

Implementasi *Framework AI Project Cycle* dalam Perancangan *Chatbot* Deteksi Tingkat Depresi Mahasiswa Kabupaten Tasikmalaya

Isna Nadifatul Istiqomah*¹, Ahmad Zamakhsyari Sidiq²

^{1,2}Sekolah Tinggi Teknologi Cipasung, Indonesia

isna10221015@student.sttcipasung.ac.id¹, ade@sttcipasung.ac.id²

Informasi Artikel

Kata Kunci:

Framework AI Project Cycle; *Deep Learning*; BiLSTM; Depresi.

Histori Artikel:

Disubmit 11 Juni 2025

Direvisi 11 Juni 2025

Diterima 11 Juni 2025

Tersedia daring 31 Juli 2025

Sitasi:

Isna Nadifatul Istiqomah and Ahmad Zamakhsyari Sidiq, "Implementasi *Framework AI Project Cycle* dalam Perancangan *Chatbot* Deteksi Tingkat Depresi Mahasiswa Kabupaten Tasikmalaya," in *Prosiding Seminar Nasional Teknik Universitas Suryakancana*, Cianjur: Fakultas Teknik Universitas Suryakancana, 2025.

* Penulis Korespondensi.

Isna Nadifatul Istiqomah

Alamat E-mail:

isna10221015@student.ac.id

Abstrak

Jawa Barat memiliki tingkat depresi tertinggi di Indonesia, disisi lain jumlah mahasiswa di Kabupaten Tasikmalaya cukup besar yaitu mencapai 161.360 jiwa, namun jumlah klinik psikolog yang tersedia sangat terbatas atau belum mencukupi kebutuhan layanan kesehatan mental di Tasikmalaya. Maka dari itu, *chatbot* dipilih sebagai solusi dari permasalahan tersebut, karena dapat beroperasi selama 24 jam dan tidak tergantung pada sumber daya manusia. Penelitian ini bertujuan untuk merancang *chatbot* Telegram yang bisa mendeteksi tingkat depresi mahasiswa Kabupaten Tasikmalaya berdasarkan pertanyaan *The Patient Health Questionnaire-9*. Perancangan dilakukan menggunakan *Framework Artificial Intelligence Project Cycle* untuk mempermudah dalam merencanakan pengembangan. Selain itu, model *Bidirectional Long Short-Term Memory* yang merupakan algoritma *deep learning*, menjadi pilihan utama karena kemampuannya dalam membaca konteks dari dua arah dan dapat memahami pola bahasa dengan lebih baik, sehingga algoritma ini cocok diimplementasikan pada *chatbot*. Rancangan sistem yang dihasilkan mencakup *Unified Modelling Language*, arsitektur teknologi, arsitektur Railway, *pseudocode* model, spesifikasi *Application Programming Interface*, dan *User Acceptance Criteria*.

1. Pendahuluan

Depresi merupakan kondisi klinis yang sering ditandai dengan perasaan sedih yang mendalam, kehilangan minat atau kesenangan dalam aktivitas yang biasa dinikmati, perubahan dalam berat badan atau nafsu makan, gangguan tidur, dan kelelahan yang berlebihan [1]. Dampak depresi dapat menimbulkan perilaku yang merugikan, seperti mudah menarik diri, rendah diri, dan isolasi sosial. Jika tidak ditangani dengan baik, depresi dapat berkembang menjadi pikiran-pikiran buruk, termasuk pikiran untuk bunuh diri [2].

Secara Nasional, prevalensi depresi di Indonesia pada tahun 2023 sebesar 1,4%. Prevalensi depresi paling tinggi ada pada kelompok anak muda 15-24 tahun yaitu sebesar 2%. Walaupun prevalensi depresi pada usia ini tinggi, namun kelompok ini adalah yang paling sedikit dalam mengakses pengobatan. Berdasarkan hasil survei Indonesia – *National Adolescent Mental Health Survey* (I-NAMHS), hanya sebagian kecil remaja yang mencari pengobatan ke fasilitas kesehatan seperti dokter dan perawat, yaitu sebesar 24,3% [3]. Mahasiswa, yang mayoritas berusia 18-30, juga termasuk dalam kelompok rentan ini [4].

Dikutip dari website Databoks yang ditulis oleh Muhamad (2024), Jawa Barat memiliki tingkat depresi tertinggi dibanding dengan provinsi lain yaitu sebesar 3,3%. Disisi lain, jumlah mahasiswa di Tasikmalaya cukup besar, yaitu mencapai 161.360 jiwa [6], jika dibandingkan dengan jumlah klinik psikolog di Tasikmalaya yang hanya memiliki 10 klinik [7], jumlah ini sangat terbatas atau tidak mencukupi kebutuhan layanan kesehatan mental di Tasikmalaya. Menurut Pusat Pengelolaan Data Pendidikan Tinggi (2024), keterbatasan jumlah klinik psikolog ini menyebabkan banyak mahasiswa kesulitan mendapatkan akses ke psikolog profesional, baik karena jarak, biaya, maupun ketersediaan tenaga ahli.

Berdasarkan latar belakang diatas, diperlukan solusi untuk mengatasi keterbatasan sumber daya klinik kesehatan mental tersebut, terutama dalam menangani permasalahan depresi yang cukup tinggi dikalangan mahasiswa. Untuk mengatasi keterbatasan ini, kecerdasan buatan bisa menjadi alternatif dari melakukan konsultasi ke pakar psikologi [9]. Salah satu solusi dalam mendukung kesehatan mental adalah dengan memanfaatkan penggunaan *chatbot* berbasis kecerdasan buatan yang dapat berinteraksi dengan pengguna secara *realtime*. Teknologi *chatbot* ini memiliki potensi besar untuk memberikan dukungan kesehatan mental karena memiliki kemampuan beroperasi selama 24 jam [10]. Pada penelitian ini, akan dilakukan perancangan saja, dengan tujuan untuk mempermudah dalam implementasi di tahap selanjutnya.

