

**KLASIFIKASI CALON NASABAH BARU
PEMBERIAN PERTANGGUNGAN ASURANSI
MENGUNAKAN ALGORITMA DECISION
TREE C4.5**

**(Studi Kasus Pada : PT. Prudential Life Assurance
GL1 Gorontalo)**

Oleh

FERDIANSYAH R. TANGKILISAN

T3119003

SKRIPSI

**Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar sarjana**



**PROGRAM SARJANA
TEKNIK INFORMATIKA
UNIVERSITAS ICHSAN GORONTALO
GORONTALO
2023**

PERSETUJUAN SKRIPSI

KLASIFIKASI CALON NASABAH BARU PEMBERIAN PERTANGGUNGAN ASURANSI MENGUNAKAN ALGORITMA DECISION TREE C4.5

**(Studi Kasus Pada : PT. Prudential Life Assurance
GL1 Gorontalo)**

Oleh

FERDIANSYAH R. TANGKILISAN

T3119003

SKRIPSI

Untuk memenuhi salah satu syarat ujian
Guna memperoleh gelar sarjana
Program Studi Teknik Informatika,
ini telah disetujui oleh Tim Pembimbing

Gorontalo, November 2023

Pembimbing I



Amiruddin, M.Kom, MCF
NIDN : 0910097601

Pembimbing II



Hastuti Dalai, M.Kom
NIDN : 0918038803

PENGESAHAN SKRIPSI
KLASIFIKASI CALON NASABAH BARU
PEMBERIAN PERTANGGUNGAN ASURANSI
MENGGUNAKAN ALGORITMA DECISION
TREE C4.5

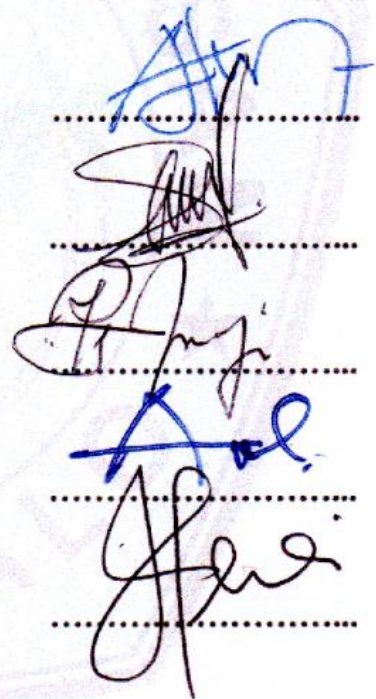
(Studi Kasus Pada : PT. Prudential Life Assurance
GL1 Gorontalo)

Oleh
FERDIANSYAH R. TANGKILISAN
T3119003

SKRIPSI

Diperiksa oleh Panitia Ujian Strata Satu (S1)
Universitas Ichsan Gorontalo

1. Ketua Penguji
Yasin Aril Mustofa, M.Kom
2. Anggota
Andi Bode, M.Kom
3. Anggota
Kartika Chandra Pelangi, M.Kom
4. Anggota
Amiruddin, M.Kom, MCF
5. Anggota
Hastuti Dalai, M.Kom



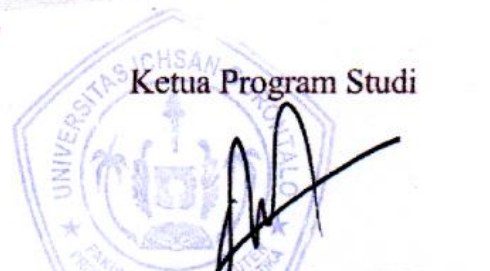
Mengetahui,

Dekan Fakultas Ilmu Komputer



Irvan A. Salihi, M.kom
NIDN : 0928028101

Ketua Program Studi



Sudirman S. Panna M.kom
NIDN : 0924038205

PERNYATAAN SKRIPSI

Dengan ini menyatakan bahwa :

1. Karya tulis (Skripsi) saya ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (Sarjana) baik di Universitas Ichsan Gorontalo maupun di perguruan tinggi lainnya.
2. Karya tulis (Skripsi) saya ini adalah murni gagasan, rumusan dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan dari tim pembimbing.
3. Dalam karya tulis (Skripsi) saya ini tidak terdapat karya atau pendapat yang telah di publikasikan orang lain, kecuali secara tertulis dicantumkan sebagai acuan/sitasi dalam naskah dan dicantumkan pula dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila dikemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya tulis ini, serta sanksi lainnya sesuai dengan norma-norma yang berlaku di Universitas Ichsan Gorontalo.

Gorontalo, November 2023

Yang Membuat Pernyataan



Ferdiansyah R. Tangkilisan

ABSTRACT

FERDIANSYAH R. TANGKILISAN. T3119003. THE DATA CLASSIFICATION OF POTENTIAL NEW CUSTOMERS IN THE INSURANCE COVERAGE GRANTING USING THE DECISION TREE ALGORITHM OF C4.5

Insurance companies often have problems that generally occur, namely customer arrears in paying premiums. Delinquent policy premium payments from customers will result in lapsed (inactive) or termination of coverage because payments are in arrears and have passed the due date set by the insurance company. Insurance companies have a large amount of data. Many do not realize that processing this data can provide beneficial information in the form of data classification of potential new customers to find whether each customer who takes part in this insurance program is suitable for the company in the future. With the data mining process from customer data, particular patterns of relationships among data can be found to be beneficial information. The C4.5 Decision Tree algorithm functions to produce a decision tree. The application of the C4.5 Decision Tree algorithm can be used to classify potential new customers to take part in an insurance program by determining a decision tree based on existing data. In this research, data classification of potential new customers is carried out using the C4.5 Decision Tree algorithm with data attributes, including gender, type of job, age, income, and health history. The classification results obtain an accuracy level of 96%, a precision value of 95%, and a Recall of 96% in the model evaluation process using the Confusion Matrix. Then, rules are obtained from the decision tree that will later become the basis for future agencies in the evaluation process for granting insurance coverage.

Keywords: data mining, classification, potential new customers, C4.5 Decision Tree

ABSTRAK

FERDIANSYAH R. TANGKILISAN. T3119003. KLASIFIKASI DATA CALON NASABAH BARU PEMBERIAN PERTANGGUNGAN ASURANSI MENGGUNAKAN ALGORITMA DECISION TREE C4.5.

Perusahaan asuransi sering kali memiliki permasalahan yang umumnya terjadi yaitu penunggakan nasabah dalam membayar premi. Penunggakan pembayaran premi polis dari nasabah akan menyebabkan lapsed (non-aktif) atau pemberhentian pertanggungan dikarenakan pembayaran yang menunggak dan telah melewati masa jatuh tempo yang telah ditetapkan perusahaan asuransi. Perusahaan asuransi mempunyai data yang besar, masih banyak yang belum menyadari bahwa dari pengolahan data-data tersebut dapat memberikan informasi yang bermanfaat berupa klasifikasi data calon nasabah baru guna untuk mengetahui setiap nasabah yang mengikuti program asuransi ini apakah layak atau tidak layak bagi perusahaan kedepannya. Dengan proses data mining dari data nasabah, dapat ditemukan polapola ataupun hubungan tertentu antara data untuk menjadi informasi yang bermanfaat. Algoritma *Decision tree* C4.5 merupakan algoritma yang digunakan untuk menghasilkan sebuah pohon keputusan, Dalam penerapan algoritma *Decision Tree* C4.5, dapat digunakan untuk melakukan klasifikasi terhadap calon nasabah baru yang berpotensi untuk mengikuti program asuransi dengan penentuan pohon keputusan berdasarkan data-data yang sudah ada. Pada penelitian ini dilakukan klasifikasi data calon nasabah baru menggunakan algoritma *Decision Tree* C4.5 dengan atribut data diantaranya : jenis kelamin, jenis pekerjaan, usia, penghasilan, dan riwayat kesehatan, didapatkan hasil klasifikasi dengan tingkat akurasi sebesar 96%, nilai *Precision* 95%, dan *Recall* 96% pada proses evaluasi model menggunakan *Confusion Matrix*. kemudian didapatkan rule dari pohon keputusan yang nantinya akan menjadi landasan bagi instansi kedepannya dalam proses evaluasi untuk pemberian pertanggungan asuransi.

Kata kunci: data mining, klasifikasi, calon nasabah baru, *Decision Tree* C4.5

KATA PENGANTAR

Puji dan syukur atas kehadiran Allah, SWT karena dengan izin dan karunia-Nya penulis dapat menyelesaikan penyusunan skripsi dengan judul **“Klasifikasi Calon Nasabah Baru Pemberian Pertanggungan Asuransi Menggunakan Algoritma Decision Tree C4.5”** Studi Kasus Pada PT. Prudential Life Assurance GL1 Gorontalo.

Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu penulis mengucapkan permohonan maaf apabila terdapat kesalahan baik dari segi penyusunan, penulisan, maupun penggunaan kata dan bahasa. Penulis sangat mengharapkan masukan kritik dan saran dari pembaca sebagai motivasi untuk penulis agar jauh lebih baik lagi di masa mendatang.

Dengan segala keikhlasan dan kerendahan hati, penulis mengucapkan terima kasih kepada pihak-pihak yang telah banyak membantu terkait penyusunan skripsi ini :

1. Ibu Dr. Dra. Juriko Abdussamad, M.Si, selaku Ketua Yayasan Pengembangan Ilmu Pengetahuan dan Teknologi (YPIPT) Ichsan Gorontalo;
2. Bapak Dr. Abdul Gaffar La Tjokke, M.Si, selaku Rektor Universitas Ichsan Gorontalo;
3. Bapak Irvan A. Salihi, S.Kom, M.Kom, selaku Dekan Fakultas Ilmu Komputer Universitas Ichsan Gorontalo;
4. Bapak Sudirman Melangi, M.Kom, selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer;
5. Ibu Irma Surya Kumala, M.Kom, selaku Wakil Dekan II bidang Administrasi Umum dan Keuangan Fakultas Ilmu Komputer;
6. Bapak Sudirman S. Panna, M.Kom, selaku Ketua Jurusan Teknik Informatika Fakultas ilmu Komputer Universitas Ichsan Gorontalo;
7. Bapak Amiruddin, M.Kom, MCF, selaku Pembimbing I yang telah membimbing dan membantu penulis dalam menyelesaikan skripsi;
8. Ibu Hastuti Dalai, M.Kom, selaku Pembimbing II yang telah membimbing dan membantu penulis dalam menyelesaikan skripsi;

9. Bapak dan Ibu Dosen Universitas Ichsan Gorontalo yang sudah mendidik dan mengajarkan berbagai ilmu yang berguna dan bermanfaat di masa depan bagi penulis;
10. Kepada kedua Orang Tua yang selalu memberikan dukungan dan doa bagi penulis;
11. Kepada seluruh teman-teman yang telah memberikan semangat bagi penulis;
12. Kepada semua pihak yang membantu dalam menyelesaikan skripsi ini.

Semoga Allah, SWT melimpahkan balasan atas jasa dari mereka kepada penulis. Dengan demikian penulis mengharapkan skripsi ini dapat menambah wawasan bagi para pembaca dan bisa bermanfaat untuk perkembangan dan peningkatan ilmu pengetahuan.

Gorontalo, November 2023

Penulis

DAFTAR ISI

PERSETUJUAN SKRIPSI.....	i
PERNYATAAN SKRIPSI.....	iii
ABSTRAK.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL	xi
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	4
1.3 Rumusan Masalah.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
BAB II LANDASAN TEORI.....	6
2.1 Tinjauan Studi.....	6
2.2 Tinjauan Pustaka.....	8
2.2.1 Pemberian Pertanggungungan Asuransi.....	8
2.2.2 <i>Data Mining</i>	9
2.2.3 Klasifikasi	11
2.2.4 Split Validation.....	12
2.2.5 Algoritma C4.5.....	12
2.2.6 Penerapan Algoritma Decision Tree C4.5	15
2.2.7 Python	20
2.2.8 Confusion Matrix.....	21
2.2.9 Perangkat Lunak.....	22
2.2.10 Kerangka Pikir.....	23
BAB III METODE PENELITIAN.....	24
3.1 Jenis, Metode, Subjek, Objek, Waktu, Dan Lokasi Penelitian	24
3.2 Pengumpulan Data	24
3.3 Pemodelan	26
3.3.1 Pra Pengolahan Data.....	26
3.3.2 Validasi Data	26

3.3.3	Pengembangan Model.....	27
3.4	Evaluasi Model	28
BAB IV	HASIL PENELITIAN	29
4.1	Hasil Pengumpulan Data	29
4.2	Hasil Pemodelan	30
4.3	Hasil Pengembangan Model.....	40
4.3.1	Seleksi Data.....	40
4.3.2	Preprocessing	41
4.3.3	Transformasi	42
4.3.4	Proses Data Mining	44
4.3.5	Evaluasi.....	46
4.3.6	Perbaikan Model.....	49
4.3.7	Pemodelan Hasil Optimalisasi	52
4.3.8	Klasifikasi Data Testing Baru	55
BAB V	PEMBAHASAN PENELITIAN	58
5.1	Pembahasan Kinerja Model.....	58
5.2	Pembahasan Hasil Pemodelan	61
5.2.1	Hasil Pemodelan Sebelum Optimalisasi	61
5.2.2	Hasil Pemodelan Setelah Optimalisasi	62
BAB VI	KESIMPULAN DAN SARAN.....	64
6.1	Kesimpulan.....	64
6.2	Saran.....	64
DAFTAR PUSTAKA.....		65
LAMPIRAN		66

DAFTAR GAMBAR

Gambar 1.1	Grafik Status Pembayaran Nasabah	2
Gambar 2.1	Proses Knowledge Discovery in Database (KDD)	9
Gambar 2.2	Tahapan Proses Klasifikasi.....	12
Gambar 2.3	Alur Proses Algoritma C4.5	13
Gambar 2.4	Contoh Pohon Keputusan Dan Rule	14
Gambar 2.5	Pohon Keputusan Hasil Perhitungan Node 1.....	19
Gambar 2.6	Pohon Keputusan Hasil Perhitungan Node 1.1.....	20
Gambar 2.7	Bagan Kerangka Pikir	23
Gambar 3.1	Pengembangan Model Klasifikasi Data Calon Nasabah Baru .	27
Gambar 4.1	Grafik Jumlah Kelas.....	32
Gambar 4.2	Grafik Jumlah Kelas Setelah Perbaikan	33
Gambar 4.3	Hasil Perhitungan Confusion Matrix Uji Coba 1	35
Gambar 4.4	Visualisasi Pemodelan Decision Tree C4.5.....	36
Gambar 4.5	Grafik Kepentingan Atribut.....	37
Gambar 4.6	Hasil Perhitungan Confusion Matrix Setelah Optimalisasi	38
Gambar 4.7	Visualisasi Model Setelah Optimalisasi	38
Gambar 4.8	Grafik Jumlah Kelas.....	42
Gambar 4.9	Confusion Matrix Uji Coba 1	47
Gambar 4.10	Visualisasi Pemodelan Decision Tree C4.5.....	48
Gambar 4.11	Grafik Presentase Atribut	49
Gambar 4.12	Confusion Matrix Setelah Optimalisasi	53
Gambar 4.13	Confusion Matrix Setelah Optimalisasi	54
Gambar 5.1	Confusion Matrix Sebelum Optimalisasi	58
Gambar 5.2	Confusion Matrix Setelah Optimalisasi	59
Gambar 5.3	Pohon Keputusan Sebelum Optimalisasi	61
Gambar 5.4	Pohon Keputusan Setelah Optimalisasi.....	62

DAFTAR TABEL

Tabel 2.1	Penelitian Terkait Tentang <i>Klasifikasi</i> Dengan Penerapan Algoritma <i>Decision Tree C4.5</i>	6
Tabel 2.2	Dataset Calon Nasabah Asuransi	15
Tabel 2.3	Kriteria Usia.....	15
Tabel 2.4	Kriteria Penghasilan	16
Tabel 2.5	Kriteria Jumlah Tanggungan	16
Tabel 2.6	Data Training	16
Tabel 2.7	Data Testing.....	17
Tabel 2.8	Hasil Perhitungan <i>Entropy</i> Total Keseluruhan	18
Tabel 2.9	Perhitungan Node 1.....	18
Tabel 2.10	Perhitungan Node 1.1	19
Tabel 2.11	Tabel <i>Confusion Matrix</i> 2 Kelas	21
Tabel 2.12	Perangkat Lunak Pendukung	22
Tabel 3.1	Data Calon Nasabah Baru.....	25
Tabel 3.2	Variabel/Atribut Data.....	25
Tabel 4.1	Hasil Pengumpulan Data Calon Nasabah Baru	29
Tabel 4.2	Dataset Pemodelan	31
Tabel 4.3	Data Tidak Konsisten	32
Tabel 4.4	Kriteria Jenis Kelamin.....	33
Tabel 4.5	Kriteria Jenis Pekerjaan.....	34
Tabel 4.6	Kriteria Penghasilan	34
Tabel 4.7	Kriteria Riwayat Kesehatan	34
Tabel 4.8	Dataset Hasil Transformasi	34
Tabel 4.9	Hasil Perhitungan Kedalaman Pohon Dan Akurasinya.....	37
Tabel 4.10	Data Testing Baru	39
Tabel 4.11	Hasil Klasifikasi Data Testing Baru.....	39
Tabel 4.12	Dataset Training.....	45
Tabel 4.13	Dataset Testing.....	45

Tabel 5.1	Tabel Perhitungan Confusion Matrix Sebelum Optimalisasi ...	59
Tabel 5.2	Perhitugan Confusion Matrix Setelah Optimalisasi.....	60
Tabel 5.3	Rule Akhir	63

BAB I

PENDAHULUAN

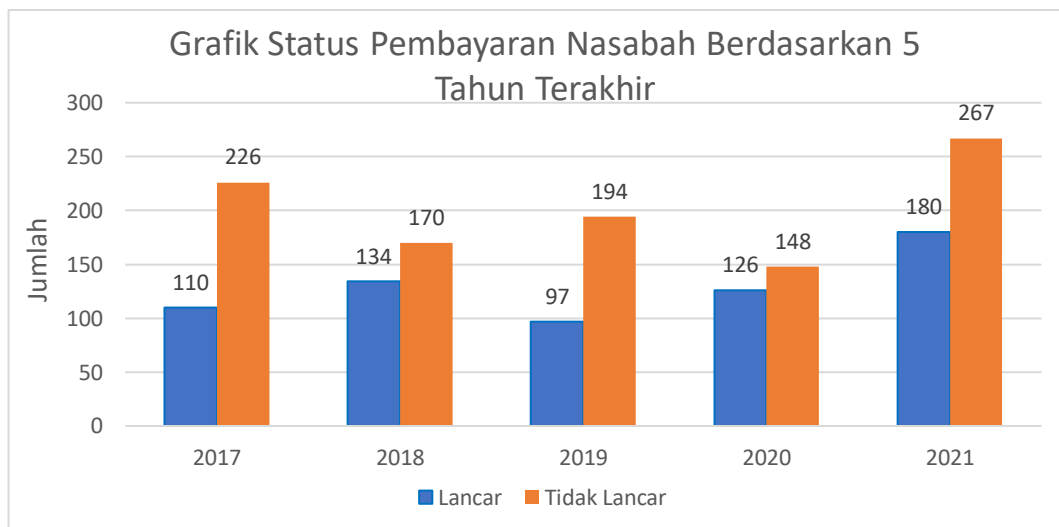
1.1 Latar Belakang

Asuransi adalah sebuah pertanggungan yang merupakan perjanjian dengan melibatkan dua belah pihak atau lebih, Yang mana pihak yang menanggung mempunyai keterkaitan dengan pihak tertanggung [1]. Dengan diterimanya polis asuransi, sebagai pemberian pertanggungan kepada pihak tertanggung yang disebabkan oleh kerugian, kerusakan dan kehilangan keuntungan yang diharapkan, atau untuk pembayaran atas dasar meninggal dan hidupnya seseorang yang di pertanggungkan.

