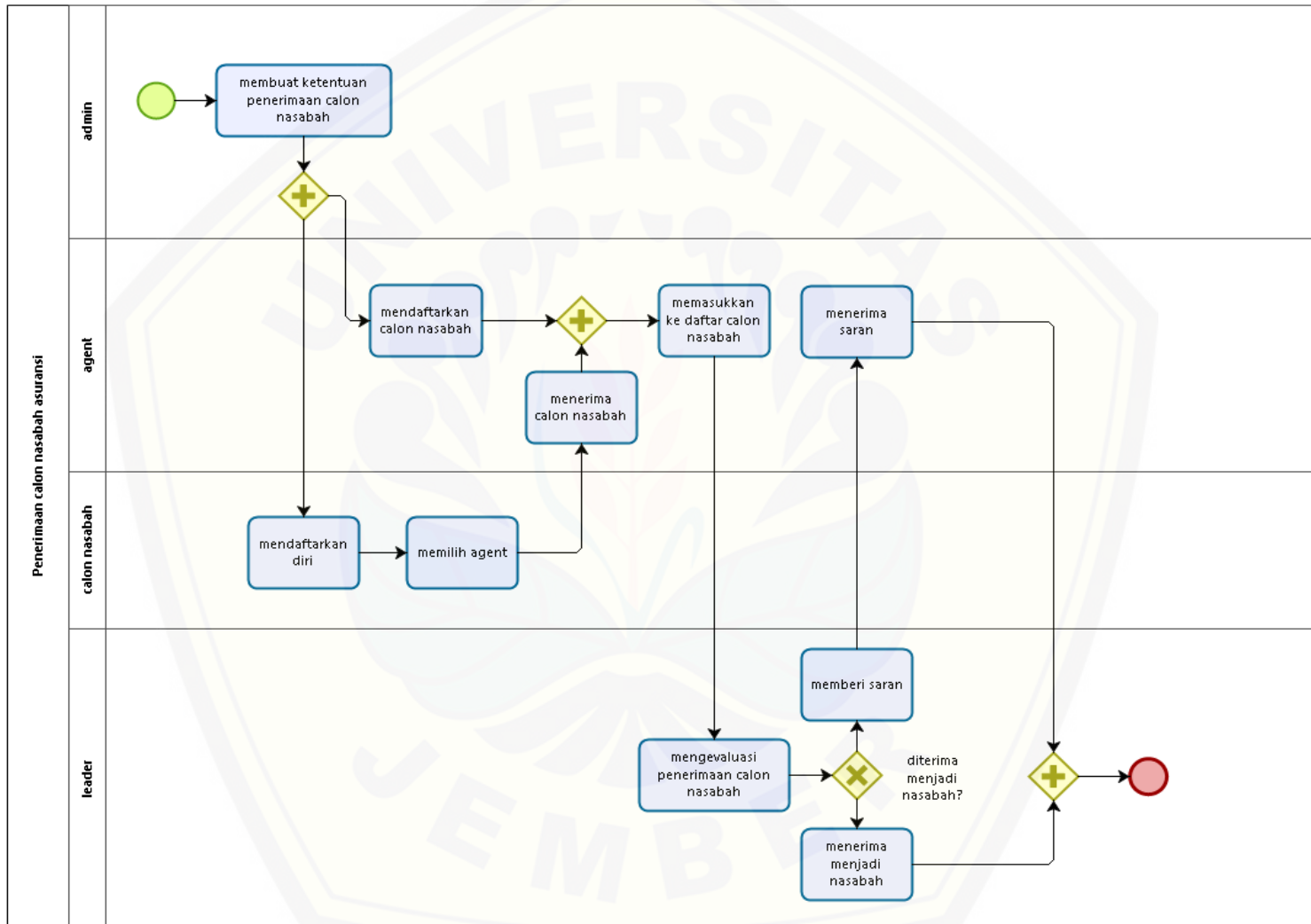
## 4.2 Analisis Kebutuhan

Berdasarkan metode pengembangan sistem *model waterfall*, tahapan awal yang dilakukan adalah tahap analisis. Tahap analisis ini dilakukan terhadap objek penelitian untuk memperoleh kebutuhan sistem yang dibangun, berupa data – data hasil wawancara, kebutuhan fungsional dan non – fungsional. Hasil analisis tersebut sangat mempengaruhi fungsionalitas sistem yang dibangun agar dapat digunakan sesuai fungsi dan kebutuhan pengguna.

### 4.2.1 Pengumpulan Data

Pengumpulan data dilakukan untuk memperoleh data dari objek penelitian secara lengkap dan sesuai yang dibutuhkan untuk proses penelitian. Cara yang dilakukan untuk mendapatkan data tersebut yaitu dengan wawancara. Data yang didapat dalam wawancara yaitu berupa bisnis proses penerimaan calon nasabah asuransi dan data nasabah asuransi yang akan dijadikan sebagai data *set*.

Berikut bisnis proses penerimaan calon nasabah asuransi yang ada pada Perusahaan Asuransi XYZ Jember dapat dilihat pada Gambar 4.1.



Gambar 4.1. Bisnis Proses Penerimaan Calon Nasabah Asuransi

Gambar 4.1 menunjukkan bisnis proses penerimaan calon nasabah asuransi pada Perusahaan Asuransi XYZ Jember yang dilakukan oleh *leader*. Bisnis proses penerimaan calon nasabah asuransi dimulai dari admin yang mempunyai hak untuk membuat ketentuan penerimaan calon nasabah asuransi. Setiap *agent* dapat mendaftarkan calon nasabahnya masing – masing. Begitu pula calon nasabah juga dapat mendaftar sebagai calon nasabah sesuai *agent* yang diinginkan yang nantinya *agent* yang dipilih oleh calon nasabah berhak menerima nasabah. Setelah itu, *leader* akan melakukan proses evaluasi penerimaan calon nasabah asuransi menggunakan ketentuan yang dibuat oleh admin. Hasil evaluasi tersebut digunakan untuk menentukan apakah calon nasabah diterima atau tidak. Apabila di terima, maka calon nasabah menjadi nasabah, jika tidak maka *leader* akan memberikan saran pada *agent* hingga calon nasabah diterima sebagai nasabah. Namun setelah menjadi nasabah, ada nasabah yang masih menunggak dalam pembayaran premi. Hal ini berdampak pada keuangan perusahaan, yaitu terjadinya kerugian pada perusahaan.

Data nasabah yang didapat pada penelitian ini berupa data yang berbentuk *excel*. Pengubahan data nasabah asli ke dalam bentuk data *set* dilakukan dengan cara mengelompokkan data nasabah berdasarkan pengelompokan nilai tiap atribut. Data *set* merupakan kumpulan atribut dan kelasnya. Data *set* dibagi kedalam dua bagian, yaitu data *training* dan data *testing*. Data *training* digunakan pada proses *learning* dalam pembangunan *classification tree*, sedangkan data *testing* digunakan pada proses uji klasifikasi. Dalam pembuatan sistem informasi pengklasifikasian calon nasabah asuransi yang di terapkan pada fitur mengelola data *training* digunakan data nasabah sebesar 90% dari jumlah data *set*, sedangkan pada fitur mengelola data *testing* digunakan data nasabah sebesar 10% dari jumlah data *set*. Atribut yang digunakan meliputi usia, status marital, jenis kelamin, penghasilan, premi, cara bayar, rencana pembayaran, dan masa pembebanan biaya. Kelas yang digunakan yaitu kelancaran. Data *set* dapat dilihat pada Lampiran A. Berikut penjelasan atribut penentuan kelancaran calon nasabah dalam pembayaran premi.

- 1) Usia  
Merupakan atribut usia calon nasabah ketika akan mengikuti program asuransi. Calon nasabah maksimal berusia 60 tahun, dan tidak memiliki persyaratan minimal usia.
- 2) Status marital  
Merupakan atribut status marital yang dimiliki calon nasabah ketika akan mengikuti program asuransi.
- 3) Jenis kelamin  
Merupakan atribut jenis kelamin yang dimiliki calon nasabah
- 4) Penghasilan  
Merupakan atribut penghasilan yang dihasilkan calon nasabah tiap bulannya.
- 5) Premi  
Merupakan atribut jumlah uang yang akan dibayar calon nasabah setiap waktu pembayaran asuransi. Persyaratan minimal premi yang harus dibayar yaitu Rp.200.000.
- 6) Cara Bayar  
Merupakan atribut frekuensi pembayaran premi yang akan dilakukan oleh calon nasabah.
- 7) Rencana Pembayaran  
Merupakan atribut waktu berapa tahun calon nasabah akan membayar angsuran premi, yaitu antara 10 hingga 15 tahun.
- 8) Masa Pembebanan Biaya  
Merupakan atribut rentang waktu pembebanan biaya yang akan didapatkan calon nasabah hingga usia tertentu.

Berikut merupakan tabel pengelompokan nilai atribut yang digunakan sebagai data *set* dalam penelitian ini dapat dilihat pada Tabel 4.1.

Tabel 4.1. Atribut dan Nilai Atribut

No	Atribut	Nilai
1	Usia	Biner (1: $\leq 30$ (Muda) atau 2: $>30$ (Tua))
2	Status Marital	Biner (1: Kawin atau 2: Tidak kawin)
3	Jenis Kelamin	Biner (1: Pria atau 2: Wanita)
4	Penghasilan	Biner (1: $\leq 3$ juta (Rendah) atau 2: $> 3$ juta (Tinggi))
5	Premi	Biner (1: $\leq 2$ juta (Rendah) atau 2: $> 2$ juta (Tinggi))
6	Cara Bayar	Kategori (1: Bulanan, 2: 3 bulanan, atau 3: Tahunan)
7	Rencana Pembayaran	Biner (1: $\geq 10 - \leq 12$ (Cepat) atau 2: $> 12 - \leq 15$ (Lama))
8	Masa Pembebanan Biaya	Biner (1: $\leq 70$ (Cepat) atau 2: $> 70$ (Lama))

Sumber: (Hasil Analisis, 2018)

Kelas pada data *set* yang digunakan pada penelitian ini berupa kelas kelancaran yang merupakan kelancaran dari nasabah yang membayar premi yang terdiri dari dua kategori yaitu lancar dan tidak lancar.

Tabel 4.2. Data Agent

No	Agent
1	Agent A
2	Agent B
3	Agent C

Tabel 4.3. Data Manfaat Asuransi

No	Nama Manfaat	Manfaat
1	<i>PRU Hospital &amp; Surgical Cover Plus</i>	Uang pertanggung dan nilai tunai apabila nasabah masuk rawat inap
2	<i>PRU Link Assurance Account</i>	Uang pertanggung dan nilai tunai apabila nasabah meninggal dunia atau cacat total selama masa asuransi.
3	<i>PRU Saver</i>	Tambahan dana investasi.
4	<i>PRU Cricis Cover Benefit Plus</i>	Uang pertanggung dan nilai tunai apabila nasabah mengalami kondisi krisis tertentu melebihi 90 hari.
5	<i>PRU Link Term</i>	Uang pertanggung dan nilai tunai apabila nasabah meninggal pada saat usia tidak melebihi 70 tahun.



Tabel 4.4. Data Persyaratan Calon Nasabah

No	Persyaratan
1	<i>Memiliki penghasilan</i>
1	Saat mendaftar, maksimal berusia 60 tahun
2	Premi minimal sebesar Rp. 200.000 dihitung perbulan
3	Batas rencana pembayaran yaitu antara 10 - 15 tahun

Tabel 4.2 merupakan tabel yang berisi data nama *agent* yang ada pada Perusahaan Asuransi XYZ Jember. Tabel 4.3 merupakan tabel yang berisi data manfaat asuransi yang didapatkan nasabah asuransi yang ada di Perusahaan Asuransi XYZ Jember. Tabel 4.4 merupakan tabel yang berisi data persyaratan yang harus dipenuhi calon nasabah untuk menjadi nasabah pada Perusahaan Asuransi XYZ Jember.

Berdasarkan hasil pengumpulan data yang telah dilakukan seperti bisnis proses yang terjadi, data nasabah asuransi, data *agent*, data manfaat asuransi, dan data persyaratan calon nasabah asuransi akan dijadikan acuan dalam menentukan kebutuhan fungsional dan kebutuhan non-fungsional sistem informasi pengklasifikasian penerimaan calon nasabah asuransi. Kebutuhan fungsional ini sangat penting untuk menggambarkan layanan-layanan yang bisa diberikan sistem kepada pengguna secara mendetail, sedangkan kebutuhan non-fungsional sistem untuk mendukung aktivitas-aktivitas sesuai dengan kebutuhan fungsional yang ada pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi. Penjelasan lebih detail dari kebutuhan fungsional dan non-fungsional dapat dilihat dibawah ini.

#### 4.2.2 Kebutuhan Fungsional

Kebutuhan fungsional sistem berisi fitur-fitur inti yang harus dipenuhi dalam sistem agar sistem berfungsi sesuai dengan tujuan dan kebutuhan pengguna terhadap sistem. Kebutuhan fungsional dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* yaitu:

1. Sistem mampu mengklasifikasikan penerimaan calon nasabah asuransi berdasarkan data calon nasabah asuransi dan data nasabah asuransi yang dikelola oleh calon nasabah dan *agent* menggunakan algoritma *classification tree* C4.5 dan *cost sensitive learning*.
2. Sistem mampu mengelola data *set* yang akan digunakan untuk proses pembangunan pohon klasifikasi penerimaan calon nasabah asuransi (*lihat, tambah, hapus*).
3. Sistem mampu melihat data *training* dan data *testing*.
4. Sistem mampu menampilkan hasil pembangunan *classification tree* dan *rule*.
5. Sistem mampu melakukan proses *testing* (uji klasifikasi).
6. Sistem mampu memvalidasi penerimaan calon nasabah asuransi.
7. Sistem mampu mengelola data *agent* (*lihat, tambah, ubah, hapus*).
8. Sistem mampu mengelola data calon nasabah asuransi (*lihat, tambah, ubah*).
9. Sistem mampu melihat dan mencetak data nasabah asuransi.
10. Sistem mampu melihat riwayat klasifikasi calon nasabah asuransi.
11. Sistem mampu mengelola data persyaratan calon nasabah asuransi (*lihat, tambah, ubah, hapus*).
12. Sistem mampu mengelola data manfaat asuransi (*lihat, tambah, ubah, hapus*).
13. Sistem mampu menambah aktivitas saran dan penerimaan calon nasabah asuransi.
14. Sistem mampu mengubah data diri.

#### 4.2.3 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional merupakan hal-hal yang dibutuhkan untuk mendukung aktivitas sistem dalam memenuhi kebutuhan fungsionalitasnya. Kebutuhan non-fungsional menitikberatkan pada properti perilaku oleh sistem. Kebutuhan non-fungsional dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree* C4.5 *algorithm* dan *cost sensitive learning* yaitu:

1. Sistem dapat dijalankan oleh beberapa *software web browser*, seperti *Internet Explore*, *Google Chrome*, dan *Mozilla Firefox*.
2. Sistem memiliki tampilan dan bahasa komunikasi sistem yang mudah dipahami oleh pengguna untuk memberikan kenyamanan pemakaian dan memudahkan pengaplikasiannya.
3. Sistem dapat digunakan oleh banyak pengguna pada saat yang bersamaan pada komputer yang berbeda.

#### 4.2.4 Fungsi Sistem

Fungsi utama dari sistem yang dibangun dalam penelitian ini terletak pada fitur *login* sesuai dengan hak akses dari setiap pengguna dalam sistem. Ketika pengguna melakukan *login*, maka sistem melakukan autentifikasi *username* dan *password* pengguna. Selanjutnya sistem akan menyajikan tampilan sistem sesuai dengan hak akses pengguna yang meliputi:

1. *Leader*

*Leader* merupakan pengguna sistem yang telah terdaftar, dan ketika *login* berhasil maka sistem akan menampilkan halaman *dashboard* dan halaman-halaman yang sesuai dengan level satu. *Leader* dapat mengakses halaman *agent*, halaman calon nasabah asuransi (yang akan diklasifikasi), halaman data nasabah asuransi, halaman riwayat klasifikasi, halaman data persyaratan calon nasabah asuransi, dan halaman data manfaat asuransi.

2. *Agent*

*Agent* merupakan pengguna sistem yang telah terdaftar, dan ketika *login* berhasil maka sistem akan menampilkan halaman *dashboard* dan halaman-halaman yang sesuai dengan level dua. *Agent* dapat mengakses halaman antrian calon nasabah asuransi (calon nasabah yang mendaftar sendiri), halaman calon nasabah asuransi (data calon nasabah masing-masing *agent*), halaman data nasabah asuransi (data nasabah masing-masing *agent*), halaman data aktivitas saran, halaman data persyaratan calon nasabah asuransi, dan halaman data manfaat asuransi. Dalam hal ini *agent* hanya mengelola data asuransi masing-masing, satu *agent* hanya dapat mengelola nasabah maupun calon nasabah miliknya.

3. Calon Nasabah

Calon nasabah merupakan pengguna sistem yang telah terdaftar, dan ketika *login* berhasil maka sistem akan menampilkan halaman *dashboard* dan halaman-halaman yang sesuai dengan level tiga. Sedangkan pengguna yang belum terdaftar, setelah melakukan register maka akan bisa melakukan *login*. Calon nasabah dapat mengakses halaman data diri, halaman daftar *agent* (jika belum daftar *agent*, maka dapat melakukan pendaftaran *agent*), halaman aktivitas pendaftaran, halaman data persyaratan calon nasabah asuransi, dan halaman data manfaat asuransi.

4. Admin

Admin merupakan pengguna sistem yang telah terdaftar, dan ketika *login* berhasil maka sistem akan menampilkan halaman *dashboard* dan halaman-halaman yang sesuai dengan level empat. Admin dapat mengakses data set, data *training* (yang

akan digunakan untuk pembuatan pohon klasifikasi) dan data *testing* (yang digunakan untuk uji klasifikasi *classification tree* yang telah dibangun).

### 4.3 Desain Sistem

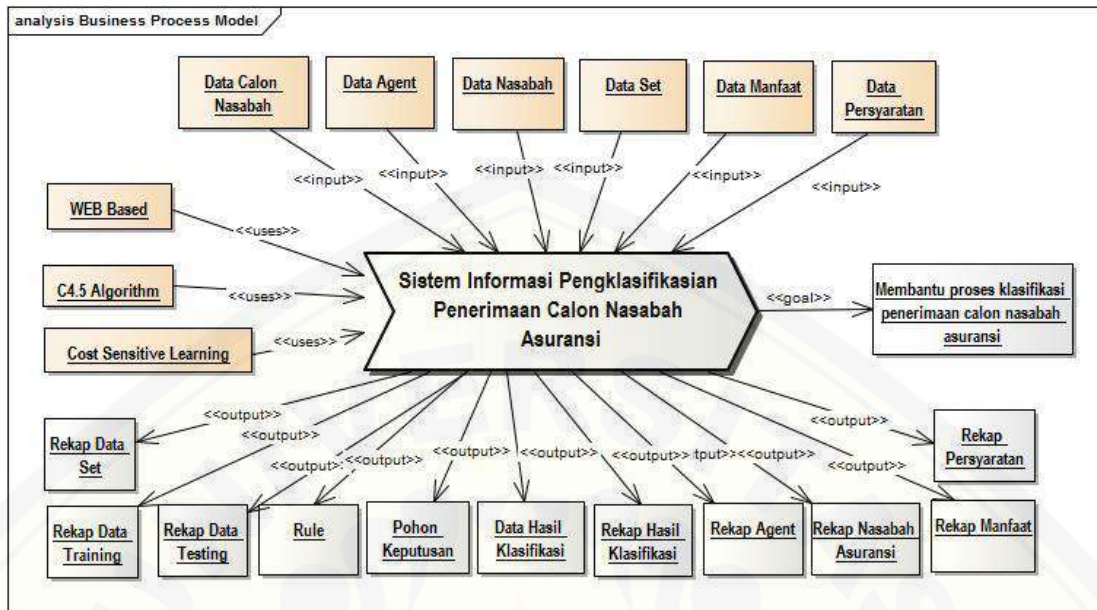
Tahapan yang dilakukan setelah melakukan analisis kebutuhan sistem yaitu tahap perencanaan pembangunan sistem yang digambarkan dengan desain sistem. Desain sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* ini meliputi *use case diagram*, *use case skenario*, *activity diagram*, *sequence diagram*, *class diagram*, dan *entity relationship diagram*.

