

PENERAPAN CASE BASED REASONING UNTUK MENDIAGNOSIS KERUSAKAN PADA KOMPUTER DESKTOP MENGGUNAKAN METODE NAIVE BAYES

Sebastianus A.S. Mola¹, Jakha Rama Phosa², dan Yelly Y. Nabuasa³

^{1,2,3}Ilmu Komputer, Universitas Nusa Cendana, Indonesia,

Email: adimola@staf.undana.ac.id¹, phosajakha@gmail.com², yellynabuasa@staf.undana.ac.id³

Abstrak

Kemajuan teknologi di bidang komputer saat ini sangatlah pesat, banyak keuntungan dari kemudahan informasi yang dapat diakses dengan mudah dan cepat melalui komputer. Menurut data yang tercatat di Badan Pusat Statistik Indonesia, kepemilikan komputer dalam rumah tangga pada tahun 2020 mengalami kenaikan menjadi 18,83%. Terkadang masalah muncul di mana komputer mengalami kerusakan saat digunakan yang berdampak pada berkurangnya produktivitas. Kerusakan pada komputer dapat diakibatkan oleh usia komputer, lingkungan dan penyebab lain yang disengaja maupun tidak disengaja. Pengetahuan yang buruk tentang komponen komputer dapat meningkatkan kebutuhan akan layanan teknisi saat terjadi kerusakan pada komputer. Kerusakan komputer dapat di diagnosis lebih cepat dan mudah dengan menggunakan sistem Case Based Reasoning (CBR). Salah satu metode yang sering digunakan dalam Case Based Reasoning adalah Naive Bayes. Naive bayes merupakan sebuah metode klasifikasi probabilistik sederhana untuk menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Keuntungan naive bayes adalah bahwa metode ini hanya membutuhkan jumlah data pelatihan (training data) yang kecil untuk menentukan estimasi parameter yang diperlukan dalam proses pengklasifikasian. Berdasarkan hasil pengujian menggunakan k-fold cross validation terhadap 200 data, didapatkan nilai rata-rata akurasi 86% dan rata-rata similarity 70% dalam waktu 0,0403 detik pada pengujian menggunakan indexing dan nilai rata-rata akurasi 85,5% dan rata-rata similarity 74% dalam waktu 0,0505 detik pada pengujian menggunakan nonindexing.

Kata Kunci: Case Based Reasoning, Kerusakan Komputer, k-Fold Cross Validation.

Abstract

Technological advances in the field of computers are currently advancing very fast, many benefits from current speed of information spread can be observed easily and quickly through computers. According to documented at the Indonesian Central Bureau of Statistics, household computer ownership in 2020 has increased to 18.83%. When technical problems happen in the devices there may be observed loss in productivity, damage to a computer can result from computer age, environment and various other causes, Poor knowledge of computer components can increase the need for a technician's services, when a computer breakdown occurs. Computer damage can be diagnosed more quickly and easily by using the Case Based Reasoning (CBR) system. One method that is often used in Case Based Reasoning is Naive Bayes. Naive Bayes is a simple probabilistic classification method for calculating a set of probabilities by summing the frequencies and combinations of values from a given dataset. The advantage of Naive Bayes is that this method only requires a small amount of training data to determine the parameter estimates needed in the classification process. Based on the test results using k-fold cross validation on 200 data, an average value of 86% accuracy and an average similarity of 70% in 0.0403 seconds was obtained in the test using indexing and an average value of 85.5% accuracy and an average 74% similar average in 0.0505 seconds in testing using non-indexing.

KeyWords: Case Based Reasoning, Kerusakan Komputer, k-Fold Cross Validation.

I. PENDAHULUAN

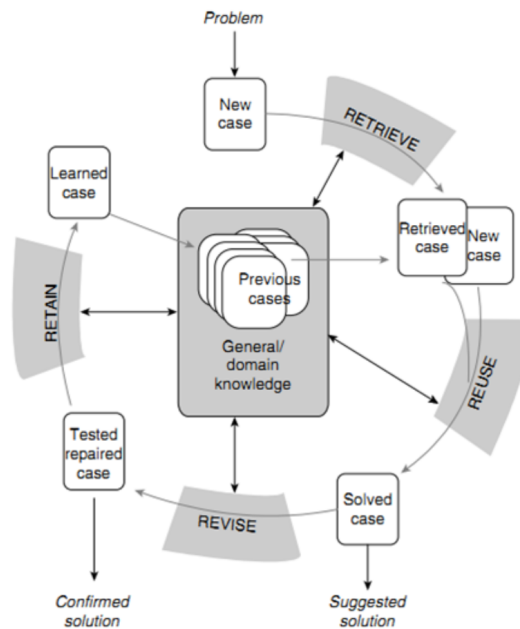
Globalisasi telah memasuki era baru yang dikenal dengan revolusi industri 4.0 [1]. Revolusi industri 4.0 membawa kita di mana kita harus selalu menggunakan teknologi dalam dunia industri contohnya seperti penggunaan komputer. Kemajuan teknologi di bidang komputer saat ini sangatlah pesat, banyak keuntungan dari kemudahan informasi yang dapat diakses dengan mudah dan cepat melalui komputer. Hanya sebagian kecil pemilik komputer yang memahami perangkat komputer dengan baik karena sebagian besar pengguna komputer hanya tertarik pada tugas menggunakannya, sehingga masalah teknis akan diserahkan kepada teknisi.

