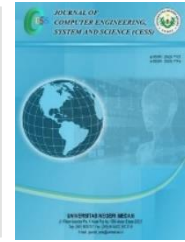


Contents list available at www.jurnal.unimed.ac.id

CESS
(Journal of Computing Engineering, System and Science)

journal homepage: <https://jurnal.unimed.ac.id/2012/index.php/cess>



Decision Support System Memprediksi Kerusakan Handphone Berbasis Web Menggunakan Algoritma C4.5 di Pioneer Service

Decision Support System Predicts Web-Based Cellphone Damage Using the C4.5 Algorithm at Pioneer Service

Maissy Masitoh Sihombing^{1*}, Mhd Basri²

^{1,2} Program Studi Sistem Informasi, Universitas Muhammadiyah Sumatera Utara Medan
Jl. Kapt. Mukhtar Basri No. 3 Medan Sumatera Utara 20238 Indonesia
email: ¹maissymasitoh9@gmail.com, ²mhd.basri@umsu.ac.id

ABSTRAK

Sistem Pendukung Keputusan (SPK) digunakan untuk memprediksi kerusakan *smartphone* di *Pioneer Service* menggunakan Algoritma C4.5. *Pioneer Service* merupakan salah satu jasa layanan perbaikan *handphone* yang setiap harinya menerima berbagai macam keluhan kerusakan *handphone*. Masalah yang dihadapi oleh *Pioneer Service* adalah banyaknya faktor yang mempengaruhi kerusakan pada *handphone* dan waktu yang diperlukan untuk menentukan jenis kerusakan pada *handphone* tersebut. Kendala lainnya adalah ketika pelanggan membutuhkan *handphone* kembali berfungsi dengan cepat. Permasalahan pada *Pioneer Service* dapat diselesaikan dengan membuat suatu Sistem Pendukung Keputusan berbasis web dengan menerapkan metode Algoritma C4.5. Aplikasi yang dirancang menggunakan *Visual Studio Code* dan *MySQL* sebagai *database*. Metodologi evaluasi yang digunakan dalam penelitian ini melibatkan pengujian sistem terhadap dataset keluhan kerusakan *handphone* untuk mengukur akurasi prediksi. Hasil dari penelitian menunjukkan bahwa sistem pendukung keputusan yang dirancang dapat meningkatkan efisiensi layanan dan kepuasan pelanggan. Penerapan metode algoritma C4.5 telah berhasil meningkatkan akurasi prediksi kerusakan *handphone*. Penerapan praktis dari Sistem Pendukung Keputusan ini juga diperjelas dengan integrasi sistem dalam operasional *Pioneer service*, memastikan manfaatnya dalam meningkatkan efisiensi layanan dan kepuasan pelanggan secara konsisten.

Kata Kunci: Sistem Pendukung Keputusan; Algoritma C4.5; Kerusakan Smartphone; Visual Studio Code, MySQL

ABSTRACT

A Decision Support System (DSS) is used to predict smartphone damage at Pioneer Service using the C4.5 Algorithm. Pioneer Service is one of the mobile phone repair service providers

*Penulis Korespondensi:
email: maissymasitoh9@gmail.com

that receives various types of damage complaints daily. The problem faced by Pioneer Service is the numerous factors affecting mobile phone damage and the time required to determine the type of damage. Another challenge is when customers need their phones to be functional again quickly. The problem at Pioneer Service can be resolved by creating a web-based Decision Support System that applies the C4.5 Algorithm. The application is designed using Visual Studio Code and MySQL as the database. The evaluation methodology used in this research involves testing the system against a dataset of mobile phone damage complaints to measure prediction accuracy. The research results show that the designed Decision Support System can improve service efficiency and customer satisfaction. The application of the C4.5 Algorithm has successfully increased the accuracy of damage predictions. The practical application of this Decision Support System is also clarified by integrating the system into the daily operations of Pioneer Service, ensuring its benefits in consistently enhancing service efficiency and customer satisfaction.

Keywords: Decision Support System; C4.5 Algorithm; Smartphone Damage; Visual Studio Code; MySQL

1. PENDAHULUAN

Pioneer Service menjadi salah satu jasa layanan perbaikan *handphone* setiap harinya. Kerusakan disebabkan oleh banyak faktor, seperti lingkungan penggunaan yang melibatkan suhu ekstrem, kelembapan, debu dan air. Kondisi perangkat yang sudah tua atau sering terjatuh juga berkontribusi, begitu pula kesalahan pengguna seperti instalasi aplikasi tidak terpecah atau gagal memperbarui perangkat lunak. Kerusakan fisik akibat benturan atau tekanan keras, serta masalah perangkat lunak seperti bug, konflik aplikasi, atau malware juga sering terjadi. Usia perangkat yang menyebabkan degradasi komponen elektronik seiring waktu menambah kompleksitas. Banyaknya faktor ini membuat diagnosa kerusakan memerlukan waktu lama, yang dapat mengurangi kepuasan pelanggan. Memahami kompleksitas ini penting untuk meningkatkan kecepatan dan akurasi diagnosa, sehingga memperbaiki *handphone* pelanggan lebih cepat dan efisien.

Pada penelitian yang dilakukan Yurmalin MZ, dkk hasil diagnosis kerusakan Lokomotif CC 201 dengan menggunakan metode *Case-Based Reasoning* dan SUPR-Q berhasil dimana penelitian menunjukkan efektivitas sistem dalam mendiagnosis masalah lokomotif CC 201 berdasarkan gejala yang diamati. Pengujian kegunaan terhadap 20 responden menggunakan metode SUPR-Q menunjukkan kemudahan penggunaan dan kepuasan sistem yang sangat baik [1]. Pada penelitian yang dilakukan Handoko, dkk hasil diagnosa kerusakan *handphone* dengan metode *Forward Chaining* membantu pengguna mengidentifikasi kerusakan dan solusi perbaikan berdasarkan gejala yang mereka alami. Dengan tingkat akurasi 88,98%, aplikasi ini memungkinkan pengguna untuk melakukan langkah-langkah awal dalam memperbaiki *handphone* mereka sebelum mencari bantuan servis profesional [2].

