

BAB V

SIMPULAN DAN SARAN

5.1 Simpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian sistem prediksi gagal jantung menggunakan metode *K-Nearest Neighbor* (KNN) dan *Naive Bayes* berbasis web, dapat diambil beberapa simpulan yang menjawab tujuan penelitian sebagai berikut:

1. Pembangunan Model Prediksi Gagal Jantung

Penelitian ini berhasil membangun model *data mining* untuk memprediksi penyakit gagal jantung dengan memanfaatkan *dataset* keberlangsungan pasien dari Institut Kardiologi Faisalabad, Rumah Sakit Allied di Faisalabad (Punjab, Pakistan). Model dikembangkan menggunakan dua algoritma, yaitu *K-Nearest Neighbor* (KNN) dan Gaussian Naive Bayes (GNB), yang telah diimplementasikan ke dalam sistem berbasis web menggunakan bahasa pemrograman *Python*, *framework Streamlit*, dan MySQL untuk penyimpanan data pengguna serta histori pelatihan.

2. Perbandingan Kinerja KNN dan *Naive Bayes*

Hasil pengujian menunjukkan bahwa algoritma KNN memiliki akurasi yang sedikit lebih tinggi yaitu 82,85% dibandingkan *Naive Bayes* sebesar 82,43%. Namun, *Naive Bayes* unggul dalam kecepatan proses prediksi. Perbandingan ini menjelaskan bahwa pemilihan metode dapat disesuaikan dengan prioritas pengguna, apakah mengutamakan akurasi prediksi atau efisiensi waktu komputasi. Dengan demikian, tujuan penelitian untuk membandingkan kinerja kedua algoritma telah tercapai dan memberikan informasi mengenai kelebihan masing-masing metode.

3. Pengembangan Aplikasi Prediksi Gagal Jantung

Penelitian ini juga berhasil mengembangkan aplikasi prediksi gagal jantung berbasis web yang memanfaatkan model yang telah dibangun. Aplikasi ini menyediakan berbagai fitur

seperti *login* dan registrasi, prediksi manual, pelatihan model untuk admin, histori penggunaan, serta logout. Hasil pengujian *Blackbox Testing* menunjukkan seluruh fitur berfungsi dengan baik, sedangkan UAT terhadap 35 responden memberikan tanggapan positif karena aplikasi dinilai mudah digunakan, informatif, dan membantu dalam memahami risiko gagal jantung. Aplikasi ini mampu menampilkan prediksi dalam bentuk persentase probabilitas dari kedua model, sehingga dapat menjadi alat bantu awal bagi tenaga medis maupun pasien dalam memperkirakan keberlangsungan hidup pasien sebelum melakukan pemeriksaan lanjutan.

5.2 Saran

Adapun beberapa saran yang dapat dipertimbangkan untuk pengembangan sistem di masa mendatang antara lain:

1. Pengembangan *Dataset*.

Disarankan untuk memperluas jumlah dan jenis data yang digunakan untuk melatih model agar sistem mampu menghasilkan prediksi yang lebih akurat dan adaptif terhadap berbagai variasi data.

2. Menjalankan kolaborasi dengan tenaga medis.

Disarankan untuk melakukan validasi sistem menggunakan data nyata dari rumah sakit atau klinik, serta menjalin kolaborasi dengan tenaga medis guna meningkatkan keandalan dan relevansi hasil prediksi terhadap kondisi pasien di dunia nyata.

3. Eksplorasi Algoritma Tambahan.

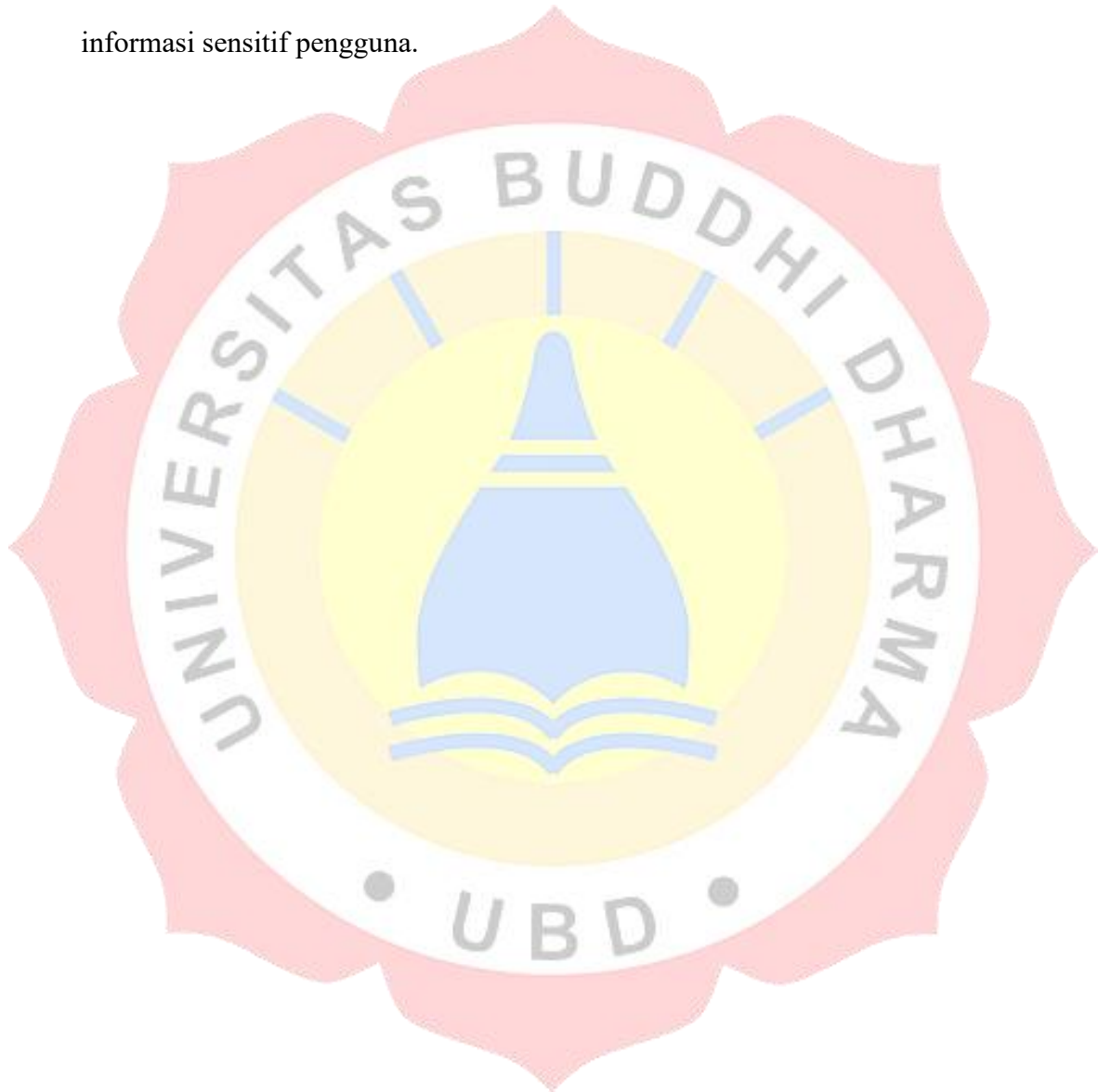
Penggunaan algoritma pembelajaran mesin lain seperti *Random Forest*, *Support Vector Machine*, atau *Decision Tree* dapat dijadikan opsi untuk membandingkan dan meningkatkan performa sistem.

