



DAGO ACADEMY

27 - 05 - 2024

# ML & DL BOT TELEGRAM

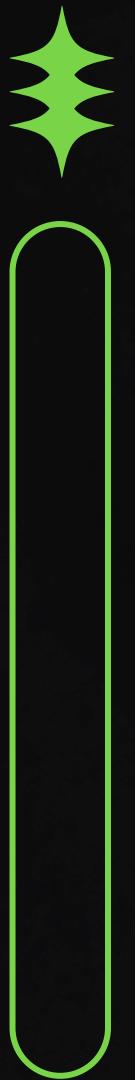
Pengembangan Machine Learning dan Deep Learning  
“DIABETES PREDICTION” dan integrasi pada bot  
telegram

# Agenda

---

Introduction (Pendahuluan)	3
User Guidance (Cara Pemakaian)	7
Algorithm (Penjelasan Algoritma)	10
Code Design (Desain Code)	20
Percobaan Prediksi Bot Telegram	24

# INTRODUCTION



# PENJELASAN IDE

---

- Pasien pengidap diabetes di Indonesia tinggi, menjadi posisi 5 tertinggi di dunia
- Dataset sudah banyak didapatkan sehingga mudah untuk digunakan dalam training model
- Teknologi training data sudah maju sehingga dapat prediksi penyakit diabetes dengan mudah
- Dapat membantu dalam pencegahan sejak dini terhadap penyakit diabetes
- Fitur yang ada dalam prediksi penyakit diabetes dapat dimengerti dengan baik



# TUJUAN

- Memngembangkan Diabetes Prediction yang terintegrasi dalam chatbot Telegram
- Mengimplementasikan Algoritma Secure Vector Machine dalam mengembangkan Diabetes Prediction



# MANFAAT

- Mempermudah pengecekan diabetes dengan deteksi dini
- Dapat lebih personalisasi pemeriksaan terhadap penyakit diabetes sesuai data pengguna
- Memberikan optimasi sumber daya kesehatan pengguna
- Meningkatkan kualitas hidup
- Membantu mencegah komplikasi