Algoritma C4.5 dipilih karena kemampuannya dalam mengatasi kerusakan kompleks dengan pengambilan keputusan berbasis pohon keputusan. Metode ini mampu menghasilkan model yang mudah dipahami dari data historis, meningkatkan akurasi prediksi dengan pola-pola yang kompleks. Dibandingkan dengan *Case-Based Reasoning* dan *Forward Chaining*, Algoritma C4.5 lebih holistik dan adaptif terhadap perubahan data dan gejala, memberikan

dasar yang kuat untuk optimalisasi diagnosa dan meningkatkan kepuasan pengguna dalam layanan teknis pada *handphone*.

Salah satu algoritma yang dapat digunakan untuk sistem pendukung keputusan ini adalah Algoritma C4.5. Algoritma C4.5 merupakan kelompok Algoritma *Decision Tree*. Algoritma ini mempunyai input berupa training samples dan *samples*. *Training samples* berupa data contoh yang akan digunakan untuk membangun sebuah *tree* yang telah diuji kebenarannya. Sedangkan *samples* merupakan *field-field* data yang nantinya akan digunakan sebagai parameter dalam melakukan klasifikasi data [3]. Algoritma ini juga menghitung *information gain* dari atribut untuk menentukan seberapa informatifnya dalam pemisahan data agar lebih optimal, dan menghitung *entropy* untuk heterogenitas data. Berdasarkan uraian masalah diatas, maka penulis membuat sistem pendukung keputusan (*Decision Support System*) dengan menggunakan Algoritma C4.5 untuk memprediksi kerusakan pada ponsel pintar dengan harapan dapat membuka peluang untuk peningkatan efisiensi dalam kegiatan operasional.

Penelitian ini tentang prediksi kerusakan *handphone* yang dimuat pada Pioneer Service. Data dalam penelitian ini berasal dari Toko Pioneer Service. Data yg di gunakan merupakan data mentah yang masih harus diperiksa, dibersihkan dan transformasi terlebih dahulu sebelum data di proses dengan Algoritma C4.5. Data awal dikumpulkan dari laporan perbaikan harian yang mencatat jenis kerusakan, gejala yang dialami, dan durasi perbaikan. Data tersebut kemudian diperiksa untuk mengidentifikasi dan menghapus data yang tidak lengkap atau salah. Proses pembersihan data meliputi penghapusan duplikasi, koreksi kesalahan penulisan, dan penanganan nilai hilang. Selain itu, data ditransformasi ke dalam format yang sesuai untuk analisis, termasuk pengkodean variabel kategorikal dan normalisasi nilai numerik. Langkah-langkah ini memastikan bahwa data yang digunakan untuk pemodelan dengan Algoritma C4.5 adalah akurat dan dapat diandalkan.

2. TINJAUAN TEORI

2.1. Sistem Pendukung Keputusan (*Decision Support System*)

Keputusan adalah suatu pilihan satu alternatif dari beberapa alternatif penyelesaian masalah untuk mengakhiri atau menyelesaikan masalah tersebut [4]. Sistem Pendukung Keputusan adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur. DSS dimaksud untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka [5]. Sistem ini bertujuan memfasilitasi proses pengambilan keputusan dengan menyediakan informasi yang terstruktur dan relevan [6].

2.2. Pohon Keputusan (*Decision Tree*)

Diantara beberapa metode atau teknik yang dapat digunakan untuk membuat prediksi dalam Sistem Pendukung Keputusan adalah metode *decision tree* atau pohon keputusan. Pohon Keputusan dapat diartikan sebagai suatu cara untuk memprediksi atau mengklarifikasi, yang mana pohon Keputusan dapat membagi kumpulan data yang besar menjadi himpunan-himpunan *record* yang lebih kecil dengan menerapkan serangkaian aturan [7].

Dengan kata lain, pohon keputusan bekerja dengan membentuk pohon keputusan yang dapat disimpulkan aturan-aturan klasifikasi tertentu, salah satu algoritma yang menerapkan pohon keputusan adalah algoritma C4.5 [8].

2.3. Algoritma C4.5

Algoritma C4.5 digunakan untuk membentuk pohon keputusan dan dapat memodelkan hasil temuan data penting sehingga pengetahuan atau informasi yang ditemukan mudah untuk dipahami [9]. Secara umum langkah-langkah yang dilakukan dalam membangun sebuah pohon keputusan dengan algoritma C4.5 adalah sebagai berikut:

1. Memilih atribut sebagai akar
2. Buat cabang untuk setiap nilai
3. Membagi kasus dari tiap cabang
4. Ulangi langkah yang sama untuk setiap cabang.

Proses pemilihan atribut akar sendiri, didasarkan pada suatu nilai yang dinyatakan sebagai nilai *information gain* tertinggi dari atribut-atribut yang sudah ada. *Information Gain* adalah nilai yang digunakan untuk dapat mengukur efektivitas suatu atribut dalam mengklasifikasikan data, untuk mengetahui nilai dari *Information Gain*, kita diharuskan menghitung *Entropy* terlebih dahulu. Dijelaskan juga bahwa, kita biasanya memakai *Entropy* sebagai parameter atau acuan untuk mengetahui keberagaman (heterogenitas) dari data. semakin beragam sampel pada data yang digunakan, maka nilai *Entropy* akan besar juga:

$$Gain(S, A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{|S|} * Entropy(S_i) \quad (2.1)$$

Dimana:

- S : Himpunan kasus
 A : Atribut
 n : Jumlah partisi atribut A
 |S_i| : Jumlah kasus pada partisi ke-i
 |S| : Jumlah kasus dalam S

Untuk perhitungan *entropy* sendiri, dapat dilihat pada persamaan dibawah ini

$$Entropy(S) = \sum_{i=1}^n - p_i * \log_2 p_i \quad (2.2)$$

Dimana:

- S : Kumpulan kasus
 N : Jumlah partisi S
 P_i : Proporsi dari S_i terhadap S [10].