4. Peningkatan Fitur *Logging* dan Pelaporan.

Perlu penambahan fitur log yang lebih rinci serta sistem pelaporan otomatis yang dapat digunakan untuk analisis penggunaan aplikasi atau pelatihan model secara berkala.

5. Mempertimbangkan penerapan standar keamanan dan privasi data medis.

Mengingat sistem ini berpotensi digunakan dalam konteks kesehatan yang melibatkan informasi sensitif pengguna.



DAFTAR PUSTAKA

- Adrianus, H., & Lasut, D. (2023). Penerapan Data Mining Untuk Memprediksi Penyakit Diabetes Menggunakan Algoritma Naive Bayes. *JURNAL ALGOR*, 4(2).
<https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- Ahmad, E., Tiwari, A., & Kumar, A. (2020). Cardiovascular Diseases (CVDs) Detection Using Machine Learning Algorithms. *International Journal for Research in Applied Science and Engineering Technology*, 8(6), 2341–2346. <https://doi.org/10.22214/ijraset.2020.6376>
- Akalın, B., Veranyurt, Ü., & Veranyurt, O. (2020). Classification of Individuals at Risk of Heart Disease Using Machine Learning. *Cumhuriyet Medical Journal*, 42(3), 283–289.
<https://doi.org/10.7197/cmj.vi.742161>
- Amin, R., & Tampubolon, E. (2024). Penerapan Algoritma K-Nearest Neighbor untuk Prediksi Akurasi Penyakit Diabetes. *Informatics And Digital Expert (INDEX)*, 6(1), 87–91. <https://e-journal.unper.ac.id/index.php/informatics>
- Amna, Gede Iwan Sudipa, I., Andi Putra, T. E., Jurnaidi Wahidin, A., Alfa Syukrilla, W., Khrisna Wardhani, A., Heryana, N., Indriyani, T., Willyanto Santoso, L., & S, W. (2023). *Data Mining* (D. Ediana & Yanto Ari, Eds.; 1st ed.). PT.GLOBALEKSEKUTIFTEKNOLOGI.
www.globaleksekutifteknologi.co.id
- Bansal, M., Goyal, A., & Choudhary, A. (2022). A Comparative Analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory Algorithms in Machine Learning. *Decision Analytics Journal*, 100071.
<https://doi.org/10.1016/j.dajour.2022.100071>
- Buani, D. C. P. (2021). Penerapan Algoritma Naïve Bayes Dengan Seleksi Fitur Algoritma Genetika Untuk Prediksi Gagal Jantung. *Jurnal Sains Dan Manajemen*, 9(2).
- Cutting, V., & Stephen, N. (2021). A Review on using Python as a Preferred Programming Language for Beginners. *International Research Journal of Engineering and Technology (IRJET)*, 08(08).
<https://www.researchgate.net/publication/359379004>
- Edric, & Tamba, S. P. (2022). Prediksi Penyakit Gagal Jantung Dengan Menggunakan Random Forest. *JUSIKOM PRIMA (Jurnal Sistem Informasi Dan Ilmu Komputer Prima)*, 5(2), 176–181.

