

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Berdasarkan penelitian tentang analisis sentimen tentang pengaruh stres akademik terhadap risiko *gastritis* pada mahasiswa Universitas Jenderal Achmad Yani Yogyakarta menggunakan metode *Naïve Bayes Classification*, tahapan yang dilakukan dalam penelitian ini yaitu pengambilan data, preprocessing, pelabelan, pembagian data, implementasi model, evaluasi model. Data diperoleh melalui kuesioner *online* menggunakan Google Form yang disebarakan kepada mahasiswa dari tiga fakultas, yaitu FTTI, FES, dan FKES. Jumlah responden sebanyak 108 mahasiswa, terdiri dari 20 laki-laki dan 88 perempuan, yang berasal dari semester 2, 4, 6, dan 8. Instrumen kuesioner telah melalui uji validitas dan reliabilitas, dengan hasil valid karena nilai $r\text{-hitung} > r\text{-tabel}$ (0,189), dan reliabel karena nilai *Cronbach's Alpha* $> 0,6$. Selanjutnya, data yang terkumpul dibagi menjadi dua bagian untuk keperluan pelatihan dan pengujian model. Sebanyak 86 data digunakan sebagai data training, sementara 22 data sisanya digunakan sebagai data *testing*.

Hasil penelitian menunjukkan bahwa tingkat stres akademik memiliki hubungan yang signifikan terhadap risiko *gastritis* pada mahasiswa. Berdasarkan analisis data, jumlah mahasiswa yang berisiko mengalami *gastritis* bervariasi pada setiap kategori stres. Kategori stres parah menunjukkan jumlah mahasiswa berisiko paling tinggi, yaitu sebanyak 84 orang, sedangkan mahasiswa yang tidak berisiko hanya 8 orang. Ini menunjukkan bahwa mayoritas mahasiswa dengan tingkat stres yang sangat tinggi cenderung memiliki risiko *gastritis*. Sementara itu, pada kategori stres ringan, terdapat 2 mahasiswa yang berisiko dan 5 mahasiswa yang tidak berisiko. Kategori sedang menunjukkan 5 mahasiswa berisiko dan 3 mahasiswa tidak berisiko. Adapun kategori tidak stres hanya terdiri dari 1 mahasiswa, dan yang bersangkutan tidak termasuk dalam kelompok berisiko *gastritis*. Temuan ini mengindikasikan bahwa semakin tinggi tingkat stres akademik yang dialami

mahasiswa, maka semakin besar pula kemungkinan mereka mengalami risiko *gastritis*. Sebaliknya, pada tingkat stres yang lebih rendah, kecenderungan mengalami *gastritis* juga lebih kecil. Hasil ini akan dibahas lebih lanjut untuk memberikan pemahaman yang lebih mendalam mengenai hubungan antara stres akademik dan risiko gangguan kesehatan pencernaan, khususnya *gastritis*, di kalangan mahasiswa Universitas Jenderal Achmad Yani Yogyakarta.

4.2 PEMBAHASAN

4.2.1 Pengumpulan Data

Pengumpulan data pada penelitian ini dilakukan melalui kuesioner yang dirancang untuk mengukur tingkat stres akademik serta riwayat *gastritis* pada mahasiswa. Data diperoleh dari 108 responden yang terdiri dari mahasiswa semester 2, 4, 6, dan 8 di Fakultas Kesehatan, Fakultas Ekonomi dan Sosial, serta Fakultas Teknik dan Teknologi Informasi Universitas Jenderal Achmad Yani Yogyakarta. Informasi yang dikumpulkan selanjutnya diproses melalui tahap pemrosesan awal (*preprocessing*) untuk memastikan hasil analisis yang diperoleh sesuai dengan yang diharapkan.

4.2.2 *Preprocessing Data*

Data yang telah dikumpulkan kemudian diproses untuk memastikan kualitas dan konsistensinya sebelum analisis. Proses ini meliputi penghapusan kolom tidak relevan, pembersihan dan encoding kolom kategorikal seperti Jenis Kelamin dan Fakultas, penghapusan kolom identitas pribadi untuk menjaga kerahasiaan, validasi data usia agar sesuai rentang wajar, serta konversi semua kolom ke bentuk numerik agar data siap digunakan dalam algoritma analisis penelitian.

1. Penghapusan Kolom Yang Tidak Penting

Langkah pertama dalam preprocessing adalah menghapus kolom seperti *Timestamp*, kolom kosong *Unnamed*, karena tidak relevan dengan analisis. Kode yang digunakan untuk menghapus kolom yang tidak penting adalah sebagai berikut:

```
if "Timestamp" in df.columns:
    df.drop(columns=["Timestamp"], inplace=True)
```

```
unnamed_cols = [col for col in df.columns if "Unnamed" in col]
df.drop(columns=unnamed_cols, inplace=True)
```

Kode diatas digunakan untuk menghapus kolom yang tidak relevan, yaitu kolom *Timestamp* dan kolom kosong *Unnamed*, agar data lebih bersih dan hanya menyisakan informasi yang diperlukan untuk analisis.

2. Membersihkan Kolom Kategorikal

Membersihkan spasi dan mengubah huruf kecil pada Jenis Kelamin, lalu meng-*encode* ke numerik serta membersihkan Fakultas dan melakukan *one-hot encoding*. Kode yang digunakan untuk membersihkan kolom kategorikal adalah sebagai berikut:

```
df["Jenis Kelamin"] = df["Jenis Kelamin"].str.strip().str.lower()
df["Jenis Kelamin"] = df["Jenis Kelamin"].map({
    "laki - laki": 1,
    "perempuan": 0
})

df["Jenis Kelamin"].value_counts()

df["Fakultas"] = df["Fakultas"].str.strip()
df = pd.get_dummies(df, columns=["Fakultas"], drop_first=False,
dtype=int)
df.head()
```

Kode di atas digunakan untuk membersihkan dan mengonversi kolom kategorikal agar siap dianalisis secara numerik. Pada kolom Jenis Kelamin, spasi di awal atau akhir kata dibersihkan dan semua huruf diubah ke huruf kecil, kemudian nilainya di-mapping ke angka 1 untuk laki-laki dan 0 untuk perempuan. Sementara itu, pada kolom Fakultas, spasi juga dibersihkan, lalu dilakukan *one-hot encoding* untuk mengubah kategori fakultas menjadi variabel numerik biner, sehingga data dapat diolah dengan algoritma analisis yang membutuhkan input numerik.