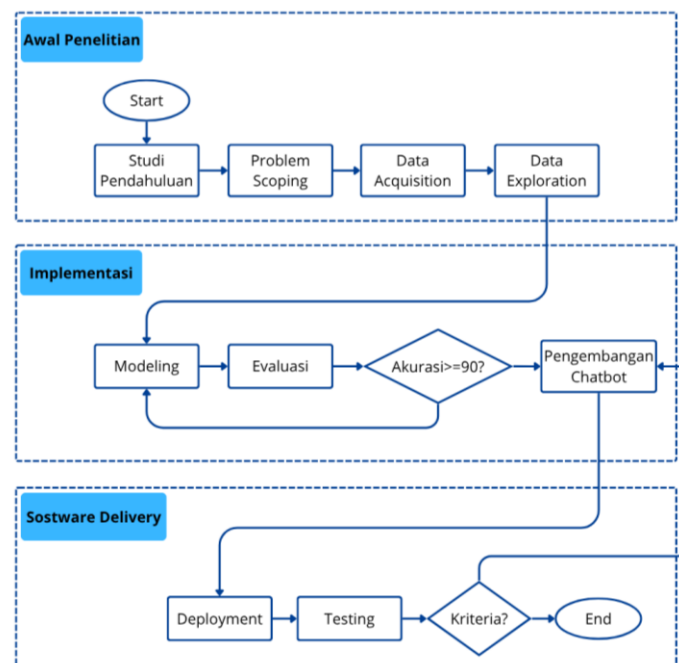
Dalam dunia medis untuk pemeriksaan klinis dalam penegakan diagnosis adalah dengan skrining awal dan diagnosis menggunakan metode wawancara dan kuisioner. Salah satu metode yang dapat digunakan dalam skrining awal depresi adalah *The Patient Health Questionnaire* (PHQ-9). *The Patient Health Questionnaire* (PHQ-9) adalah sebuah urutan pertanyaan dalam skrining awal depresi untuk melihat tingkat awal keparahan depresi. Dr. Kurt

Kroenke dkk dari Universitas Columbia mengembangkan PHQ-9 pada tahun 1999 dengan Hibah dari Pfizer. PHQ-9 merupakan alat ukur yang dapat menggambarkan tingkat keparahan depresi dalam kurun waktu 2 minggu terakhir [11].

Dalam perancangan *chatbot* untuk deteksi depresi, pemilihan model yang tepat sangat penting untuk memastikan keakuratan hasil analisis percakapan pengguna. BiLSTM (*Bidirectional Long Short-Term Memory*) yang merupakan algoritma *deep learning*, menjadi pilihan utama karena kemampuannya dalam membaca konteks dari dua arah dan dapat memahami pola bahasa dengan lebih baik, sehingga algoritma ini cocok diimplementasikan pada *chatbot*. BiLSTM dapat melakukan klasifikasi lebih baik dibandingkan dengan LSTM yang hanya bekerja dengan satu arah. Selain itu, BiLSTM mampu mencapai akurasi hingga 94.12% [12]. Maka dari itu, peneliti tertarik untuk melakukan penelitian dengan judul "Implementasi *Framework AI Project* dalam Perancangan *Chatbot* Deteksi Tingkat Depresi Mahasiswa Kabupaten Tasikmalaya".

2. Metode Penelitian

Pada bagian metodologi menjelaskan mengenai tahapan yang akan dilakukan dalam menyusun dan menyelesaikan penelitian. Dalam merancang *chatbot* deteksi tingkat depresi, digunakan *framework AI Project Cycle*. Pada artikel yang ditulis oleh Kashyap (2025), dijelaskan bahwa *Framework AI Project Cycle* menyediakan kerangka kerja untuk merencanakan, mengatur, melaksanakan, dan mengimplementasikan proyek AI untuk mencapai target. Tahapan AI Project Cycle yaitu *project scope*, *data acquisition*, *data exploration*, *modeling*, *evaluation*, dan *deployment*. Langkah-langkah pada *framework* ini sangat detail jika dibanding dengan *framework* lain, seperti *Deep Learning Life Cycle* yang melewati tahap *data exploration* [14]. Gambar 1 berikut adalah tahapan penelitian dalam studi ini.



Gambar 1. Metodologi Penelitian

2.1. Studi Pendahuluan

Pada tahap studi pendahuluan terdapat beberapa hal yang dilakukan yaitu studi literatur, wawancara, dan observasi. Tujuan dari studi pendahuluan adalah memperjelas masalah, memastikan penelitian bisa lanjut atau tidak, mendapatkan data awal dari lapangan, mengetahui penelitian sebelumnya, dan mengetahui narasumber yang dituju [15].

Studi literatur yang dilakukan adalah dengan membaca penelitian terdahulu. Sedangkan survei dilakukan dengan cara membagikan kuesioner kepada mahasiswa, mahasiswa yang dijadikan sebagai sampel penelitian, yaitu dari Sekolah Tinggi Teknologi Cipasung, Universitas Cipasung, dan Universitas Islam KH. Ruhiat. Data hasil dari survei digunakan sebagai validasi permasalahan dan masukan penelitian.

2.2. Problem Scoping

Pada tahap *problem scoping*, dilakukan pemetaan batas-batas masalah yang ingin dipecahkan dengan metode 4Ws yaitu *Who*, *What*, *Where*, dan *Why*. Permasalahan yang dipecahkan antara lain mengenai kesehatan mental yaitu depresi. Sehingga menghasilkan solusi *deep learning* yang akan diterapkan pada *chatbot* Telegram yang dapat diakses di berbagai platform.