2.4. Kerusakan pada Handphone

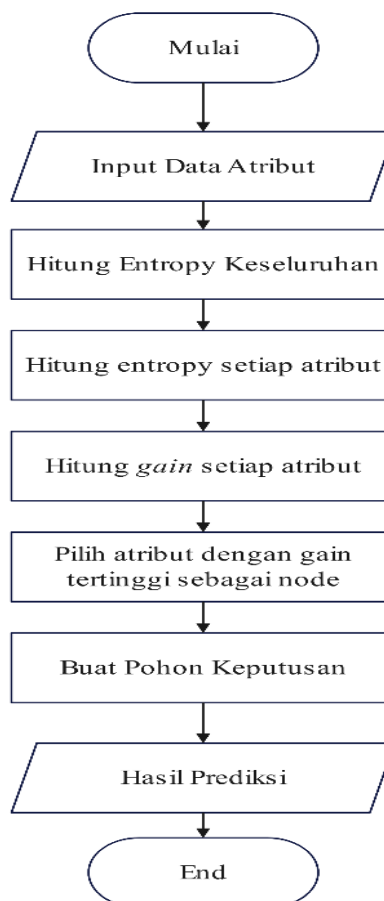
Kerusakan pada *handphone* adalah perubahan yang terjadi yang mengakibatkan menurunnya *handphone* dari kondisi awalnya, kerusakan *handphone* tidak hanya terfokus kepada perubahan pada fungsional sistem yang tidak optimal, kerusakan juga bisa terjadi pada fisik *handphone* yang membuat *handphone* tidak dapat berfungsi sebagaimana mestinya [12].

Beberapa jenis kerusakan HP yang terjadi antara lain dapat dikelompokkan sebagai berikut :

1. Kerusakan IC Power
2. Kerusakan IC Microprocessor
3. Kerusakan Sistem Operasi
4. Kerusakan IC Power Amplifier
5. Kerusakan Pada Baterai (Battery Drop)
6. Kerusakan IC eMMC
7. Kerusakan IC Charging

3. METODE

Penelitian dilakukan dengan observasi dan wawancara secara langsung di Pioneer Service untuk mendapatkan data-data yang diperlukan, yang mana data yang paling diperlukan dalam penelitian ini adalah data perbaikan *handphone* dan juga sampel data kerusakan *handphone* pada Pioneer Service pada tahun 2023 periode Januari hingga Desember. Data awal dikumpulkan dari laporan perbaikan harian yang mencatat jenis kerusakan, gejala yang dialami, dan durasi perbaikan. Data tersebut kemudian diperiksa untuk mengidentifikasi dan menghapus data yang tidak lengkap atau salah. Proses pembersihan data meliputi penghapusan duplikasi, koreksi kesalahan penulisan, dan penanganan nilai hilang. Selain itu, data ditransformasi ke dalam format yang sesuai untuk analisis, termasuk pengkodean variabel kategorikal dan normalisasi nilai numerik. Dengan Bahasa pemrograman python menyediakan library pandas yang dapat membantu pemrosesan data menjadi lebih cepat. Data yang terkumpul selanjutnya akan dianalisis menggunakan Algoritma C4.5. Dalam penelitian ini, Algoritma C4.5 dipilih karena kemampuannya dalam mengatasi kerusakan kompleks dengan pengambilan keputusan berbasis pohon keputusan. Metode ini mampu menghasilkan model yang mudah dipahami dari data historis, meningkatkan akurasi prediksi dengan pola-pola yang kompleks. Dengan menggunakan Algoritma C4.5 untuk memprediksi kerusakan pada ponsel pintar dengan harapan dapat membuka peluang untuk peningkatan efisiensi dalam kegiatan operasional. Berikut ini adalah urutan metodologi pengembangan sistem *flowchart* penelitian yang telah diterapkan oleh peneliti pada Gambar 1.



Gambar 1. Tahapan Kerangka Penelitian *Flowchart*

Berikut adalah penjelasan dari setiap langkah dalam bagan alir flowchart yang diberikan:

1. **Mulai:**
Tahap awal dari proses pembuatan sistem prediksi kerusakan handphone. Ini adalah titik awal dimana proses dimulai.
2. **Input Data Atribut:**
Data atribut yang terkait dengan handphone dan kerusakannya dimasukkan ke dalam sistem. Atribut ini bisa berupa jenis kerusakan, model handphone, usia penggunaan, kondisi fisik, dll.
3. **Hitung *Entropy* Keseluruhan:**
Entropy keseluruhan dari dataset dihitung untuk mengetahui tingkat ketidakteraturan atau ketidakpastian dalam data.
4. **Hitung *Entropy* Setiap Atribut:**
Entropy dihitung untuk setiap atribut dalam dataset. Ini membantu dalam mengukur ketidakteraturan yang ditimbulkan oleh masing-masing atribut.
5. **Hitung *Gain* Setiap Atribut:**
Gain dihitung untuk setiap atribut berdasarkan *entropy* yang telah dihitung. *Gain* menunjukkan seberapa besar pengurangan ketidakteraturan yang dapat dicapai dengan menggunakan atribut tersebut untuk memisahkan data.
6. **Pilih Atribut dengan *Gain* Tertinggi sebagai *Node*:**
Atribut dengan nilai *gain* tertinggi dipilih sebagai node (simpul) dalam pohon keputusan. Ini adalah atribut yang memberikan pengurangan ketidakteraturan terbesar dan karenanya paling berguna untuk pemisahan data.
7. **Buat Pohon Keputusan:**
Pohon keputusan dibangun dengan menggunakan node yang telah dipilih. Proses ini melibatkan pemisahan data berdasarkan atribut yang dipilih dan membuat cabang-cabang untuk setiap nilai dari atribut tersebut.
8. **Hasil Prediksi:**
Berdasarkan pohon keputusan yang telah dibuat, prediksi untuk data baru dapat dilakukan. Pohon keputusan digunakan untuk menentukan jenis kerusakan handphone berdasarkan atribut yang diberikan.
9. **End:**
Proses pembuatan dan penggunaan sistem prediksi selesai. Ini adalah titik akhir dari *flowchart*. Secara keseluruhan, *flowchart* ini menggambarkan langkah-langkah yang diperlukan untuk membangun sistem prediksi kerusakan handphone menggunakan Algoritma C4.5, mulai dari pengumpulan data atribut hingga menghasilkan prediksi akhir.