#### 4.3.1 Elemen *Input* dan *Output*

Elemen *input* dan *output* merupakan *diagram* yang menggambarkan kebutuhan elemen yang terdiri dari elemen *input* atau masukkan ke sistem, elemen *output* atau keluaran dari sistem, *goal* atau tujuan dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi dan *uses* atau media yang digunakan untuk membuat sistem. Berikut ini penjelasan mengenai elemen yang ada pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi. Elemen *input* dan *output* terdiri atas beberapa poin diantaranya:

- a. *Input* : Data yang dimasukkan ke dalam sistem
- b. *Output* : Data yang dihasilkan oleh sistem
- c. *Goal* : Tujuan dibangun suatu sistem
- d. *Uses* : *Platform* yang menjadi basis sistem
- e. *Process* : Sistem yang bekerja

Berikut elemen *input* dan *output* pada sistem ini dapat dilihat pada Gambar 4.2.

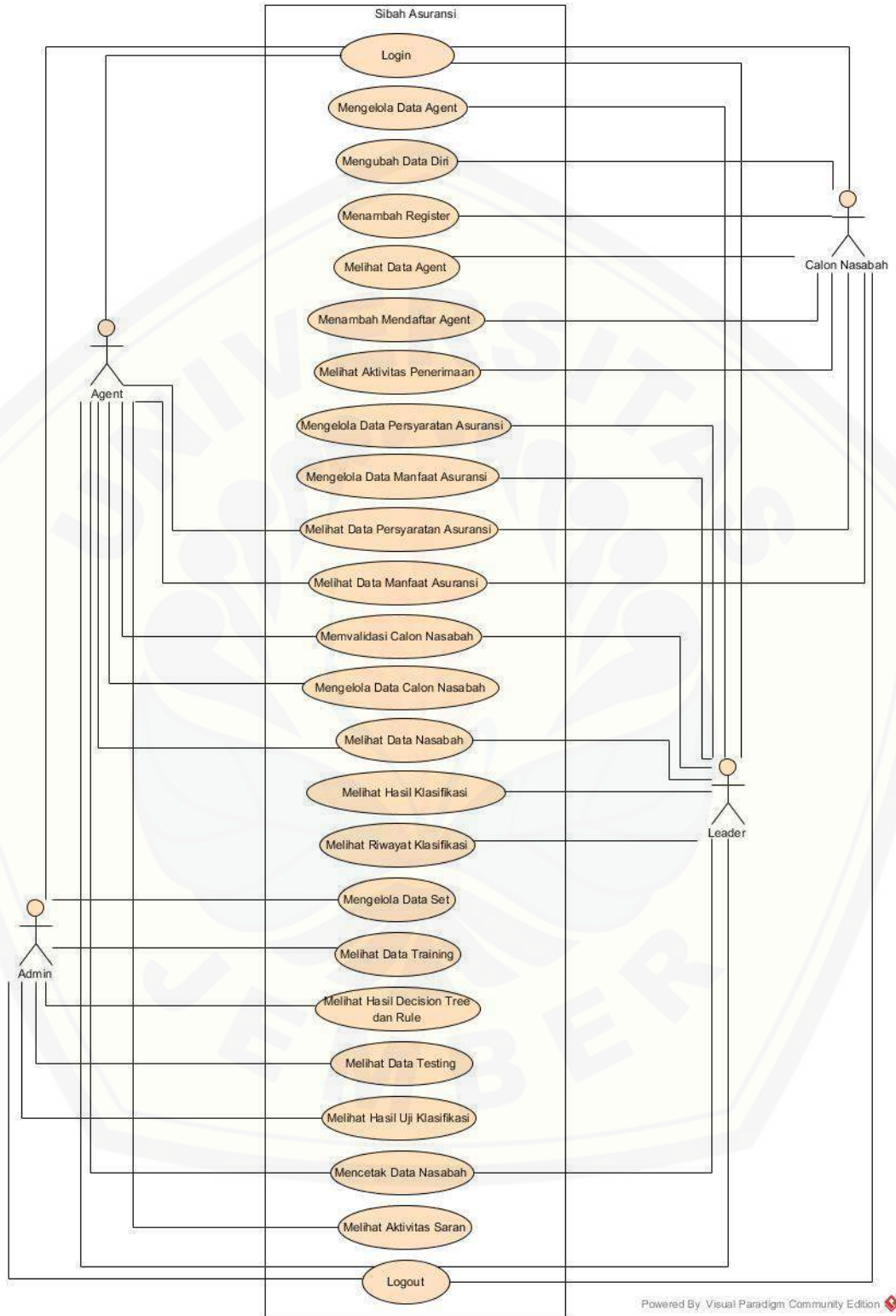


Gambar 4.2. Elemen *input* dan *output* Sibah Asuransi

Gambar 4.2 merupakan elemen *input* dan *output* dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree* C4.5 *algorithm* dan *cost sensitive learning*. Elemen *input* dan *output* menjelaskan proses *input*, *output*, *goal* dan *uses* yang diaplikasikan pada sistem.

#### 4.3.2 Use Case Diagram

*Use case diagram* merupakan pemodelan yang digunakan untuk menggambarkan interaksi antara aktor dengan sistem yang dibangun. Melalui *use case diagram* dapat diketahui interaksi yang dapat dilakukan aktor sesuai dengan hak akses yang dimiliki oleh masing-masing aktor atau pengguna. *Use case diagram* ditunjukkan pada Gambar 4.2.



Gambar 4.3. Use Case Diagram Sibah Asuransi

Gambar 4.3 menunjukkan *usecase diagram* sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* yang terdiri dari empat aktor dengan dua puluh empat *use case*.

Berdasarkan *use case diagram* pada Gambar 4.3 terdapat empat aktor atau pengguna, yaitu *Leader*, *Agent*, Calon Nasabah, dan Admin. Adapun deskripsi dari masing-masing aktor dapat dilihat pada Tabel 4.5.

Tabel 4.5. Deskripsi Aktor

No.	Aktor	Deskripsi
1.	<i>Leader</i>	Aktor <i>leader</i> merupakan aktor yang memiliki hak akses untuk melakukan <i>login</i> , mengelola data <i>agent</i> , menambah klasifikasi data calon nasabah, validasi calon nasabah, melihat data nasabah, melihat riwayat klasifikasi, mengelola data persyaratan calon nasabah asuransi, dan mengelola data manfaat asuransi.
2.	<i>Agent</i>	Aktor <i>agent</i> merupakan aktor yang memiliki hak akses untuk melakukan <i>login</i> , memvalidasi calon nasabah, mengelola data calon nasabah, melihat data nasabah, melihat aktivitas saran, melihat data persyaratan calon nasabah asuransi, dan melihat data manfaat asuransi.
3.	Calon nasabah	Aktor calon nasabah merupakan aktor yang memiliki hak akses untuk tambah register, melakukan <i>login</i> , tambah mendaftar <i>agent</i> , mengubah data diri, melihat data <i>agent</i> , dan melihat aktivitas penerimaan
4.	Admin	Aktor admin merupakan aktor yang memiliki hak akses untuk mengelola data set, melihat data <i>training</i> , melihat data <i>testing</i> , melihat hasil <i>classification tree</i> dan <i>rule</i> , dan uji klasifikasi.

Selain memiliki empat aktor, dalam *use case diagram* juga terdapat dua puluh empat *use case*. Deskripsi dari *use case* terdapat pada pada Tabel 4.6.



Tabel 4.6. Deskripsi *Use Case*

No.	<i>Usecase</i>	Penjelasan
1.	<i>Login</i>	Merupakan <i>usecase</i> yang menggambarkan proses <i>login</i> atau autentifikasi untuk masuk ke system
2.	Mengelola data <i>agent</i>	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah, dan menghapus data <i>agent</i> .
3.	Mengubah data diri	Merupakan <i>usecase</i> yang menggambarkan proses mengubah data diri masing-masing calon nasabah.
4.	Menambah data <i>register</i>	Merupakan <i>usecase</i> yang menggambarkan proses menambah data <i>register</i> .
5.	Melihat data <i>agent</i>	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data <i>agent</i>
6.	Menambah mendaftar <i>agent</i>	Merupakan <i>usecase</i> yang menggambarkan proses pendaftaran <i>agent</i> yang dilakukan oleh calon nasabah yang telah melakukan proses menambah data <i>register</i> .
7.	Melihat aktivitas penerimaan	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data aktivitas penerimaan
8.	Mengelola data persyaratan calon nasabah asuransi	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah, dan menghapus data persyaratan.
9.	Mengelola data manfaat asuransi	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah, dan menghapus data manfaat asuransi.
10.	Melihat data persyaratan calon nasabah asuransi	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data persyaratan.
11.	Melihat data manfaat asuransi	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data manfaat asuransi.
12.	Validasi calon nasabah	Merupakan <i>usecase</i> yang menggambarkan proses verifikasi penerimaan calon nasabah
13.	Mengelola data calon nasabah	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan, menambah, mengubah, dan menghapus data calon nasabah.
14.	Melihat data nasabah	Merupakan <i>usecase</i> yang menggambarkan proses menampilkan data nasabah.

No.	Usecase	Penjelasan
15.	Melihat hasil klasifikasi	Merupakan usecase yang menggambarkan proses menampilkan data hasil klasifikasi..
16.	Melihat riwayat kasifikasi	Merupakan usecase yang menggambarkan proses menampilkan riwayat klasifikasi.
17.	Mengelola data set	Merupakan usecase yang menggambarkan proses menampilkan, menambah, dan menghapus data set.
18.	Melihat data <i>training</i>	Merupakan usecase yang menggambarkan proses menampilkan data <i>training</i> .
19.	Melihat hasil <i>classification tree</i> dan <i>rule</i>	Merupakan usecase yang menggambarkan proses menampilkan hasil pembangunan <i>classification tree</i> dan <i>rule</i>
20.	Melihat data <i>testing</i>	Merupakan usecase yang menggambarkan proses menampilkan data <i>testing</i> .
21.	Melihat hasil uji klasifikasi	Merupakan usecase yang menggambarkan proses menampilkan hasil uji klasifikasi atau proses <i>testing</i> .
22.	Mencetak data nasabah	Merupakan usecase yang menggambarkan proses mencetak data nasabah.
23.	Melihat aktivitas saran	Merupakan usecase yang menggambarkan proses menampilkan aktivitas saran.
24.	<i>Logout</i>	Merupakan usecase yang menggambarkan proses <i>logout</i> atau keluar dari sistem.

#### 4.3.3 Use Case Skenario

*Usecase* skenario adalah dokumentasi terhadap kebutuhan fungsional sistem yang menjelaskan alur yang menggambarkan aksi aktor dan reaksi sistem. *Use case* skenario dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree* C4.5 *algorithm* dan *cost sensitive learning* adalah sebagai berikut:

##### 1. Skenario *Login*

Penjelasan urutan aksi aktor untuk melakukan *login* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

2. Skenario Mengelola Data *Agent*

Penjelasan urutan aksi aktor untuk mengelola data *agent* meliputi (lihat, tambah, ubah, hapus) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

3. Skenario Mengubah Data Diri

Penjelasan urutan aksi aktor untuk mengubah data diri masing-masing calon nasabah dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

4. Skenario Menambah Data Register

Penjelasan urutan aksi aktor untuk menambah data register masing-masing calon nasabah dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

5. Skenario Melihat Data *Agent*

Penjelasan urutan aksi aktor untuk melihat data *agent* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

6. Skenario Menambah Mendaftar *Agent*

Penjelasan urutan aksi aktor untuk mendaftar *agent* masing-masing calon nasabah dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.7.

Tabel 4.7. Skenario Menambah Mendaftar *Agent*

Nomor <i>Usecase</i>	USC 06
Nama	Menambah mendaftar <i>Agent</i>
Aktor	Calon Nasabah
<i>Pre Condition</i>	Calon nasabah memilih menu Daftar <i>Agent</i> dan belum memiliki <i>agent</i>
<i>Post Condition</i>	Calon nasabah berhasil mendaftar <i>agent</i>
<b>SKENARIO UTAMA MENAMBAH MENDAFTAR AGENT</b>	

Aktor	Sistem
1. Memilih menu Daftar <i>Agent</i>	2. Mengambil data <i>agent</i> dari <i>database</i>
	3. Menampilkan data <i>agent</i> pada tabel di halaman data <i>agent</i> : <ul style="list-style-type: none"> <li>- No (<i>int: 11</i>)</li> <li>- Nama <i>Agent</i> (<i>varchar: 55</i>)</li> <li>- No. Identitas (<i>int: 55</i>)</li> <li>- Alamat (<i>varchar: 255</i>)</li> <li>- <i>Actions</i> yang memuat 1 tombol yaitu               <ul style="list-style-type: none"> <li>• Daftar: untuk mendaftarkan <i>agent</i> yang dipilih</li> </ul> </li> </ul> Halaman ini juga menampilkan kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.
4. Menekan tombol ikon “ <i>Daftar</i> ” pada baris data yang akan dipilih	
	5. Menampilkan <i>form</i> modal Mendaftar <i>Agent</i> yang terdiri dari: <ul style="list-style-type: none"> <li>- ID <i>Agent</i> (<i>int: 11</i>)</li> <li>- Jenis Kelamin (<i>enum: 'Pria', 'Wanita'</i>)</li> <li>- Status Marital (<i>enum: 'Kawin', 'Tidak kawin'</i>)</li> <li>- Tempat Lahir (<i>varchar: 55</i>)</li> <li>- Tanggal Lahir (<i>date</i>)</li> <li>- Pekerjaan (<i>varchar: 55</i>)</li> <li>- Penghasilan (<i>double</i>)</li> <li>- Premi (<i>double</i>)</li> <li>- Cara Bayar (<i>enum: 'Bulanan', '3 Bulanan', 'Tahunan'</i>)</li> <li>- Masa Pembebanan Biaya (<i>int: 11</i>)</li> <li>- Rencana pembayaran (<i>int: 11</i>)</li> <li>- KTP (<i>varchar: 255</i>)</li> <li>- Slip Gaji (<i>varchar: 255</i>)</li> <li>- Tombol Kirim</li> </ul>
6. Mengisi seluruh isian <i>form</i> modal mendaftarkan <i>agent</i> : <ul style="list-style-type: none"> <li>- Jenis Kelamin (<i>enum: 'Pria', 'Wanita'</i>)</li> </ul>	

- Status Marital (*enum*: 'Kawin', 'Tidak kawin')
- Tempat Lahir (*varchar*: 55)
- Tanggal Lahir (*date*)
- Pekerjaan (*varchar*: 55)
- Penghasilan (*double*)
- Premi (*double*)
- Cara Bayar (*enum*: 'Bulanan', '3 bulanan', 'Tahunan')
- Masa Pembebanan Biaya (*int*: 11)
- Rencana pembayaran (*int*: 11)
- KTP (*varchar*: 255)
- Slip Gaji (*varchar*: 255)

7. Menekan tombol “Kirim”

8. Menampilkan *pop up* “Apakah data akan benar - benar disimpan?”

9. Menekan tombol “Oke”

10. Menyimpan masukan data dan menyimpan data aktivitas penerimaan pada tabel aktivitas penerimaan:

- *Idnasabah* (*int*: 11)
- *aktivitas* (*varchar*: 255) = "Mendaftar Nasabah"
- *statusktivitas* (*int*: 11) = "1"
- *tglaktivitas* (*timestamp*)
- *ket* (*text*) = "Anda sudah mendaftar sebagai calon nasabah. Saat ini sedang menunggu persetujuan *agent*" ke *database*

11. Mengambil data *agent* dari database

12. Menampilkan data *agent* pada tabel di halaman data *agent*

13. Menampilkan alert “Data Berhasil di Simpan”

**SKENARIO UTAMA BATAL TAMBAH MENDAFTAR AGENT KEMBALI KE FORM MODAL MENDAFTAR AGENT**

**Aktor**

**Sistem**

9. Menekan tombol “Batal”

10. Menampilkan kembali *form* modal mendaftar *agent*.

**SKENARIO ALTERNATIF MENAMBAH MENDAFTAR AGENT**

Jika aktor tidak mengisi *form* modal tambah data *mendaftar agent* secara lengkap

**Aktor**

**Sistem**

6a. Mengisi isian *form* modal mendaftar *agent* tidak lengkap

7a. Menekan tombol “Kirim”	8a. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
9a. Menekan tombol “Oke”	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> tambah data <i>agent</i> . Pesan akan tampil disetiap <i>field</i> yang kosong

#### 7. Skenario Melihat Aktivitas Penerimaan

Penjelasan urutan aksi aktor untuk melihat aktivitas penerimaan masing-masing calon nasabah dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

#### 8. Skenario Mengelola Data Persyaratan Calon Nasabah Asuransi

Penjelasan urutan aksi aktor untuk mengelola data persyaratan calon nasabah asuransi meliputi (lihat, tambah, ubah, hapus) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

#### 9. Skenario Mengelola Data Manfaat Asuransi

Penjelasan urutan aksi aktor untuk mengelola data manfaat asuransi meliputi (lihat, tambah, ubah, hapus) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

#### 10. Skenario Melihat Data Persyaratan Calon Nasabah Asuransi

Penjelasan urutan aksi aktor untuk melihat data persyaratan calon nasabah asuransi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

### 11. Skenario Melihat Data Manfaat Asuransi

Penjelasan urutan aksi aktor untuk melihat data manfaat asuransi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

### 12. Skenario Validasi Calon Nasabah Asuransi

Penjelasan urutan aksi aktor untuk memvalidasi penerimaan calon nasabah dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

### 13. Skenario Mengelola Data Calon Nasabah Asuransi

Penjelasan urutan aksi aktor untuk mengelola data calon nasabah meliputi (lihat, tambah, ubah) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.8.

Tabel 4.8. Skenario Mengelola Data Calon Nasabah

Nomor <i>Usecase</i>	USC 13
Nama	Mengelola Data Calon Nasabah Asuransi
Aktor	<i>Agent</i>
<i>Pre Condition</i>	<i>Agent</i> memilih menu Data Calon Nasabah
<i>Post Condition</i>	<i>Agent</i> berhasil melihat, menambah, dan mengubah data calon nasabah
<b>SKENARIO UTAMA MELIHAT DATA CALON NASABAH</b>	
<b>Aktor</b>	<b>Sistem</b>
1. Memilih menu Calon Nasabah	2. Mengambil data Calon Nasabah dengan ID <i>Agent</i> = ID <i>UserLogin</i> dari <i>database</i>
	3. Menampilkan data Calon Nasabah pada tabel di halaman data Calon Nasabah: <ul style="list-style-type: none"> <li>- ID Nasabah (<i>int</i>: 11)</li> <li>- No. Identitas (<i>int</i>: 55)</li> <li>- Nama (<i>varchar</i>: 55)</li> <li>- KTP (<i>varchar</i>: 255)</li> <li>- Slip Gaji (<i>varchar</i>: 255)</li> <li>- <i>Actions</i> yang memuat 2 tombol yaitu:</li> </ul>

- lihat: untuk melihat data calon nasabah
- ubah: untuk mengubah data calon nasabah

Halaman ini juga menampilkan tombol tambah data yang digunakan apabila aktor akan menambahkan data calon nasabah, serta kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.

