ALGORITMA KLASIFIKASI *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM) DAN *CLASSIFICATION AND REGRESSION TREES* (CART) UNTUK MENENTUKAN FAKTOR YANG MEMPENGARUHI PEMBAYARAN KREDIT

TESIS

Oleh :

Imas Wihdah Misshuari

NIM : 24010119410008

PROGRAM MAGISTER MATEMATIKA

FAKULTAS SAINS & MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2021

ALGORITMA KLASIFIKASI *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM) DAN *CLASSIFICATION AND REGRESSION TREES* (CART) UNTUK MENENTUKAN FAKTOR YANG MEMPENGARUHI PEMBAYARAN KREDIT

Imas Wihdah Misshuari

24010119410008

Tesis

Diajukan sebagai syarat untuk memperoleh gelar Magister Sains pada Program Studi Magister Matematika

DEPARTEMEN MATEMATIKA

FAKULTAS SAINS DAN MATEMATIKA

UNIVERSITAS DIPONEGORO

SEMARANG

2021

ALGORITMA KLASIFIKASI *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM) DAN *CLASSIFICATION AND REGRESSION TREES* (CART) UNTUK MENENTUKAN FAKTOR YANG MEMPENGARUHI PEMBAYARAN KREDIT

Oleh

Imas Wihdah Misshuari

NIM:24010119410008

(Program Magister Matematika Fakultas Sains & Matematika)

UNIVERSITAS DIPONEGORO SEMARANG

Menyetujui:

Tim Pembimbing

Tanggal 18 Juli 2021

Pembimbing I	Pembimbing II
	
Ratna Herdiana, M.Sc, PhD	**Farikhin, S.Si, M.Si, PhD**
NIP. H.7.196411242019092001	NIP. 197312202000121001

PEDOMAN PENGGUNAAN TESIS

Tesis S2 yang tidak dipublishkan terdaftar dan tersedia di Perpustakaan Fakultas Sains & Matematika Universitas Diponegoro Semarang, dan terbuka untuk umum dengan ketentuan bahwa hak cipta ada pada pengarang dengan mengikuti aturan HaKI yang berlaku di Fakultas Sains & Matematika Universitas Diponegoro Semarang. Referensi kepustakaan diperkenankan dicatat, tetapi pengutipan atau peringkasan hanya dapat dilakukan seizin pengarang dan harus disertai dengan kaidah ilmiah untuk menyebutkan sumbernya.

Sitasi hasil penelitian Tesis ini dapat ditulis dalam bahasa Indonesia sebagai berikut :

Misshuari, I.,W.(2021): *Algoritma Klasifikasi Light Gradient Boosting Machine (LightGBM) dan Classification and Regression Tree (CART) untuk Menentukkan Faktor yang Mempengaruhi Pembayaran Kredit* , Tesis Program Magister, Fakultas Sains & Matematika Universitas Diponegoro Semarang.

dan dalam bahasa Inggris sebagai berikut :

Misshuari, I.,W.(2021): *Classification Algorithm of Light Gradient Boosting Machine (LightGBM) and Classification and Regression Tree (CART) to Determine Factors Affecting Credit Payments*, Master's Program Thesis, Faculty of Science & Mathematics, University of Diponegoro Semarang.

Memperbanyak atau menerbitkan sebagian atau seluruh tesis haruslah seizin Dekan Fakultas Sains & Matematika Universitas Diponegoro Semarang.

Dipersembahkan kepada Ayahanda Mesiono, Ibunda Suridah, Dinda dan Dhawi

ABSTRAK
ALGORITMA KLASIFIKASI *LIGHT GRADIENT BOOSTING MACHINE* (LIGHTGBM) DAN *CLASSIFICATION AND REGRESSION TREES* (CART) UNTUK MENENTUKAN FAKTOR YANG MEMPENGARUHI PEMBAYARAN KREDIT

Oleh:
Imas Wihdah Misshuari
NIM : 24010119410008

Program Magister Matematika Fakultas Sains & Matematika

Machine learning adalah metode analisis data yang mengotomatiskan pembuatan model analitik. [illegible] memiliki dua tujuan; pertama, untuk mengklasifikasikan data berdasarkan model yang dikembangkan dan kemudian membuat prediksi untuk hasil masa depan berdasarkan model tersebut. Tesis ini mengkaji dua metode *Machine learning* yaitu *Light Gradient Boosting Machine* (LightGBM) dan *Classification and Regression Tree* (CART). LightGBM adalah kerangka peningkatan gradien yang menggunakan algoritma pembelajaran berbasis pohon. Ini adalah pengembangan dari *Gradient Boosting Decision Tree* (GBDT) - efisiensi, akurasi, dan interpretasi dari algoritma pembelajaran mesin yang banyak digunakan yang mampu menangani data yang tidak seimbang. CART adalah metode atau algoritma dari teknik pohon keputusan. Pandemi COVID-19 mempengaruhi seluruh segmen dunia, termasuk sektor perbankan dan kredit. Permasalahan yang dihadapi oleh perbankan khususnya Bank Perkreditan Rakyat (BPRS) Syariah MEDAN adalah masih melakukan pengecekan data kredit secara manual dengan status lancar dan tidak lancar. Sehingga menjadi tidak efisien dan kemungkinan terjadi kesalahan. Oleh karena itu perlu dilakukan pengklasifikasian data secara otomatis dan mencari faktor-faktor yang mempengaruhi pembayaran kredit nasabah. Tesis ini menerapkan dua klasifikasi masalah faktor yang mempengaruhi pembayaran kredit pada data debitur pada PT. BPRS Gebu Prima Medan tiga tahun 2018 – 2020 berisi debitur kredit lancar dan kredit macet. Tesis ini juga membahas akurasi metode LightGBM dan CART. Berdasarkan evaluasi yang dilakukan terhadap klasifikasi faktor-faktor yang mempengaruhi pembayaran kredit, metode CART. Hasil penelitian menemukan bahwa faktor-faktor yang mempengaruhi adalah pendapatan total maksimal Rp 27.750.000 dengan pagu lebih dari Rp 57.500.000 diikuti anggota keluarga maksimal. 3.5; sedangkan dengan metode LightGBM, faktornya adalah plafon, total pendapatan, anggota keluarga, usia, dan jenis kelamin dengan nilai kepentingan masing-masing 65200, 65100, 13000, 9800, dan 4200. Namun, metode CART memiliki tingkat akurasi yang lebih tinggi yaitu 85,9% dibandingkan dengan metode LightGBM, 81%.

Kata Kunci : *Machine Learning, Light Gradient Boosting Machine* (LightGBM) dan *Classification and Regression Tree* (CART), pembayaran kredit

ABTRACT

Classification Algorithm of Light Gradient Boosting Machine (LightGBM) and Classification and Regression Tree (CART) to Determine Factors Affecting Credit Payments

By :

Imas Wihdah Misshuari

NIM : 24010119410008

Master's Program Faculty of Science & Mathematics

Machine learning is a data analysis method that automates the creation of analytic models. It has two objectives; first, to classify data based on a model developed and then make predictions for future results based on that model. Machine learning has several methods, including Light Gradient Boosting Machine (LightGBM) and Classification and Regression Tree (CART). LightGBM is a gradient enhancement framework that uses a tree-based learning algorithm. It's an extension of the Gradient Boosting Decision Tree (GBDT) - efficiency, accuracy, and interpretability of widely used machine learning algorithms capable of handling unbalanced data. CART is a method or algorithm of the decision tree technique. The COVID-19 pandemic affects whole segments of the world, including the banking and credit sectors. The banking and credit sector plays an essential role in the Indonesian economy structure due to the function as a collector and channel of funds by creating products offered to people who need to use the credit services. The problem faced by banks, especially the MEDAN Syariah People's Credit Bank (BPRS), is that they are still manually checking credit data with current and non-current status. So that it becomes inefficient and the possibility of errors can occur. Therefore, it is necessary to automatically classify data and look for factors that affect customer credit payments. This tesis applies these two classification problems of factors affecting credit payments in debtor data at PT. BPRS Gebu Prima Medan of three years 2018 – 2020 contains current credit debtors and lousy credit. This thesis also discusses the accuracy of the LightGBM and CART methods. Based on the evaluation conducted on the classification of factors that affect credit payments, the CART method. The study found the factors that influence are maximum total income of IDR 27,750,000 with a ceiling of more than IDR 57,500,000 followed by maximum family members. 3.5; while by LightGBM method, the factors are ceiling, total income, family members, age, and gender with importance values of 65200, 65100, 13000, 9800, and 4200, respectively. However, the CART method has a higher accuracy rate of 85.9% than the LightGBM method, 81%.

Keywords : Machine Learning, Light Gradient Boosting Machine (LightGBM) and Classification and Regression Tree (CART), credit payment

KATA PENGANTAR

Puji syukur kepada Allah SWT yang Maha Pengasih lagi Maha Penyayang atas segala rahmat dan karunianya, sehingga penulis dapat menyelesaikan penyusunan proposal tesis yang berjudul **"**Algoritma Klasifikasi *Light Gradient Boosting Machine* (LightGBM) dan *Classification and Regression Tree* (CART) untuk Menentukkan Faktor yang Mempengaruhi Pembayaran Kredit**"**.

Dalam penyusunan Proposal Tesis ini banyak pihak yang telah membantu, maka tidak lupa penulis menyampaikan rasa hormat dan mengucapan terima kasih kepada:

1. Bapak Dr. Redemtus Heru Tjahyana S.Si, M.Si selaku ketua program studi Magister Matematika yang telah memberi ijin pembuatan proposal tesis ini.
2. Ibu Ratna Herdiana, M.Sc, PhD selaku dosen pembimbing I yang telah meluangkan waktu memberikan bimbingan dan pengarahan.
3. Bapak Farikhin, S.Si, M.Si, Ph. D. selaku dosen pembimbing II yang telah meluangkan waktu memberikan bimbingan dan pengarahan.
4. Bapak Drs. Bayu Surarso, Ph.D. selaku dosen penguji I yang telah banyak memberikan masukan dan sara dalam penyusunan tesis ini.
5. Ibu Indri Pratiwi,SE, M.Si selaku direkur utama dan pegawai PT BPR Syariah Gebu Prima Medan, yang memberikan izin dan tempat kepada penulis untuk melakukan penelitian.
6. Seluruh dosen dan staf pegawai administrasi Jurusan Magister Matematika FSM Universitas Diponegoro yang telah membantu penulis menyelesaikan tesis ini dan memberikan bimbingan kepada penulis semenjak mengikuti perkuliahan.
7. Teristimewa kepada Ayahanda terkasih Dr. Mesiono, M.Pd dan ibunda tercinta Suridah, S.PdI untuk semua kasih sayang, doa , ajaran, motivasi, dan jerih payah sehingga penulis dapat menyelesaikan studi.
8. Kepada adik-adik Dinda Hafsah Misshuari dan Ahmad Qordhawi Misshuari yang memberikan dukungan kepada penulis.
9. Kepada teman-teman Nurul Ayu Pratiwi, Hanifah Aisyah dan keluarga besar Magister Matematika 2019 yang tidak dapat penulis sebutkan satu persatu yang

telah memberikan dukungan, doa, semangat, saran, dan membantu proses persiapan dalam menyelesaikan tesis ini.

10. Semua pihak yang ikut membantu hingga selesainya penyusunan proposal tesis ini, yang tidak dapat penulis sebutkan satu per satu. Semoga Allah SWT membalas segala kebaikan yang telah diberikan.

Penulis menyadari bahwa dalam proposal tesis ini masih terdapat banyak kekurangan, baik pada redaksi penulisan maupun isi dan masih jauh dari kata sempurna. Oleh sebab itu, penulis mengharapkan kritik dan saran yang membangun guna penyempurnaan proposal tesis ini.

Semarang, 2021

Imas Wihdah Misshuari

DAFTAR ISI

DAFTAR GAMBAR DAN ILUSTRASI

DAFTAR TABEL

DAFTAR SINGKATAN DAN LAMBANG

SINGKATAN	Nama	Pemakaian pertama kali pada halaman
Min	Minimum	26
Max	Maximum	23
LAMBANG		
$\sum_{i}^{m}$	Jumlah seluruh i ke m	12
∞	Tak Berhingga	22
$\leq$	Kurang dari atau sama dengan	16
$<$	Kurang dari	20
$>$	Lebih dari	16
$\approx$	isomorfik	29
∂	Turunan parsial	29
∇	Gradien	29

BAB I

Pendahuluan

1.1 Latar Belakang

Machine learning adalah salah satu cabang dari *Artifical Intelligence* (AI) dan ilmu komputer yang berfokus untuk membuat sistem atau algoritma dengan menggunakan data dan meningkatkan akurasinya dari waktu ke waktu. Algortima ini memiliki dua tujuan yaitu untuk mengklasifikasikan data berdasarkan model yang telah dikembangkan dan membuat prediksi untuk hasil masa depan berdasarkan model tersebut. *Machine learning* adalah [illegible]ek yang mempelajari statistik, probabilitas, pohon keputusan dan lain – lain . [illegible] nurut Prabawati dkk (2019) , Hartati dkk (2012) dan Dinata dkk(2018), Pohon keputusan sangat sering digunakan dalam menyelesaikan suatu masalah , mudah diinterpretasikan dan divisualisasikan. Hasil dari pohon keputusan tersebut dapat digunakan sebagai memprediksi atau membantu pengambilan suatu keputusan dalam memecahkan suatu masalah. Beberapa algoritma *machine learning* yang telah dikembangkan berdasarkan pohon keputusan adalah CHAID (*Chi Squared Automatic Interaction Detection*) yang mana split tiap *node* berdasarkan pada *Chi Squared test* pada masing masing atribut , C4.5 yang menggunakan *gain ratio* untuk kinerja splitnya, ID3 (*Iterative Dichotomiser 3*) yang menggunakan entropy untuk kriteria splitnya, CART (*Classification And Regression Tree*), XGBoost dan LightGBM (*Light Gradient Boosting Machine*) membentuk pohon keputusan dengan perhitungan *gini index* untuk kriteria split (Maulana dkk., 2015).

Dunia sedang dilanda penyakit akibat serangan virus SARS- Cov 2 (*Severe Acute Respiratory Syndrome Coronavirus 2*) atau yang lebih dikenal dengan istilah Virus Covid 19 (*Corona Virus Disease 2019*) . Adanya pandemi Covid – 19 mempengaruhi segala sektor perekonomian di dunia. Selama penyakit ini menyerang, perekonomian dunia mengalami kerugian yang luar biasa diantaranya karena banyak perdagangan jual beli yang dihentikan sehingga tidak ada pemasukan bahkan kegiatan ekonomi lainnya. Dampak dari hal ini akan meluas ke sektor perbankan dan penkreditan. Sektor perbankan menjadi salah satu faktor yang memegang peranan penting dalam tatanan perekonomian di Indonesia karena berfungsi sebagai penghimpun dan penyalur dana melalui penciptaan

produk yang beraneka ragam untuk ditawarkan kepada masyarakat yang ingin menggunakan jasa pengkreditan seperti pengkreditan kendaraan.

Pada penelitian Budi dkk. (2010), Mewoh dkk (2016), Alexandri dkk (2020), dan Bank Pengkreditan Rakyat Syariah (BPRS) Gebu Prima Medan memiliki permasalahan pada kredit macet dengan salah satu faktornya adalah penginputan dan pengecekan data debitur dilakukan oleh operator pada bank tersebut . Oleh karna itu diperlukan suatu metode untuk mengklasifikasikan data secara otomatis , agar pihak bank dalam pengecekan data tidak terjadi kesalahan dan mampu menemukkan faktor – faktor yang mempengaruhi pembayaran kredit oleh nasabah. Untuk menentukkan faktor – faktor yang mempengaruhi pembayaran kredit oleh nasabah pada BPRS Gebu Prima MEDAN dapat diselesaikan dengan menggunakan algoritma *Decision Tree* atau Pohon Keputusan.

Classification and Regresion Trees (CART) merupakan metode atau algoritma dari salah satu teknik pohon keputusan yang dikembangkan oleh Leo Breimann, Jerome H. Freidman, Richard A. Olshen dan Charles J. Stone sekitar tahun 1980-an. Penerapan CART untuk melakukan analisis klasifikasi data statistik telah banyak digunakan dalam berbagai studi kasus . Tujuan CART adalah untuk mendapatkan pengklasifikasian suatu kelompok yang paling akurat. (Breimann dkk., 1984).

Menurut Goulin dkk (2017), *Light Gradient Boosting Machine* (LightGBM) adalah algoritma *gradient boosting* yang cepat, terdistribusi dan berperforma tinggi berdasarkan pohon keputusan yang digunakan untuk memberi peringkat, klasifikasi dan banyak tugas *machine learning* lainnya.

Alasan penelitian ini menggunakan metode CART adalah karena hasilnya lebih sederhana untuk dipahami dan ditafsirkan, menggunakan data yang mempunyai jumlah yang besar, dan mampu mengolah data dengan variabel kategorik maupun kontinu. Selain itu juga didukung oleh beberapa penelitian-penelitian sebelumnya yang menggunakan CART. Rezeki Handayani Tanjung (2017) telah melakukan penelitian dengan judul "Penerapan Metode CART Untuk Menentukan Faktor - Faktor Yang Mempengaruhi Pembayaran Kredit Oleh Nasabah" . Variabel-variabel yang digunakan adalah jenis kelamin, usia, anggota keluarga, jumlah penghasilan, plafond, dan lama pinjaman sebagai variabel

independen (X) dan status kredit sebagai variabel dependen (Y). Hasil dari penelitian menunjukkan bahwa status pembayaran kredit dipengaruhi oleh jumlah penghasilan, usia dan plafond dengan ketepatan hasil klasifikasi yang terbentuk sebesar 84,2%. Karakteristik nasabah yang memiliki status pembayaran kredit cenderung tidak lancar adalah nasabah yang mempunyai jumlah penghasilan maksimum Rp 4.550.000, dengan plafond maksimum Rp 17.500.000 usia maksimum 39,5 tahun dengan persentase 20,5%, nasabah dengan jumlah penghasilan lebih dari Rp 4.550.000 dengan persentase 18,1%, serta nasabah dengan jumlah penghasilan maksimum Rp 4.550.000 dengan plafond lebih dari Rp 17.500.000 persentase 1,4%.