Salah satu cara yang dapat digunakan untuk menyelesaikan masalah yang ada adalah dengan menggunakan Case Based Reasoning (CBR), Case Based Reasoning adalah pendekatan untuk memecahkan masalah dengan menyoroti peran pengalaman sebelumnya. Masalah baru dapat diselesaikan dengan menata ulang dan menyesuaikan insiden yang telah menyebabkan persamaan [2]. Naive bayes adalah salah satu metode dalam CBR yang merupakan sebuah pengklasifikasian probabilistik sederhana yang menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari data set yang diberikan [3]. CBR digunakan pada beberapa penelitian seperti pada penelitian [4] yang bertujuan untuk membangun sebuah sistem untuk diagnosis perawatan kulit wajah. Metode naive bayes juga digunakan dalam beberapa penelitian seperti pada penelitian [5] yang bertujuan untuk membangun sebuah sistem yang dapat membantu dalam menentukan kesesuaian lahan pertanian untuk tanaman cabai. Adapun pada penelitian lainnya dengan metode naive bayes seperti pada penelitian [6] yang bertujuan untuk membangun sebuah sistem yang dapat mendiagnosis penyakit pada ternak babi dan pada penelitian [7] yang digunakan untuk mendapatkan nilai akurasi dalam penentuan kelayakan sekolah perawatan.

II. METODE

A. Case Based Reasoning

Case based reasoning (CBR) adalah suatu teknik penyelesaian masalah (*problem solving*) yang didasarkan pada solusi dari permasalahan sebelumnya, menggunakan informasi dan situasi tersebut untuk menyelesaikan masalah [2]. CBR memiliki siklus yang disingkat menjadi 4R seperti terlihat pada Gambar 1.



Gambar 1: Siklus CBR

Berdasarkan Gambar 1 CBR memiliki 4 tahap yang dapat dijelaskan sebagai berikut:

- 1) *Retrieve* proses pencocokan atau pengambilan kembali data maupun informasi yang sudah ada pada kasus-kasus sebelumnya dengan mendeskripsikan beberapa masalah dengan tingkat kompatibilitas tertinggi. Proses *retrieve* membahas identifikasi, pencocokan awal, pencarian dan pemilihan serta eksekusi. Penilaian kesamaan (*similarity assessment*) merupakan acuan yang digunakan sebagai pertimbangan untuk menentukan pilihan dalam banyak pilihan.
- 2) *Reuse* didasarkan pada bobot kesamaan yang paling relevan dari kasus-kasus lama ke kasus baru, lalu akan menghasilkan usulan solusi yang mungkin akan memerlukan suatu adaptasi dengan masalah yang baru.
- 3) *Revise* merupakan peninjauan kembali solusi yang telah diusulkan dari proses sebelumnya, kemudian akan diuji kembali dalam kasus *actual* (nyata) dan jika perlu solusi dapat dievaluasi atau direvisi untuk menyesuaikan dengan kasus yang baru agar lebih akurat.
- 4) *Retain* atau proses terakhir dalam siklus CBR akan menyimpan kasus baru dengan solusi agar kasus yang mirip dengan kasus ini bisa digunakan di kemudian hari.

B. Naïve Bayes

Naïve bayes merupakan sebuah metode klasifikasi probabilistik sederhana untuk menghitung sekumpulan probabilitas dengan menjumlahkan frekuensi dan kombinasi nilai dari dataset yang diberikan. Teorema ini dikemukakan oleh ilmuwan Inggris, Thomas Bayes, dengan memprediksi peluang di masa depan berdasarkan pengalaman di masa sebelumnya [8]. Rumusan *naïve bayes* diturunkan dari hukum perkalian tentang probabilitas yang bersifat kondisional, di mana dua peristiwa dikatakan memiliki hubungan bersyarat jika peristiwa yang satu mempunyai syarat peristiwa yang lain [9]. Misalkan terdapat basis kasus S yang mempunyai n kelas, (C_1, C_2, \dots, C_n) yang mana $C_i \cup C_j = \phi, i \neq j, i, j = 1, 2, 3, \dots, n$ dan $\sum_{i=1}^n P(C_i) = 1$, dimana $P(i)$ merupakan probabilitas dari C_i Jika diketahui data X didalam data uji S yang memiliki kondisi probabilitas $P(i), i = 1, 2, \dots, n$ Untuk menghitung probabilitas posterior dari C_i yaitu $P(C_i|X, i = 1, 2, \dots, n)$ Persamaan naive bayes dapat dilihat pada Persamaan (1).

$$P(X) = \frac{P(C_j)P(C_i)}{P(X)} = \frac{P(C_j)P(C_i)}{\sum_{i=1}^n P(C_j)P(C_i)} \quad (1)$$

Langkah – langkah perhitungan naïve bayes sebagai berikut:

- 1) Langkah pertama menghitung nilai probabilitas *prior* pada setiap hipotesis kelas dimana $P(C)$ adalah peluang dari kelas kasus C (jenis kerusakan), jumlah kemunculan setiap kelas dibagi jumlah dataset. Untuk menghitung nilai probabilitas *prior* dapat digunakan Persamaan (2).

