

IMPLEMENTASI ALGORITMA *K-NEAREST NEIGHBOARS* UNTUK MEREKOMENDASIKAN PERAWATAN KOMPUTER DI INDOMARET CABANG BOGOR 2

Fuji Nur Firdaus¹, Agus Riyanto², Zuhana Realita Alfy³

Program Studi Teknik Informatika, Fakultas Teknik dan Ilmu Komputer, Universitas Indraprasta PGRI
Jalan Raya Tengah No 80, Kelurahan Gedong, Pasar Rebo, Jakarta Timur

¹fujinurfirdaus@gmail.com, ²Agus.Riyanto@unindra.ac.id, ³zuhanarealita28@gmail.com

ABSTRAK

Industri ritel modern, seperti Indomaret, sangat bergantung pada kecepatan dan efisiensi operasional yang didukung oleh sistem *Point of Sale* (POS) dan manajemen inventaris berbasis komputer. Di Indomaret Cabang Bogor 2, ribuan unit komputer aktif digunakan setiap hari untuk memproses transaksi dan mengelola persediaan. Namun, pemanfaatan data pemantauan perangkat—seperti suhu CPU, kecepatan jaringan, waktu booting, dan penggunaan CPU—belum optimal. Pemeliharaan masih bersifat reaktif, sehingga teknisi sering melakukan kunjungan berulang tanpa prioritas berbasis data, yang menyebabkan downtime, gangguan penjualan, dan peningkatan biaya perbaikan. Penelitian ini bertujuan untuk mengembangkan sistem otomatis untuk rekomendasi perawatan komputer dengan menerapkan algoritma *K-Nearest Neighbors*. Data historis kinerja perangkat dipra-proses dan dilabeli sesuai kebutuhan perawatan, kemudian digunakan sebagai basis klasifikasi kondisi komputer baru. Sistem ini dibangun dalam bentuk aplikasi web menggunakan framework Flask, yang memungkinkan teknisi mengakses rekomendasi melalui browser. Hasil pengujian menunjukkan bahwa model KNN memiliki performa tinggi dengan akurasi 97%, precision 100%, recall 67%, dan f1-score 80%. Sistem ini menyajikan daftar prioritas toko yang membutuhkan kunjungan berdasarkan kondisi perangkat, sehingga membantu teknisi melakukan perawatan secara lebih terencana dan efisien. Dengan demikian, penelitian ini membuktikan bahwa algoritma KNN efektif dalam mendukung pemeliharaan preventif perangkat komputer di lingkungan ritel modern.

Kata Kunci : *K-Nearest Neighbors*, Rekomendasi, Perawatan Komputer

ABSTRACT

Modern retail industries, such as Indomaret, rely on operational speed and efficiency supported by computer-based *Point of Sale* (POS) systems and inventory management. At Indomaret-Branch Bogor 2, thousands of computers are actively used daily to process transactions and manage inventory. However, the utilization of device monitoring data—such as CPU temperature, network speed, boot time, and CPU usage is not optimal yet. Maintenance is still reactive, resulting in technicians often visits without data-driven prioritization, which leads to downtime, sales disruption, and increased repair costs. This study aims at developing an automated system for computer maintenance recommendations by implementing the *K-Nearest Neighbors* algorithm. Historical device performance data is preprocessed and labeled according to maintenance needs, then used as the basis for classifying new computer conditions. The system is built as a web application using the Flask framework, allowing technicians to access recommendations through a browser. Test results show that the KNN model has high performance with 97% accuracy, 100% precision, 67% recall, and an f1-score of 80%. This system presents a prioritized list of stores requiring visits based on device condition, helping technicians perform maintenance more planned and efficiently. Therefore, this study demonstrates the effectiveness of the KNN algorithm in supporting preventative maintenance of computer devices in modern retail environments.

Key Word: *K-Nearest Neighbors*, Recommending, Computer Maintenance

PENDAHULUAN

Dalam industri ritel modern, khususnya di minimarket seperti Indomaret, kecepatan dan efisiensi layanan merupakan faktor krusial bagi kepuasan pelanggan. Sistem *Point of Sale* (POS) dan manajemen inventaris berbasis komputer menjadi komponen utama dalam kegiatan operasional sehari-hari. Kinerja komputer di toko secara signifikan mempengaruhi keberhasilan operasional.

Di Indomaret Cabang Bogor 2, terdapat sekitar 1.914 komputer di 869 toko aktif yang digunakan untuk memproses transaksi dan mengelola persediaan. Meskipun terdapat sistem pemantauan yang mencatat suhu perangkat, kecepatan jaringan, penggunaan CPU, dan waktu *booting*, data ini belum dimanfaatkan secara optimal. Pemeliharaan dilakukan hanya setelah terjadi kerusakan, sehingga teknisi sering kali harus

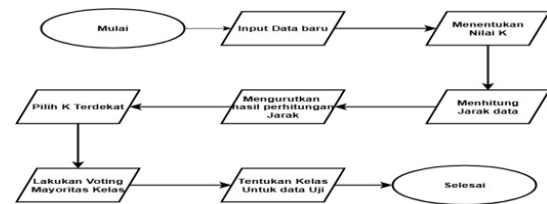
mengunjungi toko yang sama berulang kali tanpa prioritas berbasis data.