Berdasarkan penelitian terdahulu yang menggunakan metode LightGBM. Marcos Roberto Machado dkk (2019) telah melakukan penilitian dengan judul "*Light GBM : an Effective Decision Tree Gradient Boosting Method to predict Customer Loyalty in The Finance Industry*". Variabel yang digunakan adalah tanggal, jumlah pembelian, jumlah angsuran, kota dan negara bagian yang teridentifikasi berdagang, identifikasi pedagang, penjualan rata – rata , jumlah bulan dimana kartu tidak aktif, kota, negara bagian lokasi pedagang dan skor loyalitas. Hasil dari penilitian menunjukkan bahwa LightGBM memiliki akurasi yang lebih baik daripada XGBoosting. Selain itu, ada beberapa penelitian yang menggunakan *Machine Learning*. Qiangqiang Guo (2019) telah melakukan penelitian dengan judul "*Credit Risk Scoring Analysis Based on Machine Learning Models*". Hasil dari penelitian menunjukkan bahwa model LigthGBM lebih baik daripada model *Logit Regression* dan *Random Forest* dengan skor AUC sebesar 78%.

Dalam penelitian ini, jumlah penghasilan, umur, jumlah anggota keluarga, jenis kelamin dan plafond digunakan sebagai variabel - variabel yang mempengaruhi pembayaran kredit. Variabel jumlah penghasilan digunakan karena besarnya jumlah penghasilan merupakan syarat penting dalam pengajuan kredit. Variabel umur digunakan karena yang memiliki nilai produktifitas yang baik ditunjukkan kepada usia yang relatif lebih muda diabndingkan yang tua.Variabel jumlah anggota keluarga digunakan karena jika semakin banyak anggota keluarga yang ditanggung oleh pihak pengajuan kredit maka semakin banyak

pengeluaran dalam memenuhi kebutuhan hidup dan berkurangnya penghasilan yang seharusnya digunakan untuk memenuhi pembayaran kredit. Variabel jenis kelamin digunakan karena yang memiliki loyalitas tinggi dan lebih mampu dalam menjaga kepercayaan yang diberikan oleh pihak bank dalam pembayaran kredit secara tepat waktu sebelum jatuh tempo adalah jenis kelamin perempuan dibandingkan dengan jenis kelamin laki-laki. Variabel plafon digunakan karena plafon adalah batasan biaya tertinggi pemakaian kredit yang dikeluarkan oleh sebuah bank atau koperasi. Dimana, fasilitas plafon yang diberikan tersebut merupakan jumlah total kredit yang diberikan oleh pihak bank.

Berdasarkan uraian diatas penulis tertarik untuk melakukan kajian tentang metode algoritma klasifikasi *Classification and Regression Trees* (CART) dan *Light Gradient Boosting Machine* (LightGBM) , dan penerapannya di bidang perbankan yaitu dalam menentukan faktor - faktor yang mempengaruhi pembayaran kredit oleh nasabah, untuk itu tesis ini diberi judul "Algoritma Klasifikasi *Light Gradient Boosting Machine* (LightGBM) dan *Classification and Regression Tree* (CART) untuk Menentukkan Faktor yang Mempengaruhi Pembayaran Kredit".

1.2 Rumusan Masalah

Berdasarkan latar belakang masalah maka rumusan masalah dari penelitian ini adalah:

1. Faktor – faktor apa saja yang paling mempengaruhi pembayaran kredit pada nasabah dengan menggunakan algoritma *Classification And Regression Trees* (CART) di BPRS Gebu Prima Medan
2. Faktor – faktor apa saja yang paling mempengaruhi pembayaran kredit pada nasabah dengan menggunakan algoritma *Light Gradient Boosting Machine* (LightGBM) di BPRS Gebu Prima Medan
3. Bagaimana hasil perbandingan antara *Classification And Regression Trees* (CART) dan *Light Gradient Boosting Machine* (Light GBM) terhadap pembayaran kredit oleh nasabah di BPRS Gebu Prima Medan

1.3 Batasan Masalah

Adapun batasan masalah dalam peneliti ini adalah menggunakan metode *Classification and Regression Trees* (CART) dan *Light Gradient Boosting*

Machine (LightGBM) , variabel yang digunakan adalah jenis kelamin, usia, jumlah tanggungan, jumlah penghasilan dan plafon dan menggunakan data sekunder yaitu data pembayaran kredit nasabah oleh PT BPRS MEDAN 2018-2020.

1.4 Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka yang menjadi tujuan penelitian ini adalah:

1. Memperoleh faktor – faktor yang mempengaruhi pembayaran kredit menggunakan algoritma *Classification And Regression Trees* (CART).
2. Memperoleh faktor – faktor yang mempengaruhi pembayaran kredit menggunakan algoritma *Light Gradient Boosting Machine* (LightGBM).
3. Mendapatkan hasil perbandingan antara *Classification And Regression Trees* (CART) dan *Light Gradient Boosting Machine* (Light GBM) terhadap pembayaran kredit oleh nasabah di BPRS Gebu Prima Medan.

1.5 Kontribusi Penelitian

Keberhasilan dalam penelitian ini secara umum akan memberikan kontribusi dalam kemajuan perkembangan ilmu dan teknologi khususnya kajian ilmu-ilmu dasar dan terapannya, kemudia secara khusus adalah untuk menerapkan matematika dalam dunia sektor perkenomian salahsatunya adalah pengkreditan. ıfaat yang diberikan adalah berupa suatu gambaran tentang pemberian hak kredit pada nasabah di BPR Syariah Gebu Prima Medan pada masa yang akan datang. Sehingga dapat menentukan langkah, kebijakan dan perencanaan yang terbaik dalam mengurangi pembayaran kredit macet.

BAB II

Tinjauan Pustaka dan Dasar Teori

2.1 Tinjauan Pustaka

Machine Learning dapat digunakan untuk menentukkan faktor-faktor yang mempengaruhi pembayaran kredit agar dapat mengurangi debitur yang melakukan kredit macet. Salah satu metode *Machine lerning* adalah metode pohon keputusan atau *decision tree.* Metode ini adalah suatu model klasifikasi yang paling populer karena mudah diintrepetasikan. Machine learning diantaranya adalah *Light Gradient Boosting Machine* (LightGBM) dan *Classification and Regression Tree* (CART).

Penelitian *Light Gradient Boosting Machine* dan *Classification And Regression Tree* terdahulu dapat dilihat pada tabel :

Tabel II.1 Penelitian LIGHTGBM dan CART

No	Peneliti	Judul	Pembahasan
1.	Debaditya Chakraborty, Hosman Elhegazy, Hazem Elzarka and Lilianna Gutierrez (Chakraborty , Elhegazy, Elzarka, and Gutierrez, 2020)	*A novel construction cost prediction model using hybrid natural and light gradient boosting*	Penerapan LightGBM untuk memprediksi biaya konstruksi dan menghasilkan akurasi sebesar 87%
2.	Abha Parihar, Suraj Bhoste, Shubham Patil, Shreya Kaptage and Sanjeev Wagh (Parihar, Patil, Kaptage and Wagh, (2020)	*Type 2 diabetes prediction using LightGBM*	Penerapan *LightGBM* untuk memprediksi diabetes tipe 2 dan menghasilkan akurasi sebesar 88% lebih tinggi

			dibandingkan SVM dan RF.
3	Qiangqiang Guo, Zhenfang Zhu , Hongli Pei, Fuyong Xu, Qiang Lu, Dianyuan Zhang and Wenqing Wu (Guo, Zhu, Pei, Xu, Lu, Zhang, and Wu , 2019)	*Mobile user prediction based on LightGBM*	Penerapan LightGBM untuk memprediksi skor kredit pengguna china phone dan menghasilkan akurasi sebesar 66,97%
4	Elena Adriana Minastireanu dan Gabriela Mesnita (Minastireanu and Mesnita , 2019)	*Light GBM Machine Learning algorithm to online click fraud detection*	Penerapan Light GBM untuk menangani penipuan click yang lebih canggih dan menghasilkan akurasi sebesar 98%.
5	Marcos Roberto Machado, Salma Karray dan Ivaldo Tributino de Sousa (Machado, Karray and Sousa , 2019)	*Light GBM : an Effective Decision Tree Gradient Boosting Method to predict costumer loyalty in the finance industry.*	Penerapan LightGBM untuk memprediksi skor loyalitas pelanggan kartu kredit dan menghasilkan akurasi sebesar 90%.
6.	Rezeki handayani Tanjung dan Kartiko (Tanjung dan Kartiko 2017)	Penerapan metode CART untuk menentukan faktor yang mempengaruhi pembayaran kredit pada nasabah	Penerapan CART untuk menentukan faktor yang mempengaruhi pada embayaran

			kredit dan menghasilkan akurasi sebesar 84,20%
7.	Ling Jing Kao, Chih Chou Chiu dan Fon Yu Chiu (Kao, Chiu dan Chiu, 2012)	*A bayesian latent variabel model with classification and regression tree* for behavior and credit scoring.	Penerapan CART untuk meningkatkan efesiensi dalam memberi keputusan pemberian kredit dan menghasilkan akurasi lebih dari 90% dibandingkankan dengan SVM, MARS, BNP,dan *Logistic*.

2.2 Dasar Teori

2.2.1 Pembelajaran Mesin (*Machine Learning*)

Pembelajaran Mesin atau *Machine Learning* adalah aplikasi atau bagian dari kecerdasan buatan yang membuat sistem memiliki kemampuan belajar secara otomatis dan meningkatkan kemampuannya berdasarkan pengalaman tanpa diprogram secara eksplisit. *Machine learning* juga dapat didefiniskan sebagai algoritma yang bertujuan untuk menemukan dan mengaplikasikan pola-pola di dalam data. Beberapa aplikasi pengenalan pola adalah pengenalan suara, ramalan cuaca, deteksi objek dalam gambar dan lain- lain. Algortima pada *machine learning* menggunakan teknik-teknik statistik untuk menemukan pola-pola tersebut. Seringkali data yang dicari polanya berukuran besar. Ada dua aplikasi utama pada *machine learning* yaitu klasifikasi dan prediksi. Menurut Han dan Kamber (2007) klasifikasi dan prediksi adalah dua bentuk dari analisis data yang

dapat digunakan untuk mengekstrak model yang mendeskripsikan kelas data yang penting atau untuk memprediksikan tren data di masa depan. Klasifikasi adalah metode dalam machine learning yang digunakan oleh mesin untuk memilah atau mengkalsifikasikan obyek berdasarkan ciri tertentu. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka hasil dari suatu data berdasarkan data yang sudah dipelajari.

2.2.2 Tipe – Tipe Algoritma *Machine Learning*

Menurut Nurhayati dkk (2019) ,Secara garis besar ada tiga tipe algoritma *machine learning* yaitu

1. *Supervised Learning*

Supervised Learning merupakan algoritma yang terdiri dari input dan output yang diberikan oleh data training dan akan mencari pola sebagai acuan untuk data berikutnya menggunakan *set* data. Algoritma ini terdiri dari dua jenis yaitu klasifikasi dan prediksi. Klasifikasi adalah algoritma yang menggunakan data dengan target , kelas atau label berupa nilai kategorikal(nominal). Algoritma klasifikasi diantaranya adalah *Logistik Regression, Decision Trees, Random Forest, KNN, SVM, Neural Networks, Naïve Bayes*, dll. Sedangkan Prediksi adalah algoritma *forecasting* atau algoritma estimasi dimana target, kelas atau label berupa numerik dan menggunakan data rentet waktu (*time series).* Algoritma prediksi diantaranya adalah *Linear Regression, Decision Trees, Neural Networks, SVM, Trees*, dll.

2. *Unsupervised Learning*

Unsupervised Learning berbeda dengan algoritma sebelumnya. Algoritma ini bukan bersifat prediktif tetapi bersifat deskriptif sehingga tidak menggunakan *training data* set dan akan menghasilkan pengelompokan atau mengkategorikan data tersebu. Algoritma ini terdiri dari beberapa jenis algoritma diantaranya yaitu *klustering* dan *association.* Algoritma *klustering* diantaranya adalah *K-Means Clustering, Hierarchical Clustering.* Sedangkan algoritma *association* adalah *association rules.*

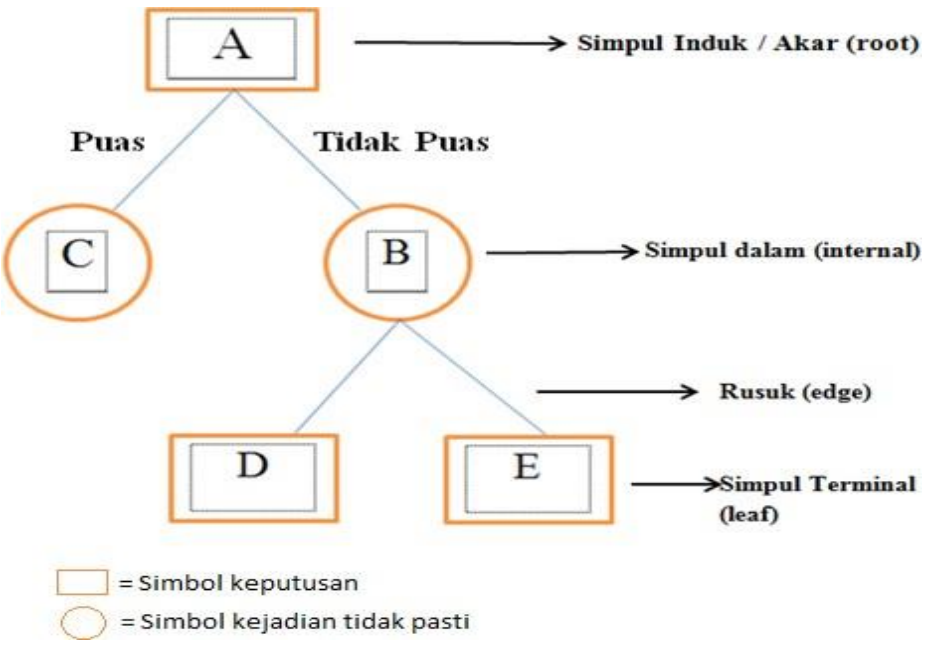
3. *Reinforcement Learning*

Algoritma ini tidak menggunakan data set yang diberikan seperti pada *supervised learning* dan *unsupervised learning*. Pada algoritma ini, terdapat dua

komponen yang utama yaitu *agent* dan *environment*. *Agent* dipaksa dengan sendirinya untuk mempelajari bagaimana ia harus bertindak dalam menghadapi *environment* agar mencapai tujuannya.

2.2.3 Decision Tree

Decision tree adalah salah satu metode klasifikasi yang menggunakan teknik pohon keputusan. Metode ini merepresentasikan atribut dengan *node*, mempresentasikan nilai atribut dengan cabang dan merepresentasikan kelas dengan *leaf*. Pada pohon keputusan terdiri dari beberapa node yaitu *root node, internal node*, dan *leaf node*. *Root node* adalah *node* awal dalam pengujian data dan berada di paling atas dari pohon keputusan, *node* ini akan memiliki beberapa cabang keluar tetapi tidak memilik cabang masuk. *Internal node* adalah *node* yang memiliki satu cabang masuk dan beberapa cabang keluar. *Leaf node* adalah *node* yang akan menyimpulkan prediksi kelas bagi data tersebut dan hanya akan memiliki satu cabang masuk dan tanpa memiliki cabang keluar. *Node* ini juga biasanya disebut dengan *terminal node.*Atribut data harus berupa data kategorik, bila kontinu maka atribut harus didiskretisasi terlebih dahulu (Indriani dan Kartini , 2018). Pada *decision tree* terdapat 3 jenis *node*, yaitu :



Gambar II.1 Pohon keputusan

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

2. *Internal Node*, merupakan *node* percabangan, pada *node* ini hanya terdapat satu input dan mempunyai output minimal dua.
3. *Leaf Node* atau *Terminal Node* , merupakan *node* akhir, pada *node* ini hanya terdapat satu input dan tidak mempunyai output.