3. Menghapus Kolom Data Pribadi

Menghapus kolom Email, Nama Lengkap, dan pertanyaan diagnosis *gastritis* untuk menjaga privasi dan fokus pada data yang diperlukan. Kode yang digunakan untuk membersihkan kolom kategorikal adalah sebagai berikut:

```
df.drop(columns=[
    "Email",
    "Nama Lengkap",
    "Apakah anda pernah didiagnosis mengalami gastritis?"
])
```

```
], inplace=True)
df.columns
```

Kode di atas digunakan untuk menghapus kolom-kolom yang memuat informasi identitas pribadi dan pertanyaan yang tidak diperlukan untuk analisis, yaitu Email, Nama Lengkap, dan kolom diagnosis *gastritis*. Langkah ini penting untuk menjaga kerahasiaan responden serta memfokuskan data hanya pada variabel yang mendukung penelitian.

4. Membersihkan Kolom Usia

Mengekstrak angka usia, memastikan usia dalam rentang 15–35 tahun, dan menghapus baris dengan usia tidak valid. Kode yang digunakan adalah sebagai berikut:

```
def bersihkan_usia(val):
    if pd.isna(val):
        return None
    val_str = str(val).strip().lower()
    angka = ''.join(filter(str.isdigit, val_str))
    if len(angka) >= 2:
        usia = int(angka[:2])
        if 15 <= usia <= 35:
            return usia
    return None

df["Usia"] = df["Usia"].apply(bersihkan_usia)
df = df[df["Usia"].notna()]

print("✅ Kolom 'Usia' dibersihkan dan divalidasi.")
```

Kode di atas digunakan untuk membersihkan data pada kolom Usia dengan mengekstrak angka dari input, kemudian memvalidasi agar hanya usia dalam rentang 15–35 tahun yang diterima. Rentang ini dipilih karena mewakili usia mahasiswa yang wajar secara akademik, sehingga data di luar rentang ini dianggap tidak valid atau merupakan kesalahan input. Baris dengan usia di luar rentang atau nilai kosong dihapus agar analisis hanya menggunakan data yang relevan dan akurat.

5. Mengonversi Semua Kolom Menjadi Numerik

Mengubah seluruh kolom ke tipe numerik agar dataset siap digunakan oleh algoritma analisis. Kode yang digunakan adalah sebagai berikut:

```
df = df.apply(pd.to_numeric, errors='coerce')
df.info()
```

Kode di atas digunakan untuk mengonversi seluruh kolom dalam dataset ke tipe numerik, sehingga memastikan semua variabel dapat diproses oleh algoritma analisis yang membutuhkan input numerik. Baris dengan nilai yang tidak bisa dikonversi otomatis akan diatur menjadi *NaN* agar dapat ditangani lebih lanjut sesuai kebutuhan analisis.

6. Mendeteksi dan Menangani Outlier

Kode yang digunakan untuk mendeteksi dan menangani outlier adalah sebagai berikut

```
def handle_outliers_iqr(df, column):
    if column not in df.columns or not
pd.api.types.is_numeric_dtype(df[column]):
        print(f"Kolom '{column}' tidak ditemukan atau bukan tipe
numerik. Melewatkan penanganan outlier.")
        return df

    Q1 = df[column].quantile(0.25)
    Q3 = df[column].quantile(0.75)
    IQR = Q3 - Q1

    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR

    outliers = df[(df[column] < lower_bound) | (df[column] >
upper_bound)]

    print(f"Menemukan {len(outliers)} outlier di kolom '{column}'
menggunakan IQR.")

    df[column] = df[column].clip(lower=lower_bound,
upper=upper_bound)
    print(f"Outlier di kolom '{column}' telah diganti dengan batas
(capping).")

    return df

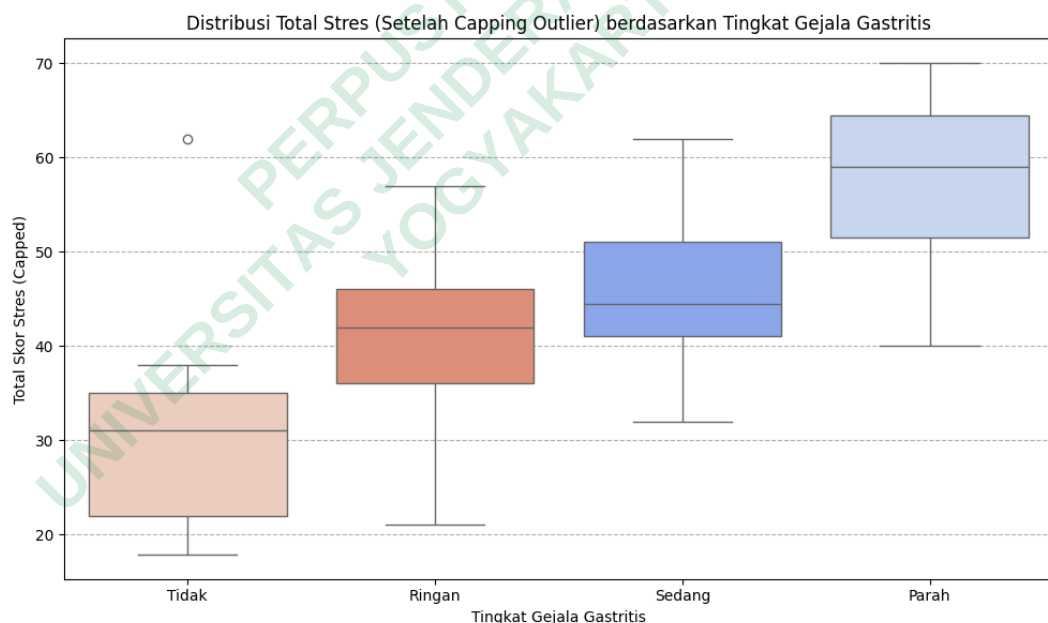
df_cleaned = df.copy()

df_cleaned = handle_outliers_iqr(df_cleaned, 'Total_Stres')
df_cleaned = handle_outliers_iqr(df_cleaned, 'Total_gastritis')

plt.figure(figsize=(10, 6))
```

```
sns.boxplot(x='Kategori_Gastritis', y='Total_Stres',
data=df_cleaned, order=["Tidak", "Ringan", "Sedang", "Parah"],
palette='coolwarm', hue='Kategori_Gastritis', legend=False)
plt.title('Distribusi Total Stres (Setelah Capping Outlier)
berdasarkan Tingkat Gejala Gastritis')
plt.xlabel('Tingkat Gejala Gastritis')
plt.ylabel('Total Skor Stres (Capped)')
plt.grid(axis='y', linestyle='--')
plt.tight_layout()
plt.show()
```

Kode diatas digunakan untuk menangani outlier pada kolom *Total_Stres* dan *Total_gastritis* dengan metode IQR (Interquartile Range) dan melakukan capping, yaitu mengganti nilai ekstrim dengan batas bawah atau atas agar distribusi data tetap stabil. Setelah itu, dilakukan visualisasi menggunakan boxplot untuk melihat distribusi skor stres berdasarkan kategori gejala *gastritis* setelah outlier ditangani, sehingga memberikan gambaran yang lebih representatif terhadap data sebenarnya. Hasil dari deteksi dan penanganan outlier dapat dilihat pada Gambar 4.1.