4. HASIL DAN PEMBAHASAN

4.1. Hasil Perhitungan Manual C4.5

Data yang diperoleh dari lokasi penelitian diproses terlebih dahulu. Data yang didapatkan berupa data pemeriksaan servis handphone di Pioneer Service pada tahun 2023, data terkait dapat dilihat dalam tabel dibawah ini:

Tabel 1. Tabel Data Pemeriksaan

| Tanggal Pemeriksaan | Nama Perangkat | Kerusakan |
|---------------------|--------------------|---------------------------------------|
| 02/01/2023 | Vivo V21 | IC Charging |
| 02/01/2023 | Samsung Galaxy S21 | IC Power |
| 05/01/2023 | Realme Narzo 30A | IC Mikroprosesor |
| 09/01/2023 | Samsung Galaxy S21 | IC eMMC IC PA (Power Amplifier) |
| 13/01/2023 | Samsung Galaxy S21 | Amplifier) |
| 14/01/2023 | Oppo Reno5 | IC Charging |
| 15/01/2023 | Oppo Reno5 | IC eMMC |
| 15/01/2023 | Samsung Galaxy A52 | IC Mikroprosesor |
| 21/01/2023 | Realme C21 | Baterai Drop |

(Sumber: Pioneer Service)

Dataset terdiri atas 262 baris data kerusakan untuk data latih dan 35 baris data untuk data uji, yang mencakup berbagai atribut yang disebutkan. Sekarang, data latih diatas akan mulai diolah mulai dari perhitungan *Entropy*, *Information Gain*, hingga pembuatan pohon keputusannya. Melalui data latih terkait, berikut merupakan ringkasan yang dapat berguna untuk perhitungan-perhitungan yang dibutuhkan:

Tabel 2. Daftar Kerusakan Berdasarkan Jumlah Kasus

| Kerusakan | Total Kasus |
|--|-------------|
| Kerusakan <i>IC Power</i> | 45 Kasus |
| Kasus Kerusakan IC PA | 33 Kasus |
| Kasus Kerusakan IC eMMC | 44 Kasus |
| Kasus Kerusakan IC <i>Charging</i> | 30 Kasus |
| Kasus Kerusakan IC <i>Microprocessor</i> | 38 Kasus |
| Kasus Kerusakan <i>Battery Drop</i> | 38 Kasus |
| Kasus Kerusakan Sistem Operasi | 34 Kasus |
| Total Pemeriksaan | 262 Kasus |

Sesuai dengan tabel diatas, maka perhitungan *entropy* dapat dilakukan dengan menggunakan rumus berikut ini. Dalam rumus ini, *entropy* dihitung dengan menjumlahkan hasil perkalian dari setiap kelas atau atribut kerusakan yang ada dengan logaritma basis 2 dari probabilitas tersebut, kemudian mengalihkan hasilnya dengan -1.

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

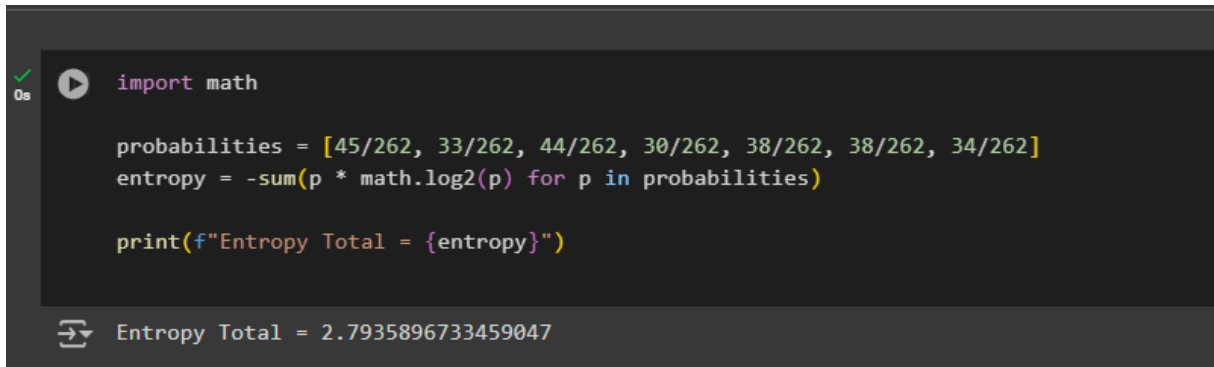
$$Entropy \text{ Total} = -\left(\frac{45}{262} \log_2 \frac{45}{262}\right) + -\left(\frac{33}{262} \log_2 \frac{33}{262}\right) + -\left(\frac{44}{262} \log_2 \frac{44}{262}\right) + \\ -\left(\frac{30}{262} \log_2 \frac{30}{262}\right) + -\left(\frac{38}{262} \log_2 \frac{38}{262}\right) + -\left(\frac{38}{262} \log_2 \frac{38}{262}\right) + -\left(\frac{34}{262} \log_2 \frac{34}{262}\right)$$

Entropy Total

$$\begin{aligned}
 &= -(0,1718 x - 2.5452) + -(0,1260 x - 2.9855) \\
 &+ -(0.1680 x - 2.5740) + -(0.1145 x - 3.1243) \\
 &+ -(0.1450 x - 2.7846) + -(0.1450 x - 2.7846) \\
 &+ -(0.1298 x - 2.9473)
 \end{aligned}$$

Entropy Total = **2,7940z**

Untuk validasi perhitungan manual diatas, dilakukan perhitungan otomatis menggunakan python dengan tampak seperti dibawah ini



```

import math

probabilities = [45/262, 33/262, 44/262, 30/262, 38/262, 38/262, 34/262]
entropy = -sum(p * math.log2(p) for p in probabilities)

print(f"Entropy Total = {entropy}")

```

Entropy Total = 2.7935896733459047

Gambar 2. Perhitungan Hasil Phyton

Nilai *entropy* diatas mencerminkan tingkat keberagaman data yang cukup tinggi, hal ini sesuai dengan data yang memiliki berbagai macam ciri-ciri kerusakan dan jenis kerusakan yang ingin diprediksi.