2.2.4 Langkah Kerja *Decision Tree*

1. Pemilahan Pohon (*Splitting node*)

Pemilihan adalah proses membagi *node* menjadi dua atau lebih untuk mengidentifikasikan setiap atribut menjadi *root node* atau *internal node*. Untuk memecahkan masalah pemilahan atribut dapat menggunakan seleksi atribut diantaranya yaitu *information gain, gain ratio dan gini index*.

a. Information Gain

Information gain merupakan salah satu seleksi atribut yang sering digunakan untuk menentukkan atribut mana yang terbaik. Nilai *information gain* diperoleh dari nilai *entropy*. *Entropy* merupakan ukuran ketidakpastian kelas dengan menggunakan probabilitas kejadian atau atribut tertentu. Pengukuran nilai ini hanya digunakan sebagai tahap awal untuk penentuan atribut yang nantinya akan digunakan atau dibuang. Atribut yang memenuhi kriteria pembobotan yang nantinya akan digunakan dalam proses klasifikasi sebuah algoritma (Hardle , 1990). *Information Gain* merupakan suatu ukuran korelasi pada model parametrik yang menggambarkan ketergantungan antara dua peubah acak X dan Y. Rumus untuk entropy sebagai berikut :

$$\text{Entropy}(S) = -\sum_{i=1}^{m} p_i \log_2(p_i) \qquad \text{(II. 1)}$$

Dengan :

m = Banyaknya nilai atribut dari kelas yang berbeda

p_i = Peluang bahwa suatu suatu sampel akan masuk dalam kelas yang telah didefenisikan

S = Jumlah seluruh sampel data

Setelah mendapatkan nilai *entropy*, maka perhitungan *information gain* dapat dilakukan dengan menggunakan :

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^{m} P(v_i|s)E(s_i) \quad \text{(II. 2)}$$

Dengan :

A = Atribut

S = banyaknya sampel data umlah seluruh sampel data

$P(v_i|s)$ = Proporsi nilai v_i *yang* muncul pada kelas dalam *node*

$E(s_i)$ = Nilai entropy untuk setiap atribut

Contoh :

Pada contoh ini diketahui :

A = cuaca, suhu , kelembapan dan angin

S = banyaknya nilai yang bermain baseball atau tidak pada atribut yang berbeda.

v_i = cerah, mendung,hujan, dingin, lembut, panas, normal, tinggi, pelan dan kencang

Tabel II.2 Contoh data klasifikasi bermain baseball atau tidak

Cuaca	Suhu	Kelembapan	Angin	Bermain
Cerah	Panas	Tinggi	Pelan	Tidak
Cerah	Panas	Tinggi	Kencang	Tidak
Mendung	Panas	Tinggi	Pelan	Ya
Hujan	Lembut	Tinggi	Pelan	Ya
Hujan	Dingin	Normal	Pelan	Ya
Hujan	Dingin	Normal	Kencang	Tidak
Mendung	Dingin	Normal	Kencang	Ya
Cerah	Lembut	Tinggi	Pelan	Tidak
Cerah	Dingin	Normal	Pelan	Ya
Hujan	Lembut	Normal	Pelan	Ya
Cerah	Lembut	Normal	Kencang	Ya
Mendung	Lembut	Tinggi	Kencang	Ya
Mendung	Panas	Normal	Pelan	Ya
Hujan	Lembut	Tinggi	Kencang	Tidak

Pada contoh ini diketahui A adalah cuaca, suhu , kelembapan dan angin sedangkan S adalah banyaknya nilai yang bermain baseball atau tidak pada atribut yang

berbeda. Dimulai dari *node* akar, menghitung entropy terlebih dahulu untuk *node* akar (semua data) terhadap komposisi kelas.

$$E(semua) = -\begin{pmatrix}(p(ya|semua)xlog_2 p(ya|semua)) + p(tidak|semua) \\ xlog_2 p(tidak|semua)\end{pmatrix}$$

$$= -\left(\left(\left(\frac{9}{14}\right)xlog_2\left(\frac{9}{14}\right)\right) + \left(\left(\frac{5}{14}\right)xlog_2\left(\frac{5}{14}\right)\right)\right) = 0.9403$$

Selanjutnya menghitung entropy untuk setiap nilai atribut terhadap kelas. Untuk entropy nilai dalam cuaca didapat :

$$E(semua_{cerah}) = -\begin{pmatrix}(p(ya|cerah)xlog_2 p(ya|cerah)) + p(tidak|cerah) \\ xlog_2 p(tidak|cerah)\end{pmatrix}$$

$$= -\left(\left(\left(\frac{2}{5}\right)xlog_2\left(\frac{2}{5}\right)\right) + \left(\left(\frac{3}{5}\right)xlog_2\left(\frac{3}{5}\right)\right)\right) = 0.9710$$

$$E(semua_{mendung}) = -\begin{pmatrix}(p(ya|mendung)xlog_2 p(ya|mendung)) + \\ p(tidak|mendung)xlog_2 p(tidak|mendung)\end{pmatrix}$$

$$= -\left(\left(\left(\frac{4}{4}\right)xlog_2\left(\frac{4}{4}\right)\right) + \left(\left(\frac{0}{4}\right)xlog_2\left(\frac{0}{4}\right)\right)\right) = 0$$

$$E(semua_{hujan}) = -\begin{pmatrix}(p(ya|hujan)xlog_2 p(ya|hujan)) + p(tidak|hujan) \\ xlog_2 p(tidak|hujan)\end{pmatrix}$$

$$= -\left(\left(\left(\frac{3}{5}\right)xlog_2\left(\frac{3}{5}\right)\right) + \left(\left(\frac{2}{5}\right)xlog_2\left(\frac{2}{5}\right)\right)\right) = 0.9710$$

Selanjutnya menentukan gain untuk setiap atribut. Dari perhitungan didapat gain seperti dibawah ini. Selengkapnya hasil perhitungan entropy dan gain untuk *node* akar pada tabel berikut:

$$G(semua, cuaca) = E(semua) - \sum_{i=1}^{n} P(v_i|semua)E(semua_{cuaca})$$

$$= E(semua) - \left(p(cerah|semua)xE(semua_{cerah})\right)$$

$$+ \left(p(mendung|semua)xE\left(semua_{mendung}\right)\right)$$

$$+ \left(p(hujan|semua)xE\left(semua_{hujan}\right)\right)$$

$$= 0.9403 - \left(\left(\frac{5}{14}\right)x0.9710\right) + \left(\left(\frac{4}{14}\right)x0\right) + \left(\left(\frac{5}{14}\right)x0.9710\right)$$

$$= 0.2467$$

$$G(semua, suhu) = 0.0292$$

$G(semua, kelembapan) = 0.1518$

$G(semua, angin) = 0.0481$

Tabel II.3 Hasil perhitungan entropy dan gain untuk *information gain*

Node			Jumlah	Ya	Tidak	Entropy	Gain
1	Total		14	9	5	0.9403	
	Cuaca						0.2467
		Cerah	5	2	3	0.9710	
		Mendung	4	4	0	0	
		Hujan	5	3	2	0.9710	
	Suhu						0.0292
		Panas	4	2	2	1.0000	
		Lembut	6	4	2	0.9183	
		Dingin	4	3	1	0.8113	
	Kelembapan						0.1518
		Tinggi	7	3	4	0.9852	
		Normal	7	6	1	0.5917	
	Angin						0.0481
		Pelan	8	6	2	0.8113	
		Kencang	6	3	3	1.0000	

Hasil yang didapat pada tabel diatas adalah menunjukkan bahwa gain tertinggi ada di atribut cuaca sehingga cuaca dijadikan *root node.*

b. Gain Ratio

Gain Ratio merupakan modifikasi dari *information gain* untuk mengurangi bias atribut yang memiliki banyak cabang. *Gain ratio* memiliki sifat bernilai besar bila data menyebar rata dan bernilai kecil bila semua data masuk dalam satu cabang. *Gain ratio* memiliki rumus :

$$Gain\ Ratio = \frac{Gain}{Split\ Info} \qquad \text{(II. 3)}$$

Dimana rumus split info seperti pada rumus *information gain* (II.1) dengan m menyatakan banyaknya split. Jenis split yang dipilih adalah split yang memiliki nilai *gain ratio* yang terbesar (Han dkk , 2012)

Contoh :

Pada contoh ini diketahui :

A = cuaca, suhu , kelembapan dan angin

S = banyaknya nilai yang bermain baseball atau tidak pada atribut yang berbeda.

v_i = cerah, mendung,hujan, dingin, lembut, panas, normal, tinggi, pelan dan kencang

Tabel II.4 Contoh data klasifikasi bermain baseball dengan tipe atribut campuran

Cuaca	Suhu	Kelembapan	Angin	Bermain
Cerah	85	85	Biasa	Tidak
Cerah	80	90	Kencang	Tidak
Mendung	83	78	Biasa	Ya
Hujan	70	96	Biasa	Ya
Hujan	68	80	Biasa	Ya
Hujan	65	70	Kencang	Tidak
Mendung	64	65	Kencang	Ya
Cerah	72	95	Biasa	Tidak
Cerah	69	70	Biasa	Ya
Hujan	75	80	Biasa	Ya
Cerahh	75	70	Kencang	Ya
Mendung	72	90	Kencang	Ya
Mendung	81	75	Biasa	Ya
Hujan	71	80	Kencang	Tidak

Dimulai dari *root node*, menghitung entropy terlebih dahulu untuk *root node*(semua data) terhadap komposisi kelas.

$$E(semua) = -\left(\begin{matrix}\left(p(ya|semua)xlog_2 p(ya|semua)\right) + p(tidak|semua) \\ xlog_2 p(tidak|semua)\end{matrix}\right)$$

$$= -\left(\left(\left(\frac{9}{14}\right)xlog_2\left(\frac{9}{14}\right)\right) + \left(\left(\frac{5}{14}\right)xlog_2\left(\frac{5}{14}\right)\right)\right) = 0.9403$$

Selanjutnya untuk atribut yang bertipe numerik harus ditentukan posisi yang terbaik untuk pemecahan atau pelabelan. Dalam contoh ini digunakan pemecahan

biner. Hasil uji coba pada atribut suhu dengan menghitung nilai gain-nya dapat dilihat pada tabel berikut :

Tabel II.5 Posisi untuk pemecahana atribut suhu di *root node*

Suhu	70		75		80	
	≤	>	≤	>	≤	>
Ya	4	5	7	2	7	2
Tidak	1	4	2	2	4	1
Gain	0.0453		0.0251		0.0005	

Dapat dilihat nilai gain tertinggi pada posisi suhu 70. Oleh karena itu untuk atribut suhu dilakukan diskretisasi pada suhu 70 ketika menghitung entropy dan gain pada semua atribut. Selanjutnya dihitung entropy untuk setiap atribut terhadap kelas, kemudian dihitung gain untuk setiap atribut. Hasilnya dapat dilihat pada tabel berikut :

Tabel II.6 Hasil perhitungan entropy dan gain untuk *gain ratio*

Node			Jumlah	Ya	Tidak	Entropy	Gain
1	Total		14	9	5	0.9403	
	Cuaca						0.2467
		Cerah	5	2	3	0.9710	
		Mendung	4	4	0	0	
		Hujan	5	3	2	0.9710	
	Suhu						0.0453
		≤ 70	5	4	1	0.7219	
		> 70	9	5	4	0.9911	
	Kelembapan						0.1022
		≤ 80	9	7	2	0.7642	
		>80	5	3	2	0.9710	
	Angin						0.0481
		Pelan	8	6	2	0.8113	
		Kencang	6	3	3	1.0000	

Hasil yang didapat di Tabel berikut menunjukkan bahwa gain tertinggi pada atribut cuaca sehingga cuaca dijadikan sebagai *root node*. Selanjutnya dihitung

posisi *split* untuk atribut cuaca dengan menghitung *ratio gain*. Hasil perhitungan *ratio gain split* untuk opsi satu sebagai berikut :

$$Split\ Info\ (semua, cuaca) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

$$Split\ Info\ (semua, cuaca)$$

$$= -\Big(\big(p(cerah|semua)xlog_2\big(p(cerah|semua)\big)\big)$$

$$+ p(mendung|semua)xlog_2\big(p(mendung|semua)\big)\Big)$$

$$+\ p(hujan|semua)xlog_2\big(p(hujan|semua)\big)$$

$$= -\left(\left(\frac{5}{14}\right)xlog_2\left(\frac{5}{14}\right)\right) + \left(\frac{4}{14}xlog_2\left(\frac{4}{14}\right)\right) + \left(\frac{5}{14}xlog_2\left(\frac{5}{14}\right)\right)$$

$$= 1.5774$$

$$Gain\ Ratio = \frac{Gain}{Split\ Info} = \frac{0.2467}{1.5774} = 0.16$$

c. Gini Index

Gini index merupakan suatu ukuran ketidaksamaan pada distribusi pendapatan dan memiliki nilai antara 0 sampai 1. Semakin rendah nilai *Gini index* maka semakin besar pula ukuran kesamaannya. *Gini index* atribut t untuk data dengan m kelas didefinisikan sebagai berikut :

$$Gini\ (t) = 1 - \sum_{i=1}^{m} p_i^2 \qquad (II.4)$$

m = jumlah kelas dari masing masing atribut

p_i = Peluang bahwa suatu suatu sampel akan masuk dalam kelas yang telah didefenisikan

Ketika simpul induk dipecah menjadi partisi m, kualitas pemisahan diberikan oleh *splitting gini index*:

$$Gini_{split} = \sum_{i=1}^{m} \frac{n_i}{n} Gini(i) \qquad (II.5)$$

Pemisahan *node* yang optimal adalah memastikan *splitting gini index* terendah dan idealnya adalah nol (Gorunescu,2011).

Pada contoh berikut ini akan dihitung *gini index* berdasarkan data pada Tabel II.4. Selanjutnya untuk atribut yang bertipe numerik harus ditentukan posisi yang terbaik untuk pemecahan atau pelabelan. Dalam contoh ini digunakan pemecahan biner. Hasil uji coba pada atribut suhu dengan menghitung nilai *gini index*nya dapat dilihat pada Tabel II.5 berikut :

Suhu	70		75		80	
	≤	>	≤	>	≤	>
Ya	4	5	7	2	7	2
Tidak	1	4	2	2	4	1

$$Gini\ (suhu\ \leq 70) = 1 - \left(\left(\frac{4}{5}\right)^2 + \left(\frac{1}{5}\right)^2\right) = 1 - 0{,}64 - 0{,}04 = 0{,}32$$

$$Gini\ (suhu\ > 70) = 1 - \left(\left(\frac{5}{9}\right)^2 + \left(\frac{4}{9}\right)^2\right) = 1 - 0{,}30 - 0{,}19 = 0{,}51$$

$$Gini_{split}(suhu = 70) = \left(\frac{5}{14}\right)(0{,}32) + \left(\frac{9}{14}\right)(0{,}51) = 0{,}11 + 0{,}32 = 0{,}43$$

2. Pemangkasan Pohon (*Pruning*)

Pemangkasan pohon yaitu salah satu teknik yang digunakan dalam mengatasi overfitting dengan mengidentifikasikan dan membuang cabang yang tidak diperlukan pada pohon yang telah terbentuk. Karena pohon keputusan yang dibangun dapat memiliki ukuran yang besar, maka dapat disederhanakan dengan melakukan pemangkasan. Pemangkasan pohon menggunakan tiga algoritma yaitu *Cost Complexity Prunning* (CCP), *Reduced Error Prunning* (REP), dan *Error Based Prunning* (EBD) (Budi dan Wijaya, 2010).

a. *Cost Complexity Prunning* (CCP)

Proses awal dari algoritma *cost complexity pruning* adalah menentukan nilai alpha, yaitu derajat kompleksitas sebuah pohon keputusan. Perhitungan nilai alpha menggunakan :

$$\alpha = \frac{r(t) - r(T_t)}{|n_{T_t}| - 1} \qquad (II.6)$$

Dengan :

$$r(T_t) = \frac{\sum_{s \in T_t} e(s)}{\sum_{s \in T_t} n(s)} \qquad (II.7)$$

Dimana :

$r(T_t)$ = *error rate* pada *subtree* T_t

$e(s)$ = jumlah data dengan kelas minoritas pada *node* s

$n(s)$ = jumlah data pada *node* s

s = *node* angka *subtree* T_t

$$r(t) = \frac{e(t)}{n(t)} \qquad (II.8)$$

Dimana :

$r(t)$ = *error rate* dari *node* t

$e(t)$ = jumlah data pada kelas minoritas pada *node* t

$n(t)$ = jumlah data pada *node* t

α = parameter kompleskitas

n_{T_t} = jumlah *leaf node* pada pohon T_t

Akan dilakukan pemangkasan terhadap *Internal node* dengan nilai α yang paling kecil dan selanjutnya menghitung nilai *error rate* dari pohon yang dipangkas. Proses tersebut akan terus dilakukan sehingga tidak terdapat *internal node* pada pohon keputusan. Seluruh nilai *error rate* dari pemangkasan dengan nilai α yang mendekati serupa satu dengan yang lain akan dijumlahkan dan dihitung nilai rata-ratanya. Nilai rata-rata tersebut akan dibandingkan dengan seluruh nilai rataan *misclassification error* yang ada.

b. *Reduced Error Prunning* (REP)

Reduced Error Prunning merupakan salah satu algoritma *prunning* yang dikemukakan oleh (Quinlan,1987) . Algoritma ini membagi data menjadi dua yaitu *training set* dan *testing set*. Setiap *internal node* pada pohon yang dihasilkan, dihitung berapa nilai *error rate internal node* tersebut menggunakan persamaan (II.6). Kemudian dengan Persamaan (II.7) dihitung nilai *error rate node* tersebut apabila *node* merupakan *leaf node*. Hasil perhitungan keduanya dibandingkan dan *prunning* dilakukan jika *error rate* hasil lebih kecil daripada *error rate*. Apabila

perubahan status pada *node* dari *internal node* menjadi *leaf node* memiliki nilai *error rate* yang sama atau lebih rendah jika *node* tersebut menjadi *internal node* maka *prunning* dilakukan. Proses tersebut terus dilakukan hingga terbentuk pohon keputusan dengan *error rate* yang terbaik dan jumlah aturan yang optimal (Febriansyah dan Afriyeni, 2019).

c. *Error Based Prunning* (EBD).

Algoritma *Error Based Prunning* biasa digunakan pada algoritma *decision tree* c4.5 . Algoritma ini mengizinkan pergantian *subtree* dengan salah satu dari *leaf node*-nya untuk membuat *decision tree* yang lebih simpel. EBP mulai melakukan *prunning* pada *internal node* dari bagian bawah *decision tree*. Pemeriksaan setiap *internal node* dilakukan dengan menggunakan :

$$e'(t) < e'(T_t) + std\left(e'(T_t)\right) \quad \text{(II.9)}$$

Dimana :

$e'(t)$ = *error rate internal node*

$e'(T_t)$ = *error rate subtree* dengan *internal node* t

$$std\left(e'(T_t)\right) = \left[e'(T_t) \times \left(\frac{n(t)-e'(T_t)}{n(t)}\right)\right]^{1/2} \quad \text{(II.10)}$$

Dengan :

$n(t)$ = jumlah *node* yang diperiksa *error rate-nya*

Apabila nilai *error rate* menjadi lebih kecil maka *prunning subtree* dilakukan (Quinlan ,1922).

2.2.5 Gradien

Gradien atau kemiringan garis adalah fungsi bernilai vektor dan sebagai vektor maka gradient memiliki arah dan besaran. Gradien juga disebut konstanta atau bilangan yang menentukan kedudukan atau posisi garis tertentu. Gradien dikelompokan ke dalam tiga katagori yaitu gradien yang bernilai positif, gradien yang bernilai nol, dan gradient yang bernilai negatif. Sebuah garis memiliki kemiringan atau gradien positif apabila posisi garis itu miring ke kanan (jatuh ke arah kanan), kemiringan atau gradien garis nol apabila garis tersebut sejajar sumbu x, dan kemiringan atau gradien garis negatif apabila posisi garis itu miring ke kiri (jatuh ke arah kiri). Sebuah garis tegak lurus sumbu x atau sejajar sumbu y didefinisikan tidak memiliki kemiringan atau gradien.