Gambar 4.1 Hasil Mendeteksi dan Menangani Outlier

Pada Gambar 4.1 ditampilkan distribusi skor stres berdasarkan tingkat gejala *gastritis* mahasiswa, yang mencakup kategori "Tidak", "Ringan", "Sedang", dan "Parah". Pada kategori "Tidak", skor stres terlihat cukup simetris dengan rentang utama antara 20–38, meskipun terdapat satu outlier ekstrem (~62) yang

menunjukkan adanya mahasiswa dengan stres tinggi meskipun tidak mengalami gejala *gastritis*. Sementara pada kategori “Ringan”, skor stres tersebar antara (~21) hingga (~57), dengan rentang utama berada pada 35–50, mencerminkan bahwa mahasiswa dengan gejala ringan cenderung mengalami stres pada tingkat sedang hingga tinggi.

Distribusi semakin melebar pada kategori “Sedang”, dengan rentang sekitar 40–55 dan satu outlier kecil di bagian bawah (~32), menandakan bahwa sebagian besar mahasiswa dalam kategori ini mengalami stres tinggi. Adapun pada kategori “Parah”, skor stres menyebar paling luas, berkisar antara 50–70, dan median yang paling tinggi di antara semua kategori tanpa adanya outlier mencolok. Pola ini memperlihatkan bahwa semakin berat gejala *gastritis* yang dialami mahasiswa, maka semakin tinggi pula tingkat stresnya. Hal ini memperkuat adanya korelasi positif antara stres akademik dan risiko gejala *gastritis*.

4.2.3 Labelling Data

Data yang telah diproses diberikan label berdasarkan kategori, yaitu mahasiswa yang berisiko mengalami *gastritis* dan yang tidak berisiko, guna mempermudah proses klasifikasi. Kode yang digunakan untuk *labelling* adalah sebagai berikut:

```
def label_risiko(row):
    if row["Total_Stres"] >= 18 and row["Total_gastritis"] >= 10:
        return 1
    else:
        return 0
df["Gastritis"] = df.apply(label_risiko, axis=1)
```

Kode di atas digunakan untuk memberikan *label* pada setiap responden berdasarkan nilai total stres dan total gejala *gastritis* yang diperoleh dari kuesioner. Fungsi `label_risiko()` dibuat untuk mengevaluasi dua variabel utama, yaitu `Total_Stres` dan `Total_gastritis`, dengan kriteria jika seorang mahasiswa memiliki skor `Total_Stres` ≥ 18 dan `Total_gastritis` ≥ 10 , maka mahasiswa tersebut dikategorikan berisiko mengalami *gastritis* dan diberikan label 1. Sebaliknya, jika tidak memenuhi kedua syarat tersebut, mahasiswa dianggap tidak berisiko dan diberi label 0.

Proses ini kemudian diterapkan ke seluruh data menggunakan metode *df.apply()* sehingga menghasilkan kolom baru bernama *Gastritis* yang merepresentasikan label risiko untuk masing-masing responden. Tahap *labeling* ini bertujuan untuk mengklasifikasikan data menjadi dua kategori yang jelas, yaitu berisiko dan tidak berisiko, sehingga data siap digunakan dalam proses pemodelan klasifikasi selanjutnya dengan algoritma yang memerlukan target variabel.

4.2.4 Pembagian Data Uji dan Data Latih

Dataset dibagi menjadi dua bagian, yaitu 80% sebagai data latih yang digunakan untuk membangun model prediksi, dan 20% sebagai data uji yang digunakan untuk menilai performa model dalam mengklasifikasikan tingkat risiko *gastritis*. Kode yang digunakan untuk *split* data adalah sebagai berikut:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42, stratify=y
)

print(f"Jumlah data training: {len(X_train)}")
print(f"Jumlah data testing: {len(X_test)}")
```

Kode di atas digunakan untuk membagi dataset menjadi dua bagian, yaitu data latih sebanyak 80% dan data uji sebanyak 20%, dengan menggunakan fungsi *train_test_split* dari *scikit-learn*. Pemisahan ini dilakukan secara acak namun mempertahankan proporsi *label* melalui parameter *stratify=y*, sehingga distribusi kategori berisiko dan tidak berisiko tetap seimbang pada kedua set data. Langkah ini penting untuk memastikan model dapat dilatih secara optimal pada data latih dan dievaluasi secara akurat pada data uji, sehingga hasil analisis memiliki tingkat keandalan yang baik.

4.2.5 Implementasi Model

Model klasifikasi dibangun dengan menggunakan algoritma *Naïve Bayes* yang menghitung kemungkinan untuk setiap kategori. berdasarkan fitur yang tersedia, sehingga model dapat mengidentifikasi pola hubungan antara stres

akademik dan *gastritis*. Kode yang digunakan untuk implementasi *Naïve Bayes* adalah sebagai berikut:

```
from sklearn.naive_bayes import GaussianNB

model_nb = GaussianNB()
model_nb.fit(X_train_scaled, y_train)
```

Kode di atas digunakan untuk membuat model klasifikasi dengan algoritma *Naïve Bayes* menggunakan metode *GaussianNB* dari *library scikit-learn*. Model ini kemudian dilatih dengan data latih yang telah melalui proses normalisasi, yaitu *X_train_scaled* sebagai fitur *input* dan *y_train* sebagai variabel target. Tahap pelatihan ini memungkinkan model untuk memahami pola hubungan antara tingkat stres akademik yang dialami mahasiswa dengan kemungkinan risiko *gastritis*.

Melalui proses ini, diharapkan model mampu mengelompokkan mahasiswa ke dalam kategori berisiko atau tidak berisiko berdasarkan data tingkat stres sehingga dapat membantu dalam menganalisis keterkaitan stres akademik dengan risiko terjadinya *gastritis* secara lebih tepat.

4.2.6 Evaluasi Model

Setelah model dibangun, langkah selanjutnya adalah menguji performanya dengan menghitung tingkat akurasi. Evaluasi dilakukan menggunakan metrik akurasi, presisi, *recall*, dan *F1-score* untuk menilai kemampuan model dalam mengklasifikasikan risiko *gastritis* secara menyeluruh. Kode yang digunakan untuk melakukan evaluasi model adalah sebagai berikut:

```
y_pred = model_nb.predict(X_test_scaled)

print("=== Confusion Matrix ===")
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import precision_score, recall_score, f1_score

akurasi = accuracy_score(y_test, y_pred)
presisi = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

print("\n=== Accuracy Score ===")
print(f"Akurasi: {accuracy_score(y_test, y_pred) * 100:.2f}%")
```

```

print("\n=== Precision Score ===")
print(f"Presisi: {precision_score(y_test, y_pred) * 100:.2f}%")

print("\n=== Recall Score ===")
print(f"Recall: {recall_score(y_test, y_pred) * 100:.2f}%")

print("\n=== F1 Score ===")
print(f"F1 Score: {f1_score(y_test, y_pred) * 100:.2f}%")

```

Kode di atas digunakan untuk mengevaluasi kinerja model *Naïve Bayes* dalam mengklasifikasikan risiko *gastritis* pada data *testing*. Hasil dari *Confusion Matrix* dapat dilihat pada Tabel 4.1.

Tabel 4.1 *Confusion Matrix*

Prediksi	Aktual	
	Positif	Negatif
Positif	18	1
Negatif	1	2

Pada Tabel 4.1 menghasilkan nilai TP sebesar 18, nilai FN sebesar 1, nilai FP sebesar 1, nilai TN sebesar 2. Hasil *accuracy*, *precision*, *recall*, *f1 score* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Akurasi Data

Akurasi	Nilai
<i>Accuracy</i>	91%
<i>Precision</i>	95%
<i>Recall</i>	95%
<i>F1-Score</i>	95%

Pada Tabel 4.2 menampilkan akurasi data yang sangat baik. Tabel di atas diperoleh nilai akurasi sebesar 91%, nilai *precision* mendapatkan 95%, nilai *recall* mendapatkan 95%, dan pada nilai *F1-Score* mendapatkan hasil sebesar 95%.

4.2.7 Hasil Analisis

Hasil dari analisis ini menggunakan kategori stres dan kategori *gastritis*. Kode yang digunakan untuk kategori stres adalah sebagai berikut:

```

stres_cols = [col for col in df.columns if any(kata in col.lower()
for kata in [
    "tugas", "ujian", "nilai", "presentasi", "skripsi", "waktu",
    "deadline", "akademik"
])]
df["Total_Stres"] = df[stres_cols].sum(axis=1)
def kategori_stres(skor):
    if skor >= 33:
        return "Parah"
    elif skor >= 25:
        return "Sedang"
    elif skor >= 18:
        return "Ringan"
    else:
        return "Tidak Stres"
df["Kategori_Stres"] = df["Total_Stres"].apply(kategori_stres)
df[["Total_Stres", "Kategori_Stres"]].head(10)

```

kode diatas digunakan untuk menghitung total skor stres akademik berdasarkan beberapa indikator, lalu mengelompokkan skor tersebut ke dalam kategori stres yaitu tidak stres, ringan, sedang, dan parah. Hasil dari kategori stress dapat dilihat pada Gambar 4.2.

	Total_Stres	Kategori_Stres
0	48	Parah
1	62	Parah
2	56	Parah
3	40	Parah
4	21	Ringan
5	29	Sedang
6	62	Parah
7	40	Parah
8	22	Ringan
9	70	Parah

Gambar 4.2 Kategori Stres

Berdasarkan Gambar 4.2 dapat diketahui bahwa mayoritas mahasiswa berada pada kategori stres parah, diikuti oleh kategori sedang, ringan, dan tidak stres. Hal ini menunjukkan bahwa sebagian besar responden mengalami tingkat stres akademik yang tinggi, yang berpotensi memengaruhi kondisi kesehatan

mereka, khususnya terkait risiko *gastritis*. Setelah itu, dilakukan kategori untuk *gastritis*, kode yang digunakan untuk kategori *gastritis* adalah sebagai berikut:

```
gastritis_cols = [col for col in df.columns if any(kata in
col.lower() for kata in [
    "nyeri", "mual", "muntah", "terbakar", "kembung", "nafsu"
])]
df["Total_gastritis"] = df[gastritis_cols].sum(axis=1)

df["Gastritis"] = df["Total_gastritis"].apply(lambda x: 1 if x >=
10 else 0)

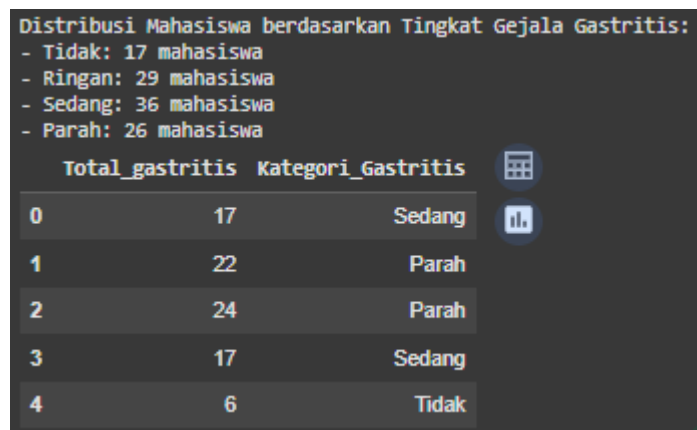
def kategori_gastritis(skor):
    if skor > 20:
        return "Parah"
    elif skor >= 15:
        return "Sedang"
    elif skor >= 10:
        return "Ringan"
    else:
        return "Tidak"

df["Kategori_Gastritis"] =
df["Total_gastritis"].apply(kategori_gastritis)

jumlah_kategori =
df["Kategori_Gastritis"].value_counts().reindex(["Tidak",
"Ringan", "Sedang", "Parah"]).fillna(0).astype(int)
print("Distribusi Mahasiswa berdasarkan Tingkat Gejala
Gastritis:")
for kategori in ["Tidak", "Ringan", "Sedang", "Parah"]:
    print(f"- {kategori}: {jumlah_kategori[kategori]} mahasiswa")

df[["Total_gastritis", "Kategori_Gastritis"]].head()
```

Kode di atas digunakan untuk menghitung total skor gejala *gastritis* berdasarkan sejumlah indikator seperti nyeri, mual, muntah, sensasi terbakar, perut kembung, dan hilangnya nafsu makan. Skor total tersebut kemudian digunakan untuk menentukan klasifikasi risiko *gastritis* ke dalam empat kategori, yaitu tidak, ringan, sedang, dan parah. Hasil dari kategori *gastritis* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Kategori *Gastritis*

Berdasarkan Gambar 4.3, proporsi gejala *gastritis* terbanyak terdapat pada kategori sedang, diikuti oleh kategori ringan dan parah, sedangkan kategori tanpa gejala menempati porsi paling kecil. Distribusi ini menggambarkan bahwa mayoritas mahasiswa mengalami gejala *gastritis* sehingga perlu mendapat perhatian khusus dalam upaya pencegahan dan penanganan lebih lanjut.

Setelah kategori selesai maka selanjutnya dilakukan analisis pengaruh stres terhadap risiko *gastritis*. Kode yang digunakan untuk menganalisis pengaruh stres terhadap *gastritis* adalah sebagai berikut:

```
print("\n=== Jumlah Mahasiswa dengan Risiko Gastritis Berdasarkan
Tingkat Stres ===")
risk_by_stres_count =
df.groupby('Kategori_Stres')['Gastritis'].value_counts().unstack
().fillna(0).astype(int)
risk_by_stres_count = risk_by_stres_count.sort_index()
print(risk_by_stres_count)

print("\n=== Interpretasi Jumlah Mahasiswa Risiko Gastritis per
Tingkat Stres ===")
for kategori, row in risk_by_stres_count.iterrows():

    jumlah_berisiko = row[1] if 1 in row else 0
    jumlah_tidak_berisiko = row[0] if 0 in row else 0
    print(
        f"- {kategori}: {jumlah_berisiko} mahasiswa berisiko, "
        f"{jumlah_tidak_berisiko} mahasiswa tidak berisiko."
    )
```

Kode di atas digunakan untuk menganalisis hubungan antara tingkat stres akademik dengan risiko *gastritis* pada mahasiswa. Proses ini dilakukan dengan

mengelompokkan data berdasarkan kategori stres, kemudian menghitung jumlah mahasiswa yang tergolong berisiko dan tidak berisiko *gastritis* pada masing-masing kategori. Hasil analisis dapat dilihat pada Gambar 4.4.

```

=== Jumlah Mahasiswa dengan Risiko Gastritis Berdasarkan Tingkat Stres ===
Gastritis      0    1
Kategori_Stres
Parah          8   84
Ringan         5    2
Sedang         3    5
Tidak Stres    1    0

=== Interpretasi Jumlah Mahasiswa Risiko Gastritis per Tingkat Stres ===
- Parah: 84 mahasiswa berisiko, 8 mahasiswa tidak berisiko.
- Ringan: 2 mahasiswa berisiko, 5 mahasiswa tidak berisiko.
- Sedang: 5 mahasiswa berisiko, 3 mahasiswa tidak berisiko.
- Tidak Stres: 0 mahasiswa berisiko, 1 mahasiswa tidak berisiko.