2.3. Data Acquisition

Pada tahap *data acquisition*, dilakukan pengumpulan dataset yang relevan untuk membuat *chatbot* yang bisa mendeteksi tingkat depresi. Dataset dikumpulkan dengan menyebarkan kuesioner kepada mahasiswa Kabupaten Tasikmalaya. Selain itu, dikumpulkan juga data sintesis yang dibuat dengan bantuan transformer, dimana atributnya terdiri dari pertanyaan PHQ-9, jawaban pengguna dan label (skala likert 0-3). Dengan demikian pada tahap ini, dibuat rancangan kuesioner dan pseudocode untuk menggunakan Transformer. Pseudocode adalah cara penulisan kode program dan algoritma menggunakan bahasa sehari-hari sehingga lebih mudah dipahami [16].

2.4. Data Exploration

Pada tahap *data exploration*, eksplorasi data dilakukan dengan tujuan untuk mempersiapkan dataset yang akan digunakan untuk melatih model agar sesuai dengan kebutuhan. Proses eksplorasi data dilakukan dengan beberapa tahap yaitu:

1. *Lemmatization*, proses ini dilakukan dengan tujuan untuk mengembalikan kata-kata yang bervariasi menjadi bentuk dasarnya.
2. *Tokenization*, proses ini dilakukan dengan tujuan untuk menghitung jumlah kata yang dikirim oleh pengguna atau pada atribut jawaban di dataset.
3. *Data visualization*, proses ini dilakukan dengan tujuan untuk melihat distribusi data, untuk memastikan penyebaran data pada setiap cluster seimbang atau tidak terlalu jauh perbandingannya.
4. *Word embedding readiness*, proses ini dilakukan dengan tujuan untuk melihat keberagaman kata dan frekuensi dalam dataset, untuk memastikan embedding dapat menangkap makna dengan baik.

Dalam penelitian ini, dataset yang akan digunakan adalah data train, data validasi, dan data test.

2.5. Modeling

Pada tahap Modeling, dibuat spesifikasi model BiLSTM yang terdiri dari *pseudocode*. Algoritma BiLSTM digunakan untuk mengklasifikasi respons pengguna ke dalam skala likert (0-3) berdasarkan jawaban yang diberikan terhadap pertanyaan PHQ-9. Proses *deep learning* dimulai dengan *embedding layer*, yaitu mengubah teks jawaban mahasiswa menjadi representasi numerik dalam bentuk vektor. Selanjutnya, data yang telah dikonversi ke dalam bentuk vektor akan masuk ke dalam dua lapisan BiLSTM, yang memiliki 64 unit dan 32 neuron, dimana setiap lapisan BiLSTM terdapat hidden state yang dihasilkan dari dua arah untuk memahami konteks kalimat. Dua hidden state dari LSTM digabungkan menjadi final hidden state, seperti yang diuraikan pada persamaan berikut [17].

$$h_t^{BiLSTM} = h_t^{forward} \oplus h_t^{backward} \quad (1)$$

Dalam pemilihan algoritma pada penelitian ini dilakukan evaluasi pada beberapa model, yang bertujuan untuk memilih model dengan akurasi tertinggi. Algoritma yang dibandingkan antara lain BiLSTM dengan LSTM.

2.6. Evaluation

Pada tahap evaluation, dilakukan evaluasi terhadap kinerja model dengan menggunakan *confusion matrix*. *Confusion matrix* adalah pengukuran kinerja model klasifikasi dimana kelas yang diprediksi akan dibandingkan dengan kelas yang sebenarnya. *Confusion matrix* terdiri dari TP (*true positive*), FP (*false positive*), FN (*false negative*), dan TN (*true negative*). Berikut adalah rumus dari *confusion matrix* [18].

$$\begin{aligned} Accuracy &= \frac{TP + TN}{TP + FP + TN + FN} \\ Recall &= \frac{TP}{TP + FN} \\ Precision &= \frac{TP}{TP + FP} \\ F1 - Score &= 2 * \frac{TP + TN}{Recall + Precision} \end{aligned}$$

2.7. Pengembangan Chatbot

Setelah model BiLSTM dilatih, tahap selanjutnya adalah pengembangan *chatbot* dengan menggunakan bahasa python dan integrasi pada API Telegram. Integrasi dilakukan dengan menggunakan token yang diberikan oleh telegram setelah selesai membuat *new bot account*. Dengan demikian, pada tahap ini dibuat spesifikasi API dan rencana struktur folder. Spesifikasi API adalah dokumen terstruktur yang mendefinisikan hal-hal penting untuk merancang, membangun, dan berinteraksi dengan API perangkat lunak. Spesifikasi API berisi *endpoint*, *request*, *response*, dan dekripsi API [19].

2.8. Deployment

Pada tahap *deployment*, model BiLSTM yang telah dilatih akan di deploy pada Railway dengan menggunakan FastAPI. Selain itu, juga digunakan layanan database MySQL, database MySQL digunakan untuk menyimpan tingkat depresi mahasiswa pada setiap perguruan tinggi. Railway adalah platform *deployment* yang dirancang untuk mempermudah *life cycle* pengembangan perangkat lunak, dimulai dengan instant *deployment* dan skalabilitas yang mudah, hingga integrasi CI/CD dan observabilitas bawaan. Layanan Railway yaitu *Persistent Service* dan *Scheduled Jobs*. *Persistent Service* selalu berjalan, seperti aplikasi web, backend APIs (FastAPI), *message queues*, database *service*, dan lain-lain. Sedangkan, layanan *scheduled Jobs* dijalankan pada jadwal tertentu atau biasa disebut juga *Cron Jobs* [20].