Setelah menghitung jumlah *Entropy* total, selanjutnya, nilai *information gain*, setiap atribut-atribut yang berbeda akan dihitung. Berikut adalah langkah perhitungan *entropy* untuk atribut "Kesalahan sistem yang terus menerus muncul".

Atribut muncul bernilai ya sebanyak 17 kali dan muncul dalam label kerusakan dibawah ini:

- Kerusakan Sistem Operasi: 17 kali
- Atribut bernilai tidak muncul sebanyak 245 kali dan tersebar dalam label kerusakan berikut ini:

Tabel 3. Sebaran Kerusakan Berdasarkan Data

| Jenis Kerusakan | Jumlah Muncul |
|--------------------------|---------------|
| Battery Drop | 38 |
| IC Charging | 30 |
| IC eMMC | 44 |
| IC Mikroprosesor | 38 |
| IC Power Amplifier | 33 |
| IC Power | 45 |
| Kerusakan Sistem Operasi | 17 |

Langkah-langkah perhitungan Information Gain untuk atribut tersebut adalah sebagai berikut:

- Hitung *Entropy* total untuk atribut "Kesalahan system yang terus menerus muncul" bernilai "ya"

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

$$Entropy\ Total = -\left(\frac{17}{17} \text{Log}_2 \frac{17}{17}\right)$$

$$Entropy\ Total = -(1 \times 0)$$

$$Entropy\ Total = 0$$

2. Hitung *Entropy* total untuk atribut “Kesalahan sistem yang terus menerus muncul” bernilai “tidak”

Perhitungan *Entropy* untuk atribut tidak adalah sebagai berikut:

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

$$Entropy\ Total = -\left(\frac{38}{245} \text{Log}_2 \frac{38}{245}\right) - \left(\frac{30}{245} \text{Log}_2 \frac{30}{245}\right) - \left(\frac{44}{245} \text{Log}_2 \frac{44}{245}\right) +$$

$$- \left(\frac{38}{245} \text{Log}_2 \frac{38}{245}\right) + - \left(\frac{33}{245} \text{Log}_2 \frac{33}{245}\right) + - \left(\frac{45}{245} \text{Log}_2 \frac{45}{245}\right) + - \left(\frac{17}{245} \text{Log}_2 \frac{17}{245}\right)$$

$$Entropy\ Total = 0,4170 + 0,3710 + 0,4445 + 0,4170 + 0,3896 + 0,4490 + 0,2671 =$$

2,7556

3. Hitung *Information Gain*

Information gain diukur sebagai pengurangan dalam entropy setelah membagi data berdasarkan atribut, rumusnya adalah:

$$Gain = Entropy\ Total - \left(\frac{|A_i|}{|S|}\right) \times Entropy_i$$

Yang mana $|A_i|$ adalah jumlah kemunculan atribut, $|S|$ adalah total data, dan $Entropy_i$ adalah *entropy* dari subset data tersebut. Dengan menggunakan rumus diatas, perhitungan *Information Gain* nya adalah

$$Gain = 2,7940 - \left(\left(\frac{17}{262}\right) \times 0 + \left(\frac{245}{262}\right) \times 2,7556\right)$$

$$Gain = 0,2172$$

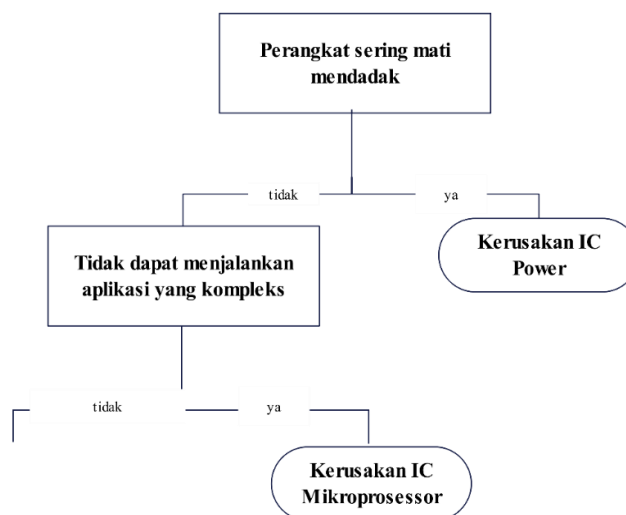
Perhitungan *gain* diatas kemudian dilanjutkan kepada 34 atribut lain dengan metode perhitungan yang sama, sehingga menghasilkan tabel berisi besar nilai *Information Gain* setiap atribut seperti dibawah ini untuk *node* pertama:

Tabel 4. Data Information Gain

| Nama Gejala | Gain |
|---|----------|
| Perangkat Sering Mati Mendadak | 0,453406 |
| Suara atau bunyi aneh saat perangkat dinyalakan | 0,370887 |
| Tidak dapat menjalankan aplikasi yang kompleks | 0,369807 |
| Tidak ada reaksi saat tombol daya ditekan | 0,355631 |
| Perangkat Tidak Menyala | 0,340718 |
| Panas berlebihan pada area mikroprosesor | 0,304091 |
| Data hilang atau tidak terbaca | 0,302659 |
| Perangkat tidak bisa diisi daya | 0,301806 |
| Perangkat menjadi sangat lambat | 0,294461 |
| Aplikasi tidak dapat dijalankan | 0,294461 |

| | |
|--|----------|
| Kesalahan sistem operasi yang tidak dapat dijelaskan | 0,288671 |
| Perangkat sering mengalami <i>hang</i> atau <i>crash</i> | 0,288671 |
| <i>Overheating</i> pada area power IC | 0,284083 |
| Kinerja Perangkat Sangat Lambat | 0,273595 |
| Baterai Tidak dapat diisi daya sepenuhnya | 0,273595 |
| Suara Terdengar kasar atau tidak stabil | 0,2673 |
| Tidak ada suara sama sekali | 0,2673 |
| Perangkat Lambat Merespons | 0,261349 |
| Perangkat tidak dapat melakukan <i>update</i> Sistem Operasi | 0,261349 |
| Perangkat tidak mendeteksi <i>charger</i> | 0,249885 |
| Kesalahan saat membuka atau menyimpan file | 0,248075 |
| Aplikasi sering keluar atau <i>crash</i> | 0,248075 |
| Distorsi suara atau suara pecah | 0,23586 |
| Perangkat panas saat digunakan dalam waktu lama | 0,23586 |
| Tidak dapat melakukan <i>update</i> atau instalasi OS | 0,231615 |
| Persentase baterai tidak akurat | 0,230217 |
| <i>Charger</i> atau <i>port</i> terasa panas saat digunakan | 0,217536 |
| Pengisian berlangsung sangat lambat | 0,217536 |
| Kesalahan Sistem Yang Terus Menerus Muncul | 0,21677 |
| Baterai mengalami pembengkakan atau deformasi | 0,216314 |
| Suara keluar dari perangkat sangat lemah | 0,205802 |
| Baterai cepat habis atau tidak bertahan lama | 0,202664 |
| Perangkat sering <i>crash</i> atau <i>restart</i> sendiri | 0,202226 |
| Indikator pengisian tidak bekerja dengan benar | 0,201934 |
| Perangkat Mati Mendadak meskipun baterai masih terisi | 0,189254 |

Dari nilai *information gain* diatas, gain Perangkat Sering Mati Mendadak adalah yang paling besar. Maka atribut tersebut dijadikan sebagai *node* awal.



Gambar 3. Pohon Keputusan

Proses rekursif akan terus terjadi hingga seluruh atribut memasuki cabang pohon keputusan. Pengembangan model prediksi berbasis Algoritma C4.5 dikembangkan dengan bahasa *Python*, model dilatih dengan data yang sudah diverifikasi dan ditransformasi sebelumnya untuk membangun pohon keputusan. Model yang sudah dilatih disimpan untuk kemudian dihubungkan sebagai fitur prediksi dalam sistem.

4.1.1. Hasil Akurasi Setelah Implementasi Algoritma

Setelah implementasi dilakukan, langkah selanjutnya adalah membandingkan label kerusakan yang ada pada data uji dengan label yang diprediksi oleh algoritma yang telah dilatih untuk mengetahui performa model, metrik yang digunakan adalah *Accuracy*, *Precision*, *Recall*, dan *F1 Score*. Berikut merupakan perbandingan hasil label antara data uji dan hasil prediksi algoritma.

Tabel 5. Perbandingan Label Sebenarnya dan Hasil Prediksi

| Index | Label Sebenarnya | Label Prediksi |
|-------|--------------------------|--------------------------|
| 125 | IC Mikroprosesor | IC Mikroprosesor |
| 129 | IC Mikroprosesor | IC Mikroprosesor |
| 82 | IC eMMC | IC eMMC |
| 41 | IC Charging | IC Charging |
| 7 | Baterai Drop | IC Mikroprosesor |
| 137 | IC Mikroprosesor | IC Mikroprosesor |
| 126 | IC Mikroprosesor | IC Mikroprosesor |
| 69 | IC eMMC | IC eMMC |
| 175 | IC PA (Power Amplifier) | IC PA (Power Amplifier) |
| 195 | IC Power | IC Power |
| 14 | Baterai Drop | Baterai Drop |
| 142 | IC Mikroprosesor | IC Mikroprosesor |
| 110 | IC eMMC | IC eMMC |
| 154 | IC PA (Power Amplifier) | IC PA (Power Amplifier) |
| 212 | IC Power | IC Power |
| 79 | IC eMMC | IC eMMC |
| 44 | IC Charging | IC Mikroprosesor |
| 253 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |
| 58 | IC Charging | IC Charging |
| 208 | IC Power | IC Mikroprosesor |
| 258 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |
| 261 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |
| 28 | Baterai Drop | Baterai Drop |
| 159 | IC PA (Power Amplifier) | IC PA (Power Amplifier) |
| 70 | IC eMMC | IC eMMC |
| 22 | Baterai Drop | IC Mikroprosesor |
| 230 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |
| 91 | IC eMMC | IC eMMC |
| 67 | IC Charging | IC Charging |
| 247 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |

| 219 | IC Power | IC Power |
|-------|--------------------------|--------------------------|
| 111 | IC eMMC | IC eMMC |
| 221 | IC Power | IC Power |
| Index | Label Sebenarnya | Label Prediksi |
| 46 | IC Charging | IC Charging |
| 108 | IC eMMC | IC eMMC |
| 145 | IC Mikroprosesor | IC Mikroprosesor |
| 176 | IC PA (Power Amplifier) | IC PA (Power Amplifier) |
| 234 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |
| 182 | IC PA (Power Amplifier) | IC PA (Power Amplifier) |
| 10 | Baterai Drop | Baterai Drop |
| 99 | IC eMMC | IC eMMC |
| 156 | IC PA (Power Amplifier) | IC PA (Power Amplifier) |
| 127 | IC Mikroprosesor | IC Mikroprosesor |
| 18 | Baterai Drop | Baterai Drop |
| 246 | Kerusakan Sistem Operasi | Kerusakan Sistem Operasi |
| 31 | Baterai Drop | Baterai Drop |
| 226 | IC Power | IC Power |
| 121 | IC Mikroprosesor | IC Mikroprosesor |
| 55 | IC Charging | IC Charging |
| 186 | IC Power | IC Power |
| 214 | IC Power | IC Power |
| 224 | IC Power | IC Power |
| 20 | Baterai Drop | Baterai Drop |