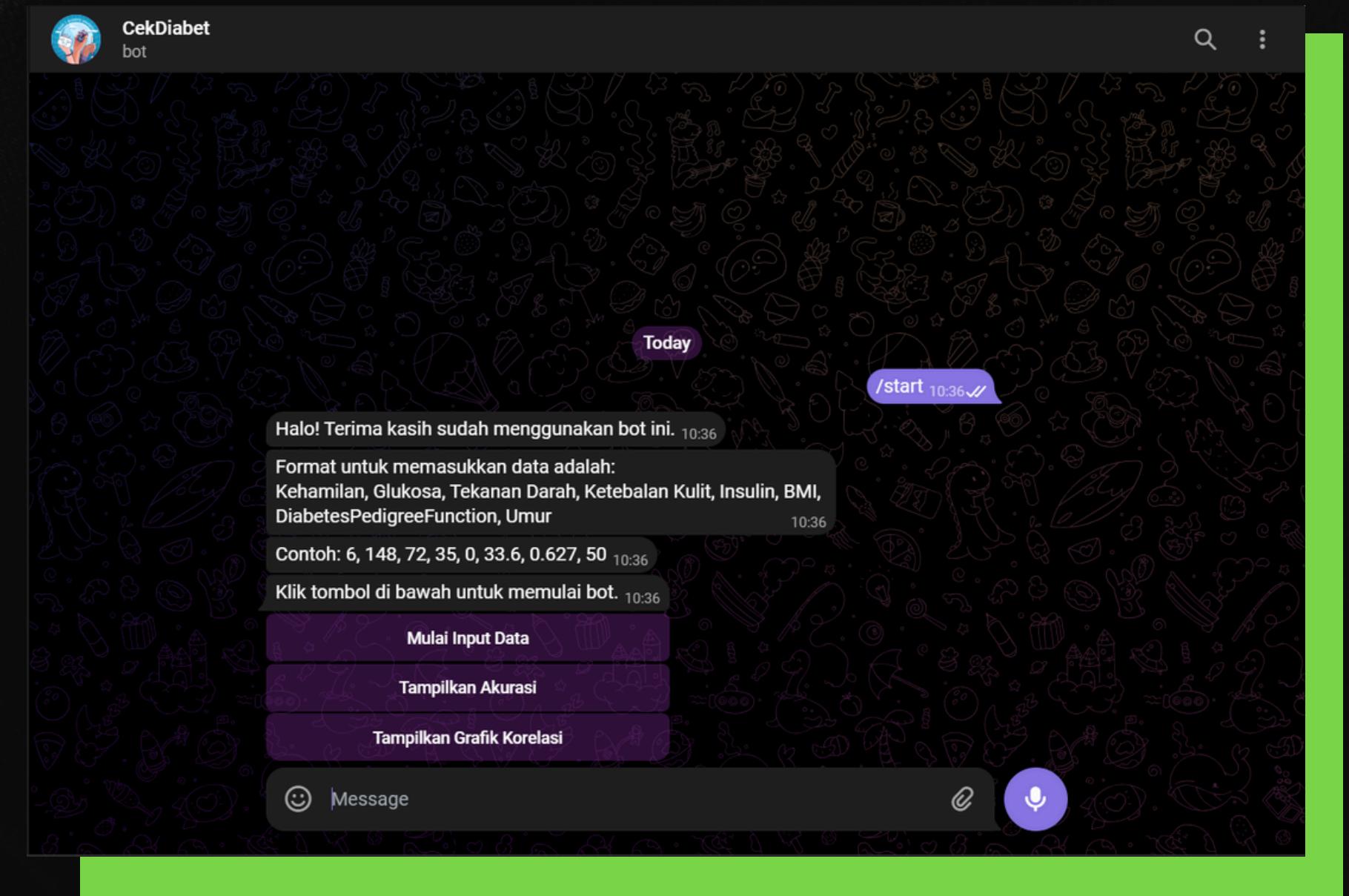


# USER GUIDANCE

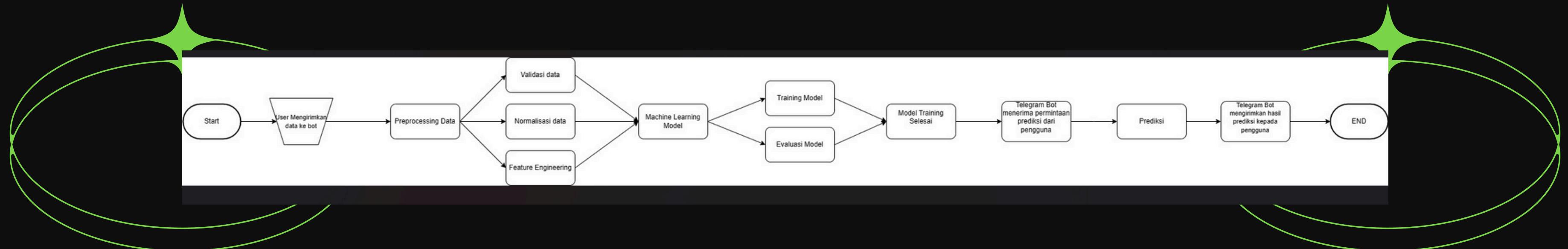
# USER GUIDANCE

## TATA CARA PEMAKAIAN

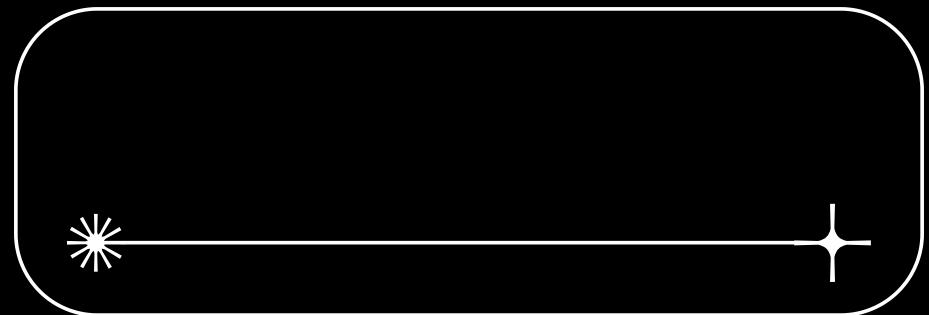
1. User diharuskan masuk chat kedalam telegram bot dengan username "CekDiabet\_Bot"
2. Pada chatbot terdapat beberapa command dan button yang dapat digunakan user:
  - /start - Command awal bot
  - /mulailnputData - Melakukan Input data user yang akan di prediksi
  - /tampilkanAkurasi - Memulai prediksi diabetes dari data user
  - /tampilkanGrafikKorelasi - Untuk melaporkan pada team, jika bot bermasalah
  - /help - Menampilkan instruksi penggunaan bot



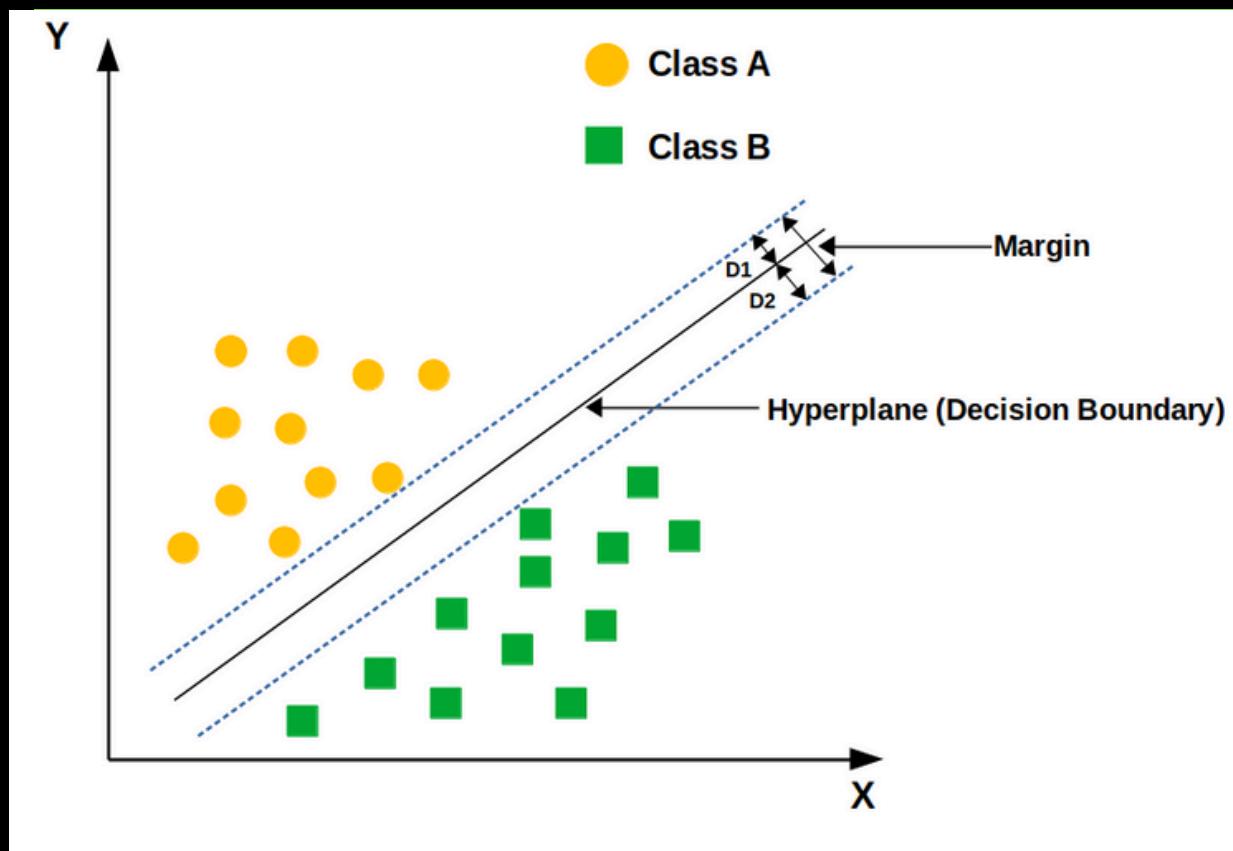
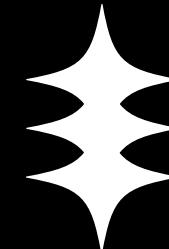
# FLOWCHART