2.9. Testing

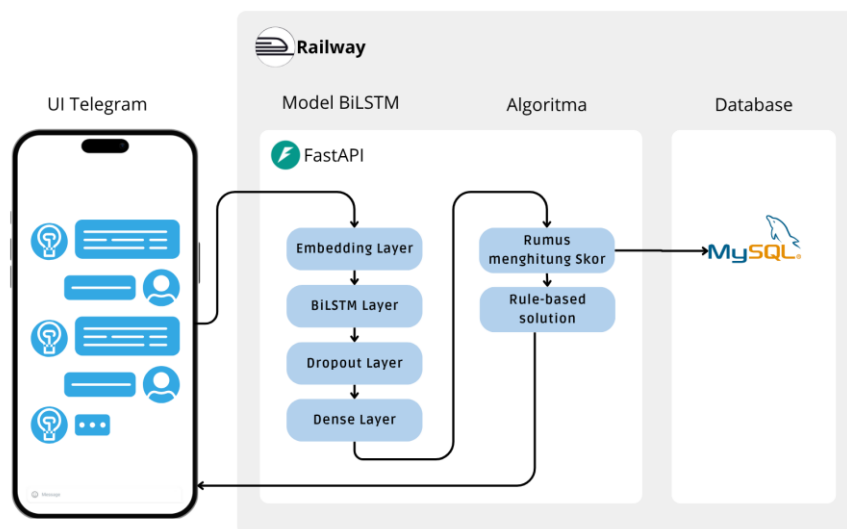
Pada tahap *testing*, dilakukan pengujian dengan menggunakan metode *crowdsourced testing*. Pengujian dilakukan melalui platform Telegram, dimana pengguna diberikan pertanyaan dari PHQ-9 dan respon pengguna di proses oleh sistem. Jika hasil yang diberikan *chatbot* sesuai dengan ekspektasi, maka sistem dianggap berhasil. Jika tidak, maka perlu dilakukan evaluasi lebih lanjut terhadap model BiLSTM atau sistem. *crowdsourced testing* dilakukan dengan menggunakan penguji yang bersumber dari banyak orang (penguji lepas) untuk melakukan pengujian yang berorientasi pada pengguna untuk *chatbot* pintar dan sistem tertentu [21]. Langkah-langkah pengujian *crowdsourced testing* yaitu keterlibatan penguji, Eksekusi skenario pengujian, Pelacakan dan pelaporan bug, Feedback dan integrasi [22].

3. Hasil dan Pembahasan

Pada hasil dan pembahasan, dijelaskan lebih detail dari proses perancangan yang dilakukan. Output dari perancangan yang dilakukan berupa UML (*Unified Modelling Language*), arsitektur teknologi, *pseudocode*, spesifikasi API, dan UAT.

3.1. Arsitektur Teknologi

Arsitektur Teknologi (*Technology Architecture*) merupakan proses merancang infrastruktur teknologi, termasuk perangkat keras, perangkat lunak, dan jaringan yang mendukung sistem informasi [23]. Arsitektur yang dibuat pada penelitian ini mencakup model BiLSTM, *framework* FastAPI, Algoritma untuk menghitung tingkat depresi, *rule-base solution*, MySQL, dan platform Railway. Gambar 2 berikut adalah arsitektur teknologi pada penelitian ini.



Gambar 2. Arsitektur Teknologi

3.2. Data Acquisition

Dataset yang dikumpulkan dengan menyebarkan kuesioner akan digunakan untuk data test. Dimana, pertanyaan nya terdiri dari identitas mahasiswa dan pertanyaan PHQ-9. Sedangkan untuk dataset train dan data validasi akan di *generate* menggunakan *transformer*. Tabel 1 berikut adalah pertanyaan kuesioner pada penelitian ini.

Tabel 1. Pertanyaan Kuesioner

No	Pertanyaan
1	Identitas magasiswa
2	Seberapa sering Anda merasa kurang tertarik atau bergairah dalam melakukan kegiatan apapun?
3	Seberapa sering Anda merasa murung, muram, atau putus asa?
4	Seberapa sering Anda mengalami sulit tidur atau mudah terbangun, atau terlalu banyak tidur?
5	Seberapa sering Anda merasa lelah atau kurang bertenaga?

No	Pertanyaan
6	Seberapa sering Anda mengalami penurunan nafsu makan atau makan berlebihan?
7	Seberapa sering Anda merasa kurang percaya diri atau merasa bahwa Anda adalah orang yang gagal atau telah mengecewakan diri sendiri atau keluarga?
8	Seberapa sering Anda mengalami kesulitan berkonsentrasi, misalnya saat membaca koran atau menonton televisi?
9	Seberapa sering Anda bergerak atau berbicara sangat lambat sehingga orang lain memperhatikannya. Atau sebaliknya, merasa resah atau gelisah sehingga Anda lebih sering bergerak dari biasanya?
10	Seberapa sering Anda merasa lebih baik mati atau ingin melukai diri sendiri dengan cara apapun?

3.3. Data Exploration

Pada bagian ini dilakukan proses *lemmatization*, *tokenization*, *data visualization*, dan *word embedding readiness*. Listing 1 berikut adalah *pseudocode* dari *data exploration*.

Listing 1. Pseudocode Data Exploration

```
#Akses direktori dataset melalui Google Drive
1. Import Google Drive('content/drive')

#Read Dataset JSON
2. Open file ← phq9_no1_synthetic_400jawaban.json
3. Ekstrak data: teks ← kolom 'jawaban',
                label ← kolom 'likert'

# Check Data
4. print(data.head)
5. print(data.info)
6. duplicate ← count_duplicate(data)
   IF duplikat > 0 THEN
       PRINT duplikat
   ENDIF

# Lemmatization (Stemming Bahasa Indonesia)
7. from library Sastrawi
8. FOR kalimat IN 'teks'DO
   → hasil ← stemmer.stem(kalimat)
   → save ke kolom baru: 'jawaban_lemmatized'
ENDFOR

# Tokenisasi dan Padding
9. calculate_kata(teks)
10. Inisialisasi tokenizer dengan OOV token,
    Fit tokenizer pada teks lemmatized,
    Konversi teks ke sequence numerik,
    Lakukan padding agar semua input punya panjang yang sama,
11. Save tokenizer to 'tokenizer.pkl'.

# Visualisasi Data
12. Plot distribusi label (likert) sebagai bar chart,
    Plot distribusi panjang jawaban (jumlah kata),
    word cloud ← generate_wordcloud(text_lemmatized),
    Print 10 kata yang paling sering muncul.
```

3.4. Modeling

3.2.1. Arsitektur BiLSTM

BiLSTM merupakan *neural network* yang berisi dua *layer* LSTM. BiLSTM memanfaatkan konteks sebelumnya dan setelahnya dengan memproses informasi dari dua arah dengan *hidden layer* terpisah. BiLSTM mengkoneksikan dua *hidden layer* yang berlawanan menuju ke satu output. *Forward layer* untuk merepresentasikan konteks sebelumnya, dan *backward layer* untuk merepresentasikan konteks setelahnya. Gambar 3 berikut adalah desain *layer* pada model BiLSTM [18].