Dari hasil klasifikasi diatas, kesimpulan mengenai performa model yang sudah dilatih adalah sebagai berikut:

1. Model yang dilatih memiliki akurasi 0.925 atau 92,5%, hal ini menunjukkan model memiliki performa umum yang baik dalam memprediksi keseluruhan label di dataset
2. Jumlah presisi, *recall*, dan *F1 score* ter bobot masing-masing adalah 0.950 (95%), 0.925 (92,5%), dan 0.928 (92,8%) yang menegaskan bahwa model memiliki performa baik secara keseluruhan dengan menghitung jumlah prediksi benar setiap sampel untuk masing-masing kelas dalam perhitungan.

Kesimpulan ini menunjukkan bahwa model dapat mengklasifikasikan kerusakan handphone yang ada dengan benar dengan tingkat kesalahan yang minimal, setelah ini, model siap untuk diintegrasikan kedalam aplikasi sistem.

4.1.2. Hasil Visualisasi Pohon Keputusan

Pohon keputusan yang dihasilkan dari hasil latih Algoritma C4.5 dapat dilihat dalam gambar dibawah ini:

4.1.2. Tampilan Menu Data Atribut

Tampilan ini berguna menyediakan deskripsi atau penjelasan singkat mengenai setiap atribut untuk membantu pengguna memahami konteks dan makna dari masing-masing atribut. Terdapat juga navigasi edit dan hapus yang dapat memudahkan pengeditan data apabila memiliki kesalahan penginputan atau atribut yang tidak digunakan lagi.

| No | Nama Atribut | Deskripsi | Action |
|----|---|--------------------|--|
| 1 | Kesalahan sistem yang terus menerus muncul | - | Edit Hapus |
| 2 | Perangkat Sering Mati Mendadak | tidak bisa menyala | Edit Hapus |
| 3 | suara atau bunyi aneh saat perangkat dinyalakan | tidak ada suara | Edit Hapus |

Gambar 6. Tampilan Menu Data Atribut

4.1.2. Tampilan Menu Data Kerusakan

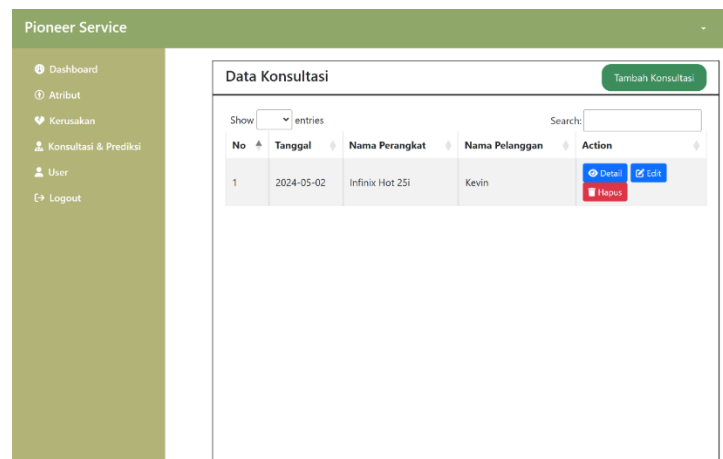
Tampilan ini berguna untuk menyediakan rincian mengenai jenis kerusakan, tanggal kerusakan, penyebab, dan langkah-langkah yang telah diambil untuk perbaikan. Menu ini juga digunakan untuk mengelola data terkait kerusakan produk atau barang. Pengguna dapat memasukkan data baru tentang kerusakan yang terjadi, memperbarui data yang ada, atau menghapus data kerusakan yang tidak relevan lagi. Jika ada sistem perbaikan atau manajemen pemeliharaan, data kerusakan ini bisa diintegrasikan untuk memastikan tindak lanjut yang cepat dan efektif.

| No | Nama Kerusakan | Deskripsi Kerusakan | Solusi Kerusakan | Action |
|----|--------------------------------|---|--|--|
| 1 | Kasus Kerusakan Battery Drop | Battery drop mengacu pada penurunan mendadak pada kapasitas baterai, sering terjadi ketika baterai cepat habis meski baru diisi daya. | Baterai yang mengalami drop biasanya perlu diganti. Juga disarankan untuk menggunakan charger asli untuk menghindari masalah serupa di masa depan. | Edit Hapus |
| 2 | Kasus Kerusakan Sistem Operasi | Kerusakan pada sistem operasi bisa disebabkan oleh file sistem yang rusak, kegagalan update, atau malware. Gejalanya termasuk boot loop, crash, atau kinerja perangkat yang lambat. | Masalah sistem operasi sering bisa diatasi dengan melakukan hard reset atau reinstalasi sistem operasi. Pastikan untuk backup data sebelum melakukan tindakan ini. | Edit Hapus |
| 3 | Kerusakan IC Charging | IC Charging mengontrol pengisian baterai perangkat. Masalah dengan IC ini bisa menyebabkan perangkat tidak mengisi daya atau mengisi daya | Ganti IC Charging dengan yang baru. Karena ini melibatkan komponen yang sangat kecil, penggantian harus dilakukan oleh teknisi | Edit Hapus |

