

BAB I PENDAHULUAN

1.1 Latar Belakang

Perusahaan Grand Indo Perkasa merupakan salah satu industri penambangan batu bara di Sumatera Selatan, yang memiliki permasalahan signifikan dalam mengelola armada alat berat mereka. Alat berat, seperti excavator, dump truck, dan bulldozer, merupakan komponen kritis dalam operasi penambangan yang efisien. Namun, kerusakan alat berat dapat menyebabkan *downtime* yang signifikan, mempengaruhi produktivitas, dan meningkatkan biaya operasional.

Permasalahan yang sering terjadi adalah kerusakan alat berat seringkali tidak dapat diprediksi secara akurat hanya dengan pemantauan manual atau jadwal pemeliharaan rutin. Akibatnya, perusahaan menghadapi risiko tinggi terhadap biaya perbaikan yang tidak terduga dan gangguan operasional. Untuk mengatasi permasalahan ini, penting untuk mengembangkan sistem yang dapat memprediksi kerusakan alat berat sebelum terjadi, sehingga tindakan pencegahan dapat diambil secara proaktif. Alasan peningkatan ketepatan prediksi kerusakan alat berat di penambangan ini adalah untuk dapat mengetahui secara lebih cepat kerusakan alat-alat berat yang ada di penambangan batu bara, agar para pekerja ditempat dapat mengefisienkan waktu pengecekan dan pengerjaan perbaikan kerusakan alat berat.

Salah satu pendekatan yang menjanjikan dalam memprediksi kerusakan alat berat adalah penggunaan algoritma machine learning, khususnya algoritma *decision tree*. *Decision tree* adalah metode yang dapat membantu dalam menganalisis data historis dan kondisi operasional untuk mengidentifikasi pola dan faktor yang berkontribusi terhadap kerusakan alat berat. Dengan memanfaatkan

algoritma ini, sistem prediksi berbasis website dapat dikembangkan untuk memberikan wawasan yang lebih baik mengenai kemungkinan kerusakan, serta memberikan rekomendasi untuk pemeliharaan yang tepat waktu.

Kelebihan dari pendekatan pohon keputusan adalah simpulan daerah pengambilan keputusan yang sebelumnya kompleks dan sangat global, dapat disederhanakan menjadi lebih ringkas dan spesifik, penghapusan perhitungan yang tidak perlu, karena saat menggunakan metode pohon keputusan, sampel diuji hanya berdasarkan kriteria atau kelas tertentu, dan fleksibilitas untuk memilih fitur dari node internal yang berbeda, dengan fitur yang dipilih membedakan suatu kriteria dari kriteria lain dalam node yang sama. (Ramadhon dkk, 2024)

Kelemahan dari pendekatan pohon keputusan ini adalah Terjadi tumpang tindih terutama ketika terdapat banyak kelas dan kriteria yang digunakan, yang dapat menyebabkan peningkatan waktu pengambilan keputusan dan penggunaan memori. (Ramadhon dkk, 2024)

Sistem prediksi berbasis *website* menawarkan keuntungan tambahan, seperti akses mudah dari berbagai lokasi, integrasi data *real-time*, dan kemudahan penggunaan. Dengan mengintegrasikan algoritma *decision tree* ke dalam sistem ini, perusahaan dapat meningkatkan akurasi prediksi kerusakan dan mengurangi risiko *downtime* yang tidak terduga. Studi ini bertujuan untuk menerapkan algoritma *decision tree* dalam pengembangan sistem prediksi kerusakan alat berat berbasis website di Perusahaan Grand Indo Perkasa. Melalui pendekatan ini, diharapkan dapat memberikan solusi yang efektif untuk mengelola dan memelihara alat berat secara lebih efisien, serta meningkatkan produktivitas dan mengurangi biaya operasional di area penambangan batu bara Sumatera Selatan.

Berdasarkan latar belakang tersebut, penulis membuat judul skripsi dengan judul **“Penerapan Algoritma *Decision Tree* untuk Meningkatkan Ketepatan Klasifikasi Kerusakan Alat Berat Berbasis Website (Studi Kasus di Grand Indo Perkasa, Penambangan Batu Bara Sumatera Selatan)”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang mendasari penulis melakukan penelitian ini, penulis merumuskan beberapa rumusan masalah antara lain sebagai berikut :

1. Bagaimana merancang dan membangun aplikasi klasifikasi jenis kerusakan alat berat berbasis web dengan menggunakan PHP dan MySQL?
2. Bagaimana menerapkan algoritma *Decision Tree* dalam mengetahui kerusakan alat berat di Grand Indo Perkasa pada penambangan batu bara di Sumatera Selatan?

1.3 Batasan Masalah

Agar pembahasan dalam penelitian ini tidak melebar dan memudahkan dalam proses penelitian maupun proses perancangan, maka diperlukan batasan masalah sebagai berikut :

1. Penelitian ini akan menggunakan data 5 jenis alat berat dengan 58 seri, 9 merek, dan 16 model alat berat dari perusahaan Grand Indo Perkasa yang ada di penambangan batu bara Sumatera Selatan.
2. Penelitian ini menggunakan 25 alat berat yang mengalami kerusakan, dan 15 total sampel kerusakan.
3. Aplikasi prediksi kerusakan alat berat berbasis web ini dirancang

dengan menggunakan bahasa pemrograman PHP dan *database* nya MySQL, dan XAMPP.

4. Penelitian ini menggunakan algoritma C4.5 sebagai perhitungan untuk menjadi tolak ukur pohon keputusan.
5. Pemodelan data menggunakan UML.

1.4 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan oleh penulis antara lain sebagai berikut :

1. Untuk membangun aplikasi prediksi kerusakan alat berat berbasis web.
2. Untuk menerapkan algoritma *Decision Tree* dalam memprediksi kerusakan alat berat berbasis *website*.

1.5 Manfaat Penelitian

Penulis berharap dari penelitian yang dilakukan dapat memberikan efek yang positif dan memberikan manfaat, antara lain:

1. Menciptakan sebuah aplikasi prediksi kerusakan alat berat berbasis web yang diharapkan dapat diimplementasikan pada PT. Grand Indo Perkasa.
2. Memberikan kemudahan bagi PT Grand Indo Perkasa dalam membuat dan mengembangkan sebuah aplikasi berbasis *website* dengan pemrograman PHP dan basis data MySQL.

1.6 Sistematika Penulisan

Sistematika penulisan skripsi sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini akan dijelaskan tentang latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, sistematika penulisan.

BAB II : TINJAUAN PUSTAKA

Pada bab ini memuat tentang teori-teori yang digunakan, konsep-konsep yang akan diterapkan dalam menyelesaikan permasalahan yang penulis teliti dalam penelitian ini.

