



*Symptoms, Causes,
and Treatment*

MENTAL HEALTH CHAT ANALYSIS

PROJECT OVERVIEW & ANALYSIS PROCESS



LATAR BELAKANG

Kesehatan mental menjadi isu krusial di era digital, terutama di kalangan anak muda dan pekerja. Banyak individu mengekspresikan perasaan stres, cemas, hingga depresi melalui media sosial atau platform diskusi daring. Sayangnya, jumlah percakapan yang sangat besar membuat psikolog dan konselor sulit memantau serta merespons secara cepat. Hal ini dapat menimbulkan risiko keterlambatan dalam memberikan dukungan, bahkan meningkatkan potensi tindakan berbahaya seperti self-harm.





TUJUAN

1. Mengklasifikasikan

Mengklasifikasi percakapan/chat berdasarkan kategori kesehatan mental, seperti normal, stres ringan, depresi, hingga suicidal risk. Serta mempermudah psikolog memahami teks panjang

2. Summarization

Melakukan summarization pada percakapan panjang agar konselor/psikolog dapat memahami inti masalah dengan cepat.

3. Insight

Memberikan insight mengenai pola bahasa, kata kunci, serta tren komunikasi yang dapat menjadi indikator kondisi mental seseorang.

PERMASALAHAN

- 1** Sulitnya deteksi dini tanda depresi dari ribuan percakapan
- 2** kurangnya sistem otomatis yang dapat membantu memprioritaskan kasus serius
- 3** keterbatasan waktu tenaga profesional dalam membaca teks panjang



PENDEKATAN



1. Pengumpulan Data

- Menggunakan dataset publik dari kaggle
- Melakukan scrapping data dari X dengan kata kunci depresi

2. Preprocessing Data

- Membersihkan teks, normalisasi bahasa, menghapus stopwords, dan tokenisasi.

3. Klasifikasi

- Menerapkan algoritma Machine Learning (SVM, Random Forest) dan model AI berbasis Transformer (BERT, DistilBERT) untuk mengkategorikan tingkat kesehatan mental.

4. Summarization

- Menggunakan metode extractive (TextRank) dan abstractive (LLM seperti T5/BART) untuk merangkum percakapan panjang.

5. Evaluasi

- Klasifikasi: akurasi, precision, recall, F1-score.
- Summarization: ROUGE score & evaluasi kualitas ringkasan.

RELEVANSI

Sosial impact tinggi:
membantu mendekripsi dini
potensi masalah mental.

*A smile can hide
a storm inside.*



Industri-ready: bisa
diintegrasikan ke chatbot
konseling atau platform
konsultasi online.

Akademik kuat: menggabungkan
classification dan summarization
dengan AI/LLM.

Pre Processing



Tahap ini adalah fondasi dari keseluruhan proyek, di mana data teks yang berantakan diubah menjadi format yang terstruktur dan siap diolah.

- Pembersihan Teks: Fungsi `preprocess_text` dijalankan untuk menstandardisasi data. Proses ini mencakup:
 - Mengubah semua teks menjadi huruf kecil (lowercase) untuk konsistensi.
 - Menghapus elemen yang tidak relevan seperti URL, mentions (@username), simbol, dan angka.
 - Meratakan spasi berlebih untuk kerapian.
- Penanganan Data Kosong: Setelah dibersihkan, setiap baris data yang menjadi kosong akan dihapus untuk menjaga integritas dan kualitas dataset.
- Pelabelan Numerik: Label kelas 'suicide' dan 'non-suicide' dikonversi menjadi format numerik (1 dan 0), format yang wajib agar dapat diproses oleh algoritma machine learning.

FEATURE ENGINEERING

Langkah ini bertujuan untuk mengekstrak sinyal atau pola penting dari teks yang mungkin tidak secara eksplisit ditangkap oleh model.