4. Menekan tombol ikon “*Lihat*” pada baris data yang akan dilihat

5. Menampilkan *form* modal lihat data calon nasabah berdasarkan yang dipilih:

- Nama (*varchar: 55*)
- No. KTP (*int: 55*)
- Alamat (*varchar: 255*)
- Jenis Kelamin (*enum: 'Pria', 'Wanita'*)
- Status Marital (*enum: 'Kawin', 'Tidak kawin'*)
- Tempat Lahir (*varchar: 55*)
- Tanggal Lahir (*date*)
- Pekerjaan (*varchar: 55*)
- Penghasilan (*double*)
- Premi (*double*)
- Cara Bayar (*enum: 'Bulanan', '3 bulanan', 'Tahunan'*)
- Masa Pembebanan Biaya (*int: 11*)
- Rencana pembayaran (*int: 11*)

#### SKENARIO UTAMA MENAMBAH DATA CALON NASABAH

Aktor	Sistem
4. Menekan tombol “Tambahkan”	5. Menampilkan <i>form</i> modal tambah data calon nasabah: <ul style="list-style-type: none"> <li>- Nama (<i>varchar: 55</i>)</li> <li>- No. KTP (<i>int: 55</i>)</li> <li>- Alamat (<i>varchar: 255</i>)</li> <li>- Jenis Kelamin (<i>enum: 'Pria', 'Wanita'</i>)</li> <li>- Status Marital (<i>enum: 'Kawin', 'Tidak kawin'</i>)</li> <li>- Tempat Lahir (<i>varchar: 55</i>)</li> <li>- Tanggal Lahir (<i>date</i>)</li> <li>- Pekerjaan (<i>varchar: 55</i>)</li> </ul>



	<ul style="list-style-type: none"> <li>- Penghasilan (<i>double</i>)</li> <li>- Premi (<i>double</i>)</li> <li>- Cara Bayar (<i>enum</i>: 'Bulanan', '3 bulanan', 'Tahunan')</li> <li>- Masa Pembebanan Biaya (<i>int</i>: 11)</li> <li>- Rencana pembayaran (<i>int</i>: 11)</li> <li>- KTP (<i>varchar</i>: 255)</li> <li>- Slip Gaji (<i>varchar</i>: 255)</li> <li>- Tombol Kirim</li> </ul>
6. Mengisi seluruh isian <i>form</i> modal tambah calon nasabah:	
<ul style="list-style-type: none"> <li>- Nama (<i>varchar</i>: 55)</li> <li>- No. KTP (<i>int</i>: 55)</li> <li>- Alamat (<i>varchar</i>: 255)</li> <li>- Jenis Kelamin (<i>enum</i>: 'Pria', 'Wanita')</li> <li>- Status Marital (<i>enum</i>: 'Kawin', 'Tidak kawin')</li> <li>- Tempat Lahir (<i>varchar</i>: 55)</li> <li>- Tanggal Lahir (<i>date</i>)</li> <li>- Pekerjaan (<i>varchar</i>: 55)</li> <li>- Penghasilan (<i>double</i>)</li> <li>- Premi (<i>double</i>)</li> <li>- Cara Bayar (<i>enum</i>: 'Bulanan', '3 bulanan', 'Tahunan')</li> <li>- Masa Pembebanan Biaya (<i>int</i>: 11)</li> <li>- Rencana pembayaran (<i>int</i>: 11)</li> <li>- KTP (<i>varchar</i>: 255)</li> <li>- Slip Gaji (<i>varchar</i>: 255)</li> </ul>	
7. Menekan tombol "Kirim"	8. Menampilkan <i>pop up</i> "Apakah data akan benar - benar disimpan?"
9. Menekan tombol "Oke"	10. Menyimpan masukan data ke <i>database</i> dan menyimpan data aktivitas penerimaan pada tabel aktivitas penerimaan: <ul style="list-style-type: none"> <li>- Idnasabah (<i>int</i>: 11)</li> <li>- aktivitas (<i>varchar</i>: 255) = "Lolos tahap <i>agent</i>"</li> <li>- statusktivitas (<i>int</i>: 11) = "1"</li> <li>- tglaktivitas (<i>timestamp</i>)</li> <li>- ket (<i>text</i>) = "Anda sudah lolos sebagai calon nasabah di <i>agent</i> yang anda pilih. Saat ini sedang menunggu persetujuan <i>leader</i>"</li> </ul>

	11. Mengambil data calon nasabah dengan ID <i>Agent</i> = ID <i>UserLogin</i> dari database
	12. Menampilkan data calon nasabah pada tabel di halaman data calon nasabah
	13. Menampilkan alert “Data Berhasil di Simpan”

**SKENARIO UTAMA BATAL TAMBAH DATA CALON NASABAH KEMBALI KE FORM MODAL TAMBAH DATA CALON NASABAH**

Aktor	Sistem
9. Menekan tombol “Batal”	10. Menampilkan <i>form</i> modal tambah data calon nasabah

**SKENARIO UTAMA MENGUBAH DATA CALON NASABAH**

Aktor	Sistem
4. Menekan tombol ikon “Ubah” pada baris data yang akan diubah	5. Mengambil data calon nasabah yang dipilih dari <i>database</i>
	6. Menampilkan <i>form</i> modal data calon nasabah yang telah terisi dengan data sebelumnya: <ul style="list-style-type: none"> <li>- Nama (<i>varchar: 55</i>)</li> <li>- No. KTP (<i>int: 55</i>)</li> <li>- Alamat (<i>varchar: 255</i>)</li> <li>- Jenis Kelamin (<i>enum: 'Pria', 'Wanita'</i>)</li> <li>- Status Marital (<i>enum: 'Kawin', 'Tidak kawin'</i>)</li> <li>- Tempat Lahir (<i>varchar: 55</i>)</li> <li>- Tanggal Lahir (<i>date</i>)</li> <li>- Pekerjaan (<i>varchar: 55</i>)</li> <li>- Penghasilan (<i>double</i>)</li> <li>- Premi (<i>double</i>)</li> <li>- Cara Bayar (<i>enum: 'Bulanan', '3 bulanan', 'Tahunan'</i>)</li> <li>- Masa Pembebanan Biaya (<i>int: 11</i>)</li> <li>- Rencana pembayaran (<i>int: 11</i>)</li> <li>- Tombol Kirim</li> </ul>
7. Mengubah isian <i>form</i> modal ubah data calon nasabah: <ul style="list-style-type: none"> <li>- Jenis Kelamin (<i>enum: 'Pria', 'Wanita'</i>)</li> <li>- Status Marital (<i>enum: 'Kawin', 'Tidak kawin'</i>)</li> <li>- Tempat Lahir (<i>varchar: 55</i>)</li> <li>- Tanggal Lahir (<i>date</i>)</li> </ul>	

- Pekerjaan ( <i>varchar: 55</i> )	
- Penghasilan ( <i>double</i> )	
- Premi ( <i>double</i> )	
- Cara Bayar ( <i>enum: 'Bulanan', '3 bulanan', 'Tahunan'</i> )	
- Masa Pembebanan Biaya ( <i>int: 11</i> )	
- Rencana pembayaran ( <i>int: 11</i> )	
8. Menekan tombol “Kirim”	9. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
10. Menekan tombol “Oke”	11. Menyimpan masukan data ke <i>database</i>
	12. Mengambil data calon nasabah dengan $ID\ Agent = ID\ UserLogin$ dari <i>database</i>
	13. Menampilkan data calon nasabah pada tabel di halaman data calon nasabah
	14. Menampilkan alert “Data Berhasil di Ubah”

#### SKENARIO UTAMA BATAL UBAH DATA CALON NASABAH KEMBALI KE FORM MODAL UBAH DATA CALON NASABAH

Aktor	Sistem
9. Menekan tombol “Batal”	10. Menampilkan <i>form</i> modal data calon nasabah yang telah terisi dengan data sebelumnya

#### SKENARIO ALTERNATIF MENAMBAH DATA CALON NASABAH

Jika aktor tidak mengisi *form* tambah data calon nasabah secara lengkap

Aktor	Sistem
6a. Mengisi isian <i>form</i> modal tambah data calon nasabah tidak lengkap	
7a. Menekan tombol “Kirim”	8a. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
9a. Menekan tombol “Oke”	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> tambah data calon nasabah. Pesan akan tampil di setiap <i>field</i> yang kosong

#### SKENARIO ALTERNATIF MENGUBAH DATA CALON NASABAH

Jika aktor tidak mengisi *form* ubah data calon nasabah secara lengkap

Aktor	Sistem
7b. Mengisi isian <i>form</i> modal ubah data calon nasabah tidak lengkap	
8b. Menekan tombol “Kirim”	

	9b. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
10b. Menekan tombol “Oke”	
	11b. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> ubah data calon nasabah. Pesan akan tampil disetiap <i>field</i> yang kosong
<b>SKENARIO ALTERNATIF MENAMBAH DATA CALON NASABAH</b>	
Jika nomor identitas calon nasabah atau <i>username</i> sudah dipakai	
<b>Aktor</b>	<b>Sistem</b>
6c. Mengisi isian <i>form</i> modal tambah data calon nasabah	
7c. Menekan tombol “Kirim”	
	8c. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
9c. Menekan tombol “Oke”	
	10c. Memeriksa nomor identitas dan <i>username</i> yang pada <i>form</i> modal tambah data calon nasabah
	11c. Mengambil data calon nasabah yang dari <i>database</i>
	12c. Menampilkan data calon nasabah pada tabel di halaman data calon nasabah
	13c. Menampilkan alert “No KTP/ <i>Username</i> tidak boleh sama”

14. Skenario Melihat Data Nasabah Asuransi

Penjelasan urutan aksi aktor untuk melihat data nasabah asuransi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase* Skenario).

15. Skenario Melihat Hasil Klasifikasi

Penjelasan urutan aksi aktor untuk melihat data hasil klasifikasi calon nasabah dan memvalidasi calon nasabah asuransi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.9.

Tabel 4.9. Skenario Melihat Hasil Klasifikasi

Nomor <i>Usecase</i>	USC 15
Nama	Melihat Hasil Klasifikasi
Aktor	<i>Leader</i>

<i>Pre Condition</i>	<i>Leader</i> memilih menu Calon Nasabah
<i>Post Condition</i>	<i>Leader</i> berhasil melihat hasil klasifikasi dan memvalidasi calon nasabah
<b>SKENARIO UTAMA MELIHAT HASIL KLASIFIKASI</b>	
<b>Aktor</b>	<b>Sistem</b>
1. Memilih menu Calon Nasabah	2. Mengambil data calon nasabah dari <i>database</i>
	3. Menampilkan data calon nasabah pada tabel di halaman data calon nasabah: <ul style="list-style-type: none"> <li>- ID Nasabah (<i>int: 11</i>)</li> <li>- No. Identitas (<i>int: 55</i>)</li> <li>- Nama Nasabah (<i>varchar: 55</i>)</li> <li>- ID Agent (<i>int: 11</i>)</li> <li>- KTP (<i>varchar: 255</i>)</li> <li>- Slip Gaji (<i>varchar: 255</i>)</li> <li>- <i>Actions</i> yang memuat 1 tombol yaitu: <ul style="list-style-type: none"> <li>• lihat: untuk melihat data calon nasabah, melihat hasil klasifikasi, dan memvalidasi calon nasabah</li> </ul> </li> </ul> <p>Halaman ini juga menampilkan kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.</p>
4. Menekan tombol ikon “ <i>Lihat</i> ” pada baris data yang akan dilihat	5. Mengambil data calon nasabah yang dipilih dari <i>database</i>
	6. Menampilkan <i>form</i> lihat data calon nasabah yang sudah terisi sebelumnya: <ul style="list-style-type: none"> <li>- ID Nasabah (<i>int: 11</i>)</li> <li>- Nama (<i>varchar: 55</i>)</li> <li>- No. KTP (<i>int: 55</i>)</li> <li>- Alamat (<i>varchar: 255</i>)</li> <li>- Jenis Kelamin (<i>enum: 'Pria', 'Wanita'</i>)</li> <li>- Status Marital (<i>enum: 'Kawin', 'Tidak kawin'</i>)</li> <li>- Tempat Lahir (<i>varchar: 55</i>)</li> <li>- Tanggal Lahir (<i>date</i>)</li> <li>- Pekerjaan (<i>varchar: 55</i>)</li> <li>- Penghasilan (<i>double</i>)</li> <li>- Premi (<i>double</i>)</li> </ul>

	<ul style="list-style-type: none"> <li>- Cara Bayar (<i>enum</i>: 'Bulanan', '3 bulanan', 'Tahunan')</li> <li>- Masa Pembebanan Biaya (<i>int</i>: 11)</li> <li>- Rencana pembayaran (<i>int</i>: 11)</li> <li>- Tombol Klasifikasi</li> </ul>
7. Menekan tombol “Klasifikasi”	
	8. Mengubah data menjadi data test
	9. Menghitung data pada <i>rule</i> dari hasil konversi <i>classification tree</i> yang dibuat menggunakan algoritma C4.5 dan <i>cost sensitive learning</i>
	10. Menyimpan hasil klasifikasi ke <i>database</i>
	11. Mengambil data hasil klasifikasi dari <i>database</i>
	12. Menampilkan data <i>form</i> data calon nasabah yang sudah terisi sebelumnya dan hasil klasifikasi pada form hasil klasifikasi di halaman data hasil klasifikasi: Data Calon Nasabah <ul style="list-style-type: none"> <li>- ID Nasabah (<i>int</i>: 11)</li> <li>- Nama nasabah (<i>varchar</i>: 55)</li> <li>- No. KTP (<i>int</i>: 55)</li> <li>- Alamat (<i>varchar</i>: 255)</li> <li>- Jenis Kelamin (<i>enum</i>: 'Pria', 'Wanita')</li> <li>- Status Marital (<i>enum</i>: 'Kawin', 'Tidak kawin')</li> <li>- Tempat Lahir (<i>varchar</i>: 55)</li> <li>- Tanggal Lahir (<i>date</i>)</li> <li>- Pekerjaan (<i>varchar</i>: 55)</li> <li>- Penghasilan (<i>double</i>)</li> <li>- Premi (<i>double</i>)</li> <li>- Cara Bayar (<i>enum</i>: 'Bulanan', '3 Bulanan', 'Tahunan')</li> <li>- Masa Pembebanan Biaya (<i>int</i>: 11)</li> <li>- Rencana pembayaran (<i>int</i>: 11)</li> </ul> Hasil Klasifikasi <ul style="list-style-type: none"> <li>- Jenis Kelamin</li> <li>- Status Marital</li> <li>- Usia</li> <li>- Penghasilan</li> <li>- Premi</li> <li>- Cara Bayar</li> <li>- Masa Pembebanan Biaya</li> </ul>

- Rencana pembayaran
- *Class*
- *Actions* yang memuat 2 tombol yaitu:
  - Terima: memvalidasi calon nasabah menjadi nasabah
  - Saran: memvalidasi calon nasabah dengan memberikan saran

#### SKENARIO UTAMA VALIDASI TERIMA CALON NASABAH ASURANSI

Aktor	Sistem
13. Menekan tombol “ <i>Terima</i> ”	14. Menampilkan <i>pop up</i> “Apakah calon nasabah akan benar - benar anda terima?”
15. Menekan tombol “Oke”	16. Mengubah status 1 ke 3 di <i>database</i> dan menyimpan data aktivitas penerimaan pada tabel aktivitas penerimaan ke <i>database</i> : <ul style="list-style-type: none"> <li>- <i>Idnasabah (int: 11)</i></li> <li>- <i>aktivitas (varchar: 255)</i> = "Lolos tahap <i>leader</i>"</li> <li>- <i>statusktivitas (int: 11)</i> = "3"</li> <li>- <i>tglaktivitas (timestamp)</i></li> <li>- <i>ket (text)</i> = "Anda sudah lolos menjadi Nasabah di Asuransi Perusahaan Asuransi XYZ"</li> </ul>
	17. Mengambil data calon nasabah dengan dari <i>database</i>
	18. Menampilkan data calon nasabah pada tabel di halaman data calon nasabah
	19. Menampilkan alert “Berhasil menerima nasabah!”

#### SKENARIO UTAMA BATAL VALIDASI TERIMA KEMBALI KE HALAMAN FORM HASIL KLASIFIKASI

Aktor	Sistem
15. Menekan tombol “Batal”	16. Menampilkan <i>form</i> hasil klasifikasi

#### SKENARIO UTAMA VALIDASI SARAN CALON NASABAH ASURANSI

Aktor	Sistem
13. Menekan tombol “ <i>Saran</i> ”	14. Menampilkan <i>form</i> modal saran: <ul style="list-style-type: none"> <li>- <i>Saran (text)</i></li> <li>- Tombol Kirim</li> </ul>

15. Mengisi form isian <i>form</i> modal saran: - Saran ( <i>text</i> )	
16. Menekan tombol “ <i>Kirim</i> ”	17. Menampilkan <i>pop up</i> “Apakah saran akan benar - benar anda kirim?”
18. Menekan tombol “Oke”	19. Mengubah status 1 ke 2 di <i>database</i> dan menyimpan data aktivitas penerimaan pada tabel aktivitas penerimaan: - Idnasabah ( <i>int: 11</i> ) - aktivitas ( <i>varchar: 255</i> ) = "Saran" - statusktivitas ( <i>int: 11</i> ) = "2" - tglaktivitas ( <i>timestamp</i> ) - ket ( <i>text</i> ) = saran
	20. Mengambil data calon nasabah dengan dari <i>database</i>
	21. Menampilkan data calon nasabah pada tabel di halaman data calon nasabah
	22. Menampilkan alert “Berhasil memberikan saran!”

#### SKENARIO UTAMA BATAL VALIDASI SARAN KEMBALI KE HALAMAN FORM HASIL KLASIFIKASI

Aktor	Sistem
18. Menekan tombol “Batal”	19. Menampilkan <i>form</i> hasil klasifikasi

#### SKENARIO ALTERNATIF VALIDASI SARAN CALON NASABAH ASURANSI

Jika aktor tidak mengisi *form* modal tambah data *saran* secara lengkap

Aktor	Sistem
15a. Mengisi isian <i>form</i> modal saran tidak lengkap	
16a. Menekan tombol “ <i>Kirim</i> ”	17a. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
18a. Menekan tombol “Oke”	19a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> tambah saran. Pesan akan tampil disetiap <i>field</i> yang kosong



## 16. Skenario Melihat Riwayat Klasifikasi

Penjelasan urutan aksi aktor untuk melihat riwayat klasifikasi calon nasabah asuransi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (Usecase Skenario).

## 17. Skenario Mengelola Data Set

Penjelasan urutan aksi aktor untuk mengelola data *set* meliputi (lihat, tambah, dan hapus) dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.10.

Tabel 4.10. Skenario Mengelola Data Set

Nomor <i>Usecase</i>	USC 17
Nama	Mengelola Data Set
Aktor	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> memilih menu Data Set
<i>Post Condition</i>	<i>Admin</i> berhasil melihat, menambah, dan menghapus data <i>set</i>
<b>SKENARIO UTAMA MELIHAT DATA SET</b>	
<b>Aktor</b>	<b>Sistem</b>
1. Memilih menu Data Set	2. Mengambil data <i>set</i> dari <i>database</i>
	3. Menampilkan data <i>set</i> pada tabel di halaman data <i>set</i> : <ul style="list-style-type: none"> <li>- No (<i>int: 11</i>)</li> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> <li>- <i>Actions</i> yang memuat 1 tombol yaitu: <ul style="list-style-type: none"> <li>• hapus: untuk menghapus data <i>training</i></li> </ul> </li> </ul> <p>Halaman ini juga menampilkan tombol tambah data yang digunakan apabila</p>

aktor akan menambahkan data *set* calon nasabah dan kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.