BAB III : METODE PENELITIAN

Pada bab ini memuat mengenai metode yang penulis gunakan dalam menyelesaikan rumusan masalah, tahap-tahap mengenai teknik pengolahan data, perancangan aplikasi, dan pembuatan aplikasi.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini memuat hasil-hasil yang didapat dari penelitian serta melakukan pembahasan atas hasil yang diperoleh. Kesulitan yang ditemukan saat perancangan dan pembuatan aplikasi.

BAB V : KESIMPULAN DAN SARAN

Pada bab ini memuat kesimpulan dan saran penulis atas penelitian yang dilakukan

BAB II

TINJAUAN PUSTAKA

2.1 Sistem

2.1.1 Pengertian Sistem

Sistem ialah suatu tatanan yang mendeskripsikan adanya rangkaian komponen-komponen yang saling berhubungan dengan tujuan yang sama secara seimbang dan serasi serta terkoordinasi dan terus berjalan dalam jangka waktu yang terencana (Mudhari, 2018).

Sistem adalah sekelompok komponen dan elemen yang digabungkan menjadi satu untuk mencapai tujuan tertentu. Sistem berasal dari bahasa Latin (*systema*) dan bahasa Yunani (*systema*) adalah suatu kesatuan yang terdiri komponen atau elemen yang dihubungkan bersama untuk memudahkan aliran informasi, materi atau energi untuk mencapai suatu tujuan. Istilah ini sering dipergunakan untuk menggambarkan suatu set entitas yang berinteraksi, di mana suatu model matematika seringkali bisa dibuat (Simangunsong & Informatika, 2018).

Kerangka kerja sistem pada dasarnya adalah berbagai komponen yang paling berdampak dalam menyelesaikan latihan bersama untuk mencapai tujuan tertentu, Kerangka kerja sistem juga memiliki karakteristik atau sifat-sifat tertentu yaitu :

1. Komponen-komponen (*components*)

Komponen-komponen satu sistem dapat berupa suatu subsistem atau bagian-bagian dari sistem yang mempunyai sifat dari sistem yang menjalankan fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan.

2. Batasan Sistem (*boundry*)

Batas sistem merupakan daerah yang membatasi antara suatu sistem dengan sistem lainnya atau dengan lingkungan luarnya.

3. Lingkungan Luar Sistem (*environment*)

Lingkungan luar sistem adalah apapun diluar batas sistem yang mempengaruhi operasi sistem.

4. Penghubung Sistem (*interface*)

Penghubung sistem merupakan media penghubung antara suatu subsistem dengan subsistem yang lain untuk berinteraksi membentuk satu kesatuan.

5. Masukan Sistem (*input*)

Masukan adalah energi yang dimasukkan kedalam sistem. Masukan dapat berupa masukan perawatan (*maintenance input*) dan masukan sinyal (*signal input*).

6. Keluaran Sistem (*output*)

Keluaran adalah hasil dari energi yang diolah dan diklasifikasikan menjadi keluaran yang berguna dan sisa pembuangan.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu bagian pengolah yang akan merubah masukan menjadi keluaran.

8. Sasaran sistem

Suatu sistem pasti mempunyai tujuan (*goal*) atau sasaran (*objectives*). Sasaran dari sistem sangat menentukan sekali masukan yang dibutuhkan

sistem dan keluaran yang akan dihasilkan sistem (Muhammad Dedi Irawan, 2017).

2.1.2 Jenis – Jenis Sistem

1. Sistem Abstrak dan Sistem Fisik

Sistem abstrak (*abstract system*) adalah sistem yang berisi gagasan atau konsep, misalnya sistem teologi yang berisi gagasan tentang hubungan manusia dan tuhan. Sedangkan sistem fisik (*physical system*) adalah sistem yang secara fisik dapat dilihat, misalnya sistem komputer, sistem sekolah, sistem akuntansi dan sistem transportasi.

2. Sistem Deterministik dan Sistem Probabilistik

Sistem deterministik (*deterministic system*) adalah suatu sistem yang operasinya dapat diprediksi secara tepat, misalnya sistem komputer. Sedangkan sistem probabilistik (*probabilistic system*) adalah sistem yang tak dapat diramal dengan pasti karena mengandung unsur probabilitas, misalnya sistem arisan dan sistem sediaan, kebutuhan rata-rata dan waktu untuk memulihkan jumlah sediaan dapat ditentukan tetapi nilai yang tepat sesaat tidak dapat ditentukan dengan pasti.

3. Sistem Tertutup dan Sistem Terbuka

Sistem tertutup (*closed system*) adalah sistem yang tidak bertukar materi, informasi, atau energi dengan lingkungan, dengan kata lain sistem ini tidak berinteraksi dan tidak dipengaruhi oleh lingkungan, misalnya reaksi kimia dalam tabung yang terisolasi. Sedangkan sistem terbuka (*open system*) adalah sistem yang berhubungan dengan lingkungan dan dipengaruhi oleh lingkungan, misalnya sistem perusahaan dagang.

4. Sistem Alamiah dan Sistem Buatan Manusia

Sistem Alamiah (*natural system*) adalah sistem yang terjadi karena alam, misalnya sistem tata surya. Sedangkan sistem buatan manusia (*human made system*) adalah sistem yang dibuat oleh manusia, misalnya sistem komputer.

5. Sistem Sederhana dan Sistem Kompleks

Berdasarkan tingkat kerumitannya, sistem dibedakan menjadi sistem sederhana (misalnya sepeda) dan sistem kompleks (misalnya otak manusia).

2.2 Aplikasi

2.2.1 Pengertian Aplikasi

Pengertian aplikasi menurut (Rachmad hakim S, 2018) merupakan perangkat lunak yang digunakan untuk tujuan tertentu, seperti mengelola dokumen, mengatur Windows dan permainan (game). Sedangkan menurut (Harip Santoso, 2017) bahwa aplikasi adalah suatu kelompok file (form, class, rePort) yang memiliki tujuan untuk melakukan aktivitas tertentu yang saling terkait, misalnya aplikasi payroll, aplikasi fixed asset, dan sebagainya. Aplikasi berasal dari kata application yang artinya penerapan lamaran penggunaan. Dapat disimpulkan dari kedua pendapat ahli tersebut, bahwa aplikasi adalah sebuah perangkat lunak yang memiliki fungsi untuk mengelola dokumen, contohnya aplikasi payroll dan fixed asset.

Aplikasi adalah program siap pakai yang dapat digunakan untuk menjalankan perintah-perintah dari pengguna aplikasi tersebut dengan tujuan mendapatkan hasil yang lebih akurat sesuai dengan tujuan pembuatan aplikasi

tersebut. (Alfeno dan Devi, 2017)

2.2.2 Jenis-Jenis Aplikasi

Aplikasi Komputer atau Aplikasi *Software* adalah Program komputer yang ditulis dalam suatu bahasa pemrograman dan dipergunakan untuk menyelesaikan masalah tertentu. Contoh aplikasi komputer (Faderubun dan Maskhuliah, 2023):

1. Aplikasi Grafik

Aplikasi komputer grafik awal perkembangannya dimulai pada akhir tahun 1950-an dan awal 1960-an. Merupakan suatu program komputer yang digunakan dengan menarik sebuah gambar yang memberikan dimensi baru pada bidang komputer saat itu. Ada 2 cara nyata yang dapat memasukkan informasi grafik ke dalam komputer menjadi suatu aplikasi, yaitu :

- *Direct Programming Graphics*, dengan menuliskannya ke dalam bahasa pemrograman.
- *Digitizing (or Graphics) Tablets*, dengan menggunakan mouse.