# ALGORITHM & DATASET



# ALGORITHM



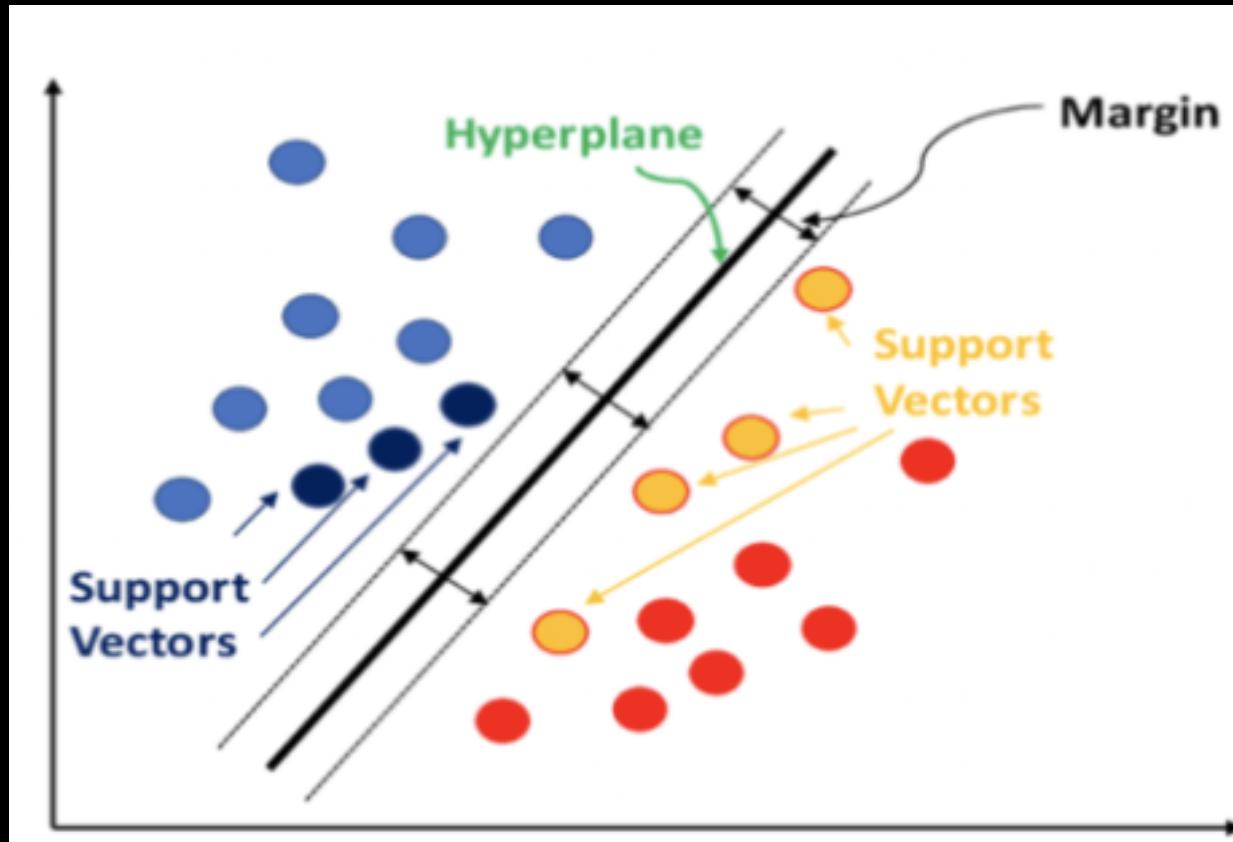
## SECURE VECTOR MACHINE

SVM (Support Vector Machine) adalah algoritma pembelajaran mesin yang digunakan untuk klasifikasi dan regresi.

Tujuan utama SVM adalah untuk menemukan hyperplane terbaik yang memisahkan dua kelas dalam ruang fitur.

Hyperplane ini dipilih sedemikian rupa sehingga jarak antara hyperplane dan titik-titik terdekat dari kedua kelas, yang disebut vektor pendukung, maksimum.

# ALGORITHM

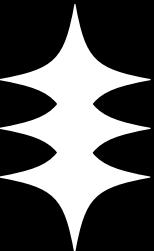


## SECURE VECTOR CLASSIFIER

SVC merupakan singkatan dari Support Vector Classifier, sebuah algoritma pembelajaran yang digunakan untuk tugas klasifikasi dalam machine learning

Algoritma ini bekerja dengan mencari hyperplane yang terbaik memisahkan kelas-kelas dalam ruang fitur.

Tujuannya adalah untuk menemukan hyperplane optimal yang memaksimalkan margin antara hyperplane dan titik data terdekat dari setiap kelas, disebut vektor pendukung.



# ✳ DATASET

## FITUR - FITUR DATASET

Kehamilan: Jumlah kehamilan yang pernah dialami oleh individu.

Glukosa: Konsentrasi glukosa dalam plasma 2 jam setelah uji toleransi glukosa oral.

Tekanan Darah: Tekanan darah diastolik (mm Hg) pada saat pemeriksaan medis.

Ketebalan Kulit: Ketebalan lipatan kulit trisep (mm), yang diukur pada daerah trisep lengan kanan menggunakan kaliper kulit.

Kehamilan,Glukosa,Tekanan Darah,Ketebalan Kulit,

,Insulin,BMI,DiabetesPedigreeFunction,Umur,Hasil

1,85,66,29,0,26.6,0.351,31,0  
8,183,64,0,0,23.3,0.672,32,1  
1,89,66,23,94,28.1,0.167,21,0  
0,137,40,35,168,43.1,2.288,33,1  
5,116,74,0,0,25.6,0.201,30,0  
3,78,50,32,88,31.0,0.248,26,1  
10,115,0,0,0,35.3,0.134,29,0  
2,197,70,45,543,30.5,0.158,53,1



# ✳ DATASET

## FITUR - FITUR DATASET

Insulin: Tingkat insulin dalam serum, diukur dalam mu U/ml.

BMI (Body Mass Index): Indeks massa tubuh, dihitung sebagai berat badan (kg) dibagi dengan tinggi badan (m) kuadrat.

Diabetes Pedigree Function: Skor fungsi pedigree diabetes yang menggambarkan seberapa tinggi kemungkinan adanya riwayat diabetes dalam keluarga penderita.