**SKENARIO UTAMA MENAMBAH DATA SET**

Aktor	Sistem
4. Menekan tombol “Tambahkan”	5. Menampilkan <i>form</i> modal tambah data <i>set</i> : <ul style="list-style-type: none"> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> <li>- Tombol Kirim</li> </ul>
6. Mengisi seluruh isian <i>form</i> modal tambah data <i>set</i> : <ul style="list-style-type: none"> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> </ul>	
7. Menekan tombol “Kirim”	8. Menampilkan <i>pop up</i> “Apakah data akan benar - benar disimpan?”
9. Menekan tombol “Oke”	10. Menyimpan masukan data ke <i>database</i>
	11. Mengambil data <i>set</i> dari <i>database</i>
	12. Menampilkan data <i>set</i> pada tabel di halaman data <i>set</i>
	13. Menampilkan alert “Data Berhasil di Simpan”

**SKENARIO UTAMA BATAL TAMBAH DATA SET KEMBALI KE FORM MODAL TAMBAH DATA SET**

Aktor	Sistem
9. Menekan tombol “Batal”	10. Menampilkan <i>form</i> modal tambah data <i>set</i>

**SKENARIO UTAMA MENGHAPUS DATA SET**

4. Menekan tombol ikon “Hapus” pada baris data yang akan dihapus	5. Menampilkan <i>pop up</i> “Apakah data akan benar - benar dihapus?”
6. Menekan tombol “Oke”	7. Menghapus data <i>set</i> yang dipilih di <i>database</i>
	8. Mengambil data <i>set</i> yang dari <i>database</i>
	9. Menampilkan data <i>set</i> pada tabel di halaman data <i>set</i>
	10. Menampilkan alert “Data Berhasil di Hapus”

**SKENARIO UTAMA BATAL HAPUS DATA SET KEMBALI KE HALAMAN DATA SET**

6. Klik “Batal”	7. Menampilkan data <i>set</i> pada tabel di halaman data <i>set</i>
-----------------	--

18. *Skenario Melihat Data Training*

Penjelasan urutan aksi aktor untuk melihat data *training* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.11.

Tabel 4.11. Skenario Melihat Data *Training*

Nomor <i>Usecase</i>	USC 18
Nama	Melihat Data <i>Training</i>
Aktor	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> memilih menu Data <i>Training</i>
<i>Post Condition</i>	<i>Admin</i> berhasil melihat data <i>training</i>

**SKENARIO UTAMA MELIHAT DATA TRAINING**

Aktor	Sistem
1. Memilih menu Data <i>Training</i>	2. Mengambil data <i>training</i> dari <i>database</i>

- 
3. Menampilkan data *training* pada tabel di halaman data *training*
    - No (*int: 11*)
    - Nama (*varchar: 200*)
    - Usia (*varchar: 2*)
    - Jenis Kelamin (*varchar: 2*)
    - Status Marital (*varchar: 2*)
    - Penghasilan (*varchar: 2*)
    - Premi (*varchar: 2*)
    - Cara Bayar (*varchar: 2*)
    - Masa Pembebanan Biaya (*varchar: 2*)
    - Rencana pembayaran (*varchar: 2*)
    - *Class* (*varchar: 2*)

Halaman ini juga menampilkan tombol *classification tree* apabila admin melakukan proses melihat hasil *classification tree*, tombol *rule* apabila admin melakukan proses melihat hasil *rule*, dan kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.

---

#### 19. Skenario Melihat Hasil *Classification Tree* dan *Rule*

Penjelasan urutan aksi aktor untuk melihat hasil *classification tree* dan *rule* serta reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.12.

Tabel 4.12. Skenario Melihat Hasil *Classification Tree* dan *Rule*

Nomor <i>Usecase</i>	USC 19
Nama	Melihat Hasil <i>Classification tree</i> dan <i>Rule</i>
Aktor	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> memilih menu <i>Data Training</i>
<i>Post Condition</i>	<i>Admin</i> berhasil melihat hasil <i>classification tree</i> dan <i>rule</i>

#### **SKENARIO UTAMA MELIHAT HASIL *CLASSIFICATION TREE***

Aktor	Sistem
1. Memilih menu Data <i>Training</i>	2. Mengambil data <i>training</i> dari <i>database</i>
	3. Menampilkan data <i>training</i> pada tabel di halaman data <i>training</i> <ul style="list-style-type: none"> <li>- No (<i>int: 11</i>)</li> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> </ul> Halaman ini juga menampilkan tombol <i>classification tree</i> apabila admin melakukan proses melihat hasil <i>classification tree</i> , tombol <i>rule</i> apabila admin melakukan proses melihat hasil <i>rule</i> , dan kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.
4. Menekan tombol “ <i>Decision Tree</i> ”	5. Menampilkan pohon klasifikasi dan tombol <i>update Decision Tree</i>
6. Menekan tombol “ <i>Update Decision Tree</i> ”	7. Menampilkan <i>form</i> modal bobot <i>cost function</i> : <ul style="list-style-type: none"> <li>a. Lancar ke Tidak Lancar</li> <li>b. Tidak Lancar ke Lancar</li> <li>c. Tombol Mulai</li> </ul>
8. Mengisi <i>form</i> modal bobot <i>cost function</i> : <ul style="list-style-type: none"> <li>- Lancar ke Tidak Lancar</li> <li>- Tidak Lancar ke Lancar</li> </ul>	
9. Menekan tombol “Mulai”	10. Melakukan pembagian data <i>set</i>
	11. Mereset data tabel perhitungan dan pohon klasifikasi pada <i>database</i>
	12. Menghitung proses pembuatan pohon klasifikasi menggunakan <i>classification</i>

	<p><i>tree C4.5 algorithm</i> dan <i>cost sensitive learning</i>:</p> <ol style="list-style-type: none"> <li>Menghitung jumlah nilai atribut tiap <i>class</i> dan tiap nilai atribut berkategori lancar dan tidak lancar</li> <li>Menghitung nilai <i>entropy</i> tiap <i>class</i> dan tiap nilai atribut</li> <li>Menghitung nilai <i>gain</i> tiap atribut</li> <li>Memilih atribut dengan nilai <i>gain</i> tertinggi dan menjadikannya akar</li> <li>Mengulang kembali proses a. – d. hingga semua atribut telah menjadi akar.</li> <li>Penentuan <i>leaf node</i> pada atribut terakhir menggunakan metode <i>cost sensitive learning</i> yaitu mengalikan nilai atribut tiap <i>class</i> dengan bobot yang telah ditentukan.</li> <li>Memilih <i>cost function</i> terkecil sebagai <i>class</i>.</li> </ol>
	13. Sistem melakukan proses pembentukan <i>classification tree</i> dan <i>rule</i>
	14. Menampilkan hasil <i>update classification tree</i>
<b>SKENARIO UTAMA MELIHAT CLASSIFICATION TREE</b>	
<b>Aktor</b>	<b>Sistem</b>
4. Menekan tombol “ <i>Decision Tree</i> ”	5. Sistem melakukan proses pembentukan <i>classification tree</i>
	6. Menampilkan <i>classification tree</i>
<b>SKENARIO UTAMA MELIHAT RULE</b>	
<b>Aktor</b>	<b>Sistem</b>
4. Menekan tombol “ <i>Rule</i> ”	5. Sistem melakukan proses konversi <i>classification tree</i> ke dalam bentuk <i>rule</i>
	6. Menampilkan <i>rule</i>
<b>SKENARIO ALTERNATIF MENAMBAH BOBOT COST FUNCTION</b>	
Jika aktor tidak mengisi <i>form</i> modal bobot <i>cost function</i> secara lengkap	
<b>Aktor</b>	<b>Sistem</b>
8a. Mengisi isian <i>form</i> modal bobot <i>cost function</i> tidak lengkap	
9a. Menekan tombol “ <i>Kirim</i> ”	10a. Menampilkan pesan dengan menunjuk lokasi isian yang belum diisi pada <i>form</i> modal bobot <i>cost function</i> . Pesan akan tampil di setiap <i>field</i> yang kosong

20. Skenario Melihat Data *Testing*

Penjelasan urutan aksi aktor untuk melihat data *testing* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.13.

Tabel 4.13. Skenario Melihat Data *Testing*

Nomor <i>Usecase</i>	USC 20
Nama	Melihat Data <i>Testing</i>
Aktor	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> memilih menu Data <i>Testing</i>
<i>Post Condition</i>	<i>Admin</i> berhasil melihat data <i>testing</i>
<b>SKENARIO UTAMA MELIHAT DATA TESTING</b>	
Aktor	Sistem
4. Memilih menu Data <i>Testing</i>	5. Mengambil data <i>testing</i> dari <i>database</i>
	6. Menampilkan data <i>testing</i> pada tabel di halaman data <i>testing</i>
	<ul style="list-style-type: none"> <li>- No (<i>int: 11</i>)</li> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> <li>- <i>Actions</i> yang memuat 2 tombol yaitu: <ul style="list-style-type: none"> <li>• hapus: untuk menghapus data <i>test</i></li> </ul> </li> </ul>
	Halaman ini juga menampilkan tombol proses <i>testing</i> yang digunakan apabila aktor akan menguji hasil pohon klasifikasi, serta kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.

## 21. Skenario Melihat Hasil Uji Klasifikasi

Penjelasan urutan aksi aktor untuk melihat hasil uji klasifikasi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Tabel 4.14.

Tabel 4.14. Skenario Melihat Hasil Uji Klasifikasi

Nomor <i>Usecase</i>	USC 21
Nama	Melihat Hasil Uji Klasifikasi
Aktor	<i>Admin</i>
<i>Pre Condition</i>	<i>Admin</i> memilih menu <i>Data Testing</i>
<i>Post Condition</i>	<i>Admin</i> berhasil melihat hasil uji klasifikasi (proses <i>testing</i> )
<b>SKENARIO UTAMA MELIHAT HASIL UJI KLASIFIKASI</b>	
Aktor	Sistem
7. Memilih menu <i>Data Testing</i>	8. Mengambil data <i>testing</i> dari <i>database</i>
	9. Menampilkan data <i>testing</i> pada tabel di halaman data <i>testing</i>
	- No ( <i>int: 11</i> )
	- Nama ( <i>varchar: 200</i> )
	- Usia ( <i>varchar: 2</i> )
	- Jenis Kelamin ( <i>varchar: 2</i> )
	- Status Marital ( <i>varchar: 2</i> )
	- Penghasilan ( <i>varchar: 2</i> )
	- Premi ( <i>varchar: 2</i> )
	- Cara Bayar ( <i>varchar: 2</i> )
	- Masa Pembebanan Biaya ( <i>varchar: 2</i> )
	- Rencana pembayaran ( <i>varchar: 2</i> )
	- <i>Class</i> ( <i>varchar: 2</i> )
	- <i>Actions</i> yang memuat 2 tombol yaitu:
	• hapus: untuk menghapus data <i>test</i>
	Halaman ini juga menampilkan tombol proses <i>testing</i> yang digunakan apabila aktor akan menguji hasil pohon klasifikasi, serta kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.
4. Menekan tombol “Proses <i>Testing</i> ”	5. Menghitung jumlah <i>record</i> data <i>testing</i>
	6. Melakukan klasifikasi menggunakan <i>rule</i> yang telah dikonversi dari



	<i>classification tree</i> sebanyak jumlah <i>record data testing</i>
7.	Menghitung frekuensi data <i>testing</i> : <ul style="list-style-type: none"> <li>- Prediksi “Lancar”, Aktual “Lancar”</li> <li>- Prediksi “Lancar”, Aktual “Tidak lancar”</li> <li>- Prediksi “Tidak lancar”, Aktual “Lancar”</li> <li>- Prediksi “Tidak lancar”, Aktual “Tidak lancar”</li> </ul>
8.	Menghitung <i>accuracy</i> , <i>precision</i> , dan <i>recall</i>
9.	Menyimpan hasil perhitungan
10.	Menampilkan hasil perhitungan

#### 22. Skenario Mencetak Data Nasabah

Penjelasan urutan aksi aktor untuk mencetak data nasabah asuransi dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase Skenario*).

#### 23. Skenario Melihat Aktivitas Saran

Penjelasan urutan aksi aktor untuk melihat aktivitas saran dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase Skenario*).

#### 24. Skenario *Logout*

Penjelasan urutan aksi aktor untuk melakukan *logout* dan reaksi sistem pada skenario normal dan skenario alternatif terdapat pada Lampiran B (*Usecase Skenario*).

#### 4.3.4 *Activity Diagram*

*Activity diagram* dokumentasi desain yang menggambarkan aliran aktivitas dalam sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* yang akan dibangun. Sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* memiliki empat puluh *activity diagram* yaitu sebagai berikut:

1. *Activity Diagram Login*

*Activity diagram login* menggambarkan alur aktivitas dari proses *login* agar *user* dapat mengakses fitur sistem. Aktor yang melakukan *login* adalah admin, calon nasabah, *agent*, dan *leader*. Setiap *user* harus memiliki *username* dan *password* yang terdaftar pada sistem. Hanya *user* berstatus aktif yang memiliki hak akses terhadap sistem. *Activity diagram login* terdapat pada Lampiran C (*Activity Diagram*).

2. *Activity Diagram Mengelola Data Agent (lihat)*

*Activity diagram* mengelola data *agent* (*lihat*) menggambarkan alur aktivitas proses melihat data *agent* yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk melihat data *agent* dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* mengelola data *agent* (*lihat*) terdapat pada Lampiran C (*Activity Diagram*).

3. *Activity Diagram Mengelola Data Agent (tambah)*

*Activity diagram* mengelola data *agent* (*tambah*) menggambarkan alur aktivitas proses memasukkan data *agent* yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk menambah data *agent* dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity diagram* mengelola data *agent* (*tambah*) terdapat pada Lampiran C (*Activity Diagram*).

4. *Activity Diagram Mengelola Data Agent (ubah)*

*Activity diagram* mengelola data *agent* (*ubah*) menggambarkan alur aktivitas proses mengubah data *agent* yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk mengubah data *agent* dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* mengelola data *agent* (*ubah*) terdapat pada Lampiran C (*Activity Diagram*).

5. *Activity Diagram* Mengelola Data Agent (hapus)

*Activity diagram* mengelola data *agent* (hapus) menggambarkan alur aktivitas proses menghapus data *agent* yang dilakukan oleh *leader*. Menghapus dalam hal ini tidak menghilangkan data dari database melainkan mengubah status *use r* menjadi “Tidak Aktif” sehingga tidak terdapat di *interface* sistem. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk menghapus data *agent* dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity diagram* mengelola data *agent* (hapus) terdapat pada Lampiran C (*Activity Diagram*).

6. *Activity Diagram* Mengubah Data Diri

*Activity diagram* mengubah data diri menggambarkan alur aktivitas proses mengubah data diri yang dilakukan oleh calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan calon nasabah untuk mengubah data diri dan reaksi sistem ketika calon nasabah memberikan aksi terhadap sistem. *Activity* mengubah data diri terdapat pada Lampiran C (*Activity Diagram*).

7. *Activity Diagram* Menambah Data Register

*Activity diagram* menambah data register menggambarkan alur aktivitas proses menambah data register yang dilakukan oleh calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan calon nasabah untuk menambah data register dan reaksi sistem ketika calon nasabah memberikan aksi terhadap sistem. *Activity diagram* menambah data register terdapat pada Lampiran C (*Activity Diagram*).

8. *Activity Diagram* Melihat Data Agent

*Activity diagram* melihat data *agent* menggambarkan alur aktivitas proses melihat data *agent* yang dilakukan oleh calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan calon nasabah untuk melihat data calon

nasabah dan reaksi sistem ketika calon nasabah memberikan aksi terhadap sistem. *Activity* melihat data *agent* terdapat pada Lampiran C (*Activity Diagram*).

9. *Activity Diagram* Menambah Mendaftar *Agent*

*Activity diagram* menambah mendaftar *agent* menggambarkan alur aktivitas proses mendaftar *agent* yang dilakukan oleh calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan calon nasabah untuk mendaftar *agent* dan reaksi sistem ketika calon nasabah memberikan aksi terhadap sistem. *Activity diagram* menambah mendaftar *agent* terdapat pada Lampiran C (*Activity Diagram*).

10. *Activity Diagram* Melihat Aktivitas Penerimaan

*Activity diagram* melihat aktivitas penerimaan menggambarkan alur aktivitas proses melihat aktivitas penerimaan yang dilakukan oleh calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan calon nasabah untuk melihat aktivitas penerimaan dan reaksi sistem ketika calon nasabah memberikan aksi terhadap sistem. *Activity diagram* melihat aktivitas penerimaan terdapat pada Lampiran C (*Activity Diagram*).

11. *Activity Diagram* Mengelola Data Persyaratan Calon Nasabah Asuransi (lihat)

*Activity diagram* mengelola data persyaratan calon nasabah asuransi (lihat) menggambarkan alur aktivitas proses melihat data persyaratan calon nasabah asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk melihat data persyaratan calon nasabah asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* mengelola data persyaratan calon nasabah asuransi (lihat) terdapat pada Lampiran C (*Activity Diagram*).

12. *Activity Diagram* Mengelola Data Persyaratan Calon Nasabah Asuransi (tambah)

*Activity diagram* mengelola data persyaratan calon nasabah asuransi (tambah) menggambarkan alur aktivitas proses memasukkan data persyaratan calon nasabah asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk menambah data persyaratan calon nasabah asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity diagram* mengelola data persyaratan calon nasabah asuransi (tambah) terdapat pada Lampiran C (*Activity Diagram*).

13. *Activity Diagram* Mengelola Data Persyaratan Calon Nasabah Asuransi (ubah)

*Activity diagram* mengelola data persyaratan calon nasabah asuransi (ubah) menggambarkan alur aktivitas proses mengubah data persyaratan calon nasabah asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk mengubah data persyaratan calon nasabah asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* mengelola data persyaratan calon nasabah asuransi (ubah) terdapat pada Lampiran C (*Activity Diagram*).

14. *Activity Diagram* Mengelola Data Persyaratan Calon Nasabah Asuransi (hapus)

*Activity diagram* mengelola data persyaratan calon nasabah asuransi (hapus) menggambarkan alur aktivitas proses menghapus data persyaratan calon nasabah asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk menghapus data persyaratan calon nasabah asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity diagram* mengelola data persyaratan calon nasabah asuransi (hapus) terdapat pada Lampiran C (*Activity Diagram*).

15. *Activity Diagram* Mengelola Data Manfaat Asuransi (lihat)

*Activity diagram* mengelola data manfaat asuransi (lihat) menggambarkan alur aktivitas proses melihat data manfaat asuransi yang dilakukan oleh *leader*. *Activity*

*diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk melihat data manfaat asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* mengelola data manfaat asuransi (lihat) terdapat pada Lampiran C (*Activity Diagram*).

16. *Activity Diagram* Mengelola Data Manfaat Asuransi (tambah)

*Activity diagram* mengelola data manfaat asuransi (tambah) menggambarkan alur aktivitas proses memasukkan data manfaat asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk menambah data manfaat asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity diagram* mengelola data manfaat asuransi (tambah) terdapat pada Lampiran C (*Activity Diagram*).

17. *Activity Diagram* Mengelola Data Manfaat Asuransi (ubah)

*Activity diagram* mengelola data manfaat asuransi (ubah) menggambarkan alur aktivitas proses mengubah data manfaat asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk mengubah data manfaat asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* mengelola data manfaat asuransi (ubah) terdapat pada Lampiran C (*Activity Diagram*).

18. *Activity Diagram* Mengelola Data Manfaat Asuransi (hapus)

*Activity diagram* mengelola data manfaat asuransi (hapus) menggambarkan alur aktivitas proses menghapus data manfaat asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk menghapus data manfaat asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity diagram* mengelola data manfaat asuransi (hapus) terdapat pada Lampiran C (*Activity Diagram*).