- Effendy, E., Siregar, E. A., Fitri, P. C., & Damanik, I. A. S. (2023). Mengenal Sistem Informasi Manajemen Dakwah (Pengertian Sistem, Karakteristik Sistem). *Jurnal Pendidikan Dan Konseling*, 5, 4343–4349. <https://doi.org/10.31004/jpdk.v5i2.14061>
- Gan, W., Ye, Z., Wan, S., & Yu, P. S. (2023). Web 3.0: The Future of Internet. *Companion Proceedings of the ACM Web Conference 2023 (WWW '23 Companion)*, 1266–1275. <https://doi.org/https://doi.org/10.1145/3543873.3587583>
- Gholizadeh, S. (2022). Top Popular Python Libraries in Research. *Journal of Robotics and Automation Research*, 3(2), 142–145. <https://doi.org/10.22541/au.164580055.55493761/v1>
- Hansen, & Hariyanto, S. (2023). Perbandingan Algoritma Data Mining Dalam Mengklasifikasi Penyakit Diabetes Menggunakan Model C4.5 Dan Naive Bayes. *JURNAL ALGOR*, 4(2). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- Hardiyanto, I., Rose Irawati, A., Heningtyas, Y., & Andrian, R. (2033). *Evaluasi Performa Sistem Informasi FMIPA Universitas Lampung Menggunakan Information Systems Functional Scorecard (ISFS)* (Vol. 4, Issue 2). <https://siakadu.unila.ac.id>
- Iftikhar, H., Khan, M., Khan, Z., Khan, F., Alshanbari, H. M., & Ahmad, Z. (2023). A Comparative Analysis of Machine Learning Models: A Case Study in Predicting Chronic Kidney Disease. *Sustainability (Switzerland)*, 15(3). <https://doi.org/10.3390/su15032754>
- Joseph, V. R. (2022). Optimal ratio for data splitting. *Statistical Analysis and Data Mining*, 15(4), 531–538. <https://doi.org/10.1002/sam.11583>
- Kartika Dewi, N., & Yahfizham. (2023). Pengenalan Dasar Algoritma Pemograman Bagi Mahasiswa. *JournalOfInformaticsAndBusisnes*, 01(03), 156–161.
- Kumar, D. (2020). Cardiovascular Disease Prediction Using Machine Learning. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 6(5), 46–54. <https://doi.org/10.32628/CSEIT20659>
- Muharrom, M. (2023). Analisis Komparasi Algoritma Data Mining Naive Bayes, K-Nearest Neighbors dan Regresi Linier Dalam Prediksi Harga Emas. *BULLETIN OF INFORMATION TECHNOLOGY (BIT)*, 4(4), 430–438. <https://doi.org/10.47065/bit.v3i1>
- Nimonkar, A. R. (2024). A Machine Learning Model for Early Prediction of Multiple Diseases to Cure Lives. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT(IJSREM)*, 08(06), 1–4. <https://doi.org/10.55041/IJSREM36158>

- Pan, Z., Wang, Y., & Pan, Y. (2020). A New Locally Adaptive K-Nearest Neighbor Algorithm Based On Discrimination Class. *Knowledge-Based Systems*, 204. <https://doi.org/10.1016/j.knosys.2020.106185>
- Priandani, Kusumajaya, H., & Permatasari, I. (2024). Faktor-Faktor Yang Berhubungan Dengan Kejadian Congestive Heart Failure (CHF) Pasien. *Jurnal Penelitian Perawat Profesional*, 6(1), 273–284. <http://jurnal.globalhealthsciencegroup.com/index.php/JPPP>
- Rahmianti, N. D., & Trisna, N. P. A. (2020). Ekokardiografi Pada Gagal Jantung. *Medicinus*, 33(1).
- Renaldy, & Surya Dwi Putra, D. (2023). Aplikasi Prediksi Harga Ayam Dengan Metode Naives Bayes Pada Supplier Ayam Potong. *JURNAL ALGOR*, 4(2). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- Rino. (2021). Comparison of Data Mining Methods Using C4.5 Algorithm and Naive Bayes in Predicting Heart Disease. *JournalTECH-E*, 4(2). <http://bsti.ubd.ac.id/e-jurnal>
- Sinaga, L. M., Sawaluddin, & Suwilo, S. (2020). Analysis of Classification and Naïve Bayes Algorithm K-Nearest Neighbor in Data Mining. *IOP Conference Series: Materials Science and Engineering*, 725, 1–5. <https://doi.org/10.1088/1757-899X/725/1/012106>
- Tarsini, I., & Anggraeni, R. (2024). Explore Flowchart and Pseudocode Concepts In Algorithms and Programming. *Indonesian Journal Of Multidisciplinary Science*, 3(5).
- Timotius, A., & Fenriana, I. (2024). Perancangan Aplikasi Prediksi Penyakit Jantung Menggunakan Metode Naive Bayes. *Jurnal Algor*, 5(2). <https://jurnal.buddhidharma.ac.id/index.php/algor/index>
- Viega, M. T., Marvin, E., Jayadi, R., & Mauritsius, T. (2020). Heart Disease Prediction System using Data Mining Classification Techniques: Naïve Bayes, KNN, and Decision Tree. *International Journal of Advanced Trends in Computer Science and Engineering*, 9(3), 3028–3035. <https://doi.org/10.30534/ijatcse/2020/82932020>

LAMPIRAN

PERBANDINGAN ALGORITMA DATA MINING DALAM MEMPREDIKSI KEBERLANGSUNGAN PASIEN AKIBAT GAGAL JANTUNG MENGGUNAKAN MODEL K-NEAREST NEIGHBOR (K-NN) DAN NAIVE BAYES

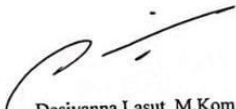
Nama : SUKARNI
Umur : 60 TH
Pekerjaan : IRT

Requirement Elicitation

No	Analisis Kebutuhan Aplikasi	Keterangan
	Saya ingin sistem dapat:	
1.	Memberikan edukasi mengenai penyakit gagal jantung	✓
2.	Memberikan saran perubahan gaya hidup	✓
3.	Memberikan petunjuk penggunaan aplikasi	✓
4.	Memberikan prediksi penyakit gagal jantung	✓
5.	Memberikan saran apabila hasil positif atau negatif	✓
6.	Tampilan aplikasi mudah digunakan dan dipahami	✓
7.	Keamanan aplikasi	✓