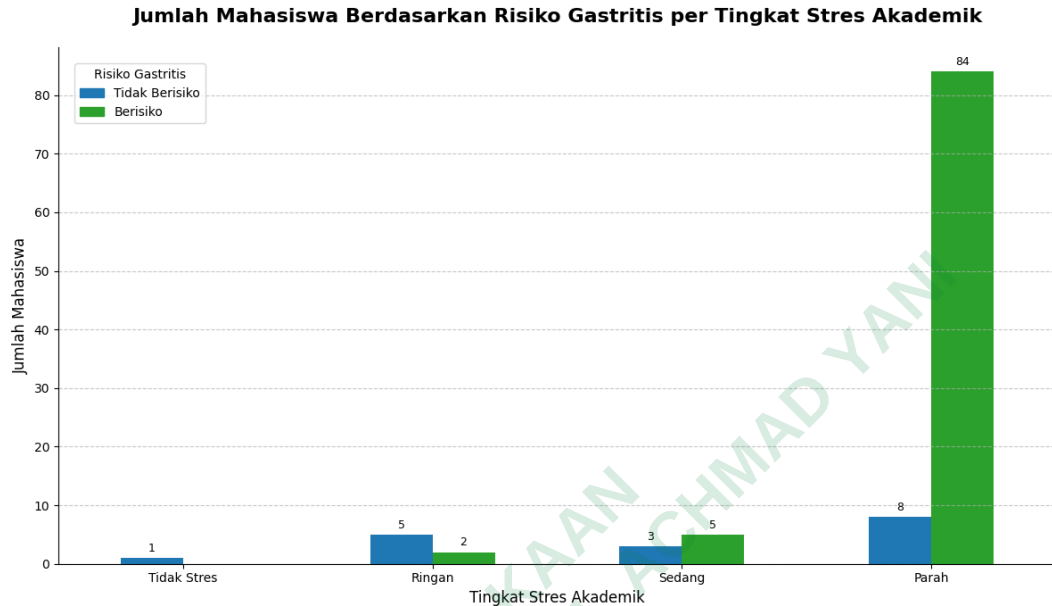
```

Gambar 4.4 Hasil Analisis

Gambar 4.4 memperlihatkan bahwa sebagian besar mahasiswa yang berada pada tingkat stres parah cenderung memiliki risiko gastritis yang tinggi, di mana tercatat sebanyak 84 mahasiswa termasuk dalam kategori berisiko dan hanya 8 orang yang tidak berisiko. Untuk kategori stres sedang, terdapat 5 mahasiswa yang berisiko mengalami gastritis dan 3 lainnya tidak berisiko. Pada tingkat stres ringan, hanya 2 mahasiswa yang termasuk berisiko, sedangkan 5 mahasiswa tidak berisiko. Adapun pada kelompok mahasiswa tanpa stres, hanya terdapat 1 orang yang tidak menunjukkan risiko gastritis sama sekali. Temuan ini mengindikasikan bahwa semakin tinggi tingkat stres akademik, maka semakin besar pula kemungkinan mahasiswa mengalami gastritis. Sebaliknya, semakin rendah tingkat stres, jumlah mahasiswa yang tidak berisiko mengalami gastritis cenderung lebih tinggi.

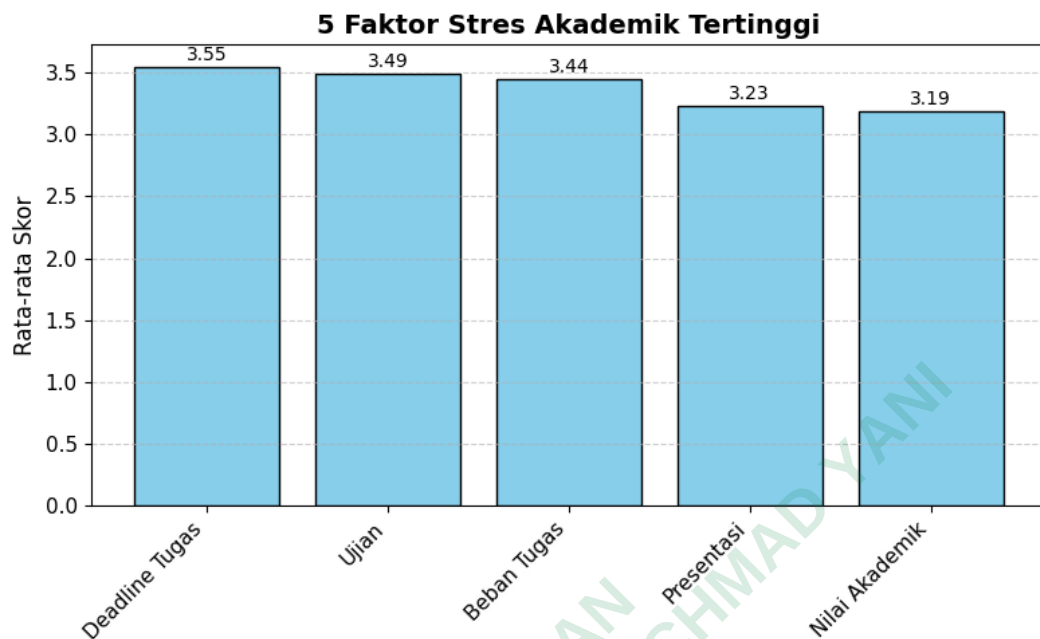
Selanjutnya, pola distribusi ini mencerminkan adanya keterkaitan yang signifikan dan bersifat positif antara tingkat stres akademik dan risiko gastritis. Peningkatan stres secara nyata sejalan dengan meningkatnya jumlah mahasiswa yang menunjukkan gejala gastritis. Bukti hubungan ini tidak hanya dapat diamati melalui angka-angka distribusi, tetapi juga diperkuat secara visual dan analitis dalam penelitian. Hal ini menegaskan bahwa stres akademik berperan penting sebagai faktor yang dapat memicu munculnya gangguan kesehatan pencernaan

seperti gastritis di kalangan mahasiswa. Visualisasi lebih lanjut dari analisis ini dapat dilihat pada Gambar 4.5.



Gambar 4.5 Risiko *Gastritis* per Tingkat Stres Akademik

Berdasarkan Gambar 4.5, terlihat bahwa terdapat hubungan yang signifikan antara tingkat stres akademik dengan jumlah mahasiswa yang mengalami risiko *gastritis*. Faktor Faktor yang paling berpengaruh terhadap stress akademik pada mahasiswa dapat dilihat pada Gambar 4.6.



Gambar 4.6 Faktor Stres Akademik

Berdasarkan Gambar 4.6, dapat diketahui bahwa terdapat lima faktor utama yang paling berkontribusi terhadap stres akademik mahasiswa. Faktor dengan skor rata-rata tertinggi adalah deadline tugas dengan skor 3,55, menunjukkan bahwa tekanan waktu dalam penyelesaian tugas merupakan penyebab stres yang paling dominan. Diikuti oleh faktor ujian dengan skor 3,49 dan beban tugas dengan skor 3,44, yang juga menunjukkan tingginya tekanan akademik yang dirasakan mahasiswa dalam hal evaluasi dan jumlah pekerjaan yang harus diselesaikan. Sementara itu, presentasi dan nilai akademik menempati urutan keempat dan kelima dengan skor masing-masing 3,23 dan 3,19. Temuan ini memperlihatkan bahwa beban akademik yang berhubungan langsung dengan penilaian dan tenggat waktu memiliki peran signifikan dalam memicu stres pada mahasiswa.