$$P(C_i) = C_i \vee \frac{---}{Sv} \quad (2)$$

- 2) Langkah kedua yaitu menghitung probabilitas *posterior* dimana atribut X akan diuji dengan kelas C dimana $P(x|C)$ merupakan probabilitas atribut X terhadap terpilihnya hipotesis kelas C (probabilitas *posterior*), jumlah atribut X pada kelas C dibagi jumlah kelas C . Untuk menghitung nilai probabilitas *posterior* dapat digunakan Persamaan (3).

$$P(C_i) = \frac{n(C_i \vee C_i)}{C_i} \quad (3)$$

- 3) Langkah ketiga yaitu menghitung probabilitas *posterior* untuk mendapatkan hasil akhir dari atribut X terhadap kelas C dengan cara mengalikan setiap hasil probabilitas setiap *posterior*. Untuk menghitung langkah ketiga dapat digunakan Persamaan (4).

$$P(C_i) = \prod_{j=i}^n P(X_i \vee C_i) \quad (4)$$

- 4) Langkah keempat yaitu menghitung semua hasil probabilitas *posterior* dari hasil langkah kedua dan dikalikan dengan probabilitas *prior* dari hasil langkah pertama. Untuk menghitung nilai akhir dapat digunakan Persamaan (5).

$$P(C_i)P(C_i) \quad (5)$$

Keterangan :

C	=	Hipotesis data X yang merupakan kelas
S	=	Basis kasus data (dataset)
X	=	Atribut dengan kelas yang belum diketahui
$P(C X)$	=	Peluang hipotesis C jika terdapat atribut X (probabilitas <i>posterior</i>)
$P(X C)$	=	Peluang atribut X terhadap terpilihnya hipotesis C
$P(C)$	=	Peluang hipotesis C (probabilitas <i>prior</i>)
$P(X)$	=	Peluang atribut X yang muncul (<i>evidence</i>)

C. K-Nearest Neighbour

Algoritma KNN adalah algoritma yang digunakan untuk melakukan klasifikasi terhadap suatu objek, berdasarkan K buah data latih yang jaraknya paling dekat dengan objek tersebut. Syarat nilai K adalah tidak boleh lebih besar dari jumlah data latih, dan nilai K harus ganjil dan lebih dari satu [10]. Dekat atau jauhnya jarak data latih dengan objek yang akan diklasifikasi dapat dihitung dengan menggunakan Persamaan (6).

$$SM_{pq} = \frac{\alpha (common)}{\alpha (common) + \beta (different)} \quad (6)$$

Keterangan:

SM	=	Similarity Measure
P	=	Kasus baru
Q	=	Kasus yang tersimpan di basis kasus
$\alpha (common)$	=	Fungsi jumlah atribut yang sama
$\beta (different)$	=	Fungsi jumlah atribut yang berbeda

Atribut yang sama pada kasus lama dan kasus baru akan diberi nilai 1 dan atribut kasus baru dan kasus lama yang tidak sama maka akan diberikan nilai 0. Perhitungan yang digunakan dalam proses penelusuran kemiripan (*similarity*) antara kasus lama dengan kasus baru yaitu perhitungan *similarity* menggunakan metode KNN dengan $K=1$ artinya nilai yang diambil adalah 1 kasus dengan nilai *similarity* tertinggi.

III. HASIL DAN PEMBAHASAN

Penelitian ini menghasilkan sebuah sistem penalaran berbasis kasus diagnosa kerusakan komputer yang sesuai dengan tujuan penelitian. Berdasarkan data yang diperoleh dari Toko Palapa Komputer sebanyak 200 data, terdapat 32 jenis kerusakan dan 66 gejala. Pengujian data dilakukan menggunakan *k-fold cross validation*.

A. Ilustrasi Perancangan Sistem Penalaran Berbasis Kasus

Contoh kasus, misalkan ada sebuah kasus baru dengan data kasus sebagai berikut.

- 1) *Hard-disk* tidak terdeteksi
- 2) Komputer sering hang
- 3) Komputer terasa lambat/berat
- 4) Terdengar suara aneh pada *hard-disk*

Tahap awal yang dilakukan adalah Sistem PBK akan mengambil kasus yang paling mirip (*retrieve*) pada saat kasus baru muncul. Kasus yang paling mirip tersebut akan digunakan kembali (*reuse*) untuk mendapatkan solusi yang tepat bagi kasus baru. Tahap *revise* akan menguatkan solusi yang didapatkan dari kemiripan kasus tersebut sehingga kasus baru dapat disimpan dalam basis kasus untuk digunakan pada kasus baru lainnya dan tahap ini disebut dengan *retain*.

Berikut rincian ilustrasi perancangan penalaran berbasis kasus.

- Perancangan Proses *Retrieve*

Proses *retrieve* merupakan tahapan untuk menemukan kembali kasus yang paling mirip atau sama dengan kasus yang baru yang dimasukkan. Tahap pertama adalah dilakukan proses *indexing* menggunakan metode *naive bayes*, pada saat proses *indexing* selesai dilanjutkan dengan tahapan untuk mencari nilai kemiripan. Perhitungan nilai kemiripan menggunakan metode *k-nearest neighbor* berdasarkan Persamaan (6), di mana dilakukan perhitungan nilai *similarity* satu persatu terhadap setiap kasus yang memiliki nilai indeks yang sama.