Kehamilan,Glukosa,Tekanan Darah,Ketebalan Kulit,

,Insulin,BMI,DiabetesPedigreeFunction,Umur,Hasil

1,85,66,29,0,26.6,0.351,31,0  
8,183,64,0,0,23.3,0.672,32,1  
1,89,66,23,94,28.1,0.167,21,0  
0,137,40,35,168,43.1,2.288,33,1  
5,116,74,0,0,25.6,0.201,30,0  
3,78,50,32,88,31.0,0.248,26,1  
10,115,0,0,0,35.3,0.134,29,0  
2,197,70,45,543,30.5,0.158,53,1



# ✳ DATASET

## FITUR - FITUR DATASET

Umur (Age): Umur pasien.

Hasil (Outcome): Variabel target yang menunjukkan apakah pasien menderita diabetes atau tidak. 1 menunjukkan diabetes, sedangkan 0 menunjukkan tidak diabetes.

Kehamilan,Glukosa,Tekanan Darah,Ketebalan Kulit,

,Insulin,BMI,DiabetesPedigreeFunction,Umur,Hasil

1,85,66,29,0,26.6,0.351,31,0  
8,183,64,0,0,23.3,0.672,32,1  
1,89,66,23,94,28.1,0.167,21,0  
0,137,40,35,168,43.1,2.288,33,1  
5,116,74,0,0,25.6,0.201,30,0  
3,78,50,32,88,31.0,0.248,26,1  
10,115,0,0,0,35.3,0.134,29,0  
2,197,70,45,543,30.5,0.158,53,1



# DATA PREPROCESSING AND SPLITTING

```
# Muat dataset dan abaikan kolom indeks jika ada
df = pd.read_csv('diabetes.csv', index_col=0) # Ganti dengan path dataset Anda

# Pisahkan fitur dan Label
x = df.drop(columns=['Hasil'])
y = df['Hasil']

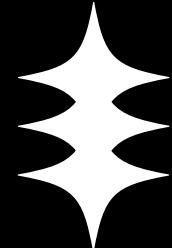
# Normalisasi fitur
scaler = StandardScaler()
scaler.fit(x)
x = scaler.transform(x)

# Pecah dataset menjadi data latih dan data uji
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)
```

## DATA PROCESSING

Pemuatan dan Persiapan Data: Memuat kumpulan data, memisahkan fitur (x) dan label (y), normalisasi fitur menggunakan StandardScaler, dan membagi data menjadi kumpulan pelatihan dan pengujian menggunakan train\_test\_split

# MODEL IMPLEMENTATION



```
# Latih model dengan SVM
clf = SVC(kernel='linear', probability=True)
clf.fit(x_train, y_train)
```

## TRAINING ALGORITMA SVM

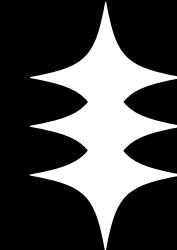
Menginisialisasi model SVM (SVC) dengan kernel linier dan melatihnya menggunakan data pelatihan (clf.fit(x\_train, y\_train)).

```
# Prediksi dan hitung akurasi
y_pred = clf.predict(x_test)
CLF_acc = accuracy_score(y_pred, y_test)
classification_report_dict = classification_report(y_test, y_pred, output_dict=True)
```

## PREDIKSI

Predict: Untuk melakukan prediksi pada data test dan melakukan kalkulasi pada akurasi (accuracy)

# MODEL IMPLEMENTATION



```
// Buat string format laporan klasifikasi secara manual
classification_report_str = (
    "Laporan Klasifikasi:\n\n"
    "Kelas 0 (Negatif Diabetes):\n"
    f"Precision: {classification_report_dict['0']['precision']:.2f}\n"
    f"Recall: {classification_report_dict['0']['recall']:.2f}\n"
    f"F1-Score: {classification_report_dict['0']['f1-score']:.2f}\n"
    f"Support: {classification_report_dict['0']['support']}\n\n"
    "Kelas 1 (Positif Diabetes):\n"
    f"Precision: {classification_report_dict['1']['precision']:.2f}\n"
    f"Recall: {classification_report_dict['1']['recall']:.2f}\n"
    f"F1-Score: {classification_report_dict['1']['f1-score']:.2f}\n"
    f"Support: {classification_report_dict['1']['support']}\n\n"
    "Rata-rata Makro:\n"
    f"Precision: {classification_report_dict['macro avg']['precision']:.2f}\n"
    f"Recall: {classification_report_dict['macro avg']['recall']:.2f}\n"
    f"F1-Score: {classification_report_dict['macro avg']['f1-score']:.2f}\n\n"
    "Rata-rata Tertimbang:\n"
    f"Precision: {classification_report_dict['weighted avg']['precision']:.2f}\n"
    f"Recall: {classification_report_dict['weighted avg']['recall']:.2f}\n"
    f"F1-Score: {classification_report_dict['weighted avg']['f1-score']:.2f}\n\n"
    f"Akurasi Model: {CLF_acc:.2f} (atau {CLF_acc * 100:.2f}%)"
```