Premi adalah pendapatan yang diperoleh perusahaan asuransi dengan jumlahnya telah ditentukan dalam presentase dan tarif tertentu dari jumlah yang di pertanggungkan [1]. Dari segi pandang tertanggung premi merupakan sebuah hal yang harus dipenuhi bahkan menjadi beban yang harus ditempuh oleh tertanggung sampai batas kontrak dengan perusahaan asuransi telah terpenuhi. Pendapatan dari premi untuk perusahaan asuransi ditentukan dari jumlah premi yang dibayarkan oleh nasabah. Perusahaan asuransi mempertimbangkan hal-hal yang akan menentukan tarif dari premi nasabahnya seperti profesi, jenis kelamin, riwayat kesehatan, nilai premi, usia.

PT. Prudential Life Assurance berdiri pada tahun 1995 dan juga merupakan bagian dari Prudential PLC, sebuah perusahaan jasa keuangan terkemuka di Inggris. Sebagai bagian dari grup yang memiliki pengalaman selama 168 tahun di perindustrian asuransi jiwa [2]. PT. Prudential Life Assurance menawarkan premi yang setiap preminya terdapat jenis pertanggungan di antaranya yaitu perlindungan rawat inap, kecelakaan, sakit kritis, meninggal atau warisan, dan pendidikan untuk anak-anak. Masing – masing jenis pertanggungan dari premi memiliki jumlah kontribusi yang berbeda.

Pada PT. Prudential Life Assurance GL1 Gorontalo didapati permasalahan yang umumnya terjadi yaitu penunggakan nasabah dalam membayar premi, akibat dari menunggaknya pembayaran premi polis dari nasabah akan mengalami lapsed (non-aktif) atau pemberhentian pertanggungan dikarenakan pembayaran yang menunggak dan telah melewati masa jatuh tempo yang telah ditetapkan perusahaan asuransi. Hal ini tentu dapat mempengaruhi nilai RBC (Risk Based Capital) dari perusahaan asuransi. Maka dari itu diperlukan upaya untuk mengetahui faktor-faktor apa saja yang dapat mempengaruhi kemampuan nasabah dalam mengikuti program asuransi.



Gambar 1.1 Grafik Status Pembayaran Nasabah

Pada gambar 1.1 menunjukkan grafik dari nasabah yang status pembayarannya lancar dan tidak lancar dalam membayar premi asuransi. Berdasarkan 5 tahun terakhir periode 2017 sampai dengan 2021 silam terdapat peningkatan dari nasabah dengan status pembayarannya di katakan tidak lancar. ketidaklancaran dari nasabah yang membayar premi disebabkan oleh beberapa faktor termasuk: ekonomi dari nasabah, nasabah yang kurang pemahaman terhadap asuransi, nasabah yang sering tidak tepat waktu dalam membayar premi, nasabah yang sering menunda-nunda pembayaran premi polis.

Selain itu adanya kurang kesadaran atau keseriusan dari nasabah dalam mengikuti program asuransi yang mengakibatkan polis dari nasabah tersebut mengalami lapsed atau non-aktif. Diperlukan sebuah strategi kedepannya untuk mengatasi hal yang memungkinkan pihak perusahaan asuransi dapat menentukan nasabah baru yang layak dan tidak layak dalam mengikuti asuransi.

Klasifikasi adalah sebuah proses menyusun data secara sistematis dalam mengelompokkan sesuatu dengan proses membedakan dan mendistribusikan jenis “hal” ke dalam kelompok yang berbeda. Sebuah perusahaan asuransi mempunyai data yang besar, masih banyak yang belum menyadari bahwa dari pengolahan data-data tersebut dapat memberikan informasi yang bermanfaat berupa klasifikasi data calon nasabah baru guna untuk mengetahui setiap nasabah yang mengikuti program asuransi ini apakah layak atau tidak layak bagi perusahaan asuransi kedepannya. Dengan proses data mining dari data nasabah, dapat ditemukan pola-pola ataupun hubungan tertentu antara data untuk menjadi informasi yang bermanfaat.

Algoritma C4.5 merupakan algoritma yang digunakan untuk menghasilkan sebuah pohon keputusan yang dikembangkan oleh Ross quinlan. Dasarnya algoritma ini memiliki proses pembuatan pohon keputusan berdasarkan pemilihan atribut dengan prioritas tertinggi atau bisa disebut mempunyai nilai gain tertinggi dari nilai *entropy* atribut tersebut. Dalam penerapan algoritma C4.5, dapat digunakan untuk melakukan klasifikasi terhadap calon nasabah baru yang berpotensi untuk mengikuti program asuransi dengan penentuan pohon keputusan berdasarkan data-data yang sudah ada dan melakukan klasifikasi terhadap calon nasabah baru yang ingin mengikuti program asuransi.

Berdasarkan penelitian yang membahas tentang klasifikasi calon nasabah baru menggunakan algoritma C4.5 oleh (Retno Ayu Syahfitri, Agus Perdana Windarto, Harly Okprana, 2021) yang memperoleh hasil dari *performance vector* bahwa menerapkan algoritma C4.5 dalam mengklasifikasikan data calon nasabah baru menggunakan *software RapidMiner* memperoleh hasil tingkat angka akurasi sebesar 96,67% dengan nilai *precision* sebesar 91,38% dan nilai *recall* sebesar 75,00% [3].

Dari uraian latar belakang diatas, penulis tertarik melakukan sebuah penelitian dengan judul **“Klasifikasi Calon Nasabah Baru Pemberian Pertanggungans Asuransi Menggunakan Algoritma *Decision Tree C4.5* (Studi Kasus Pada : PT. Prudential Life Assurance GL1 Gorontalo)”**.

1.2 Identifikasi Masalah

Sesuai dengan hasil uraian latar belakang, perusahaan memiliki permasalahan pada nasabah yang sering kali mengalami lapsed atau (non aktif) polis berdasarkan faktor-faktor tertentu, hal ini dapat mempengaruhi nilai keuntungan dari suatu perusahaan asuransi dan akan merugikan pihak asuransi dan nasabahnya.

1.3 Rumusan Masalah

Dari hasil penjelasan masalah pada latar belakang diperoleh rumusan masalah :

1. Bagaimana penerapan *Decision Tree C4.5* terhadap klasifikasi data calon nasabah baru pada PT.Prudential Life Assurance GL1 Gorontalo?.
2. Bagaimana tingkat akurasi penerapan metode *Decision tree C4.5* dalam klasifikasi data calon nasabah baru?.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini yaitu senantiasa menyelesaikan permasalahan yang terjadi pada perusahaan yakni :

1. Menerapkan metode *Decision Tree C4.5* dalam klasifikasi data calon nasabah baru pada PT.Prudential Life Assurance GL1 Gorontalo.
2. Mengukur tingkat akurasi dari hasil penerapan metode *Decision Tree C4.5* dalam klasifikasi data calon nasabah baru.

1.5 Manfaat Penelitian

Dengan adanya penelitian yang dilakukan diharapkan memberikan manfaat secara teoritis dan praktis :

1. Secara teoritis, penelitian ini dapat mengembangkan ilmu pengetahuan, khususnya pada bidang ilmu komputer dengan penerapan metode yang digunakan.
2. Secara praktis, penelitian ini diharapkan dapat memberikan wawasan dan pengetahuan bagi instansi terhadap nasabah asuransi yang layak dan bijak dalam berasuransi.

BAB II

LANDASAN TEORI

2.1 Tinjauan Studi

Adapun penelitian terkait tentang *Klasifikasi* dengan penerapan algoritma *Decision Tree C4.5* sebagai berikut :

Tabel 2.1 Penelitian Terkait Tentang *Klasifikasi* Dengan Penerapan Algoritma *Decision Tree C4.5*

NO	PENELITI	JUDUL	HASIL
1.	Retno Ayu Syahfitri, Agus Perdana Windarto, Harly Okprana (2021) [3].	Klasifikasi Calon Nasabah Baru Menggunakan C.45 Sebagai Dasar Pemberian Pertanggungan Asuransi Di PT Asuransi Central Asia Pematangsiantar.	Dari hasil performance vector diatas dapat diketahui bahwa penerapan Algoritma C4.5 dalam mengklasifikasikan calon nasabah baru dengan menggunakan software RapidMiner memperoleh tingkat akurasi sebesar 96,67%. klasifikasi layak memiliki nilai 26 dan klasifikasi tidak layak memiliki nilai 3 dengan nilai precision sebesar 91.38% dan nilai recall sebesar 75,00% [3].

NO	PENELITI	JUDUL	HASIL
2.	Sunjana (2010) [4].	Klasifikasi Data Nasabah Sebuah Asuransi Menggunakan Algoritma C4.5	Atribut-atribut yang digunakan dalam penelitian ini adalah penghasilan, premi_dasar, cara_pembayaran, mata_uang, dan status sedang label yang digunakan untuk pengklasifikasian adalah lancar dan tidak lancar Aplikasi dapat menyimpulkan bahwa rata-rata nasabah memiliki status L dikarenakan pembayaran premi yang melebihi 10% dari penghasilan. Dengan persentase atribut Premi Dasar dan Penghasilan, maka dapat diketahui rata-rata status nasabah memiliki nilai P atau L [4].

NO	PENELITI	JUDUL	HASIL
3.	Dian Adriansyah, Walim Walim (2018) [5].	Algoritma C4.5 Untuk Klasifikasi Calon Peserta Lomba Cerdas Cermat Siswa Smp Dengan Menggunakan Aplikasi Rapid Miner.	Berdasarkan hasil pengujian pada klasifikasi Calon peserta cerdas cermat dengan menggunakan aplikasi rapid miner terdapat kesimpulan di antaranya, klasifikasi proses seleksi calon peserta lomba siswa SMP dapat mengklasifikasikan siswa dalam tahapan lolos atau tidaknya dalam seleksi. Dari 33 data siswa yang digunakan menunjukkan tingkat akurasi dengan algoritma C4.5 sebesar 81,81% [5].

2.2 Tinjauan Pustaka

2.2.1 Pemberian Pertanggungan Asuransi

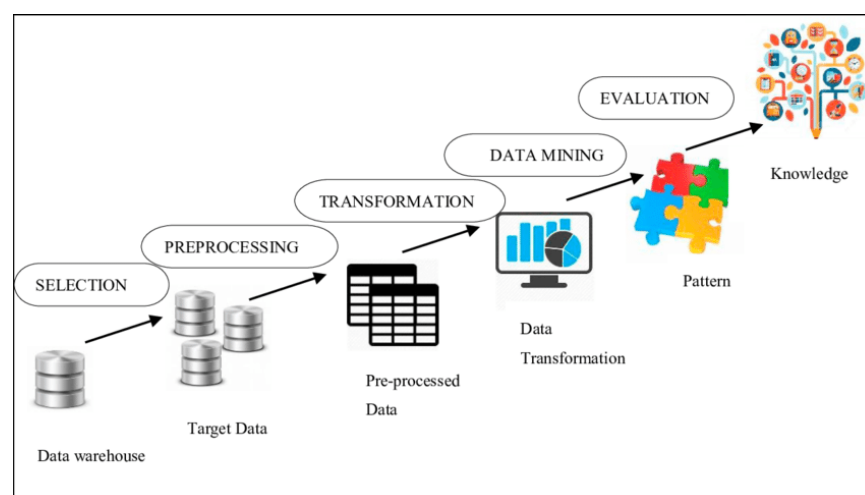
Pertanggungan dalam asuransi adalah penanggulangan risiko terhadap pemegang polis atau tertanggung dalam hal memberikan penggantian karena kerusakan, kerugian, meninggal dunia atau tetap hidup. Dalam hal ini asuransi memberikan jaminan penggantian terhadap tertanggung dengan syarat kontrak yang telah ditetapkan.

Pemberian pertanggungan asuransi melibatkan dua belah pihak atau lebih dengan dibuatnya perjanjian antara penanggung sebagai pihak yang memberikan jasa dalam penanggulangan risiko kepada tertanggung atau seseorang yang diasuransikan. Dengan dibuatnya polis asuransi yang akan ditandatangani oleh pihak tertanggung, maka pihak tertanggung telah melakukan persetujuan antar kedua belah pihak bahwa tertanggung menerima segala peraturan yang ditetapkan oleh pihak penanggung.

2.2.2 Data Mining

Data mining adalah proses pencarian pola atau informasi dari sebuah data terpilih yang menggunakan teknik dan metode tertentu. Teknik dan metode atau algoritma dalam *data mining* mempunyai banyak variasi [6]. Penentuan metode atau algoritma yang tepat tergantung pada tujuan dan proses *Knowledge Discovery in Database (KDD)* secara menyeluruh.

Proses *Knowledge Discovery in Database (KDD)* dapat digambarkan seperti pada gambar 2.1 :



Gambar 2.1 Proses Knowledge Discovery in Database (KDD)

1. *Data Selection*

Pemilihan (seleksi) data dari kumpulan data operasional yang perlu dilakukan sebelum tahap penggalian informasi dalam *Knowledge Discovery in Database (KDD)* dimulai [6]. Hasil dari seleksi data yang nantinya akan digunakan dalam proses *data mining*, akan disimpan pada suatu berkas secara terpisah dari basis data operasional.

2. *Pre – Processing / Cleaning*

Sebelum proses *data mining* dilakukan, perlu adanya proses cleaning pada data yang akan menjadi fokus dari *Knowledge Discovery in Database (KDD)* [6]. Proses *cleaning* mencakup berbagai proses yakni membuang duplikat pada data, memeriksa data yang inkonsisten, dan memperbaiki kesalahan yang terdapat pada data, seperti kesalahan dalam mencetak.

3. *Transformation*

Dilakukan proses transformasi dengan *Coding* pada data yang terpilih, sehingga data tersebut akan sesuai dengan proses *data mining*. Proses *Coding* pada *Knowledge Discovery in Database (KDD)* merupakan proses yang memiliki kreativitas dan sangat tergantung dengan jenis atau pola informasi [6]. yang akan dicari pada basis data.

4. *Data Mining*

Data mining mempunyai proses yang menggunakan teknik statistik, matematika, kecerdasan buatan, dan *machine learning* [6]. untuk ekstraksi dan identifikasi suatu informasi yang berguna juga memiliki pengetahuan yang terkait di berbagai database besar.

5. *Interpretation / Evaluation*

Pola dari informasi yang dihasilkan pada saat proses *data mining* perlu ditampilkan dalam yang mudah dipahami. Pada tahapan ini mencakup pemeriksaan apakah pola atau informasi yang ditemukan nantinya bertentangan dengan fakta atau hipotesis yang ada sebelumnya.

2.2.3 Klasifikasi

Secara Etimologi, kata Klasifikasi diangkat dari kata bahasa Inggris yaitu “*Classification*” dan kata ini berasal dari kata “*To Classy*” yang berarti menggolongkan atau menempatkan sesuatu pada tempatnya. Istilah ini ditunjukkan kepada sebuah metode untuk menyusun sebuah data secara sistematis [7].

Menurut *Harrolds Librarians Glossary*, Towa P. Hamakonda, dan J.N.B Tairas (2022) [7]. menyebutkan bahwa klasifikasi adalah pengelompokan yang sistematis dan logis dari obyek, gagasan, buku atau benda lain ke dalam kelas atau golongan tertentu sesuai dengan ciri-ciri yang sama.

Proses klasifikasi memiliki beberapa tahapan dalam melakukan *data mining* :

1. Validasi (*Validation*)

Validasi adalah sebuah proses pembagian data awal menjadi dua bagian data, *data training* dan *data testing*. *Data training* adalah data yang akan diolah menggunakan algoritma klasifikasi dan *data testing* adalah data yang akan digunakan dalam proses pengujian menggunakan program komputer. Proses validasi dalam klasifikasi terbagi menjadi dua proses yaitu *cross validation* dan *split validation*. *Cross validation* melakukan pembagian data secara acak sedangkan *split validation* melakukan pembagian data menjadi dua bagian berdasarkan jumlah data.

2. Implementasi Algoritma Klasifikasi

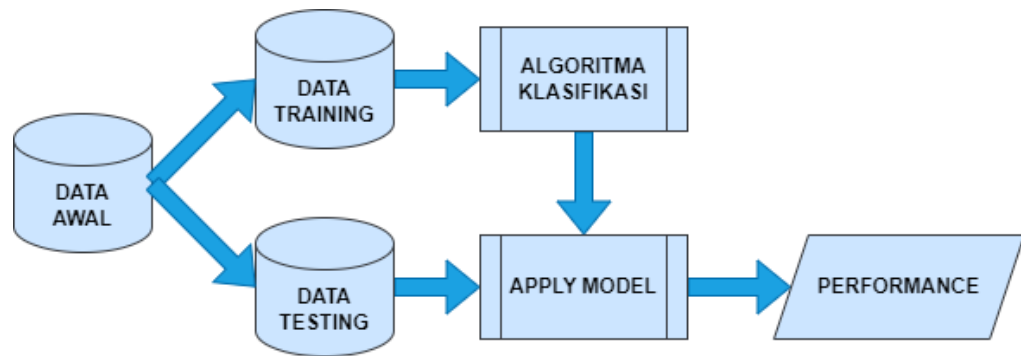
Proses klasifikasi melakukan pengolahan data dengan mengimplementasikan algoritma klasifikasi seperti *support vector machine*, *decision tree*, *k-means* dan *naive bayes*.

3. Apply Model

Apply model digunakan untuk penerapan model klasifikasi yang telah dilatih sebelumnya menggunakan *data training* yang juga akan digunakan dalam pengujian *data testing*.

4. Performance

Performance adalah hasil dari pengujian yang dilakukan dalam proses klasifikasi. Hasil yang diukur pada tahap *performance* seperti akurasi, *precision*, dan *recall*.



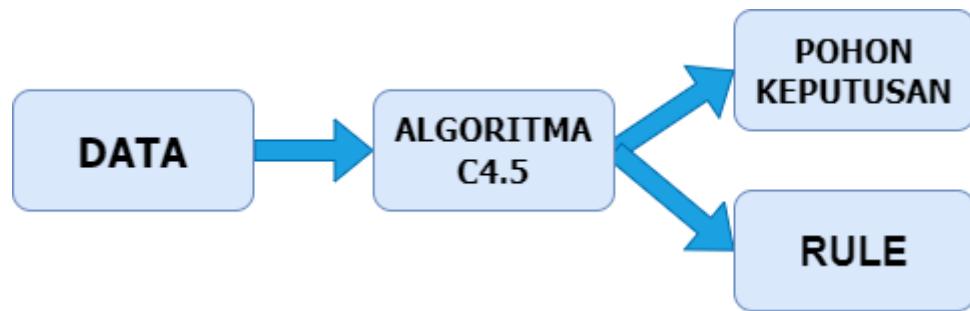
Gambar 2.2 Tahapan Proses Klasifikasi

2.2.4 Split Validation

Split Validation adalah teknik validasi dengan membagi data menjadi dua bagian secara acak, bagian pertama untuk *data training* yang digunakan mengevaluasi atau pengujian pada data dan bagian kedua *data testing* digunakan untuk melatih model.