2.2.6 Loss Function (Fungsi Kerugian)

Loss Function adalah fungsi yang memperkirakan seberapa baik model dalam membuat prediksi dengan data yang diberikan. Ini dapat bervariasi tergantung pada masalah yang dihadapi. *Loss function* yang baik adalah fungsi yang menghasilkan error yang diharapkan paling rendah. Ketika suatu model memiliki kelas yang cukup banyak, perlu adanya cara untuk mengukur perbedaan antara probabilitas hasil hipotesis dan probabilitas kebeneran yang asli, dan selama pelatihan banyak algortima yang dapat menyesuaikan parameter sehingga perbedaan ini diminimalkan. Menurut Hastie dkk (2001) Fungsi kerugian terbagi dua yaitu :

1. Fungsi Kerugian Regresi

Model regresi bertujuan untuk mengembalikan nilai target yang berkelanjutan. Secara matematis , model regresi dapat dirumuskan sebagai:

$$min_{f(x)} \sum_{i=1}^{n} l\big(y - f(x_i)\big) + R_\lambda(f) \qquad (II.11)$$

Dimana :

$f(x_i)$ = Model regresi

$y - f(x_i)$ = Deviasi antara $f(x_i)$ dan nilai target

$l(r)$ = Fungsi kerugian yang mengukur kerugian yang ditimbulkan oleh deviasi

$R_\lambda(f)$ = Istilah regularisasi untuk mengurangi resiko *overfitting.*

$L_1, L_2\ dan\ Hubber$ adalah fungsi kerugian dalam pembelajaran mesin yang dugunakan untuk meminimalkan kesalahan. Fungsi kerugian L_1 adalah singkatan dari *Mean Absolute Error* (MAE), fungsi kerugian L_2 adalah singkatan dari *Mean Squared Errors* (MSE) dan fungsi kerugian *hubber* adalah kombinasi dari $L_1 dan\ L_2$.

- Fungsi Kerugiian L_1

Fungsi ini digunakan untuk meminimalkan kesalahan yang merupakan jumlah dari semua perbedaan mutlak antara nilai sebenarnya (y) dan nilai prediksi(γ). Turunan MAE tidak kontinu dan gradiennya besar sehingga tidak efisien dan sulit dicari solusinya.

$$Loss\ Function\ L_1 = \sum_{i=1}^{n} |y_i - \gamma_i| \qquad (II.12)$$

Pseudo Residual dari fungsi kerugiaan ini adalah

$$r_{im} = -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = \frac{\partial}{\partial \gamma_j}\sum_{i=1}^{n}|y_i - \gamma_i| = \frac{\partial}{\partial \gamma_j}|y_j - \gamma_j|$$

$$= sign\left(y_j - \gamma_j\right)\frac{\partial}{\partial \gamma_j}\left(y_j - \gamma_j\right)$$

$$= -sign\left(y_j - \gamma_j\right) = sign(y_i - \gamma) \quad (II.13)$$

- Fungsi Kerugian L_2

Fungsi ini digunakan untuk meminimalkan kesalahan yang merupakan jumlah dari semua selisih kuadrat antara nilai sebenarnya (y) dan nilai prediksi(γ). Fungsi kerugian MSE adalah fungsi kerugian yang paling umum digunakan untuk masalah regresi dan mudah untuk menghitung gradien.

$$Loss\ Function\ L_2 = \frac{1}{2}\sum_{i=1}^{n}(y_i - \gamma_i)^2 \quad (II.14)$$

Pseudo Residual dari fungsi kerugiaan ini adalah

$$r_{im} = -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = \frac{\partial}{\partial \gamma_j}\frac{1}{2}\sum_{i=1}^{n}(y_i - \gamma_i)^2$$

$$= \frac{1}{2}\frac{\partial}{\partial \gamma_j}\sum_{i=1}^{n}\left(y_j - \gamma_j\right)^2$$

$menggunakan\ chain\ rule$

$$r_{im} = -\left[\frac{\partial L\left(y_i, F(x_i)\right)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = 2\left(y_j - \gamma_j\right)\frac{\partial}{\partial \gamma}\left(y_j - \gamma_j\right)$$

$$= 2\left(y_j - \gamma_j\right)\frac{\partial}{\partial \gamma}\left(y_j - \gamma_j\right)$$

$$= \frac{1}{2}2\left(y_j - \gamma_j\right)(-1)$$

$$= -\left(y_j - \gamma_j\right) = (y_i - \gamma) \quad (II.15)$$

- Fungsi Kerugian *Hubber*

Fungsi ini adalah kombinasi MAE dan MSE untuk mempelajari keunggulannya dapat didiferensial pada 0.

$$L_\delta(y_i,\gamma) = \begin{cases} \frac{1}{2}(y_i-\gamma)^2, & untuk\ |y_i-\gamma| \le \delta \\ \delta|y_i-\gamma| - \frac{1}{2}\delta^2, & untuk\ yang\ lainnya \end{cases} \quad (II.16)$$

δ adalah parameter penyetelan yang menentukkan apa yang ingin dipertimbangkan sebagai outlier.

Pseudo Residual dari fungsi kerugiaan ini adalah

$$r_{im} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = \begin{cases} \frac{1}{2}(y_i-\gamma)^2, & untuk\ |y_i-\gamma| \le \delta \\ \delta|y_i-\gamma| - \frac{1}{2}\delta^2, untuk\ yang\ lainnya \end{cases}$$

$$r_{im} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} = \begin{cases} y_i-\gamma, & untuk\ |y_i-\gamma| \le \delta \\ \delta.sign(y_i-\gamma), untuk\ |y_i-\gamma| > \delta \end{cases} \quad (II.17)$$

2. Fungsi Kerugiaan Klasifikasi

Model klasifikasi brtujuan untuk mengembalikan nilai diskrit. Model klasifikasi biner dapat dirumuskan sebagai :

$$min_{f(x)} \sum_{i=1}^{n} l\big(y_i f(x_i)\big) + R_\lambda(f) \quad (II.18)$$

Dimana :

y_i = Label pada x_i

$y_i\, f(x_i)$ (margin) = Deviasi antara $f(x_i)$ dan *hyperplane.*$f(x_i) = 0$ adalah *hyperplane.*

Aturan pada fungsi kerugian klasifikasi adalah dengan margin positif $y_i\, f(x_i) > 0$, diklasifikasikan dengan benar , sedangkan margin $y_i\, f(x_i) < 0$ diklasifikasikan salah. Tujuan dari algoritma klasifikasi adalah untuk menghasilkan margin positif sesering mungkin.

- Fungsi Kerugian 0-1 (*Missclassification Error*)

Fungsi ini memberikan penalti untuk nilai margin negatif dan tidak ada penalti sama sekali untuk nilai positif, Fungsi ini *non convex* dan tidak dapat dibedakan.

$$\sum_{i=1}^{n} I(y_i\, f(x_i) < 0\,)\ OR \sum_{i=1}^{n} I(y_i\ \neq f(x_i)\,) \quad (II.19)$$

- Fungsi Kerugian *Logistic* (*Binomial Deviance)*

Fungsi Kerugian *The negative binomial log-likelihood* untuk klasifikasi biner. Model awal diberikan oleh rasio *odds log.* Fungsi kerugian *logistic* disimpulkan

dengan menggunakan notasi $y \in \{0,1\}$, notasi ini yang diturunkan dari model probailitas Bernouli. Namun notasi ini membuat kerugian *logistic* sulit untuk dibandingkan dengan fngsi kerugian lainnya. Pada dasarnya , dengan menggunakan notasi $y \in \{-1,1\}$ dapat mengintegrasikan label dan dipredisi bersama-sama ke dalam fungsi probabilitas. Kemudian didapatkan fungsi kerugian MLE. Kelebihan dan kekurangan pada fungsi ini adalah dapat dibedakan dan selalu mengalami kerugian.

$$L(y_i, \gamma) = \log[1 + e^{-y_i \gamma}] \qquad (II.20)$$

- Fungsi Kerugian Eksponensial

Fungsi ini dapat memprediksi yang salah lebih banyak dan memiliki gradien yang lebih besar.

$$Exp\ Loss = e^{-y_i \gamma} \qquad (II.21)$$

2.2.7 Gradient Descent

Gradient descent adalah algoritma pengoptimalan yang digunakan untuk meminimalkan beberapa fungsi dan bergerak secara berulang kearah penurunan paling curam seperti yang ditentukan oleh gradien negatif. Dalam *machine learning*, penurunan gradien atau *gradient descent* digunakan untuk memperbarui parameter model yang digunakan. Gradient descent menemukan parameter yang meminimalkan *loss function* (kesalahan dalam prediksi). *Gradient descent* melakukannya dengan bergerak secara berulang kearah sekumpulan nilai parameter yang meminimalkan fungsi, mengambil kearah yang berlawanan dengan gradien (Anjar , 2019).

Menurut Idaman dkk (2018), *gradient descent* adalah metode optimasi yang digunakan untuk menyelesaikan nilai minimum dari fungsi tujuan. Penurunan gradien menganggap bahwa arah gradien yang berlawanan adalah arah penurunan tercepat, sehingga setiap klai variabel dipindahkan sejauh tertentu sepanjang arah gradien yang berlawanan, fungsi tujuan secara bertahap akan menurun dan akhirnya mencapai minimum.

$$f(x) = f(x_0) + \nabla f(x)(x - x_0) + \frac{1}{2}(x - x_0)^T \nabla^2 f(x)(x - x_0) + \cdots$$

$$f(x) \approx f(x_0) + \nabla f(x)(x - x_0)$$

Jadi jika $(x - x_0)$ dan $f(x)$ berlawanan arah, pergesereran jarak yang sama, $f(x)$ berkurang paling banyak. Jadi langkah inti dari *gardient descent* adalah $x_{k+1} = x_k - a\nabla f(x)$ diantaranya, a adalah ukuran langkah, yang dapat ditentukkan.

Kecepatan pembelajaran adalah angka yang dipilih secara acak yang memberi tahu seberapa jauh untuk memindahkan bobot. Menurut Senov dan Granichin (2017) Jika nilai kecepatan kecil , maka akan membutuhkan waktu yang lama untuk mencapai nilai minimum. Jika nilai kecepatan besar, maka akan melewati nilai minimum. Biasanya nilai kecepatan dipilih secara manual yaitu 0.1 , 0,.01, 0.001 sebagai nilai umum. Algoritma *gradient descent* mengalikan gradien dengan kecepatan pembelajaran untuk menentukkan titik berikutnya.

2.2.7 Gradient *Boosting*

Gradient boosting berawal dari pengamatan Leo Breiman bahwa boosting dapat diartikan sebagai algoritma optimasi fungsi pada biaya yang sesuai. *Gradient boosting* adalah teknik *machine learning* untuk masalah regresi dan klasifikasi, yang menghasilkan model prediksi berupa ansambel model prediksi yang lemah , biasanya pohon keputusan. Ketika pohon keputusan adalah *learner* yang lemah, algoritma yang dihasilkan disebut dengan *gradient decision tree*. Saat mempertimbangkan *ensemble learning*, ada dua metode utama yaitu *bagging* dan *boosting*. *Boosting* melatih model secara berurutan , di mana setiap model belajar dari kesalahan model sebelumnya. Dimulai dengan model dasar yang lemah, model dilatih secara berulang, masing-masing ditambahkan ke prediksi model sebelumnya untuk menghasilkan prediksi keseluruhan yang kuat. Dalam kasus *gradient boosted decision tree*, model yang berurutan ditemukan dengan menerapkan penurunan gradien ke arah gradien rata-rata, yang dihitung sehubungan dengan residual kesalahan fungsi kerugian, dari simpul daun model sebelumnya (Rahayu dkk., 2015) .

Mempertimbangkan pohon keputusan, dapat dilanjutkan langkah berikut. Dimulai dengan kecocokan awal $\boldsymbol{F_0}$, dari data : nilai konstan yang meminimalkan fungsi kerugian **L**:

$$F_0(x) = arg_\gamma min \sum_{i=1}^{n} L(y_i, \gamma) \qquad (II.22)$$

dalam kasus mengoptimalkan kesalahan kuadrat rata-rata, dapat mengambil mean dari nilai target:

$$F_0(x) = \frac{1}{n}\sum_{i=1}^{n} y_i \qquad (II.23)$$

dengan *initial guess* untuk $\boldsymbol{F_0}$ lalu dapat menghitung gradien, atau pseudo residuals, dari **L** sehubungan dengan $\boldsymbol{F_0}$:

$$r_{im} = -\left[\frac{\partial L\big(y_i, F(x_i)\big)}{\partial F(x_i)}\right] \qquad (II.24)$$

Selanjutnya menyesuaikan pohon keputusan $h_1(x)$, ke residuals. Dengan menggunakan pohon regresi, ini akan menghasilkan gradien rata-rata untuk setiap simpul daun dengan menerapkan penurunan gradien untuk meminimalkan kerugian setiap daun dengan melangkah ke arah gradien rata-rata untuk simpul daun seperti yang terdapat dalam pohon keputusan $h_1(x)$. Ukuran langkah ditentukan oleh multiplier γ_1 yang dapat dioptimalkan dengan melakukan pencarian garis. Ukuran langkah selanjutnya menyusut menggunakan kecepatan learning λ_1, sehingga menghasilkan kecocokan data yang ditingkatkan:

$$F_1(x) = F_0(x) + \lambda_1 \gamma_1 h_1(x)$$

BAB III

Metodologi Penelitian

3.1 Sumber Data

Sumber data yang digunakan dalam penelitian ini bersumber dari data derbitur di PT. BPR Syariah Gebu Prima Medan , yang merupakan 1138 data derbitur kredit lancar dan macet pada tahun 2018-2020.

3.2 Identifikasi Variabel Penelitian

Variabel yang digunakan penelitian ini terdiri dari variabel independen dan variabel dependen. Variabel dependen dan variabel independen yang digunakan dalam penelitian ini dapat dilihat pada tabel berikut :

Tabel III.1 Identifikasi variabel penelitian

Variabel	Keterangan	Kategori	Skala
Y	Status Kredit Debitur	0 = Kredit Macet 1 = Kredit Lancar	Kategorik
X1	Jenis Kelamin	1 = Laki-laki 2 = Perempuan	Kategorik
X2	Usia		Numerik
X3	Jumlah Anggota		Numerik
X4	Penghasilan	1 = <3000000 2 = 3000000 – 5000000 3 = > 5000000	Kategorik
X5	Plafond		Numerik

3.3 Prosedur Penelitian

1. akukan Studi Literatur
2. Melakukan pengumpulan data yang diperoleh dari BPRS Gebu Prima Medan
3. Melakukan penginputan data untuk metode *Classification and Regression Tree* menggunakan software SPSS dan metode *Light Gradient Boosting Machine* menggunakan Phyton.
4. Menentukan variabel independen dan variabel dependen dalam penelitian.

5. Melakukan pembagian data menjadi Data Training dan Testing. Pada penilitian akan digunakan 80% Data Training dan 20% Data Testing.
6. Melakukan analisis menggunakan metode *Classification and Regression Tree* dengan menggunakan SPSS :

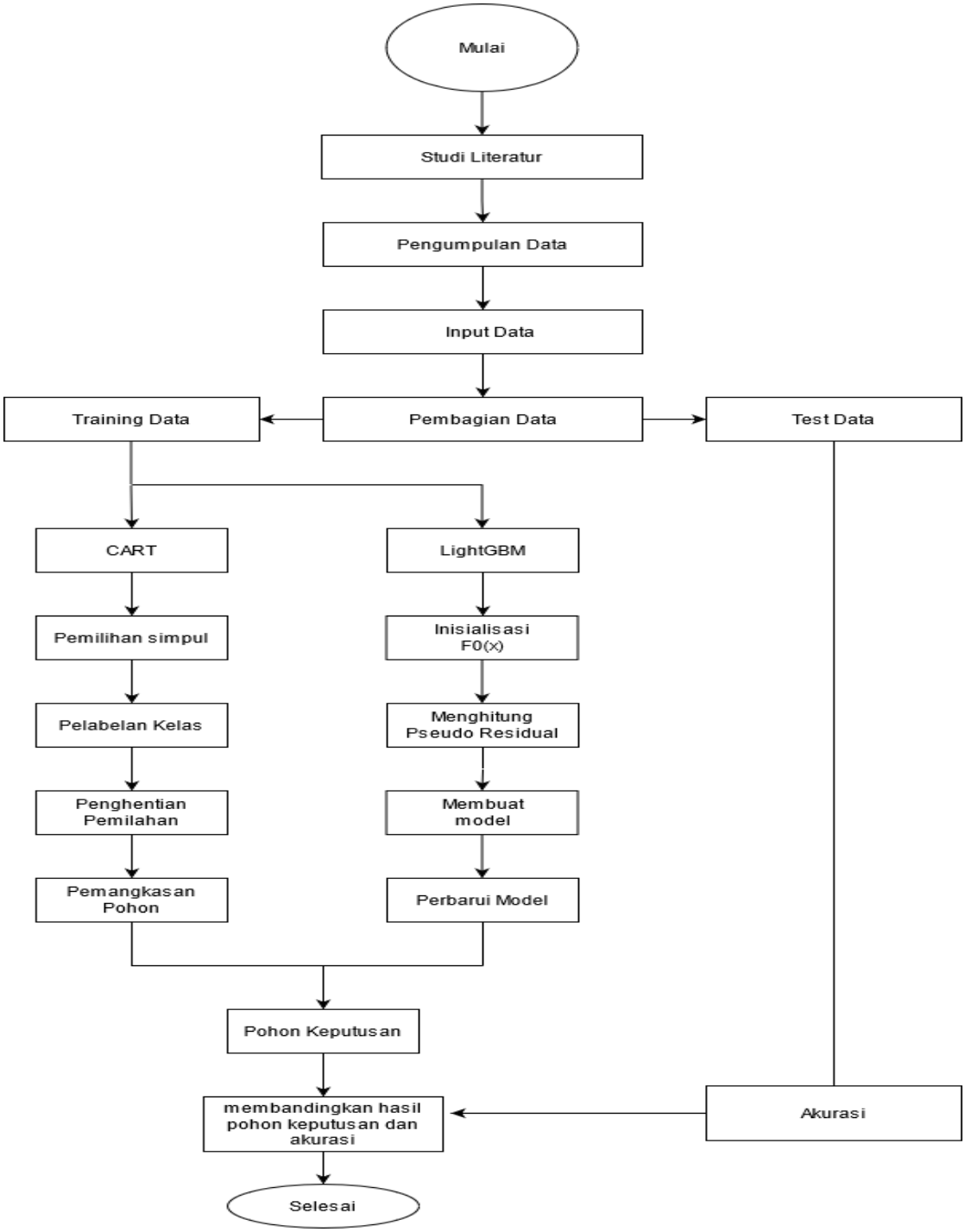
a. Proses pemecahan simpul dengan nilai *improvement* dari nilai Index Gini tertinggi sebagai kriteria dalam memilih variabel yang digunakan untuk memecah sebuah simpul.
b. Pelabelan kelas adalah proses pengidentifikasian tiap simpul – simpul pada suatu kelas tertentu.
c. Proses penghentian pohon keputusan akan berhenti apabila sudah tidak dimungkinkan lagi dilakukan proses pemecahan. Proses pemecahan akan berhenti apabila hanya tersisa satu objek saja yang ada didalam simpul terakhir atau semua objek yang berada didalam sebuah simpul merupakan anggota kelas yang sama (homogen).
d. Menginterpretasikan pohon klasifikasi.