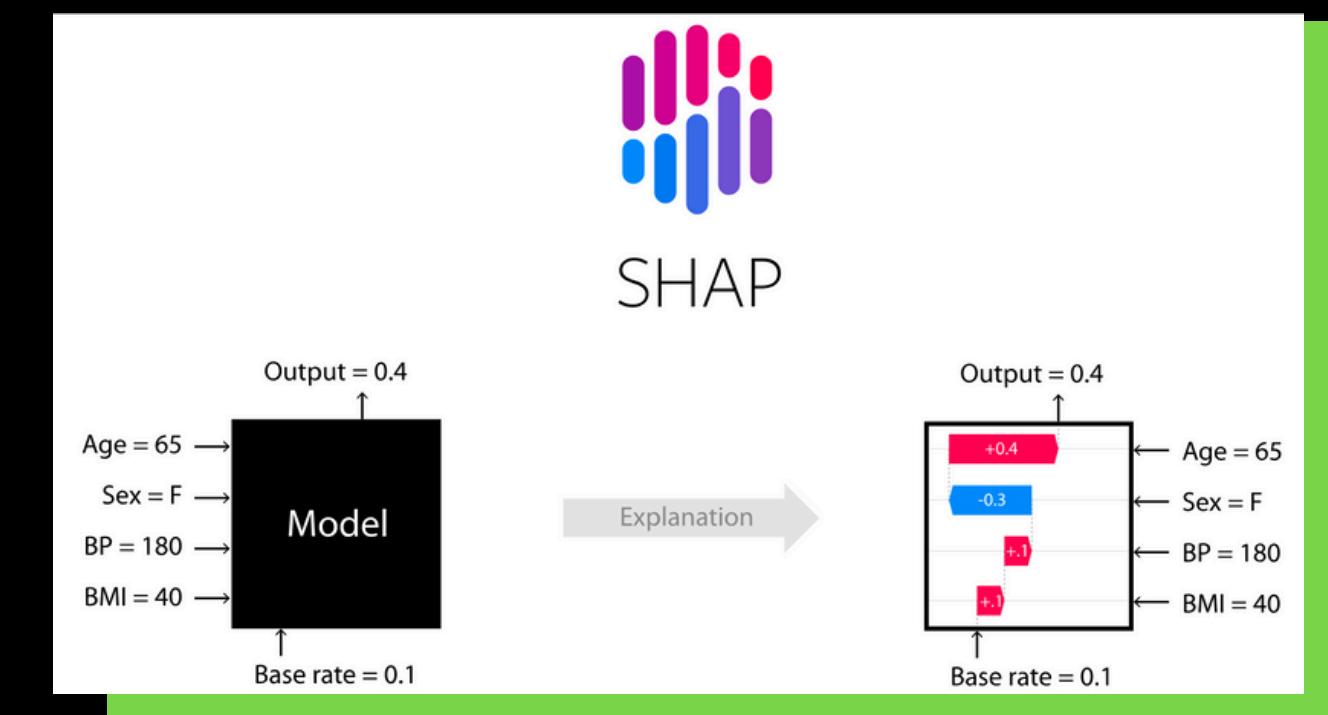
## CLASSIFICATION REPORT

Digunakan untuk dan menghasilkan laporan klasifikasi dari model dengan terperinci (classification\_report).

# SHAP

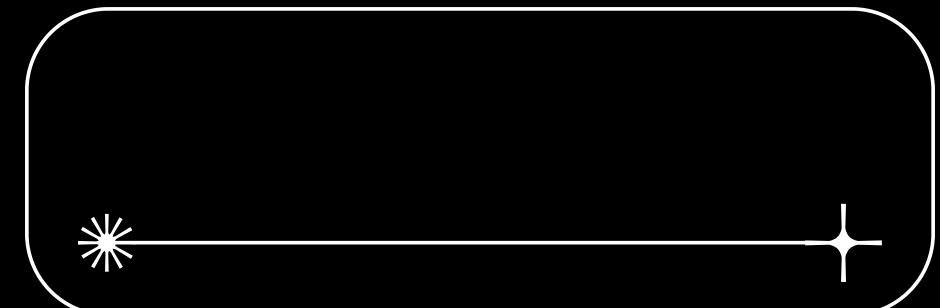
## SHAP (SHAPLEY ADDITIVE EXPLANATIONS) ANALYSIS

Menginisialisasi penjelasan SHAP (`shap.KernelExplainer`) untuk model SVM yang dilatih dan menghitung nilai SHAP untuk set pengujian. Nilai SHAP membantu menjelaskan keluaran model dengan menghitung kontribusi setiap fitur terhadap prediksi.



```
# Inisialisasi SHAP
explainer = shap.KernelExplainer(clf.predict, x_train)
shap_values = explainer.shap_values(x_test)
```

# CODE DESIGN



# CODE DESIGN

## BOT TELEGRAM

Pengimplementasian Code pada telegram bot, menggunakan skema yang sama dengan chatbot sebelumnya dengan beberapa penambahan pada UI User dan command untuk melakukan prediksi Diabetes

```
# Main Logic
if __name__ == '__main__':
    app = Application.builder().token(TOKEN).build()

# Command Handlers
app.add_handler(CommandHandler('start', start_command))
app.add_handler(CommandHandler('help', help_command))
app.add_handler(CommandHandler('custom', custom_command))
app.add_handler(CommandHandler('infoDiabetes', infoDiabetes_command))
app.add_handler(CommandHandler('mulaiPrediksi', mulaiPrediksi_command))
```

```
async def infoDiabetes_command(update: Update, context):
    keyboard = InlineKeyboardMarkup([
        [InlineKeyboardButton("Pengertian", callback_data='pengertian')],
        [InlineKeyboardButton("Pencegahan", callback_data='pencegahan')],
        [InlineKeyboardButton("Penanganan", callback_data='penanganan')]
    ])
    await update.message.reply_text("Berikut adalah informasi Diabetes yang kami sediakan ", reply_markup=keyboard)
```

# CODE DESIGN

## DATASET

Untuk code dataset yang digunakan seperti berikut dengan beberapa fitur seperti Kehamilan, Glukosa, Ketebalan Kulit, Insulin, BMi, DiabetespredgreeFunction Umur, Hasil

Import data

```
df = pd.read_csv('/content/drive/MyDrive/Klasifikasi/data/Diabetes.csv', index_col=0)
df
```

	Kehamilan	Glukosa	Tekanan Darah	Ketebalan Kulit	Insulin	BMI	DiabetesPedigreeFunction	Umur	Hasil	
0	6	148	72	35	0	33.6		0.627	50	1
1	1	85	66	29	0	26.6		0.351	31	0
2	8	183	64	0	0	23.3		0.672	32	1
3	1	89	66	23	94	28.1		0.167	21	0
4	0	137	40	35	168	43.1		2.288	33	1
...	...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9		0.171	63	0
764	2	122	70	27	0	36.8		0.340	27	0
765	5	121	72	23	112	26.2		0.245	30	0
766	1	126	60	0	0	30.1		0.349	47	1
767	1	93	70	31	0	30.4		0.315	23	0