Gambar 3. Arsitektur BiLSTM

1. Embedding Layer

Data input untuk layer ini merupakan teks dari jawaban pengguna yang telah diolah pada tahap *data exploration* atau *pre-processing*, yang berupa token. Lalu, token tersebut di *encode* sebagai bilangan bulat

dan dipetakan sebagai vektor representasi melalui *embedding layer*. *Layer* ini mempelajari representasi kata selama proses pelatihan.

2. BiLSTM Layer

Setelah *embedding layer*, vektor kata dikirim ke BiLSTM layer. Layer ini membaca urutan teks dari dua arah, berikut adalah contoh penerapannya.

- Forward* : [Saya] → [merasa] → [putus asa] → [hari] → [ini], akan menghasilkan output "Saya merasa putus asa".
- Backward* : [ini] → [hari] → [putus asa] → [merasa] → [saya], akan menghasilkan output "putus asa hari ini".

Kedua hasil ini digabung (*concatenation*) menjadi satu vektor representasi gabungan yang mencakup konteks dari kedua arah proses algoritma BiLSTM.

3. Dropout Layer

Dropout layer merupakan proses menonaktifkan unit tertentu (*neuron*) dalam *layer* dengan probabilitas tertentu. Penggunaan *dropout* dilakukan dengan menetapkan setengah dari *activity layer* menjadi nol. Penerapan lapisan *dropout* bertujuan untuk mempercepat proses pelatihan model.

4. Dense Layer

Dense layer digunakan sebagai output *layer* yang *fully connected*. Lapisan ini menggunakan beberapa parameter, seperti units dan aktivasi. Units merupakan bilangan positif yang menunjukkan ukuran keluaran lapisan, sesuai dengan jumlah label atau target yang diprediksi. Pada kasus ini, kalimat "Saya merasa putus asa hari ini" akan dipetakan oleh model pada skala likert PHQ-9 yaitu 0 sampai 3.

3.2.2. Pseudocode Model BiLSTM

Arsitektur model dikonstruksi menggunakan pendekatan *Sequential*, dimana BiLSTM terdiri dari empat *layer* yaitu *embedding layer*, BiLSTM *layer*, *dropout layer*, dan *dense layer*. *Embedding layer* memiliki dimensi input 10.000, output 128, panjang input 50, BiLSTM dengan 64 unit, lapisan *dropout* sebesar 0.5 untuk mencegah overfitting, lalu *dense layer* berjumlah 32 *neuron* dengan aktivasi ReLU. Setelah arsitektur dibangun, proses pelatihan dilakukan dengan *batch size* 32 selama 10 *epoch*, serta divalidasi menggunakan data uji. Setelah proses pelatihan selesai, model disimpan dalam format. Keras dan dapat diunduh untuk keperluan implementasi lebih lanjut. Listing 2 berikut adalah pseudocode dari model BiLSTM.

Listing 2. Pseudocode model BiLSTM

```
Pseudocode: BiLSTM-based Depression Classification from Text Responses
# Pseudocode: BiLSTM-Based Depression Classification
# Input: teks dari pengguna (kalimat atau jawaban)
# Output: skor klasifikasi (0, 1, 2, atau 3)

Begin

# Label Encoding
1. Encode label (likert) menjadi nilai numerik

# Split Data Sintesis
2. x_train, x_val, y_train, y_val ← split(x, y val_size = 0.2)

# Process Data Questioner
3. Lemmatize and Tokenize (x_test, y_test)

# Arsitektur Model BiLSTM
4. model ← Sequential()
5. Embedding Layer (input_dim=10.000, output_dim=128, input_length=50)
6. BiLSTM Layer (64 unit)
7. Dropout Layer (rate 0.5)
8. Dense Layer (32 neuron, ReLU)
   Output Layer (4 neuron, softmax)
9. model.compile:
   → Loss: sparse_categorical_crossentropy
   → Optimizer: Adam
   → Metrics: Accuracy

# Train Model
10. Model.compile
   → batch_size = 32
   → epochs = 10
   → validasi dengan data test

# Simpan Model
```

```
11. model.save('model_bilstm.keras')
12. Download('model_bilstm.keras')

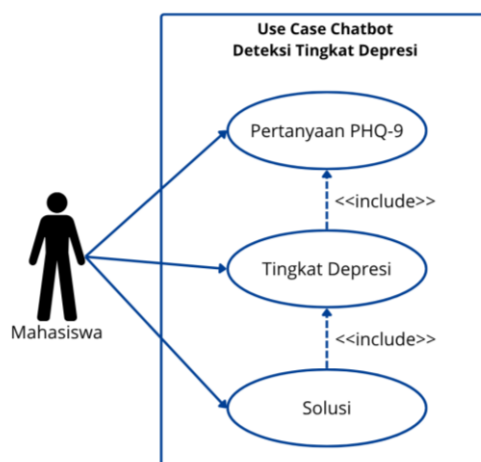
# Evaluasi Model
14. Prediksi X_test,
    Konversi hasil prediksi probabilitas menjadi kelas (argmax),
    Buat confusion matrix,
    Plot confusion matrix (4 kelas: 0-3),
    Print classification report (precision, recall, f1-score).

End
```

3.5. Pengembangan Chatbot

3.4.1. Use Case Diagram

Use case diagram adalah salah satu dari jenis diagram UML (*Unified Modelling Language*) yang menggambarkan hubungan interaksi antara sistem dan aktor [24]. Dalam use case ini, dijelaskan bahwa mahasiswa yang mengakses *chatbot* akan menjawab pertanyaan PHQ-9. Lalu, setelah selesai menjawab pertanyaan mahasiswa akan mendapatkan tingkat depresi dan solusi. Gambar 4 berikut adalah use case *chatbot* deteksi tingkat depresi tersebut.