Situasi ini menyebabkan gangguan operasional, gangguan penjualan, peningkatan biaya perbaikan darurat, dan kurangnya efektivitas sistem pemantauan. Suhu CPU yang tinggi terdeteksi namun tanpa adanya rekomendasi otomatis, yang berisiko menyebabkan kerusakan serius. Diperlukan sistem yang dapat memberikan rekomendasi otomatis untuk prioritas pemeliharaan perangkat. Penerapan algoritma *K-Nearest Neighbors* untuk mendeteksi anomali yang mengindikasikan kerusakan dan merekomendasikan perangkat yang perlu ditangani, sehingga teknisi dapat melakukan kunjungan yang lebih terencana dan efisien. "Rekomendasi merupakan saran berbasis data historis untuk mendukung keputusan otomatis" (Ivania & Harani, 2024). "Perawatan adalah kegiatan rutin dan terencana untuk menjaga perangkat tetap optimal"(Jono, 2015), seperti pembersihan dan pengecekan (Andria, 2023).

Solusinya adalah algoritma *K-Nearest Neighbors* (K-NN), yang "membandingkan data baru dengan data historis berlabel"(Putry & Nurina Sari, 2022) dan "mengklasifikasikan berdasarkan jarak terdekat"(Dwi Fasnuari et al., 2022). Sistem dibangun dengan basis *website* agar mudah digunakan teknisi. "*Website* menyajikan informasi dalam berbagai format" (Elgamar, 2020). Aplikasi dikembangkan menggunakan *Flask (Python)*, *Hypertext markup Language (HTML)*, *CSS*, dan database *MySQL*. "*Database* adalah data terstruktur yang bisa diakses oleh program" (Pradana & Hardi, 2021). "HTML digunakan untuk tampilan web" (Hidayat et al., 2019). *Python* dipilih karena "fleksibel dan mudah dipelajari" (Syafi'i et al., 2025), dan "*Flask* digunakan sebagai *framework* web berbasis *Python*" (Novindri et al., 2022).

METODE PENELITIAN

Penulis melaksanakan studi di PT. Indomareo Prismatama Cabang Bogor 2. Ini adalah salah satu cabang Indomaret yang terletak di Jl. Kp. Nangewer No.227, Nangewer Mekar, Kec. Cibinong, Kabupaten Bogor, Jawa Barat 16912.



Gambar 1. Flowchart *k-nearest neighbors*

1. Dataset Awal
 Penulis mengumpulkan data historis kinerja komputer dari sistem local Indomaret. Data mencakup parameter dan label kondisi.
2. Membagi dataset
 Pembagian dilakukan dengan rasio 70:30, di mana 70 % sebagai data *training* dan 30 % sebagai data *testing*.
3. Menentukan nilai K
 Menentukan nilai K adalah memilih jumlah tetangga terdekat yang akan dipertimbangkan. Nilai K biasanya bilangan ganjil untuk menghindari hasil seri dalam klasifikasi biner (Safitri et al., 2024, h. 2).
4. Menghitung Jarak *Euclidean*
 "Semakin kecil jarak *Euclidean*, semakin mirip kedua data tersebut".(Yudhana & Agus Jaka Sri Hartanta, 2020). Jarak *Euclidean* dapat dihitung menggunakan persamaan (1) berikut. (Yudhana & Agus Jaka Sri Hartanta, 2020, h. 3).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i \text{ training} - y_i \text{ testing})^2} \quad (1)$$

$d(x, y)$	= Jarak
$x_i \text{ training}$	= Data <i>Training</i>
$y_i \text{ testing}$	= Data <i>Testing</i>
n	= dimensi data
i	= <i>Variable Data</i>
5. Mengurutkan hasil perhitungan Jarak
 Setelah semua jarak dihitung, data latih diurutkan dari yang paling mirip (jarak terkecil) .
6. Pilih K terdekat
 Pilih K titik data *training* yang memiliki jarak terdekat dengan data *testing*.
7. Tentukan Kelas Rekomendasi untuk Data *Testing*
 Sebagai contoh, jika K=5 dan 3 tetangga berlabel "Ya" sementara 2 lainnya berlabel "Tidak", maka data uji tersebut akan diklasifikasikan sebagai "Ya".