- Perancangan *Indexing*

Indexing adalah tahap awal dalam proses menggunakan metode *naive bayes*. Hasil akhir dari proses *indexing* ini berupa hasil diagnosa kerusakan komputer. Pada tahap ini akan dihitung prediksi kasus kerusakan pada komputer berdasarkan dataset yang telah ditentukan. Dataset kasus kerusakan komputer dapat dilihat pada Tabel I.

Tabel I: Dataset kasus kerusakan pada komputer

No	Gejala	Kelas Kasus
1	Komputer terasa lambat/berat Komputer sering hang Komputer blue screen saat dinyalakan Beberapa file corrupt/hilang dalam hard disk	Hard Disk Bad sector (C1)
2	Komputer terasa lambat/berat Hard disk tidak terdeteksi Komputer sering hang Beberapa file corrupt/hilang dalam hard disk Terdengar suara aneh pada hard disk	Hard Disk Bad sector (C1)
3	Komputer terasa lambat/berat Hard disk tidak terdeteksi Komputer sering hang Komputer blue screen saat dinyalakan Terdengar suara aneh pada hard disk	Hard Disk Bad sector (C1)
4	Komputer terasa lambat/berat Komputer blue screen saat dinyalakan Hard disk tidak terdeteksi Beberapa file corrupt/hilang dalam hard disk Terdengar suara aneh pada hard disk	Hard Disk Bad sector (C1)
5	Komputer sering hang Komputer blue screen saat dinyalakan Beberapa file corrupt/hilang dalam hard disk Terdengar suara aneh pada hard disk	Hard Disk Bad sector (C1)
6	Hard disk tidak terdeteksi Hard disk terasa sangat panas	Board Hard Disk Terbakar (C2)
7	Komputer sering hang Hard disk tidak terdeteksi Hard disk tidak merespon	Hard Disk Failure (C3)
8	Komputer sering hang Pesan “Hard disk Error” / “Hard disk Failure” saat booting Hard disk tidak merespon	Hard Disk Failure (C3)
9	Komputer sering hang Hard disk tidak terdeteksi Pesan “Hard disk Error” / “Hard disk Failure” saat booting	Hard Disk Failure (C3)
10	Komputer terasa lambat/berat Komputer sering hang Hard disk tidak terdeteksi	Hard Disk Corrupt (C4)
11	Komputer terasa lambat/berat Beberapa file corrupt/hilang dalam hard disk Komputer sering hang	Hard Disk Corrupt (C4)
12	Hard disk tidak terdeteksi Hard disk tidak merespon	Head Hard Disk Rusak (C5)

Langkah-langkah perhitungan contoh kasus baru menggunakan metode naive bayes adalah sebagai berikut:

1) Hitung nilai $P(C_i)$ menggunakan Persamaan (2)

- $P(C_1) = P(\text{Diagnosa} = \text{"Hard disk bad sector"}) = \frac{|C_1|}{|S|} = \frac{5}{12} = 0,41$
- $P(C_2) = P(\text{Diagnosa} = \text{"Board hard disk terbakar"}) = \frac{|C_2|}{|S|} = \frac{1}{12} = 0,08$
- $P(C_3) = P(\text{Diagnosa} = \text{"Hard disk failure"}) = \frac{|C_3|}{|S|} = \frac{3}{12} = 0,25$
- $P(C_4) = P(\text{Diagnosa} = \text{"Hard disk corrupt"}) = \frac{|C_4|}{|S|} = \frac{2}{12} = 0,16$
- $P(C_5) = P(\text{Diagnosa} = \text{"Head hard disk rusak"}) = \frac{|C_5|}{|S|} = \frac{1}{12} = 0,08$

2) Hitung nilai $P(x_j|C_i)$ menggunakan Persamaan (3), $i = 1,2,3,4,5$ dan $j = 1,2,3,4$

a. Hard disk tidak terdeteksi

- $P(C_1) = P(\text{Gejala} = \text{"Hard disk tidak terdeteksi"} \mid \text{Diagnosa} = \text{"Hard disk bad sector"}) = \frac{3}{5} = 0.6$
- $P(C_2) = P(\text{Gejala} = \text{"Hard disk tidak terdeteksi"} \mid \text{Diagnosa} = \text{"Board hard disk terbakar"}) = \frac{1}{1} = 1$
- $P(C_3) = P(\text{Gejala} = \text{"Hard disk tidak terdeteksi"} \mid \text{Diagnosa} = \text{"Hard disk failure"}) = \frac{2}{3} = 0,6$
- $P(C_4) = P(\text{Gejala} = \text{"Hard disk tidak terdeteksi"} \mid \text{Diagnosa} = \text{"Hard disk corrupt"}) = \frac{1}{2} = 0.5$
- $P(C_5) = P(\text{Gejala} = \text{"Hard disk tidak terdeteksi"} \mid \text{Diagnosa} = \text{"Head hard disk rusak"}) = \frac{1}{1} = 1$