Gambar 4. Use Case Diagram

3.4.2. Spesifikasi API

Terdapat dua *endpoint* utama yang akan digunakan untuk membangun *chatbot* deteksi tingkat depresi yaitu *endpoint* `/predict` dan `/result`. *Endpoint* `/predict` berfungsi untuk menangani *request* dari *chatbot* berupa jawaban pengguna, lalu response berupa prediksi label berdasarkan skala likert serta pertanyaan berikutnya. Sedangkan, *endpoint* `/result` berfungsi untuk menampilkan tingkat depresi mahasiswa dan solusi berdasarkan tingkat depresi tersebut. Listing 3 berikut adalah spesifikasi API pada penelitian ini.

Listing 3. Spesifikasi API

```
Endpoint: /predict
Method: POST
Content-Type: application/json
Description: Menerima input teks dari chatbot, memproses teks, dan memprediksi label ke dalam skala likert.

Request Body:
{
  "session_id": "abcd123",
  "answer": "Saya merasa kurang bergairah setiap hari"
}

Response:
{
  "completed": false,
  "next_question": "Seberapa sering Anda merasa murung, muram, atau putus asa?",
  "next_index": 2
  "predicted_label": "2"
}

Endpoint: /Result?session_id=abcd123
Method: GET
```

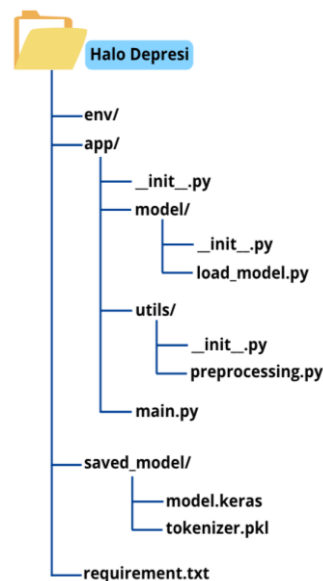
Description: Mengambil score tingkat depresi dan solusinya

Response:

```
{
  "Score": 15,
  "level": "Sedang",
  "saran": "Anda mengalami depresi sedang, dianjurkan untuk konsultasi ke ahli profesional atau psikoterapi "
}
```

3.4.3. Struktur Folder API

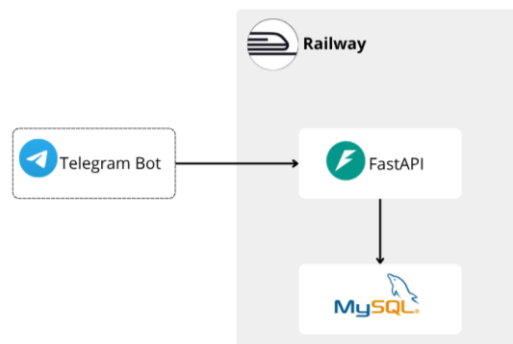
Folder env merupakan lingkungan *virtual* untuk mengelola dependensi Python. Folder app berisi kode utama aplikasi, yang terdiri dari beberapa modul. Modul model berfungsi untuk *load* model BiLSTM yang telah dilatih, modul *utils* berfungsi untuk membersihkan dan mempersiapkan data teks sebelum diproses oleh model, dan file main.py berfungsi untuk mengatur *routing* API. Lalu folder saved_model berfungsi untuk menyimpan file model yang telah dilatih dan menyimpan file tokenizer.py. Gambar 5 berikut adalah struktur directory API menggunakan *framework* FastAPI.



Gambar 5. Struktur Directory API

3.6. Deployment

Pada penelitian ini *deployment* akan dilakukan pada platform Railway. *Service* yang digunakan yaitu FastAPI dan MySQL. FastAPI merupakan bagian dari *Persistent Service* adalah kerangka kerja web yang modern dan cepat (berkinerja tinggi) untuk membangun API dengan Python berdasarkan petunjuk tipe Python standar. Lalu, MySQL yang juga merupakan bagian *Persistent Service* digunakan untuk menyimpan jawaban dari pengguna dan tingkat depresi yang dihasilkan. Layanan Railway sangat fleksibel, sehingga hanya diperlukan beberapa setup untuk menggunakan layanan pada Railway [20]. Gambar 6 berikut adalah arsitektur railway pada penelitian ini.



Gambar 6. Arsitektur Railway

Terdapat beberapa cara untuk *deployment* pada Railway, dimana Railway telah mempermudah metode *deployment*. Metode *deployment* pada Railway antara lain:

1. *One-click deploy from a template* merupakan metode di mana deploy bisa langsung dilakukan hanya dengan satu klik dari sebuah template (proyek contoh yang sudah disiapkan).
2. *From a GitHub repository* untuk *delivery* aplikasi dari GitHub. Platform Railway akan menarik (*pull*) kode dari GitHub, lalu otomatis membangun dan menjalankan aplikasinya.
3. *Using the CLI* merupakan metode dengan deploy aplikasi melalui *Command Line Interface* (CLI).
4. *Using a dockerfile* yang berisi instruksi untuk membangun *image docker* (aplikasi + *environment*-nya). Digunakan ketika telah memiliki direktori atau repo di komputer.