4.3 IMPLEMENTASI *DASHBOARD*

4.3.1 Menu Navigasi

Pada tampilan awal *dashboard*, ditampilkan menu navigasi yang terletak di sisi kiri antarmuka. Menu ini berfungsi sebagai kontrol utama yang memungkinkan

pengguna untuk memilih dan mengakses halaman yang diinginkan. Tampilan menu navigasi tersebut dilihat pada Gambar 4.7.

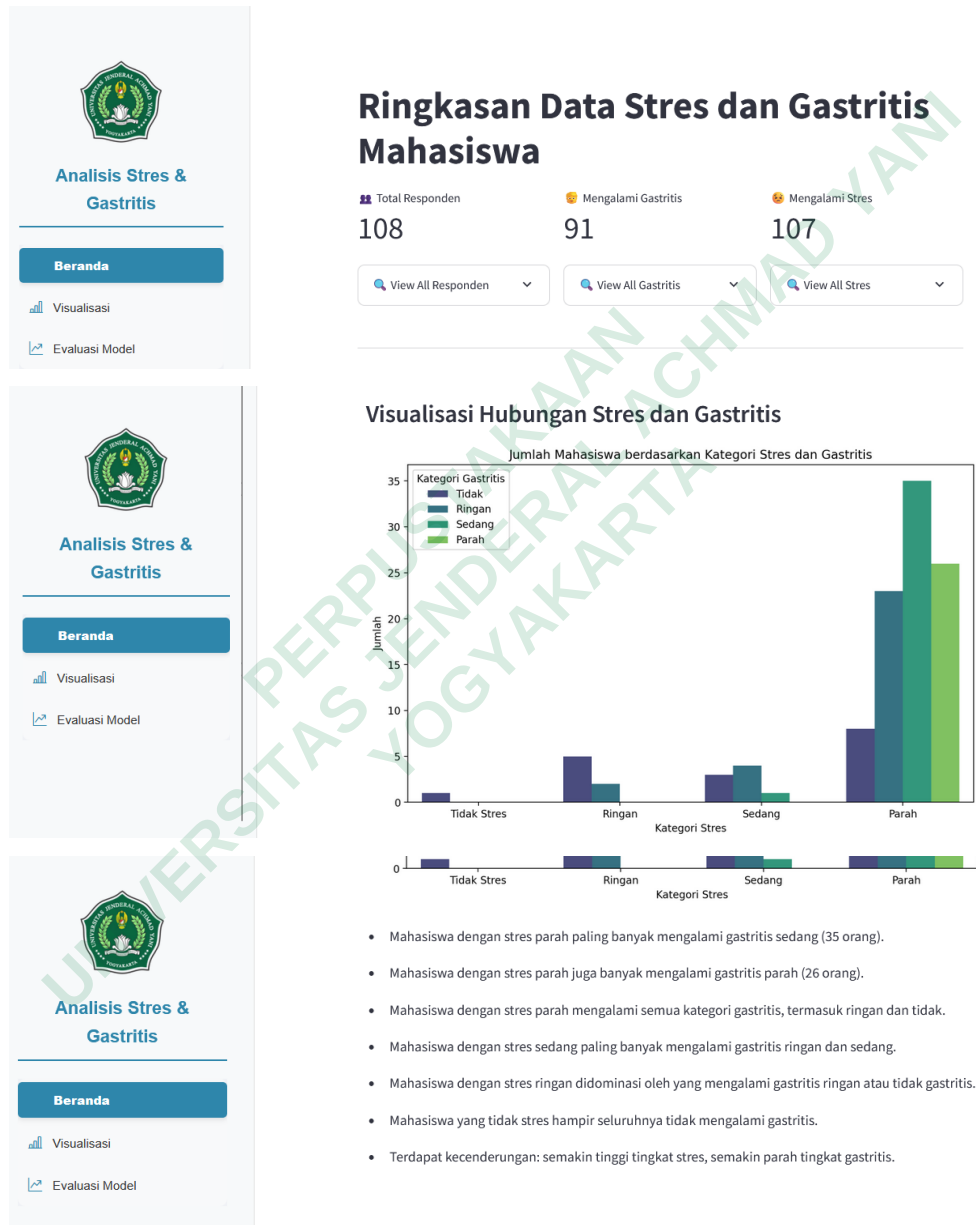


Gambar 4.7 Tampilan Menu Navigasi

Pada Gambar 4.7 menampilkan tampilan menu navigasi yang terletak di sisi kiri antarmuka *dashboard*. Menu navigasi ini dirancang untuk memudahkan pengguna dalam mengakses halaman yang tersedia pada sistem. Terdapat tiga bagian utama dalam menu ini, yaitu Beranda, Visualisasi, dan Evaluasi Model. Halaman beranda berfungsi sebagai halaman utama yang menyajikan ringkasan data. Halaman visualisasi menyajikan berbagai grafik dan diagram yang menggambarkan pola hubungan antara tingkat stres akademik dan risiko *gastritis* pada mahasiswa. Sementara itu, halaman Evaluasi Model menyediakan informasi mengenai performa algoritma *NAÏVE BAYES CLASSIFICATION* yang digunakan dalam penelitian ini, seperti nilai akurasi, presisi, recall, dan F1-score.

4.3.2 Halaman Beranda

Halaman Beranda merupakan tampilan pertama yang muncul saat pengguna mengakses *dashboard*. Halaman ini menyajikan ringkasan data yang digunakan dalam penelitian sebagai informasi awal. Tampilan Halaman Beranda dapat dilihat pada Gambar 4.8.

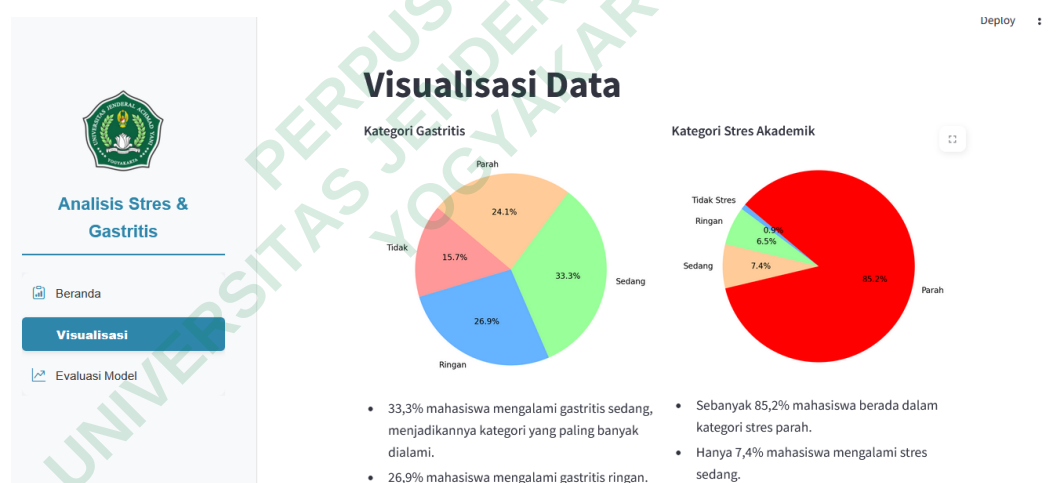


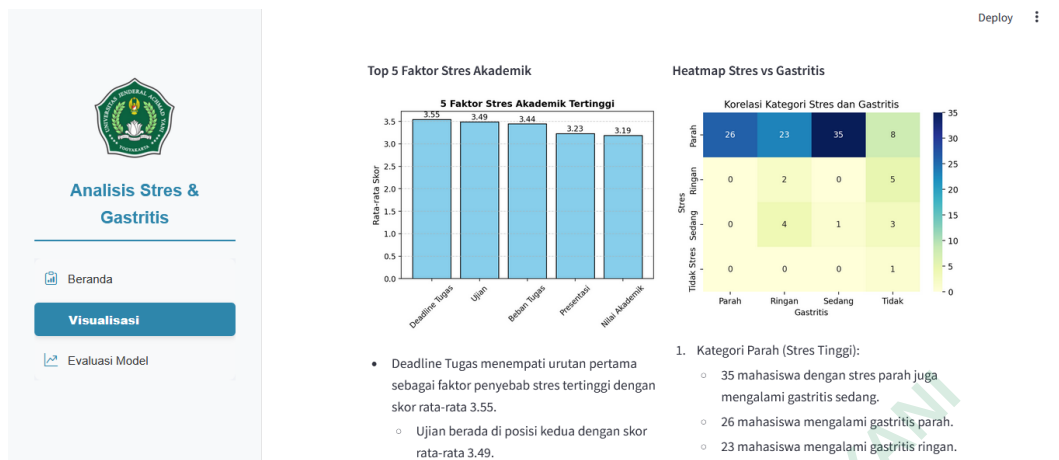
Gambar 4.8 Tampilan Halaman Beranda

Pada Gambar 4.8 menampilkan halaman beranda yang berisi informasi jumlah responden sebanyak 108 orang, dengan 91 di antaranya mengalami *gastritis* dan 107 mengalami stres akademik. Penyajian ringkasan data ini bertujuan untuk memberikan gambaran awal mengenai karakteristik data yang akan dianalisis lebih lanjut. Di bawah ringkasan tersebut, terdapat grafik visualisasi yang menggambarkan hubungan antara kategori tingkat stres akademik (Tidak Stres, Ringan, Sedang, dan Parah) dengan tingkat keparahan *gastritis* (Tidak, Ringan, Sedang, dan Parah). Selain grafik, ditampilkan pula penjelasan berupa wawasan atau insight yang diperoleh dari visualisasi tersebut guna membantu pengguna memahami pola keterkaitan antara stres akademik dan risiko *gastritis*.

4.3.3 Halaman Visualisasi

Halaman Visualisasi menampilkan grafik untuk menunjukkan hubungan antara tingkat stres akademik dan keparahan *gastritis* pada mahasiswa. Tampilan Halaman Visualisasi dapat dilihat pada Gambar 4.9.





Gambar 4.9 Tampilan Halaman Visualisasi