8. Evaluasi

Penulis menggunakan metode *confusion matrix* sebagai metode evaluasi performa model, dengan mengukur metrik akurasi, *precision*, *recall*, dan *f1-score* berdasarkan nilai *True Positive (TP)* yang merupakan "jumlah data yang benar-benar ya, dan diprediksi sebagai ya oleh model" (Safitri et al., 2024), *True Negative (TN)* atau "jumlah data yang sebenarnya ya, tapi diprediksi sebagai tidak oleh model" (Safitri et al., 2024), *False Positive (FP)* didefinisikan sebagai "jumlah data yang sebenarnya tidak, tapi diprediksi ya oleh model" (Safitri et al., 2024) dan *False Negative (FN)* sebagai "jumlah data yang benar-benar belum perlu, dan diprediksi sesuai oleh model" (Safitri et al., 2024).

HASIL DAN PEMBAHASAN

Dataset Awal

Dalam penelitian ini, penulis menggunakan 100 toko sampel dari Indomaret.

Tabel 1. Dataset Awal

Kode	Usage	LAN	Suhu	Boot
TXPJ	96	1000	44	0.87
TDVH	96	100	46	0.86
TJTX	94	1000	49	0.98
FGQ2	94	1000	62	0.95
TWAT	93	1000	54	1.78
TCOJ	93	1000	53	1.05
TKLQ	93	1000	53	0.8
TYQE	91	1000	60	1.58
TB1K	91	1000	56	0.89
F19G	91	1000	64	0.88
TGWI	90	100	100	0.66
THLY	9	1000	49	0.47
TJ49	9	1000	51	0.42

Pemberian kelas (*Preprocessing*)

Setelah data dimasukkan, sistem akan mengelompokkan komputer berdasarkan aturan tabel 2.

Tabel 2. Ketentuan pemberian kelas

Kelas	Keterangan
Ya	Memerlukan Perawatan
Tidak	Belum Memerlukan Perawatan

Pemberian label didasarkan pada total bobot atau skor tersebut. Tabel 3 menyajikan ketentuan pemberian label.

Tabel 3. Bobot masing masing fitur

Fitur	Kondisi	Bobot	Keterangan
Suhu CPU	> 80	20	Diberi bobot 20, Dapat menurunkan performa dan merusak hardware jika dibiarkan terus-menerus.
Bootin g Time	> 240 detik	30	Diberi bobot 30, Mempengaruhi kesiapan komputer saat toko dibuka atau restart.
LAN Speed	< 1000 Mbps	40	Diberi bobot 40, penting untuk transaksi dan komunikasi data antar sistem toko.
CPU Usage	> 80	10	Diberi bobot 10 karena data bisa berubah dengan cepat.

Tabel 4. Pemberian label

Total Skor	Kelas Rekomendasi perawatan
≥ 30	Ya
< 30	Tidak

Total skor ≥ 30 diberi label "Ya" karena menunjukkan dua atau lebih masalah signifikan. Skor < 30 dianggap "Belum Diperlukan Perawatan" atau "tidak", karena kondisi normal. Disajikan 10 data dari 100 dataset yang telah melewati *preprocessing* untuk pemberian label.

Tabel 5. Dataset yang sudah di beri label

Kdtk	Dataset				Pembobotan					Label
	Usage (%)	Lan (Mbps)	Suhu (°C)	Boot (detik)	Usage	Lan	Suhu	Boot	Total	
TXPJ	96	1000	44	52.2	10	0	0	0	10	tidak
TDVH	96	100	46	51.6	10	40	0	0	50	ya
TJTX	94	1000	49	58.8	10	0	0	0	10	tidak
FGQ2	94	1000	62	57	10	0	0	0	10	tidak
TWAT	93	1000	54	106.8	10	0	0	0	10	tidak
TCOJ	93	1000	53	63	10	0	0	0	10	tidak
TKLQ	93	1000	53	48	10	0	0	0	10	tidak
TYQE	91	1000	60	94.8	10	0	0	0	10	tidak
TB1K	91	1000	56	53.4	10	0	0	0	10	tidak
F19G	91	1000	64	52.8	10	0	0	0	10	tidak

Pada Tabel 5, kita melihat komputer toko TXPJ. Penggunaan CPU mencapai 96 %, lebih dari 80 %, skornya bertambah 10. Kecepatan LAN 1000 Mbps dan suhu tidak lebih 80°, skornya bertambah nol. Booting time 52,2 detik, tidak lebih dari 240 detik, skornya bertambah nol. Total skornya 10, kurang dari sama dengan 30, sehingga label rekomendasi perawatan adalah "Tidak". Berikut hasil data yang telah diberikan kelas pada label.

Perhitungan *K-Nearest Neighbors*

1. Pembagian dataset *training* dan *testing*
 Dalam penelitian ini, dari 100 data

komputer, 70 data digunakan untuk melatih model KNN, dan 30 data digunakan untuk pengujian.