2. Animasi

Merupakan bentuk seni yang tampak secara spontan menimbulkan gerakan kehidupan pada suatu obyek. Untuk mencapai efek tersebut, animator harus membangun rangkaian frame/gambar dari sebuah subyek, yang setiap frame berikutnya sedikit berbeda dari frame sebelumnya. Sebenarnya terdapat beberapa fungsi yang berbeda untuk menghasilkan animasi berbasis komputer dan satu daripadanya ialah animasi tiga dimensi (3D). Satu daripada tekniknya ialah dengan mencipta objek yang

kemudian disesuaikan dan digerakkan, di mana akhirnya akan menghasilkan animasi 3D yang lengkap. Fungsi lain untuk mencipta animasi komputer ialah dengan menggunakan alat pengecatan komputer yang standard untuk mengecat frame-frame tunggal sebelum dilakukan proses penggabungan. Ini kemudian disimpan sebagai sebuah file gambar (movie).

3. Aplikasi Web

Merupakan bagian dari internet sebagai komunitas jaringan komputer yang memberikan pelayanan http (*world wide web*). Dengan demikian, definisi teknis dari world wide web adalah semua sumber daya dan semua pengguna di internet yang menggunakan HTTP (*Hypertext Transfer Protocol*). WWW adalah aplikasi yang paling menarik di Internet dan seperti *email*, aplikasi ini sangat penting dan banyak digunakan. Aplikasi ini kadang disebut "*The Killer Application*" atau "*the world is at your fingertip*" karena kita bisa mendapatkan hasil dengan begitu mudah dan bukan sekadar teks yang bisa kita dapatkan namun gambar (*images*), maupun multimedia.

4. Aplikasi Pendidikan

- *Computer Assisted Instruction (CAI)* Komputer secara langsung digunakan dalam proses belajar, sebagai pengganti pengajar ataupun buku. Beberapa aplikasi CAI adalah : Drill and Practice, Tutorial dan Simulasi.
- *Computer Managed Instruction (CMI)* Para pengajar memanfaatkan komputer untuk merencanakan kuliah, disesuaikan dengan kondisi

para siswa, yang terdiri dari acara belajar dengan bantuan komputer, membaca, dan ujian.

5. DBMS

Komputer juga dapat digunakan secara efektif dalam bisnis manajemen. Melalui komputerisasi *data base management system* (DBMS) dan SIM, seorang manajer atau eksekutif perusahaan dapat dibantu dalam membuat keputusan tentang seluruh operasi bisnisnya dan dapat memberikan bimbingan dan instruksi kepada bawahannya.

6. Aplikasi bisnis / perkantoran

Setiap bisnis, besar atau kecil, memproses sejumlah data yang besar dalam operasi hariannya. Data diperoleh dari *payroll*, faktur customer dan rekening keuangan, inventaris manajemen, dan rekening biaya customer yang semuanya harus dikalkulasi, disimpan, diklasifikasikan, disortir, diretriev, diproses kembali, dilaporkan, dikomunikasi dan disebarkan pada suatu waktu. Pemrosesan data seperti itu dikenal sebagai *Business Data Processing* atau *Data Processing* (DP).

2.3 Algoritma

Algoritma menurut (Kani, 2020) adalah suatu upaya dengan urutan operasi yang disusun secara logis dan sistematis untuk menyelesaikan suatu masalah untuk menghasilkan suatu output tertentu.

Menurut Rinaldi Munir (2011:10), Para ahli bahasa menemukan kata *algorism* berasal dari nama cendikiawan muslim yang terkenal yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarijmi (Al-Khuwarijmi dibaca oleh orang Barat menjadi *algorism*) dalam bukunya yang berjudul *Kitab Aljabar Wal-muqabala*,

yang artinya “Buku Pemugaran dan Pengurangan” (The book of restoration and reduction). Dari judul buku itu kita memperoleh kata “aljabar” (algebra). Perubahan dari kata algorism menjadi algorithm muncul karena kata algorism sering dikelirukan dengan arithmetic sehingga akhiran –sm berubah menjadi –thm.

Menurut Thomas H. Cormen (2009:5), Algoritma adalah prosedur komputasi yang mengambil beberapa nilai atau kumpulan nilai sebagai input kemudian di proses sebagai output sehingga algoritma merupakan urutan langkah komputasi yang mengubah input menjadi output. Algoritma dapat diartikan sebagai urutan langkah-langkah untuk menyelesaikan suatu persoalan (Munir & Leony, 2016).

2.4 Data Mining

Penambangan data atau *data mining* merupakan suatu proses menentukan suatu patron dan informasi dalam jumlah yang besar dari suatu data. Data sumbernya berupa basis data, gudang data, data internet yang sistemnya bersifat dinamis dan sebagai tempat penyimpanan informasi lainnya (Han dkk., 2011).

Data mining adalah pengamatan database dengan suatu metode baru yang bermanfaat dan bisa dipahami oleh orang yang memiliki data secara analisis untuk meringkas data. Pada dasarnya, data mining melakukan proses implementasinya sebagai berikut: prediksi, deskripsi, klasifikasi, estimasi, clustering dan asosiasi (Issad dkk., 2019).

Ada beberapa Karakteristik data mining yaitu (Heroe Santoso, I Putu Hariyadi, 2016):

1. Data mining berhubungan dengan penemuan sesuatu yang tersembunyi dan pola data tertentu yang tidak diketahui sebelumnya.
2. Data mining biasa menggunakan data yang sangat besar.

3. Data mining berguna untuk membuat keputusan yang kritis, terutama dalam strategi.

2.5 Decision Tree

2.5.1 Pengertian Decision Tree

Decision tree atau pohon keputusan merupakan metode pendukung keputusan dengan menggunakan model keputusan berbentuk seperti pohon atau *tree*. *Decision tree* dapat mengatasi suatu masalah dengan cara memetakan berbagai alternatif yang mungkin, dan terdapat juga kemungkinan faktor-faktor yang dapat dipengaruhi oleh alternatif tersebut beserta estimasi yang ada. Pohon keputusan penggunaannya dapat menampilkan algoritma dimana hanya berisi pernyataan kontrol bersyarat, menjelaskan urutan proses didahului bertahap dari bagian atas sampai bagian bawah yang akan menghasilkan nilai secara bertahap sehingga menghasilkan keputusan yang akurat (Han dkk., 2019).