Tangerang, 15 Desember 2024

Pembimbing


Desiyanna Lasut, M.Kom
0402128601

Responden


SUKARNI

Mahasiswa


Adisty Maharani
20211000055

Lampiran 1 Form Requirement Elicitation 1

U B D

**PERBANDINGAN ALGORITMA DATA MINING DALAM MEMPREDIKSI
KEBERLANGSUNGAN PASIEN AKIBAT GAGAL JANTUNG MENGGUNAKAN
MODEL K-NEAREST NEIGHBOR (K-NN) DAN NAIVE BAYES**

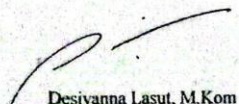
Nama : ERNA EKAWATI
 Umur : 46
 Pekerjaan : Petawak

Requirement Elicitation

No	Analisis Kebutuhan Aplikasi	Keterangan
	Saya ingin sistem dapat:	
1.	Memberikan edukasi mengenai penyakit gagal jantung	✓
2.	Memberikan saran perubahan gaya hidup	✓
3.	Memberikan petunjuk penggunaan aplikasi	✓
4.	Memberikan prediksi penyakit gagal jantung	✓
5.	Memberikan saran apabila hasil positif atau negatif	✓
6.	Tampilan aplikasi mudah digunakan dan dipahami	✓
7.	Keamanan aplikasi	✓

Tangerang, 15 Desember 2024

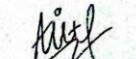
Pembimbing


 Desiyanna Lasut, M.Kom
 0402128601

Responden


 ERNA EKAWATI

Mahasiswa


 Adisty Maharani
 20211000055

Lampiran 2 Form Requirement Elicitation 2

U B D

LAMPIRAN

```
import time
import os
import pandas as pd
import joblib
import json
import logging
import streamlit as st
import hashlib
import mysql.connector
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.metrics import accuracy_score, confusion_matrix,
classification_report
import seaborn as sns
import matplotlib.pyplot as plt

# Konfigurasi Logging
logging.basicConfig(
    filename="app.log",
    level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(message)s"
)

# Konfigurasi database MySQL
DB_CONFIG = {
    "host": "localhost",
    "user": "root",
    "password": "",
    "database": "heart_failure_db"
}

def create_connection():
    try:
        conn = mysql.connector.connect(**DB_CONFIG)
        return conn
    except mysql.connector.Error as e:
        logging.error(f"Error connecting to database: {e}")
        st.error("Gagal terhubung ke database.")
        return None

# Fungsi untuk membuat hash MD5
def hash_password(password):
    return hashlib.md5(password.encode()).hexdigest()
```

```

# Fungsi untuk autentikasi user dengan hash MD5
def authenticate_user(username, password):
    hashed_password = hash_password(password)
    conn = create_connection()
    if conn:
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT * FROM users WHERE username = %s AND password
= %s", (username, hashed_password))
        user = cursor.fetchone()
        conn.close()
        return user
    return None

# Fungsi untuk memuat riwayat training
def load_train_history():
    conn = create_connection()
    if conn:
        cursor = conn.cursor(dictionary=True)
        cursor.execute("SELECT * FROM train_history")
        history = cursor.fetchall()
        conn.close()
        return history
    return []

# Fungsi untuk menyimpan riwayat pelatihan
def save_train_history(entry):
    conn = create_connection()
    if conn:
        cursor = conn.cursor()
        cursor.execute(
            """
            INSERT INTO train_history (timestamp, dataset_name,
knn_train_accuracy, nb_train_accuracy, duration_seconds)
            VALUES (%s, %s, %s, %s, %s)
            """,
            (entry['timestamp'], entry['dataset_name'],
entry['knn_train_accuracy'], entry['nb_train_accuracy'],
entry['duration_seconds'])
        )
        conn.commit()
        conn.close()

# Fungsi load uploaded file
def load_file(uploaded_file):
    if uploaded_file.name.endswith(".csv"):
        return pd.read_csv(uploaded_file)
    else:

```

```

        return pd.read_excel(uploaded_file)

# Streamlit app
if "logged_in" not in st.session_state:
    st.session_state["logged_in"] = False
    st.session_state["user"] = None

# Halaman login
# Halaman login atau register
if not st.session_state["logged_in"]:
    st.title("🔑 Login / Register")
    tab1, tab2 = st.tabs(["Login", "Register"])

    # Tab Login
    with tab1:
        username = st.text_input("Username", key="login_username")
        password = st.text_input("Password", type="password",
key="login_password")
        if st.button("Login"):
            user = authenticate_user(username, password)
            if user:
                st.session_state["logged_in"] = True
                st.session_state["user"] = user
                st.success(f"Selamat datang, {user['username']}!")
                st.rerun()
            else:
                st.error("Username atau password salah.")

    # Tab Register
    with tab2:
        reg_name = st.text_input("Nama Lengkap", key="name")
        reg_username = st.text_input("Username", key="register_username")
        reg_password = st.text_input("Password", type="password",
key="register_password")
        reg_confirm_password = st.text_input("Konfirmasi Password",
type="password", key="register_confirm_password")

        if st.button("Register"):
            if not reg_name or not reg_username or not reg_password:
                st.error("Semua kolom harus diisi!")
            elif reg_password != reg_confirm_password:
                st.error("Password dan Konfirmasi Password tidak cocok!")
            else:
                # Proses penyimpanan ke database
                conn = create_connection()
                if conn:
                    cursor = conn.cursor()
                    try:
                        # Validasi apakah username sudah ada