	CPU_USAGE	LAN_SPEED	SUHU	BOOT_TIME	label
11	9.0	1000.0	49.0	28.2	Tidak
47	9.0	1000.0	43.0	32.4	Tidak
85	8.0	1000.0	54.0	136.2	Tidak
26	9.0	1000.0	57.0	30.6	Tidak
93	8.0	1000.0	50.0	16.8	Tidak
...
60	9.0	1000.0	55.0	37.2	Tidak
71	85.0	1000.0	59.0	111.0	Tidak
14	9.0	100.0	52.0	21.6	ya
92	8.0	1000.0	46.0	25.8	Tidak
51	9.0	1000.0	63.0	46.8	Tidak

Gambar 2. Data training

83	80.0	1000.0	61.0	39.0	Tidak
53	9.0	1000.0	33.0	48.0	Tidak
70	85.0	1000.0	32.0	48.0	Tidak
45	9.0	1000.0	49.0	33.6	Tidak
44	9.0	1000.0	50.0	24.0	Tidak
39	9.0	1000.0	69.0	17.4	Tidak
22	9.0	1000.0	43.0	37.8	Tidak
80	82.0	1000.0	69.0	40.8	Tidak
10	90.0	100.0	100.0	39.6	ya
9	96.0	1000.0	44.0	52.2	Tidak
18	9.0	1000.0	58.0	27.0	Tidak
30	9.0	1000.0	37.0	47.4	Tidak
73	84.0	100.0	39.0	57.6	ya
33	9.0	1000.0	67.0	19.2	Tidak
90	8.0	1000.0	47.0	54.0	Tidak
4	93.0	1000.0	54.0	106.8	Tidak
76	83.0	1000.0	59.0	46.2	Tidak
77	82.0	1000.0	42.0	46.8	Tidak
12	9.0	1000.0	51.0	25.2	Tidak
31	9.0	1000.0	33.0	159.0	Tidak
55	9.0	1000.0	0.0	336.6	ya
88	8.0	1000.0	47.0	30.6	Tidak
25	9.0	1000.0	49.0	39.6	Tidak
42	9.0	1000.0	66.0	27.6	Tidak
69	87.0	1000.0	41.0	49.8	Tidak
15	9.0	1000.0	55.0	26.4	Tidak
40	9.0	1000.0	52.0	98.4	Tidak
96	8.0	1000.0	56.0	0.0	Tidak
9	91.0	1000.0	64.0	52.8	Tidak
72	85.0	1000.0	41.0	79.2	Tidak

Gambar 3. Data testing

2. Perhitungan euclidean distance

Hasil perhitungan euclidean distance menggunakan persamaan (1) sehingga menghasilkan tabel berikut.

Tabel 6 hasil perhitungan jarak euclidean

Jarak TXPJ ke-	USAGE (%)	LAN (Mbps)	SUHU (°C)	BOOT (Detik)	Jarak
TDVH	96.0	100.0	46.0	51.6	900.0
TJTX	94.0	1000.0	49.0	58.8	8.52
FGQ2	94.0	1000.0	62.0	57.0	18.73
TCOJ	93.0	1000.0	53.0	48.0	10.37
TKLQ	93.0	1000.0	53.0	48.0	10.37
TYQE	91.0	1000.0	51.0	44.0	11.88
TBIK	91.0	1000.0	56.0	28.4	27.12
THLY	9.0	1000.0	28.2	22.6	93.25
FIOI	9.0	1000.0	52.0	21.6	92.57
TS77	9.0	1000.0	41.0	40.2	87.87

3. Mengurutkan euclidean distance terkecil

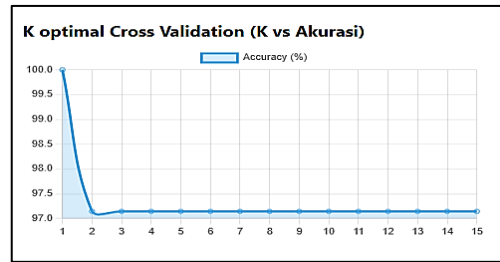
Tabel 7 hasil perhitungan jarak euclidean yang sudah diurutkan

Jarak TXPJ ke-	Usage	LAN	Suhu	Boot	Jarak
TJTX	94.0	1000	49.0	58.8	8.52
TCOJ	93.0	1000	53.0	48.0	10.37
TKLQ	93.0	1000	53.0	48.0	10.37
TYQE	91.0	1000	51.0	44.0	11.88
FGQ2	94.0	1000	62.0	57.0	18.73
TBIK	91.0	1000	56.0	28.4	27.12
TS77	9.0	1000	41.0	40.2	87.87
FIOI	9.0	100	52.0	21.6	92.57
THLY	9.0	1000	28.2	22.6	93.25
TDVH	96	100	46.0	51.6	900

4. Penentuan Nilai K

Penulis menggunakan k-fold cross validation. Dimana nilai K dengan akurasi

tertinggi akan menjadi nilai K penentu hasil rekomendasi.