2.2.5 Algoritma C4.5

Algoritma C4.5 adalah algoritma yang dapat digunakan untuk membuat pohon keputusan *decison tree* dan merupakan algoritma yang paling banyak digunakan karena memiliki kelebihan utama dari algoritma lainnya [3]. Algoritma ini juga akan menghasilkan rule yang nantinya akan mempermudah dalam proses menggali sebuah data dan informasi.



Gambar 2.3 Alur Proses Algoritma C4.5

Terdapat beberapa tahapan dalam pembuatan pohon keputusan *decision tree* algoritma C4.5 yaitu :

1. Menyiapkan *data training*. *Data training* dapat diambil dari data awal yang sudah ada sebelumnya dan sudah dikelompokkan berdasarkan kelas tertentu. *Data training* adalah data yang akan diolah menggunakan algoritma dalam pemodelan pohon keputusan dan rule.
2. Menghitung akar pohon. Akar nantinya akan diambil pada atribut data yang dipilih dengan cara menghitung nilai *entropy* dan *gain* dari setiap atribut, nilai *gain* tertinggi akan menjadi akar di bagian pertama. Sebelum mencari nilai *gain* maka perlu diketahui dahulu nilai *entropy* dari setiap atribut.

Perhitungan nilai *entropy* digunakan rumus :

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

Keterangan :

S : Himpunan Kasus

i : Jumlah Partisi *S*

pi : Proporsi Dari *Si* terhadap *S*

Jika nilai *entropy* pada masing-masing atribut sudah diketahui, kemudian mencari nilai *gain* dengan rumus :

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

Keterangan :

S : Himpunan Kasus

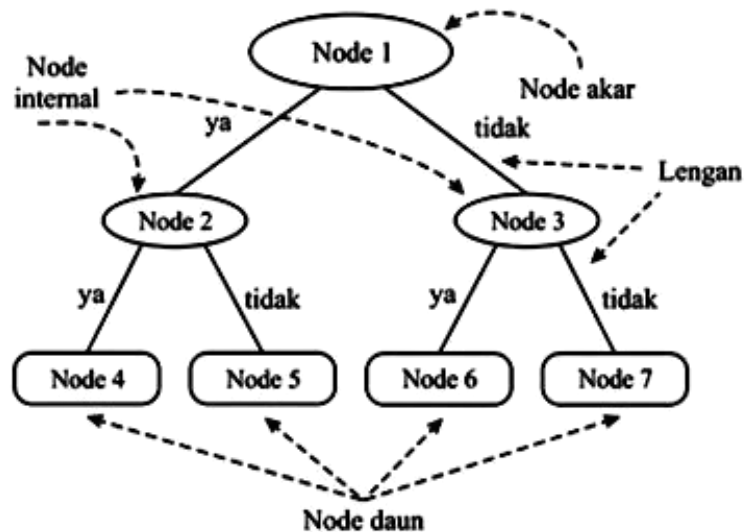
A : Fitur

N : Jumlah Partisi Atribut A

$|S_i|$: Jumlah Kasus Pada Partisi Ke- i

$|S|$: Jumlah Kasus Dalam S

Setelah proses perhitungan nilai *entropy* dan *gain* didapatkan, akan menghasilkan model pohon keputusan dan rule yang terbentuk dari proses penggalan data.



Gambar 2.4 Contoh Pohon Keputusan Dan Rule

2.2.6 Penerapan Algoritma Decision Tree C4.5

Terdapat penelitian terkait dengan penerapan algoritma *decision tree C4.5* yang dikutip dari penelitian sebelumnya oleh Retno Ayu Syahfitri, Agus Perdana Windarto, dan Harly Okprana pada tahun 2021 dengan judul “Klasifikasi Calon Nasabah Baru Sebagai Dasar Pemberian Pertanggungan Asuransi Menggunakan Algoritma *Decision Tree C4.5* Di PT. Asuransi Central Asia Pematangsiantar” dengan beberapa tahapan :

1. Tahap *Selection and Transformation*

Berdasarkan data yang akan diolah dilakukan proses seleksi dan transformasi data sehingga ada kriteria yang akan digunakan sebanyak 3 seperti yang ditunjukkan pada tabel berikut :

Tabel 2.2 Dataset Calon Nasabah Asuransi

No.	Nama	Usia	Penghasilan	Jumlah Tanggungan (Orang)	Keputusan
1	Situlus Sigalingging	30	3500000	4	Layak
2	Marnaek Marojahan Sinaga	33	4000000	6	Tidak Layak
3	Pintor Sitorus	45	4450000	4	Layak
4	Erni R Marpaung	27	3000000	1	Layak
5	Pancasila Sibarani	29	4800000	6	Tidak Layak
6	Manro Nababan	34	2950000	4	Tidak Layak
...
100	Ratna Masni Sagala	39	4500000	0	Layak

Tabel 2.3 Kriteria Usia

Umur	Kriteria
24 - 40	Baik
40-50	Cukup
50-70	Kurang

Tabel 2.4 Kriteria Penghasilan

Penghasilan	Kriteria
Rp. 2.500.000 – Rp. 3.000.000	Cukup
Rp. 3.000.000 – Rp. 6.000.000	Baik
> Rp. 6.000.000	Sangat Baik

Tabel 2.5 Kriteria Jumlah Tanggungan

Jumlah Tanggungan	Kriteria
0-1	Sangat Baik
2-4	Baik
>6	<u>Cukup</u>

2. Tahap *Validasi*

Pada tahapan ini data akan dibagi menjadi dua bagian yaitu *data training* dan *data testing* pada tabel berikut :

Tabel 2.6 Data Training

No.	Nama	Usia	Penghasilan	Jumlah Tanggungan (Orang)
1	A1	Baik	Baik	Baik
2	A2	Baik	Baik	Cukup
3	A3	Cukup Baik	Baik	Baik
4	A4	Baik	Baik	Sangat Baik
5	A5	Baik	Baik	Cukup
6	A6	Baik	Cukup	Baik
...
50	A50	Baik	Baik	Sangat Baik

Tabel 2.7 Data Testing

No.	Nama	Usia	Penghasilan	Jumlah Tanggungan (Orang)
51	A51	Cukup Baik	Baik	Sangat Baik
52	A52	Baik	Baik	Baik
53	A53	Cukup Baik	Baik	Baik
54	A54	Baik	Cukup	Baik
55	A55	Baik	Baik	Sangat Baik
56	A56	Kurang	Baik	Baik
...
100	A100	Baik	Baik	Sangat Baik

3. Tahap Implementasi Algoritma C4.5

Tahap ini terdiri dari beberapa proses yakni :

- Menyiapkan *data training* yang ditunjukkan pada tabel 2.6
- Melakukan perhitungan nilai *entropy* dan *gain*
- Melakukan klasifikasi menggunakan algoritma C4.5

Dengan melakukan perhitungan jumlah kasus, jumlah kasus untuk kategori layak dan tidak layak kemudian *entropy* dari semua kasus berdasarkan atribut yang digunakan dengan rumus mencari *entropy* pada persamaan 2 :

$$\begin{aligned}
 Entropy\ S &= \left(- \left(\frac{17}{100} \right) \log_2 \left(\frac{17}{100} \right) \right) + \left(- \left(\frac{83}{100} \right) * \log_2 \left(\frac{83}{100} \right) \right) \\
 &= 0,43458686924 + 0,22311790949 \\
 &= 0,6577047787
 \end{aligned}$$

Tabel 2.8 Hasil Perhitungan *Entropy* Total Keseluruhan

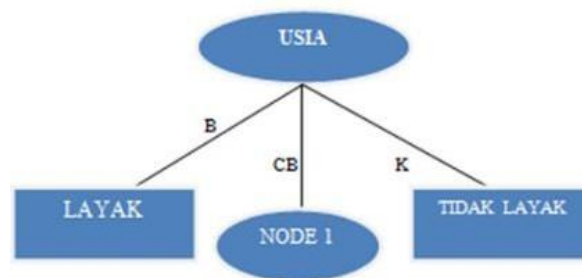
Total Kasus	Sum (Layak)	Sum (Tidak Layak)	Entropy Total
100	83	17	0,6577047787

Setelah melakukan perhitungan nilai *entropy* dan *gain* untuk semua atribut dilakukan untuk mencari nilai *gain* tertinggi yang nantinya akan dijadikan sebagai akar. Hasil dari perhitungannya dapat dilihat pada tabel 2.9 berikut :

Tabel 2.9 Perhitungan Node 1

Atribut		Jumlah	Layak	Tidak Layak	Entropy	Gain
Total		100	83	17	0.65770478	
						0.175626
Usia	Baik	67	63	4	0.32625881	
	Cukup Baik	24	18	6	0.81127812	
	Kurang	9	2	7	0.76420451	
						-0.04704
Penghasilan	Cukup	8	4	4	1	
	Baik	81	68	13	0.63548964	
	Sangat Baik	11	11	0	0	
						0.067637
Jumlah Tanggungan	Sangat Baik	39	37	2	0.29181826	
	Baik	55	43	12	0.75683363	
	Cukup	6	3	3	1	

Dari tabel 2.9 diatas ditunjukkan bahwa atribut yang memperoleh nilai gain tertinggi adalah atribut usia dengan nilai 0.175626 [3]. Atribut usia yang nantinya akan menjadi node akar dalam pemodelan pohon keputusan dengan memiliki tiga nilai yaitu baik, cukup baik, dan kurang. Atribut baik pada usia akan mengklasifikasikan kasus layak sedangkan atribut kurang akan mengklasifikasikan kasus tidak layak. Dari hasil perhitungan node 1 yang diperoleh dapat digambarkan model dari pohon keputusan seperti pada gambar 2.5 :



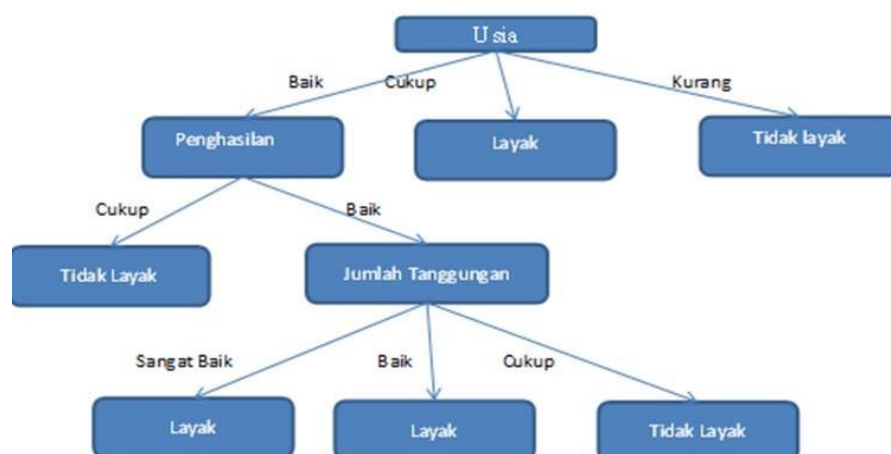
Gambar 2.5 Pohon Keputusan Hasil Perhitungan Node 1

Tabel 2.10 Perhitungan Node 1.1

Atribut		Jumlah	Layak	Tidak Layak	Entropy	Gain
Total		100	83	17	0.657704779	0.220659
Penghasilan	Cukup	15	13	15	0.178924094	
	Baik	27	26	9	0.580752033	
	Sangat Baik	9	5	1	0.823323282	
Jumlah Tanggungan						0.444965
	Sangat Baik	13	10	13	0.291162787	
	Baik	18	15	11	0.653385739	
	Cukup	20	19	1	0.286396957	

Pada tabel 2.10 diatas atribut yang memiliki nilai gain tertinggi adalah atribut jumlah tanggungan dengan nilai 0.444965. dengan demikian atribut jumlah tanggungan menjadi cabang keputusan terakhir pada kasus ini [3]. Terdapat 3 nilai pada atribut jumlah tanggungan yaitu sangat baik, baik, dan cukup. Sangat baik pada atribut jumlah tanggungan mengklasifikasikan kasus layak sedangkan atribut baik mengklasifikasikan kasus layak dan atribut cukup mengklasifikasikan kasus tidak layak.

Dari hasil perhitungan node 1.1 yang ditunjukkan pada tabel 2.10 dapat digambarkan model pohon keputusan seperti pada gambar 2.6 berikut :



Gambar 2.6 Pohon Keputusan Hasil Perhitungan Node 1.1

2.2.7 Python

Python adalah sebuah bahasa pemrograman yang bertujuan umum dan memiliki level tingkat tinggi. Pertama kali dirilis pada tahun 1991 oleh Guido van Rossum selaku pembuat, *python* digunakan untuk pembuatan aplikasi, sebagai alat untuk memerintah suatu komputer, dan melakukan analisis data. *Python* sebagai general-purpose language dapat digunakan untuk membuat sebuah program apa saja dan dapat menyelesaikan berbagai permasalahan yang terjadi [8].

Python juga dikategorikan sebagai bahasa pemrograman multi-paradigma. Pemrograman berorientasi objek dan pemrograman terstruktur juga didukung penuh, dan memiliki banyak fitur yang mendukung pemrograman fungsional.

2.2.8 Confusion Matrix

Confusion matrix adalah sebuah *tool* untuk mengevaluasi model dari klasifikasi untuk memprediksi objek yang bernilai benar atau salah [5]. *Matrix* dari hasil prediksi yang nantinya akan dibandingkan dengan kelas asli inputan atau informasi nilai aktual dan prediksi pada klasifikasi.

Tabel 2.11 Tabel *Confusion Matrix* 2 Kelas

<i>Classification</i>	<i>Predicted Class</i>	
	<i>Class = Yes</i>	<i>Class = No</i>
<i>Class = Yes</i>	<i>a (true positive-TP)</i>	<i>b (false negative-FN)</i>
<i>Class = No</i>	<i>c (false positive-FP)</i>	<i>d (true negative-TN)</i>

Rumus perhitungan akurasi, presisi, dan recall pada matriks :

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+TN+FN}$$

$$\text{Precision} = \frac{TP}{FP+TP}$$

$$\text{Recall} = \frac{TP}{FN+TP}$$

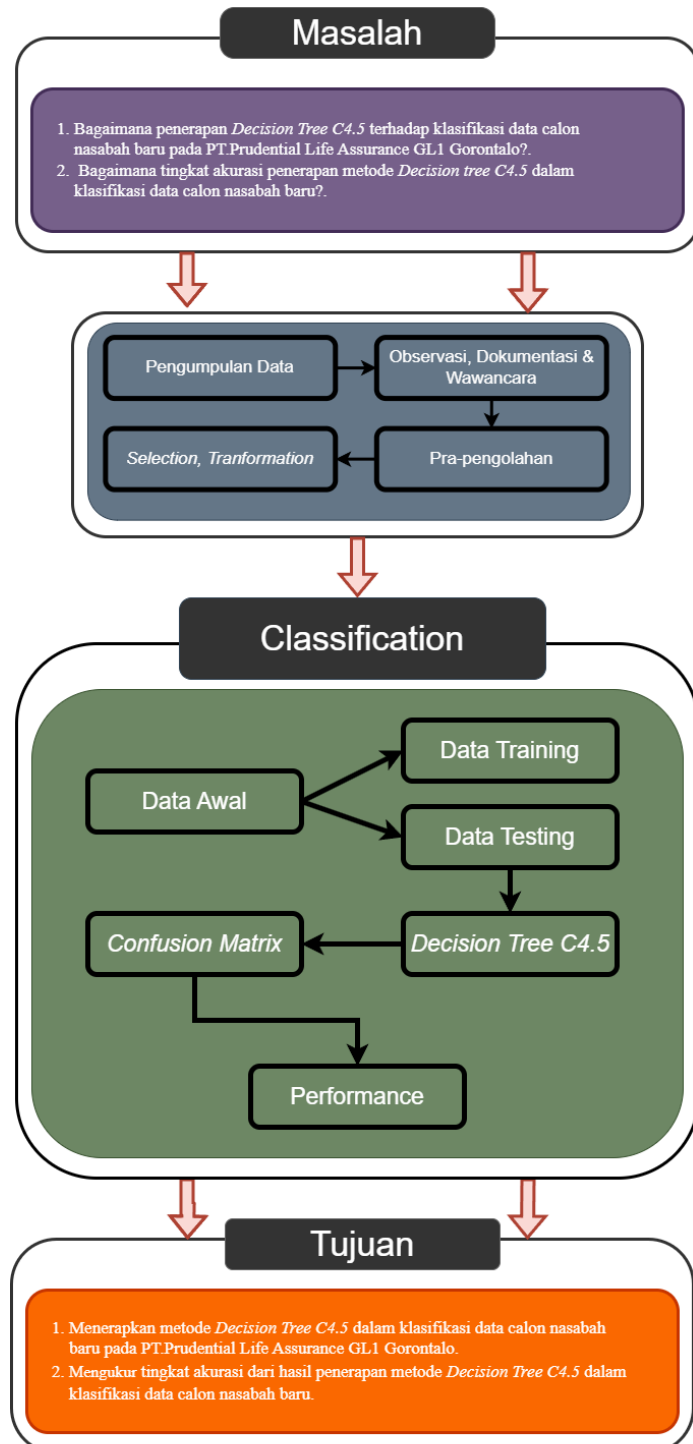
2.2.9 Perangkat Lunak

Perangkat lunak pendukung yang akan digunakan pada proses pengujian untuk klasifikasi data :

Tabel 2.12 Perangkat Lunak Pendukung

No	Perangkat Lunak	Keterangan
1.	<i>python</i>	<i>Python</i> adalah bahasa pemrograman yang paling sering digunakan dalam ilmu data, pengembangan perangkat lunak, aplikasi web, dan machine learning [9]. Setiap developer memilih menggunakan <i>python</i> karena lebih efisien dan dapat dengan mudah dipahami serta bisa dijalankan di berbagai platform.
2.	<i>Google Colaboratory</i>	<i>Google Colaboratory</i> , juga dikenal sebagai <i>Google Colab</i> , adalah platform komputasi cloud gratis yang disediakan oleh Google. Ini memungkinkan pengguna untuk menjalankan kode Python secara interaktif dalam lingkungan berbasis web tanpa perlu menginstal perangkat lunak atau mengatur infrastruktur komputasi sendiri.

2.2.10 Kerangka Pikir



Gambar 2.7 Bagan Kerangka Pikir

BAB III

METODE PENELITIAN

3.1 Jenis, Metode, Subjek, Objek, Waktu, Dan Lokasi Penelitian

Dari segi pandangan tingkat penerapannya, maka penelitian ini merupakan penelitian terapan. Dari segi jenis informasi yang diolah, penelitian ini merupakan penelitian kuantitatif. Dari segi perlakuan terhadap data yang diolah, maka penelitian ini merupakan penelitian konfirmatori.

Subjek dari penelitian ini adalah klasifikasi pada objek calon nasabah baru sebagai dasar pemberian pertanggungan. Penelitian ini dimulai pada bulan November tahun 2022 sampai dengan bulan Desember tahun 2023 yang berlokasi di PT. Prudential Life Assurance GL1 Gorontalo.