```

```

        cursor.execute("SELECT username FROM users WHERE
username = %s", (reg_username,))
        existing_user = cursor.fetchone()
        if existing_user:
            st.error("Username sudah digunakan. Silakan
pilih username lain.")
        else:
            # Simpan ke database
            hashed_password = hash_password(reg_password)
            cursor.execute(
                "INSERT INTO users (username, password,
name) VALUES (%s, %s, %s)",
                (reg_username, hashed_password, reg_name)
            )
            conn.commit()
            st.success("Registrasi berhasil! Silakan
login.")
        except mysql.connector.Error as e:
            st.error(f"Terjadi kesalahan: {e}")
            logging.error(f"Error during registration: {e}")
        finally:
            conn.close()

    else:
        # Halaman utama setelah login
        st.sidebar.title(f"Selamat Datang,
{st.session_state['user']['username']}!")
        # Tombol Logout berada di bawah title (masih di sidebar)
        if st.sidebar.button("🚪 Logout"):
            st.session_state["logged_in"] = False
            st.session_state["user"] = None
            st.rerun()
        # Menu navigasi
        menu = st.sidebar.selectbox("Pilih Menu", ["Beranda", "Prediksi Manual",
"Lihat Histori"])

        # Menu Logout
        if menu == "Logout":
            st.session_state["logged_in"] = False
            st.session_state["user"] = None
            st.rerun()

        # Menu Beranda
        elif menu == "Beranda":
            st.subheader("🚫 Selamat Datang di Sistem Prediksi Gagal Jantung")
            # Main page
            st.title("Selamat Datang di Sistem Prediksi Gagal Jantung")

```

```

st.write("""
**🔗 Apa Itu Gagal Jantung?**

Terjadi karena sindrom klinis kompleks yang diakibatkan oleh
kerusakan structural atau fungsional akibat pengisian vantrikel atau pemompa
darah (Nursyamsiah et al., dalam Yuliani Yuri, 2022). Serta gagal jantung itu
sendiri yang dimana jantung gagal untuk memompa yang cukup dalam memenuhi
kebutuhan tubuh (Healen et al., dalam Yuliani Yuri, 2022)

**🔗 Mengapa Gagal Jantung Penting untuk Diperhatikan?**

Gagal jantung adalah salah satu jenis penyakit kardiovaskular
yang semakin sering terjadi, terutama pada orang berusia lanjut. Penting
untuk memahami dan mendeteksi gejalanya lebih awal agar dapat ditangani
dengan baik.
""")

st.markdown("""
**Apabila hasil prediksi <span style='color:red'>POSITIF</span>,
segera konsultasikan dengan dokter**
""", unsafe_allow_html=True)

st.write("""
**Pilih menu di sidebar untuk**:
```

- Prediksi Manual: Halaman prediksi.
- Lihat Histori: Menampilkan histori aplikasi.
- Log Out: Apabila pengguna ingin keluar dari aplikasi.

```

""")

st.write("""
**Perubahan gaya hidup yang disarankan apabila pengguna terprediksi
terkena penyakit gagal jantung (Sumber: halodoc.com)**:
```

- 🥤 Mengurangi asupan natrium.
- 🥗 Konsumsi makanan sehat seperti;
 - buah dan sayur
 - Kacang-kacangan dan polong-polongan,
 - susu rendah lemak,
 - Unggas dan ikan tanpa kulit.
- 💧 Membatasi asupan cairan.
- ⚖️ Mengontrol berat badan.
- 🥞 Membatasi konsumsi garam, gula, dan lemak jenuh.
- 🚭 Berhenti merokok.
- 🏃 Berolahraga secara teratur dan terukur.

```

""")

# Menu: Lihat Log
elif menu == "Lihat Histori":
    st.subheader("📄 Histori Aplikasi")

```

```

# Tombol Hapus Log
if st.button("🗑️ Hapus Histori"): try:
    open("app.log", "w").close() # Mengosongkan isi file log
    st.success("Histori berhasil dihapus.")
    logging.info(f"User {st.session_state['user']['username']}
menghapus histori penggunaan aplikasi.")
except Exception as e:
    st.error("Gagal menghapus Histori.")
    logging.error(f"Gagal menghapus Histori: {e}")

# Tampilkan isi log
try:
    with open("app.log", "r") as log_file:
        log_content = log_file.read()
        if log_content.strip():
            # Ganti 0 dan 1 dengan label Positif/Negatif di histori
            log_content = log_content.replace("Hasil KNN: 1", "Hasil
KNN: Positif")
            log_content = log_content.replace("Hasil KNN: 0", "Hasil
KNN: Negatif")
            log_content = log_content.replace("Hasil NB: 1", "Hasil
NB: Positif")
            log_content = log_content.replace("Hasil NB: 0", "Hasil
NB: Negatif")
            st.text(log_content)
        else:
            st.info("Belum ada histori aplikasi.")
except FileNotFoundError:
    st.warning("File histori tidak ditemukan.")

#Menu Prediksi
if menu == "Prediksi Manual":
    st.subheader("🔍 Prediksi Manual")

    # Form Input
    st.write("### Masukkan Data untuk Prediksi Penyakit Gagal
Jantung 🫀")
    age = st.number_input("Usia", min_value=0, max_value=120,
value=None)
    anaemia = st.selectbox("Anemia", [None, 0, 1], format_func=lambda x:
"Pilih" if x is None else "Tidak" if x == 0 else "Ya")
    creatinine_phosphokinase = st.number_input("Kreatinin Fosfokinase",
min_value=0, value=None)
    diabetes = st.selectbox("Diabetes", [None, 0, 1], format_func=lambda
x: "Pilih" if x is None else "Tidak" if x == 0 else "Ya")
    ejection_fraction = st.number_input("Persentase Pompa Jantung (%)",
min_value=0, max_value=100, value=None)