- Fitur Panjang Teks: Menghitung jumlah karakter dan kata pada setiap teks. Fitur ini dapat memberikan petunjuk tentang gaya penulisan atau keadaan emosional.
- Analisis Sentimen: Menggunakan VADER dari NLTK untuk mengekstrak skor sentimen (positif, negatif, gabungan). Ini secara langsung mengukur nuansa emosional dalam tulisan.
- Identifikasi Kata Kunci: Membuat fitur biner yang menandai keberadaan kata-kata kunci kritis terkait krisis mental (contoh: 'bunuh', 'mati', 'sakit', 'putus asa').
- Penghitungan Kata Ganti Orang Pertama: Menghitung frekuensi kata seperti 'aku', 'saya', dan 'diriku'. Jumlah yang tinggi bisa mengindikasikan fokus pada diri sendiri atau perenungan internal yang mendalam.

Persiapan Model BERT

Tahap ini menjembatani antara data teks olahan dengan arsitektur model Transformer.

- Pembagian Data: Dataset dibagi menjadi data latih (80%) dan data uji (20%). Proses ini menggunakan stratifikasi untuk memastikan distribusi kelas yang seimbang di kedua set, mencegah bias.
- Tokenisasi: BertTokenizer mengubah setiap kalimat menjadi serangkaian token numerik. Proses ini mencakup:
 - Menambahkan token spesial [CLS] dan [SEP].
 - Menyamakan panjang semua input menjadi 128 token melalui padding atau truncation.
 - Membuat attention mask untuk memberitahu model token mana yang harus diperhatikan.
- Pembuatan DataLoader: Data yang sudah di-tokenisasi diorganisir ke dalam DataLoader PyTorch, yang menyajikannya ke model dalam bentuk batch untuk efisiensi pelatihan pada GPU.

Pelatihan Model

Ini adalah inti dari proses pembelajaran, di mana model menyesuaikan pengetahuannya untuk tugas klasifikasi yang spesifik ini.