b. Komputer sering hang

- $P(C_1) = P(\text{Gejala} = \text{"Komputer sering hang"} \mid \text{Diagnosa} = \text{"Hard disk bad sector"}) = \frac{3}{5} = 0.6$
- $P(C_2) = P(\text{Gejala} = \text{"Komputer sering hang"} \mid \text{Diagnosa} = \text{"Board hard disk terbakar"}) = \frac{0}{1} = 0$
- $P(C_3) = P(\text{Gejala} = \text{"Komputer sering hang"} \mid \text{Diagnosa} = \text{"Hard disk failure"}) = \frac{3}{3} = 1$
- $P(C_4) = P(\text{Gejala} = \text{"Komputer sering hang"} \mid \text{Diagnosa} = \text{"Hard disk corrupt"}) = \frac{3}{2} = 1.5$
- $P(C_5) = P(\text{Gejala} = \text{"Komputer sering hang"} \mid \text{Diagnosa} = \text{"Head hard disk rusak"}) = \frac{0}{1} = 0$

c. Komputer terasa lambat/berat

- $P(X_3|C_1) = P(\text{Gejala} = \text{"Komputer terasa lambat/berat"} \mid \text{Diagnosa} = \text{"Hard disk bad sector"}) = \frac{4}{5} = 0.8$
- $P(X_3|C_2) = P(\text{Gejala} = \text{"Komputer terasa lambat/berat"} \mid \text{Diagnosa} = \text{"Board hard disk terbakar"}) = \frac{0}{1} = 0$
- $P(X_3|C_3) = P(\text{Gejala} = \text{"Komputer terasa lambat/berat"} \mid \text{Diagnosa} = \text{"Hard disk failure"}) = \frac{0}{3} = 0$
- $P(X_3|C_4) = P(\text{Gejala} = \text{"Komputer terasa lambat/berat"} \mid \text{Diagnosa} = \text{"Hard disk corrupt"}) = \frac{2}{2} = 1$
- $P(X_3|C_5) = P(\text{Gejala} = \text{"Komputer terasa lambat/berat"} \mid \text{Diagnosa} = \text{"Head hard disk rusak"}) = \frac{0}{1} = 0$

d. Terdengar suara aneh pada hard disk

- $P(C_1) = P(\text{Gejala} = \text{"Terdengar suara aneh pada hard disk"} \mid \text{Diagnosa} = \text{"Hard disk bad sector"}) = \frac{4}{5} = 0.8$
- $P(C_2) = P(\text{Gejala} = \text{"Terdengar suara aneh pada hard disk"} \mid \text{Diagnosa} = \text{"Board hard disk terbakar"}) = \frac{0}{1} = 0$
- $P(C_3) = P(\text{Gejala} = \text{"Terdengar suara aneh pada hard disk"} \mid \text{Diagnosa} = \text{"Hard disk failure"}) = \frac{0}{3} = 0$
- $P(C_4) = P(\text{Gejala} = \text{"Terdengar suara aneh pada hard disk"} \mid \text{Diagnosa} = \text{"Hard disk corrupt"}) = \frac{0}{2} = 0$
- $P(C_5) = P(\text{Gejala} = \text{"Terdengar suara aneh pada hard disk"} \mid \text{Diagnosa} = \text{"Head hard disk rusak"}) = \frac{0}{1} = 0$

3) Hitung nilai $P(C_i)$ menggunakan Persamaan (4), $i = 1,2,3,4,5$

- $P(C_1) = \prod_{j=1}^4 P(x_j|C_1) = P(\text{Diagnosa} = \text{"Hard disk bad sector"}) = 0,6 * 0,8 * 0,8 * 0,8 = 0,3072$
- $P(C_2) = \prod_{j=1}^4 P(x_j|C_2) = P(\text{Diagnosa} = \text{"Board hard disk terbakar"}) = 1 * 0 * 0 * 0 = 0$
- $P(C_3) = \prod_{j=1}^4 P(x_j|C_3) = P(\text{Diagnosa} = \text{"Hard disk failure"}) = 0,6 * 1 * 0 * 0 = 0$
- $P(C_4) = \prod_{j=1}^4 P(x_j|C_4) = P(\text{Diagnosa} = \text{"Hard disk corrupt"}) = 0,5 * 1 * 1 * 0 = 0$
- $P(C_5) = \prod_{j=1}^4 P(x_j|C_5) = P(\text{Diagnosa} = \text{"Head hard disk rusak"}) = 1 * 0 * 0 * 0 = 0$

4) Hitung nilai $P(C_i)P(C_i)$ menggunakan Persamaan (5), $i = 1,2,3,4,5$

- $P(C_1)P(C_1) = P(\text{Diagnosa} = \text{"Hard disk bad sector"}) P(\text{Diagnosa} = \text{"Hard disk bad sector"}) = 0,3072 * 0,41 = 0,12$
- $P(C_2)P(C_2) = P(\text{Diagnosa} = \text{"Board hard disk terbakar"}) P(\text{Diagnosa} = \text{"Board hard disk terbakar"}) = 0 * 0,08 = 0$
- $P(C_3)P(C_3) = P(\text{Diagnosa} = \text{"Hard disk failure"}) P(\text{Diagnosa} = \text{"Hard disk failure"}) = 0 * 0,25 = 0$
- $P(C_4)P(C_4) = P(\text{Diagnosa} = \text{"Hard disk corrupt"}) P(\text{Diagnosa} = \text{"Hard disk corrupt"}) = 0 * 0,16 = 0$
- $P(C_5)P(C_5) = P(\text{Diagnosa} = \text{"Head hard disk rusak"}) P(\text{Diagnosa} = \text{"Head hard disk rusak"}) = 0 * 0,08 = 0$

Berdasarkan perhitungan menggunakan metode naive bayes dengan membandingkan hasil dari setiap kelas dari C_1 hingga C_5 , dapat diperoleh hasil nilai bobot terbesar dimiliki oleh gejala *hard disk bad sector* atau C_1 dan dapat disimpulkan bahwa kasus baru tersebut masuk dalam diagnosa kerusakan *hard disk bad sector*.