3.2 Pengumpulan Data

Penelitian ini menggunakan pengumpulan data dengan jenis data primer yang dikumpulkan dengan teknik dokumentasi, wawancara, dan observasi yang didapatkan langsung di lapangan. Sedangkan untuk data sekunder diperoleh dari penelitian sebelumnya seperti jurnal yang membahas tentang *klasifikasi* dengan menerapkan algoritma *decision tree C4.5* baik dari internet maupun perpustakaan.

Adapun sampel data dan variabel/atribut dengan tipe datanya masing-masing ditunjukkan pada tabel 3.1 dan tabel 3.2 berikut ini.

Tabel 3.1 Data Calon Nasabah Baru

No.	Nama	Jenis Kelamin	Alamat	Jenis Pekerjaan	Usia	Penghasilan	Format Polis	Riwayat Kesehatan	Ket
0	nasabah 1	L	GORONTALO	Wiraswasta	52	7.000.000 sd 10.000.000	Polis Elektronik	Tidak Ada	Layak
1	nasabah 2	P	PAGUYAMAN	Swasta	34	10.000.000 sd 25.000.000	Polis Elektronik	Tidak Ada	Layak
2	nasabah 3	P	GORONTALO	Swasta	50	10.000.000 sd 25.000.000	Polis Elektronik	Tidak Ada	Layak
3	nasabah 4	L	KABILA	PNS	31	7.000.000 sd 10.000.000	Polis Non Elektronik	Tidak Ada	Layak
4	nasabah 5	L	SUAWA	Wiraswasta	31	5.000.000 sd 7.000.000	Polis Non Elektronik	Penyakit Kritis	Tidak Layak
5	nasabah 6	L	GORONTALO	Wiraswasta	41	10.000.000 sd 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
6	nasabah 7	L	GORONTALO	Swasta	34	10.000.000 sd 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
7	nasabah 8	P	GORONTALO	Wiraswasta	27	5.000.000 sd 7.000.000	Polis Elektronik	Penyakit Kritis	Tidak Layak
8	nasabah 9	L	LIMBOTO	Wiraswasta	30	5.000.000 sd 7.000.000	Polis Non Elektronik	Penyakit Kritis	Tidak Layak
9	nasabah 10	L	KABILA	PNS	36	10.000.000 sd 25.000.000	Polis Elektronik	Tidak Ada	Layak
10	nasabah 11	L	GORONTALO	Wiraswasta	38	7.000.000 sd 10.000.000	Polis Non Elektronik	Tidak Ada	Layak
...
519	nasabah 520	L	GORONTALO	Wiraswasta	49	5.000.000 sd 7.000.000	Polis Elektronik	Penyakit Kritis	Tidak Layak

Tabel 3.2 Variabel/Atribut Data

NO.	NAME	TYPE	VALUE	KETERANGAN
1	Usia	Integer	Diketahui	Variabel Input
2	Jenis Kelamin	Tuple	Diketahui	Variabel Input
3	Jenis Pekerjaan	Varchar	Diketahui	Variabel Input
4	Riwayat Kesehatan	Varchar	Diketahui	Variabel Input
5	Penghasilan	Varchar	Diketahui	Variabel Input
6	Class Label	Varchar	Layak, tidak layak	Variabel Output

3.3 Pemodelan

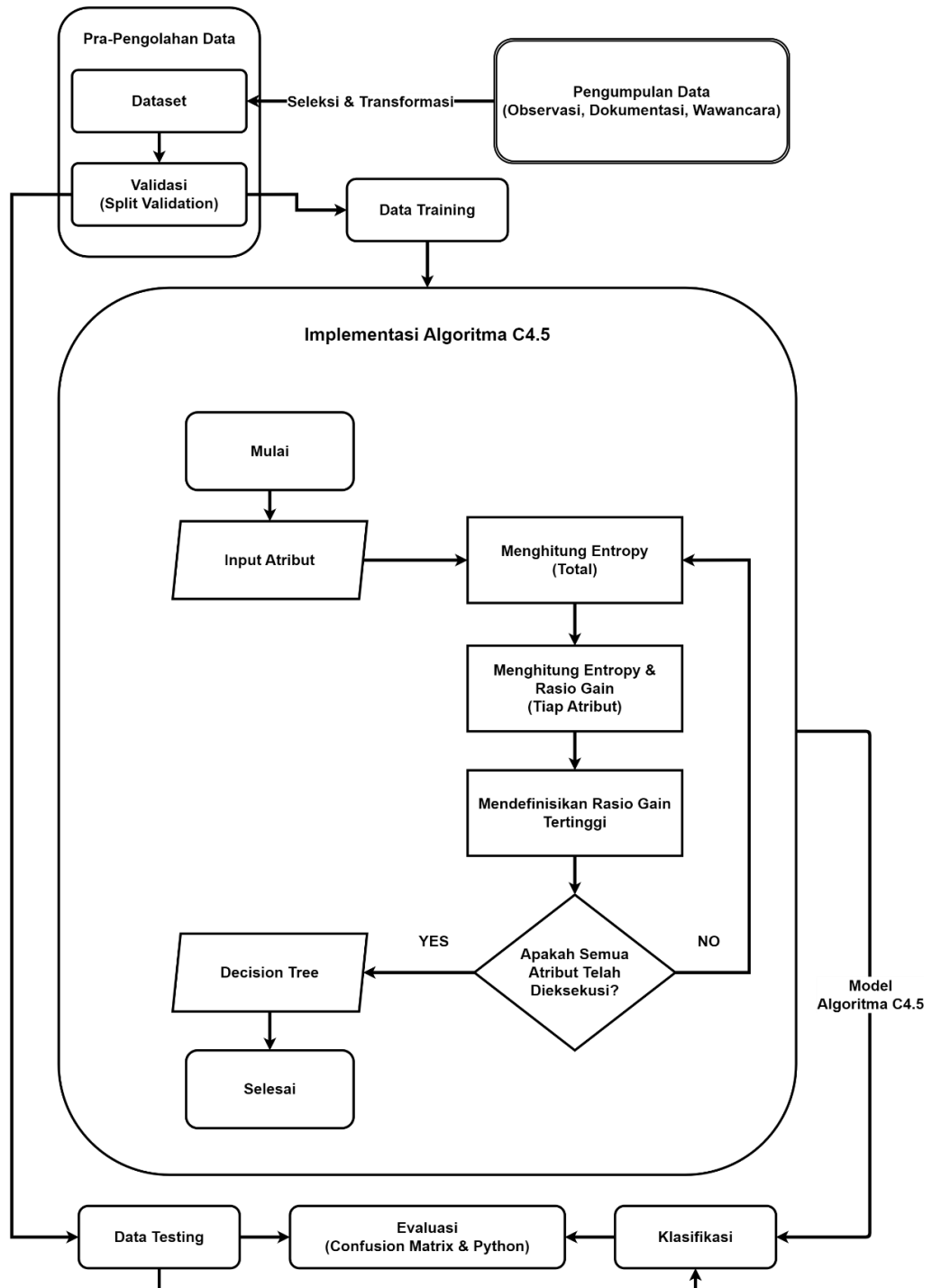
3.3.1 Pra Pengolahan Data

Data yang akan diolah nanti terlebih dahulu dilakukan *selection* dan *transformation* data. Hal ini dilakukan untuk menentukan kriteria dari atribut yang nantinya akan menjadi nilai ukur dari tiap atribut untuk proses klasifikasi data.

3.3.2 Validasi Data

Validasi dilakukan untuk membagi data awal menjadi dua bagian data, *data training* dan data *testing*. *Data training* adalah data yang akan diolah menggunakan algoritma klasifikasi dan *data testing* adalah data yang akan digunakan dalam proses pengujian menggunakan program komputer. Teknik validasi yang digunakan pada tahap ini yaitu *split validation* dengan melakukan pembagian data menjadi dua bagian berdasarkan jumlah data.

3.3.3 Pengembangan Model



Gambar 3.1 Pengembangan Model Klasifikasi Data Calon Nasabah Baru

3.4 Evaluasi Model

Model yang dihasilkan kemudian dievaluasi dengan menggunakan *Python*, dan *Confusion Matrix* Untuk mengetahui tingkat *akurasi*, *presisi*, dan *recall*.

BAB IV

HASIL PENELITIAN

4.1 Hasil Pengumpulan Data

Pada tahapan ini, data yang digunakan didapatkan dari hasil pengumpulan data dengan melakukan observasi langsung ke lokasi penelitian yakni PT.Prudential Life Assurance GL1 Gorontalo dan melakukan studi dokumen untuk mendapatkan data tersebut. Dengan observasi peneliti mendapatkan data calon nasabah baru yang terdapat pada tabel 4.1 Hasil Pengumpulan Data Calon Nasabah Baru.

Tabel 4.1 Hasil Pengumpulan Data Calon Nasabah Baru

No.	Nama	Jenis Kelamin	Alamat	Jenis Pekerjaan	Usia
0	nasabah 1	L	GORONTALO	Wiraswasta	52
1	nasabah 2	P	PAGUYAMAN	Swasta	34
2	nasabah 3	P	GORONTALO	Swasta	50
3	nasabah 4	L	KABILA	PNS	31
4	nasabah 5	L	SUWAWA	Wiraswasta	31
5	nasabah 6	L	GORONTALO	Wiraswasta	41
6	nasabah 7	L	GORONTALO	Swasta	34
7	nasabah 8	P	GORONTALO	Wiraswasta	27
8	nasabah 9	L	LIMBOTO	Wiraswasta	30
9	nasabah 10	L	KABILA	PNS	36
10	nasabah 11	L	GORONTALO	Wiraswasta	38
11	nasabah 12	L	GORONTALO	PNS	45
12	nasabah 13	L	GORONTALO	Swasta	32
13	nasabah 14	L	BONE PANTAI	PNS	36
14	nasabah 15	L	PALOPO	Wiraswasta	38
15	nasabah 16	P	BULULI	Swasta	39
16	nasabah 17	L	GORONTALO	Swasta	44
17	nasabah 18	P	MINAHASA	Swasta	48
18	nasabah 19	L	GORONTALO	Swasta	52
19	nasabah 20	P	GORONTALO	Wiraswasta	32
20	nasabah 21	L	GORONTALO	Wiraswasta	31
21	nasabah 22	L	PAGUYAMAN	Nelayan	27
...
519	nasabah 520	P	GORONTALO	Wiraswasta	49

No.	Penghasilan	Format Polis	Riwayat Kesehatan	Keterangan
0	7.000.000 s.d. 10.000.000	Polis Elektronik	Tidak Ada	Layak
1	10.000.000 s.d. 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
2	10.000.000 s.d. 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
3	7.000.000 s.d. 10.000.000	Polis Non Elektronik	Tidak Ada	Layak
4	5.000.000 s.d. 7.000.000	Polis Non Elektronik	Penyakit Kritis	Tidak Layak
5	10.000.000 s.d. 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
6	10.000.000 s.d. 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
7	5.000.000 s.d. 7.000.000	Polis Elektronik	Penyakit Kritis	Tidak Layak
8	5.000.000 s.d. 7.000.000	Polis Non Elektronik	Penyakit Kritis	Tidak Layak
9	10.000.000 s.d. 25.000.000	Polis Elektronik	Tidak Ada	Layak
10	7.000.000 s.d. 10.000.000	Polis Non Elektronik	Tidak Ada	Layak
11	5.000.000 s.d. 7.000.000	Polis Elektronik	Penyakit Kritis	Tidak Layak
12	5.000.000 s.d. 7.000.000	Polis Non Elektronik	Penyakit Kritis	Tidak Layak
13	25.000.000 s.d. 50.000.000	Polis Non Elektronik	Tidak Ada	Layak
14	25.000.000 s.d. 50.000.000	Polis Non Elektronik	Tidak Ada	Layak
15	7.000.000 s.d. 10.000.000	Polis Elektronik	Tidak Ada	Layak
16	25.000.000 s.d. 50.000.000	Polis Elektronik	Penyakit Kritis	Tidak Layak
17	7.000.000 s.d. 10.000.000	Polis Elektronik	Tidak Ada	Layak
18	10.000.000 s.d. 25.000.000	Polis Non Elektronik	Tidak Ada	Layak
19	7.000.000 s.d. 10.000.000	Polis Elektronik	Tidak Ada	Layak
20	7.000.000 s.d. 10.000.000	Polis Elektronik	Tidak Ada	Layak
21	7.000.000 s.d. 10.000.000	Polis Elektronik	Tidak Ada	Layak
...
519	5.000.000 s.d. 7.000.000	Polis Non Elektronik	Penyakit Kritis	Tidak Layak

4.2 Hasil Pemodelan

Berdasarkan tabel 4.1 di atas adalah data yang didapatkan dari hasil pengumpulan data yang masih harus dilakukan proses pra-pengolahan data awal. pada penerapan *Decision Tree C4.5* nantinya dilakukan beberapa proses pra-pengolahan data sebelum diterapkan pada model *Decision Tree C4.5* diantaranya dilakukan tahapan pengolahan sebagai berikut :

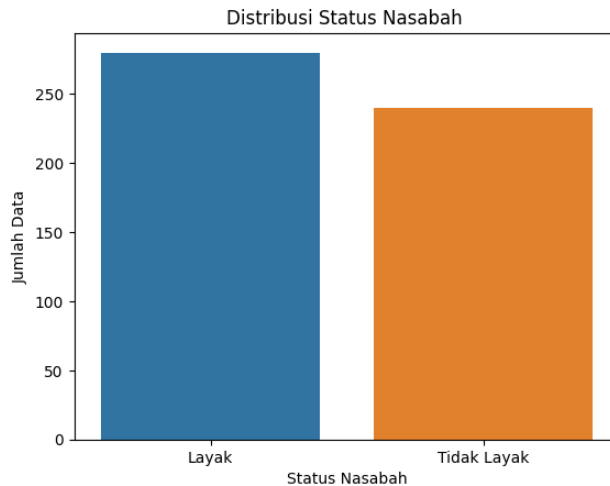
1. Seleksi Data

Pada tahap ini dilakukan seleksi atribut yang digunakan berdasarkan pengumpulan data pada tabel 4.1 sebelumnya. Atribut yang akan digunakan dari hasil seleksi yaitu : Jenis Kelamin, Jenis Pekerjaan, Usia, Penghasilan, dan Riwayat Kesehatan sehingga dataset yang digunakan pada proses pemodelan ditunjukkan pada tabel 4.2 berikut :

Tabel 4.2 Dataset Pemodelan

No.	Nama	JK	Jenis Pekerjaan	Usia	Penghasilan	Riwayat Kesehatan	Ket
0	nasabah 1	L	Wiraswasta	52	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
1	nasabah 2	P	Swasta	34	10.000.000 s.d. 25.000.000	Tidak Ada	Layak
2	nasabah 3	P	Swasta	50	10.000.000 s.d. 25.000.000	Tidak Ada	Layak
3	nasabah 4	L	PNS	31	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
4	nasabah 5	L	Wiraswasta	31	5.000.000 s.d. 7.000.000	Penyakit Kritis	Tidak Layak
5	nasabah 6	L	Wiraswasta	41	10.000.000 s.d. 25.000.000	Tidak Ada	Layak
6	nasabah 7	L	Swasta	34	10.000.000 s.d. 25.000.000	Tidak Ada	Layak
7	nasabah 8	P	Wiraswasta	27	5.000.000 s.d. 7.000.000	Penyakit Kritis	Tidak Layak
8	nasabah 9	L	Wiraswasta	30	5.000.000 s.d. 7.000.000	Penyakit Kritis	Tidak Layak
9	nasabah 10	L	PNS	36	10.000.000 s.d. 25.000.000	Tidak Ada	Layak
10	nasabah 11	L	Wiraswasta	38	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
11	nasabah 12	L	PNS	45	5.000.000 s.d. 7.000.000	Penyakit Kritis	Tidak Layak
12	nasabah 13	L	Swasta	32	5.000.000 s.d. 7.000.000	Penyakit Kritis	Tidak Layak
13	nasabah 14	L	PNS	36	25.000.000 s.d. 50.000.000	Tidak Ada	Layak
14	nasabah 15	L	Wiraswasta	38	25.000.000 s.d. 50.000.000	Tidak Ada	Layak
15	nasabah 16	P	Swasta	39	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
16	nasabah 17	L	Swasta	44	25.000.000 s.d. 50.000.000	Penyakit Kritis	Tidak Layak
17	nasabah 18	P	Swasta	48	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
18	nasabah 19	L	Swasta	52	10.000.000 s.d. 25.000.000	Tidak Ada	Layak
19	nasabah 20	P	Wiraswasta	32	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
20	nasabah 21	L	Wiraswasta	31	7.000.000 s.d. 10.000.000	Tidak Ada	Layak
...	
519	nasabah 520	L	Wiraswasta	49	5.000.000 s.d. 7.000.000	Penyakit Kritis	Tidak Layak

Berdasarkan tabel 4.2 di atas dilakukan perhitungan jumlah data untuk masing-masing kelas hasilnya ditampilkan pada grafik jumlah kelas dengan jumlah kelas layak 280 orang dan tidak layak berjumlah 240 orang yang dapat dilihat pada gambar 4.1 berikut :



Gambar 4.1 Grafik Jumlah Kelas

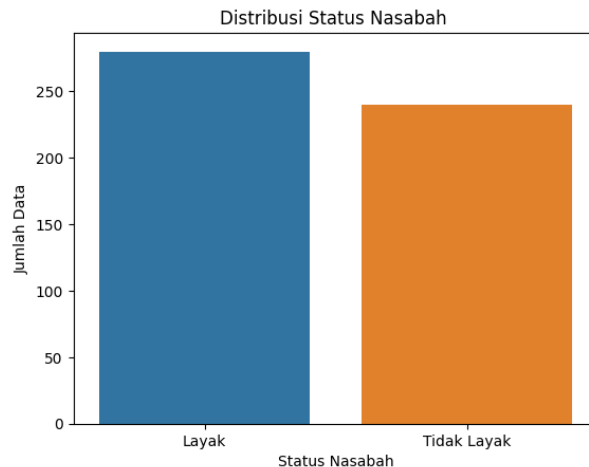
2. Preprocessing

Pada tahapan ini dilakukan pengecekan data yang kosong atau data yang tidak konsisten terhadap hasil akhir atau kelas output. Berikut contoh sampel data yang tidak konsisten dapat dilihat pada Tabel 4.3 berikut :

Tabel 4.3 Data Tidak Konsisten

No.	JK	Jenis Pekerjaan	Usia	Penghasilan	Riwayat Kesehatan	Kelas
1	L	Swasta	35	7.000.000 s.d. 10.000.000	Tidak Ada	Layak , Tidak Layak
2	L	Swasta	38	7.000.000 s.d. 10.000.000	Tidak Ada	Layak , Tidak Layak
3	P	Wiraswasta	36	7.000.000 s.d. 10.000.000	Tidak Ada	Layak , Tidak Layak
4	P	Wiraswasta	42	7.000.000 s.d. 10.000.000	Tidak Ada	Tidak layak , Layak

Setelah dilakukan perbaikan data kemudian dihitung kembali jumlah data masing-masing kelas dapat dilihat pada Gambar 4.2 berikut, jumlah data dari kelas status nasabah tidak memiliki perubahan dari segi jumlah, yaitu kelas layak berjumlah 280 dan tidak layak berjumlah 240 :