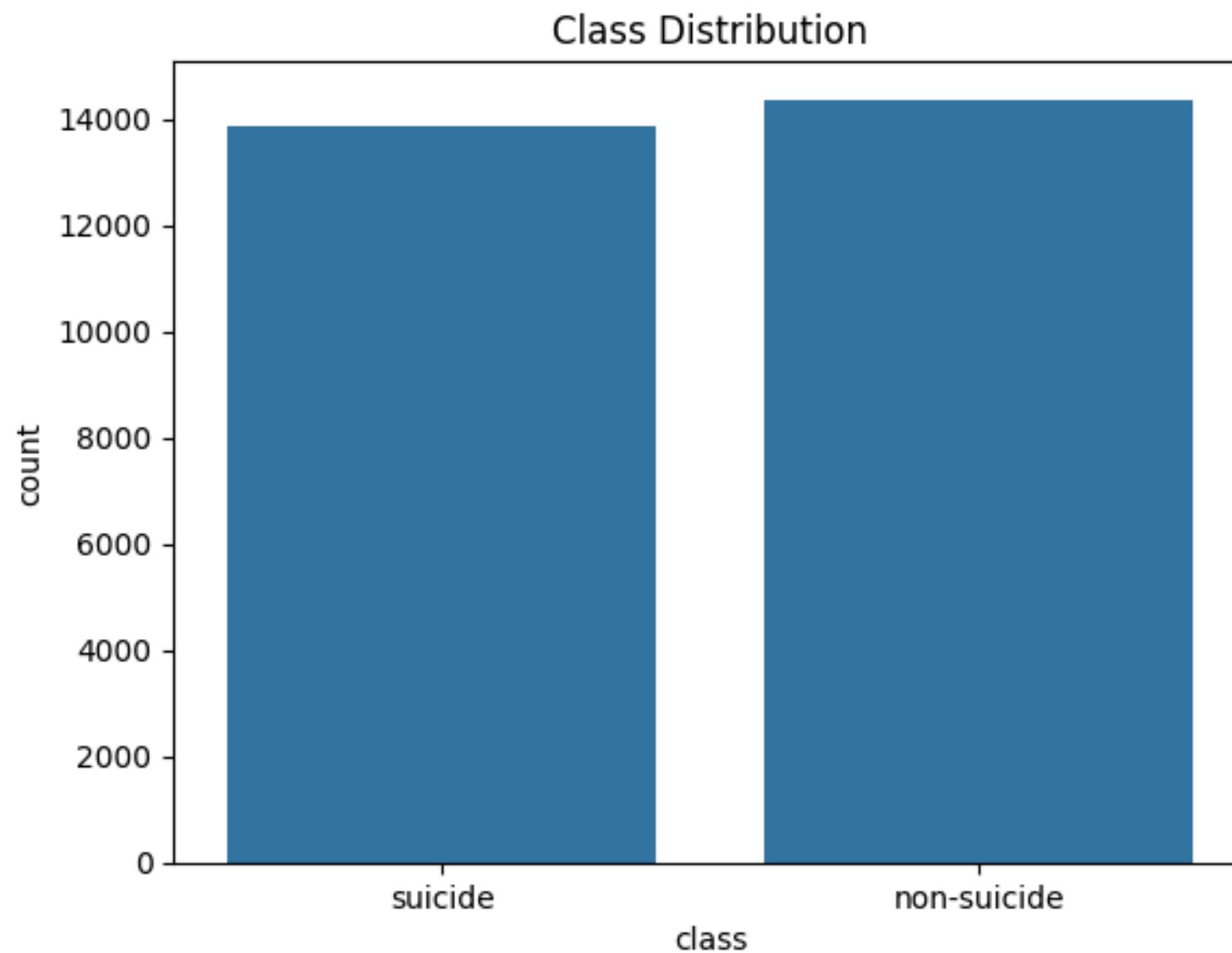
- Inisialisasi Model: Menggunakan model BertForSequenceClassification dengan bobot dari 'bert-base-uncased'. Hanya lapisan klasifikasi terakhir yang diinisialisasi ulang untuk disesuaikan dengan dua kelas (suicide/non-suicide).
- Optimisasi:
 - Menggunakan optimizer AdamW, yang sangat cocok untuk model Transformer.
 - Menerapkan learning rate scheduler dengan warmup untuk mengatur laju pembelajaran secara dinamis, yang membantu mencapai konvergensi yang lebih stabil.
- Proses Pelatihan: Model dilatih dengan data latih. Setiap batch melalui siklus:
 - a. Forward Pass: Model membuat prediksi.
 - b. Loss Calculation: Menghitung tingkat kesalahan prediksi.
 - c. Backward Pass (Backpropagation): Model belajar dari kesalahannya dan memperbarui bobotnya.

Evaluasi Model

Tahap ini memvalidasi apakah model yang telah dilatih benar-benar mampu melakukan tugasnya dengan baik.

- Metrik Kinerja Standar: Performa model diukur menggunakan metrik-metrik berikut:
 - Akurasi: Persentase total prediksi yang benar.
 - Presisi: Seberapa akurat prediksi positif yang dibuat model? (Dari yang diprediksi 'suicide', berapa yang benar?).
 - Recall: Seberapa baik model menemukan semua kasus positif? (Dari semua kasus 'suicide' asli, berapa yang terdeteksi?).
 - F1-Score: Rata-rata harmonik dari presisi dan recall, memberikan skor tunggal yang seimbang.
- Analisis Mendalam:
 - Laporan Klasifikasi: Memberikan rincian metrik untuk setiap kelas secara terpisah.
 - Confusion Matrix: Visualisasi yang menunjukkan jenis-jenis kesalahan yang dibuat model (misalnya, berapa banyak 'non-suicide' yang salah diklasifikasikan sebagai 'suicide').