Gambar 4. Hasil K-fold validation

Gambar 4 menunjukkan bahwa akurasi tertinggi adalah k = 1, dengan akurasi mencapai 100 %. Setelah menentukan nilai K.

5. Penentuan Kelas Rekomendasi

Dari tabel 6 jarak euclidean terkecil yaitu data komputer toko TJTX dengan jarak euclidean 8,52. Dari hasil itu bisa kita simpulkan bahwa label klasifikasi prediksi pada data testing TXPJ adalah sama dengan label klasifikasi TJTX yaitu "tidak". Bisa kita lihat hasil lengkap pada gambar 5.

	CPU_USAGE	LAN_SPEED	SUHU	BOOT_TIME	label	prediksi
83	80.0	1000.0	61.0	39.0	Tidak	Tidak
53	9.0	1000.0	33.0	48.0	Tidak	Tidak
70	85.0	1000.0	32.0	48.0	Tidak	Tidak
45	9.0	1000.0	49.0	33.6	Tidak	Tidak
44	9.0	1000.0	50.0	24.0	Tidak	Tidak
39	9.0	1000.0	69.0	17.4	Tidak	Tidak
22	9.0	1000.0	43.0	37.8	Tidak	Tidak
80	82.0	1000.0	69.0	40.8	Tidak	Tidak
10	90.0	100.0	100.0	39.6	ya	ya
0	96.0	1000.0	44.0	52.2	Tidak	Tidak

Gambar 5. Hasil lengkap rekomendasi

Evaluasi.

Actual \ Predicted	tidak	ya
tidak	27	0
ya	1	2

Gambar 6. Confusion matrix

TP = True Positive sebanyak 2 data yang diprediksi Ya dan benar. TN = True Negatif sebanyak 27 data yang di Tidak dan benar. FP = False Positive sebanyak 0 data yang diprediksi Ya dan salah. FN = False Negative sebanyak 1 data yang diprediksi Tidak dan salah.

1. Accuracy adalah “ rasio prediksi benar (positif dan negatif) dengan keseluruhan

data.Accuracy" (Safitri et al., 2024, h. 4).

$$Acc = \frac{TP+TN}{TP + FP + FN + TN} \times 100\% \quad (2)$$

2. "Precision adalah perbandingan antara jumlah prediksi benar positif dengan semua hasil yang diprediksi positif." (Safitri et al., 2024, h. 4).

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (3)$$

3. Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data negatif, Rumus *Recal* (Safitri et al., 2024, h. 2).

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (4)$$

4. F1- Score, "merupakan rasio dari hasil *precision* dan *recall*, dihitung dengan persamaan berikut" (Safitri et al., 2024, h. 4).

$$F1- = 2 * \frac{precision * recall}{precision + recall} \times 100\% \quad (5)$$

Dari perhitungan evaluasi, dengan persamaan (2), (3), (4), dan (5), menghasilkan akurasi sebesar 97% dan tabel 8 berikut.

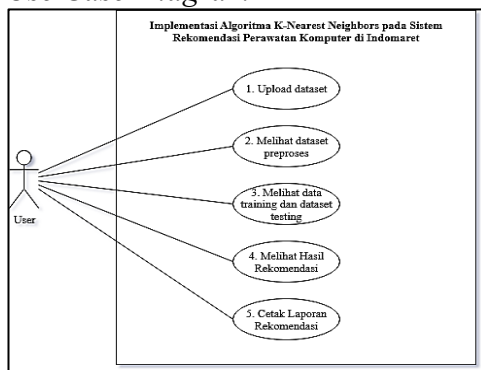
Classification Report:					
	precision	recall	f1-score	support	
Ya	0.96	1.00	0.98	27	
Tidak	1.00	0.67	0.80	3	
accuracy			0.97	30	
macro avg	0.98	0.83	0.89	30	
weighted avg	0.97	0.97	0.96	30	

Gambar 7. Hasil Klasifikasi

Model *K-Nearest Neighbors* menunjukkan hasil sangat baik dengan recall 100% pada kelas "Ya" dan precision 96%. Untuk kelas "Tidak", *precision* mencapai 100%. Model K-NN ini efektif untuk deteksi awal masalah komputer.