7. Melakukan analisis menggunakan metode *Light Gradient Boosting Machine* dengan menggunakan Phyton :

a. Inisialisasi learner yang lemah atau $F_0(x)$
b. Menghitung M kali secara berulang- ulang dan akan menghasilkan M weak learner. Dengan cara menghitung Menghitung pseudo residual untuk mencari perbedaan antara nilai yang diamati dan nilai yang diprediksi menghasilkan residual.
c. Mencocokan dengan weak learner untuk menggabungkan weak learner awal dan prediksi pseudo residual agar menjadi strong learner.
d. Memperbarui prediksi model.
e. Menginterpretasikan pohon keputusan.

8. Menghitung uji ketepatan klasifikasi dengan menggunakan data *testing* pada kedua metode.
9. Membandingkan hasil pohon keputusan dan akurasi dari metode *Classification and Regression Tree* dan *Light Gradient Boosting Machine.*
10. Memilih metode mana yang memiliki nilai akurasi yang tertinggi.

3.4 Alur Penelitian



Gambar III.1 Alur penelitian

BAB IV

Hasil dan Pembahasan

4.1 *Classification and Regression Tree* (CART)

CART adalah salah satu metode atau algoritma dari salah satu teknik eksplorasi data, yaitu teknik pohon keputusan. Metode ini dikembangkan oleh Leo Breimann, Jerome H. Friedman, Richard A. Olshen dan Charles J. Stone sekitar tahun 1980-an.CART merupakan metode statistika nonparametrik yang dapat menggambarkan hubungan antara variabel dependen dengan satu atau lebih variabel independen. CART dikembangkan untuk topik analisis klasifikasi, baik untuk variabel dependen kategorik maupun kontinu. CART menghasilkan sebuah pohon klasifikasi (*classification trees*) ,jika variabel dependennya kategorik dan menghasilkan pohon regresi (*regression trees*), jika variabelnya kontinu. Statistik nonparametrik yang digunakan untuk analisis klasifikasi yaitu dengan teknik pohon keputusan (Maulana dan Al-Kanomi, 2015).

CART dapat menyeleksi variabel-variabel dan interaksi-interaksi variabel yang paling penting dalam penentuan hasil. Tujuan utama CART adalah untuk mendapatkan suatu kelompok data yang akurat sebagai penciri dari suatu pengklasifikasian. CART mempunyai beberapa kelebihan dibandingkan dengan metode pengelompokan yang klasik, seperti hasilnya lebih mudah diinterpretasikan,lebih akurat, dan lebih cepat perhitungannya.Tingkat kepercayaan yang dapat digunakan dalam pengklasifikasian data baru pada CART adalah akurasi yang dihasilkan oleh pohon klasifikasi yang murni dibentuk dari data yang mempunyai kesamaan kondisi.

CART merupakan metode yang bisa diterapkan untuk himpunan data yang memiliki jumlah besar, variabel independennya banyak dengan skala variabel campuran dilakukan melalui prosedur pemilahan biner, sejauh terlihat dalam Gambar II.1. Pada Gambar II.1, A, B,C, D dan E merupakan variabel independen yang terpilih menjadi simpul. A merupakan simpul induk atau *root node*, B merupakan simpul dalam atau *internal node*, sementara C,D dan E merupakan simpul akhir atau *terminal node* yang tidak bercabang lagi. Setiap simpul terminal merupakan titik akhir dari suatu pemilahan berstruktur pohon, simpul ini tidak bisa dipilah kembali menjadi simpul lain atau dengan kata lain simpul

terminal merupakan simpul yang mengandung amatan-amatan yang homogen dan akhirnya akan dimasukkan sebagai suatu kelas tertentu.

Variabel independen yang dianggap berpengaruh terhadap variabel dependen adalah variabel independen yang muncul sebagai pemisah. Tahapan dalam pembuatan pohon klasifikasi adalah membuat pohon yang besar yaitu dengan simpul banyak. Pohon yang terbentuk kemudian disederhanakan dengan cara memangkas beberapa cabang untuk mendapatkan struktur pohon yang layak dengan aturan-aturan tertentu sehingga terbentuk sebuah pohon optimal (Indriani dan Kartini , 2018).

4.2 Langkah Kerja CART

1. Proses Pemilahan Simpul (*Splitting Nodes*)

Proses pemilahan dimulai dari simpul utama yang terdiri dari data yang akan dipilah. Pemilahan dilakukan untuk memilah data menjadi dua kelompok yaitu kelompok yang masuk simpul kiri dan yang masuk simpul kanan. Aturan pemilahannya sebagai berikut :

a. Tiap pemilahan bergantung pada satu nilai pemilah yang hanya berasal dari satu variable independen.

b. Untuk variable independen yaitu X_g, pemilahan berasal dari pertanyaan "Apakah $x_g \leq c_n$?" jika ruang sampel berukuran N dan terdapat n nilai amatan yang berbeda pada variabel X_g dengan nilai $c_n = (-\infty,\infty)$ dan c_n adalah nilai tengah antara dua nilai amatan variabel berukuran X_g berbeda.

c. Untuk variable kategorik, pemilahan berasal dari semua kemungkinan pemilahan. Apabila merupakan variable kategorik bertaraf, maka akan diperoleh pemilahan terbanyak.Proses pemilahan pada masing-masing simpul induk didasarkan pada *goodness of split criterion* (kriteria pemilahan terbaik). Kriteria pemilahan terbaik ini dibentuk berdasarkan fungsi *impurity* (fungsi keheterogenan) yaitu untuk mengukur tingkat keheterogenan suatu kelas dari suatu simpul tertentu dalam pohon klasifikasi.

Proses pemilahan pada masing-masing simpul induk didasarkan pada *goodness of split criterion* (kriteria pemilahan terbaik). Kriteria pemilihan terbaik ini dibentuk berdasarkan fungsi *impurity* (fungsi keheterogenan) yaitu untuk mengukur tingkat

keheterogenan suatu kelas dari suatu simpul tertentu dalam pohon klasifikasi. Fungsi kehetogenan yang dapat digunakan adalah indeks gini, yaitu :

$$i(t) = \sum_{\substack{j=1 \\ j \neq k}}^{J} P(j|t)P(k|t) \qquad (IV.1)$$

Dimana :

$P(j|t)$ = Proporsi kelas j pada simpul t

$P(k|t)$ = Proporsi kelas k pada simpul t

Goodness of split merupakan suatu evaluasi pemilihan oleh pemilah s pada simpul t. *Goodness of split* $\Delta i(s,t)$ didefenisikan sebagai keheterogenan dan dinyatakan sebagai berikut :

$$\Delta i(s,t) = i(t) - P_L i(t_L) - P_R i(t_R) \qquad (IV.2)$$

Nilai $\Delta i(s,t)$ digunakan sebagai uji *goodness of split criterion* (kriteria uji pemilahan terbaik). Pengembangan pohon dilakukan dengan mencari semua kemungkinan pemilah pada simpul t_1 sehingga ditemukan pemilah s^* yang memberikan nilai penurunan keheterogenan tertinggi yaitu :

$$\Delta i(s^*,t) = max_j \Delta i(s,t_1) \qquad (IV.3)$$

d. Kemudian t_1 dipilah menjadi t_2 dan t_3 menggunakan pemilah s^* dan dengan prosedur yang sama diulangi pada simpul t_2 dan t_3 secara terpisah dan kemudian pada simpul-simpul selanjutnya sampai terbentuk pohon klasifikasi maksimal.

2. Proses Pelabelan Kelas (*Class Assignment*)

Pelabelan kelas dilakukan mulai dari awal pemilahan simpul hingga simpul akhir terbentuk, karena setiap simpul yang dibentuk memiliki kesempatan menjadi simpul akhir. Pelabelan tiap simpul akhir berdasarkan aturan jumlah anggota kelas terbanyak yaitu jika :

$$P(J_0|t) = max_j\ P(j|t) = \ max_j\ \frac{N_j(t)}{N(t)} \qquad (IV.4)$$

Dimana :

$P(J_0|t)$= Proporsi kelas J_0 pada simpul t.

$P(j|t)$ = Proporsi kelas j pada simpul t.

$N_j(t)$ = Banyak pengamatan kelas j pada simpul t.

$N(t)$ = Jumlah pengamatan pada simpul t.

3. Proses Penghentian Pemilahan (*Stop the Splitting*)

Suatu simpul akan menjadi simpul akhir atau tidak akan dipilah kembali apabila hanya terdapat satu pengamatan dalam tiap simpul anak, semua pengamatan dalam tiap simpul anak memiliki distribusi variabel dependen yang identik, dan adanya batasan jumlah kedalaman pohon maksimal yang ditentukan oleh peneliti(Lewis,2000). Apabila hal tersebut terpenuhi, maka pengembangan pohon dihentikan dan diperoleh pohon klasifikasi maksimal atau *maximal tree*.

4. Proses Pemangkasan Pohon Klasifikasi (*Prunning*)

Pohon klasifikasi maksimal yang terbentuk dimungkinkan berukuran sangat besar. Semakin banyak pemilahan yang dilakukan maka tingkat akurasi semakin tinggi, tetapi dengan ukuran yang sangat besar akan sulit dipahami sehingga menyebabkan *overfitting* (pencocokan nilai yang sangat kompleks) untuk data baru. Masalah tersebut diatasi dengan pemangkasan pada pohon klasifikasi maksimal untuk mendapatkan pohon klasifikasi yang optimal. Ukuran pemangkasan yang digunakan untuk memperoleh ukuran pohon yang layak yaitu:

$$R(t) = r(t)P(t) \qquad (IV.5)$$

Dimana :

$R(T)$: *misclassification rate*

$r(t)$: *error rate*

$P(t)$: Probabilitas data

dengan *resubstitution estimate* adalah *tree missclassification cost* atau *tree resubstitution cost* (probabilitas terjadinya kesalahan klasifikasi yang disebabkan oleh pohon klasifikasi yang terbentuk), adalah proporsi amatan yang masuk dalam simpul, merupakan himpunan simpul akhir, sedangkan *resubstitution estimate* adalah probabilitas terjadinya kesalahan klasifikasi di dalam sebuah simpul *t* tertentu yang didefinisikan sebagai berikut :

$$r(t) = 1 - \max_j P(j|t) \qquad (IV.6)$$

Langkah awal proses pemangkasan dilakukan terhadap r(t) yaitu subpohon dari dengan mengambil yang merupakan simpul anak kiri dan yang merupakan simpul

anak kanan hasil dari pemilahan simpul induk. Jika diperoleh dua simpul anak dan simpul induk yang memenuhi persamaan

$$R(t) = R(t_L) + R(t_R) \qquad (IV.7)$$

maka simpul anak dan dipangkas. Proses ini diulangi sampai tidak ada lagi pemangkasan yang mungkin (Machado dkk., 2019).

4.3 Klasifikasi Kredit Macet Menggunakan CART

Berdasarkan data pada lampiran dapat diperoleh faktor – faktor yang mempengaruhi pembayaran kredit pada nasabah dengan metode *Classification and Regression Tree* (CART) sebagai berikut :

1. Proses Pemilihan Simpul (*Splittng Node*)

Suatu *split* akan digunakan untuk memilah simpul t menjadi dua buah simpul yaitu simpul kiri (t_L) dan simpul kanan (t_R) jika s memaksimalkan nilai $\Delta i(s,t)$ (IV.2).

Tabel IV.1 Data calon simpul kiri dan simpul kanan

Nomor Calon Simpul	Calon Simpul Kiri	Calon Simpul Kanan
1	Jenis Kelamin Laki – Laki	Jenis Kelamin Perempuan
2	Usia ≤ 45 tahun	Usia > 45 tahun
3	Penghasilan ≤ 27.750.000	Penghasilan > 27.750.000
4	Plafond ≤ 57.500.000	Plafond > 57.500.000
5	Anggota Keluarga ≤ 3,5	Anggota Keluarga > 3,5

Pada data *training*, untuk menentukkan puncak pohon keputusan, diambil dari 5 artibut yaitu jenis kelamin, usia, jumlah penghasilan, plafond dan anggota keluarga. Kemudian data – data tersebut disusun menjadi calon simpul . Kemudian hitung kandidatan *split purity left* (P_L) dan *purity right* (P_R) menggunakan persamaan :

$$P_L = \frac{Calon\ cabang\ kiri}{Jumlah\ keseluruhan\ data}$$

$$P_R = \frac{Calon\ cabang\ kanan}{Jumlah\ keseluruhan\ data}$$

Tabel IV.2 Hasil perhitungan probilitas cabang

No.	P_L	P_R
1	$\frac{640}{1138} = 0{,}5623$	$\frac{498}{1138} = 0{,}4376$
2	$\frac{204}{1138} = 0{,}1792$	$\frac{934}{1138} = 0{,}8207$
3	$\frac{453}{1138} = 0{,}3980$	$\frac{685}{1138} = 0{,}6019$
4	$\frac{400}{1138} = 0{,}3514$	$\frac{738}{1138} = 0{,}6485$
5	$\frac{323}{1138} = 0{,}2838$	$\frac{815}{1138} = 0{,}7161$

Selanjutnya dengan hasil *split purity left* (P_L) dan *purity right* (P_R) dihitung $P(j|t_L)$ dan $P(j|t_R)$. Hasil perhitungan ditunjukkan pada tabel berikut :

$$P(j|t_L) = \frac{Jumlah\ data\ pada\ kelas\ j\ di\ calon\ cabang\ kiri}{Jumlah\ data\ pada\ calon\ cabang\ kiri}$$

$$P(j|t_R) = \frac{Jumlah\ data\ pada\ kelas\ j\ di\ calon\ cabang\ kanan}{Jumlah\ data\ pada\ calon\ cabang\ kanan}$$

Tabel IV.3 Hasil perhitungan probabilitas cabang berdasarkan status kredit

No	Status Kredit	$P(j\|t_L)$	$P(j\|t_R)$
1	Macet	$\frac{195}{640} = 0{,}3046$	$\frac{155}{498} = 0{,}3112$
	Lancar	$\frac{445}{640} = 0{,}6953$	$\frac{343}{498} = 0{,}6887$
2	Macet	$\frac{35}{204} = 0{,}1715$	$\frac{315}{934} = 0{,}3372$
	Lancar	$\frac{169}{204} = 0{,}8284$	$\frac{619}{934} = 0{,}6627$

3	Macet	$\frac{11}{453} = 0{,}0242$	$\frac{339}{685} = 0{,}4948$
	Lancar	$\frac{442}{453} = 0{,}9757$	$\frac{346}{685} = 0{,}5051$
4	Macet	$\frac{117}{400} = 0{,}2925$	$\frac{233}{738} = 0{,}3157$
	Lancar	$\frac{283}{400} = 0{,}7075$	$\frac{505}{738} = 0{,}6842$
5	Macet	$\frac{125}{323} = 0{,}3869$	$\frac{225}{815} = 0{,}2760$
	Lancar	$\frac{198}{323} = 0{,}6130$	$\frac{590}{815} = 0{,}7239$

Dari Tabel IV.2, kemudian menghitung nilai gini index untuk setiap cabang dengan menggunakan persamaan (II.4) :

$$Gini\ (Jenis\ kelamin) = 1 - (0{,}5623)^2 - (0{,}4376)^2 = 0{,}4923$$

$$Gini\ (Usia) = 1 - (0{,}1792)^2 - (0{,}8207)^2 = 0{,}2944$$

$$Gini\ (Penghasilan) = 1 - (0{,}3980)^2 - (0{,}6019)^2 = 0{,}4793$$

$$Gini\ (plafond) = 1 - (0{,}3514)^2 - (0{,}6485)^2 = 0{,}4559$$

$$Gini\ (anggota\ keluarga) = 1 - (0{,}2838)^2 - (0{,}7161)^2 = 0{,}4066$$

Perhitungan indeks gini tersebut dilakukan untuk semua calon simpul dan diperoleh nilai sebagai berikut:

Tabel IV.4 Hasil Indeks Gini

Cabang	P_L	P_R	$Gini\ (t)$
1	Laki - Laki	Perempuan	0,4923
2	Usia ≤ 45 tahun	Usia > 45 tahun	0,2944
3	Penghasilan ≤ 27.750.000	Penghasilan > 27.750.000	0,4793