RISK FACTORS



Distribusi antarkelas:

- Suicide 13854 (49.1%)
- Non-Suicide 14361 (13.854%)

Accuracy	0.9692
Precision(Weighted)	0.9692
Recall(Weighted)	0.9692
F1(Weighted)	0.9692

Model BERT-base(transformer) yang di fine tune untuk klasifikasi biner (Suicidal vs Non-Suicidal)

AI SUPPORT

&

EXPLANATION

BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

1 Embedding Kontekstual:
BERT memahami makna
kata berdasarkan konteks
kalima

2 Attention Mechanism: Memungkinkan
model fokus pada kata-kata kunci
yang relevan dengan risiko bunuh diri

3 Transfer Learning: Memanfaatkan
pengetahuan linguistik yang sudah
dipelajari dari data besar



HASIL UJI BERT

Accuracy

Proporsi prediksi benar dari seluruh data. 96.29% model konsisten pada mayoritas kasus

Precision

Ketepatan model ketika klaim dari suatu class. 96.97



Recall

Kemampuan menangkap contoh dari sebuah kelass. 96.9% model berhasil .

F1

Menilai keseimbangan

BAGAIMANA MODEL BEKERJA?

Bidirectional Encoder Representations from Transformer bekerja dengan memproses teks secara dua arah, menggunakan arsitektur transformer yang dilengkapi mekanisme self-attention, sehingga dapat memahami konteks makna kata berdasarkan kata-kata sebelum dan sesudahnya, serta mampu menangkap konteks halus(negasi, intensitas emosi, dan rujukan kata).

Proses:

1. Tokenisasi (WordPiece) → ubah teks jadi token + attention mask dengan max_length=128.
2. Fine-tuning: tambahkan classification head (linear) di atas [CLS]. Latih end-to-end (loss cross-entropy).
3. Inferensi: logit → softmax → probabilitas dua kelas; ambang default 0,5 (kelas dengan probabilitas tertinggi dipilih).
4. Fitur tambahan (sentimen VADER, indikator kata kunci, pronouns) dihitung untuk analisis forensik (menjelaskan prediksi), namun keputusan klasifikasi berasal dari BERT (bukan feature engineering manual).



CARA KERJA SISTEM

Cara Kerja Sistem Prediksi Suicide Risk

1. Prediksi dengan Model AI (BERT/Transformer)

- Teks → Tokenizer → Angka (token ID).
- Model → keluarkan probabilitas Suicidal vs Non-Suicidal.
- Kelas dengan probabilitas tertinggi → hasil prediksi.

2. Analisis Tambahan (Explainability Layer)

- Panjang teks (jumlah kata/karakter).
- Jumlah kata ganti orang pertama (I, me, my).
- Sentiment score (positif/negatif).
- Deteksi kata berbahaya (tired, end, pain, die).
- Hasil akhir → Risk Level (Low, Medium, High).

Cara Menggunakan Sistem

- Load model yang sudah disimpan (`./model_save/`).
- Input teks baru ke fungsi `suicide_risk_assessment_system()`.
- Sistem memberi hasil berupa:
 - Prediksi & Probabilitas
 - Risk Level
 - System Response
 - Analisis detail



Cara Menggunakan?

1. Training & Save Model

Setelah selesai Training model. code akan otomatis SAVE model yang akan kita gunakan untuk memprediksi kata secara langsung.

```
model.save_pretrained("./model_save/")
tokenizer.save_pretrained("./model_save/")
```

Model & tokenizer sekarang tersimpan di folder ./model_save/.

2. Load Model untuk Prediksi

```
from transformers import BertTokenizer, BertForSequenceClassification
import torch
device = torch.device("cuda" if torch.cuda.is_available() else "cp")
tokenizer = BertTokenizer.from_pretrained("./model_save/")
model
BertForSequenceClassification.from_pretrained("./model_save/").to(device)
```

Code ini digunakan untuk mengubah teks ke token agar bisa diproses oleh model. lalu kita gunakan Model yang sudah kita latih sebelumnya



Cara Menggunakan?