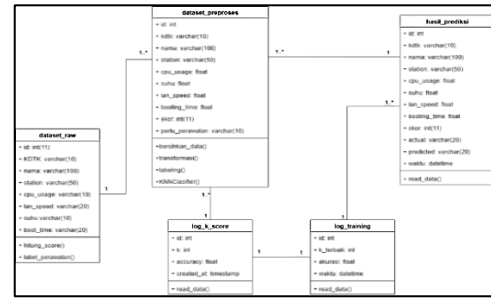
UML (Unified Modeling Language)

1. Use Case Diagram



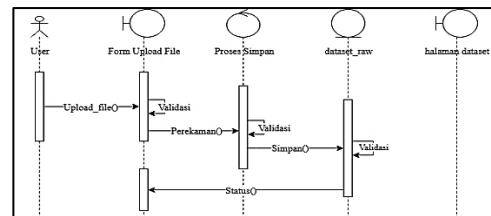
Gambar 8. Use case diagram

2. Class diagram, menunjukkan gambar tentang struktur kelas dalam sebuah sistem.

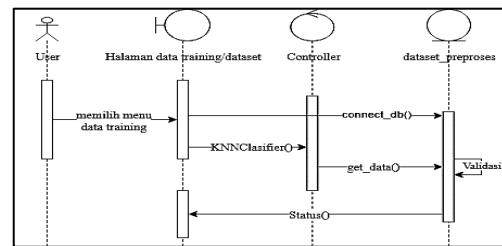


Gambar 9. Class diagram

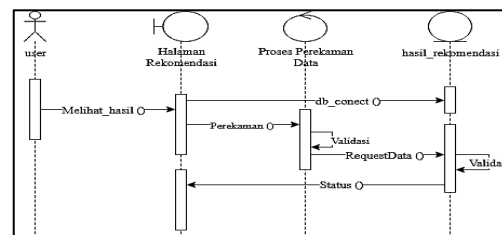
3. Sequence diagram, "sequence diagram merupakan gambaran proses pelaksanaan suatu operasi, termasuk pesan yang dikirimkan serta waktu pengirimannya" (Sano, 2020).



Gambar 10. Sequence diagram upload dataset

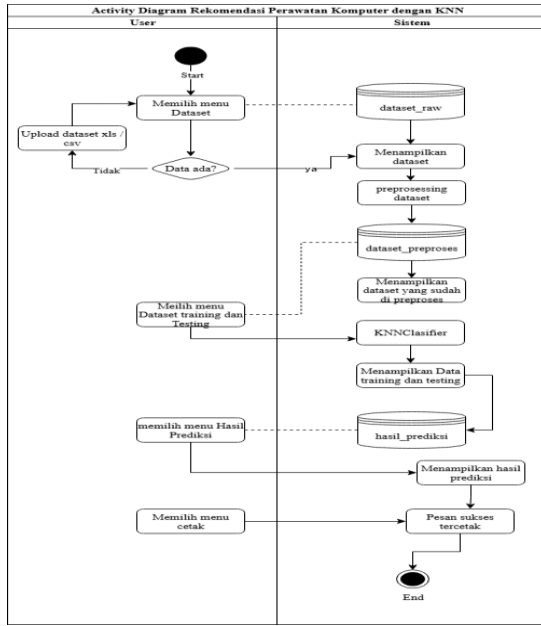


Gambar 11. Sequence diagram training dan testing



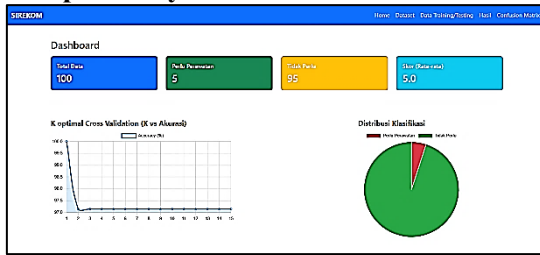
Gambar 12. Sequence diagram hasil rekomendasi

4. Activity diagram, "Diagram aktivitas didefinisikan sebagai pemodelan yang dilakukan pada suatu sistem dan menggambarkan aktivitas sistem yang sedang berjalan." (Puturu, 2022).



Gambar 13. Activity diagram

Tampilan Layar



Gambar 14. Tampilan layar dashboard

ID	KODE	NAMA	STATION	CPU USAGE	LAN SPEED	RAM	BOOT TIME
1	T01	SACS JAYA	1	16	100 MB/s	44	0:57 Menit
2	T02	TRUSMI P. RAHAYU K	1	16	100 MB/s	46	0:58 Menit
3	T03	HAJAR GUNUNG	1	14	100 MB/s	49	0:58 Menit
4	T04	SERING RAYA	1	34	100 MB/s	52	0:59 Menit
5	T05	CTSA MIDAN 2	1	15	100 MB/s	54	0:59 Menit
6	T06	SAWIR DEWAS 2	1	16	100 MB/s	55	0:59 Menit
7	T07	NAGATI	1	11	100 MB/s	53	0:58 Menit
8	T08	KONDOL SAMPUR	2	11	100 MB/s	58	1:00 Menit
9	T09	LEUWANGUNGGUNG 2	1	11	100 MB/s	56	0:58 Menit
10	T10	TEUREBEK	1	15	100 MB/s	54	0:58 Menit

Gambar 15. Tampilan layar dataset awal

Gambar 15 menampilkan halaman dataset awal, serta form untuk upload dataset baru.