Gambar 4.2 Grafik Jumlah Kelas Setelah Perbaikan

3. Transformasi

Pada tahapan ini dilakukan transformasi data beberapa atribut dengan kriteria nilai yaitu: Jenis Kelamin diubah ke bentuk nilai 1 dan 2, jenis pekerjaan diubah dalam bentuk nilai 1-4, penghasilan diubah dalam bentuk nilai 1-4, dan Riwayat Kesehatan diubah dalam bentuk nilai 1-2 agar mempermudah proses pemodelan dengan metode *Decision Tree C4.5* Sampel hasil transformasi dapat dilihat pada tabel 4.4 sampai 4.8 berikut [9]. :

Tabel 4.4 Kriteria Jenis Kelamin

Jenis Kelamin	Nilai
L	1
P	2

Tabel 4.5 Kriteria Jenis Pekerjaan

Jenis Pekerjaan	Nilai
Nelayan	1
Swasta	2
PNS	3
Wiraswasta	4

Tabel 4.6 Kriteria Penghasilan

Penghasilan	Nilai
5.000.000 s.d. 7.000.000	1
7.000.000 s.d. 10.000.000	2
10.000.000 s.d. 25.000.000	3
25.000.000 s.d. 50.000.000	4

Tabel 4.7 Kriteria Riwayat Kesehatan

Riwayat Kesehatan	Nilai
Penyakit Kritis	1
Tidak Ada	2

Tabel 4.8 Dataset Hasil Transformasi

No	Nama	JK	Jenis Pekerjaan	Usia	Penghasilan	Riwayat Kesehatan	Ket
1	Nasabah 1	1	4	52	2	2	layak
2	Nasabah 2	2	2	34	3	2	layak
3	Nasabah 3	2	2	50	3	2	layak
4	Nasabah 4	1	3	31	2	2	layak
5	Nasabah 5	1	4	31	1	1	Tidak layak

4. Proses Data Mining

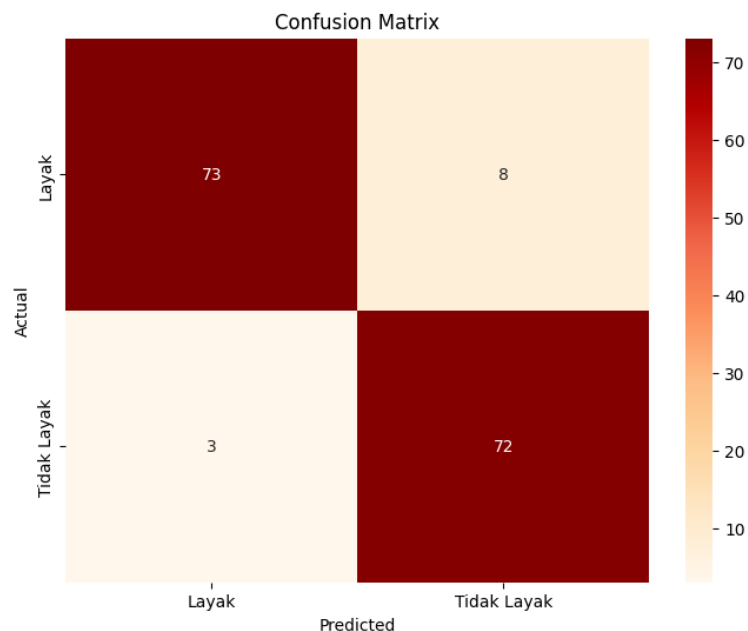
Setelah dilakukan tahapan di atas, maka selanjutnya dilakukan proses data mining terdiri dari penentuan data atribut (Jenis Pekerjaan, Usia, Penghasilan, Riwayat Kesehatan) dan kelas (Status nasabah), pembagian data training dan data testing dengan perbandingan 70:30 sehingga didapat Data Training sebanyak 364 data dan Data Testing sebanyak 156 data dari dataset sebanyak 520. Setelah itu dilakukan proses pemodelan dengan metode decision tree C4.5

5. Evaluasi

Setelah dilakukan Proses data mining perlu dilakukan evaluasi terhadap model yang sudah dibuat dengan melihat hasil akurasi dan visualisasi hasil pemodelan.

a. Evaluasi Kinerja Model

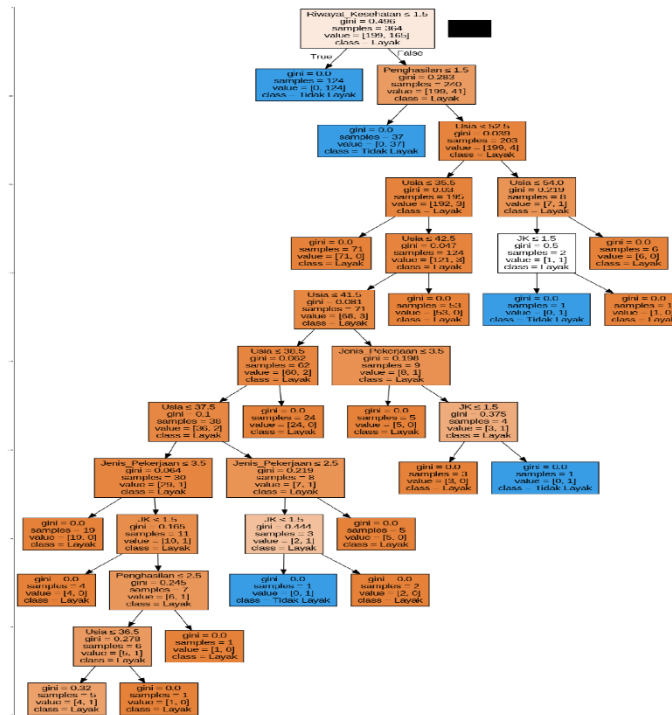
Evaluasi kinerja model decision tree C4.5 menggunakan metode Confusion Matrix dengan hasil perhitungan dapat dilihat pada Gambar 4.3 berikut :



Gambar 4.3 Hasil Perhitungan Confusion Matrix Uji Coba 1

b. Visualisasi Hasil Pemodelan

Hasil pemodelan metode decision tree C4.5 dapat dilihat divisualisasikan pada Gambar 4.4 berikut :



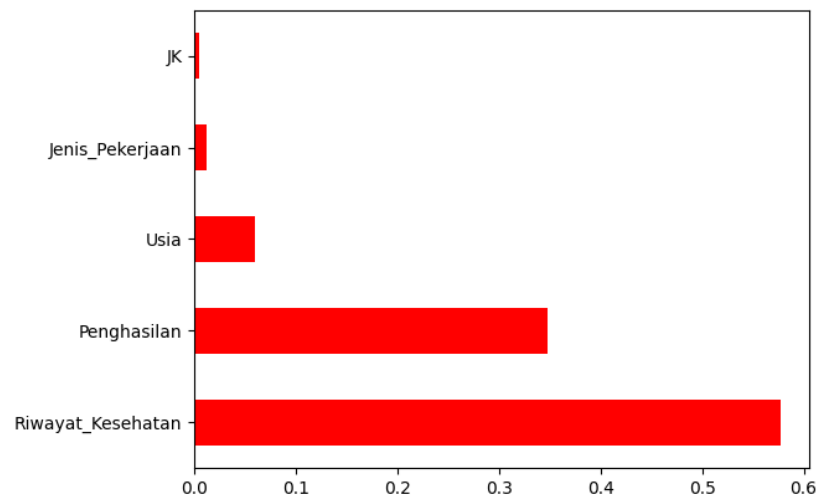
Gambar 4.4 Visualisasi Pemodelan Decision Tree C4.5

6. Perbaikan Model

Berdasarkan hasil evaluasi kinerja model di atas, belum didapatkan hasil yang maksimal sehingga perlu dilakukan perbaikan atau optimalisasi Model diantaranya :

a. Optimalisasi Atribut

Beberapa metode bisa digunakan dalam pemilihan atribut yang berpengaruh ke hasil akhir. Pada penelitian digunakan metode Feature Importances dengan hasil perhitungan dapat dilihat pada Gambar 4.5 berikut :



Gambar 4.5 Grafik Kepentingan Atribut

Berdasarkan Gambar 4.6 di atas, maka atribut yang akan digunakan pada pemodelan kedua ini yaitu : Usia, Penghasilan, Dan Riwayat Kesehatan.

b. Mengukur Kedalaman Pohon

Salah satu teknik untuk mendapat pohon keputusan yang tidak terlalu panjang, maka dapat dilakukan pengukuran kedalaman pohon yang maksimal dengan akurasi yang tinggi. Berikut hasil perhitungan kedalaman dan akurasi dapat dilihat pada Tabel 4.5 berikut :

Tabel 4.9 Hasil Perhitungan Kedalaman Pohon Dan Akurasinya

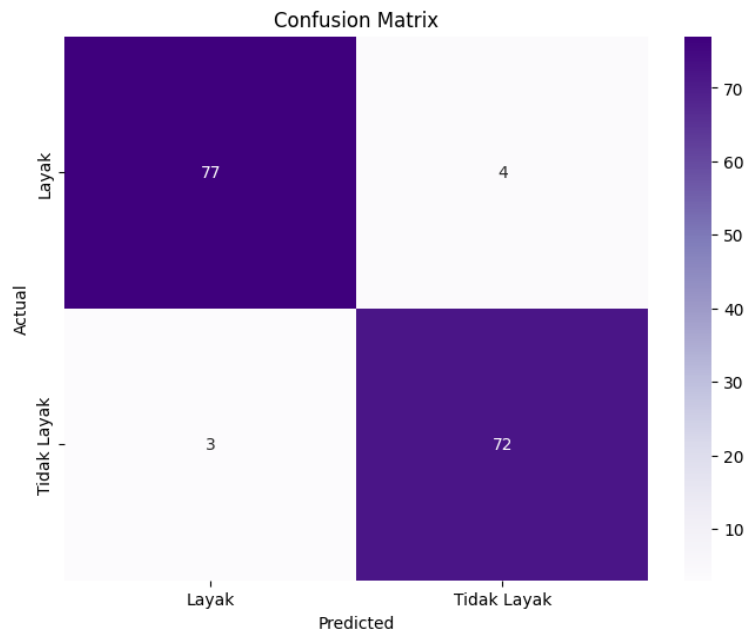
Kedalaman Pohon	Akurasi
1	85%
2 - 10	96%

c. Pemodelan Hasil Optimalisasi

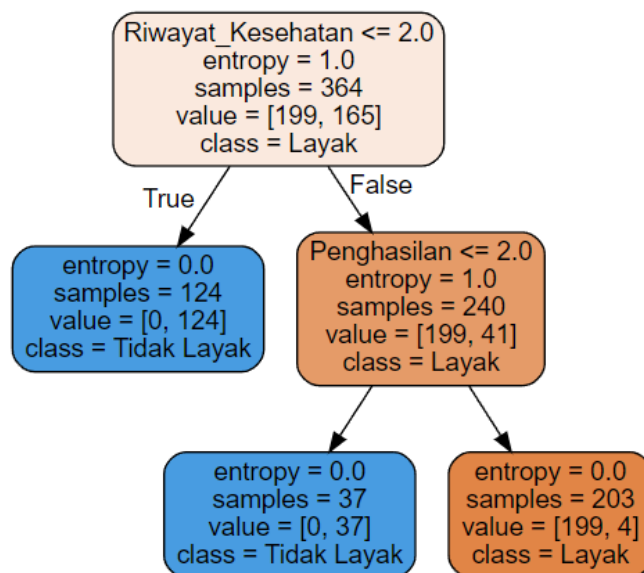
Setelah dilakukan beberapa optimalisasi, selanjutnya diterapkan pada model *Decision Tree C4.5*, kemudian dilakukan evaluasi kinerja kembali.

d. Evaluasi Kinerja Model

Hasil evaluasi kinerja model setelah dilakukan optimalisasi dapat dilihat pada Gambar 4.6 dan Gambar 4.7 berikut ini :



Gambar 4.6 Hasil Perhitungan Confusion Matrix Setelah Optimalisasi



Gambar 4.7 Visualisasi Model Setelah Optimalisasi

7. Testing Data Baru

Setelah didapatkan hasil pemodelan yang optimal, maka model tersebut dapat digunakan untuk melakukan klasifikasi data testing baru dengan memasukkan nilai atribut sesuai dengan model yang sudah dilatih. Adapun hasil ujicoba klasifikasi data baru dapat dilihat pada Tabel 4.6 dan Tabel 4.7 berikut ini :

Tabel 4.10 Data Testing Baru

No.	Usia	Penghasilan	Riwayat Kesehatan	Ket
1	34	3	2	?
4	31	1	1	?
6	34	3	2	?
7	27	1	1	?
8	30	1	1	?

Tabel 4.11 Hasil Klasifikasi Data Testing Baru

No.	Usia	Penghasilan	Riwayat Kesehatan	Ket
1	34	3	2	Layak
4	31	1	1	Tidak Layak
6	34	3	2	Layak
7	27	1	1	Tidak Layak
8	30	1	1	Tidak Layak

4.3 Hasil Pengembangan Model

Pengembangan Model *Decision Tree C4.5* menggunakan bahasa pemrograman Python dengan tools Google Colaboratory dengan tahapan-tahapan berikut :

4.3.1 Seleksi Data

Pada Tahapan Seleksi Data ini dilakukan beberapa proses diantara :

1. Mengaktifkan penggunaan Goole Drive sebagai tempat penyimpanan, adapun coding dengan Python sebagai berikut :

```
#Proses Pengaitan dengan Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

2. Membaca Dataset File Excel

```
#Membaca Dataset
df = pd.read_excel("/content/drive/MyDrive/Folder
Skripsi/Dataset/Dataset4.xlsx")
df.head()
```

Hasil output sebagai berikut :

No.	Nama	JK	Jenis_Pekerjaan	Usia	Penghasilan	Riwayat_Kesehatan	Ket
1	Nasabah 1	L	Wiraswasta	34	7.000.000_sd_10.000.000	Tidak_Ada	Layak
2	Nasabah 2	P	Swasta	31	10.000.000_sd_25.000.000	Tidak_Ada	Tidak Layak
3	Nasabah 3	P	Swasta	34	10.000.000_sd_25.000.000	Tidak_Ada	Layak
4	Nasabah 4	L	PNS	27	7.000.000_sd_10.000.000	Tidak_Ada	Tidak Llayk
5	Nasabah 5	L	Wiraswasta	30	5.000.000_sd_7.000.000	Penyakit_Kritis	Tidak Layak

4.3.2 Preprocessing

Pada tahapan preprocessing dilakukan beberapa proses seperti :

1. Melakukan pengecekan data yang kosong atau data yang tidak konsisten terhadap hasil akhir atau kelas output dengan melakukan perintah sebagai berikut:

```
# Mengelompokkan data berdasarkan atribut yang
ingin diperiksa
grouped_data = df.groupby(['JK',
'Jenis_Pekerjaan', 'Usia', 'Penghasilan',
'Riwayat_Kesehatan'])

# Memeriksa kategori kelas untuk setiap kelompok
for key, group in grouped_data:
    unique_classes = group['Ket'].unique()
    if len(unique_classes) > 1:
        print(f"Kelompok {key}: Kelas yang berbeda
= {unique_classes}")
```

Dan hasil keluaran dari perintah diatas sebagai berikut :

```
Kelompok ('L', 'Swasta', 35, '7.000.000_sd_10.000.000',
'Tidak_Ada'): Kelas yang berbeda = ['Layak' 'Tidak Layak']
Kelompok ('L', 'Swasta', 38, '7.000.000_sd_10.000.000',
'Tidak_Ada'): Kelas yang berbeda = ['Layak' 'Tidak Layak']
Kelompok ('P', 'Wiraswasta', 36, '7.000.000_sd_10.000.000',
'Tidak_Ada'): Kelas yang berbeda = ['Layak' 'Tidak Layak']
Kelompok ('P', 'Wiraswasta', 42, '7.000.000_sd_10.000.000',
'Tidak_Ada'): Kelas yang berbeda = ['Tidak Layak' 'Layak']
```

2. Melakukan proses validasi data setelah dilakukan perbaikan data dan menghitung kembali jumlah data masing-masing kelas dengan perintah berikut :

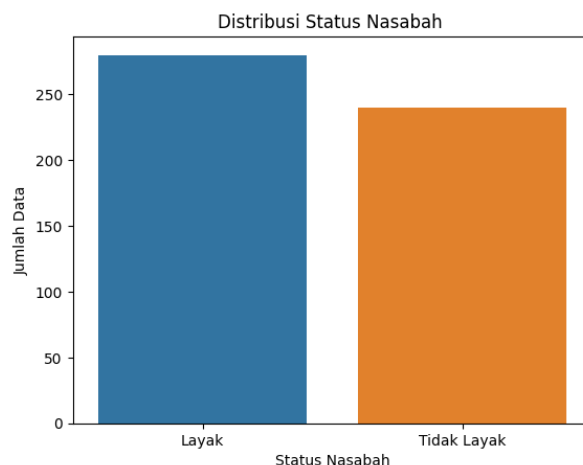
```
# Menghitung jumlah setiap kategori dalam kolom
'Keterangan'
countplot = sns.countplot(data=df, x='Ket')

# Menambahkan label pada sumbu x dan y
plt.xlabel('Status Nasabah')
plt.ylabel('Jumlah Data')

# Menambahkan judul
plt.title('Distribusi Status Nasabah ')

# Menampilkan grafik
plt.show()
```

Hasil keluaran dari perintah diatas ditampilkan pada gambar 4.8 sebagai berikut :



Gambar 4.8 Grafik Jumlah Kelas

4.3.3 Transformasi

Pada tahap tranformasi terdiri dari beberapa proses yakni :

1. Rekonstruksi Data

Proses rekonstruksi data dilakukan untuk mengubah atau mengkonversi atribut yang bernilai string menjadi numerik. Terdapat atribut string pada dataset antara lain yaitu atribut jenis kelamin, jenis pekerjaan, penghasilan, dan riwayat kesehatan. Untuk proses konversi atribut ini diperlukan perintah :

```
#Proses Konversi Nilai String ke Numerik Jenis Kelamin

df['JK'].replace(['L','P'],[1,2], inplace=True)
df.head()
```

Pada atribut jenis kelamin terdapat 2 jenis nilai string yaitu L dan P, kedua nilai ini dikonversi kedalam nilai numerik sehingga nilai akhir menjadi 1 dan 2 dimana L dikonversi ke 1 dan P dikonversi ke 2.

Perintah untuk rekonstruksi atribut jenis pekerjaan :

```
#Proses Konversi Nilai String ke Numerik Atribut Jenis
Pekerjaan
df['Jenis_Pekerjaan'].replace(['Nelayan','Swasta','PNS',
'Wiraswasta'],[1,2,3,4], inplace=True)
df.head()
```

Untuk atribut jenis pekerjaan terdapat 4 jenis nilai string seperti nelayan, swasta, pns, dan wiraswasta nilai ini dikonversi ke numerik menjadi 1 – 4 dimana nelayan bernilai 1, swasta bernilai 2, pns bernilai 3, dan wiraswasta bernilai 4.

Perintah untuk rekonstruksi atribut penghasilan :