19. *Activity Diagram* Melihat Data Persyaratan Calon Nasabah Asuransi

*Activity diagram* melihat data persyaratan calon nasabah asuransi menggambarkan alur aktivitas proses melihat data persyaratan calon nasabah asuransi yang dilakukan oleh *agent* dan calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* dan calon nasabah untuk melihat data persyaratan calon nasabah asuransi dan reaksi sistem ketika *agent* dan calon nasabah memberikan aksi terhadap sistem. *Activity* melihat data persyaratan calon nasabah asuransi terdapat pada Lampiran C (*Activity Diagram*).



20. *Activity Diagram* Melihat Data Manfaat Asuransi

*Activity diagram* melihat data manfaat asuransi menggambarkan alur aktivitas proses melihat data manfaat asuransi yang dilakukan oleh *agent* dan calon nasabah. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* dan calon nasabah untuk melihat data manfaat asuransi dan reaksi sistem ketika *agent* dan calon nasabah memberikan aksi terhadap sistem. *Activity* melihat data manfaat asuransi terdapat pada Lampiran C (*Activity Diagram*).

21. *Activity Diagram* Validasi Calon Nasabah Asuransi

*Activity diagram* validasi calon nasabah asuransi menggambarkan alur aktivitas proses validasi calon nasabah asuransi yang dilakukan oleh *agent*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* untuk memvalidasi calon nasabah asuransi dan reaksi sistem ketika *agent* memberikan aksi terhadap sistem. *Activity diagram* validasi calon nasabah asuransi terdapat pada Lampiran C (*Activity Diagram*).

22. *Activity Diagram* Mengelola Data Calon Nasabah Asuransi (lihat)

*Activity diagram* mengelola data calon nasabah asuransi (lihat) menggambarkan alur aktivitas proses melihat data calon nasabah asuransi yang dilakukan oleh *agent*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* untuk melihat data calon nasabah asuransi dan reaksi sistem ketika *agent* memberikan aksi terhadap sistem. *Activity* mengelola data calon nasabah asuransi (lihat) terdapat pada Lampiran C (*Activity Diagram*).

23. *Activity Diagram* Mengelola Data Calon Nasabah Asuransi (tambah)

*Activity diagram* mengelola data calon nasabah asuransi (tambah) menggambarkan alur aktivitas proses memasukkan data calon nasabah asuransi yang dilakukan oleh *agent*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* untuk menambah data calon nasabah asuransi dan reaksi sistem ketika



*agent* memberikan aksi terhadap sistem. *Activity diagram* mengelola data calon nasabah asuransi (tambah) terdapat pada Lampiran C (*Activity Diagram*).

24. *Activity Diagram* Mengelola Data Calon Nasabah Asuransi (ubah)

*Activity diagram* mengelola data calon nasabah asuransi (ubah) menggambarkan alur aktivitas proses mengubah data calon nasabah asuransi yang dilakukan oleh *agent*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* untuk mengubah data calon nasabah asuransi dan reaksi sistem ketika *agent* memberikan aksi terhadap sistem. *Activity* mengelola data calon nasabah asuransi (ubah) terdapat pada Lampiran C (*Activity Diagram*).

25. *Activity Diagram* Melihat Data Nasabah Asuransi

*Activity diagram* melihat data nasabah asuransi menggambarkan alur aktivitas proses melihat data nasabah asuransi yang dilakukan oleh *agent* dan *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* dan *leader* untuk melihat data nasabah asuransi dan reaksi sistem ketika *agent* dan *leader* memberikan aksi terhadap sistem. *Activity* melihat data nasabah asuransi terdapat pada Lampiran C (*Activity Diagram*).

26. *Activity Diagram* Melihat Hasil Klasifikasi (lihat)

*Activity diagram* melihat hasil klasifikasi (lihat) menggambarkan alur aktivitas proses melihat hasil klasifikasi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk melihat hasil klasifikasi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* melihat hasil klasifikasi (lihat) terdapat pada Lampiran C (*Activity Diagram*).

27. *Activity Diagram* Melihat Hasil Klasifikasi (validasi terima)

*Activity diagram* melihat hasil klasifikasi (validasi terima) menggambarkan alur aktivitas proses memvalidasi terima calon nasabah asuransi yang dilakukan oleh

*leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk memvalidasi terima calon nasabah asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* melihat hasil klasifikasi (validasi terima) terdapat pada Lampiran C (*Activity Diagram*).

28. *Activity Diagram* Melihat Hasil Klasifikasi (validasi saran)

*Activity diagram* melihat hasil klasifikasi (validasi saran) menggambarkan alur aktivitas proses memberi saran pada calon nasabah asuransi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk memberi saran pada calon nasabah asuransi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* melihat hasil klasifikasi (validasi saran) terdapat pada Lampiran C (*Activity Diagram*).

29. *Activity Diagram* Melihat Riwayat Klasifikasi

*Activity diagram* melihat riwayat klasifikasi menggambarkan alur aktivitas proses melihat riwayat klasifikasi yang dilakukan oleh *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *leader* untuk melihat riwayat klasifikasi dan reaksi sistem ketika *leader* memberikan aksi terhadap sistem. *Activity* melihat riwayat klasifikasi terdapat pada Lampiran C (*Activity Diagram*).

30. *Activity Diagram* Mengelola Data Set (lihat)

*Activity diagram* mengelola data *set* (lihat) menggambarkan alur aktivitas proses melihat data *set* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data *set* dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* mengelola data *set* (lihat) terdapat pada Lampiran C (*Activity Diagram*).

31. *Activity Diagram* Mengelola Data Set (tambah)

*Activity diagram* mengelola data *set* (tambah) menggambarkan alur aktivitas proses memasukkan data *set* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menambah data *set* dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram* mengelola data *set* (tambah) terdapat pada **Error! Reference source not found.**

32. *Activity Diagram* Mengelola Data *Set* (hapus)

*Activity diagram* mengelola data *set* (hapus) menggambarkan alur aktivitas proses menghapus data *set* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk menghapus data *set* dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram* mengelola data *set* (hapus) terdapat pada Lampiran C (*Activity Diagram*).

33. *Activity Diagram* Melihat Data *Training*

*Activity diagram* melihat data *training* menggambarkan alur aktivitas proses melihat data *training* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data *training* dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* melihat data *training* terdapat pada Lampiran C (*Activity Diagram*).

34. *Activity Diagram* Melihat *Classification Tree*

*Activity diagram* melihat *classification tree* menggambarkan alur aktivitas proses melihat *classification tree* menggunakan algoritma C4.5 dan *cost sensitive learning* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk proses melihat pohon klasifikasi (*classification tree*) dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* melihat *classification tree* terdapat pada Lampiran C (*Activity Diagram*).

35. *Activity Diagram* Melihat Hasil *Classification Tree*

*Activity diagram* melihat hasil *classification tree* (*update classification tree*) menggambarkan alur aktivitas proses mengupdate pohon klasifikasi (*classification tree*) menggunakan algoritma C4.5 dan *cost sensitive learning* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk proses *update* pohon klasifikasi (*classification tree*) dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram* melihat hasil *classification tree* (*update classification tree*) terdapat pada Lampiran C (*Activity Diagram*).

36. *Activity Diagram* Melihat *Rule*

*Activity diagram* melihat *rule* menggambarkan alur aktivitas proses melihat *rule* dari pohon klasifikasi (*classification tree*) yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk proses melihat *rule* dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity diagram* melihat *rule* terdapat pada Lampiran C (*Activity Diagram*).

37. *Activity Diagram* Melihat Data *Testing*

*Activity diagram* melihat data *testing* menggambarkan alur aktivitas proses melihat data *testing* yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk melihat data *testing* dan reaksi sistem ketika admin memberikan aksi terhadap sistem. *Activity* melihat data *testing* terdapat pada Lampiran C (*Activity Diagram*).

38. *Activity Diagram* Uji Klasifikasi (proses *testing*)

*Activity diagram* uji klasifikasi (proses *testing*) menggambarkan alur aktivitas proses mengevaluasi pohon klasifikasi (*classification tree*) yang telah dibuat saat proses *data mining* pada *activity diagram* Gambar 4.16 yang dilakukan oleh admin. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan admin untuk proses mengevaluasi pohon klasifikasi (*classification tree*) dan reaksi sistem ketika admin

memberikan aksi terhadap sistem. *Activity diagram* uji klasifikasi (proses *testing*) terdapat pada Lampiran C (*Activity Diagram*).

#### 39. *Activity Diagram* Mencetak Data Nasabah Asuransi

*Activity diagram* mencetak data nasabah asuransi menggambarkan alur aktivitas proses mencetak data nasabah asuransi yang dilakukan oleh *agent* dan *leader*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* dan *leader* untuk mencetak data nasabah asuransi dan reaksi sistem ketika *agent* dan *leader* memberikan aksi terhadap sistem. *Activity* mencetak data nasabah asuransi terdapat pada Lampiran C (*Activity Diagram*).

#### 40. *Activity Diagram* Melihat Aktivitas Saran

*Activity diagram* melihat aktivitas saran menggambarkan alur aktivitas proses melihat aktivitas saran yang dilakukan oleh *agent*. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *agent* untuk melihat aktivitas saran untuk calon nasabah *agent* dan reaksi sistem ketika *agent* memberikan aksi terhadap sistem. *Activity diagram* melihat aktivitas saran terdapat pada Lampiran C (*Activity Diagram*).

#### 41. *Activity Diagram* Logout

*Activity diagram user* menggambarkan alur aktivitas dari proses *logout* agar *user* dapat keluar dari sistem. *Activity diagram* ini menjelaskan tentang alur yang harus dilakukan *user* untuk keluar dari sistem dan reaksi sistem ketika *user* memberikan aksi *user* terhadap sistem. *Activity diagram logout* terdapat pada Lampiran C (*Activity Diagram*).

#### 4.3.5 *Sequence Diagram*

*Sequence diagram* merupakan dokumentasi desain berbentuk *diagram* terurut yang menampilkan interaksi-interaksi antar objek dalam sistem. *Sequence diagram* digunakan untuk menggambarkan skenario dan memodelkan aliran logika dalam

sistem secara *visual*. *Sequence diagram* dari sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* adalah sebagai berikut:

1. *Sequence Diagram Login*

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses *login*. Semua aktor yang berstatus aktif dapat *login* dalam *sequence* ini. Pada *sequence* ini terdapat *class view* *index*, *v\_homeLeader*, *v\_homeAdmin*, *v\_homeAgent*, dan *v\_homeNasabah*, *class controller* *c\_login* dan *c\_home*, *class model* *m\_login* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram login* dapat dilihat pada Lampiran D (*Sequence Diagram*).

2. *Sequence Diagram Mengelola Data Agent* (lihat)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data *agent*. Aktor yang dapat melihat data *agent* dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view* *dashboard* dan *v\_agent*, *class controller* *c\_agent*, *class model* *m\_agent* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data *agent* (lihat) dapat dilihat pada Lampiran D (*Sequence Diagram*).

3. *Sequence Diagram Mengelola Data Agent* (tambah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menambah data *agent*. Aktor yang dapat menambah data *agent* dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view* *dashboard* dan *v\_agent*, *class controller* *c\_agent*, *class model* *m\_agent* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data *agent* (tambah) dapat dilihat pada Lampiran D (*Sequence Diagram*).



4. *Sequence Diagram* Mengelola Data *Agent* (ubah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengubah data *agent*. Aktor yang dapat mengubah data *agent* dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_agent*, *class controller c\_agent*, *class model m\_agent* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data *agent* (ubah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

5. *Sequence Diagram* Mengelola Data *Agent* (hapus)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menghapus data *agent*. Aktor yang dapat menghapus data *agent* dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_agent*, *class controller c\_agent*, *class model m\_agent* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data *agent* (hapus) dapat dilihat pada Lampiran D (*Sequence Diagram*).

6. *Sequence Diagram* Mengubah Data Diri

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengubah data diri. Aktor yang dapat mengubah data diri dalam *sequence* ini adalah calon nasabah. Pada *sequence* ini terdapat *class view dashboard* dan *v\_dataDiri*, *class controller c\_nasabah*, *class model m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengubah data diri dapat dilihat pada Lampiran D (*Sequence Diagram*).



#### 7. *Sequence Diagram* Menambah Data Register

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menambah data register. Aktor yang dapat mengubah data diri dalam *sequence* ini adalah calon nasabah. Pada *sequence* ini terdapat *class view* index, *class controller* c\_register, *class model* m\_register serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* menambah data register dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 8. *Sequence Diagram* Melihat Data Agent

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data *agent*. Aktor yang dapat melihat data *agent* dalam *sequence* ini adalah calon nasabah. Pada *sequence* ini terdapat *class view* dashboard dan v\_agent, *class controller* c\_agent, *class model* m\_nasabah dan m\_agent serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat data *agent* dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 9. *Sequence Diagram* Menambah Mendaftar Agent

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mendaftar pada *agent*. Aktor yang dapat mendaftar pada *agent* dalam *sequence* ini adalah calon nasabah. Pada *sequence* ini terdapat *class view* dashboard dan v\_agent, *class controller* c\_agent, *class model* m\_nasabah dan m\_agent serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* menambah mendaftar *agent* dapat dilihat pada Gambar 4.20.

#### 10. *Sequence Diagram* Melihat Aktivitas Penerimaan

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat aktivitas penerimaan. Aktor yang dapat melihat aktivitas penerimaan dalam *sequence* ini adalah calon nasabah. Pada *sequence* ini terdapat *class view dashboard* dan *v\_aktivitas penerimaan*, *class controller c\_nasabah*, *class model m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat aktivitas penerimaan dapat dilihat pada Lampiran D (*Sequence Diagram*).

11. *Sequence Diagram* Mengelola Data Persyaratan Nasabah Asuransi (lihat)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data persyaratan nasabah asuransi. Aktor yang dapat melihat data persyaratan nasabah asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_persyaratan*, *class controller c\_persyaratan*, *class model m\_persyaratan* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data persyaratan nasabah asuransi (lihat) dapat dilihat pada Lampiran D (*Sequence Diagram*).

12. *Sequence Diagram* Mengelola Data Persyaratan Nasabah Asuransi (tambah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menambah data persyaratan nasabah asuransi. Aktor yang dapat menambah data persyaratan nasabah asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_persyaratan*, *class controller c\_persyaratan*, *class model m\_persyaratan* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data persyaratan nasabah asuransi (tambah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

13. *Sequence Diagram* Mengelola Data Persyaratan Nasabah Asuransi (ubah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengubah data persyaratan nasabah asuransi. Aktor yang dapat mengubah data persyaratan nasabah asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_persyaratan*, *class controller c\_persyaratan*, *class model m\_persyaratan* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data persyaratan nasabah asuransi (ubah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

14. *Sequence Diagram* Mengelola Data Persyaratan Nasabah Asuransi (hapus)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menghapus data persyaratan nasabah asuransi. Aktor yang dapat menghapus data persyaratan nasabah asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_persyaratan*, *class controller c\_persyaratan*, *class model m\_persyaratan* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data persyaratan nasabah asuransi (hapus) dapat dilihat pada Lampiran D (*Sequence Diagram*).

15. *Sequence Diagram* Mengelola Data Manfaat Asuransi (lihat)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data manfaat asuransi. Aktor yang dapat melihat data manfaat asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_manfaat*, *class controller c\_manfaat*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data manfaat asuransi (lihat) dapat dilihat pada Lampiran D (*Sequence Diagram*).

16. *Sequence Diagram* Mengelola Data Manfaat Asuransi (tambah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menambah data manfaat asuransi. Aktor yang dapat menambah data manfaat asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_manfaat*, *class controller c\_manfaat*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini.

*Sequence diagram* mengelola data manfaat asuransi (tambah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

17. *Sequence Diagram* Mengelola Data Manfaat Asuransi (ubah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengubah data manfaat asuransi. Aktor yang dapat mengubah data manfaat asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_manfaat*, *class controller c\_manfaat*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data manfaat asuransi (ubah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

18. *Sequence Diagram* Mengelola Data Manfaat Asuransi (hapus)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menghapus data manfaat asuransi. Aktor yang dapat menghapus data manfaat asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_manfaat*, *class controller c\_manfaat*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data manfaat asuransi (hapus) dapat dilihat pada Lampiran D (*Sequence Diagram*).

19. *Sequence Diagram* Melihat Data Persyaratan Nasabah Asuransi

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data persyaratan nasabah asuransi. Aktor yang dapat melihat data persyaratan nasabah asuransi dalam *sequence* ini adalah *agent* dan calon nasabah. Pada *sequence* ini terdapat *class view dashboard* dan *v\_persyaratan*, *class controller*

*c\_persyaratan*, *class model m\_persyaratan* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat data persyaratan nasabah asuransi dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 20. *Sequence Diagram* Melihat Data Manfaat Asuransi

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data manfaat asuransi. Aktor yang dapat melihat data persyaratan nasabah asuransi dalam *sequence* ini adalah *agent* dan calon nasabah. Pada *sequence* ini terdapat *class view dashboard* dan *v\_manfaat*, *class controller c\_manfaat*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat data manfaat asuransi dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 21. *Sequence Diagram* Validasi Calon Nasabah Asuransi

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses validasi calon nasabah asuransi. Aktor yang dapat memvalidasi calon nasabah asuransi dalam *sequence* ini adalah *agent*. Pada *sequence* ini terdapat *class view dashboard*, *v\_antrianCalon* dan *v\_detailAntrianCalon*, *class controller c\_nasabah*, *class model m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* validasi calon nasabah asuransi dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 22. *Sequence Diagram* Mengelola Data Calon Nasabah Asuransi (lihat)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data calon nasabah asuransi. Aktor yang dapat melihat data calon nasabah asuransi dalam *sequence* ini adalah *agent*. Pada *sequence* ini terdapat *class*

*view dashboard* dan *v\_calonNasabah*, *class controller c\_nasabah*, *class model m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data calon nasabah asuransi (lihat) dapat dilihat pada Lampiran D (*Sequence Diagram*).

23. *Sequence Diagram* Mengelola Data Calon Nasabah Asuransi (tambah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menambah data calon nasabah asuransi. Aktor yang dapat menambah data calon nasabah asuransi dalam *sequence* ini adalah *agent*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_calonNasabah*, *class controller c\_nasabah*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data calon nasabah asuransi (tambah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

24. *Sequence Diagram* Mengelola Data Calon Nasabah Asuransi (ubah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengubah data calon nasabah asuransi. Aktor yang dapat mengubah data calon nasabah asuransi dalam *sequence* ini adalah *agent*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_calonNasabah*, *class controller c\_nasabah*, *class model m\_manfaat* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data calon nasabah asuransi (ubah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

25. *Sequence Diagram* Melihat Data Nasabah Asuransi

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data nasabah asuransi. Aktor yang dapat melihat data nasabah asuransi

dalam *sequence* ini adalah *agent* dan *leader*. Pada *sequence* ini terdapat *class view* *dashboard* dan *v\_dataNasabah*, *class controller* *c\_nasabah*, *class model* *m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat data nasabah asuransi dapat dilihat pada Lampiran D (*Sequence Diagram*).

26. *Sequence Diagram* Melihat Hasil Klasifikasi (lihat)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat hasil klasifikasi. Aktor yang dapat melihat hasil klasifikasi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view* *v\_dataCalonNasabah*, *v\_detailCalonNasabah*, dan *v\_klasifikasi*, *class controller* *c\_nasabah* dan *c\_klasifikasi*, *class model* *m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat hasil klasifikasi (lihat) dapat dilihat pada Lampiran D (*Sequence Diagram*).

27. *Sequence Diagram* Melihat Hasil Klasifikasi (validasi terima)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses memvalidasi terima calon nasabah asuransi. Aktor yang dapat melihat hasil klasifikasi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view* *v\_dataCalonNasabah*, *v\_detailCalonNasabah*, dan *v\_klasifikasi*, *class controller* *c\_nasabah* dan *c\_klasifikasi*, *class model* *m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat hasil klasifikasi (validasi terima) dapat dilihat pada Lampiran D (*Sequence Diagram*).