3.7. Testing

Pengujian *chatbot* akan dilakukan dengan menggunakan metode *crowdsourced testing*, dimana pengujian akan dilakukan oleh 10 orang yang telah memenuhi kriteria. Dalam pengujian ini, ditetapkan skenario pengujian yang disusun dalam format *User Acceptance Criteria* (UAT) untuk menilai apakah sistem sudah memenuhi kebutuhan pengguna. Tabel 1 berikut adalah skenario pengujian yang digunakan dalam tahap ini.

Tabel 2. UAT (User Acceptance Criteria)

No	Fitur	Langkah pengujian	Data	Hasil yang diharapkan	Status
1	Akses <i>chatbot</i>	Pengguna membuka <i>chatbot</i> via Telegram.	-	<i>Chatbot</i> merespon sapaan awal dan instruksi penggunaan.	
2	Input jawaban pengguna	Pengguna mengisi semua pertanyaan PHQ-9	Jawaban pengguna berupa kalimat atau teks	Jawaban pengguna berhasil diinput.	
3	Pertanyaan PHA-9	-	-	<i>Chatbot</i> merespon atau memberikan pertanyaan selanjutnya secara berurutan.	
4	Deteksi tingkat depresi	Pengguna selesai menjawab semua pertanyaan PHQ-9	Jawaban pengguna	<i>Chatbot</i> menampilkan tingkat depresi.	
5	Solusi	Pengguna selesai menjawab semua pertanyaan PHQ-9	tingkat depresi	<i>Chatbot</i> menampilkan solusi untuk mengatasi depresi berdasarkan tingkat depresinya.	
6	Penanganan input tidak sesuai	Pengguna memberikan input tidak relevan atau kosong	ahdkahdakh atau skip jawaban	<i>Chatbot</i> meminta ulang memberi instruksi perbaikan pada jawaban.	

4. Kesimpulan

Penelitian ini menerapkan *Framework AI Project* secara terstruktur, untuk melakukan perencanaan *data acquisition*, *data exploration*, *modeling*, *evaluation*, pengembangan *chatbot*, *deployment*, dan *testing*. Dengan mengikuti alur pengembangan pada *framework* tersebut perancangan *chatbot* deteksi tingkat depresi dapat dilakukan secara efisien dan lebih mudah karena tahapannya detail. Pada penelitian ini, dibuat perancangan sistem mencakup *Unified Modelling Language* (UML), arsitektur teknologi, arsitektur Railway, *pseudocode* model, spesifikasi *Application Programming Interface* (API), dan *User Acceptance Criteria* (UAT). Perancangan yang dihasilkan menggambarkan sistem secara menyeluruh dan detail, sehingga dapat dijadikan acuan dalam tahap implementasi sistem tersebut.

Notasi

Dimana:

$h_t^{forward}$: representasi hidden state yang dihitung dari kiri ke kanan (urutan maju)

$h_t^{backward}$: representasi hidden state yang dihitung dari kanan ke kiri (urutan mundur)

$\oplus$: operasi concatenation (penggabungan vektor) yang menggabungkan informasi dari kedua arah.

True Positive (TP): mendefinisikan pelanggan yang melakukan churn (positif) dan diklasifikasikan sebagai churn (positif).

True Negative (TN): mendefinisikan nasabah yang tidak melakukan churn (negatif) dan diklasifikasikan sebagai tidak melakukan churn (negatif).

False Positive (FP): mengidentifikasi pelanggan yang tidak melakukan churn (negatif) dan diklasifikasikan sebagai churn (positif).

False Negative (FN): mengidentifikasi nasabah yang melakukan churn (positif) dan diklasifikasikan sebagai tidak melakukan churn (negatif).

Ucapan Terimakasih

Terima kasih kepada Fakultas Teknik Universitas Suryakencana yang telah menjadi wadah bagi para peneliti untuk mengembangkan penelitian jurnal ini. Semoga penelitian ini dapat memberikan kontribusi yang besar bagi kemajuan ilmu pengetahuan di Indonesia. Selain itu, terima kasih kepada Sekolah Tinggi Teknologi Cipasung dan pembimbing saya yang telah memberikan arahan, dukungan, dan motivasi selama proses penyusunan penelitian ini.