ID	KODE	NAMA	STATION	CPU USAGE	LAN SPEED	RAM	BOOT TIME	PERLU PERAWATAN
400	T01	SACS JAYA	1	16	100	44	0:57	Ya
401	T02	TRUSMI P. RAHAYU K	1	16	100	46	0:58	Ya
402	T03	HAJAR GUNUNG	1	14	100	49	0:58	Ya
403	T04	SERING RAYA	1	34	100	52	0:59	Ya
404	T05	CTSA MIDAN 2	1	15	100	54	0:59	Ya
405	T06	SAWIR DEWAS 2	1	16	100	55	0:59	Ya
406	T07	NAGATI	1	11	100	53	0:58	Ya
407	T08	KONDOL SAMPUR	2	11	100	58	1:00	Ya
408	T09	LEUWANGUNGGUNG 2	1	11	100	56	0:58	Ya
409	T10	TEUREBEK	1	15	100	54	0:58	Ya

Gambar 16. Tampilan layar data training

Gambar 16 dataset training hasil pembagian, juga ditampilkan k- optimal yang dipakai.

ID	KODE	NAMA	STATION	CPU USAGE	LAN SPEED	RAM	BOOT TIME	PERLU PERAWATAN
400	T01	SACS JAYA	1	16	100	44	0:57	Ya
401	T02	TRUSMI P. RAHAYU K	1	16	100	46	0:58	Ya
402	T03	HAJAR GUNUNG	1	14	100	49	0:58	Ya
403	T04	SERING RAYA	1	34	100	52	0:59	Ya
404	T05	CTSA MIDAN 2	1	15	100	54	0:59	Ya
405	T06	SAWIR DEWAS 2	1	16	100	55	0:59	Ya
406	T07	NAGATI	1	11	100	53	0:58	Ya
407	T08	KONDOL SAMPUR	2	11	100	58	1:00	Ya
408	T09	LEUWANGUNGGUNG 2	1	11	100	56	0:58	Ya
409	T10	TEUREBEK	1	15	100	54	0:58	Ya

Gambar 17. Tampilan layar dataset testing

Gambar 17 dataset testing hasil pembagian dari dataset awal.

	precision	recall	f1 score	support
accuracy	0.97	0.97	0.97	0.97
macro avg	0.98	0.93	0.93	30.03
micro avg	0.96	1.00	0.98	27.03
weighted avg	0.97	0.97	0.96	30.03
pr	1.00	0.67	0.80	1.00

Gambar 18. Tampilan layar hasil perhitungan KNN

ID	KODE	NAMA	STATION	RAM	boot_time	cpu_usage	book_time	lan	prediksi	status	waktu
4010	T04	SERING RAYA	1	34	100:00	52	390	0	Ya	Ya	2025-07-04 23:32:17
4011	T02	TRUSMI	1	16	100:00	46	450	0	Ya	Ya	2025-07-04 23:32:17
4012	T01	SACS JAYA	1	16	100:00	44	450	0	Ya	Ya	2025-07-04 23:32:17
4013	T03	HAJAR GUNUNG	1	14	100:00	49	350	0	Ya	Ya	2025-07-04 23:32:17
4014	T05	CTSA MIDAN 2	1	15	100:00	54	360	0	Ya	Ya	2025-07-04 23:32:17
4015	T06	SAWIR DEWAS 2	1	16	100:00	55	360	0	Ya	Ya	2025-07-04 23:32:17
4016	T07	NAGATI	1	11	100:00	53	360	0	Ya	Ya	2025-07-04 23:32:17
4017	T08	KONDOL SAMPUR	2	11	100:00	58	360	0	Ya	Ya	2025-07-04 23:32:17
4018	T09	LEUWANGUNGGUNG 2	1	11	100:00	56	360	0	Ya	Ya	2025-07-04 23:32:17
4019	T10	TEUREBEK	1	15	100:00	54	360	0	Ya	Ya	2025-07-04 23:32:17

Gambar 19. Hasil Prediksi

Gambar 19 hasil prediksi dimana berisi komputer komputer yang perlu di lakukan perawatan.

Actual \ Predicted	tidak	ya
Actual's Predicted	27	0
ya	1	2

Gambar 20 Confusion matrix

SIMPULAN DAN SARAN

Simpulan

Sebuah sistem untuk rekomendasi perawatan komputer di Indomaret Cabang Bogor 2 telah dibuat dalam bentuk aplikasi web. Sistem menggunakan data suhu CPU, booting time, LAN speed, dan CPU usage untuk memberikan saran perawatan. Algoritma K-NN digunakan membandingkan data dalam menentukan kondisi perangkat, dengan akurasi mencapai 97 %. Sistem ini membantu teknisi menentukan prioritas perawatan toko

melalui pemrosesan data monitoring dan klasifikasi.