Pada tahap *indexing* telah diprediksi kasus kerusakan yang dialami dengan gejala-gejala yang dimasukkan adalah *hard disk bad sector*, maka dilanjutkan dengan melihat dataset yang memiliki kesimpulan yang sama yaitu kasus *hard disk bad sector*. Kasus dengan kesimpulan *hard disk bad sector* dapat dilihat pada Tabel II.

Tabel II: Kasus dengan kesimpulan hard disk bad sector

No	Gejala	Kasus
1	Komputer terasa lambat/berat	<i>Hard disk bad sector</i>
	Komputer sering hang	
	Komputer blue screen saat dinyalakan	
	Beberapa file corrupt/hilang dalam hard disk	
2	Komputer terasa lambat/berat	<i>Hard disk bad sector</i>
	Hard disk tidak terdeteksi	
	Komputer sering hang	
	Beberapa file corrupt/hilang dalam hard disk Terdengar suara aneh pada hard disk	
3	Komputer terasa lambat/berat	<i>Hard disk bad sector</i>
	Hard disk tidak terdeteksi	
	Komputer sering hang	
	Komputer blue screen saat dinyalakan Terdengar suara aneh pada hard disk	
4	Komputer terasa lambat/berat	<i>Hard disk bad sector</i>
	Komputer blue screen saat dinyalakan	
	Hard disk tidak terdeteksi	
	Beberapa file corrupt/hilang dalam hard disk Terdengar suara aneh pada hard disk	
5	Komputer sering hang	<i>Hard disk bad sector</i>
	Komputer blue screen saat dinyalakan	
	Beberapa file corrupt/hilang dalam hard disk	
	Terdengar suara aneh pada hard disk	

a. Perhitungan kasus ke-1

Tabel III: Perhitungan Kasus Ke-1

Kasus Lama	Kasus Baru	Similarity
Komputer terasa lambat/berat	Komputer terasa lambat/berat	1
Komputer sering hang	Komputer sering hang	1
Komputer blue screen saat dinyalakan	-	0
Beberapa file corrupt/hilang dalam hard disk	-	0
-	<i>Hard disk tidak terdeteksi</i>	0
-	Terdengar suara aneh pada hard disk	0

$$Similarity(X, 1) = \frac{2}{2 + 4} = \frac{2}{6} = 0,5$$

b. Perhitungan kasus ke-2

Tabel IV: Perhitungan Kasus Ke-2

Kasus Lama	Kasus Baru	Similarity
Komputer terasa lambat/berat	Komputer terasa lambat/berat	1
<i>Hard disk tidak terdeteksi</i>	<i>Hard disk tidak terdeteksi</i>	1
Komputer sering hang	Komputer sering hang	1
Beberapa file corrupt/hilang dalam hard disk	-	0
Terdengar suara aneh pada hard disk	Terdengar suara aneh pada hard disk	1

$$Similarity(X, 2) = \frac{4}{4 + 1} = \frac{4}{5} = 0,8$$

c. Perhitungan kasus ke-3

Tabel V: Perhitungan Kasus Ke-3

Kasus Lama	Kasus Baru	Similarity
Komputer terasa lambat/berat	Komputer terasa lambat/berat	1
<i>Hard disk tidak terdeteksi</i>	<i>Hard disk tidak terdeteksi</i>	1
Komputer sering hang	Komputer sering hang	1
Komputer blue screen saat dinyalakan	-	0
Terdengar suara aneh pada hard disk	Terdengar suara aneh pada hard disk	1

$$Similarity(X, 3) = \frac{4}{4 + 1} = \frac{4}{5} = 0,8$$

d. Perhitungan kasus ke-4

Tabel VI: Perhitungan Kasus Ke-4

Kasus Lama	Kasus Baru	Similarity
Komputer terasa lambat/berat	Komputer terasa lambat/berat	1
Komputer blue screen saat dinyalakan	-	0
Hard disk tidak terdeteksi	Hard disk tidak terdeteksi	1
Beberapa file corrupt/hilang dalam hard disk	-	0
Terdengar suara aneh pada hard disk	Terdengar suara aneh pada hard disk	1
-	Komputer sering hang	0

$$Similarity(X, 4) = \frac{3}{3 + 3} = \frac{3}{6} = 0,5$$

e. Perhitungan kasus ke-5

Tabel VII: Perhitungan Kasus Ke-5

Kasus Lama	Kasus Baru	Similarity
Komputer sering hang	Komputer sering hang	1
Komputer blue screen saat dinyalakan	-	0
Beberapa file corrupt/hilang dalam hard disk	-	0
Terdengar suara aneh pada hard disk	Terdengar suara aneh pada hard disk	1
-	Komputer terasa lambat/berat	0
-	Hard disk tidak terdeteksi	0

$$Similarity(X, 5) = \frac{2}{2 + 4} = \frac{2}{6} = 0,5$$

Hasil dari perhitungan similarity kasus ke-1 hingga kasus ke-5 dapat dilihat pada Tabel VIII.