Pada Gambar 4.9 ditampilkan visualisasi yang menggambarkan distribusi tingkat keparahan *gastritis* dan tingkat stres akademik pada mahasiswa. Diagram lingkaran kategori *gastritis* menunjukkan bahwa sebagian besar mahasiswa mengalami *gastritis*, dengan kategori sedang sebagai yang paling dominan (33,3%), diikuti oleh ringan (26,9%), parah (24,1%), dan hanya 15,7% yang tidak mengalami *gastritis*. Sementara itu, diagram kategori stres akademik memperlihatkan bahwa mayoritas mahasiswa berada dalam kategori stres parah (85,2%), sedangkan sisanya terbagi dalam kategori stres sedang (7,4%), stres ringan (6,5%), dan tidak stres (0,9%). Grafik batang memperlihatkan lima faktor utama penyebab stres, dengan ‘Deadline Tugas’ sebagai penyebab tertinggi. *Heatmap* menunjukkan bahwa mahasiswa dengan stres parah paling banyak mengalami *gastritis* sedang dan parah, sedangkan mereka yang tidak mengalami stres cenderung tidak mengalami *gastritis*. Visualisasi ini menguatkan bahwa semakin tinggi tingkat stres akademik, semakin tinggi pula risiko *gastritis* yang dialami mahasiswa.

4.3.4 Halaman Evaluasi Model

Halaman Evaluasi Model menampilkan nilai *accuracy*, *precision*, *recall*, *f1-score*, dan menampilkan *confusion matrix* dengan keterangan TN, TP, FN, FP. Tampilan Halaman Evaluasi Model dapat dilihat pada Gambar 4.10.

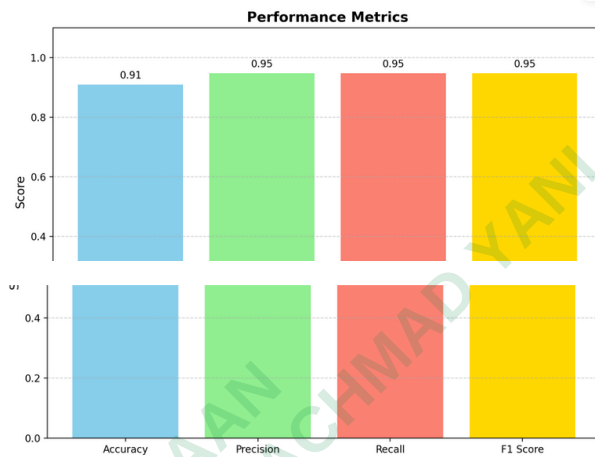


Evaluasi Model Naive Bayes

Jumlah data training: 86

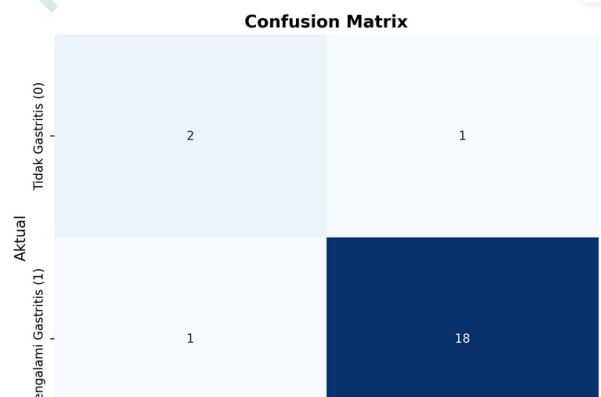
Jumlah data testing: 22

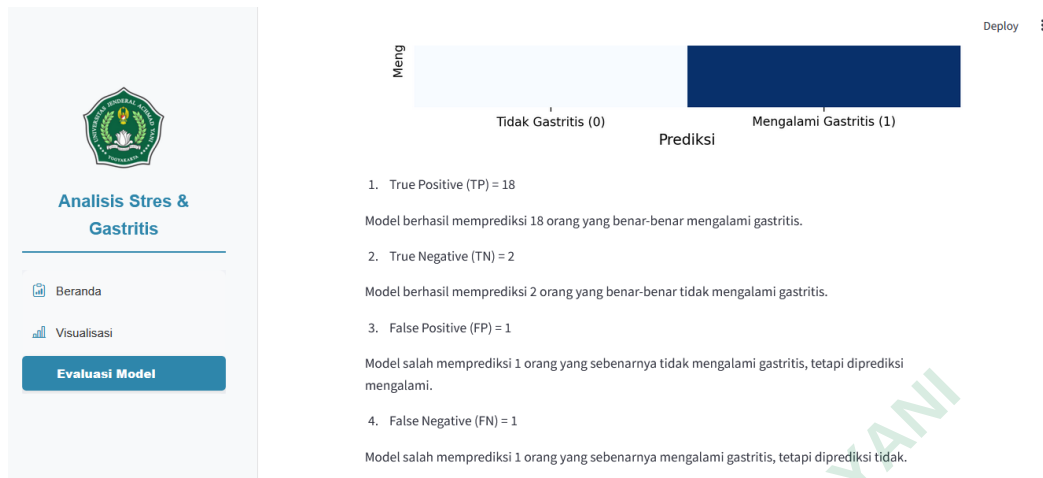
Evaluation Metrics



1. Accuracy (Akurasi) = 0.91 Artinya, 91% prediksi model sudah tepat. Dari seluruh data yang diuji, hanya 9% yang salah klasifikasi. Ini menunjukkan model cukup andal secara keseluruhan.
2. Precision (Presisi) = 0.95 Artinya, dari semua prediksi positif (misalnya: mahasiswa terindikasi gastritis), sebanyak 95% memang benar-benar positif. Ini penting untuk menghindari false positive, yaitu salah mengklasifikasikan mahasiswa sehat sebagai berisiko.
3. Recall = 0.95 Artinya, model mampu menemukan 95% kasus yang benar-benar positif. Semakin tinggi recall, semakin sedikit kasus gastritis yang terlewat oleh model. Ini penting untuk deteksi dini.
4. F1 Score = 0.95 Ini adalah gabungan dari precision dan recall. Nilai 0.95 menunjukkan model sangat seimbang antara ketepatan dan kelengkapan

Confusion Matrix





Gambar 4.10 Tampilan Halaman Evaluasi Model

Pada Gambar 4.10 menampilkan hasil dari nilai *accuracy*, *precision*, *recall*, *f1-score* dan *confussion matrix*. Model menunjukkan akurasi sebesar 0,91, dengan *precision*, *recall*, dan *f1-score* masing-masing sebesar 0,95, menandakan performa klasifikasi yang sangat baik. *Confusion matrix* juga memperlihatkan prediksi yang akurat pada tiap kelas. Hasil ini menunjukkan bahwa model efektif dalam memprediksi risiko *gastritis* berdasarkan tingkat stres akademik mahasiswa.