```

```

        high_blood_pressure = st.selectbox("Tekanan Darah Tinggi", [None, 0, 1], format_func=lambda x: "Pilih" if x is None else "Tidak" if x == 0 else "Ya")
        platelets = st.number_input("Jumlah Trombosit", min_value=0.0, value=None)
        serum_creatinine = st.number_input("Kadar Serum Kreatinin", min_value=0.0, value=None)
        serum_sodium = st.number_input("Kadar Natrium Serum", min_value=0, max_value=200, value=None)
        sex = st.selectbox("Jenis Kelamin", [None, 0, 1], format_func=lambda x: "Pilih" if x is None else "Wanita" if x == 0 else "Pria")
        smoking = st.selectbox("Kebiasaan Merokok", [None, 0, 1], format_func=lambda x: "Pilih" if x is None else "Tidak" if x == 0 else "Ya")
        time_input = 40 # default value

    if st.button("Prediksi"):
        try:
            knn = joblib.load("knn_model.pkl")
            nb = joblib.load("nb_model.pkl")
            scaler = joblib.load("scaler.pkl")

            input_data = pd.DataFrame({
                "age": [age],
                "anaemia": [anaemia],
                "creatinine_phosphokinase": [creatinine_phosphokinase],
                "diabetes": [diabetes],
                "ejection_fraction": [ejection_fraction],
                "high_blood_pressure": [high_blood_pressure],
                "platelets": [platelets],
                "serum_creatinine": [serum_creatinine],
                "serum_sodium": [serum_sodium],
                "sex": [sex],
                "smoking": [smoking],
                "time": [time_input]
            })

            input_scaled = scaler.transform(input_data)

            knn_prediction = knn.predict(input_scaled)[0]
            knn_proba = knn.predict_proba(input_scaled)[0]
            nb_prediction = nb.predict(input_data)[0]
            nb_proba = nb.predict_proba(input_data)[0]

            st.write("### Hasil Prediksi")
            st.write(f"***K-NN Prediction**": {'Positif' if knn_prediction == 1 else 'Negatif'} ({knn_proba[1] * 100:.2f}%)")
            st.write(f"***Naive Bayes Prediction**": {'Positif' if nb_prediction == 1 else 'Negatif'} ({nb_proba[1] * 100:.2f}%)")

```

```

        if knn_prediction != nb_prediction:
            st.warning("⚠ Hasil prediksi dari K-NN dan Naive Bayes
tidak sejalan. Diperlukan pemeriksaan medis lebih lanjut.")
            elif knn_prediction == 1 and nb_prediction == 1:
                st.error("🚫 Hasil menunjukkan kemungkinan kuat gagal
jantung. Segera konsultasikan ke dokter.")
            else:
                st.success("✅ Hasil menunjukkan kondisi normal. Tetap
jaga gaya hidup sehat.")

        logging.info(f"Hasil KNN: {knn_prediction}, Hasil NB:
{nb_prediction}")
    except FileNotFoundError:
        st.error("Model belum tersedia. Silakan lakukan training
model terlebih dahulu.")

# Menu Train Model
elif menu == "Train Model":
    st.subheader("🔍 Train Models (K-NN & Naive Bayes)")

    # Layout dengan kolom
    col1, col2 = st.columns(2)
    with col1:
        st.info("***Langkah 1** : Upload dataset Anda dalam format
CSV/Excel.")
        uploaded_file = st.file_uploader("Upload Dataset", type=["csv",
"xlsx"])
    with col2:
        st.info("***Langkah 2** : Klik **Train Models** untuk memulai
pelatihan.")

    # Display training history
    st.write("### 📖 Riwayat Pelatihan")
    history = load_train_history()
    if history:
        st.write(pd.DataFrame(history))
    else:
        st.write("Belum ada riwayat pelatihan.")

    if uploaded_file:
        try:
            data = load_file(uploaded_file)
            logging.info(f"Dataset '{uploaded_file.name}' berhasil
diunggah.")
            st.write("### Dataset Preview")
            st.write(data.head())

            if "DEATH_EVENT" not in data.columns:
                st.error("Dataset harus memiliki kolom 'DEATH_EVENT'")

```

```

        logging.error("Kolom 'DEATH_EVENT' tidak ditemukan dalam
dataset.")
    else:
        if st.button("Train Models"):
            start_time = time.time()

            # Pembagian data
            X = data.drop(columns=["DEATH_EVENT"])
            y = data["DEATH_EVENT"]
            X_train, X_test, y_train, y_test = train_test_split(
                X, y, test_size=0.2, random_state=42, stratify=y
            )

            # Scaling data
            scaler = StandardScaler()
            X_train_scaled = scaler.fit_transform(X_train)
            X_test_scaled = scaler.transform(X_test)

            # Train K-NN
            knn = KNeighborsClassifier(n_neighbors=5)
            knn.fit(X_train_scaled, y_train)

            # Train Naive Bayes
            nb = GaussianNB()
            nb.fit(X_train, y_train)

            # Save model and scaler
            joblib.dump(knn, "knn_model.pkl")
            joblib.dump(nb, "nb_model.pkl")
            joblib.dump(scaler, "scaler.pkl")

            end_time = time.time()
            duration = end_time - start_time
            logging.info("Model berhasil dilatih.")

            st.success("Model berhasil dilatih!")

            # Akurasi Training
            knn_train_acc = accuracy_score(y_train,
knn.predict(X_train_scaled))
            nb_train_acc = accuracy_score(y_train,
nb.predict(X_train))

            # Save history
            new_entry = {
                "timestamp": time.strftime("%Y-%m-%d %H:%M:%S"),
                "dataset_name": uploaded_file.name,
                "knn_train_accuracy": knn_train_acc,
                "nb_train_accuracy": nb_train_acc,