3. Fungsi Prediksi

```
def predict(text, model, tokenizer, device):
    inputs = tokenizer(text, return_tensors="pt", padding=True, truncation=True).to(device)
    outputs = model(**inputs)
    probs = torch.nn.functional.softmax(outputs.logits, dim=-1)
    pred_class = torch.argmax(probs, dim=1).item()
    return pred_class, probs.detach().cpu().numpy()
```

Fungsi dari code ini adalah untuk mengubah logits menjadi probabilitas. lalu probabilitas tertinggi akan digunakan untuk menentukan akan masuk ke kelas mana text tersebut.

4. Gunakan untuk Prediksi

```
labels = ["Non-Suicidal", "Suicidal"]
text = "I don't want to live anymore"
pred_class, probs = predict(text, model, tokenizer, device)

print("Text:", text)
print("Prediction:", labels[pred_class])
print("Probabilities:", probs)
```



**INSIGHT
&
FINDINGS**

INTERPRETASI

Vader sentimen, digunakan untuk analisis sentimen berbasis leksikon dan dirancang untuk teks media sosial, memberikan skor sentimen yang menunjukkan tingkat, positif, negatif, dan netral. Sentimen vader, indikator kata kunci, dan pronouns dihitung untuk analisis prediksi

Medium Risk

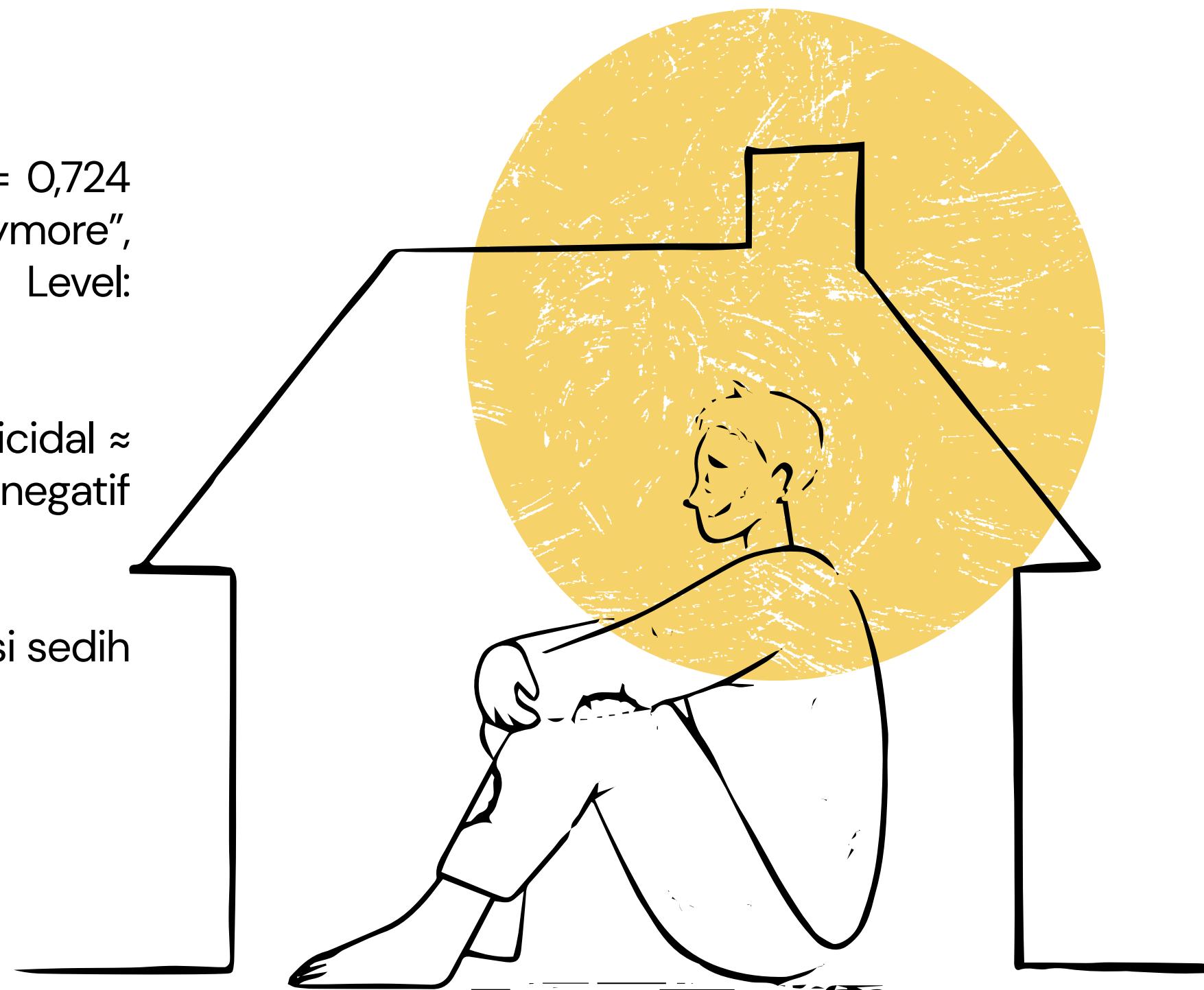
- "I don't know if I can keep going anymore..." → Prob. suicidal = 0,724 (confidence model tinggi). Bahasa keputusasaan ("anymore", "hopeless") meningkatkan sinyal risiko; sistem memberi Risk Level: Medium sesuai aturan ($>0,5$).