```
#Proses Konversi Nilai String ke Numerik Atribut
Penghasilan
df
df['Penghasilan'].replace(['5.000.000_sd_7.000.000','7.0
00.000_sd_10.000.000','10.000.000_sd_25.000.000','25.0
00.000_sd_50.000.000'],[1,2,3,4], inplace=True)
df.head()
```

Atribut penghasilan terdapat 4 jenis nilai string dan dikonversi ke numerik menjadi 1 – 4 dimana 5.000.000 s.d. 7.000.000 bernilai 1, 7.000.000 s.d. 10.000.000 bernilai 2, 10.000.000 s.d. 25.000.000 bernilai 3, dan 25.000.000 s.d. 50.000.000 bernilai 4.

Perintah untuk rekonstruksi atribut riwayat kesehatan :

```
#Proses Konversi Nilai String ke Numerik Atribut
Riwayat Kesehatan
df['Riwayat_Kesehatan'].replace(['Penyakit_Kritis', 'Ti
dak_Ada'], [1,2], inplace=True)
df.head()
```

Atribut riwayat kesehatan terdapat 2 jenis nilai string, kedua nilai ini kemudian dikonversi kedalam nilai numerik dimana penyakit kritis bernilai 1 dan tidak ada bernilai 2.

Dan hasil dari perintah diatas adalah sebagai berikut :

No.	Nama	JK	Jenis_Pekerjaan	Usia	Penghasilan	Riwayat_Kesehatan	Ket
1	Nasabah 1	1	4	52	2	2	Layak
2	Nasabah 2	2	2	34	3	2	Layak
3	Nasabah 3	2	2	50	3	2	Layak
4	Nasabah 4	1	3	31	2	2	Layak
5	Nasabah 5	1	4	31	1	1	Tidak Layak

4.3.4 Proses Data Mining

1. Proses Pembagian Data Training Dan Data Testing

Pada tahap ini, data akan dibagi menjadi dua bagian yaitu data training dan data testing. Untuk proses pembagian dengan membagi data training sebesar 70% dan data testing sebesar 30% dimana data training berjumlah 364 data dan data testing berjumlah 156 data. Untuk proses pembagiannya dapat dilakukan dengan perintah :

```
# Membagi dataset menjadi data latih & data uji
from sklearn.model_selection import train_test_split
X_train1, X_test1, y_train1, y_test1 =
train_test_split(X, y, test_size=0.3, random_state=0)
print ('Train set:', X_train1.shape1, y_train1.shape)
print ('Test set:', X_test1.shape1, y_test1.shape)
```

Kemudian akan menghasilkan keluaran berikut :

Train set: (364, 5) (364,)
Test set: (156, 5) (156,)

Adapun juga data training dan testing yang digunakan ada pada tabel 4.8 dan 4.9 berikut :

Tabel 4.12 Dataset Training

No.	JK	Jenis Pekerjaan	Usia	Penghasilan	Riwayat Kesehatan	Ket
0	1	4	52	2	2	Layak
2	2	2	50	3	2	Layak
3	1	3	31	2	2	Layak
5	1	4	41	3	2	Layak
9	1	3	36	3	2	Layak
11	1	3	45	1	1	Tidak Layak
13	1	3	36	4	2	Layak
14	1	4	38	4	2	Layak
...
519	1	4	49	1	1	Tidak Layak

Tabel 4.13 Dataset Testing

No.	JK	Jenis Pekerjaan	Usia	Penghasilan	Riwayat Kesehatan	Ket
1	2	2	34	3	2	Tidak Layak
4	1	4	31	1	1	Layak
6	1	2	34	3	2	Layak
7	2	4	27	1	1	Tidak Layak
8	1	4	30	1	1	Layak
10	1	4	38	2	2	Layak
12	1	2	32	1	1	Tidak Layak
15	2	2	39	2	2	Layak
21	1	1	27	2	2	Layak
26	1	2	55	2	2	Layak

4.3.5 Evaluasi

1. Evaluasi Kinerja Model

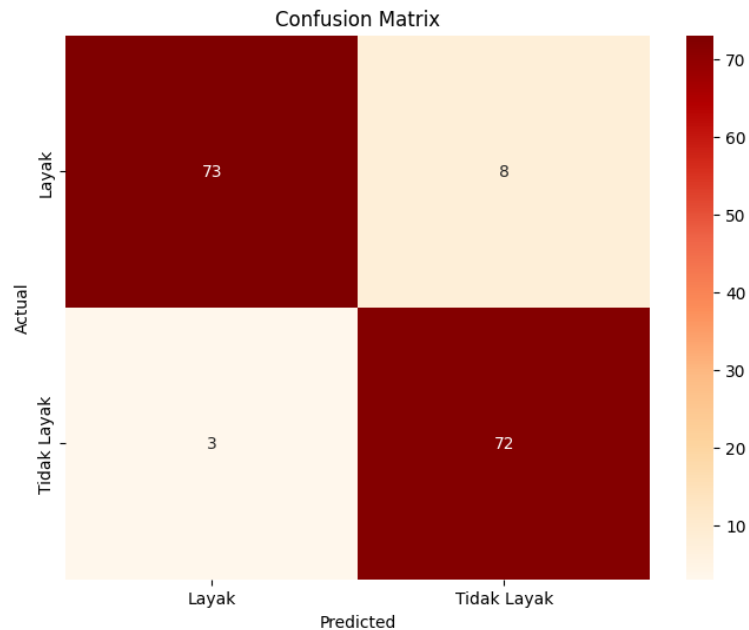
Pada tahap ini, akan dilakukan evaluasi kinerja model dari *Decision Tree* menggunakan metode *Confusion Matrix*, dimana proses ini akan dilakukan proses evaluasi dari model sebelumnya. Dataset yang diprediksi akan menggunakan data hasil uji coba 1 sebelumnya. Dengan memasukan perintah :

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# 6. Evaluasi Model
y_pred = tree_model.predict(X_test1)
labels=y
# Membuat confusion matrix
cm = confusion_matrix(y_test1, y_pred)

# Mencetak confusion matrix
labels = list(y.unique()) # Pastikan y adalah DataFrame
atau Series yang memiliki label kelas
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', xticklabels=labels,
yticklabels=labels, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

Setelah perintah dijalankan, maka akan menampilkan hasil keluaran seperti pada gambar 4.9 berikut :



Gambar 4.9 Confusion Matrix Uji Coba 1

Pada gambar 4.9 diatas, menunjukan hasil uji coba 1 dari proses evaluasi dengan *Confusion Matrix* yang diperoleh dari data testing yang berjumlah 156 data. Akan tetapi, perlu adanya perbaikan model karena masih dilakukan penghapusan atribut yang tidak berpengaruh.

2. Visualisasi Hasil Pemodelan

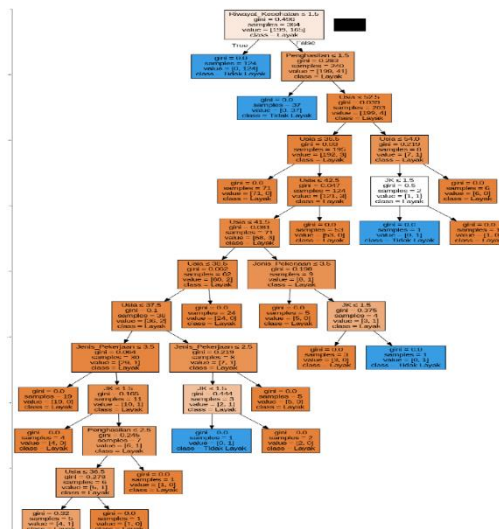
Pada tahap ini dilakukan proses pemodelan dengan metode *decision tree*, dimana nantinya pemodelan ini akan menghasilkan pohon keputusan akhir. Untuk perintah yang digunakan dalam proses pemodelan ini sebagai berikut :

```
from six import StringIO
import pydotplus
import matplotlib.image as mpimg
from sklearn import tree

%matplotlib inline

dot_data = StringIO()
filename = "/content/drive/MyDrive/Folder
Skripsi/pohon1.png"
featureNames = ['JK', 'Jenis_Pekerjaan', 'Usia',
'Penghasilan', 'Riwayat_Kesehatan']
targetNames = ['Layak', 'Tidak Layak']
out = tree.export_graphviz(tree_model,
feature_names=featureNames, out_file=dot_data,
class_names= np.unique(y_train1),
filled=True, special_characters=True, rotate=False)
graph =
pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png(filename)
img = mpimg.imread(filename)
plt.figure(figsize=(200, 100))
plt.imshow(img, interpolation='nearest')
```

Setelah melakukan perintah diatas, akan menampilkan hasil keluaran seperti pada gambar 4.10 berikut :



Gambar 4.10 Visualisasi Pemodelan Decision Tree C4.5

4.3.6 Perbaikan Model

1. Optimalisasi Atribut

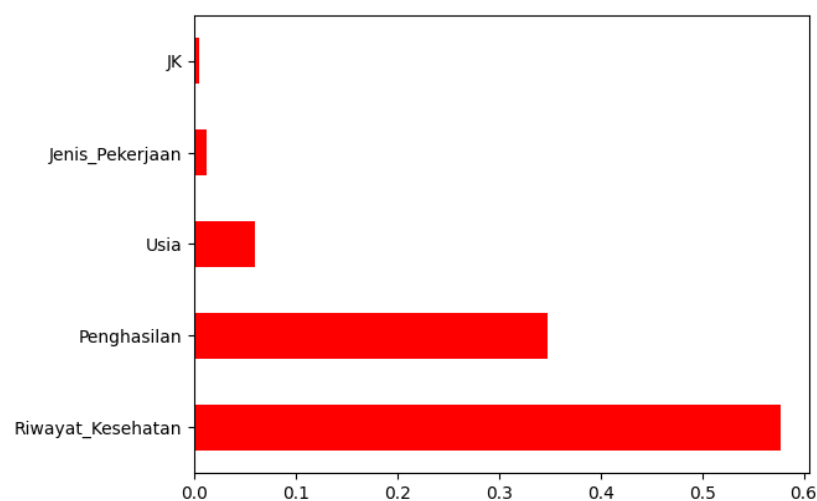
Tahapan ini akan dilakukan proses evaluasi terhadap variable atau atribut yang tidak berpengaruh dengan penerapan feature importance dengan perintah seperti berikut :

```
from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X,y)

print(model.feature_importances_) #use inbuilt class
feature_importances of tree based classifiers

#plot graph of feature importances for better
visualization
feat_importances = pd.Series(model.feature_importances_,
index=X.columns)
feat_importances.nlargest(10).plot(kind='barh',
color='red')
plt.show()
```

Hasil keluaran dari perintah diatas ditunjukkan pada gambar 4.11 berikut :



Gambar 4.11 Grafik Presentase Atribut

Berdasarkan grafik pada gambar 4.11, terdapat atribut jenis kelamin dan jenis pekerjaan yang tidak berpengaruh dalam proses klasifikasi dan pada hasil pohon keputusan, sehingga adanya penghapusan atribut.

2. Pemilahan Atribut

Pada tahap ini, dilakukan proses pemilahan terhadap atribut yang tidak digunakan. Beberapa atribut yang tidak digunakan akan dikeluarkan sehingga hasil dari pemilahan atribut data adalah sebagai berikut :

```
#Proses Penghapusan Atribut Yang tidak Penting
df.drop(['No.', 'Nama', 'JK', 'Jenis_Pekerjaan'], axis=1, inplace=True)
df.head()
```

Setelah perintah diatas dijalankan, hasil keluarannya sebagai berikut:

	Usia	Penghasilan	Riwayat_Kesehatan	Ket
0	52	2	2	Layak
1	34	3	2	Layak
2	50	3	2	Layak
3	31	2	2	Layak
4	31	1	1	Tidak Layak

3. Mengukur Kedalaman Pohon

Untuk mendapatkan pohon dengan model yang minimalis dilakukan proses pengukuran kedalaman pohon yang maksimal berdasarkan tingkat akurasi tertinggi yang didapatkan dengan menjalankan perintah sebagai berikut :

```

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Mencoba berbagai nilai kedalaman pohon
for max_depth in range(1, 11):
    clf = DecisionTreeClassifier(max_depth=max_depth)
    clf.fit(X_train2, y_train2)
    y_pred = clf.predict(X_test)
    accuracy = accuracy_score(y_test2, y_pred)
    print(f"Max Depth = {max_depth}, Accuracy =
{accuracy:.2f}")

# Mencoba berbagai nilai minimum sampel untuk
percabangan
for min_samples_split in range(2, 11):
    clf =
DecisionTreeClassifier(min_samples_split=min_samples_s
plit)
    clf.fit(X_train2, y_train2)
    y_pred = clf.predict(X_test2)
    accuracy = accuracy_score(y_test2, y_pred)
    print(f"Min Samples Split = {min_samples_split},
Accuracy = {accuracy:.2f}")

```

Kemudian hasil keluaran perintah diatas seperti :

```

Max Depth = 1, Accuracy = 0.85
Max Depth = 2, Accuracy = 0.96
Max Depth = 3, Accuracy = 0.96
Max Depth = 4, Accuracy = 0.96
Max Depth = 5, Accuracy = 0.96
Max Depth = 6, Accuracy = 0.96
Max Depth = 7, Accuracy = 0.96
Max Depth = 8, Accuracy = 0.96
Max Depth = 9, Accuracy = 0.96
Max Depth = 10, Accuracy = 0.96
Min Samples Split = 2, Accuracy = 0.96
Min Samples Split = 3, Accuracy = 0.96
Min Samples Split = 4, Accuracy = 0.96
Min Samples Split = 5, Accuracy = 0.96
Min Samples Split = 6, Accuracy = 0.96
Min Samples Split = 7, Accuracy = 0.96
Min Samples Split = 8, Accuracy = 0.96
Min Samples Split = 9, Accuracy = 0.96
Min Samples Split = 10, Accuracy = 0.96

```

Berdasarkan hasil dari keluaran di atas, maka dipilih kedalaman pohon dengan nilai 96%.

4.3.7 Pemodelan Hasil Optimalisasi

Setelah dilakukan beberapa optimalisasi, selanjutnya pada tahap ini diterapkan model *Decision Tree C4.5* dengan hasil optimalisasi, kemudian dilakukan evaluasi kinerja kembali dengan tahapan berikut :

1. Evaluasi Kinerja Model

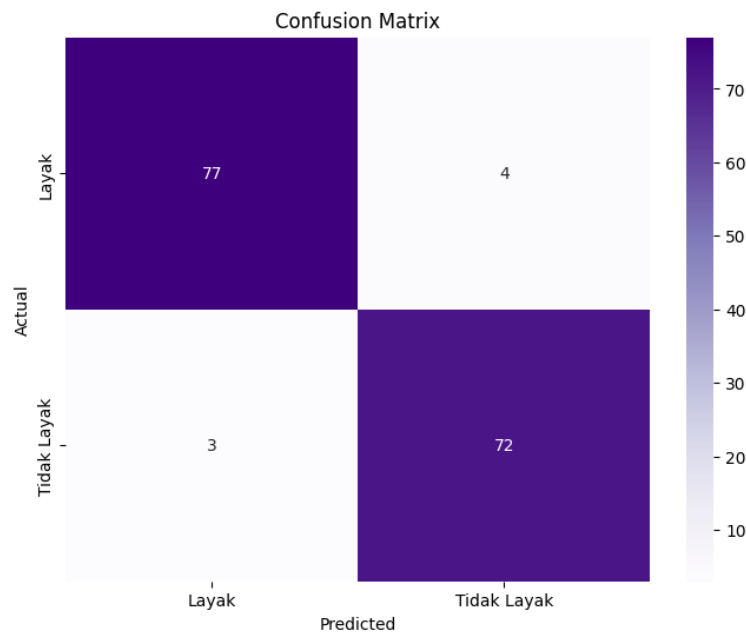
Pada tahap ini, akan dilakukan proses evaluasi kinerja model dari *Decision Tree* kembali menggunakan metode *Confusion Matrix*.

```
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# 6. Evaluasi Model 2
y_pred = tree_model2.predict(X_test2)
labels=y
# Membuat confusion matrix
cm = confusion_matrix(y_test2, y_pred)

# Mencetak confusion matrix
labels = list(y.unique()) # Pastikan y adalah DataFrame
atau Series yang memiliki label kelas
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', xticklabels=labels,
yticklabels=labels, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

Setelah perintah dijalankan, maka akan menampilkan hasil keluaran seperti pada gambar 4.12 berikut :



Gambar 4.12 Confusion Matrix Setelah Optimalisasi

Pada gambar 4.12 diatas, menunjukan hasil evaluasi yang di optimalkan dengan menggunakan *Confusion Matrix* dan memperoleh nilai akurasi sebesar 95%. Pada tahap selanjutnya akan dilakukan proses prediksi nilai akurasi sesuai dengan model dari *Confusion Matrix* yang didapatkan.