```

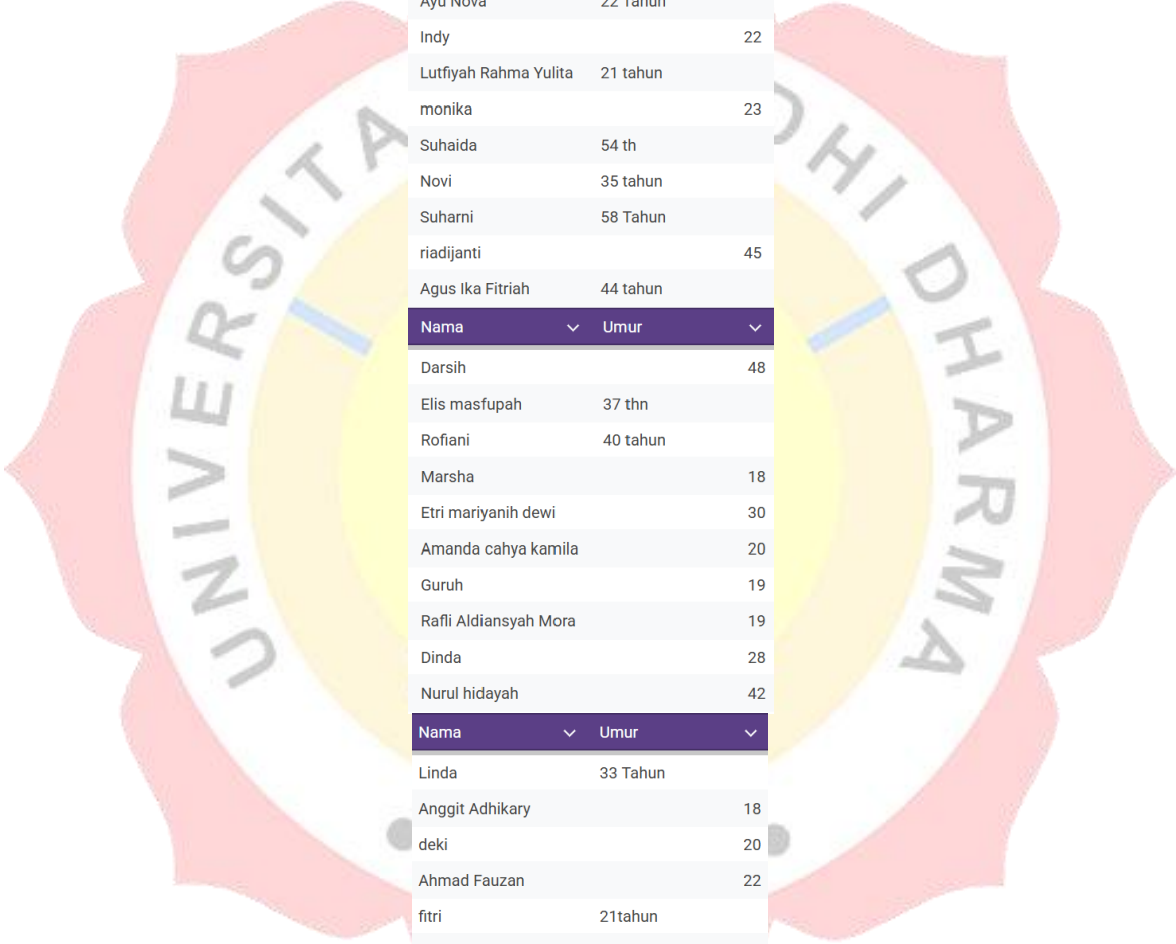
```

        "duration_seconds": duration,
    }
    save_train_history(new_entry)

except Exception as e:
    st.error("Terjadi kesalahan saat memproses dataset.")
    logging.error(f"Error: {e}")

```

Lampiran 3 Listing Program

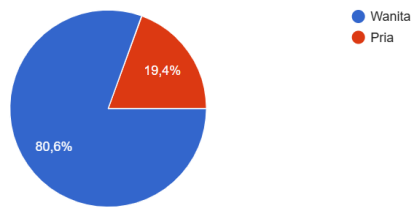


Nama	Umur
nasyie	20
Ayu Nova	22 Tahun
Indy	22
Lutfiyah Rahma Yulita	21 tahun
monika	23
Suhaida	54 th
Novi	35 tahun
Suharni	58 Tahun
riadijanti	45
Agus Ika Fitriah	44 tahun
Nama	Umur
Darsih	48
Elis masfupah	37 thn
Rofiani	40 tahun
Marsha	18
Etri mariyanihi dewi	30
Amanda cahya kamila	20
Guruh	19
Rafli Aldiansyah Mora	19
Dinda	28
Nurul hidayah	42
Nama	Umur
Linda	33 Tahun
Anggit Adhikary	18
deki	20
Ahmad Fauzan	22
fitri	21tahun
nasyie	20 th
Leli	22
Muhamad Calvin	22
Husnul Khotimi	25
Zakiyyah Kafka	19
Nama	Umur
Husna	22
Aslamul Ihsan	22
Arya	22
Irma Muthia	21 Tahun
Erna Ekawati	45
Sukarmi	60

Jenis Kelamin

36 jawaban

[Salin diagram](#)