Medium – High Risk

- "I can't take this pain anymore... want it all to end..." → Prob. suicidal ≈ 0,78; terdapat kata kunci kuat ("pain", "end") + sentimen sangat negatif (VADER -0,75).

Low Risk

- "I've been feeling down..." → Prob. non-suicidal dominan; ekspresi sedih umum tanpa niat eksplisit; Risk Level: Low.



INSIGHT

PERFORMA MODEL

- Akurasi: 96.92% – Sangat tinggi dan konsisten
- Presisi & Recall: 96.92% – Seimbang untuk kedua kelas
- F1-Score: 96.92% – Model andal dan stabil

POLA LINGUISTIK TERIDENTIFIKASI

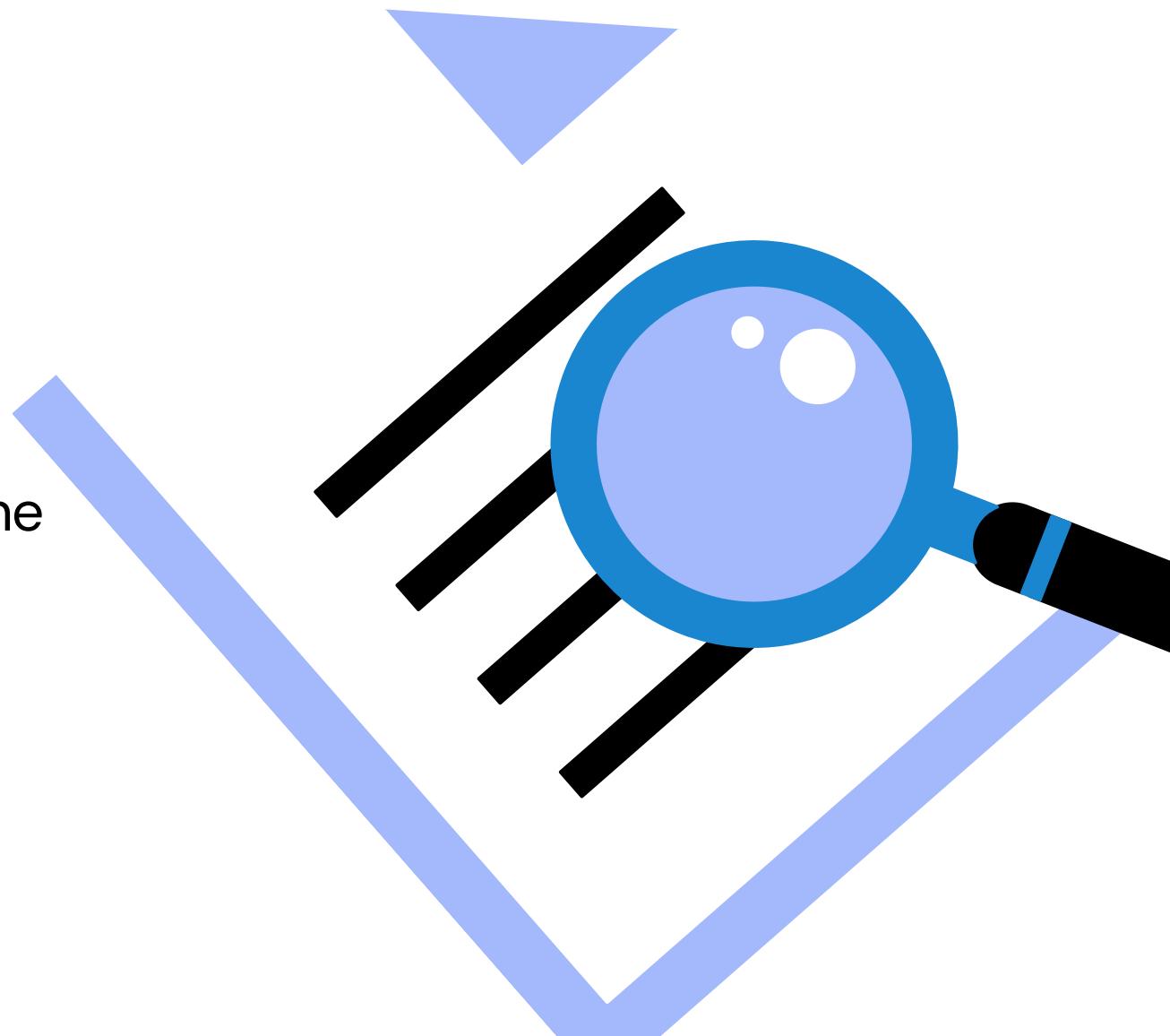
- Kata kunci prediktif: pain, want, end, hope, help
- Penggunaan kata ganti orang pertama (I, me, my) meningkat pada teks suicidal
- Teks suicidal lebih panjang dan elaboratif
- Pola emosional spesifik terdeteksi oleh model

KEMAMPUAN MODEL

- Deteksi pola halus yang mungkin terlewat manusia
- Respons instan untuk skrining awal
- Konsistensi tinggi tanpa fatigue
- Generalisasi baik pada data baru

BATASAN DAN NUANSA

- 3% error rate masih signifikan untuk konteks high-risk
- Confidence score bervariasi (53-86%) – butuh human review untuk kasus borderline
- Terdapat false negatives yang berpotensi berakibat serius
- Kesulitan dengan sarkasme dan nuansa budaya



INSIGHT

PERFORMA MODEL

- Akurasi: 96.92% – Sangat tinggi dan konsisten
- Presisi & Recall: 96.92% – Seimbang untuk kedua kelas
- F1-Score: 96.92% – Model andal dan stabil