Gambar 7. Tampilan Menu Data Kerusakan

4.1.2. Tampilan Menu Data Konsultasi

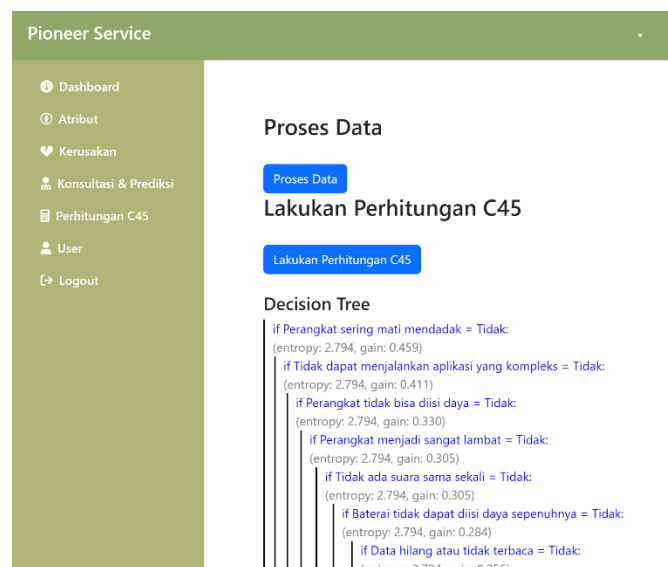
Tampilan Menu Data Konsultasi berisi daftar konsultasi yang sudah tercatat dalam sistem, terdapat navigasi menuju detail konsultasi untuk melihat masing-masing konsultasi secara detail dan hapus konsultasi. Keamanan data konsultasi melibatkan enkripsi data saat transit dan saat disimpan, serta pseudonimisasi dan anonimisasi data untuk melindungi identitas individu. Kebijakan retensi data yang tepat dan metode penghapusan aman diterapkan untuk mengelola siklus hidup data.



Gambar 8. Tampilan Menu Data Konsultasi

4.1.2. Tampilan Menu Perhitungan Algoritma C4.5

Menu ini digunakan untuk membentuk model pohon keputusan menggunakan algoritma C4.5 berdasarkan data yang telah dimasukkan. Pengguna dapat mengatur parameter seperti nilai gain ratio, pruning, dan lain-lain untuk menyesuaikan algoritma dengan kebutuhan analisis. Setelah parameter ditentukan, pengguna dapat menjalankan algoritma untuk melihat hasilnya, termasuk pohon keputusan yang dihasilkan dan akurasinya. Jika ingin memprediksi kategori produk yang paling mungkin mengalami kerusakan berdasarkan atribut tertentu, Anda akan menggunakan menu ini untuk membangun model prediksi tersebut.



Gambar 9. Tampilan Menu Perhitungan Algoritma C4.5

5. KESIMPULAN

Penerapan metode Algoritma C4.5 mendapatkan akurasi baik dalam memprediksi kerusakan HP berdasarkan data atribut dan historis Pioneer Service. Akurasi diukur dengan membandingkan prediksi model dengan label sebenarnya, memastikan validitas dalam aplikasi praktis. Sistem ini dapat dijalankan di laptop dengan spesifikasi minimum: prosesor dual-core, RAM 4GB, dan penyimpanan 10GB, sehingga mudah diimplementasikan. Prediksi

akurat ini meningkatkan kepuasan pelanggan Pioneer Service dengan diagnosa cepat dan tepat, mengurangi waktu tunggu, dan meningkatkan efisiensi operasional. Hasil ini dapat digunakan untuk menyusun strategi layanan yang lebih baik dan diverifikasi layanan. Langkah selanjutnya meliputi evaluasi pengembangan sistem untuk lebih mengoptimalkan layanan, memberikan dampak positif bagi Pioneer Service dan pelanggannya.

REFERENSI

- [1] Yurmalin, S. Rahayu, and E. B. Jemmy, "Penerapan Case-Based Reasoning dan SUPR-Q dalam Sistem Diagnosis Kerusakan pada Lokomotif CC 201," 2024. <https://doi.org/10.33369/pseudocode.11.1.7-14>.
- [2] R. T. Handako, "Aplikasi Diagnosa Kerusakan Pada Handphone Menggunakan Metode Forward Chaining," *Jurnal Teknologi Pintar*, vol.2, no.5, 2022.
- [3] W. Nurhadi, E. Marselina, And F. Mujatia, "Penerapan Algoritma Decision Tree C.45 Untuk Klasifikasi Data Status Huni Rumah Rehabilitasi Pasca Erupsi Merapi," 2020. [Online]. Available: <https://prosiding.respati.ac.id/index.php/PSN/article/view/309>
- [4] D. Pribadi, R.A. Saputra, And J.M. Hudin, "Sistem Pendukung Keputusan in Graha Ilmu," ISBN, 2020.
- [5] H. A. Septilia, Styawati, "Sistem Pendukung Keputusan Pemberian Dana Bantuan Menggunakan Metode AHP," *Jurnal Teknologi dan Sistem Informasi (JTSI)*, vol.1, no.2, 2020.
- [6] Lusiana, A. Sianturi, N. A. And Yulianty, "Penerapan Metode Peramalan (Forecasting) Pada Permintaan Atap di PT X. Industri Inovatif," 2020. doi: <https://doi.org/10.36040/industry.v10i1.2530>.
- [7] H. Hafijan, And A. N. Putri, "Penerapan Metode Klasifikasi Decision Tree Pada Status Gizi Balita di Kabupaten Simalungun," *KESATRIA: Jurnal Penerapan Sistem Informasi Komputer dan Jaringan*, vol.1, no.2, 2020.
- [8] N. Azwant, "Analisa Algoritma C4.5 Untuk Memprediksi Penjualan Motor Pada PT. Capella Dinamik Nusantara Cabang Muka Kuning," *Jurnal Ilmiah Ilmu Komputer*, vol.13, no.1, 2018.
- [9] D. Y. H. Tanjung, "Optimalisasi Algoritma C4.5 untuk Prediksi Kerusakan Mesin ATM," 2021. doi: 10.22303/Infosys.6.1.2021.12-21.
- [10] Suyanto, "Artificial Intelligence," ISBN, 2021.
- [11] S. N. Arif, M. Syahril, S. Kusnasari, And H. Winata, "Sistem Pakar Mendiagnosa Kerusakan Handphone Oppo Dengan Menggunakan Teorema Bayes," *Jurnal Teknologi Sistem Informasi dan Sistem Komputer TGD (J-SISKO)*, vol.4, no.1, 2021.