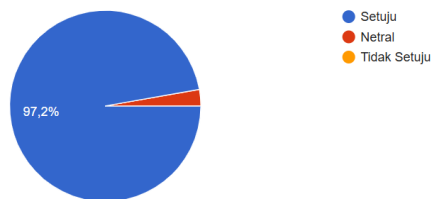


User Acceptance Testing (UAT): Sistem Prediksi Gagal Jantung

1. Tampilan aplikasi mudah digunakan dan dipahami

36 jawaban

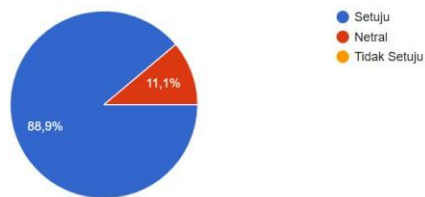
[Salin diagram](#)



2. Hasil prediksi yang ditampilkan mudah dipahami dan disajikan secara jelas

36 jawaban

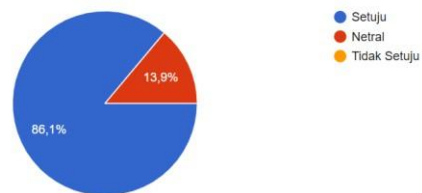
[Salin diagram](#)



3. Saya merasa terbantu dengan adanya fitur prediksi yang disediakan dalam aplikasi ini

36 jawaban

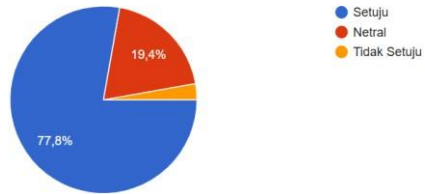
[Salin diagram](#)



4. Pertanyaan mengenai gejala disampaikan dengan jelas dan mudah untuk dijawab

36 jawaban

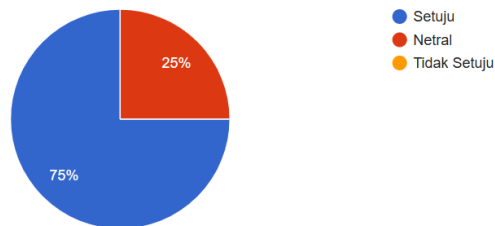
[Salin diagram](#)



5. Proses prediksi berjalan dengan cepat

36 jawaban

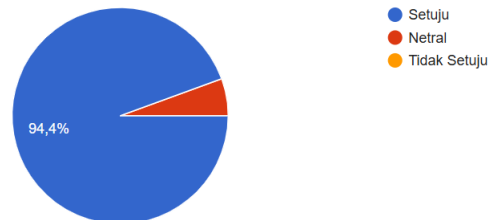
[Salin diagram](#)



6. Hasil prediksi yang tersedia dapat membantu saya dalam memantau kondisi kesehatan secara lebih efektif

36 jawaban

[Salin diagram](#)



Lampiran 4 Hasil Jawaban pada Google Form

DAFTAR RIWAYAT HIDUP



Data Pribadi

Nama Lengkap : Adisty Maharani
Tempat, Tanggal Lahir : Tangerang, 05 Mei 2002
Jenis Kelamin : Perempuan
Alamat : Persada Raya Blok E.10 No.19
Agama : Islam
No Kontak : 085770126902
eMail : Adistyy05@gmail.com

Pendidikan Formal

2007 - 2008 : Taman Kanak-Kanak Fajar
2008 - 2014 : Sekolah Dasar Negeri Total Persada
2014 - 2017 : Sekolah Menengah Pertama Negeri 15 Kota Tangerang
2017 - 2020 : Sekolah Menengah Atas Negeri 15 Kota Tangerang
2021 - Sekarang : Teknik Informatika, *Data Science*
Universitas Buddhi Dharma

Pengalaman Kerja

2022 - 2024 : Himpunan Mahasiswa Teknik Informatika, Universitas Buddhi Dharma, Ketua Divisi *Relation, Reasearch and Development*

Tangerang, 04 Agustus 2025

Adisty Maharani

**UNIVERSITAS BUDDHI DHARMA**

Jl. Imam Bonjol No. 41 Karawaci Ilir, Tangerang
021 5517853 / 021 5586822 | admin@buddhidharma.ac.id

KARTU BIMBINGAN TA/SKRIPSI

NIM : 20211000055
Nama Mahasiswa : ADISTY MAHARANI
Fakultas : Sains dan Teknologi
Program Studi : Teknik Informatika
Jenjang : Strata Satu
Tahun Akademik/Semester : 2024/2025 Genap
Dosen Pembimbing : Desiyanna Lasut, S.Kom., M.Kom
Judul Skripsi : PERBANDINGAN ALGORITMA DATA MINING DALAM
MEMPREDIKSI KEBERLANGSUNGAN HIDUP PASIEN AKIBAT
GAGAL JANTUNG MENGGUNAKAN MODEL K-NEAREST
NEIGHBOR (K-NN) DAN NAIVE BAYES

Tanggal	Catatan	Paraf
2025-03-04	Pengajuan judul	
2025-03-18	Acc Judul dan jurnal referensi	
2025-04-08	Bab 1	
2025-04-22	Bab 2	
2025-04-29	Bab 3	
2025-05-13	Bab 4	
2025-05-27	Bab 5	
2025-06-03	Final program	
2025-06-10	Persentasi	

Mengetahui
Ketua Program Studi



Hartana Wijaya, M.Kom

Tangerang, 23 June 2025
Pembimbing



Desiyanna Lasut, S.Kom., M.Kom

Lampiran 5 Kartu Bimbingan