POLA LINGUISTIK TERIDENTIFIKASI

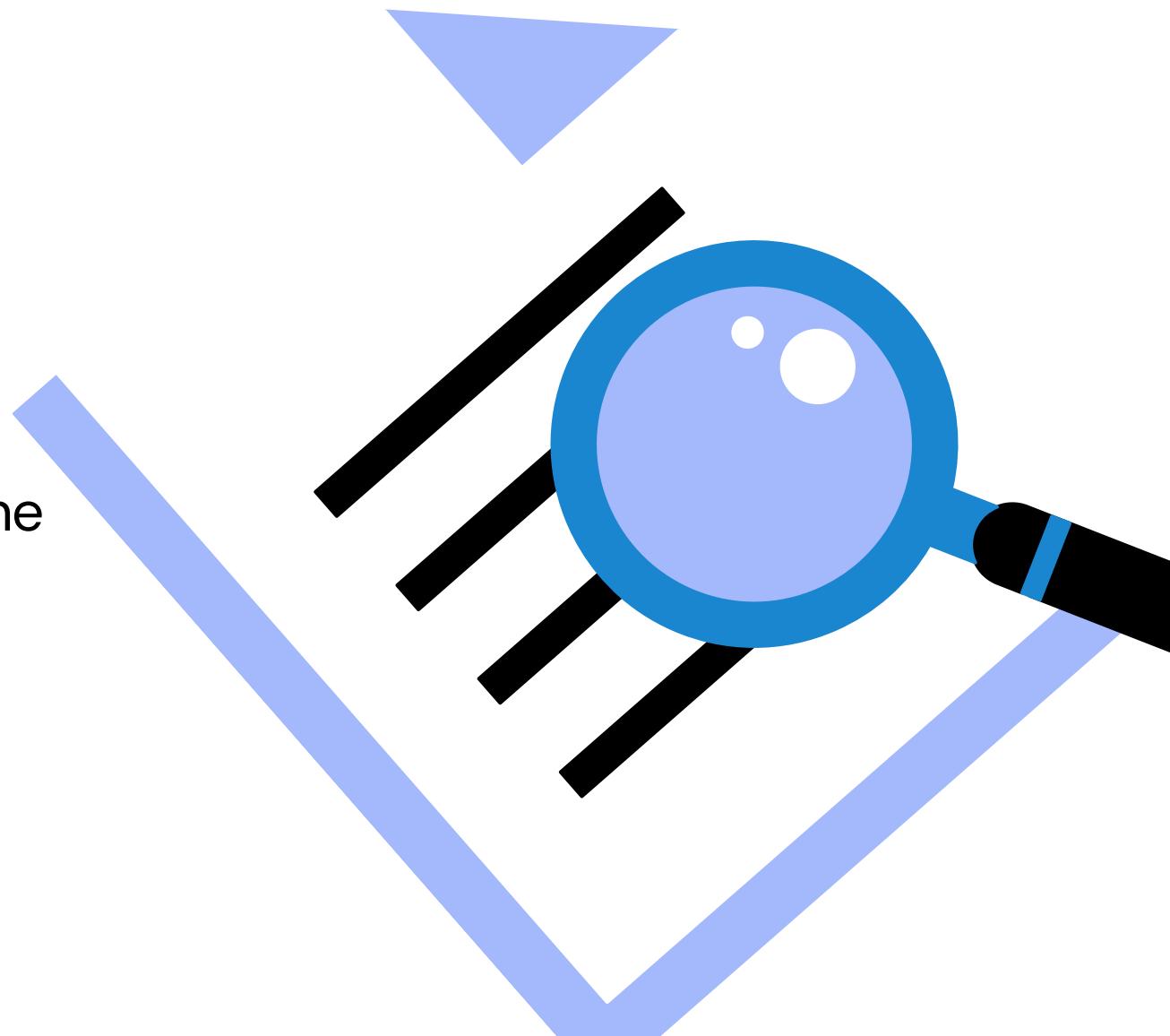
- Kata kunci prediktif: pain, want, end, hope, help
- Penggunaan kata ganti orang pertama (I, me, my) meningkat pada teks suicidal
- Teks suicidal lebih panjang dan elaboratif
- Pola emosional spesifik terdeteksi oleh model

KEMAMPUAN MODEL

- Deteksi pola halus yang mungkin terlewat manusia
- Respons instan untuk skrining awal
- Konsistensi tinggi tanpa fatigue
- Generalisasi baik pada data baru

BATASAN DAN NUANSA

- 3% error rate masih signifikan untuk konteks high-risk
- Confidence score bervariasi (53-86%) – butuh human review untuk kasus borderline
- Terdapat false negatives yang berpotensi berakibat serius
- Kesulitan dengan sarkasme dan nuansa budaya



LINK

BIT.LY/HACKTIVE CLASSIFICATION

KAGGLE.DATASET