28. *Sequence Diagram* Melihat Hasil Klasifikasi (saran)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan



proses menambah data saran pada calon nasabah asuransi. Aktor yang dapat menambah data saran pada calon nasabah asuransi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view* v\_dataCalonNasabah, v\_detailCalonNasabah, dan v\_klasifikasi, *class controller* c\_nasabah dan c\_klasifikasi, *class model* m\_nasabah serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat hasil klasifikasi (saran) dapat dilihat pada Lampiran D (*Sequence Diagram*).

29. *Sequence Diagram* Melihat Riwayat Klasifikasi

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat riwayat klasifikasi. Aktor yang dapat melihat riwayat klasifikasi dalam *sequence* ini adalah *leader*. Pada *sequence* ini terdapat *class view* dashboard dan v\_riwayatKlasifikasi, *class controller* c\_klasifikasi, *class model* m\_klasifikasi serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat riwayat klasifikasi dapat dilihat pada Lampiran D (*Sequence Diagram*).

30. *Sequence Diagram* Mengelola Data Set (lihat)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data set. Aktor yang dapat melihat data set dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view* dashboard dan v\_datatset, *class controller* c\_klasifikasi, *class model* m\_klasifikasi serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data set (lihat) dapat dilihat pada Lampiran D (*Sequence Diagram*).

31. *Sequence Diagram* Mengelola Data Set (tambah)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menambah data set. Aktor yang dapat menambah data set dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard* dan *v\_dataset*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data set (tambah) dapat dilihat pada Lampiran D (*Sequence Diagram*).

32. *Sequence Diagram* Mengelola Data Set (hapus)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses menghapus data set. Aktor yang dapat menghapus data set dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard* dan *v\_dataset*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mengelola data set (hapus) dapat dilihat pada Lampiran D (*Sequence Diagram*).

33. *Sequence Diagram* Melihat Data Training

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data *training*. Aktor yang dapat melakukan proses melihat data *training* dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard* dan *v\_datatraining*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat data *training* dapat dilihat pada Lampiran D.

34. *Sequence Diagram* Melihat Classification tree

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan

proses melihat pohon klasifikasi (*classification tree*). Aktor yang dapat melakukan proses melihat pohon klasifikasi (*classification tree*) dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard*, *v\_datatraining*, dan *v\_classification tree*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat *classification tree* dapat dilihat pada Lampiran D (*Sequence Diagram*).

### 35. *Sequence Diagram* Melihat Hasil *Classification tree*

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat hasil pohon klasifikasi (*classification tree*). Aktor yang dapat melakukan melihat hasil pohon klasifikasi (*classification tree*) dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard*, *v\_datatraining*, dan *v\_classification tree*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat hasil *classification tree* dapat dilihat pada Lampiran D (*Sequence Diagram*).

### 36. *Sequence Diagram* Melihat *Rule*

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat *rule* dari pohon klasifikasi (*classification tree*). Aktor yang dapat melakukan proses melihat *rule* dari pohon klasifikasi (*classification tree*) dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard*, *v\_datatraining*, dan *v\_rule*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat *rule* dapat dilihat pada Lampiran D (*Sequence Diagram*).

### 37. *Sequence Diagram* Melihat Data *Testing*

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat data *testing*. Aktor yang dapat melihat data *testing* dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard* dan *v\_datatest*, *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat data *testing* dapat dilihat pada Lampiran D (*Sequence Diagram*).

38. *Sequence Diagram* Uji Klasifikasi (proses *testing*)

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mengevaluasi pohon klasifikasi (*classification tree*) yang telah dibuat saat proses *data mining* pada *sequence diagram* Gambar 4.28. Aktor yang dapat melakukan proses mengevaluasi pohon klasifikasi (*classification tree*) dalam *sequence* ini adalah admin. Pada *sequence* ini terdapat *class view dashboard*, *v\_datatest*, dan *v\_ujimetode*, dan *class controller c\_klasifikasi*, *class model m\_klasifikasi* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* uji klasifikasi (proses *testing*) dapat dilihat pada Lampiran D (*Sequence Diagram*).

39. *Sequence Diagram* Mencetak Data Nasabah Asuransi

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses mencetak data nasabah asuransi. Aktor yang dapat mencetak data nasabah asuransi dalam *sequence* ini adalah *agent* dan *leader*. Pada *sequence* ini terdapat *class view dashboard*, *v\_dataNasabah*, dan *pdfCetak*, *class controller c\_nasabah*, *class model m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* mencetak data nasabah asuransi dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 40. *Sequence Diagram* Melihat Aktivitas Saran

*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses melihat aktivitas saran. Aktor yang dapat melihat aktivitas saran dalam *sequence* ini adalah *agent*. Pada *sequence* ini terdapat *class view dashboard* dan *v\_aktivitasSaran*, *class controller c\_nasabah*, *class model m\_nasabah* serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram* melihat aktivitas saran dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 41. *Sequence Diagram Logout*

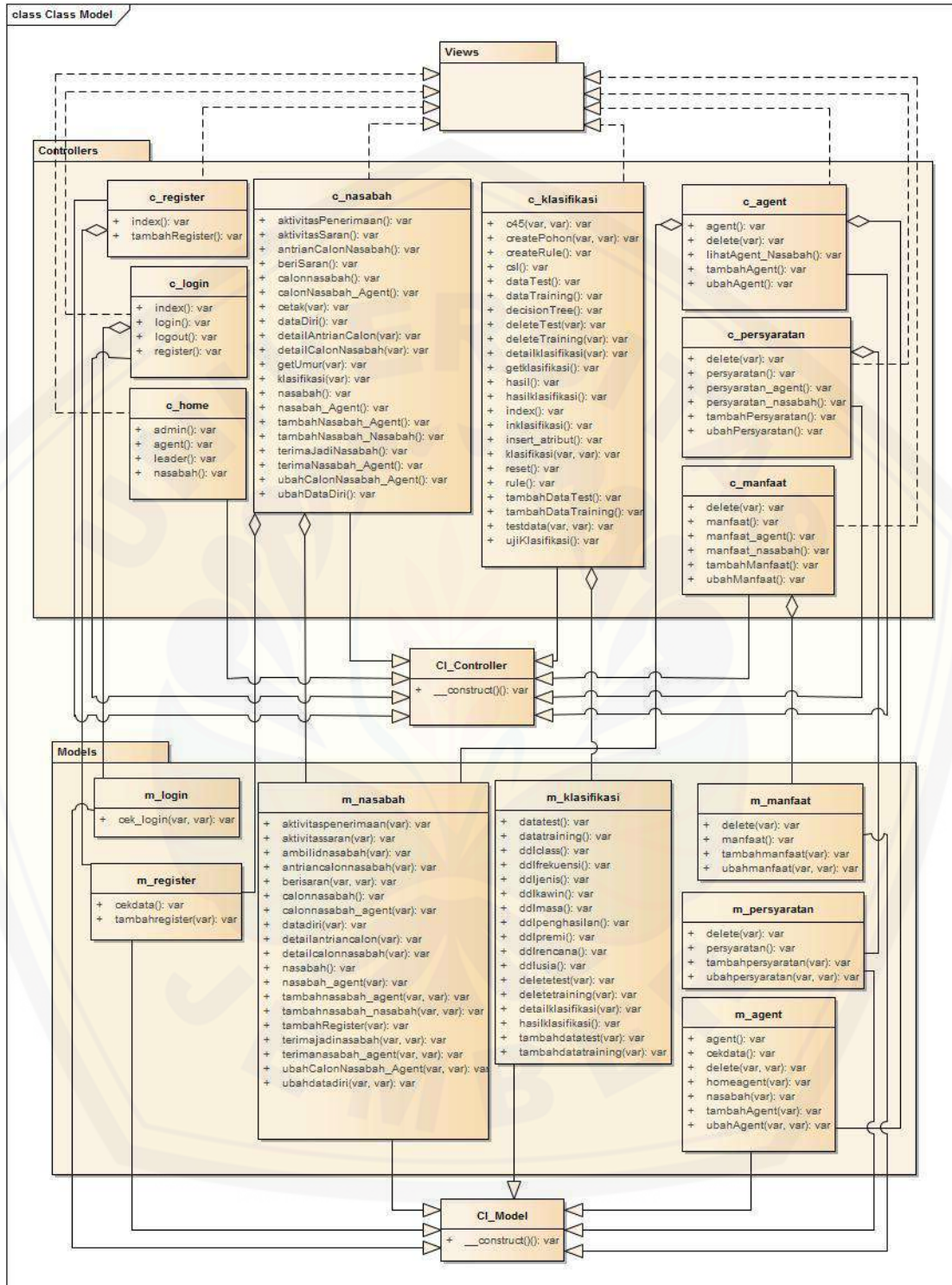
*Sequence diagram* ini menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk melakukan proses *logout*. Semua aktor dapat melakukan proses *logout* dalam *sequence* ini. Pada *sequence* ini terdapat *class view dashboard* dan *index*, *class controller c\_login*, serta dalam *class* tersebut terdapat berbagai *method* yang dibutuhkan dalam proses ini. *Sequence diagram logout* dapat dilihat pada Lampiran D (*Sequence Diagram*).

#### 4.3.6 *Class Diagram*

*Class diagram* menggambarkan hubungan antar kelas yang digunakan untuk membangun suatu sistem, kelas yang digambarkan berisi atribut dan nama *method* pada setiap kelasnya. Dalam paradigma OOD (*Object Oriented Desain*) terdapat 3 jenis kelas yaitu *model*, *view* dan *controller* dan memiliki berbagai macam relasi yang menggambarkan hubungan antar kelas. Berdasarkan *sequence diagram* yang telah dibangun, *class diagram* sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* terdiri dari 7 kelas *model* yaitu *m\_agent*, *m\_klasifikasi*, *m\_login*, *m\_manfaat*, *m\_nasabah*, *m\_persyaratan*, dan *m\_register*, sementara *view* terdiri dari 38 kelas yang digambarkan dengan *loader view* dan 8 kelas *controller* yaitu *c\_agent*, *c\_home*,

*c\_klasifikasi*, *c\_login*, *c\_manfaat*, *c\_nasabah*, *c\_persyaratan*, dan *c\_register*. *Class diagram* sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* dapat dilihat pada Gambar 4.36.

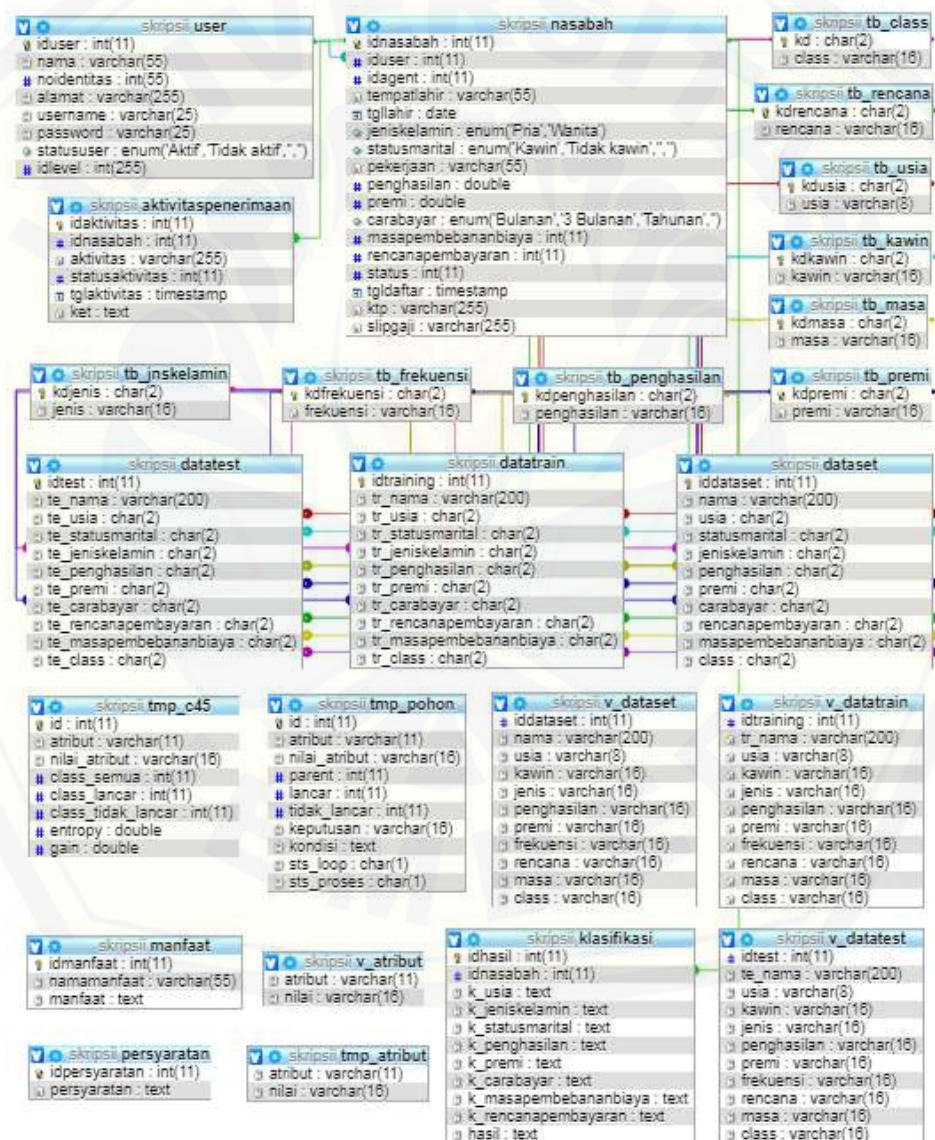




Gambar 4.4. Class Diagram Sibah Asuransi

4.3.7 Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* merupakan gambaran komponen dan struktur database yang saling berhubungan untuk digunakan dalam pembuatan sistem. ERD yang diimplementasikan pada sistem ini terdiri dari 25 entitas yang dapat dilihat pada Gambar 4.5.



Gambar 4.5. Entity Relationship Diagram (ERD) Sibah Asuransi



#### 4.4 Penulisan Kode Program

Desain yang telah dibuat akan diimplementasikan ke dalam kode program. Penulisan kode program (*coding*) menggunakan bahasa pemrograman *Page Hyper Text Pre-Processor (PHP)*, *HyperText Markup Language (HTML)*, *Cascading Style Sheet (CSS)* dan dengan bantuan *framework Code Igniter (CI)*, serta manajemen basisdata menggunakan *DBMS MySQL*.

##### 4.3.1. Perhitungan *Classification Tree C4.5 Algorithm* dan *Cost Sensitive Learning*

Potongan kode program perhitungan *classification tree C4.5 algorithm* dan *cost sensitive learning* terdapat pada *class c\_klasifikasi* pada *package controllers* dan pada *class m\_klasifikasi* pada *package models* kemudian ditampilkan pada *package view v\_decisiontree*. Potongan kode program perhitungan *classification tree C4.5 algorithm* dan *cost sensitive learning* tersebut dapat dilihat pada Tabel 4.15 dan Tabel 4.16.

Tabel 4.15. Potongan Kode Program Perhitungan *C4.5 Algorithm*

No	Kode Program
1	<code>public function index(){</code>
2	<code>    \$this-&gt;reset();</code>
3	<code>    \$this-&gt;insert_atribut();</code>
4	<code>    \$this-&gt;c45(", ");</code>
5	<code>    }</code>
6	<code>public function reset(){</code>
7	<code>    \$this-&gt;db-&gt;query('TRUNCATE tmp_c45');</code>
8	<code>    \$this-&gt;db-&gt;query('TRUNCATE tmp_pohon');</code>
9	<code>    }</code>
10	<code>public function insert_atribut(){</code>
11	<code>    \$this-&gt;db-&gt;query('TRUNCATE tmp_atribut');</code>
12	<code>    \$this-&gt;db-&gt;query('INSERT INTO tmp_atribut SELECT * FROM v_atribut');</code>
	<code>    }</code>
13	<code>public function c45(\$atribut, \$nilai_atribut){</code>
14	<code>    if (empty(\$atribut) &amp;&amp; empty(\$nilai_atribut)) {</code>
15	<code>        \$konndisi_atribut = ";</code>
16	<code>    } elseif (!empty(\$atribut) AND !empty(\$nilai_atribut)) {</code>

No	Kode Program
17	\$get_kondisi = \$this->db->query("SELECT kondisi FROM tmp_pohon
18	WHERE atribut = '\$atribut' AND nilai_atribut = '\$nilai_atribut' order by id
	DESC LIMIT 1")->row_array();
	\$kondisi_atribut = str_replace("*", "", \$get_kondisi['kondisi']);
19	}
20	//Hitung jumlah semua nilai tiap class dan nilai tiap atribut per class
21	\$semua_atribut = \$this->db->query("SELECT DISTINCT atribut FROM
22	tmp_atribut ORDER BY atribut ASC")->result_array();
	foreach (\$semua_atribut as \$data):
23	if (\$data['atribut'] == 'class') {
24	\$sarr_class = \$this->db->query("select count(*) as jumlah_semua,
25	count(case class when 'Lancar' then 1 else NULL end) as
	jumlah_lancar, count(case class when 'Tidak Lancar' then 1 else NULL
	end) as jumlah_tidak_lancar from v_datatrain WHERE 1=1
	\$kondisi_atribut")->row_array();
26	\$this->db->query("INSERT INTO tmp_c45 VALUES(", 'class', 'class', ""
	. \$sarr_class['jumlah_semua'] . ", " . \$sarr_class['jumlah_lancar'] . ", " .
	\$sarr_class['jumlah_tidak_lancar'] . ", ", ")");
	} else {
27	\$sarr_nilai_atribut = \$this->db->query("select nilai from tmp_atribut
28	where atribut = " . \$data['atribut'] . """)->result_array();
	foreach (\$sarr_nilai_atribut as \$ana):
29	\$sarr_class = \$this->db->query("select count(*) as jumlah_semua,
30	count(case class when 'Lancar' then 1 else NULL end) as
	jumlah_lancar, count(case class when 'Tidak Lancar' then 1 else NULL
	end) as jumlah_tidak_lancar from v_datatrain where " . \$data['atribut'] .
	" = " . \$ana['nilai'] . " " \$kondisi_atribut")->row_array();
31	\$this->db->query("INSERT INTO tmp_c45 VALUES(", " .
	\$data['atribut'] . ", " . \$ana['nilai'] . ", " . \$sarr_class['jumlah_semua'] .
	", " . \$sarr_class['jumlah_lancar'] . ", " .
	\$sarr_class['jumlah_tidak_lancar'] . ", ", ")");
	endforeach;
	}
	endforeach;
32	}
33	
34	
35	

Tabel 4.15 merupakan potongan kode program perhitungan *classification tree* C4.5 *algorithm*. Baris 1 – 5 merupakan *method* kode program untuk proses seluruh pembuatan pohon. Baris 6 – 9 merupakan *method* kode program untuk proses mengosongkan tabel tmp\_c45 dan tmp\_pohon. Baris 10 – 13 merupakan *method* kode program untuk proses mengosongkan tabel tmp\_atribut dan mengisi kembali tabel dari view v\_atribut. Baris 14 – seterusnya merupakan *function* kode program untuk proses perhitungan *classification tree* C4.5 *algorithm* yang terdiri dari proses menghitung jumlah semua nilai tiap *class* dan nilai tiap atribut per *class*, menghitung nilai *entropy* tiap *class* dan tiap nilai atribut, menghitung nilai *gain* tiap atribut, memilih *gain* tertinggi dan menjadikannya sebagai akar. Proses pada kode program tersebut dilakukan hingga semua nilai atribut memiliki *class* atau menjadi *node* murni.