Tabel VIII: Hasil Perhitungan Similarity Setiap Kasus Hard Disk Bad Sector

Kasus	Gejala	Diagnosa	Similarity
Kasus 2	Komputer terasa lambat/berat	<i>Hard disk bad sector</i>	80%
	Hard disk tidak terdeteksi		
	Komputer sering hang		
	Beberapa file corrupt/hilang dalam hard disk		
Kasus 3	Terdengar suara aneh pada hard disk	<i>Hard disk bad sector</i>	80%
	Komputer terasa lambat/berat		
	Hard disk tidak terdeteksi		
	Komputer sering hang		
Kasus 4	Komputer blue screen saat dinyalakan	<i>Hard disk bad sector</i>	50%
	Hard disk tidak terdeteksi		
	Beberapa file corrupt/hilang dalam hard disk		
	Terdengar suara aneh pada hard disk		
Kasus 5	Komputer terasa lambat/berat	<i>Hard disk bad sector</i>	33%
	Komputer blue screen saat dinyalakan		
	Hard disk tidak terdeteksi		
	Beberapa file corrupt/hilang dalam hard disk		
Kasus 1	Terdengar suara aneh pada hard disk	<i>Hard disk bad sector</i>	33%
	Komputer terasa lambat/berat		
	Komputer sering hang		
	Komputer blue screen saat dinyalakan		
	Beberapa file corrupt/hilang dalam hard disk		

Pada Tabel VIII dapat dilihat bahwa kasus ke-2 dan ke-3 yang mempunyai nilai kemiripan tertinggi dengan kasus baru yang dimasukkan. Maka salah satu dari kasus ke-2 dan ke-3 ini yang akan dianjurkan oleh sistem PBK sebagai solusi.

a. Perancangan Proses Reuse

Pada tahap ini diambil kembali kasus-kasus yang tersimpan dalam basis kasus yang akan digunakan sebagai solusi. Kriteria untuk pemilihan kasus adalah kasus yang memiliki hasil perhitungan paling tinggi yang telah dilakukan pada tahap *indexing*. Berdasarkan perhitungan pada tahap sebelumnya, didapatkan satu kasus lama yang memiliki tingkat kemiripan paling tinggi dengan kasus baru dibandingkan dengan kasus yang lainnya yaitu kasus hard disk bad sector dengan *similarity* sebesar 80

b. Perancangan Proses Revise

Proses *revise* dilakukan oleh pakar yang akan melakukan revisi dengan pertimbangan dari pakar sendiri terhadap solusi yang diberikan oleh sistem dan akan dilakukan perbaikan solusi melalui proses revisi pakar.

c. Perancangan Proses Retain

Proses *retain* merupakan proses untuk menyimpan kasus baru yang telah berhasil mendapatkan solusi ke dalam basis kasus, agar dapat digunakan oleh kasus-kasus selanjutnya yang mirip dengan kasus tersebut. Pada proses *retain* solusi yang didapatkan akan disimpan dalam database.

B. Pengujian K-Fold Cross Validation

Pengujian sistem *k-fold cross validation* terdapat 2 hasil pengujian yaitu menggunakan *indexing* dan *non-indexing*. Setiap proses pengujian *k-fold* akan divalidasi dengan membagi data menjadi 2 kelompok yaitu data latih dan data uji. Data akan divalidasi menggunakan *k-fold* dimana data dibagi sebanyak 10-fold sehingga memiliki 10 dataset dimana setiap dataset memiliki 20 data. Fold 1 dataset 1 akan menjadi data uji dan dataset lainnya akan menjadi data latih. Setiap fold akan mendapatkan nilai akurasi dan rata-rata waktu pengujian sebanyak n kali. Nilai akurasi didapat dari jumlah nilai yang benar dibagi total data uji, sedangkan nilai rata-rata *similarity* didapat dari rata-rata nilai *similarity* tiap data dalam data uji. Hasil pengujian 10-fold dengan *indexing* dan *non-indexing* sebanyak 10 kali dapat dilihat pada Tabel IX.

Tabel IX: K-fold cross validation

Fold	Indexing			Non-Indexing		
	Similarity	Akurasi	(Detik)	Similarity	Akurasi	(Detik)
1	76%	90%	45	66%	75%	0,05
2	79%	100%	42	76%	90%	51
3	72%	90%	36	77%	100%	0,05
4	62%	75%	44	71%	85%	51
5	76%	100%	44	78%	90%	0,05
6	65%	85%	36	74%	75%	51
7	61%	65%	38	76%	80%	51
8	76%	90%	0,04	72%	90%	51
9	78%	90%	38	70%	80%	0,05
10	57%	75%	0,04	81%	90%	0,05
Rata-rata	70%	86%	0,0403	74%	85,5%	0,0505

Berdasarkan hasil pada Tabel IX pengujian *k-fold cross validation indexing* dan *non-indexing*, memiliki perbedaan dimana hasil akurasi dan waktu yang lebih baik yaitu menggunakan *indexing*, sedangkan hasil *similarity* lebih baik pada pengujian *non-indexing*. *Indexing* lebih baik akurasi dan waktu karena proses *indexing* hanya mencari hasil berdasarkan kelas yang telah di *index*, sedangkan *similarity* pada *indexing* lebih rendah dari *non-indexing* karena *similarity* pada *indexing* hanya fokus pada kelas yang memiliki peluang yang lebih besar tanpa melihat kelas lain. *Non-indexing* lebih lama karena semua data akan di cari kemiripannya antara data kasus lama dan kasus baru. *Similarity* pada *non-indexing* lebih baik karena mencari semua nilai dan mengambil nilai k terdekat. Hal ini menunjukkan bahwa pengujian menggunakan *indexing* lebih akurat dan lebih cepat dibandingkan pengujian *non-indexing* karena setiap pengujian *indexing* tidak memerlukan waktu yang banyak untuk proses pencarian kemiripan.