768 rows × 9 columns

# CODE DESIGN

## ALGORITMA SVM

Ketika menggunakan algortima svm dari dataset yang sudah di test mendapatkan akurasi sekitar 70% ini berakti data tersebut sudah bisa di gunakan dengan baik untuk pembuatan model test nya

```
[24] x = df.drop(columns=['Hasil'])
y = df['Hasil']

print("x :", x.shape)
print("y :", y.shape)

x : (768, 8)
y : (768,)

scaler = StandardScaler()
scaler.fit(x)
x = scaler.transform(x)

[27] x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=42)

Modeling & Evaluation

[28] clf = SVC(kernel='linear')
clf.fit(x_train, y_train)

y_pred = clf.predict(x_test)
CLF_acc = accuracy_score(y_pred, y_test)

print(classification_report(y_test, y_pred))
print("Akurasi SVM : {:.2f}%".format(CLF_acc * 100))

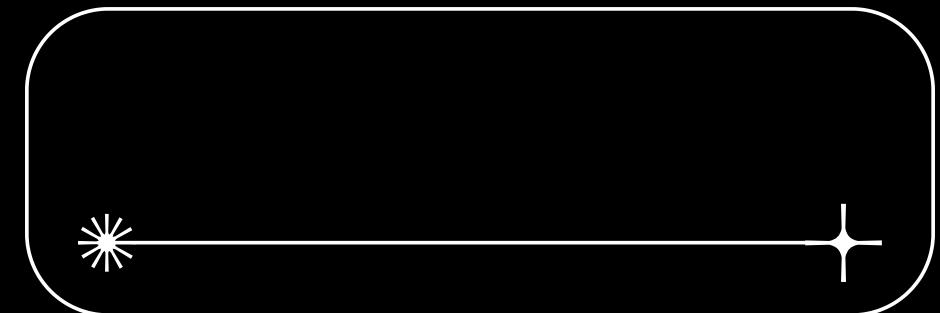
          precision    recall  f1-score   support

             0       0.81      0.81      0.81      99
             1       0.65      0.65      0.65      55

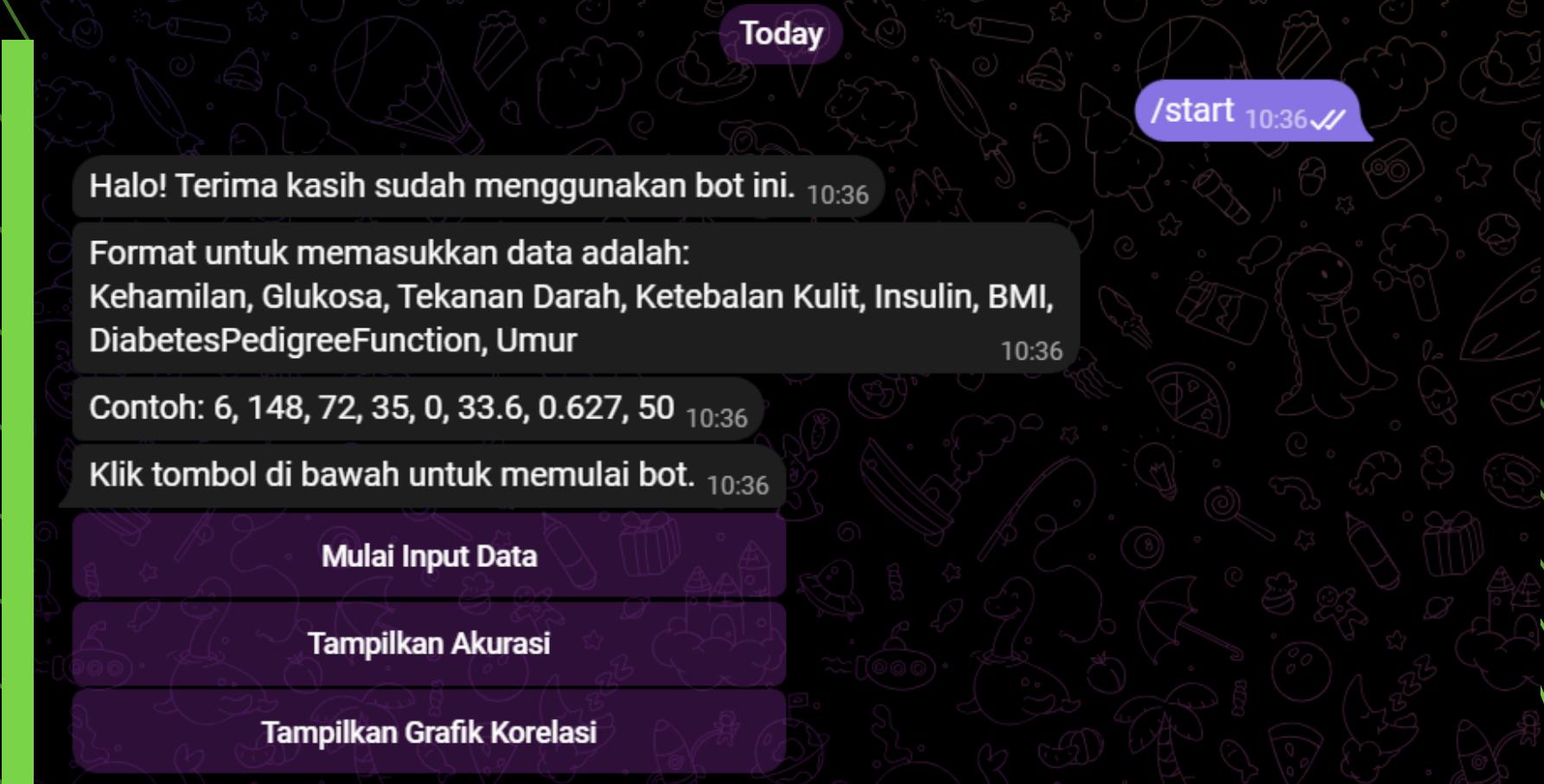
      accuracy                           0.75      154
     macro avg       0.73      0.73      0.73      154
  weighted avg       0.75      0.75      0.75      154

Akurasi SVM : 75.32%
```