Decision tree teknik pemodelan prediktif yang dapat digunakan untuk klasifikasi dan prediksi tugas. Pohon keputusan menggunakan teknik "bagi dan taklukkan" untuk membagi ruang pencarian masalah menjadi kumpulan masalah. Proses pohon keputusan adalah memformat ulang data tabular menjadi model pohon. Pohon model menghasilkan aturan dan disederhanakan (Bahri, 2020).

Decision tree adalah struktur data yang terdiri dari node dan edge. Simpul pohon dibagi menjadi tiga bagian, yaitu simpul akar (akar/simpul), simpul cabang/internal (cabang/internal simpul) dan simpul daun. Pohon keputusan adalah presentasi sederhana metode klasifikasi untuk banyak kelas, di mana simpul internal dan simpul akar ditandai dengan nama atribut, rusuk-rusuk diberi diberi label nilai atribut yang mungkin dan simpul daun ditandai dengan kelas- kelas yang berbeda

(A.H. Nasrullah, 2021).

1. Kelebihan dan Kekurangan *Decision Tree*

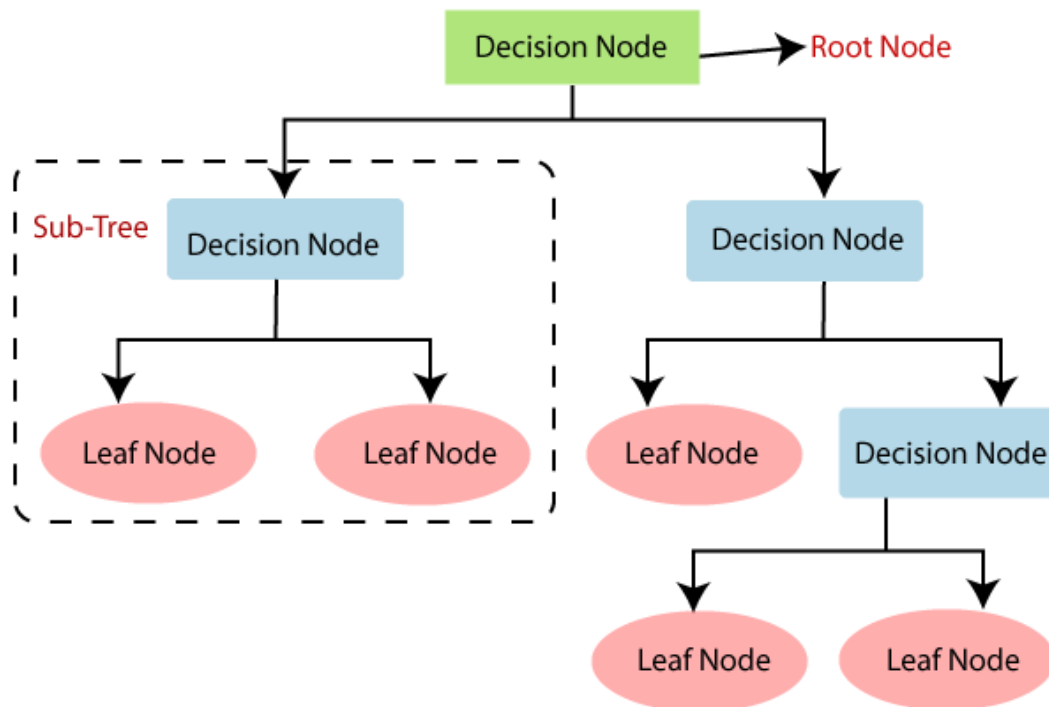
Berikut ini kelebihan dan kekurangan dari pendekatan pohon keputusan melibatkan (Ramadhon dkk, 2024):

Tabel 2.1 Tabel Perbandingan Kelebihan dan Kekurangan *Decision Tree*

Kelebihan	Kekurangan
Simpulan daerah pengambilan keputusan yang sebelumnya kompleks dan sangat global, dapat disederhanakan menjadi lebih ringkas dan spesifik.	Terjadi tumpang tindih terutama ketika terdapat banyak kelas dan kriteria yang digunakan, yang dapat menyebabkan peningkatan waktu pengambilan keputusan dan penggunaan memori.
Penghapusan perhitungan yang tidak perlu, karena saat menggunakan metode pohon keputusan, sampel diuji hanya berdasarkan kriteria atau kelas tertentu.	Akumulasi kesalahan dari setiap tingkat dalam pohon keputusan yang besar.
Fleksibilitas untuk memilih fitur dari node internal yang berbeda, dengan fitur yang dipilih membedakan suatu kriteria dari kriteria lain dalam node yang sama. Kefleksibelan ini meningkatkan kualitas keputusan dibandingkan dengan metode penghitungan satu tahap yang lebih konvensional.	Kesulitan dalam merancang pohon keputusan yang optimal.
Dalam analisis multivariat, terutama dengan jumlah kriteria dan kelas yang tinggi, pohon keputusan dapat menghindari masalah dengan menggunakan kriteria yang lebih sedikit pada setiap node internal tanpa mengurangi kualitas keputusan.	Kualitas keputusan yang dihasilkan sangat tergantung pada desain pohon tersebut.

2. Diagram Pohon Algoritma *Decision Tree*

Berikut ini merupakan diagram yang menggambarkan struktur umum dari algoritma *Decision Tree*:



Gambar 2.1 Diagram Umum *Decision Tree*

- Jenis-jenis *Decision Tree*

Adapun jenis-jenis *decision tree*, yaitu:

1. Categorical variable *decision tree*

Decision tree yang memiliki variabel target kategori. Misalnya, kategorinya bisa ya atau tidak. Kategori berarti bahwa setiap tahap proses keputusan masuk tepat ke dalam satu kelompok, dan tidak ada di antaranya.

2. Continuous variable *decision tree*

Decision tree yang variabel targetnya kontinu. Misalnya, pendapatan individu yang pendapatannya tidak diketahui dapat diprediksi berdasarkan

informasi yang tersedia seperti pekerjaan, usia, dan variabel kontinu lainnya.