4	Plafond ≤ 57.500.000	Plafond > 57.500.000	0,4559
5	Anggota Keluarga ≤ 3,5	Anggota Keluarga > 3,5	0,4066

Selanjutnya dilakukan pemilahan pemilah atau cabang simpul yang akan menjadi *root node* atau simpul akar dengan menggunakan kriteria *goodness of split.* Untuk menghitung nilai *goodness of split* cabang simpul pertama maka menggunakan rumus (II.11) :

$$Gini\ (Jenis\ kelamin_{laki-laki}) = 1 - (0{,}3046)^2 - (0{,}6953)^2 = 0{,}4238$$

$$Gini\ \left(Jenis\ kelamin_{perempuan}\right) = 1 - (0{,}3112)^2 - (0{,}6887)^2 = 0{,}4288$$

$$\Delta i(s, jenis\ kelamin) = 0{,}4923 - \left(0{,}5623(0{,}4238)\right) - \left(0{,}4376(0{,}4288)\right)$$
$$= 0{,}0663$$

$$Gini\ (Usia_{\leq 45\ \text{tahun}}) = 1 - (0{,}1715)^2 - (0{,}8284)^2 = 0{,}2843$$

$$Gini\ (Usia_{> 45\ \text{tahun}}) = 1 - (0{,}3372)^2 - (0{,}6627)^2 = 0{,}4471$$

$$\Delta i(s, usia) = 0{,}2944 - \left(0{,}1792(0{,}2843)\right) - \left(0{,}8207(0{,}4471)\right) = -0{,}1234$$

$$Gini\ (Penghasilan_{\leq 27.750.000}) = 1 - (0{,}0242)^2 - (0{,}9757)^2 = 0{,}0474$$

$$Gini\ (Penghasilan_{> 27.750.000}) = 1 - (0{,}4948)^2 - (0{,}5051)^2 = 0{,}5000$$

$$\Delta i(s, penghasilan) = 0{,}4793 - \left(0{,}3980(0{,}0474)\right) - \left(0{,}6019(0{,}5000)\right)$$
$$= 0{,}1594$$

$$Gini\ (Plafond_{\leq 57.500.000}) = 1 - (0{,}2925)^2 - (0{,}7075)^2 = 0{,}4138$$

$$Gini\ (Plafond_{> 57.500.000}) = 1 - (0{,}3157)^2 - (0{,}6842)^2 = 0{,}4322$$

$$\Delta i(s, plafond) = 0{,}4559 - \left(0{,}3514(0{,}4138)\right) - \left(0{,}6485(0{,}4322)\right) = 0{,}0342$$

$$Gini\ \left(Anggota\ Keluarga_{\leq 3{,}5}\right) = 1 - (0{,}3869)^2 - (0{,}6130)^2 = 0{,}4745$$

$$Gini\ \left(Anggota\ Keluarga_{> 3{,}5}\right) = 1 - (0{,}2760)^2 - (0{,}7239)^2 = 0{,}3997$$

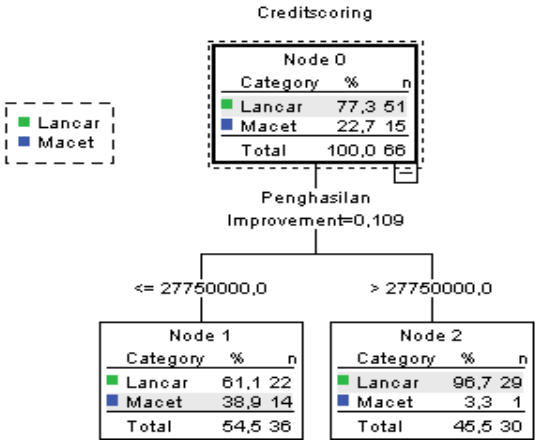
$$\Delta i(s, anggota\ keluarga) = 0{,}4066 - \left(0{,}2838(0{,}4745)\right) - \left(0{,}7161(0{,}3997)\right)$$
$$= -0{,}0142$$

Sehinnga diperoleh nilai *goodness of split* untuk setiap calon cabang dapat dilihat pada Tabel IV.4 :

Tabel IV.5 Hasil *goodness of split*

Cabang	$\Delta i(s,t)$	Kriteria kebaikan
1	0,0663	2
2	−0,1234	5
3	0,1594	1
4	0,0342	3
5	−0,0142	4

Hasil perhitungan *goodness of split* untuk calon cabang ,menunjukkan bahwa calon cabang tertinggi adalah calon cabang 3 sebesar 0,1594, yaitu jumlah penghasilan. Pemilah terbaik untuk simpul 0 atau simpul akar adalah penghasilan. Variabel tersebut terpilih karena memiliki nilai *goodness of split / improvement* tertinggi dari variabel lainnya. Proses pemilihan dapat dilihat pada Gambar IV.1 berikut :



Gambar IV.1 Pemilahan Simpul Akar

2. Proses Pelabelan Kelas (*Class Assigment*)

Proses pelabelan kelas pada simpul – simpul yang terbentuk berdasarkan aturan jumlah anggota kelas terbanyak yaitu $P(k_0|t) = max_j P(k|t)$, maka $j_0 = j$ dengan j = lancar dan macet. Sebagai contoh yaitu simpul 1 pada gambar IV.1 berikut :

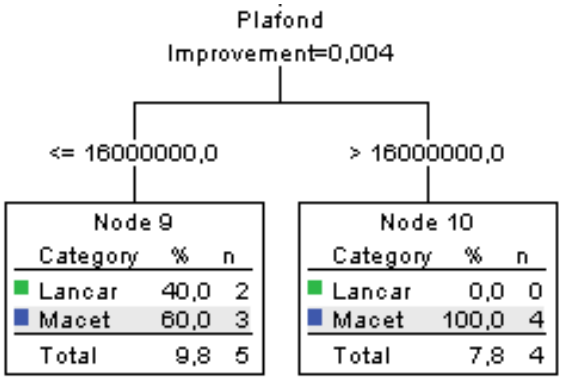
$$P(Lancar|Simpul\ 1) = \frac{22}{36} = 0{,}611 = 61{,}1\%$$

$$P(Macet|Simpul\ 1) = \frac{14}{36} = 0{,}389 = 38{,}9\%$$

Sehingga simpul 1 diberi kelas lancar, karena posisi kelas lancar lebih besar proporsi kelas tidak lancar.

3. Proses Penghentian Pemilahan.

Pohon klasifikasi maksimal yang pertama memiliki 6 simpul dalam dan 7 simpul terminal. Pada pembahasan disini kedalaman maksimal sebesar 4 tingkatan. Proses penghentian pemilahan dapat dilihat pada contoh simpul 9 dan simpul 10 pada Gambar IV.2. Pada simpul 10 terdapat 4 amatan pada kelas yang sama (homogen) dan pada simpul 9 terdapat 5 amatan pada kedalaman maksimal yaitu kedalaman 4 sehingga proses pemilahan dihentikan.



Gambar IV.2 Simpul Penghentian Pemilahan

4. Pemangkasan Pohon Klasifikasi

Pohon klasifikasi maksimal yang dihasilkan kemudian dilakukan pemangkasan pohon klasifikasi dimulai dengan mengambil nilai dari cabang kiri dan cabang kanan. Jika diperoleh dua simpul anak dan simpul induk yang memenuhi persamaan (II.16), maka simpul anak pada cabang kiri dan kanan dipangkas. Proses tersebut diulang sampai tidak ada lagi pemangkasan yang mungkin dilakukan. Sebagai contoh simpul berdasarkan kriteria *cost complexity minimum.* Dalam mendapatkan pohon yang baik maka dapat dilakukan dengan melakukan pemangkasan pohon berdasarkan persamaan (II.15) dan (II.16). Sebagai contoh simpul yang dipangkas yaitu simpul 5.

Pada simpul 5 diperoleh

$$r\ (\ simpul\ 5\ ) = 1 - \max_j P(j|simpul\ 5) = 1 - 0{,}778 = 0{,}222$$

$$P\ (Simpul\ 5\ ) = \frac{9}{1138} = 0{,}007$$

$$R\ (\ Simpul\ 5\ ) = r\ (\ simpul\ 5\ )x\ P\ (\ Simpul\ 5\ ) = 0{,}222\ x\ 0{,}007 = 0{,}001$$

Selanjutnya dihitung nilai $R(t_L)$ dan $R(t_R)$ pada simpul anak, yaitu simpul 9 dan simpul 10.

Pada simpul 9 diperoleh

$$r\ (\ simpul\ 9\ ) = 1 - \max_j P(j|simpul\ 9) = 1 - 0{,}60 = 0{,}4$$

$$P\ (Simpul\ 9\ ) = \frac{5}{1138} = 0{,}004$$

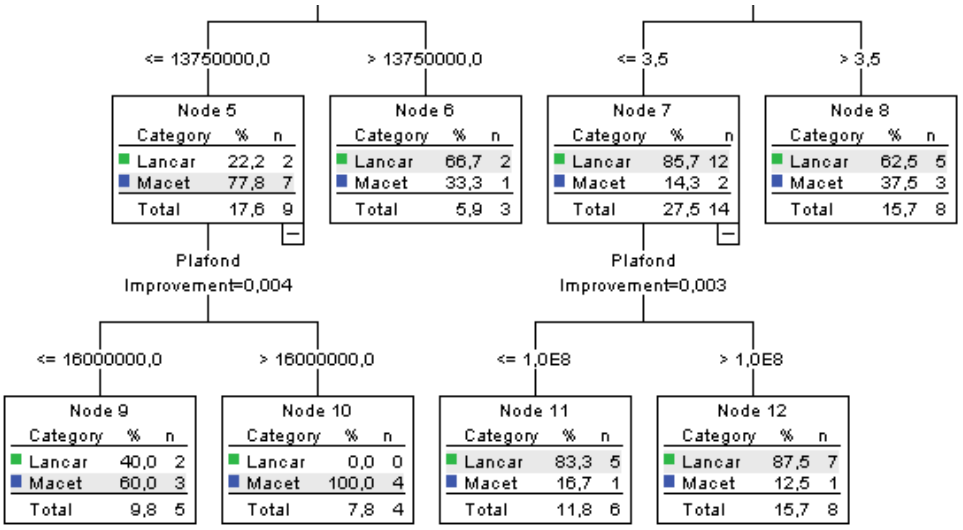
$$R\ (\ Simpul\ 9\ ) = r\ (\ simpul\ 9\ )x\ P\ (\ Simpul\ 9\ ) = 0{,}4\ x\ 0{,}004 = 0{,}0016$$
$$= 0{,}001$$

Pada simpul 10 diperoleh

$$r\ (\ simpul\ 10\ ) = 1 - \max_j P(j|simpul\ 10) = 1 - 1 = 0$$

$$P\ (Simpul\ 10\ ) = \frac{4}{1138} = 0{,}003$$

$$R\ (\ Simpul\ 10\ ) = r\ (\ simpul\ 10\ )x\ P\ (\ Simpul\ 10\ ) = 0\ x\ 0{,}003 = 0$$



Dengan demikian persamaan (II.16) dengan

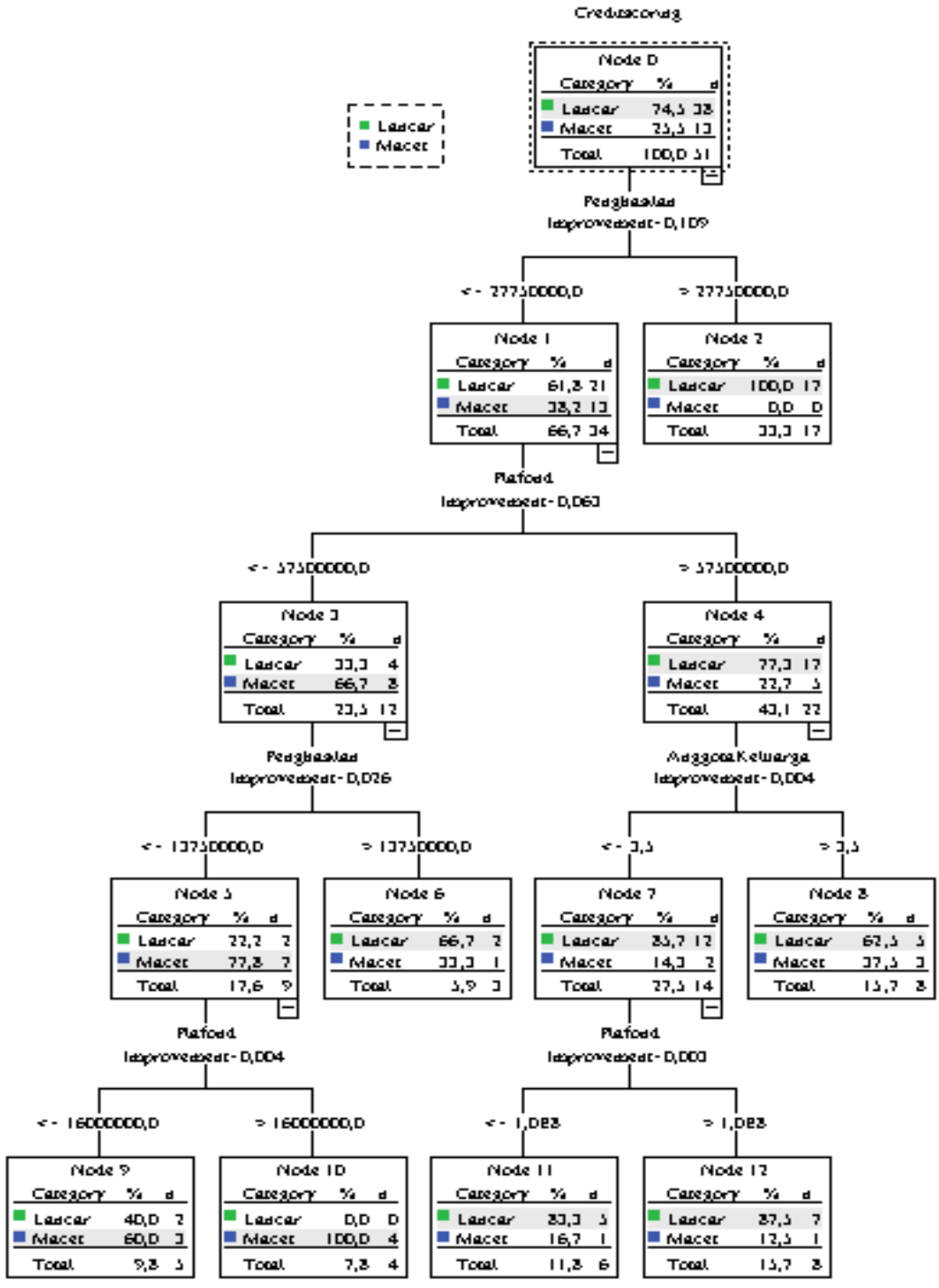
$$R\ (\ Simpul\ 9\ ) + R\ (\ Simpul\ 10\ ) = 0{,}001 + 0 = 0{,}001 =\ R\ (\ Simpul\ 5)$$

Terpenuhi, sehingga dilakukan pemangkasan pada simpul tersebut.

5. Hasil Pohon Klasifikasi.

Analisis yang dilakukan terhadap data debitur PT BPR Syariah Gebu Prima Medan menggunakan klasifikasi CART menghasilkan 13 simpul yang terdiri dari 1 simpul induk, 6 simpul dalam dan 7 simpul terminal. Variabel independen yang masuk ke dalam pohon klasifikasi dan diurutkan berdasarkan besarnya nilai *improvement*

adalah penghasilan, plafond, anggota keluarga, usia dan jenis kelamin dengan kedalaman pohon sebesar 4 sebagaimana dijelaskan pada Gambar IV.3.



Gambar IV.3 Hasil Pohon Keputusan CART

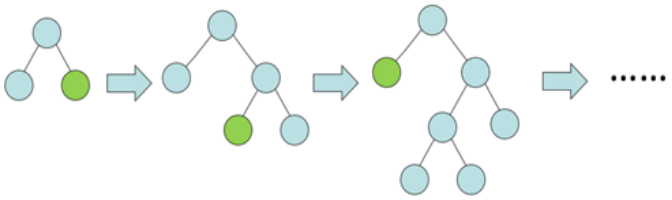
Berdasarkan pohon klasifikasi yang sudah terbentuk menggunakan metode CART, pemilah dengan nilai improvement tertinggi digunakan sebagai pemilah awal dalam

pohon klasifikasi. Dapat dijelaskan bahwa pemilah awal pada simpul induk (simpul 0) adalah penghasilan dengan nilai *improvement* sebesar 0,106 yaitu nasabah yang memiliki penghasilan lebih dari atau sama dengan 27750000 tahun pada simpul 1 dan nasabah yang memiliki penghasilan maksimum 27750000 pada simpul 2. Selanjutnya pada simpul 1 menghasilkan pemilah berdasarkan plafond dengan nilai *improvement* sebesar 0,064 yiatu nasabah dengan plafond lebih dari atau sama dengan 57500000 pada simpul 3 dan nasabah yag memiliki plafond maksimum 57500000 pada simpul 4. Pada simpul 3 menghasilkan pemilah berdasarkan penghasian dengan nilai *improvement* sebesar 0,026 yaitu nasabah yang memiliki penghasilan lebih dari atau sama dengan 13750000 pada simpul 5 dan nasabah yang meiliki penghasilan maksimum 13750000 pada simpul 6. Pada simpul 4 menghasilkan pemilah berdasarkan anggota keluarga dengan nilai *improvement* sebesar 0,004 yaitu nasabah yang memiliki anggota keluarga lebih dari atau sama dengan 3,5 pada simpul 6 dan nasabah yang memiliki anggota keluarga maksimum 3,5 pada simpul 7. Pada simpul 5 menghasilkan pemilah berdasarkan plafond dengan nilai *improvement* sebesar 0,004 yaitu nasabah yang memiliki plafond lebih besar dari atau sama dengan 16000000 pada simpul 9 dan nasabah yang memiliki plafond maksimum 16000000 pada simpul 10. Pada simpul 7 menghasilkan oemilah berdasarkan plafond dengan nilai *improvement* sebesar 0,003 yaitu nasabah yang memiliki plafond lebih dari atau sama dengan 101500000 pada simpul 11 dan nasabah yang memiliki plafond maksimum 101500000 pada simpul 12.

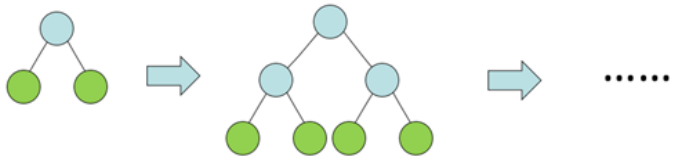
4.4 *Light Gradient Boosting Machine*

Light GBM adalah *framework* peningkatan gradient yang cepat, terdistribusi dan berperforma tinggi berdasarkan pohon keputusan, yang digunakan untuk memberi peringkat, klasifikasi dan banyak tugas *learning* lainnya. LightGBM menggunakan algoritme berbasis histogram, yang memasukkan nilai atribut kontinu ke dalam bin diskrit. Ini mempercepat pelatihan dan mengurangi penggunaan memori. satu hal yang perlu diperhatikan di sini adalah LightGBM menggunakan algoritme berbasis histogram, yang memasukkan nilai atribut berkelanjutan ke dalam bin terpisah. Ini mempercepat pelatihan dan mengurangi penggunaan memori.Selain itu, LightGBM menggunakan jenis *Decision Tree* yang berbeda yang mengoptimalkan *leaf*, bukan

kedalaman seperti *Decision Tree* biasa yaitu, menghitung semua *leaf* yang mungkin dan memilih salah satu dengan kesalahan paling sedikit. Peningkatan gradien adalah teknik pembelajaran mesin yang menghasilkan model prediksi dalam bentuk ensembel pengklasifikasian yang lemah, mengoptimalkan *loss function* yang dapat dibedakan. Ada dua strategi berbeda untuk menghitung pohon keputusan yaitu *level wise* dan *leaf wise*. Strategi *level wise* menumbuhkan pohon dengan tingkat demi tingkat . Dalam strategi ini, setiap *node* membagi data dengan memprioritaskan *node* yang lebih dekat ke akar pohon. Strategi *leaf wise* menumbuhkan pohon dengan memisahkan data pada *node* dengan perubahaan *loss* tertinggi. Pertumbuhan *level wise* biasanya lebih baik untuk kumpulan data yang lebih kecil dibandingkan *leaf wise* akan cenderung untuk overfitting. Pertumbuhan *leaf wise* cenderung unggul dalam data yang lebih besar yang lebih cepat pertumbuhannya daripada *level wise*.