# PERCOBAAN PREDIKSI BOT TELEGRAM



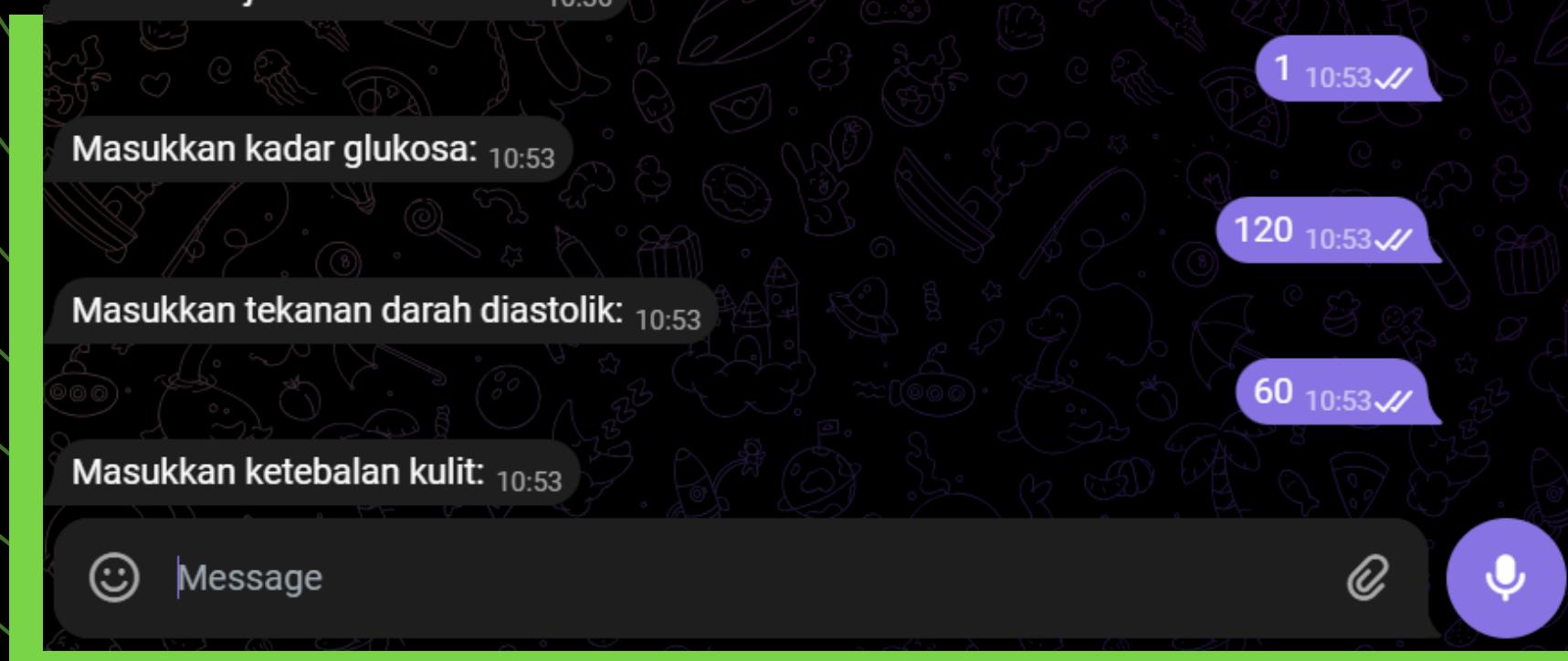
# UJI COBA BOT



## /START

- Pengguna memulai bot dengan perintah /start.
- Bot menyambut pengguna dan memberikan instruksi untuk memasukkan data pasien dalam format tertentu.
- Pengguna diarahkan untuk menekan tombol untuk memulai input data.