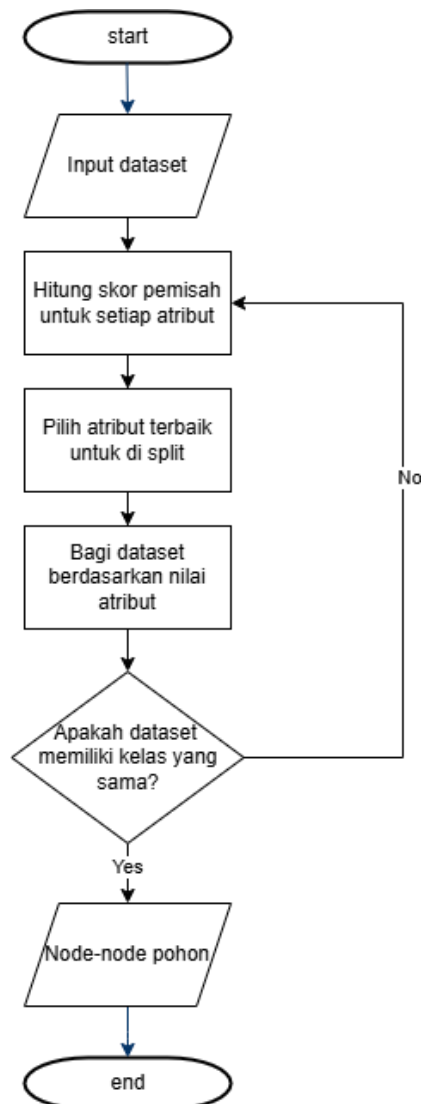
- Cara Kerja Algoritma Decision Tree

Untuk memprediksi kelas dari dataset yang diberikan, algoritma Decision tree dimulai dari simpul akar pohon. Algoritma ini membandingkan nilai atribut root dengan atribut record. Berdasarkan perbandingan tersebut, algoritma menelusuri cabang dan menuju ke simpul berikutnya. Untuk simpul berikutnya, algoritma kembali membandingkan nilai atribut dengan sub-simpul lainnya dan bergerak menuju simpul yang lebih dalam. Tujuannya untuk melanjutkan proses sampai mencapai simpul daun (*node leaf*).

Langkah-langkah dari algoritma *Decision tree* adalah sebagai berikut:

1. Mulai dari simpul akar, kita misalkan sebagai S, yang berisi dataset lengkap.
2. Ambil atribut terbaik dalam dataset menggunakan *Attribute Selection Measure* (ASM). ASM yang bisa digunakan di antaranya *Information Gain* dan Gini Index
3. Pisahkan himpunan S menjadi himpunan bagian yang berisi kemungkinan nilai untuk atribut terbaik.
4. Buat simpul *decision tree* yang berisi atribut terbaik.
5. Buat simpul *decision tree* baru secara rekursif menggunakan himpunan bagian dari kumpulan data yang dibuat pada langkah 3. Lanjutkan proses ini sampai tahap terakhir di mana kita tidak dapat mengklasifikasikan simpul lebih lanjut. Simpul ini yang menjadi simpul akhir atau disebut sebagai simpul daun (*leaf node*).

2.5.2 Penerapan *Decision Tree*



Gambar 2.2 *Flowchart Decision Tree*

Uraian *Flowchart* Algoritma *Decision Tree*:

1. Mulai (*Start*): Algoritma dimulai dari titik ini.
2. *Input Dataset*: *Dataset* dimasukkan sebagai input untuk membentuk pohon keputusan.
3. *Hitung Skor Pemisah untuk Setiap Atribut*: Pada langkah ini, hitung skor pemisahan (seperti *Information Gain*, *Gini Index*, atau *Gain Ratio*) untuk setiap atribut yang ada. Ini menentukan seberapa baik atribut tersebut

memisahkan data ke dalam kelas-kelas yang berbeda.

4. Pilih Atribut Terbaik untuk Di-Split: Pilih atribut yang memiliki skor pemisahan terbaik untuk digunakan dalam memisahkan data pada node saat ini.
5. Bagi Dataset Berdasarkan Nilai Atribut: Dataset dibagi ke dalam subset-subset berdasarkan nilai-nilai dari atribut yang terpilih, sehingga setiap cabang pada node mewakili satu nilai atribut tersebut.
6. Apakah Dataset/atribut Memiliki Kelas yang Sama?

Ini adalah titik keputusan:

- Yes: Jika semua data dalam dataset memiliki kelas yang sama, maka tidak perlu melanjutkan pemisahan, dan kita bisa langsung membuat node akhir (*leaf*).
- No: Jika tidak semua data memiliki kelas yang sama, lanjutkan ke langkah berikutnya.

7. Node-node Pohon: Hasil akhir dataset yang sudah menjadi sebuah pohon keputusan.
8. Akhir (End): Proses selesai ketika seluruh data dalam pohon telah terklasifikasi pada node-node akhir.

2.6 Algoritma C4.5

Ada beberapa tahapan dalam membuat sebuah pohon keputusan dalam algoritma C4.5, Larose yaitu :

1. Mempersiapkan data training. Data training biasanya diambil dari data

histori yang pernah terjadi sebelumnya atau disebut data masa lalu dan sudah dikelompokkan dalam kelas-kelas tertentu.

2. Menghitung akar dari pohon. Akar akan diambil dari atribut yang akan terpilih, dengan cara menghitung nilai gain dari masing-masing atribut. Nilai gain adalah ukuran untuk menentukan atribut mana yang paling informatif dalam membagi dataset selama proses pembuatan pohon keputusan. Nilai gain yang paling tinggi yang akan menjadi akar pertama. Sebelum menghitung nilai gain dari atribut, hitung dahulu nilai entropy. Entropy adalah ukuran ketidakpastian atau keragaman data dalam sebuah dataset. Entropy digunakan untuk menghitung *information gain*, yang membantu memilih atribut terbaik untuk membagi dataset.

Rumus Nilai Entropy

$$\mathbf{Entropy}(S) = \sum -p_i \times \log_2 p_i \quad (2.1)$$

Keterangan :

S = Himpunan Kasus.

n = Jumlah partisi atribut A

|S_i| = Proporsi S_i terhadap S

|S| = jumlah kasus dalam S

3. Menghitung nilai *Gain* menggunakan persamaan.

Rumus Gain

$$\mathbf{Gain}(S, A) = \mathbf{Entropy}(S) - \sum_{v \in \text{values}(A)} \frac{|S_v|}{|S|} \mathbf{Entropy}(S_v) \quad (2.2)$$

Keterangan:

S = Himpunan Kasus

$A = \text{Fitur}$

$n = \text{Jumlah partisi atribut } A$

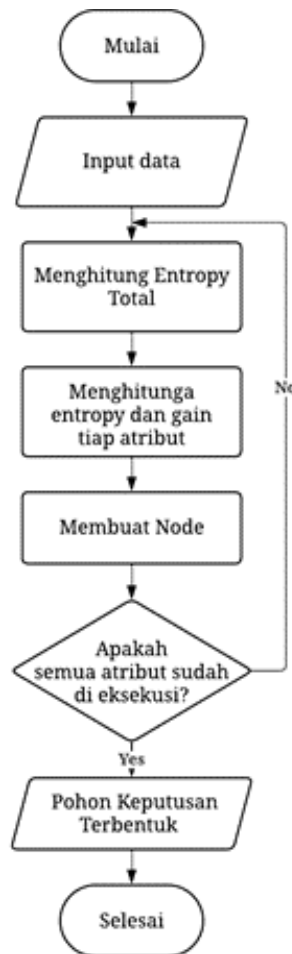
$|S_v| = \text{Proporsi } S_v \text{ terhadap } S$

$|S| = \text{jumlah kasus dalam } S$

4. Ulangi langkah ke 2 dan langkah ke 3 hingga semua *record* terpartisi.
5. Proses partisi pohon keputusan akan berhenti saat:
 - a. Semua *record* dalam simpul N mendapat kelas yang sama.
 - b. Tidak ada atribut di dalam *record* yang dipartisi lagi. Tidak ada *record* di dalam cabang yang kosong.