IV. SIMPULAN

Pengujian sistem menggunakan *k-fold cross validation*. Pengujian sistem k-fold dengan *indexing* menggunakan metode naïve bayes dan k-nearest neighbor menghasilkan nilai rata-rata akurasi 86%, rata-rata *similarity* 70% dengan rata-rata waktu pengujian selama 0,0403 detik. Pengujian *k-fold* dengan *non-indexing* menggunakan metode k-nearest neighbor menghasilkan nilai rata-rata akurasi 85,5%, rata-rata *similarity* 74% dengan rata-rata waktu pengujian selama 0,0505 detik. Berdasarkan hasil pengujian sistem dari *indexing* dan *non-indexing* sebanyak 10 kali uji dapat disimpulkan bahwa nilai akurasi dan waktu pada pengujian *indexing* lebih baik dari pengujian *non-indexing*, dikarenakan kasus yang diuji terlebih dahulu dihitung peluang kelasnya sehingga perhitungan kemiripannya hanya melihat pada kelas yang memiliki peluang lebih besar. Nilai *similarity* pada *non-indexing* lebih baik dari pengujian *indexing* dengan selisih 4% karena, kasus yang diuji menghitung nilai kemiripan berdasarkan tetangga terdekat tanpa melihat kelas.

PUSTAKA

[1] K. Schwab, "The Fourth Industrial Revolution. New York: Crown Business; 2017," dalam Originally published by World Economic Forum, Geneva, Switzerland, 2016.

[2] A. Aamodt dan E. Plaza, "Case-based reasoning: Foundational issues, methodological variations, and system approaches," AI communications, vol. 7, no. 1, hlm. 39–59, 1994.

[3] A. Triawan, "Penerapan Metode Naive Bayes Untuk Rekomendasi Topik Tugas Akhir Berdasarkan Daftar Hasil Studi Mahasiswa di Perguruan Tinggi," Teknois: Jurnal Ilmiah Teknologi Informasi dan Sains, vol. 10, no. 2, hlm. 58–70, 2020.

[4] A. Labellapansa, A. D. Fitriani, dan A. Yulianti, "PENALARAN BERBASIS KASUS UNTUK MENENTUKAN PERAWATAN KULIT WAJAH," Seminar Nasional Teknologi Informasi dan Komunikasi (SEMNASITIK), vol. 1, no. 1, Art. no. 1, Okt 2018.

[5] W. Sinlae, S. A. S. Mola, dan N. D. Rumlaklak, "PENENTUAN KESESUAIAN LAHAN PERTANIAN TANAMAN CABAI MENGGUNAKAN METODE NAÏVE BAYES DI KABUPATEN KUPANG," jicon, vol. 9, no. 1, hlm. 56–64, Mar 2021, doi: 10.35508/jicon.v9i1.3848.

- [6] J. S. AngDjadi, S. A. S. Mola, dan D. Prasetyo, “Penerapan Case Based Reasoning Untuk Mendiagnosa Penyakit Pada Ternak Babi Menggunakan Metode Naive Bayes,” *JURNAL TEKNIK INFORMATIKA INOVATIF WIRA WACANA*, vol. 1, no. 2, Art. no. 2, Des 2022, doi: 10.58300/inovatif-wira-wacana.v1i2.285.
- [7] I. W. Supriana dan K. D. Prebiana, “METODE PENALARAN BERBASIS KASUS (CASE BASE REASONING) DALAM PENENTUAN KELAYAKAN SEKOLAH PERAWATAN,” *jurnalresistor*, vol. 3, no. 1, hlm. 57–65, Apr 2020, doi: 10.31598/jurnalresistor.v3i1.554.
- [8] B. S. Gandhi, D. A. Megawaty, dan D. Alita, “Aplikasi Monitoring Dan Penentuan Peringkat Kelas Menggunakan Naïve Bayes Classifier,” *Jurnal Informatika Dan Rekayasa Perangkat Lunak*, vol. 2, no. 1, hlm. 54–63, 2021.
- [9] A. S. Bararah dan D. Andreswari, “IMPLEMENTASI CASE BASED REASONING UNTUK DIAGNOSA PENYAKIT BERDASARKAN GEJALA KLINIS DAN HASIL PEMERIKSAAN HEMATOLOGI DENGAN PROBABILITAS BAYES,” vol. 5, no. 1, hlm. 12, 2017.
- [10] J. Han, M. Kamber, dan J. Pei, “Data Mining: Concepts and Techniques (tretja izd.). Waltham, ZDA.” Morgan Kaufmann Publishers, Elsevier, 2012.