2. Prediksi Nilai Akurasi

Pada proses ini dilakukan prediksi nilai akurasi dari hasil evaluasi model dari *Decision Tree*, Dataset yang diprediksi akan menggunakan data uji yang telah dibagi sebelumnya. Dengan memasukan perintah :

```
# Evaluasi Model
from sklearn.metrics import accuracy_score

y_pred = tree_model2.predict(X_test2)

acc_score = round(accuracy_score(y_pred, y_test2), 3)

print('Accuracy: ', acc_score)
```


Setelah perintah diatas dijalankan maka akan menghasilkan keluaran nilai akurasi sebagai berikut :

```
Accuracy: 0.955
```

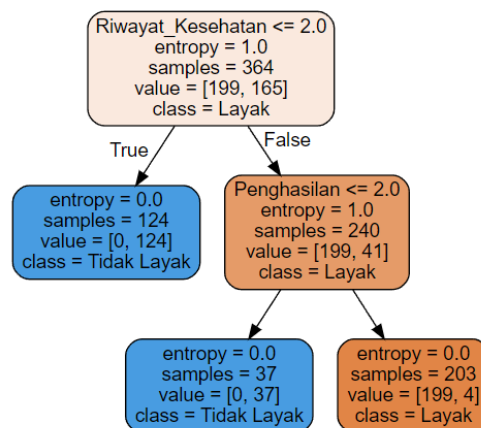
3. Pemodelan Dengan Decision Tree

Pada tahap ini dilakukan proses pemodelan dengan metode *decision tree*, dimana nantinya pemodelan ini akan menghasilkan pohon keputusan akhir. Untuk perintah yang digunakan dalam proses pemodelan ini sebagai berikut :

```
from sklearn.tree import DecisionTreeClassifier
# membuat model Decision Tree
tree_model2 = DecisionTreeClassifier()
# Melatih model dengan menggunakan data latih
tree_model2 = tree_model.fit(X_train2, y_train2)
```

4. Visualisasi Model

Setelah pemodelan dengan *decision tree*, pada tahap ini akan ditampilkan pohon keputusan dari hasil pemodelan yang telah dibuat sebelumnya yang terdiri dari beberapa node (rule). Hasil dari visualisasi model *Decision Tree* dapat dilihat pada gambar 4.13 berikut :



Gambar 4.13 Confusion Matrix Setelah Optimalisasi

4.3.8 Klasifikasi Data Testing Baru

Pada tahap ini, dilakukan proses pemodelan hasil setelah optimalisasi, model tersebut digunakan untuk melakukan klasifikasi data testing baru dengan memasukkan nilai atribut sesuai dengan model yang sudah dilatih. Dengan beberapa tahapan sebagai berikut :

1. Membaca Dataset Prediksi

```
#Membaca Dataset Prediksi
datapred = pd.read_excel("/content/drive/MyDrive/Folder
Skripsi/Dataset/Datapredict.xlsx")
datapred.head()
```

Hasil keluaran perintah diatas sebagai berikut :

	No.	Usia	Penghasilan	Riwayat Kesehatan	Ket
0	1	34	3	2	Layak
1	4	31	1	1	Tidak Layak
2	6	34	3	2	Layak
3	7	27	1	1	Tidak Llayk
4	8	30	1	1	Tidak Layak

2. Konversi File Excel Ke CSV

```
#Membaca Dataset Prediksi
datapred = pd.read_excel("/content/drive/MyDrive/Folder
Skripsi/Dataset/Datapredict.xlsx")
datapred.head()
```

Kemudian hasil keluaran perintah diatas sebagai berikut :

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5 entries, 0 to 4
Data columns (total 5 columns):
#   Column                Non-Null Count  Dtype
---  -
0   No.                    5 non-null     int64
1   Usia                   5 non-null     int64
2   Penghasilan            5 non-null     int64
3   Riwayat_Kesehatan     5 non-null     int64
4   Ket                    5 non-null     object
dtypes: int64(4), object(1)
memory usage: 328.0+ bytes

```

3. Proses Cetak Dataset Prediksi

Setelah file dikonversi ke format csv, kemudian dilakukan langkah selanjutnya yaitu proses mencetak dataset prediksi dengan melakukan perintah :

```

dataX = datapred[['Usia', 'Penghasilan',
'Riwayat_Kesehatan']]
print(dataX)

```

hasil keluaran dari perintah diatas menampilkan :

	No.	Usia	Penghasilan	Riwayat Kesehatan
0	1	34	3	2
1	4	31	1	1
2	6	34	3	2
3	7	27	1	1
4	8	30	1	1

4. Proses Prediksi Model Dengan Data Baru

Setelah ditampilkan dataset prediksi kemudia dilakukan proses memprediksi model menggunakan data baru dengan perintah berikut :

```
# prediksi model dengan data baru  
(Usia,Penghasilan,Riwayat_Kesehatan)  
hasil=tree_model.predict(dataX)  
print(hasil)
```

hasil keluaran dari perintah diatas sebagai berikut :

```
['Layak' 'Tidak Layak' 'Layak' 'Tidak Layak' 'Tidak Layak']
```

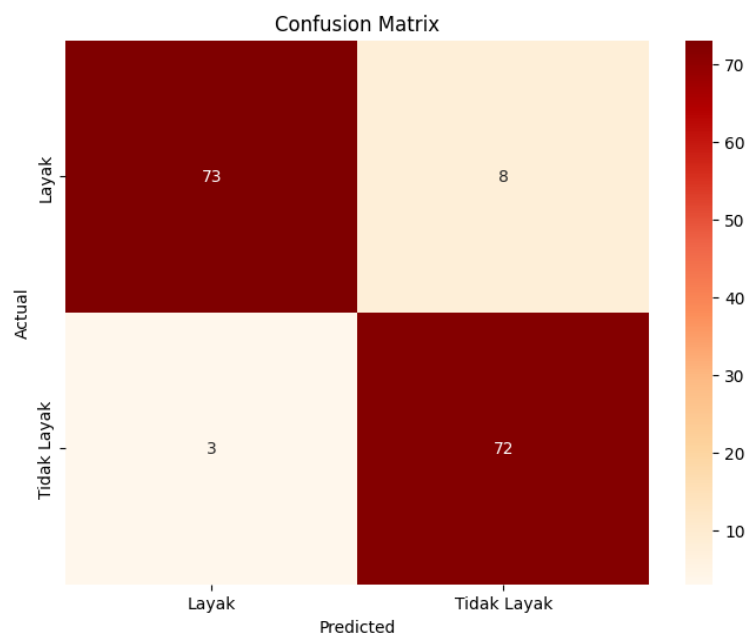
BAB V

PEMBAHASAN PENELITIAN

5.1 Pembahasan Kinerja Model

Kinerja dari metode *Decision Tree* dalam pengklasifikasian data tentu sangat berdasarkan dari banyaknya data yang diperoleh, dengan meminimalisir waktu dalam melakukan uji coba beberapa jumlah prediksi yang tepat, maka diperlukan suatu teknik untuk menentukan jumlah prediksi dengan hasil yang optimal menggunakan teknik *Confusion Matrix*. Berikut adalah optimalisasi *Confusion Matrix* dengan 2 kali pengujian jumlah prediksi berdasarkan hasil optimalisasi :

1. *Confusion Matrix* Sebelum Optimalisasi



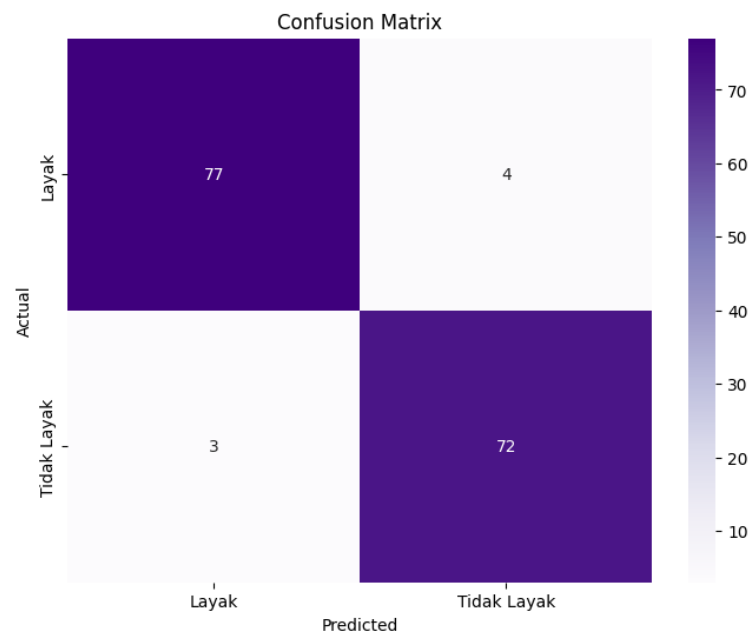
Gambar 5.1 Confusion Matrix Sebelum Optimalisasi

Pada gambar 5.1 menunjukkan hasil perhitungan dengan *confusion matrix* sebelum dioptimalisasi. Pada hasil ini didapatkan tingkat *accuracy* skor sebesar 93%, *precision* 90%, dan *recall* 96%. Untuk detail info perhitungannya dapat dilihat pada tabel 5.1 berikut :

Tabel 5.1 Tabel Perhitungan Confusion Matrix Sebelum Optimalisasi

Keterangan	Precision	Recall	F1-Score	Jumlah Data
Layak	0.96	0.90	0.93	81
Tidak Layak	0.90	0.96	0.93	75
Accuracy			0.93	156

2. Confusion Matrix Setelah Optimalisasi

**Gambar 5.2** Confusion Matrix Setelah Optimalisasi

Pada gambar 5.2 menunjukkan hasil perhitungan dengan *confusion matrix* setelah dioptimalisasi dengan proses penghapusan atribut yang tidak penting, mengukur kedalaman pohon berdasarkan akurasi tertinggi. Pada hasil ini didapatkan tingkat *accuracy* skor sebesar 96%, *precision* 95%, dan *recall* 96%. Untuk detail info perhitungannya dapat dilihat pada tabel 5.2 berikut :

Tabel 5.2 Perhitungan Confusion Matrix Setelah Optimalisasi

Keterangan	Precision	Recall	F1-Score	Jumlah Data
Layak	0.96	0.95	0.96	81
Tidak Layak	0.95	0.96	0.95	75
Accuracy			0.96	156

- Accuracy (akurasi), adalah sebuah metrik yang mengukur sejauh mana model klasifikasi dapat mengidentifikasi jumlah total prediksi dengan benar.
- Precision (presisi), adalah metrik yang melakukan pengukuran sejauh mana prediksi dari model (True Positive) adalah benar. Presisi mengukur seberapa baik model dalam menghindari dan memberikan prediksi positif yang salah.
- Recall adalah metrik yang berguna ketika kesalahan negatif (False Negative) memiliki konsekuensi yang serius atau ketika tujuan utama adalah untuk mengidentifikasi semua instance positif yang ada dalam data.
- F1-Score adalah metrik evaluasi yang menggabungkan presisi (Precision) dan recall dalam satu angka tunggal untuk gambaran yang lebih komprehensif tentang performa model klasifikasi

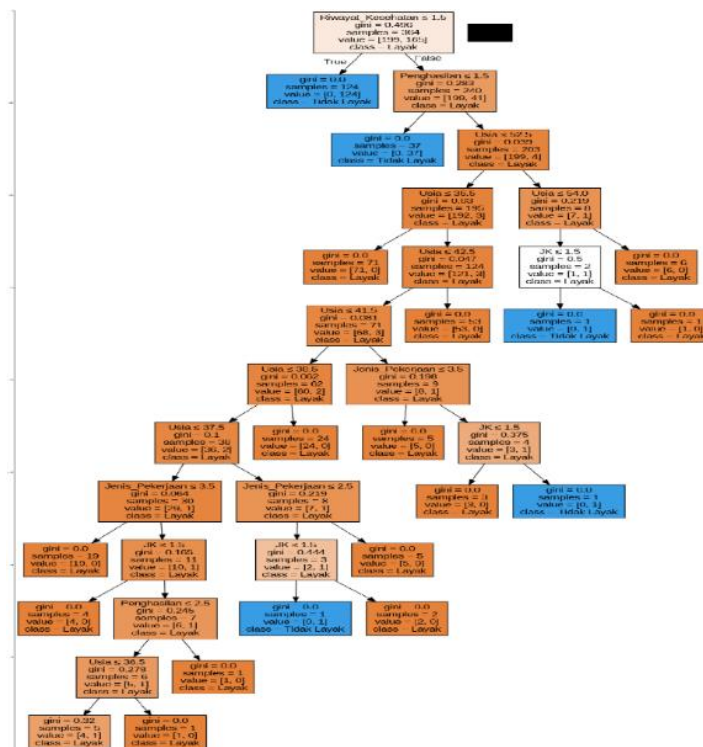
Berdasarkan pengujian kinerja model *Decision Tree* yang dilakukan dengan 2 kali uji coba yaitu sebelum optimisasi model didapatkan hasil akurasi sebesar 93% sebagaimana dapat dilihat pada Tabel 5.1 di atas dan setelah dilakukan optimalisasi model didapatkan hasil akurasi sebesar 96% seperti pada Tabel 5.2 di atas, dengan demikian dapat dilihat bahwa terjadi peningkatan akurasi sebesar 3% setelah optimalisasi model dilakukan.

5.2 Pembahasan Hasil Pemodelan

Hasil pemodelan *Decision Tree* pada klasifikasi data calon nasabah baru dilakukan 2 kali tahapan uji coba yaitu pemodelan sebelum adanya optimalisasi yang dilakukan seperti seleksi atribut dan mengukur kedalaman pohon. Adapun hasil masing-masing uji coba tersebut diuraikan pada pembahasan berikut.

5.2.1 Hasil Pemodelan Sebelum Optimalisasi

Hasil pemodelan *Decision Tree* sebelum dilakukan optimalisasi didapatkan hasil visualisasi pohon keputusan sebagaimana ditampilkan pada Gambar 5.3 berikut ini :



Gambar 5.3 Pohon Keputusan Sebelum Optimalisasi

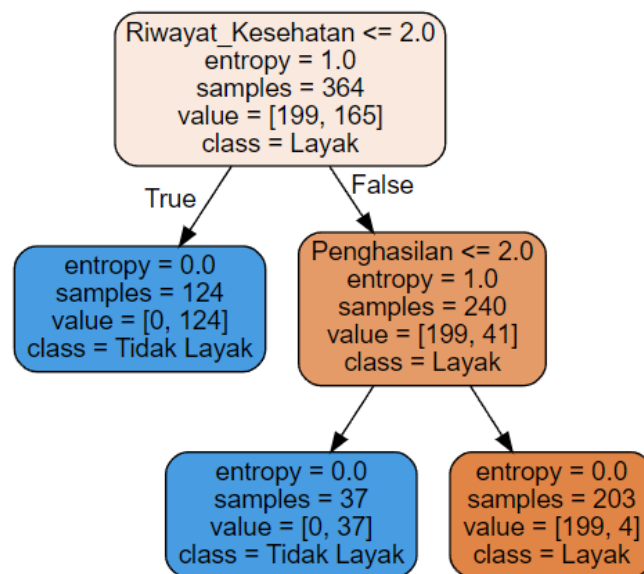
Berdasarkan Gambar 5.3 di atas dapat dilihat bahwa pohon keputusan yang terbentuk belum maksimal dan masih terlalu banyak jalur keputusan yang tidak signifikan yaitu terdapat lebih dari 30 jalur keputusan, hal ini terjadi karena masih terdapat data yang tidak konsisten terhadap kelas output, sehingga untuk mengatasi masalah tersebut perlu dilakukan proses optimalisasi model.

5.2.2 Hasil Pemodelan Setelah Optimalisasi

Hasil pemodelan *Decision Tree* setelah dilakukan optimalisasi model terkait seleksi atribut, dan mengukur kedalaman pohon. Pada optimalisasi Seleksi atribut terpilih atribut usia, penghasilan, dan Riwayat kesehatan sesuai hasil pengukuran menggunakan metode *Decision Tree C4.5*.

Pada optimalisasi pengukuran kedalaman didapatkan hasil kedalaman pohon 2 yang memiliki tingkat akurasi tertinggi, sehingga nilai kedalaman ini yang digunakan dalam pemodelan *Decision Tree*.

Berdasarkan hasil optimalisasi tersebut di atas dijadikan sebagai pemodelan *Decision Tree* yang digunakan dalam penelitian ini dengan hasil visualisasi pohon keputusan dapat dilihat pada Gambar 5.4 berikut ini :



Gambar 5.4 Pohon Keputusan Setelah Optimalisasi

Berdasarkan Gambar 5.4 di atas dapat dilihat bahwa pohon keputusan yang terbentuk sudah lebih sederhana dan jumlah jalur atau rule keputusan yang terbentuk sisa 3 rule. Adapun jalur atau rule keputusan secara detail diuraikan pada Tabel 5.3 berikut ini :

Tabel 5.3 Rule Akhir

No.	Rule
1	Jika Riwayat Kesehatan ≤ 2.0 (Penyakit kritis) Maka Tidak Layak
2	Jika Riwayat kesehatan > 2.0 (Tidak Ada) Dan Penghasilan ≤ 2.0 (5.000.000 sd 7.000.000) Maka Tidak Layak
3	Jika Riwayat kesehatan > 2.0 (Tidak Ada) Dan Penghasilan > 2.0 (10.000.000 sd 50.000.000) Maka Layak

Terdapat 3 aturan (Rule) yang ditemukan untuk klasifikasi status kelayakan nasabah baru. Selanjutnya 3 aturan tersebut dapat digunakan sebagai kelas atau kelompok untuk klasifikasi calon nasabah baru sebagai dasar pemberian pertanggungan asuransi pada PT. Prudential Life Assurance GL1 Gorontalo. Berdasarkan rule keputusan di atas, klasifikasi data calon nasabah baru dapat ditentukan dengan cukup melihat pohon atau rule keputusan tersebut. Rule keputusan ini juga bisa diterapkan pada sistem informasi *pruForce* PT. Prudential Life Assurance kedepannya agar dengan mudah bisa dilakukan monitoring atau evaluasi pada pemberian pertanggungan asuransi.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil penelitian dengan sistem untuk klasifikasi calon nasabah baru sebagai dasar pemberian pertanggungan asuransi menggunakan algoritma *Decision Tree C4.5* pada PT.Prudential Life Assurance GL1 Gorontalo, maka pada akhir laporan penelitian ini penulis menyimpulkan bahwa :

1. Penerapan metode *Decision Tree C4.5* pada klasifikasi data calon nasabah baru didapatkan rule keputusan sebanyak 3 rule dengan atribut yang terpilih yaitu : usia, penghasilan, dan riwayat kesehatan. hasil ini didapatkan berdasarkan hasil optimalisasi model dengan menggunakan seleksi atribut dan pengukuran kedalaman pohon.
2. Penerapan metode *decision Tree C4.5* ini didapatkan hasil akurasi sebesar 96%, *Precision* 95%, dan *Recall* 96% pada proses evaluasi model menggunakan *Confusion Matrix*. dengan demikian metode klasifikasi *decision tree C4.5* dapat menjadi solusi yang efektif untuk proses klasifikasi data calon nasabah baru.

6.2 Saran

Sesuai dengan kesimpulan dari laporan diatas, peneliti dapat memberikan saran atau masukan terhadap peneliti selanjutnya, yaitu:

1. Dengan Penelitian selanjutnya diharapkan dapat dilakukan dengan menggunakan data yang lebih besar dan lebih beragam, hal ini akan dapat memberikan informasi yang lebih lengkap agar dapat membantu algoritma klasifikasi untuk membuat keputusan yang lebih akurat.
2. Diharapkan dapat meningkatkan akurasi untuk proses klasifikasi data calon nasabah baru menggunakan algoritma *Decision Tree C4.5* sehingga perusahaan asuransi dapat lebih efektif dalam pengelolaan nasabah dan meningkatkan profitabilitas.

DAFTAR PUSTAKA

- [1] Betrisandi, “Klasifikasi Nasabah Asuransi Jiwa Menggunakan Algoritma Naive Bayes Berbasis Backward Elimination,” 2017.
- [2] Prudential Indonesia, “About Prudential Indonesia,” Prudential Indonesia.
- [3] R. A. Syahfitri, A. P. Windarto, and H. Okprana, “Klasifikasi Calon Nasabah Baru Menggunakan C.45 Sebagai Dasar Pemberian Pertanggunggunaan Asuransi di PT Asuransi Central Asia Pematangsiantar,” *Media Online*), vol. 1, no. 1, pp. 40–48, 2021, [Online]. Available: <https://ejurnal.seminar-id.com/index.php/bulletinds>
- [4] Sunjana, “Klasifikasi Data Nasabah Sebuah Asuransi Menggunakan Algoritma C4.5,” 2010.
- [5] W. W. Dian Adriansyah, *Algoritma C4.5 Untuk Klasifikasi Calon Peserta Lomba Cerdas Cermat Siswa Smp*. Stationery Office, 2018. Accessed: Nov. 09, 2022. [Online]. Available: <http://www.politeknikmeta.ac.id/meta/ojs/>
- [6] P. Bidang Komputer Sains dan Pendidikan Informatika, D. Akademi Perekam dan Informasi Kesehatan Iris Padang Jl Gajah Mada No, and S. Barat, “Jurnal Edik Informatika Data Mining : Klasifikasi Menggunakan Algoritma C4.5 Yuli Mardi,” 2017.
- [7] D. A. Novilla, R. Goejantoro, F. Deny, and T. Amijaya, “Classification of Insurance Data Customers Using Naive Bayes Method (Case Study : PT. Prudential Life MT. Haryono Street Samarinda),” *Jurnal EKSPONENSIAL*, vol. 10, no. 2, 2019.
- [8] Wikipedia, “Python (Bahasa Pemrograman),” Wikipedia.
- [9] Suswantia Andriani, “IMPLEMENTASI ALGORITMA K-NEAREST NEIGHBOR UNTUK MENENTUKANKLASIFIKASI KELULUSAN MAHASISWA TEKNIK INFORMATIKA,” 2023.
- [10] Aws Amazon, “Apa itu Python?,” Aws Amazon.

LAMPIRAN