Tabel 4.16. Potongan Kode Program Perhitungan *Cost Sensitive Learning*

No	Kode Program
1	public <i>function</i> csl(){
2	\$csl_res = \$this->db->query("SELECT * FROM tmp_pohon WHERE keputusan = '?' and id NOT IN (SELECT parent FROM tmp_pohon)")->result_array();
3	\$bobot_LTL = 30;
4	\$bobot_TLL = 70;
5	foreach(\$csl_res as \$d):
6	\$LTL = \$d['lancar'] * \$bobot_LTL;
7	\$TLL = \$d['tidak_lancar'] * \$bobot_TLL;
8	if(\$LTL < \$TLL){
9	\$keputusan = 'Tidak Lancar';

Tabel 4.16 merupakan potongan kode program perhitungan *cost sensitive learning*. *Method* *csl* ini berfungsi untuk melakukan perhitungan *cost function* sebagai pelabelan *leaf node*.

#### 4.3.2. Perhitungan Uji Klasifikasi

Potongan kode program perhitungan pengujian algoritma terdapat pada *class* *c\_klasifikasi* pada *package controllers* dan pada *class* *m\_klasifikasi* pada *package models* kemudian ditampilkan pada *package view* *v\_test*. Potongan kode program perhitungan uji klasifikasi terdapat tersebut dapat dilihat pada Tabel 4.17.

Tabel 4.17. Potongan Kode Program Uji Klasifikasi

1	<code>public function testdata(\$idparent, \$sarr_post){</code>
2	<code>    if(\$sarr_post == '0'){</code>
3	<code>        \$sarr_post = \$_POST;</code>
4	<code>    }</code>
5	<code>    \$result = \$this-&gt;db-&gt;query("select * from tmp_pohon where parent =</code>
	<code>        '\$idparent')-&gt;result_array();</code>
6	<code>    foreach(\$result as \$dt):</code>
7	<code>        \$atribut = \$dt['atribut'];</code>
8	<code>        \$nilai_atribut = \$dt['nilai_atribut'];</code>
9	<code>        if(\$sarr_post[\$atribut] == \$nilai_atribut){</code>
10	<code>            if(\$dt['keputusan'] === 'Lancar'){</code>
11	<code>                echo 'Lancar';</code>
12	<code>                break;</code>
13	<code>            }elseif(\$dt['keputusan'] === 'Tidak Lancar'){</code>
14	<code>                echo 'Tidak Lancar';</code>
15	<code>                break;</code>
16	<code>            }</code>
17	<code>            \$this-&gt;testdata(\$dt['id'], \$sarr_post);</code>
18	<code>        }</code>
19	<code>    endforeach;</code>
20	<code>}</code>

Tabel 4.17 merupakan potongan kode program perhitungan klasifikasi data test. Setelah diklasifikasi, hasil klasifikasi akan dibandingkan dengan *class* aktual dari data *testing*.

Tabel 4.18. Potongan Kode Program *Confusion matrix*

1	<code>for(var x = 0;x&lt;arr_class.length;x++){</code>
2	<code>    if(arr_prediksi[x] == 'Lancar' &amp;&amp; arr_class[x] == 'Lancar'){</code>
3	<code>        col_1++;</code>
4	<code>    }</code>
5	<code>    if(arr_prediksi[x] == 'Tidak Lancar' &amp;&amp; arr_class[x] == 'Lancar'){</code>

```
6         col_2++;
7     }
8     if(arr_prediksi[x] == 'Lancar' && arr_class[x] == 'Tidak Lancar'){
9         col_3++;
10    }
11    if(arr_prediksi[x] == 'Tidak Lancar' && arr_class[x] == 'Tidak Lancar'){
12        col_4++;
13    }
14    }
15    $('#col_1').text(col_1);
16    $('#col_2').text(col_2);
17    $('#col_3').text(col_3);
18    $('#col_4').text(col_4);
19    var accuracy = (col_1 + col_4) / (col_1 + col_2 + col_3 + col_4);
20    var recall = col_1 / (col_1 + col_2);
21    var precision = col_1 / (col_1 + col_3);
22    $('#accuracy_1').text(accuracy);
23    $('#accuracy_2').text((accuracy * 100).toFixed(2)+"%");
24    $('#recall_1').text(recall);
25    $('#recall_2').text((recall * 100).toFixed(2)+"%");
26    $('#precision_1').text(precision);
27    $('#precision_2').text((precision * 100).toFixed(2)+"%");
```

Tabel 4.18 merupakan potongan kode program perhitungan *confusion matrix* yang terdiri *accuracy*, *precision*, dan *recall*. Melalui perhitungan ini, akan diketahui hasil *accuracy*, *precision*, dan *recall* dari pembangunan *classification tree* yang telah dibuat menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning*.

#### 4.5 Pengujian Sistem

Pengujian sistem dilakukan berguna untuk mengetahui ketidaksesuaian yang telah diimplementasikan dalam kode program dan rancangan sistem. Pengujian dilakukan dengan dua metode yaitu pengujian *White box* dan *Black box*. Hasil pengujian sebagai berikut:

#### 4.6.1 White Box Testing

Pengujian sistem dengan metode *white box* dilakukan untuk menguji sistem dari segi desain dan kode program. Pengujian bertujuan untuk mengevaluasi apakah sistem mampu menghasilkan fungsi, masukan, dan keluaran yang sesuai dengan spesifikasi dari kebutuhan sistem. Tahapan pengujian metode *white box* meliputi:

1. *Listing* program

*Listing* program merupakan kumpulan baris kode yang diuji. Setiap langkah dari kode yang ada diberi nomor ketika menjalankan *statement* biasa atau penggunaan kondisi dalam program, terdapat pada poin A.

2. *Diagram* alir

*Diagram* alir merupakan notasi yang digunakan untuk merepresentasikan aliran kontrol. Aliran kontrol yang digambarkan merupakan hasil penomoran dari *listing* program. *Diagram* alir digambarkan dengan *node-node* (simpul) yang dihubungkan dengan *edge-edge* (garis) yang menggambarkan alur jalannya program, terdapat pada poin B.

3. Kompleksitas siklomatik (*cyclomatic complexity*)

Kompleksitas siklomatik merupakan metrik perangkat lunak yang menyediakan ukuran kuantitatif dari kompleksitas logis suatu program. Bila digunakan dalam konteks teknik pengujian jalur dasar, nilai yang dihitung untuk kompleksitas siklomatik mendefinisikan jumlah jalur independen dalam basis *set* suatu program, terdapat pada poin C. Perhitungan kompleksitas siklomatik menggunakan rumus:

$$V(G) = E - N + 2 \quad (\text{Persamaan 8})$$

Keterangan:

$V(G)$  = Kompleksitas siklomatik

$E$  = Jumlah *edge* (garis)

$N$  = Jumlah *node* (simpul)

#### 4. Jalur independen (*Independent Path*)

Jalur independen adalah setiap jalur yang melalui program, menunjukkan setidaknya satu kumpulan pernyataan-pernyataan pemrosesan atau kondisi baru. Bila dinyatakan dalam grafik alir, jalur independen harus bergerak setidaknya sepanjang satu *edge* yang belum dilintasi sebelum jalur tersebut didefinisi, terdapat pada poin D.

#### 5. Pengujian Basis Set (*Test Case*)

Pada bagian ini diberikan contoh data yang menggambarkan pelaksanaan jalur di basis set. Data yang dieksekusi dimasukkan ke dalam grafik alir apakah sudah melewati basis *set* yang tersedia. Sistem telah memenuhi syarat kelayakan perangkat lunak jika salah satu jalur yang dieksekusi setidaknya satu kali, terdapat pada poin E dan Lampiran F.

Pengujian sistem informasi pengklasifikasian penerimaan calon nasabah asuransi menggunakan *classification tree C4.5 algorithm* dan *cost sensitive learning* akan diterapkan pada fitur yang dinilai dapat mewakili sistem informasi pengklasifikasian penerimaan calon nasabah asuransi sebagai berikut:

##### A. Listing Program

```
17     public function index(){
18         $this->bagidataset();
19         $this->reset();
20         $this->insert_atribut();
21         $this->c45('', '');
22     }
```

Gambar 4.6. Listing Program *index()*

```

29     public function bagidataset(){
30         $this->db->query('TRUNCATE datatrain');
31         $this->db->query('TRUNCATE datatest');
32         $jumlahL = $this->db->query('SELECT COUNT(iddataset) as jumlah FROM dataset where class
= "L")->result_array();
33         $jumlahTL = $this->db->query('SELECT COUNT(iddataset) as jumlah FROM dataset where
class = "TL")->result_array();
34         $prosentL = round(0.9*$jumlahL[0]["jumlah"]);
35         $prosentTL = round(0.9*$jumlahTL[0]["jumlah"]);
36
37         $this->db->query('INSERT INTO datatrain SELECT * FROM dataset where class = "L" limit '.
$prosentL);
38         $this->db->query('INSERT INTO datatrain SELECT * FROM dataset where class = "TL" limit '
.$prosentTL);
39         $this->db->query('INSERT INTO datatest SELECT * FROM dataset where iddataset NOT IN (
SELECT idtraining FROM datatrain)');
40     }

```

Gambar 4.7. Listing Program bagidataset()

```

12     public function reset(){
13         $this->db->query('TRUNCATE tmp_c45');
14         $this->db->query('TRUNCATE tmp_pohon');
15     }

```

Gambar 4.8. Listing Program reset()

```

17     public function insert_atribut(){
18         $this->db->query('TRUNCATE tmp_atribut');
19         $this->db->query('INSERT INTO tmp_atribut SELECT * FROM v_atribut');
20     }

```

Gambar 4.9. Listing Program insert\_atribut()

```

22     public function c45($atribut, $nilai_atribut){
23
24         if (empty($atribut) && empty($nilai_atribut)) {
25             $kondisi_atribut = '';
26         } elseif (!empty($atribut) AND !empty($nilai_atribut)) {
27             $get_kondisi = $this->db->query("SELECT kondisi FROM tmp_pohon WHERE atribut = '$atribut' AND nilai_atribut = '$
nilai_atribut' order by id DESC LIMIT 1")->row_array();
28             $kondisi_atribut = str_replace("=", "", $get_kondisi['kondisi']);
29         }
30         //Hitung jumlah semua nilai tiap class dan nilai tiap atribut per class
31         $semua_atribut = $this->db->query("SELECT DISTINCT atribut FROM tmp_atribut ORDER BY atribut ASC")->result_array();
32         foreach ($semua_atribut as $data):
33             if ($data['atribut'] == 'class') {
34                 $sarr_class = $this->db->query("select count(*) as jumlah_semua, count(case class when 'Lancar' then 1 else NULL
end) as jumlah_lancar, count(case class when 'Tidak Lancar' then 1 else NULL end) as jumlah_tidak_lancar from
v_datatrain WHERE 1=1 $kondisi_atribut")->row_array();
35                 $this->db->query("INSERT INTO tmp_c45 VALUES('', 'class', 'class', '' . $sarr_class['jumlah_semua'] . '', '' . $
arr_class['jumlah_lancar'] . '', '' . $sarr_class['jumlah_tidak_lancar'] . '', '', ''");
36             } else {
37                 $sarr_nilai_atribut = $this->db->query("select nilai from tmp_atribut where atribut = '' . $data['atribut'] . ''")
->result_array();
38                 foreach ($sarr_nilai_atribut as $ana):
39                     $sarr_class = $this->db->query("select count(*) as jumlah_semua, count(case class when 'Lancar' then 1 else
NULL end) as jumlah_lancar, count(case class when 'Tidak Lancar' then 1 else NULL end) as
jumlah_tidak_lancar from v_datatrain where '' . $data['atribut'] . '' = '' . $ana['nilai'] . '' $
kondisi_atribut")->row_array();
40                     $this->db->query("INSERT INTO tmp_c45 VALUES('', '' . $data['atribut'] . '', '' . $ana['nilai'] . '', '' . $
arr_class['jumlah_semua'] . '', '' . $sarr_class['jumlah_lancar'] . '', '' . $sarr_class['
jumlah_tidak_lancar'] . '', '', ''");
41                 endforeach;
42             }
43         endforeach;

```



```

44 // Hitung nilai Entropy
45 $arr_entropy = $this->db->query("SELECT id, class_semua, class_lancar, class_tidak_lancar FROM tmp_c45")->result_array();
46 foreach ($arr_entropy as $ae):
47     if ($ae['class_semua'] == 0 || $ae['class_lancar'] == '0' || $ae['class_tidak_lancar'] == '0') {
48         $entropy = '0';
49     } elseif ($ae['class_lancar'] == $ae['class_tidak_lancar']) {
50         $entropy = '1';
51     } else {
52         $per_lancar = $ae['class_lancar'] / $ae['class_semua'];
53         $per_tidak_lancar = $ae['class_tidak_lancar'] / $ae['class_semua'];
54         $entropy = round(-($per_lancar) * log($per_lancar, 2)) + (-($per_tidak_lancar) * log($per_tidak_lancar, 2)), 9);
55     }
56     $this->db->query("UPDATE tmp_c45 SET entropy = '" . $entropy . "' WHERE id = '" . $ae['id'] . "'");
57 endforeach;
58
59 // Hitung nilai Gain
60 $entropy_total = $this->db->query("SELECT class_semua, entropy FROM tmp_c45 WHERE atribut = 'class'")->row_array();
61 foreach ($semua_atribut as $data):
62     $ambil_nilai = $this->db->query("SELECT class_semua, entropy FROM tmp_c45 WHERE atribut = '" . $data['atribut'] . "'")
63     ->result_array();
64     $jumlah_semua = 0;
65     foreach ($ambil_nilai as $an):
66         $jumlah_semua += ($an['class_semua'] / $entropy_total['class_semua']) * $an['entropy'];
67     endforeach;
68     $gain = round($entropy_total['entropy'] - $jumlah_semua, 9);
69     $this->db->query("UPDATE tmp_c45 SET gain = '" . $gain . "' WHERE atribut = '" . $data['atribut'] . "'");
70 endforeach;
71
72 // Buat Pohon
73 $ambil_max_gain = $this->db->query("SELECT DISTINCT atribut, gain FROM tmp_c45 WHERE gain = (select max(gain) from
74 tmp_c45) LIMIT 1")->row_array();
75 if ($ambil_max_gain['gain'] > 0) {
76     $ambil_atribut = $this->db->query("SELECT * FROM tmp_c45 WHERE atribut = '" . $ambil_max_gain['atribut'] . "'")->
77     result_array();
78     $pes_error_inc = 0;
79     foreach ($ambil_atribut as $aa):
80         if ($aa['class_lancar'] == 0 && $aa['class_tidak_lancar'] == 0) {
81             $keputusan = 'Kosong';
82         } elseif ($aa['class_lancar'] != 0 && $aa['class_tidak_lancar'] == 0) {
83             $keputusan = 'Lancar';
84         } elseif ($aa['class_lancar'] == 0 && $aa['class_tidak_lancar'] != 0) {
85             $keputusan = 'Tidak Lancar';
86         } else {
87             $keputusan = '?';
88         }
89         if (empty($atribut) && empty($nilai_atribut)) {
90             $kondisi_atribut = "AND " . $aa['atribut'] . " = '" . $aa['nilai_atribut'] . "'";
91             $this->db->query("INSERT INTO tmp_pohon VALUES('' . $aa['atribut'] . "', '" . $aa['nilai_atribut'] . "', '" .
92             $aa['class_lancar'] . "', '" . $aa['class_tidak_lancar'] . "', '$keputusan', '$kondisi_atribut', '0', '0')");
93             echo "INSERT INTO tmp_pohon VALUES('' . $aa['atribut'] . "', '" . $aa['nilai_atribut'] . "', '0', '" . $aa
94             ['class_lancar'] . "', '" . $aa['class_tidak_lancar'] . "', '$keputusan', '$kondisi_atribut', '0', '0')<
95             br />";
96         } elseif (!empty($atribut) && !empty($nilai_atribut)) {
97             $parent = $this->db->query("SELECT id, atribut, nilai_atribut, lancar, tidak_lancar FROM tmp_pohon WHERE
98             atribut = '$atribut' AND nilai_atribut = '$nilai_atribut' ORDER BY id DESC LIMIT 1")->row_array();
99             $this->db->query("INSERT INTO tmp_pohon VALUES('' . $aa['atribut'] . "', '" . $aa['nilai_atribut'] . "', '" .
100             $parent['id'] . "', '" . $aa['class_lancar'] . "', '" . $aa['class_tidak_lancar'] . "', '$keputusan',
101             '" . $aa['class_lancar'] . "', '" . $aa['class_tidak_lancar'] . "', '$keputusan', '0', '0')");
102             echo "INSERT INTO tmp_pohon VALUES('' . $aa['atribut'] . "', '" . $aa['nilai_atribut'] . "', '" . $parent[
103             'id'] . "', '" . $aa['class_lancar'] . "', '" . $aa['class_tidak_lancar'] . "', '$keputusan', '0',
104             '0')<br />";
105         }
106     }
107 endforeach;
108 }

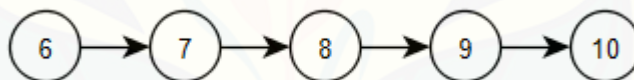
```

```

98 $kondisi_pohon = $this->db->query("SELECT id, kondisi FROM tmp_pohon")->result_array();
99 foreach ($kondisi_pohon as $kp):
100     $parent_belum = $this->db->query("SELECT * FROM tmp_pohon WHERE parent = '" . $kp['id'] . "' AND sts_loop = '0'")->
        result_array();
101     foreach ($parent_belum as $pb):
102         $kondisi_atribut = $kp['kondisi'] . " AND " . $pb['atribut'] . " = " . $pb['nilai_atribut'] . """;
103         $this->db->query("UPDATE tmp_pohon SET kondisi = '$kondisi_atribut', sts_loop = '1' WHERE id = '" . $pb['id'] . "'
        ");
104     endforeach;
105 endforeach;
106
107 $gain_max_atribut = $this->db->query("SELECT distinct atribut FROM tmp_c45 WHERE gain in (SELECT max(gain) FROM tmp_c45)
        LIMIT 1")->row_array();
108 if (!empty($atribut) && !empty($nilai_atribut)) {
109     $this->db->query("UPDATE tmp_pohon SET sts_proses = '1' WHERE nilai_atribut = '$nilai_atribut'");
110 }
111
112 $belum_keputusan = $this->db->query("SELECT * FROM tmp_pohon WHERE keputusan = '?' and sts_proses = '0' AND atribut = '" .
        $gain_max_atribut['atribut'] . """);
113 foreach ($belum_keputusan as $bk):
114     if ($bk['parent'] == 0) {
115         $this->insert_atribut();
116     }
117     $atribut = $bk['atribut'];
118     $nilai_atribut = $bk['nilai_atribut'];
119     $kondisi_atribut = "AND " . $atribut . " = " . $nilai_atribut . """;
120     $this->db->query("TRUNCATE tmp_c45");
121     $this->db->query("DELETE FROM tmp_atribut WHERE atribut = '" . $gain_max_atribut['atribut'] . """);
122     $this->c45($atribut, $nilai_atribut);
123     $this->insert_atribut();
124 endforeach;
125 echo '-----<br/>';
126 }
    
```

Gambar 4.10. Listing Program c45()

B. Diagram Alir



Gambar 4.11. Diagram Alir index()