2.6.1 Flowchart Algoritma C4.5

Berikut ini merupakan *flowchart* algoritma c4.5 yang akan penulis gunakan sebagai algoritma yang membantu membuat pohon keputusan atau *decision tree*.



Gambar 2.3 *Flowchart* Algoritma C4.5

Uraian *Flowchart* Algoritma C4.5:

1. Mulai: Proses pembuatan pohon keputusan dimulai.
2. Input data: Data dimasukkan, yang berisi fitur (atribut) dan target klasifikasi.
3. Menghitung Entropy Total: Entropy total dihitung untuk menentukan seberapa homogen atau tidaknya data tersebut dalam kaitannya dengan target klasifikasi.
4. Menghitung entropy dan gain tiap atribut: Entropy dan gain dihitung untuk masing-masing atribut. Gain digunakan untuk mengukur seberapa besar pengaruh atribut tersebut terhadap pengurangan ketidakpastian (entropy)

dalam data.

5. Membuat Node: Berdasarkan gain tertinggi, node (simpul) dibuat dengan atribut yang paling informatif sebagai pembagi (splitter).
6. Apakah semua atribut sudah dieksekusi? Pertanyaan ini muncul untuk mengecek apakah semua atribut sudah dipertimbangkan atau dihitung dalam proses pembuatan node.
 - No: Jika tidak, proses berlanjut untuk memproses atribut yang tersisa.
 - Yes: Jika semua atribut telah diproses, lanjut ke langkah berikutnya.
7. Pohon Keputusan Terbentuk: Pohon keputusan selesai dibuat dengan semua atribut yang telah dievaluasi.
8. Selesai: Proses pembentukan pohon keputusan selesai

2.7 Alat Berat Pertambangan Batu Bara

2.7.1 Pengertian Alat Berat Batu Bara

Alat berat batu bara adalah peralatan mesin berukuran besar yang didesain untuk melaksanakan fungsi konstruksi pertambangan. Alat berat merupakan salah satu faktor penting didalam proyek-proyek konstruksi dengan skala yang besar. Penggunaan alat berat bertujuan untuk memudahkan manusia dalam mengerjakan pekerjaannya sehingga hasil yang diharapkan dapat dicapai dengan waktu yang singkat.

2.7.2 Jenis-Jenis Alat Berat Pertambangan Batu Bara

Berikut ini beberapa alat berat yang digunakan pada pertambangan batu bara Grand Indo Perkasa:

1. Excavator



Gambar 2.4 Alat Berat Excavator

2. Loader



Gambar 2.5 Alat Berat Loader

3. Dozer



Gambar 2.6 Alat Berat Dozer

4. Compactor



Gambar 2.7 Alat Berat Compactor

5. Greder



Gambar 2.8 Alat Berat Greder

2.8 Flowchart

Menurut (Pahlevi dan Astutik, 2021) *Flowchart* merupakan suatu bagan dengan simbol-simbol tertentu yang menggambarkan urutan proses secara mendetail dan hubungan antara suatu proses atau instruksi dengan proses lainnya dalam suatu program. *Flowchart* hanya untuk menyederhanakan rangkaian proses untuk memudahkan pemahaman pengguna terhadap informasi tersebut. *Flowchart* bisa untuk segala sesuatu yang mengandung rangkaian kegiatan. *Flowchart* menggunakan simbol yang terstandarisasi secara internasional. Hal ini bertujuan

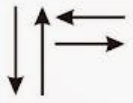
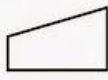



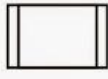


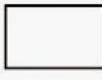
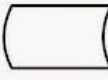

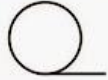

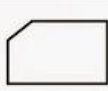


untuk mempermudah orang untuk memahami berbagai variasi *Flowchart*.

Pengertian *Flowchart* (Diagram Alir) atau disebut *Flowchart* merupakan bagan (*Chart*) yang mengarahkan alir (*flow*) di dalam prosedur atau program sistem secara logika. *Flowchart* adalah cara untuk menjelaskan tahap-tahap pemecahan masalah dengan merepresentasikan simbol- simbol tertentu yang mudah dipahami, mudah digunakan dan standar. Tujuan penggunaan *Flowchart* adalah untuk menggambarkan suatu tahapan penyelesaian masalah secara sederhana, terurai dan rapi dengan menggunakan simbol-simbol yang standar yang dapat dimengerti oleh *programmer*. Tahapan penyelesaian masalah yang disajikan harus tepat, sederhana, dan jelas. Pengertian *Android* adalah sistem operasi yang berbasis OS Linux untuk telepon seluler seperti smartphone dan komputer tablet (Syamsiah, 2019).

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urutan-urutan prosedur dari suatu program. *Flowchart* menolong analyst dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. *Flowchart* biasanya mempermudah penyelesaian suatu masalah khususnya masalah yang perlu dipelajari dan dievaluasi lebih lanjut. *Flowchart* adalah bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. *Flowchart* digunakan untuk merepresentasikan maupun mendesain program. Oleh karena itu *flowchart* harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman.(Adelia dan Setiawan, 2011)

Flowchart memiliki beberapa simbol, yang termuat dalam tabel 2.2. (Informatikalogi, 2019)

Tabel 2.2 Tabel *Flowchart*

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/procedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

2.9 Unified Modelling Language (UML)

UML (*Unified Modeling Language*) merupakan salah satu bahasa spesifikasi standar yang dapat digunakan untuk membangun sebuah perangkat lunak. Dimana UML ini juga merupakan suatu metodologi untuk pengembangan sistem dengan berorientasi objek (Suendri, 2018).

Unified Modeling Language atau yang biasa disebut dengan UML merupakan sebuah sistem yang bermodelkan arsitektur dan berjalan pada OOAD atau *object-Oriented Analysis Design* yang mana cara kerjanya dengan menggunakan bahasa bersifat konsisten dalam mengkontruksi, visualisasi, mendokumentasi artifact (Sebagian informasi yang bertujuan untuk menghasilkan

sebuah proses rekayasa *software*, baik dalam bentuk model, *software*, maupun deskripsi) semua itu terdapat dalam sistem dari sebuah *software* tersebut (Aziz et al., 2022).