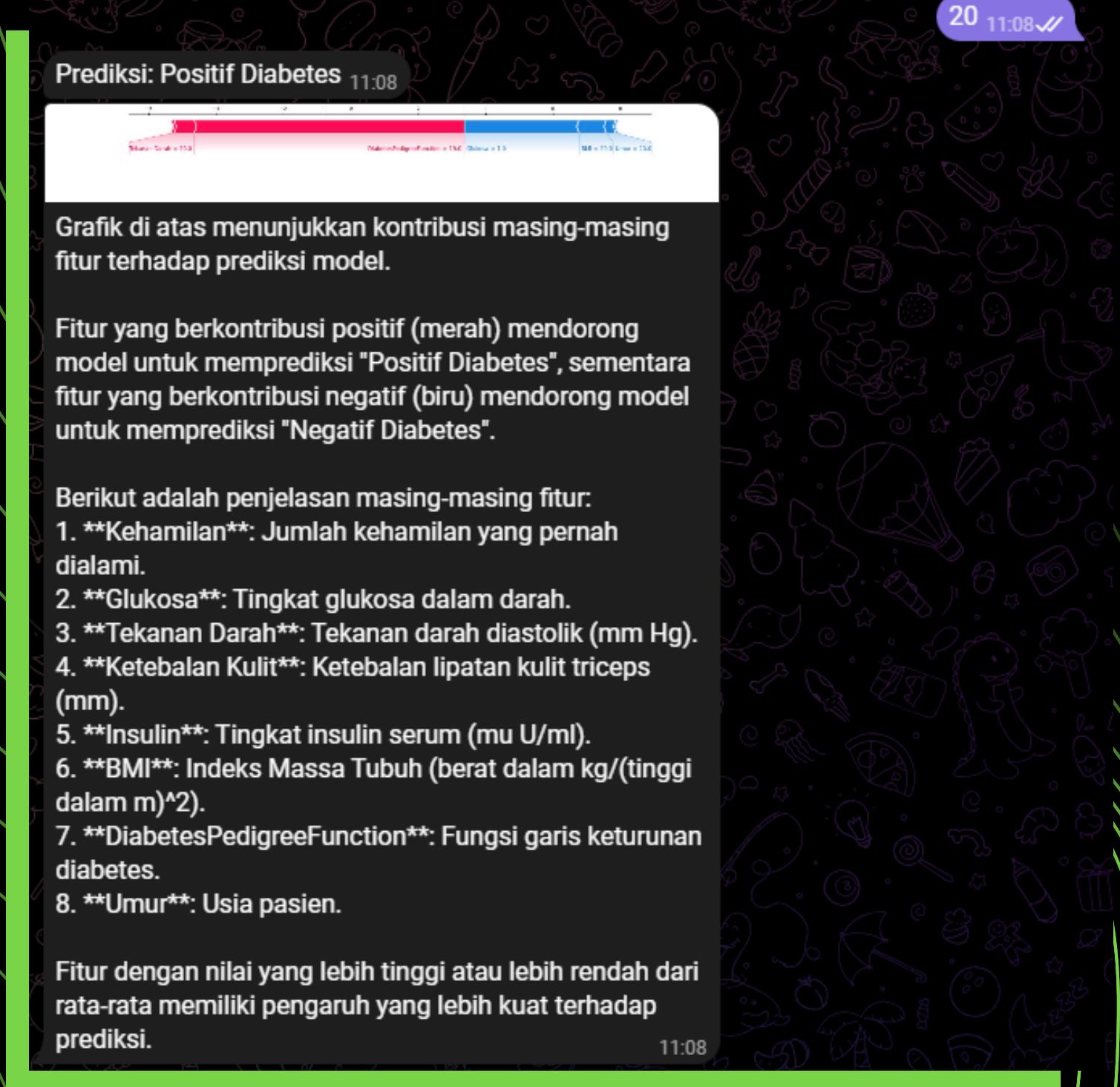
# UJI COBA BOT



## INPUT DATA PASIEN

- Pengguna akan diminta memasukkan data pasien secara bertahap.
- Bot menanyakan satu per satu nilai fitur yang diperlukan, seperti jumlah kehamilan, kadar glukosa, tekanan darah, ketebalan kulit, insulin, BMI, fungsi silsilah diabetes, dan umur.
- Pengguna memasukkan nilai-nilai tersebut satu per satu sesuai permintaan bot.

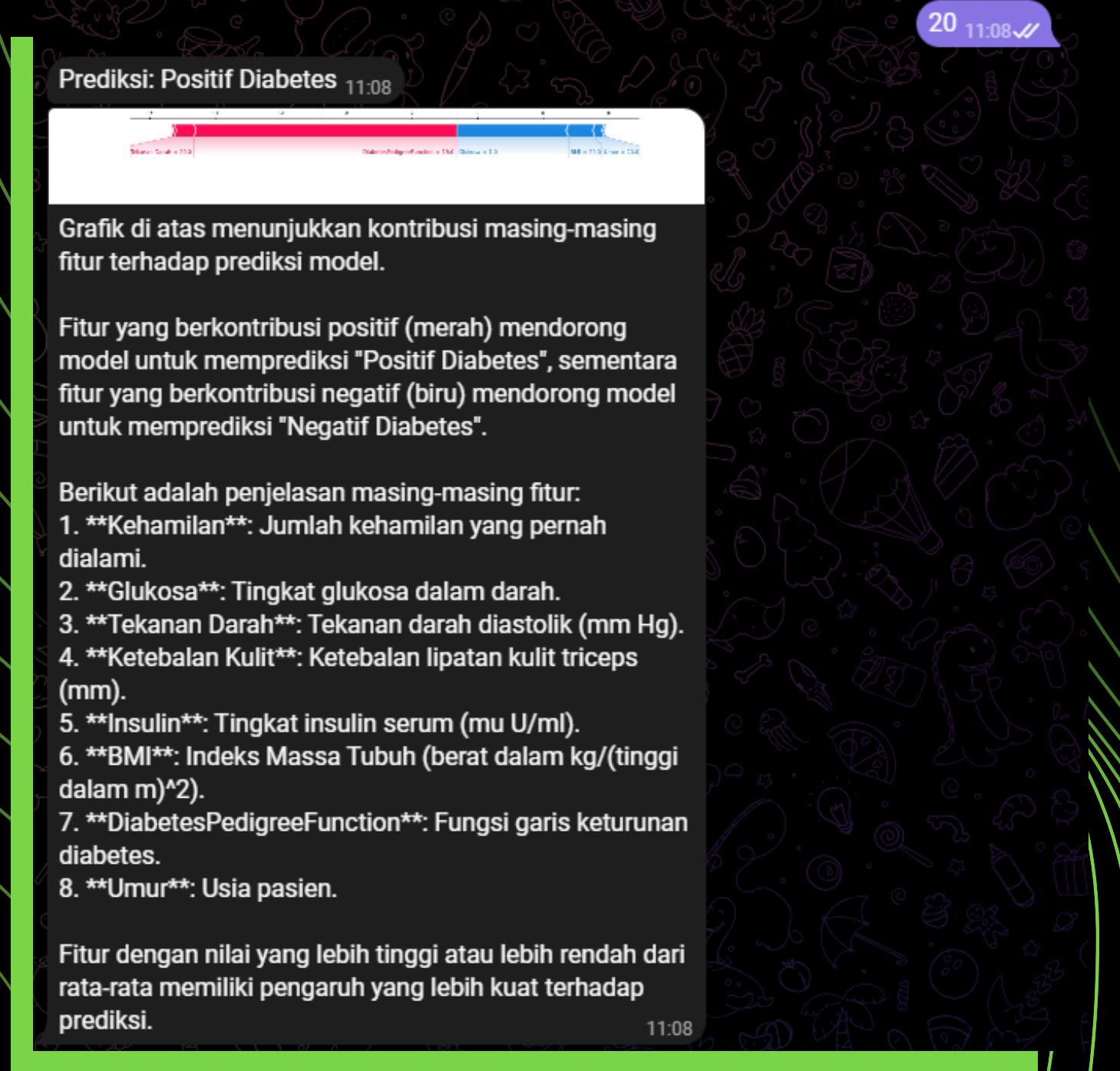
# UJI COBA BOT



## PROSES PREDIKSI

- Setelah semua data dimasukkan, bot akan memproses data tersebut.
- Data pengguna akan di-normalisasi dan digunakan untuk membuat prediksi menggunakan model SVM yang telah dilatih.
- Bot memberikan hasil prediksi kepada pengguna, menyatakan apakah pasien kemungkinan positif atau negatif diabetes.

# UJI COBA BOT



## PROSES PREDIKSI DAN PROSES SHAP

- Setelah semua data dimasukkan, bot akan memproses data tersebut.
- Data pengguna akan di-normalisasi dan digunakan untuk membuat prediksi menggunakan model SVM yang telah dilatih.
- Bot memberikan hasil prediksi kepada pengguna, menyatakan apakah pasien kemungkinan positif atau negatif diabetes.
- Bot juga dapat memberikan visualisasi dan penjelasan lebih lanjut tentang kontribusi setiap fitur terhadap prediksi menggunakan nilai SHAP.
- Grafik SHAP ditampilkan untuk memberikan wawasan lebih lanjut kepada pengguna tentang bagaimana model membuat prediksi tersebut.

# TERIMAKASIH

---