Gambar IV.4 *Leaf Wise Tree Growth*



Gambar IV.5 *Level Wise Tree Growth*

LightGBM menggunakan strategi pertumbuhan berdasarkan *leaf wise* dengan batas kedalaman untuk menemukan simpul daun dengan perolehan split terbesar di semua simpul daun saat ini, kemudian membelah, dan seterusnya (Wang dan Wang , 2020).

Pada penelitian ini LightGBM digunakan pada masalah klasifikasi dengan beberapa langkah yaitu :

1. Inisialisasi *learner* yang lemah

$$F_0(x) = arg_\gamma min \sum_{i=1}^{n} L(y_i, \gamma)$$

Dimana :

$F_0(x)$ = Fungsi dasar yang disebut *learner* yang lemah

$L(y_i, \gamma)$ = Fungsi kerugian

n = Banyaknya sampel

y_i = Nilai yang diamati (*Observed value)*

γ = Nilai untuk *log(odds)*

Untuk memprediksi *log likelihood* dari data observasi yang mana prediksinya telah diberikan adalah sebagai berikut :

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction) = [y_i * \log(p) + (1 - y_i * \log(1 - p)]$$

Dimana :

y_i = *Observed Value 0 or 1*

P = Probabilitas dari nilai yang diprediksi

Tujuannya adalah untuk memaksimalkan fungsi *loglikelihood* . Oleh karena itu, jika ingin menggunakan *log(likelihood)* sebagai fungsi kerugian dimana nilai yang lebih kecil mewakili model yang lebih baik, maka :

$$\log(likelihood) * (-1)$$

Sekarang *log(likelihood)* adalah fungsi dari probabilitas yang diprediksi (p) tetapi *log(likelihood)* dibutuhkan juga sebagai fungsi dari *log(odds)*. Maka rumus dapat diubah menjadi :

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction) = -[y_i * \log(p) + (1 - y_i * \log(1 - p)]$$

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction) = -[y_i * \log(p) + (1 - y_i)(\log(1 - p))]$$

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction) = -[[y_i * \log(p)] - \log(1 - p) + y_i \log(1 - p)]$$

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction)$$
$$= -y_i * [\log(p) - \log(1-p)] - \log(1-p)$$
$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction)$$
$$= -y_i * \left[log\left(\frac{p}{1-p}\right)\right] - \log(1-p)$$

Diketahui bahwa $log\left(\frac{p}{1-p}\right) = \log(odds)$, maka subsitusikan

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction)$$
$$= -y_i * \log(odds) - \log(1-p)$$

Lalu, $p = \frac{e^{\log(odds)}}{1+e^{\log(odds)}}$

$$\log(1-p) = log\left(1 - \frac{e^{\log(odds)}}{1+e^{\log(odds)}}\right)$$
$$\log(1-p) = log\left(\frac{1+e^{\log(odds)}}{1+e^{\log(odds)}} - \frac{e^{\log(odds)}}{1+e^{\log(odds)}}\right)$$
$$\log(1-p) = log\left(\frac{1}{1+e^{\log(odds)}}\right)$$
$$\log(1-p) = \log(1) - \log(1+e^{\log(odds)})$$
$$\log(1-p) = -\log(1+e^{\log(odds)})$$

Kemudian

$$\log(likelihood\ of\ the\ observed\ data\ given\ the\ prediction)$$
$$= -y_i * \log(odds) + \log(1+e^{\log(odds)})$$

2. Untuk m = 1 sampai M:
   a. Hitung pseudo residual:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \qquad untuk\ i = 1, \dots, n.$$

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right] = \frac{\partial}{\partial \log(odds)} y_i * \log(odds) + \log(1+e^{\log(odds)})$$

$$= -y_i + \frac{e^{\log(odds)}}{1+e^{\log(odds)}} = -(-y_i + \gamma)$$
$$= y_i - \gamma = pseudo\ residual$$

   b. Mencocokkan dengan learner yang lemah $h_m(x)$ ke pseudo residual, yaitu melatihnya menggunakan set pelatihan $\{(x_i, r_{im})\}_{i=1}^{n}$.

c. Hitung multiplier γ_m dengan memecahkan masalah one dimensional optimization berikut :

$$\gamma_m = arg_\gamma min \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma\ h_m(x_i))$$

d. Perbarui model :

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x)$$

3. Hasil $F_M(x)$.

Pentingnya atribut dihitung sebagai pengurangan total kriteria yang dinormalisasi yang dibawa oleh atribut tersebut. Yang dikenal sebagai Gini index dapat dilihat pada persamaan (II.4).

4.5 Klasifikasi Kredit Macet Menggunakan LightGBM

Untuk lebih memahami algoritma LightGBM, berikut ini diberikan contoh dengan mengambil sebagian kecil data nasabah (6 nasabah) yang diberikan pada tabel IV.6 berikut :

Tabel IV.6 Contoh data nasabah PT BPRS Gebu Prima

Credit Scoring (Y)	Plafond (X1)	Anggota Keluarga (X2)	Jenis Kelamin (X3)	Penghasilan (X4)	Usia (X5)
0	Rp.400.000.000	4	Laki-laki	Rp.8.000.000	50
0	Rp.400.000.000	4	Perempuan	Rp11.855.000	50
0	Rp.350.000.000	4	Laki-laki	Rp.5.000.000	50
0	Rp.350.000.000	2	Laki-laki	Rp.2.500.000	40
1	Rp.500.000.000	4	Laki-laki	Rp.6.000.000	40
1	Rp.500.000.000	2	perempuan	Rp.3.500.000	40

1. Ketika menggunakan GB untuk klasifikasi, maka dimulai dengan memprediksi nilai awal menggunakan *log(odds). Log(odds)* adalah setara dengan rata-rata dari nilai *observed* (Y). Karena pada penelitian ini terdapat 4 nasabah yang memiliki status kredit macet dan 2 nasabah yang memiliki status kredit lancar, *log(odds) bahwa nasabah memiliki status kredit macet adalah :*

$$\text{odds} = \left(\frac{4}{2}\right) = 2$$

$$\log(\text{odds}) = \ln\left(\frac{4}{2}\right) = \ln(2) = 0{,}69 = 0{,}7$$

Cara termudah untuk menggunakan *log(odds)* untuk klasifikasi adalah dengan mengubahnya menjadi probailitas. Maka dapat menggunakan rumus tersebut :

$$P(macet) = \frac{e^{\log(odds)}}{1 + e^{\log(odds)}} = \frac{e^{0,7}}{1 + e^{0,7}} = 0{,}67 = 0{,}7$$

Jika probabilitas pada kredit macet lebih besar dari 0,5 maka dapat diklasifikasikan semua nasabah dalam *training data* memiliki status kredit macet.(0,5 adalah *commong threshold* yang digunakan untuk keputusan klasifikasi yang dibuat berdasarkan probabilitas).

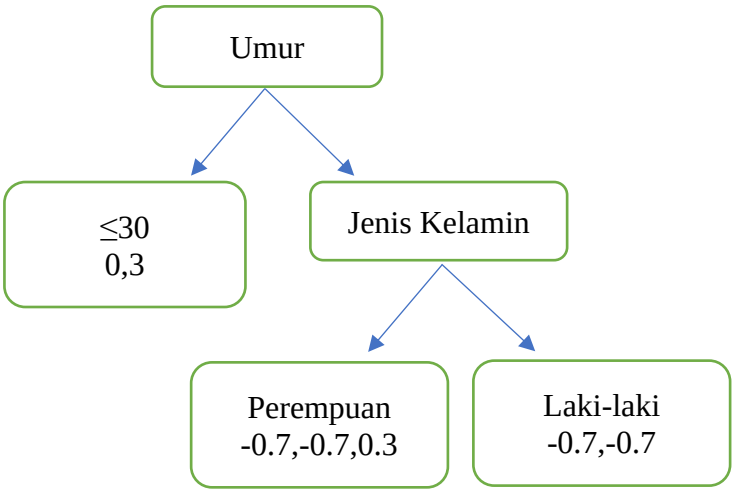
2. Menghitung *Pseudo residual*, yaitu perbedaan antara nilai yang diamati dan nilai yang diprediksi. Pada penelitian ini nilai prediksi diketahui dari status kredit, nasabah yang memiliki status kredit macet bernilai 0 dan nasabah yang memiliki status kredit lancar bernilai 1. Maka nilai *residual menjadi :*

$$residual = observed - predicted$$

Table IV.7 Hasil residual

Plafond (X1)	Anggota Keluarga (X2)	JenKel (X3)	Penghasilan (X4)	Usia (X5)	Credit Scoring (Y)	Residual
Rp.400.000.000	4	Laki-laki	Rp.8.000.000	50	0	-0.7
Rp.400.000.000	5	Perempuan	Rp11.855.000	60	0	-0.7
Rp.350.000.000	3	Laki-laki	Rp.5.000.000	40	0	-0.7
Rp.350.000.000	2	perempuan	Rp.2.500.000	60	0	-0.7
Rp.500.000.000	3	Laki-laki	Rp.6.000.000	30	1	0.3
Rp.500.000.000	2	perempuan	Rp.3.500.000	40	1	0.3

Dengan nilai *residual* dapat membangun pohon :



Gradient boosting memiliki kisaran antara 8 *leaf* hingga 32 *leaf* tetapi pada penelitian ini hanya menggunakan batas dua *leaf* untuk menyederhanakan. Karena batasan pada *leaf*, satu *leaf* dapat memiliki banyak niai. Prediksi dalam bentuk *log(odds)* tetapi *leaf* ini berasal dari probabilitas yang menyebabkan perbedaan. Jadi, tidak bisa hanya menambahkan satu *leaf* yang didapatkan sebelumnya dan dari pohon yang dibentuk mendapatkan prediksi yang baru. Untuk masalah ini dapat menggunakan semacam transformasi. Bentuk transformasi yang paling umum digunakan dalam Gradient Boost for Classification adalah :

$$Transformasi\ Gradient\ boost = \frac{\sum Residual}{\sum[PreviProb * (1 - PrevPprob)]}$$

Leaf pertama hanya memiliki satu nilai *residual* yaitu 0,3, dan karena ini adalah pohon pertama, probabilitas sebelumnya akan menjadi nilai dari *leaf* awal, Karenanya :

$$Transformasi\ Gradient\ boost = \frac{0{,}3}{[0{,}7 * (1 - 0{,}7)]} = 1{,}43$$

Untuk *leaf* kedua ,

$$Transformasi\ Gradient\ boost = \frac{-0{,}7 - 0{,}7}{[0{,}7 * (1 - 0{,}7)] + [0{,}7 * (1 - 0{,}7)]}$$

$$= -6{,}67$$

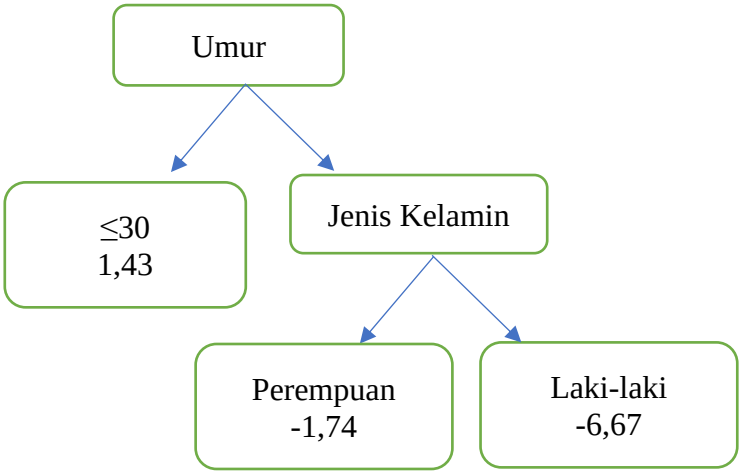
Untuk *leaf* ketiga,

$Transformasi\ Gradient\ boost$

$$= \frac{-0{,}7 - 0{,}7 + 0{,}3}{[0{,}7 * (1 - 0{,}7)] + [0{,}7 * (1 - 0{,}7)] + [0{,}7 * (1 - 0{,}7)]}$$

$$= -1{,}74$$

Sekarang pohon yang ditransformasi terlihat seperti :



3. Setelah mendapatkan transformasi selanjutnya menambahkan nilai awal dengan pohon baru dengan *learning rate*.

$$probabilitas\ baru\ = Oldtree + Learningrate * newtree$$

Learning Rate digunakan untuk mengukur kontribusi dari pohon baru. Ini menghasilkan langkah kecil ke arah prediksi yang benar. Bukti empiris telah membuktikan bahwa mengambil banyak langkah kecil ke arah yang benar menghasilkan prediksi yang lebih baik dengan kumpulan data pengujian yaitu kumpulan data yang belum pernah dilihat model dibandingkan dengan prediksi sempurna pada langkah pertama. Tingkat pembelajaran biasanya berupa angka kecil seperti 0,1. Sekarang dapat menghitung prediksi *log(odds)* yang baru dan setelahnya mencari nilai probabilitas yang baru.

Misalnya, untuk nasabah pertama, nilai awal = 0,7. Tingkat Pembelajaran yang tetap sama untuk semua catatan sama dengan 0,1 dan dengan menskalakan pohon baru, didapatkan hasil nilainya menjadi -1,74. Oleh karena itu, substitusikan ke dalam rumus peroleh:

$$probabilitas\ baru\ = 0{,}7 + 0{,}1(-1{,}74) = 0{,}526$$

Dapat mengganti dan menemukan *log(odds)* baru untuk setiap nasabah dan menemukan probabilitas. Menggunakan probabilitas baru akan menghitung residu baru. Proses ini berulang sampai membuat jumlah pohon maksimum yang ditentukan atau residu menjadi sangat kecil.

Klasifikasi kredit macet untuk studi kasus Bank Pengkreditan Rakyat Syariah Gebu Prima Medan menggunakan LightGBM diselesaikan dengan bantuan software Phyton versi 3.6.5. Berikut ini langkah-langkah yang dilakukan :

Langkah 1 - *Import the library*

```
In [70]: import pandas as pd
         from sklearn.model_selection import train_test_split
         from sklearn.metrics import roc_auc_score
         from sklearn.metrics import classification_report
         from sklearn.metrics import accuracy_score
         from sklearn.metrics import confusion_matrix

         import numpy as np

         import lightgbm as lgb
```

Langkah 2 - Menyiapkan data untuk pengklasifikasi

```
In [33]: df=pd.read_excel('Data LIGHTGBM.xlsx')
```

Langkah 3 - Periksa distribusi variabel target

```
In [44]: y.value_counts()
Out[44]: 0    788
         1    350
```

Variabel target adalah *Credit scoring*. Karena pada penelitian ini masalahnya adalah klasifikasi biner maka terdapat 2 nilai : 0 dan 1. 0 untuk memprediksi lancar dan 1 untuk prediksi macet.

Langkah 4 – Splitting data set kedalam *training set* dan *testing set.*

```
In [36]: X_train, X_test, y_train, y_test = train_test_split(X, y, test_
         size=0.3, random_state=123)
```

Pada penelitian ini digunakan 80% *training set* dan 20% *testing set*.

Langkah 5 – Membangun model LightGBM

```
In [37]: params = {'min_child_weight': 0.6715,
                   'max_depth': 12,
                   'num_leaves': 20,
                   'min_child_samples' :24,
                   'bagging_fraction' : 0.8538,
                   'lambda_l1' : 0.7467,
                   'lambda_l2' : 0.6911
                                                }
         model_lgb = lgb.LGBMClassifier(**params)
         model_lgb.fit(X_train,y_train)

         [LightGBM] [Warning] lambda_l1 is set=0.7467, reg_alpha=0.0 w
         ill be ignored. Current value: lambda_l1=0.7467
         [LightGBM] [Warning] bagging_fraction is set=0.8538, subsampl
         e=1.0 will be ignored. Current value: bagging_fraction=0.8538
         [LightGBM] [Warning] lambda_l2 is set=0.6911, reg_lambda=0.0
         will be ignored. Current value: lambda_l2=0.6911

Out[37]: LGBMClassifier(bagging_fraction=0.8538, lambda_l1=0.7467, lam
         bda_l2=0.6911,
                         max_depth=12, min_child_samples=24, min_child_
         weight=0.6715,
                         num_leaves=20)
```

Langkah 6 – Memprediksi model dan Akurasi

```
In [38]: y_pred  = model_lgb.predict(X_test)
         roc_auc_score(y_pred,y_test)

Out[38]: 0.7878632478632479
```

Di sini, *y_test* adalah label kelas yang sebenarnya dan *y_pred* adalah label kelas yang diprediksi dalam *testing set*.

Langkah 7 - Membandingkan akurasi *training set* dan testing set

Membandingkan akurasi dari kedua set digunakan untuk memeriksa *overfitting.*

```
In [40]: accuracy=accuracy_score(y_pred, y_test)
         print('Model accuracy score: {0:0.4f}'.format(accuracy_score(y_
         test, y_pred)))

         Model accuracy score: 0.8099
```

Langkah 8 – *Overfitting*

```
In [41]: print('Training set score: {:.4f}'.format(model_lgb.score(X_tra
         in, y_train)))

         print('Test set score: {:.4f}'.format(model_lgb.score(X_test, y
         _test)))

         Training set score: 0.9523
         Test set score: 0.8099
```

Akurasi *training set* dan *testin set* cukup sebanding. Jadi tidak bisa dikatakan ada overfitting.