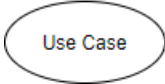

UML adalah struktur maupun teknik untuk memodelkan desain program yang berorientasi objek beserta aplikasinya. Dimana dalam pembuatan model UML ini memiliki aturan yang harus diikuti. elemen padamodel yang dibuat harus terhubung satu sama lain serta harus mengikuti standar yang terdapat pada UML bukan hanya sekedar diagram namun jugamenjelaskan konteksnya (Adyan et al, 2020).

a. *Use case Diagram*

Use case merupakan suatu bentuk pemodelan yang menjelaskan kelakuan (*behavior*). Dimana *use case* ini dapat digunakan untuk menjelaskan interaksi antara aktor dengan sistem yang dibuat (Tabrani & Priyandaru, 2021).

Tabel *Use case Diagram* ditunjukkan pada Tabel 2.3 di bawah ini:

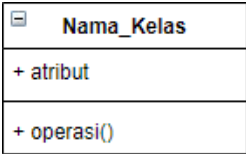

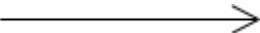
Tabel 2.3 *Usecase Diagram* (Tabrani & Priyandaru, 2021)

Nama	Simbol	Keterangan
Aktor	 Actor	Seseorang atau admin yang berhubungan dengan sistem yang sedang dibangun.
<i>Use case</i>	 Use Case	Digunakan untuk menggambarkan aktivitas apa saja yang dapat dilakukan oleh aktor.
Relasi		Relasi digunakan untuk menunjukkan relasi antara aktor dan <i>use case</i> .

b. Class Diagram

Class diagram merupakan diagram yang menjelaskan mengenai gambaran dari struktur sistem dengan mendefinisikan kelas-kelas yang nantinya akan digunakan untuk pembangunan sistem (Tabrani & Priyandaru, 2021). Tabel *Class* Diagram dapat ditunjukkan pada Tabel 2.4 di bawah ini.

Tabel 2.4 *Class* Diagram (Tabrani & Priyandaru, 2021)





Nama	Simbol	Keterangan
Kelas		Kelas pada struktur sistem.
Asosiasi (<i>Association</i>)		Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
Asosiasi berarah (<i>Directed association</i>)		Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .

c. Activity Diagram

Activity diagram merupakan diagram yang menjelaskan mengenai gambaran dari aliran aktifitas sistem (Tabrani & Priyandaru, 2021). Tabel berikut ini akan menjelaskan simbol-simbol apa saja yang terdapat pada *activity* diagram. Tabel *Activity* Diagram dapat ditunjukkan pada Tabel 2.5 di bawah ini.

Tabel 2.5 *Activity* Diagram (Tabrani & Priyandaru, 2021)


Nama	Simbol	Keterangan
------	--------	------------


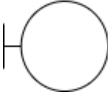

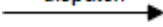
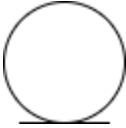

<i>Start</i> (Titik Awal)		Menunjukkan suatu Objek yang dibentuk untuk memulai aktivitas.
<i>Stop</i> (Titik Akhir)		Menyatakan berhentinya suatu aktivitas.
<i>Activity</i>		Menunjukkan kelas yang saling berinteraksi.
<i>Decision</i>		Digunakan untuk mengambil sebuah keputusan.

a. *Sequence Diagram*

Sequence diagram merupakan diagram yang menjelaskan mengenai aktivitas objek pada *use case* dengan menggambarkan waktu hidup objek dan pesan yang akan dikirim maupun diterima oleh objek (Tabrani & Priyandaru, 2021). Tabel Simbol *Sequence Diagram* dapat ditunjukkan pada Tabel 2.6 di bawah ini.

Tabel 2.6 *Sequence Diagram* (Tabrani & Priyandaru, 2021)

Nama	Simbol	Keterangan
Aktor		Mendefinisikan tentang peran <i>user</i> yang dapat berinteraksi dengan sistem yang akan dibuat.

Garis Hidup (<i>Lifeline</i>)		Garis titik-titik terputus menyatakan kehidupan suatu objek atau keterhubungan dengan suatu objek.
<i>Boundary Class</i>		Menyatakan objek yang berinteraksi antar aktor dengan sistem.
Waktu aktif		Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya.
Pesan tipe <i>create</i>		Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat.
<i>Entity Class</i>		Digunakan untuk menyimpan data atau informasi, berupa gambaran dari <i>tabel</i> .
<i>Control Class</i>		Digunakan untuk menggambarkan proses yang di jalankan oleh sistem.

2.10 Website

Menurut Yuhefizar (2021), *Website* merupakan metode untuk menampilkan informasi di internet, berupa gambar, video, teks dan suara maupun

interaktif yang menghubungkan (link) dari dokumen satu dengan dokumen lainnya (hypertext) yang bisa diakses melalui browser.

Adapun jenis – jenis website adalah sebagai berikut : *Website* statis adalah suatu *website* yang memiliki halaman yang tidak berubah. Artinya, untuk melakukan sebuah perubahan pada suatu halaman hanya dapat dilakukan secara manual, yaitu dengan mengedit kode - kode yang menjadi struktur dari *website* itu sendiri. *Website* dinamis adalah suatu *website* yang secara strukturnya diperuntukkan untuk update sesering mungkin. *Website* ini selain utamanya untuk diakses oleh para pengguna juga telah disediakan halaman back-end yaitu untuk mengedit kode dari website tersebut. *Website* interaktif adalah suatu website yang diperuntukkan untuk berinteraksi dengan orang lain secara online. Pengguna website jenis ini biasanya komunitas atau pengguna internet aktif. Pengguna di *website* ini dapat berinteraksi dan beradu argumen tentang apa yang sedang mereka pikirkan.

2.11 PHP

PHP merupakan singkatan dari PHP (Hypertext Preprocessor). PHP merupakan bahasa pemrograman script yang diletakkan dalam server yang biasa digunakan untuk membuat aplikasi web yang bersifat dinamis (Winanjar, J., & Susanti, D. 2021).

Hypertext Preprocessor (PHP) merupakan server-side programming, yaitu bahasa pemrograman yang diproses di sisi server. Fungsi utama PHP dalam membangun website adalah untuk melakukan pengolahan data pada database. Data website akan dimasukkan ke database, di-edit, dihapus, dan ditampilkan pada website yang diatur oleh PHP. PHP merupakan bahasa pemrograman universal untuk penanganan pembuatan dan pengembangan sebuah website dan bisa

digunakan bersamaan dengan HTML (Resman, K. I. K., Gunadnya, I. B. P., & Budisanjaya, I. P. G. 2021).

PHP (Hypertext Preprocessor) adalah suatu bahasa pemrograman yang digunakan untuk menterjemahkan basis kode program menjadi kode mesin yang dapat dimengerti oleh komputer yang bersifat server side yang ditambahkan ke HTML (Hari Utami, 2022).