Referensi

- [1] Y. P. Omega and S. Herman, "Penanganan Depresi melalui Dimensi Rohani di Kota Bandung," *Fidei: Jurnal Teologi Sistemika dan Praktika*, vol. 7, no. 1, pp. 1–20, Jun. 2024, doi: 10.34081/fidei.v7i1.488.
- [2] N. A. Surachman, M. A. Jaya, R. P. I. Abdullah, I. K. A. Bakhtiar, and W. N. Aisyah, "Gambaran Derajat Stres, Cemas Dan Depresi Pada Mahasiswa Fakultas Kedokteran Universitas Muslim Indonesia Angkatan 2021," *Journal Of Multidisciplinary Research and Development*, vol. 6, no. 6, Sep. 2024, doi: <https://doi.org/10.38035/rj.v6i6>.
- [3] Kemenkes, "Depresi pada Anak Muda di Indonesia." Badankebijakan.kemkes, 2023. [Online]. Available: chrome-extension://efaidnbmnnnibpcjpcglclefindmkaj/https://repository.badankebijakan.kemkes.go.id/id/eprint/5532/1/03%20factsheet%20Keswa_bahasa.pdf
- [4] V. S. Alya et al., "Analisis pengaruh penerapan sistem blok terhadap motivasi belajar mahasiswa pendidikan biologi di universitas negeri medan," *Jurnal intelek insan cendikia*, vol. 1, no. 4, Jun. 2024.
- [5] N. Muhamad, "Jawa Barat, Provinsi dengan Angka Depresi Tertinggi," Databoks, Jun. 14, 2024. Accessed: Mar. 16, 2025. [Online]. Available: <https://databoks.katadata.co.id/layanan-konsumen-kesehatan/statistik/2ffe4020a79b997/jawa-barat-provinsi-dengan-angka-depresi-tertinggi>
- [6] Open Data Tasikmalaya, "Open data," Open Data Tasikmalaya. Accessed: Mar. 16, 2025. [Online]. Available: <https://opendata.tasikmalayakota.go.id/artikel/kependudukan-kota-tasikmalaya-tahun-2023-semester-i>
- [7] Ikatan Psikolog Klinis Indonesia, "Statistik Keanggotaan ikatan psikolog klinis Indonesia Wilayah Jawa Barat," data.ipkindonesia. Accessed: Mar. 10, 2025. [Online]. Available: <https://data.ipkindonesia.or.id/statistik/wilayah/jawa-barat>
- [8] P2DPT, "Meningkatkan Akses ke Layanan Kesehatan Mental," Pusat Pengelolaan Data Pendidikan Tinggi. Accessed: Mar. 10, 2025. [Online]. Available: <https://p2dpt.uma.ac.id/2024/10/23/meningkatkan-akses-ke-layanan-kesehatan-mental/>
- [9] S. Suliati, S. Achmadi, and D. Rudhistiar, "PENERAPAN SISTEM PAKAR UNTUK DETEKSI DINI MENTAL ILLNESS DENGAN MENGGUNAKAN METODE FORWARD CHAINING DAN CERTAINTY FACTOR BERBASIS WEBSITE," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 6, no. 2, pp. 1087–1095, Jan. 2023, doi: 10.36040/jati.v6i2.5464.
- [10] S. P. Afrisia, F. M. Hana, and W. C. Wahyudin, "Implementasi Metode Long Short Term Memory (LSTM) pada Chatbot Kesehatan Mental Mahasiswa," *Sainteks*, vol. 21, no. 2, p. 107, Oct. 2024, doi: 10.30595/sainteks.v21i2.23869.
- [11] M. N. D. Miharja and S. Adhkar, "Implementasi Chatbot Deteksi Depresi Dini Pada Mahasiswa dengan PHQ-9 (Patient Health Questionnaire) menggunakan NLP (Natural Language Processing)," *Prosiding SAINTEK*, vol. 1, no. 1, pp. 2962–3545, Jul. 2022, doi: <https://gotriple.eu/documents/ftunivpelbanojs%3Aoi%3Aois.pkp.sfu.ca%3Aarticle%2F1156>.
- [12] K. S. Nugroho1, I. Akbar, A. N. Suksmawati, and Istiadi, "Deteksi depresi dan kecemasan pengguna twitter menggunakan bidirectional lstm." pp. 2622–1284, Dec. 15, 2021.
- [13] V. Kashyap, "Introducing AI project cycle - Vikas kashyap," Medium, Mar. 28, 2025. Accessed: Apr. 05, 2025. [Online]. Available: https://medium.com/@vikas_kashyap/introducing-ai-project-cycle-5ad0025a6b0c
- [14] H. Nafi'ah, "Deep learning - Hanin nafi'ah," Medium, Oct. 03, 2021. Accessed: Apr. 05, 2025. [Online]. Available: <https://medium.com/@hannnfh/deep-learning-3840922fc77b>
- [15] Y. A. Azis, "Studi Pendahuluan: Definisi, Tujuan, Cara Membuat dan Contoh," Deepublish Store, May 15, 2023. Accessed: Mar. 05, 2025. [Online]. Available: <https://deepublishstore.com/blog/studi-pendahuluan/>
- [16] N. Roviery, F. Baskoro, and A. S. Ahmadiyah, "Otomatisasi Pembuatan Storyboard Video Tutorial dengan Menggunakan ChatGPT pada Website Pelatihan Vokasi," *Jurnal Teknik ITS*, vol. 14, no. 1, Apr. 2025, doi: 10.12962/j23373539.v14i1.150940.
- [17] J. Amalia, Juanda Pakpahan, Melani Pakpahan, and Yeni Panjaitan, "Model Klasifikasi Berita Palsu Menggunakan Bidirectional LSTM dan Word2vec sebagai Vektorisasi," *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 4, pp. 3319–3331, Dec. 2022, doi: 10.35957/jatisi.v9i4.1332.
- [18] A. F. Azmi and A. Voutama, "Prediksi churn nasabah bank menggunakan klasifikasi random forest dan decision tree dengan evaluasi confusion matrix," *KOMPUTA: Jurnal Ilmiah Komputer dan Informatika*, vol. 13, no. 1, Apr. 2024.
- [19] AppMaster, "Spesifikasi API," AppMaster - ultimate all-in no-code platform. Accessed: Jun. 09, 2025. [Online]. Available: <https://appmaster.io/id/glossary/spesifikasi-api>
- [20] Railway, "About Railway," About Railway | Railway Docs. Accessed: Mar. 21, 2025. [Online]. Available: <https://docs.railway.com/overview/about-railway>
- [21] J. Gao, R. Agarwal, and P. Garsole, "AI testing for intelligent chatbots – a case study," *MDPI AG*, Feb. 2025. Accessed: Mar. 20, 2025. [Online]. Available: <https://doi.org/10.20944/preprints202502.1742.v1>
- [22] Testlio, "Crowdsourced testing: The ultimate crowdtesting guide," Testlio. Accessed: Mar. 20, 2025. [Online]. Available: <https://testlio.com/blog/what-is-crowdsourced-testing/>
- [23] A. Tryana and L. Larasati, "Perencanaan Arsitektur Sistem Informasi Pengarsipan Surat dengan Menggunakan Metode EAP," *Jurnal Accounting Information System (AIMS)*, vol. 8, no. 1, pp. 57–70, Mar. 2025, doi: 10.32627/aims.v8i1.1343.
- [24] D. Intern, "Contoh Use Case Diagram Lengkap dengan Penjelasannya," Dicoding Blog. Accessed: Jun. 09, 2025. [Online]. Available: <https://www.dicoding.com/blog/contoh-use-case-diagram/>