Saran

Sistem perlu diuji di lebih banyak cabang untuk melihat kinerjanya. Disarankan menambah data seperti umur perangkat dan riwayat kerusakan agar rekomendasi lebih tepat. Sistem bisa ditambah fitur notifikasi otomatis lewat Telegram agar teknisi tidak perlu memeriksa secara manual.

DAFTAR PUSTAKA

- Andria, A. (2023). Penerapan Sistem Informasi Monitoring Maintenance and Repair Hardware Di UPT Komputer Universitas PGRI Madiun. *Fountain of Informatics Journal*, 7(3), 24–33. <https://doi.org/10.21111/fij.v7i3.9421>
- Dwi Fasnuari, H. A., Yuana, H., & Chulkamdi, M. T. (2022). Penerapan Algoritma K-Nearest Neighbor Untuk Klasifikasi Penyakit Diabetes Melitus. *Antivirus: Jurnal Ilmiah Teknik Informatika*, 16(2), 133–142. <https://doi.org/10.35457/antivirus.v16i2.2445>
- Elgamar. (2020). *Konsep Dasar Pemrograman website dengan PHP* (N. Pangesti, Ed.). CV. Multimedia Edukasi.
- Hidayat, A., Yani, A., Rusidi, & Saadulloh. (2019). Membangun Website SMA PGRI Gunung Raya Ranau Menggunakan PHP dan MySQL. *JTIM (JURNAL TEKNIK INFORMATIKA MAHA KARYA)*, 2(2), 41–52.
- Ivania Sidora, L., & Hanum Harani, N. (2024). Sistem Rekomendasi Musik Spotify Menggunakan Knn Dan Algoritma Genetika. *JATI (Jurnal Mahasiswa Teknik Informatika)*, 7(4), 2585–2591. <https://doi.org/10.36040/jati.v7i4.7073>
- Jono. (2015). Total Productive Maintenance (TPM) pada Perawatan Mesin Boiler Menggunakan Metode Overall Equipment Effectiveness (OEE). *TEKINFO-JURNAL ILMIAH TEKNIK INDUSTRI DAN INFORMASI*, 3. <https://doi.org/https://doi.org/10.31001/tekinfo.v3i2.117>
- Novindri, G. F., Ocsa, P., & Saian, N. (2022). Implementasi Flask pada Sistem Penentuan Minimal Order untuk tiap Item Barang di Distribution Center pada PT XYZ Berbasis Website. In *Jurnal MNEMONIC* (Vol. 5, Issue 2).
- Pradana, A., & Hardi, I. (2021). *SISTEM INFORMASI ALAT KESEHATAN BERBASIS WEB* (Vol. 1, Issue 1). <http://ejurnal.provisi.ac.id/index.php/FKB>page14
- Putry, N. M., & Nurina Sari, B. (2022). Komparasi Algoritma Knn Dan Naïve Bayes Untuk Klasifikasi Diagnosis Penyakit Diabetes Mellitus. *EVOLUSI: Jurnal Sains Dan Manajemen*, 10(1). <https://doi.org/10.31294/evolusi.v10i1.12514>
- Puturu, V. (2022). SISTEM INFORMASI MANAJEMEN PENELITIAN DAN PENGABDIAN PNPB PADA POLITEKNIK NEGERI AMBON. *JURNAL SIMETRIK*, 12(1). <https://eternalsunshineoftheismind.wordpress.com/>
- Safitri, N., Kusnandar, D., & Martha, S. (2024). Implementasi Algoritma K-Nearest Neighbors dengan Normalisasi Z-Score dalam Klasifikasi Penerima Bantuan Sosial Desa Serunai. In *Buletin Ilmiah Math. Stat. dan Terapannya (Bimaster)* (Vol. 13, Issue 1).
- Sano, A. V. D. (2020). *Diagram Sequence Dalam Analisa Desain Sistem Informasi*. <https://binus.ac.id/malang/2020/12/diagram-sequence-dalam-analisa-desain-sistem-informasi/>
- Syafi'i, N. F., Harahap, R. R., & Rizal, C. (2025). Pembuatan Aplikasi Berita Menggunakan Flask Python. *Jurnal Minfo Polgan*, 13(2), 2515–2520. <https://doi.org/10.33395/jmp.v13i2.14537>
- Yudhana, A., & Agus Jaka Sri Hartanta, dan. (2020). ALGORITMA K-NN DENGAN EUCLIDEAN DISTANCE UNTUK PREDIKSI HASIL PENGGERGAJIAN KAYU SENGON. *TRANSMISI*, 22(4). <https://doi.org/10.14710/transmisi.22.4.107-141>