Langkah 9 – Mengurutkan nilai *importance* tertinggi – terendah

```
In [69]: featureImp= []
         for feat, importance in zip(X_test.columns, model_lgb.feature_i
         mportances_):
             temp = [feat, importance]
             featureImp.append(temp)

         fT_df = pd.DataFrame(featureImp, columns = ['Feature', 'Importa
         nce'])
         print (fT_df.sort_values('Importance', ascending = False))

                      Feature  Importance
         0           Plafond         652
         3       Penghasilan         651
         1  Anggota Keluarga         130
         4              Usia          98
         2     Jenis Kelamin          42
```

Langkah 10 - *Confusion Matrix*

```
In [42]: confusion_matrix(y_test, y_pred)

Out[42]: array([[193,  33],
                [ 32,  84]], dtype=int64)
```

Dimana :

True Positive (TP)= 193 *False Positive* = 33

True Negative (TN)= 84 *False Negative* = 32

4.3 Hasil Klasifikasi CART dan LightGBM

Menurut Prasetyo (2012) , sebuah sistem yang melakukan klasifikasi diharapkan dapat melakukan klasifikasi semua himpunan data dengan benar, tetapi tidak dipungkiri bahwa kinerja suatu sistem tidak bisa 100% benar sehinggga sebuah sistem klasifikasi harus diukur tingkat ketepatan klasifikasinya menggunakan matriks konfusi.

Tabel II.6 Matriks konfusi untuk klasifikasi dua kelas

f_{ij}		Kelas hasil prediksi (j)	
		Kelas =1	Kelas = 0
Kelas asli (i)	Kelas = 1	f_{11}	f_{10}
	Kelas = 0	f_{01}	f_{00}

Matriks konfusi merupakan tabel pencatat hasil kerja ketepatan klasifikasi. Tabel 1 merupakan matriks konfusi yang melakukasn klasifikasi biner (dua kelas) yaitu kelas 0 dan 1. Setiap sel f_{ij} dalam matriks menyatakan jumlah record atau data dari kelas I yang hasil prediksinya masuk kekelas j. Misalkan f_{11} adalah jumlah data dalam kelas 1 yang secara benar dipetakan ke kelas 1, dan f_{10} adalah data dalam kelas 1 yang dipetakan secara salah ke kelas 0 .

Berdasarkan isis matriks konfusi, maka dapat diketahui jumlah data dari masing-masing kelas yang diprediksi secara benar yaitu $(f_{11} + f_{00})$ dan data yang diklasifikasikan secara saah yaitu $(f_{10} + f_{01})$. Kuantitas matiks konfusi dapat diringkas menjadi dua nilai , yaitu akurasi dan laju error. Dengan mengetahui jumlah data yang diklasifikasikan secara benar maka dapat diketahui akurasi hasil prediksi, dan data dengan mengetahui jumlah data yang diklasifikasikan secara salah maka dapat diketahui laju error dari prediksi yang salah. Dua kuantitas ini digunakan sebagai matriks ketepatan klasifikasi.

Untuk mengetahui akurasi klasifikasi dengan formula

$$Akurasi = \frac{jumlah\ data\ yang\ diprediksi\ secara\ benar}{jumlah\ data\ yang\ dilakukan}$$

$$Akurasi = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{01} + f_{00}} \qquad (II.31)$$

Untuk mengetahui laju error (kesalahan prediksi) digunakan formula

$$Laju\ Error = \frac{jumlah\ data\ yang\ diprediksi\ secara\ salah}{jumlah\ data\ yang\ dilakukan}$$

$$Laju\ Error = \frac{f_{10} + f_{01}}{f_{11} + f_{00} + f_{01} + f_{00}} \qquad (II.32)$$

Berikut merupakan Tabel *matrix confusion* untuk data *testing* pada metode *Classification and Regression Tree* dan *Light Gradient Boosting Machine* :

Tabel IV.8 Matriks confusion CART

f_{ij}		Kelas hasil prediksi (j)	
		Lancar	Macet
Kelas asli (i)	Lancar	36	6
	Macet	2	7

Berdasarkan Tabel IV.8 dapat dijelaskan bahwa untuk status kredit yang lancar berdasarkan prediksi sebesar 36 nasabah sedangkan status kredit yang macet berdasarkan prediksi sebesar 7 nasabah. Untuk perhitungan akurasi dan laju error pada data *testing* dengan persamaan (II.31) dan (II.32) maka diperoleh hasil berikut :

$$Akurasi = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{01} + f_{00}} = \frac{36 + 7}{36 + 7 + 6 + 2} = 0{,}8431 = 84{,}31\%$$

$$Laju\ Error = \frac{f_{10} + f_{01}}{f_{11} + f_{00} + f_{01} + f_{00}} = \frac{2 + 6}{36 + 7 + 6 + 2} = 0{,}1568 = 15{,}68\%$$

Tabel IV.9 Matriks confusion LightGBM

f_{ij}		Kelas hasil prediksi (j)	
		Lancar	Macet
Kelas asli (i)	Lancar	193	33
	Macet	32	84

Berdasarkan IV.9 dapat dijelaskan bahwa untuk status kredit yang lancar berdasarkan prediksi sebesar 193 nasabah sedangkan status kredit yang macet

berdasarkan prediksi sebesar 84 nasabah. Untuk perhitungan akurasi dan laju error pada data *testing* dengan persamaan (II.31) dan (II.32) maka diperoleh hasil berikut :

$$Akurasi = \frac{f_{11} + f_{00}}{f_{11} + f_{00} + f_{01} + f_{00}} = \frac{193 + 84}{193 + 84 + 33 + 32} = 0{,}8099 = 80{,}99\%$$

$$Laju\ Error = \frac{f_{10} + f_{01}}{f_{11} + f_{00} + f_{01} + f_{00}} = \frac{33 + 32}{193 + 84 + 33 + 32} = 0{,}1900 = 19\%$$

Dari perhitungan yang dilakukan, nilai akurasi yang digunakan untuk mengukur ketepatan klasifikasi untuk data prediksi sebesar 84,31% dan 80,99% maka dapat dikatakan bahwa pohon optimal yang terbentuk mampu mengklasifikasikan data baru sebesar sebesar 84,31% dan 80,99%.

BAB V

Kesimpulan

5.1 Kesimpulan

Berdasarkan hasil analisis dan pembahasan, didapat beberapa kesimpulan bahwa pohon klasifikasi CART dan LightGBM dapat menentukan faktor-faktor yang mempengaruhi keputusan bank dalam pembayaran kredit oleh nasabah yang diurutkan berdasarkan besarnya nilai *goodness of split* dan akurasi yang diuji berdasarkan ketepatan klasifinya adalah berbeda. Pohon klasifikasi CART memperoleh penghasilan, plafond, anggota keluarga, usia dan jenis kelamin sedangkan LightGBM memperoleh plafond, penghasilan, anggota keluarga, usia dan jenis kelamin. Hasil akurasi faktor – faktor yang mempengaruhi keputusan Bank dalam pembayaran kredit oleh nasabah berdasarkan *Classification And Regression Trees* (CART) adalah 84,3% dan *Light Gradient Boosting Machine* (Light GBM) adalah 81% . Dapat dilihat bahwa pada penelitian ini CART memiliki akurasi yang lebih baik daripada LightGBM.

5.2 Saran

Berdasarkan pada kesimpulan yang telah diambil dari hasil penelitian maka klasifikasi pohon keputusan pada *machine learning* masih dapat dikembangkan dalam pemilahan pohon keputusan dengan tujuan mendapatkan akurasi yang terbaik, hal itu bisa didapat dengan membandingkan setiap pengembangan yang telah dilakukan.

DAFTAR PUSTAKA

Alexandri, M. B., dan Sujatna, M.P.C. (2020) : Analisis faktor penyebab kredit macet pada pt bpr banjar arthasariguna tasikmalaya, Jurnal Pemikiran dan Penelitian Administrasi, Sosial, Humaniora, dan Kebijakan Publik, **4**, 77-86.

Breiman, L., Friedman. J., Olshen, R.A., dan Charles, J.S. (1984): *Classification and regression trees*, Chapman and Hall, New York.

Budi, M., Rindang, K., dan Wijaya, S. H. (2010): Perbandingan algoritma pruning pada d*ecision tree* yang dikembangkan dengan algoritma CART, Jurnal Ilmiah Komputer, **15**, 7-13.

Chakraborty, D., Elhegazy, H., Elzarka, H., dan Gutierrez, L. (2020): *A novel construction model using hybrid natural and light gradient boosting*, Advanced Engginering Informatics,**46**

Dinata, R.K., Fajrina dan Khairunnisa. (2018) : Penerapan Algoritma Classification and Regression Tree (CART) pada Penerimaan Anggota Baru Unit Kegiatan Mahasiswa (UKM) di Universitas Malikulsaleh Berbasis Web, Jurnal TECHSI, **10**, 74-81.

Febriansyah, I., dan Afriyeni. (2019): Penyelesaian kredit bermasalah pt. bank pembangunan daerah (BPD) sumatera barat cabang alahan panjang kabupaten solok, Akademi Keuangan dan Perbankan Padang.

Guo, Q., Zhu, Z., Pei, H., Xu, F., Lu, Q., Zhang, D., dan Wu, W. (2019): *Mobile user credit prediction based on lightgbm*, International Conference on Big Data Electronics and Communication Engineering, **29**, 140-144.

Gorunescu, F. (2011): *Data mining : concepts and technique*, Mofgan Kaufmann Publisher Springer, San Fransisco, 166.

Han, J., Kamber, M., dan Pei, J. (2012) : *Data mining : concepts and technique*, Mofgan Kaufmann Publisher Springer, USA, 340-341

Hardle, W. (1990): *Applied nonparametric regression*, Cambridge University Press, New York.

Hartati, A., Zain, I., dan Ulama, B.S.S. (2012) : Analisis CART (Classification and Regression Tree) Faktor – Faktor yang Mempengaruhi Kepala Rumah Tangga di Jawa Timur Melakukan Urbanisasi, Jurnal Sains dan Seni ITS, **1**, 100-105.

Hastie, T., Tibshirani, R., dan Frieman, J. (2001): *The elements of statistical learning : data mining, inference and prediction*, Springer Series in Statistics, Newyork, 360

Indriani,F.,dan Kartini, D. (2018) : Pola klasifikasi sektor usaha UMKM dengan CART menggunakan seleksi atribut *information gain*, Seminar Nasional Teknik Elektro dan Informatika(SNTEI),277-281.

Junaidi (2015): Statistik non-paramaterik, *Universitas Jambi*, .

Kao, L.,J., Chiu, C.,C., dan Chiu, F.,Y.(2012): *A bayesian latent variabel model with classification and regression tree for behavior and credit scoring*, Knowledeg Based System, **36**, 245-252

Kasali, R., (2007): Membidik pasar indonesia segmentasi, targeting, dan positioning, PT Gramedia Pustaka Utama, Jakarta.

Kasih, P. (2019) : Pemodelan data mining decision tree dengan *classification error* untuk seleksi calon anggota tim paduan suara, Journal Innovation in Research of Informatics (INNOVATICS), **1**, 63-69.

Kasmir, S.E., M. (2005) : *Dasar-Dasar Perbankan*, PT Raja Grafindo Persada, Jakarta.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., dan Liu, T.Y. (2017) : *Lightgbm : a highly efficient gradient boosting decesion tree*, Neural Information Processing System, **2017**, 3146-3154.

Khairani, N., 2016: *Statistik Nonparametrik,* Universitas Negeri Medan, Medan.

Lewis, R. J., (2000): *An Introduction to Classification and Reggression Trees (CART) Analysis*. Presented at the 2000 Annual Meeting of Society for Academic Emergency Medicine of Sanfransisco. California.

Machado, M.R., Karray, S., dan Sousa, I.V. (2019) : *Lightgbm: an effective decision tree gradient boosting method to predict customer loyalty in the*

finance industry, International Conference on Computer Science and Education,1111-1116.

Maulana, M.R., dan Al-Kanomi, M.A., (2015): Information gain untuk mengetahui pengaruh atribut terhadap klasifikasi persetujuan kredit, Jurnal Litbang Kota Pekalongan, **9**.

Mewoh, F. C.,Sumampow, H., dan Tamengkel, L. (2016) : Analisis kredit macet (PT Bank Sulut, Tbk Di Manado), Jurnal Administrasi Bisnis, **4**.

Minastireanu, E.A., dan Mesnita, G. (2019) : *LightGBM machine learning algorithm to online click fraud detection*, Journal of Information Assurance and Cybersecurity, **2019.**

Nurhayati, B., dan Iswara, R.P. (2019) : Pengembangan algoritma *unsupervised learning technique* pada big data analysis dimedia sosial sebagai dimedia promosi online bagi masyarakat, Jurnal Teknik Informatika, **12**, 79-96.

Parihar, A., Bhoste, S., Patil, S., Kaptage, S., dan Wagh, S. (2020): *Type 2 diabetes predistion using LightGBM*, International Journal of Future Generation Communication and Networking, **13**, 1365-1373

Patel, H.H., dan Prajapati, P. (2018) : *Study and analysis of decision tree based classification algorithm*, International Journal Of Improved Computer Science and Engineering, **6**, 74-78.

Prabawati, N.H., Widodo, dan Ajie, H. (2019) : Kinerja Algoritma Classification and regression Tree (CART) dalam Mengklasifikasikan Lama Masa Studi Mahasiswa yang Mengikuti Organisasi di Universitas Negeri Jakarta, Jurnal PINTER, **3**.

Prasetyo, E. (2012) : Data mining konsep dan aplikasi menggunakan MATLAB, ANDI Yogyakarta, Yogyakarta.

Quinlan JR. (1987) : *Simplifying Decision Trees*. International Journal of Machine Studies, **27**, 221-234.

Quinlan, J.,R.(1992): *C4.5: Programs for Machine Learning*, San Mateo,Morgan Kaufmann.

Rahayu, E. S., Wahono, R. S., dan Supriyanto, C. (2015) : Penerapan metode average gain, thresold prunning, dan cost complexity prunning untuk split atribut pada algoritma C4.5, Journal of Intelligent Systems, **1**, 91-97.

Senov, A., dan Granichin, O. (2017) : *Projective Approximation Based Gradient Descent Modification* , IFAC-PapersOnLine, **50**, 3899-3904.

Tanjung, R.H., dan Kartiko. (2017): Penerapan metode CART untuk menentukan faktor-faktor yang mempengaruhi pembayaran kredit oleh nasabah, Jurnal Statistika Industri dan Komputasi, **2**,78-83.

Tian, Z., Xiao, J., Feng, H., dan Wei, Y. (2020): *Credit risk assesment based on gradient boosting decision tree*, Procedia Computer Science, **174**,150-160

Timoveef, R. (2004): *Classification and regression trees* (CART) *theory and aplications*, Center of Applied Statistics and Economics humboldt University,Berlin.

Untung, B. (2002): Kredit perbankan di indonesia, ANDI, Yogyakarta.

Wang, Y., dan Wang, T. (2020). *Application of improved lightgbm model in blood glucose prediction*, Applied Science, **10**, 9.

Wanto, A. (2019) : Prediksi Angka Partisipasi Sekolah dengan Fungsi Pelatihan Gradient Descent With Momentum & Adaptive LR, Algoritma: Jurnal Ilmu Komputer dan Informatika, **03**, 09-20

LAMPIRAN

LAMPIRAN

Pada tesis ini penulis melakukan penelitian berkaitan dengan status pembayaran kredit yang diambil dari data di PT BPRS GEBU PRIMA Medan , data tersebut berupa data status kredit nasabah, jumlah penghasilan dalam rupiah, jenis kelamin, plafond dalam rupiah, usia dan jumlah anggota keluarga dari tahun 2018-2020.

No	Credit Scoring (Y)	Plafond (X1)	Anggota Keluarga (X2)	Jenis Kelamin (X3)	Penghasilan (X4)	Usia (X5)
1	2	Rp 400.000.000	4	1	Rp 8.000.000	50
2	2	Rp 400.000.000	4	2	Rp 11.855.000	50
3	2	Rp 350.000.000	4	1	Rp 5.000.000	50
4	2	Rp 350.000.000	4	1	Rp 4.915.000	60
5	2	Rp 350.000.000	3	1	Rp 3.000.000	30
6	2	Rp 350.000.000	2	1	Rp 2.500.000	40
7	2	Rp 300.000.000	2	1	Rp 4.000.000	40
8	2	Rp 300.000.000	3	2	Rp 4.000.000	40
9	2	Rp 300.000.000	4	1	Rp 6.000.000	40
10	2	Rp 300.000.000	2	1	Rp 4.000.000	40
11	2	Rp 300.000.000	3	1	Rp 3.155.000	40
12	2	Rp 300.000.000	3	2	Rp 3.500.000	40
13	2	Rp 270.000.000	3	1	Rp 2.745.000	40
14	2	Rp 270.000.000	4	1	Rp 3.850.000	40
15	2	Rp 250.000.000	3	2	Rp 35.000.000	40
16	2	Rp 250.000.000	4	1	Rp 13.000.000	60
17	2	Rp 250.000.000	5	1	Rp 4.750.000	40
18	2	Rp 250.000.000	3	2	Rp 3.800.000	40
19	2	Rp 250.000.000	5	2	Rp 7.815.000	50
20	2	Rp 250.000.000	4	1	Rp 25.000.000	40
...	...	...	...	...	...	...
1128	1	Rp 5.000.000	4	2	Rp 30.000.000	40
1129	1	Rp 5.000.000	3	2	Rp 24.228.000	30
1130	1	Rp 4.600.000	3	2	Rp 240.000.000	40
1131	1	Rp 4.000.000	3	1	Rp 180.000.000	40
1132	1	Rp 4.000.000	4	2	Rp 120.000.000	50
1133	1	Rp 4.000.000	2	1	Rp 360.000.000	50
1134	1	Rp 4.000.000	3	1	Rp 240.000.000	50
1135	1	Rp 3.000.000	2	2	Rp 240.000.000	60
1136	1	Rp 3.000.000	3	2	Rp 84.000.000	30
1137	1	Rp 2.700.000	4	2	Rp 68.400.000	40
1138	1	Rp 2.510.000	3	2	Rp 120.000.000	40