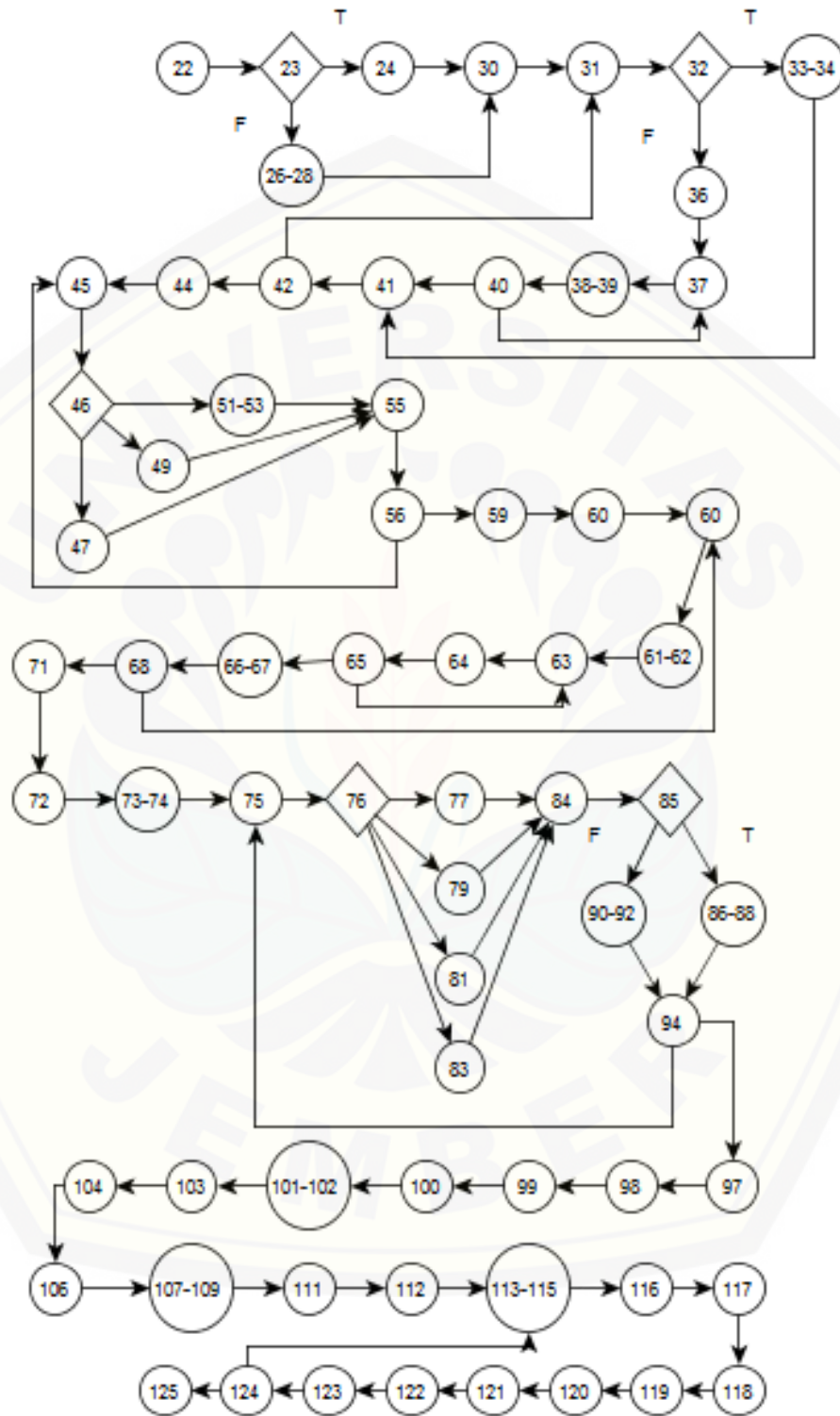
Gambar 4.12. Diagram Alir bagidataset()



Gambar 4.13. Diagram Alir reset()



Gambar 4.14. Diagram Alir insert\_atribut()



Gambar 4.15. Diagram Alir c45()

**C. Kompleksitas Siklomatik**

1. index()

$$V(G) = E - N + 2$$

$$V(G) = 4 - 5 + 2 = 1$$

2. bagidataset()

$$V(G) = E - N + 2$$

$$V(G) = 5 - 6 + 2 = 1$$

3. reset()

$$V(G) = E - N + 2$$

$$V(G) = 3 - 4 + 2 = 1$$

4. insert\_atribut()

$$V(G) = E - N + 2$$

$$V(G) = 3 - 4 + 2 = 1$$

5. c45()

$$V(G) = E - N + 2$$

$$V(G) = 82 - 67 + 2 = 17$$

**D. Jalur Independen**

1. index()

$$\text{Jalur} = 1 - 2 - 3 - 4 - 5$$

2. bagidataset()

$$\text{Jalur} = 1 - 2 - 3 - 4 - 5 - 6$$

3. reset()

$$\text{Jalur} = 1 - 2 - 3 - 4$$

4. insert\_atribut()

$$\text{Jalur} = 1 - 2 - 3 - 4$$

## 5. c45()

Jalur 1 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16 – 17 – 18 – 21 – 22 – 23  
– 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 –  
41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 67

Jalur 2 = 1 – 2 – 4 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16 – 17 – 18 – 21 – 22 – 23  
– 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 –  
41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 67

Jalur 3 = 1 – 2 – 3 – 5 – 6 – 7 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18  
– 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 –  
35 – 36 – 37 – 41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 –  
53 – 67

Jalur 4 = 1 – 2 – 3 – 5 – 6 – 7 – 9 – 10 – 11 – 12 – 10 – 11 – 12 – 13 – 14 – 15  
– 16 – 17 – 18 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 –  
32 – 33 – 34 – 35 – 36 – 37 – 41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 –  
50 – 51 – 52 – 53 – 67

Jalur 5 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 13 – 14 – 6 – 7 – 8 – 13 – 14 – 15 – 16 –  
17 – 18 – 21 – 22 – 23 – 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 –  
33 – 34 – 35 – 36 – 37 – 41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 –  
51 – 52 – 53 – 67

Jalur 6 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16 – 17 – 18 – 21 – 22 – 23  
– 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 –  
41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 67

Jalur 7 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16 – 17 – 19 – 21 – 22 – 23  
– 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 –  
41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 67

Jalur 8 = 1 – 2 – 3 – 5 – 6 – 7 – 8 – 13 – 14 – 15 – 16 – 17 – 20 – 21 – 22 – 23  
– 24 – 25 – 26 – 27 – 28 – 29 – 30 – 31 – 32 – 33 – 34 – 35 – 36 – 37 –  
41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 67



41 – 42 – 43 – 45 – 46 – 47 – 48 – 49 – 50 – 51 – 52 – 53 – 54 – 55 –  
56 – 57 – 66 – 57 – 66 – 67

### E. Pengujian Basis Set

Tabel 4.19. *Test Case Function Index()*

<i>Test case function index()</i>	
<i>Test Case</i>	Jika baris 17-22 dijalankan
Target yang diharapkan	Berhasil melakukan beberapa method yang ada pada method index
Hasil pengujian	Benar
<i>Path/ Jalur</i>	1 – 2 – 3 – 4 – 5

Tabel 4.20. *Test Case Function bagidataset()*

<i>Test case function bagidataset()</i>	
<i>Test Case</i>	Jika baris 29 – 40 dijalankan
Target yang diharapkan	Berhasil melakukan pembagian dataset tabel data set kebel dalam dua bagian, yaitu ke dalam tabel datatrain dan data set.
Hasil pengujian	Benar
<i>Path/ Jalur</i>	1 – 2 – 3 – 4 – 5 – 6

Tabel 4.21. *Test Case Function reset()*

<i>Test case function reset()</i>	
<i>Test Case</i>	Jika baris 12 – 15 dijalankan
Target yang diharapkan	Berhasil melakukan pengosongan tabel perhitungan C4.5 dan tabel pohon
Hasil pengujian	Benar
<i>Path/ Jalur</i>	1 – 2 – 3 – 4

Tabel 4.22. *Test Case Function insert\_atribut()*

<i>Test case function insert_atribut ()</i>	
<i>Test Case</i>	Jika baris 17 – 20 dijalankan
Target yang diharapkan	Berhasil melakukan pengosongan tabel atribut dan mengisi kembali tabel tersebut dari v_atribut
Hasil pengujian	Benar
<i>Path/ Jalur</i>	1 – 2 – 3 – 4

#### 4.6.2 Black Box Testing

Pengujian sistem dengan metode *black box* dilakukan dari segi fungsionalitas *software* yang berkaitan dengan struktur internal atau kerja. Metode ini memfokuskan pada keperluan fungsionalitas dari *software*. Pengujian *black box* pada sistem informasi pengklasifikasian penerimaan calon nasabah asuransi dilakukan untuk mengetahui masukan dan keluaran dari sistem telah sesuai dengan kebutuhan fungsional atau tidak. Pengujian akan diterapkan pada fitur yang dinilai dapat mewakili sistem informasi pengklasifikasian penerimaan calon nasabah asuransi sebagai berikut:

Tabel 4.23. *Black Box Testing* Melihat Hasil Pembangunan *Classification tree*

Nomor <i>Usecase</i>	USC 17		
Nama	Melihat Hasil Pembangunan <i>Classification tree</i>		
Aktor	<i>Admin</i>		
<i>Pre Condition</i>	<i>Admin</i> memilih menu <i>Data Training</i>		
<i>Post Condition</i>	<i>Admin</i> berhasil menampilkan hasil pembangunan <i>classification tree</i> , melihat hasil pembangunan <i>classification tree</i> dan <i>rule</i>		
<b>SKENARIO MELIHAT HASIL PEMBANGUNAN CLASSIFICATION TREE</b>			
Aksi Aktor	Reaksi Sistem	Kesesuaian	
		Benar	Salah
1. Memilih menu <i>Data Training</i>		√	
	2. Menampilkan data <i>training</i> pada tabel di halaman data <i>training</i> : <ul style="list-style-type: none"> <li>- No (<i>int: 11</i>)</li> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> <li>- <i>Actions</i> yang memuat 2 tombol yaitu:               <ul style="list-style-type: none"> <li>• hapus: untuk menghapus data <i>training</i></li> </ul> </li> </ul> Halaman ini juga menampilkan tombol tambah data yang digunakan apabila aktor akan menambahkan data calon	√	



	nasabah, tombol <i>classification tree</i> yang digunakan apabila aktor akan melihat pohon klasifikasi dan <i>update classification tree</i> , tombol <i>rule</i> yang digunakan apabila aktor akan melihat <i>rule</i> , serta kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.		
4. Menekan tombol “ <i>Decision Tree</i> ”		√	
	5. Menampilkan pohon klasifikasi dan tombol <i>update Decision Tree</i>	√	
6. Menekan tombol “ <i>Update Decision Tree</i> ”		√	
	7. Menampilkan <i>form</i> modal bobot <i>cost function</i> : - Lancar ke Tidak Lancar - Tidak Lancar ke Lancar - Tombol Mulai	√	
8. Mengisi <i>form</i> modal bobot <i>cost function</i> : - Lancar ke Tidak Lancar - Tidak Lancar ke Lancar		√	
9. Menekan tombol “Mulai”		√	
	10. Menampilkan hasil <i>update classification tree</i>	√	

Tabel 4.24. *Black Box Testing* Melihat Hasil Uji Klasifikasi

Nomor <i>Usecase</i>	USC 18		
Nama	Melihat Hasil Uji Klasifikasi		
Aktor	<i>Admin</i>		
<i>Pre Condition</i>	<i>Admin</i> memilih menu <i>Data Testing</i>		
<i>Post Condition</i>	<i>Admin</i> berhasil menampilkan hasil uji klasifikasi		
<b>SKENARIO UTAMA MELIHAT HASIL UJI KLASIFIKASI</b>			
Aktor	Sistem	Kesesuaian	
		Benar	Salah
1. Memilih menu <i>Data Testing</i>		√	

2. Mengambil data <i>testing</i> dari <i>database</i>	√	
<p>3. Menampilkan data <i>testing</i> pada tabel di halaman data <i>testing</i></p> <ul style="list-style-type: none"> <li>- No (<i>int: 11</i>)</li> <li>- Nama (<i>varchar: 200</i>)</li> <li>- Usia (<i>varchar: 2</i>)</li> <li>- Jenis Kelamin (<i>varchar: 2</i>)</li> <li>- Status Marital (<i>varchar: 2</i>)</li> <li>- Penghasilan (<i>varchar: 2</i>)</li> <li>- Premi (<i>varchar: 2</i>)</li> <li>- Cara Bayar (<i>varchar: 2</i>)</li> <li>- Masa Pembebanan Biaya (<i>varchar: 2</i>)</li> <li>- Rencana pembayaran (<i>varchar: 2</i>)</li> <li>- <i>Class</i> (<i>varchar: 2</i>)</li> <li>- <i>Actions</i> yang memuat 2 tombol yaitu: <ul style="list-style-type: none"> <li>• hapus: untuk menghapus data <i>test</i></li> </ul> </li> </ul> <p>Halaman ini juga menampilkan tombol tambah data yang digunakan apabila aktor akan menambahkan data <i>testing</i>, tombol proses <i>testing</i> yang digunakan apabila aktor akan menguji hasil pohon klasifikasi, serta kolom pencarian yang digunakan untuk melakukan pencarian data berdasarkan nama.</p>	√	
4. Menekan tombol “Proses <i>Testing</i> ”	√	
5. Menampilkan hasil perhitungan, yaitu <i>matrix</i> perbandingan frekuensi <i>class</i> aktual dan <i>class</i> hasil klasifikasi, serta nilai <i>accuracy</i> , <i>precision</i> , dan <i>recall</i> .	√	

## BAB 6. PENUTUP

Bab ini berisi mengenai kesimpulan dan saran dari peneliti tentang penelitian yang telah dilakukan. Kesimpulan dan saran tersebut diharapkan dapat digunakan sebagai acuan pada penelitian selanjutnya.

### 6.1. Kesimpulan

Kesimpulan dari penelitian yang telah dilakukan adalah sebagai berikut:

1. Memprediksi kelancaran calon nasabah dalam pembayaran premi menggunakan *classification tree c4.5 algorithm* dan *cost sensitive learning*. Algoritma tersebut digunakan untuk proses pembangunan *classification tree* yang dibutuhkan untuk menghasilkan *rule* yang dikonversi dari pohon klasifikasi. *Rule* digunakan untuk proses klasifikasi kelancaran calon nasabah dalam pembayaran premi. *Rule* yang dihasilkan menggunakan metode tersebut menghasilkan nilai *accuracy* sebesar 88.24%, *precision* sebesar 90.29%, dan *recall* sebesar 94.87%, sedangkan *rule* yang dihasilkan *classification tree c4.5 algorithm* tanpa menggunakan *cost sensitive learning* menghasilkan nilai *accuracy* sebesar 76.47%, *precision* sebesar 87.3%, dan *recall* sebesar 82.05%. Data yang digunakan dalam pembangunan *classification tree* menggunakan data calon nasabah asuransi sebanyak 170 data yang terdiri dari atribut usia, jenis kelamin, status marital, penghasilan, premi, cara bayar, rencana pembayaran, dan masa pembebanan biaya.
2. Penerapan *cost sensitive learning* digunakan untuk meminimalisir *misclassification cost* dalam klasifikasi penerimaan calon nasabah asuransi. Penggunaan yang dilakukan adalah *cost function* sebagai kriteria untuk pelabelan *leaf node*. Proses pelabelan *leaf node* pada pembangunan *classification tree*, digunakan bobot *cost function* (nilai kerugian) yang dikalikan dengan nilai atribut

tiap *class* dan memilih nilai atribut dengan nilai *cost function* terkecil sebagai *class*. Kenaikan nilai *recall* pada hasil uji klasifikasi, membuktikan bahwa *cost sensitive learning* dapat meminimalisir *misclassification cost*, sehingga dapat meningkatkan banyaknya proporsi calon nasabah lancar yang bisa ditemukan oleh sistem.

3. Sistem informasi pengklasifikasian penerimaan calon nasabah asuransi dibangun berbasis *website* dengan empat hak akses yaitu *admin*, calon nasabah, *agent*, dan *leader* dengan fitur utama klasifikasi kelancaran calon nasabah dalam pembayaran premi oleh *leader*. Sistem dikembangkan dengan menggunakan SDLC model *waterfall*. Model *waterfall* digunakan karena kebutuhan sistem telah terdefiniskan diawal pembangunan sistem. Tahap pengembangan meliputi analisis kebutuhan, desain sistem, implementasi, pengujian sistem, dan pemeliharaan.

## 6.2. Saran

Adapun saran yang ditujukan untuk memberikan masukan yang lebih baik adalah sebagai berikut:

1. Klasifikasi penerimaan calon nasabah asuransi disarankan tidak hanya ditentukan dari segi kelancaran calon nasabah dalam pembayaran premi, namun juga dengan menambah atribut yang lebih banyak lagi sesuai dengan peraturan penerimaan calon nasabah asuransi agar nilai akurasi yang dihasilkan lebih tinggi, serta menggunakan data *set* yang lebih banyak.
2. Proses pengelompokan penerimaan calon nasabah asuransi dapat dilakukan dengan metode yang berbeda atau mengkombinasikan *classification tree c4.5 algorithm* dan *cost sensitive learning* dengan metode lainnya.

**DAFTAR PUSTAKA**

- Aini, Z. Q. (2015). Sistem Informasi Klinik Mata Dengan Prediksi Waktu Pemeriksaan Menggunakan Algoritma C4.5 Pada Fitur Registrasi Pasien.
- Alshamrani, A., & Bahttab, A. (2015). A Comparison Between Three SDLC Model Waterfall Model, Spiral Model, and Incremental/Iterative Model. *International Journal of Computer Science Issue*, 106 - 111.
- Amin, R. K., Indwiarti, & Sibaroni, Y. (2015, April 1). Implementasi Klasifikasi Decision Tree dengan Algoritma C4.5 dalam Pengambilan Keputusan Permohonan Kredit Oleh Debitur (Studi Kasus: Bank Pasar Daerah Istimewa Yogyakarta). *Jurnal Tugas Akhir*, 1.
- Bassil, Y. (2012). A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering & Technology (iJET)*, 2049 - 3444.
- Bertalya. (2009). *Konsep Data Mining (Klasifikasi: Pohon Keputusan)*. Universitas Gunadarma.
- Gulo, W. (2000). *Metodologi Penelitian*. Jakarta: Gasindo.
- Han, J., Kamber, M., & Pei, J. (2001). *Data Mining: Concepts and Techniques*. USA: Morgan Kaufmann.
- Hartono, S. R. (1995). *Hukum Asuransi dan Perusahaan Asuransi*. Jakarta: Sinar Grafika.
- Jefri. (2013). Implementasi Algoritma C4.5 Dalam Aplikasi Untuk Memprediksi Jumlah Mahasiswa Yang Mengulang Mata Kuliah Di STMIK AMIKOM Yogyakarta.

- Jogiyanto, H. M. (2005). *Analisis dan Desain Sistem Informasi : pendekatan Terstruktur Teori dan Praktik Aplikasi Bisnis*. Yogyakarta: Andi Publisher.
- Kadir, A. (2014). *Pengenalan Sistem Informasi Edisi Revisi*. Yogyakarta: Andi Publisher.
- Kusrini, & Luthfi, E. T. (2009). *Algoritma Data Mining*. Yogyakarta: Andi Offset.
- Larose, D. T. (2014). *Discovering Knowledge in Data: An Introduction to Data Mining, Second Edition*. Canada: John Wiley & Sons, Inc.
- Rahayu, E. S., Wahono, R. S., & Supriyanto, C. (2015). Penerapan Metode Average Gain, Threshold Pruning dan Cost Complexity Pruning untuk Split Atribut pada Algoritma C4.5. *Journal of Intelligent Systems, 1*.
- Sartika, D., & Sensuse, D. I. (2017, Maret 2). Perbandingan Algoritma Klasifikasi Naive Bayes, K-Nearest Neighbour, dan Decision Tree pada Studi Kasus Pengambilan Keputusan Pemilihan Pola Pakaian. *Jatiji, 1(2)*, 151-161.
- Swastina, L. (2013, Juni). Penerapan Algoritma C4.5 Untuk Penentuan Jurusan Mahasiswa. *Jurnal GEMA AKTUALITA, 2(1)*.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Boston.
- Turban, E., Aronson, J., & Liang, T.-P. (2005). *Decision Support Systems and Intelligent Systems*. USA: Pearson Education.
- Wicaksono, S. A. (2010). Pembangunan Model Prediksi Defect Pengembangan Perangkat Lunak Menggunakan Metode Ensemble Decision Tree Dan Cost Sensitive Learning.