2.12 HTML

HTML adalah suatu bahasa pemrograman yang digunakan untuk pembuatan halaman website agar dapat menampilkan berbagai informasi baik tulisan maupun gambar pada sebuah web browser. Saat ini bahasa HTML masih terus dikembangkan. Hal ini dikarenakan pengguna internet semakin hari semakin berkembang pesat. Oleh karena itu bahasa HTML harus ditingkatkan lagi agar bisa menciptakan halaman web yang lebih berkualitas (Noviantoro, A., Silviana, A. B., Fitriani, R. R., & Permatasari, H. P. 2022)

2.13 XAMPP

XAMPP merupakan *software server apache* dimana dalam XAMPP yang telah tersedia *database server* seperti MySQL dan PHP programming. XAMPP memiliki keunggulan yaitu cukup mudah dioperasikan, tidak memerlukan biaya serta mendukung instalasi pada *Windows* dan *Linux*. Keuntungan lain yang didapatkan adalah hanya dengan melakukan instalasi cukup satu kali kemudian didalamnya tersedia MySQL, *apache web server*, *database server* PHP. (Mawaddah & Fauzi, 2018)

2.14 MySQL

My Structured Query Language (MySQL) adalah suatu sistem basis data *relation* atau *Relational Database Managemnt System* (RDBMS) yang mampu bekerja secara cepat dan mudah digunakan. MySQL juga merupakan program pengakses *database* yang bersifat jaringan, sehingga dapat digunakan untuk aplikasi *multiuser* (Resman, K. I. K., Gunadnya, I.B. P., & Budisanjaya, I. P. G. 2021).

MySQL adalah salah satu jenis *database server* yang sangat populer, hal ini disebabkan karena MySQL menggunakan SQL sebagai bahasa dasar untuk mengakses *database*-nya. MySQL bersifat *Open Source, Software* ini dilengkapi dengan *Source code* (kode yang dipakai untuk membuat MySQL) (Winanjar, J., & Susanti, D. 2021).

2.15 Profil Perusahaan

PT. Grand Indo Perkasa adalah perusahaan yang bergerak di bidang pengadaan alat berat untuk industri penambangan, khususnya penambangan batu bara. Didirikan pada bulan Oktober 2016, perusahaan ini telah tumbuh menjadi salah satu penyedia terpercaya untuk berbagai jenis alat berat yang digunakan dalam operasi tambang, termasuk *excavator, bulldozer, dump truck*, dan peralatan khusus lainnya. Dengan komitmen pada kualitas dan efisiensi, PT. Grand Indo Perkasa menyediakan solusi pengadaan alat berat yang disesuaikan dengan kebutuhan klien. Kami bermitra dengan produsen alat berat terkemuka dunia untuk memastikan bahwa setiap peralatan yang kami sediakan memiliki standar tertinggi dalam hal ketahanan, performa, dan efisiensi operasional.



Gambar 2.9 Logo Perusahaan

Visi:

Menjadi penyedia alat berat terkemuka di Indonesia yang mendukung kemajuan industri penambangan batu bara melalui produk berkualitas tinggi dan layanan profesional.

Misi:

1. Memberikan solusi terbaik dalam pengadaan alat berat yang mendukung operasional penambangan secara efisien.
2. Memastikan ketersediaan peralatan yang handal dengan layanan purna jual yang profesional dan responsif.
3. Menjalin kemitraan jangka panjang dengan klien berdasarkan kepercayaan dan kepuasan.
4. Terus berinovasi untuk menyediakan peralatan terbaru yang ramah lingkungan dan efisien dalam penggunaan energi.

Layanan Utama:

- Pengadaan alat berat untuk penambangan batu bara
- Penyewaan alat berat
- Layanan purna jual, termasuk perawatan dan suku cadang
- Konsultasi teknis terkait penggunaan alat berat di sektor penambangan

Keunggulan:

- Pengalaman lebih dari 7 tahun dalam industri alat berat
- Kerjasama dengan produsen alat berat global

- Dukungan teknis dan layanan purna jual yang handal
- Jaringan distribusi yang luas di seluruh wilayah penambangan Indonesia

PT. Grand Indo Perkasa terus berkomitmen untuk memberikan kontribusi pada perkembangan sektor pertambangan batu bara di Indonesia dengan menyediakan peralatan dan layanan terbaik yang mendukung keberhasilan operasional klien kami.

2.16 Penelitian Terdahulu

1. Asmaul Husnah Nasrullah, Implementasi Algoritma *Decision Tree* Untuk Klasifikasi Produk Laris, 2021

Kesimpulan pada penelitian ini adalah mengimplementasikan algoritma *Decision Tree* dengan C4.5 untuk mengklasifikasikan produk laris. penelitian jurnal ini bertujuan untuk menguji akurasi dari algoritma C4.5 dalam melakukan klasifikasi produk laris (data privat). Adapun hasil akurasi model klasifikasi produk laris menggunakan Decision Tree C4.5 yang diperoleh dari penelitian ini adalah sebesar 90% dan nilai AUC 0.709 dimana nilai ini termasuk dalam Good Classification.

2. Rafael Nuansa Ramadhon, Aldino Ogi, Ari Permana Agung, Ryandra Putra, Siti Sarah Febrihartina, Uus Firdaus, Implementasi Algoritma Decision Tree untuk Klasifikasi Pelanggan Aktif atau Tidak Aktif pada Data Bank, 2024

Kesimpulan pada penelitian ini adalah mengimplementasikan penerapan pohon keputusan (*decision tree*) sebagai metode analisis dalam konteks pengambilan keputusan. Penelitian ini berfokus pada penerapan algoritma *decision tree* untuk klasifikasi pelanggan aktif dan tidak aktif di sektor

perbankan. Tujuan utama penelitian pada jurnal ini yaitu untuk mengidentifikasi faktor-faktor yang paling berpengaruh dalam menentukan status pelanggan, terutama dalam pemasaran layanan perbankan. Kami menggunakan dataset yang mencakup berbagai atribut diantaranya data lokasi, ID produk, nama pelanggan, saldo, dan status.

3. Syaiful Bahri, Akhyar Lubis, Metode Klasifikasi Decision Tree Untuk Memprediksi Juara *English Premier League*, 2020

Kesimpulan penelitian ini adalah untuk memprediksi juara *English Premier League*, penelitian ini menerapkan metode klasifikasi *decision tree* dengan algoritma C4.5 menggunakan *software RapidMiner 8.1*. dan Club yang akan diuji adalah *Manchester United*. Data set yang digunakan adalah data club dari tahun 1992 hingga tahun 2016, jumlah goal yang diciptakan setiap musimnya, jumlah kemasukan goal tiap musimnya, jumlah points diakhir musim dan status juara atau tidak di akhir musim. Hasil penelitian dengan algoritma C4.5 ini menunjukkan bahwa club tersebut akan mampu menjadi juara di akhir musim jika mampu menciptakan jumlah goal diatas 65 dan point yang diperoleh diatas 89 point di setiap akhir musim. Jika jumlah perolehan points club tersebut diakhir musim dibawah 65 points, maka club tersebut dipastikan tidak pernah juara diakhir musim.