```
#Mengimport Library
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
#Proses Pengaitan dengan Google Drive
from google.colab import drive
drive.mount('/content/drive')

# 1. Seleksi Data

#Membaca Dataset
df = pd.read_excel("/content/drive/MyDrive/Folder Skripsi/Dataset/Dataset
Test.xlsx")
df.head()

#Mengkonversi File Excel .xlsx ke .csv
df.to_csv("/content/drive/MyDrive/Folder Skripsi/Dataset/Dataset Test",
index=False)
df.info()

# Menghitung jumlah setiap kategori dalam kolom 'Keterangan'
countplot = sns.countplot(data=df, x='Ket')

# Menambahkan label pada sumbu x dan y
plt.xlabel('Status Nasabah')
plt.ylabel('Jumlah Data')

# Menambahkan judul
plt.title('Distribusi Status Nasabah ')

# Menampilkan grafik
plt.show()

# 2. Preprocessing
# Mengelompokkan data berdasarkan atribut yang ingin diperiksa
grouped_data = df.groupby(['JK', 'Jenis_Pekerjaan', 'Usia', 'Penghasilan',
'Riwayat_Kesehatan'])
```

```

# Memeriksa kategori kelas untuk setiap kelompok
for key, group in grouped_data:
    unique_classes = group['Ket'].unique()
    if len(unique_classes) > 1:
        print(f"Kelompok {key}: Kelas yang berbeda = {unique_classes}")

# Menghitung jumlah setiap kategori dalam kolom 'Keterangan'
countplot = sns.countplot(data=df, x='Ket')

# Menambahkan label pada sumbu x dan y
plt.xlabel('Status Nasabah')
plt.ylabel('Jumlah Data')

# Menambahkan judul
plt.title('Distribusi Status Nasabah ')

# Menampilkan grafik
plt.show()

# 3. Transformasi
#Proses Konversi Nilai String ke Numerik Atribut Jenis Kelamin
df['JK'].replace(['L', 'P'], [1, 2], inplace=True)
df.head()

#Proses Konversi Nilai String ke Numerik Atribut Jenis Pekerjaan
df['Jenis_Pekerjaan'].replace(['Nelayan', 'Swasta', 'PNS', 'Wiraswasta'], [1, 2, 3, 4], inplace=True)
df.head()

#Proses Konversi Nilai String ke Numerik Atribut Penghasilan
df
df['Penghasilan'].replace(['5.000.000_sd_7.000.000', '7.000.000_sd_10.000.000', '10.000.000_sd_25.000.000', '25.000.000_sd_50.000.000'], [1, 2, 3, 4], inplace=True)
df.head()

#Proses Konversi Nilai String ke Numerik Atribut Riwayat Kesehatan
df['Riwayat_Kesehatan'].replace(['Penyakit_Kritis', 'Tidak_Ada'], [1, 2], inplace=True)
df.head()

#Menghitung data statistik
df.describe()

```

4. Proses Data Mining

memisahkan atribut dan label

```
X = df[['JK', 'Jenis_Pekerjaan', 'Usia', 'Penghasilan', 'Riwayat_Kesehatan']]
y = df['Ket']
```

Membagi dataset menjadi data latih & data uji

```
from sklearn.model_selection import train_test_split
X_train, X_test1, y_train1, y_test1 = train_test_split(X, y,
test_size=0.3, random_state=0)
print ('Train set:', X_train1.shape, y_train1.shape)
print ('Test set:', X_test1.shape, y_test1.shape)
```

```
from sklearn.tree import DecisionTreeClassifier
```

membuat model Decision Tree

```
tree_model = DecisionTreeClassifier()
```

Melatih model dengan menggunakan data latih

```
tree_model = tree_model.fit(X_train1, y_train1)
```

5. Visualisasi Pohon Keputusan Pertama

```
from six import StringIO
import pydotplus
import matplotlib.image as mpimg
from sklearn import tree
```

```
%matplotlib inline
```

```
dot_data = StringIO()
filename = "/content/drive/MyDrive/Folder Skripsi/pohon1.png"
featureNames = ['JK', 'Jenis_Pekerjaan', 'Usia',
'Penghasilan', 'Riwayat_Kesehatan']
targetNames = ['Layak', 'Tidak Layak' ]
out = tree.export_graphviz(tree_model, feature_names=featureNames,
out_file=dot_data, class_names= np.unique(y_train1),
filled=True, special_characters=True, rotate=False)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png(filename)
img = mpimg.imread(filename)
plt.figure(figsize=(200, 100))
plt.imshow(img, interpolation='nearest')
```

```

#Evaluasi Model
from sklearn.metrics import accuracy_score

y_pred = tree_model.predict(X_test1)

acc_score = round(accuracy_score(y_pred, y_test1), 3)

print('Accuracy: ', acc_score)

from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# 6. Evaluasi Model
y_pred = tree_model.predict(X_test1)
labels=y
# Membuat confusion matrix
cm = confusion_matrix(y_test1, y_pred)

# Mencetak confusion matrix
labels = list(y.unique()) # Pastikan y adalah DataFrame atau Series yang
memiliki label kelas
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', xticklabels=labels,
yticklabels=labels, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

from sklearn.metrics import confusion_matrix, classification_report

# Menghitung precision, recall, f1-score, dan support
report = classification_report(y_test1, y_pred, target_names=labels)
print(report)

# 6. Pengoptimalan Model

from sklearn.ensemble import ExtraTreesClassifier
import matplotlib.pyplot as plt
model = ExtraTreesClassifier()
model.fit(X,y)

```



```

print(model.feature_importances_) #use inbuilt class feature_importances
of tree based classifiers

#plot graph of feature importances for better visualization
feat_importances = pd.Series(model.feature_importances_, index=X.columns)
feat_importances.nlargest(10).plot(kind='barh')
plt.show()

X = df[['Usia', 'Penghasilan', 'Riwayat_Kesehatan']]
y = df['Ket']

#Proses Penghapusan Atribut Yang tidak Penting
df.drop(['No.', 'Nama', 'JK', 'Jenis_Pekerjaan'], axis=1, inplace=True)
df.head()

# Membagi dataset menjadi data latih & data uji
from sklearn.model_selection import train_test_split
X_train2, X_test2, y_train2, y_test2 = train_test_split(X, y,
test_size=0.3, random_state=0)
print ('Train set:', X_train2.shape, y_train2.shape)
print ('Test set:', X_test2.shape, y_test2.shape)

from sklearn.tree import DecisionTreeClassifier

# membuat model Decision Tree
tree_model2 = DecisionTreeClassifier()

# Melatih model dengan menggunakan data latih
tree_model2 = tree_model.fit(X_train2, y_train2)

from six import StringIO
import pydotplus
import matplotlib.image as mpimg
from sklearn import tree

%matplotlib inline

dot_data = StringIO()
filename = "/content/drive/MyDrive/Folder Skripsi/pohon2.png"
featureNames = ['Usia', 'Penghasilan', 'Riwayat_Kesehatan']
targetNames = ['Layak', 'Tidak Layak']

```

```

out = tree.export_graphviz(tree_model2, precision=0,
feature_names=featureNames, out_file=dot_data, class_names=
np.unique(y_train2), filled=True, special_characters=True, rotate=False)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
graph.write_png(filename)
img = mpimg.imread(filename)
plt.figure(figsize=(200, 100))
plt.imshow(img, interpolation='nearest')

```

7. Mengukur Kedalaman Pohon

```

from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
# Mencoba berbagai nilai kedalaman pohon
for max_depth in range(1, 11):
    clf = DecisionTreeClassifier(max_depth=max_depth)
    clf.fit(X_train2, y_train2)
    y_pred = clf.predict(X_test2)
    accuracy = accuracy_score(y_test2, y_pred)
    print(f"Max Depth = {max_depth}, Accuracy = {accuracy:.2f}")

# Mencoba berbagai nilai minimum sampel untuk percabangan
for min_samples_split in range(2, 11):
    clf = DecisionTreeClassifier(min_samples_split=min_samples_split)
    clf.fit(X_train2, y_train2)
    y_pred = clf.predict(X_test2)
    accuracy = accuracy_score(y_test2, y_pred)
    print(f"Min Samples Split = {min_samples_split}, Accuracy =
{accuracy:.2f}")

```

membuat model Decision Tree

```

clf = DecisionTreeClassifier(max_depth=2, min_samples_split=2,
min_samples_leaf=5, criterion="entropy")
# Melatih model dengan menggunakan data latih
tree_model2 = clf.fit(X_train2, y_train2)

```

8. Visualisasi Model

```

# Evaluasi Model
from sklearn.metrics import accuracy_score
y_pred = tree_model2.predict(X_test2)
acc_score = round(accuracy_score(y_pred2, y_test), 3)
print('Accuracy : ', acc_score)

```

```

from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

# 6. Evaluasi Model 2
y_pred = tree_model2.predict(X_test2)
labels=y
# Membuat confusion matrix
cm = confusion_matrix(y_test2, y_pred)

# Mencetak confusion matrix
labels = list(y.unique()) # Pastikan y adalah DataFrame atau Series yang
memiliki label kelas
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt='d', xticklabels=labels,
yticklabels=labels, cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()

```

```

from sklearn.metrics import confusion_matrix, classification_report

```

```

# Menghitung precision, recall, f1-score, dan support
report = classification_report(y_test2, y_pred, target_names=labels)
print(report)

```

```

# 9. Pemodelan Pohon Keputusan Kedua

```

```

import os

# Where to save the figures
#content/drive/MyDrive/Folder Skripsi
PROJECT_ROOT_DIR = "."
CHAPTER_ID = "decision_trees"
IMAGES_PATH = os.path.join(PROJECT_ROOT_DIR, "Skrpsi", CHAPTER_ID)
os.makedirs(IMAGES_PATH, exist_ok=True)

def save_fig(tree_model, tight_layout=True, fig_extension="png",
resolution=300):
    path = os.path.join(IMAGES_PATH, tree_model2 + "." + fig_extension)
    print("Saving figure", tree_model2)
    if tight_layout:
        plt.tight_layout()
    plt.savefig(path, format=fig_extension, dpi=resolution)

```

```
from graphviz import Source
from sklearn.tree import export_graphviz
export_graphviz(
    tree_model2, precision = 0,
    out_file=os.path.join(IMAGES_PATH, "nasabah_tree7.dot"),
    feature_names = ['Usia', 'Penghasilan', 'Riwayat_Kesehatan'],
    class_names = ['Layak', 'Tidak Layak' ],
    rounded= True,
    filled = True)
Source.from_file(os.path.join(IMAGES_PATH, "nasabah_tree7.dot"))
```



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
LEMBAGA PENELITIAN

Kampus Unisan Gorontalo Lt.3 - Jln. Achmad Nadjamuddin No. 17 Kota Gorontalo
Telp: (0435) 8724466, 829975 E-Mail: lembagapenelitian@unisan.ac.id

Nomor : 4385/PIP/LEMLIT-UNISAN/GTO/XI/2022

Lampiran : -

Hal : Permohonan Izin Penelitian

Kepada Yth,

Kepala PT. Prudential Life Assurance GLI Gorontalo

di,-

Tempat

Yang bertanda tangan di bawah ini :

Nama : Dr. Rahmisyari, ST.,SE.,MM

NIDN : 0929117202

Jabatan : Ketua Lembaga Penelitian

Meminta kesediannya untuk memberikan izin pengambilan data dalam rangka penyusunan **Proposal / Skripsi**, kepada :

Nama Mahasiswa : Ferdiansyah R. Tangkilisan

NIM : T3119003

Fakultas : Fakultas Ilmu Komputer

Program Studi : Teknik Informatika

Lokasi Penelitian : PT. PRUDENTIAL LIFE ASSURANCE GLI GORONTALO

Judul Penelitian : KLASIFIKASI CALON NASABAH BARU PEMBERIAN
PERTANGGUNGAN ASURANSI MENGGUNAKAN
ALGORITMA DECISION TREE C4.5 (STUDI KASUS PADA
PT. PRUDENTIAL LIFE ASSURANCE GLI GORONTALO)

Atas kebijakan dan kerja samanya diucapkan banyak terima kasih.



Gorontalo, 07 November 2022
Ketua

Dr. Rahmisyari, ST.,SE.,MM
NIDN 0929117202

Gorontalo, 2023

Kepada Yth,
Ketua Program Studi
Fakultas Ilmu Komputer
Universitas Ichsan Gorontalo
Di Tempat

SURAT KETERANGAN

No. : 291/BlessingGTO/Prudential/VIII/2023

Dengan ini Menerangkan bahwa :

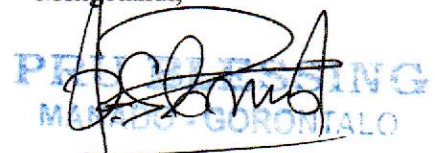
Nama : Ferdiansyah R. Tangkilisan
NIM : T3119003
Program Studi : Teknik Informatika – Universitas Ichsan Gorontalo
Jenjang : S1

Telah kami setuju untuk melaksanakan penelitian di instansi kami PT. Prudential Life Assurance sebagai persyaratan penyusunan skripsi dengan judul **“KLASIFIKASI CALON NASABAH BARU PEMBERIAN PERTANGGUNGAN ASURANSI MENGGUNAKAN ALGORITMA DECISION TREE C4.5”** yang dimulai pada tanggal 27 November 2022 s/d selesai.

Demikian surat ini kami ajukan untuk digunakan seperlunya, atas perhatian dan kerjasamanya, PT. Prudential Life Assurance mengucapkan terima kasih.

Gorontalo, 10 November 2023

Mengetahui,



SABRIA SY. USMAN

Associate Agency Director



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO
FAKULTAS ILMU KOMPUTER

SURAT KEPUTUSAN MENDIKNAS RI NOMOR 84/D/O/2001

Jl. Achmad Najamuddin No. 17 Telp. (0435) 829975 Fax (0435) 829976 Gorontalo

SURAT REKOMENDASI BEBAS PLAGIASI

No. 463 /FIKOM-UIG/R/XI/2023

Yang bertanda tangan di bawah ini :

Nama : Irvan Abraham Salihi, M.Kom
NIDN : 0928028101
Jabatan : Dekan Fakultas Ilmu Komputer

Dengan ini menerangkan bahwa :

Nama Mahasiswa : Ferdiansyah Tangkilisan
NIM : T3119003
Program Studi : Teknik Informatika (S1)
Fakultas : Fakultas Ilmu Komputer
Judul Skripsi : Klasifikasi Data Calon Nasabah Baru Pemberian
Pertanggungan Asuransi Menggunakan Algoritma
Decision Tree C4.5

Sesuai hasil pengecekan tingkat kemiripan skripsi melalui aplikasi **Turnitin** untuk judul skripsi di atas diperoleh hasil *Similarity* sebesar **23%**, berdasarkan Peraturan Rektor No. 32 Tahun 2019 tentang Pendeteksian Plagiat pada Setiap Karya Ilmiah di Lingkungan Universitas Ichsan Gorontalo dan persyaratan pemberian surat rekomendasi verifikasi calon wisudawan dari LLDIKTI Wil. XVI, bahwa batas kemiripan skripsi maksimal 30%, untuk itu skripsi tersebut di atas dinyatakan **BEBAS PLAGIASI** dan layak untuk diujikan.

Demikian surat rekomendasi ini dibuat untuk digunakan sebagaimana mestinya.

Mengetahui
Dekan,

Irvan Abraham Salihi, M.Kom
NIDN. 0928028101

Gorontalo, 13 November 2023
Tim Verifikasi,

Zulfrianto Y. Lamasigi, M.Kom
NIDN. 0914089101

Terlampir :
Hasil Pengecekan Turnitin

PAPER NAME

SKRIPSI_T3119003_FERDIANSYAH R. T
ANGKILISAN.pdf

AUTHOR

Ferdiansyah R. Tangkilisan ferdyt64@gmail.com

WORD COUNT

13104 Words

CHARACTER COUNT

79598 Characters

PAGE COUNT

85 Pages

FILE SIZE

1.6MB

SUBMISSION DATE

Nov 11, 2023 4:15 PM GMT+8

REPORT DATE

Nov 11, 2023 4:17 PM GMT+8

● **23% Overall Similarity**

The combined total of all matches, including overlapping sources, for each database.

- 23% Internet database
- 9% Publications database
- Crossref database
- Crossref Posted Content database
- 2% Submitted Works database

● **Excluded from Similarity Report**

- Bibliographic material
- Quoted material
- Cited material
- Small Matches (Less than 15 words)



**KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
UNIVERSITAS ICHSAN GORONTALO**

FAKULTAS ILMU KOMPUTER

UPT. PERPUSTAKAAN FAKULTAS

SK. MENDIKNAS RI NO. 84/D/0/2001

Jl. Achmad Nadjamuddin No.17 Telp(0435) 829975 Fax. (0435) 829976 Gorontalo

SURAT KETERANGAN BEBAS PUSTAKA

No : 013/Perpustakaan-Fikom/XI/2023

Perpustakaan Fakultas Ilmu Komputer (FIKOM) Universitas Ichsan Gorontalo dengan ini menerangkan bahwa :

Nama Anggota : Ferdiansyah R. Tangkilisan

No. Induk : T3119003

No. Anggota : M202375

Terhitung mulai hari, tanggal : Jumat, 10 November 2023, dinyatakan telah bebas pinjam buku dan koleksi perpustakaan lainnya.

Demikian keterangan ini di buat untuk di pergunakan sebagaimana mestinya.



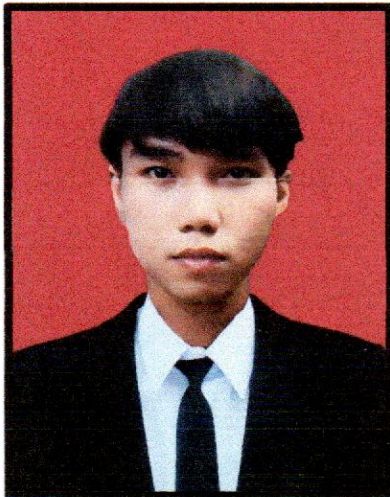
Gorontalo, 10 November 2023

**Mengetahui,
Kepala Perpustakaan**

Apriyanto Alhamad, M.Kom

NIDN : 0924048601

RIWAYAT HIDUP PENELITI



Nama	: Ferdiansyah R. Tangkilisan
NIM	: T3119003
Tempat, Tgl Lahir	: Gorontalo, 28 November 2001
Agama	: Islam
Jenis Kelamin	: Laki-Laki
Fakultas/Jurusan	: Ilmu Komputer / Teknik Informatika
Konsentrasi	: Software Engineering
Alamat	: Jl. Cendrawasih No. 10

Riwayat Pendidikan :

1. Tahun 2013, Menyelesaikan Pendidikan di SD Negeri 67, Kecamatan Kota Timur, Kota Gorontalo, Provinsi Gorontalo.
2. Tahun 2016, Menyelesaikan Pendidikan di SMP Negeri 06 Gorontalo, Kecamatan Kota Selatan, Kota Gorontalo, Provinsi Gorontalo.
3. Tahun 2019, Menyelesaikan Pendidikan di SMK Negeri 01 Gorontalo Kecamatan Sipatana, Kota Gorontalo, Provinsi Gorontalo.
4. Tahun 2019, Telah di terima menjadi Mahasiswa di Perguruan Tinggi Swasta Universitas Ichsan Gorontalo.