DAFTAR LAMPIRAN

Lampiran 1

Surat Keterangan Bebas Plagiat

UNIVERSITAS DARMA PERSADA
UPT PERPUSTAKAAN
Gedung Rektorat Lantai 3,
Jl.Taman Malaka Selatan, Pondok Kelapa – Jakarta Timur 13450

SURAT KETERANGAN
HASIL PENGECEKAN TURNITIN

UPT Perpustakaan Universitas Darma Persada menerangkan telah selesai melakukan pemeriksaan duplikasi/*similarity* menggunakan perangkat lunak Turnitin terhadap hasil karya sebagai berikut:

Judul	: Pengembangan Chatbot Layanan Sistem Informasi Menggunakan Natural Language Processing dan BERT di Fakultas Teknik Universitas Darma Persada
Penulis	: Wisnu Anggoro
NIM	: **2021230010**
Tgl pemeriksaan	: 28 Juli 2025

Dengan hasil Tingkat Kesamaan (*similarity index*) **15%**

Demikian Surat Keterangan kami buat, untuk dipergunakan sebagaimana mestinya.

Jakarta, 28 Juli 2025

Ka.UPT Perpustakaan Unsada

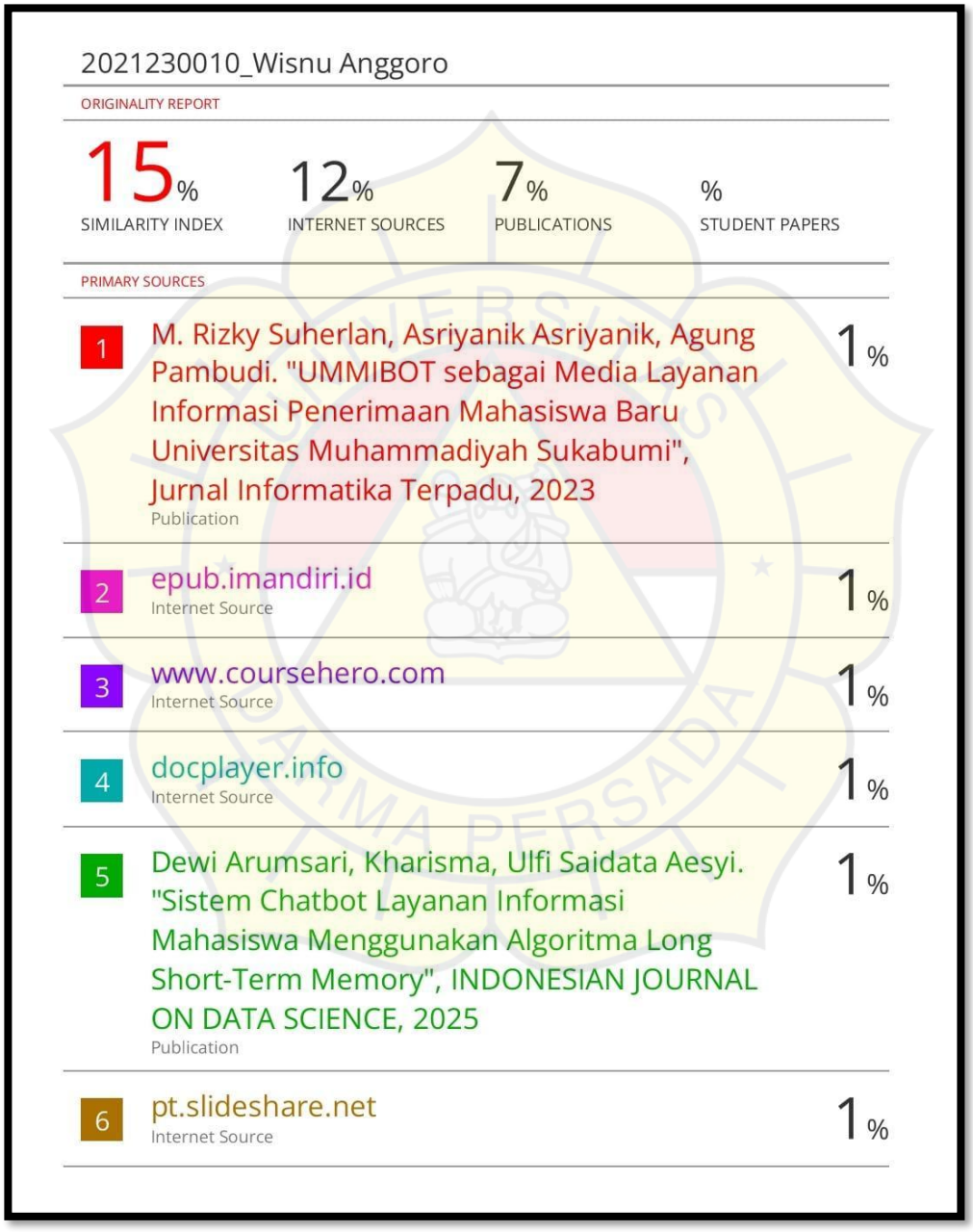
Yus Rusmiyati, SS., MM

Batas maksimal similarity 30% untuk Fakultas Sastra dan Ekonomi

Batas maksimal similarity 25% untuk Fakultas Teknik, Kelautan dan Pasca Sarjana

Lampiran 2

Hasil Turnitin



2021230010_Wisnu Anggoro

ORIGINALITY REPORT

15%	12%	7%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	M. Rizky Suherlan, Asriyanik Asriyanik, Agung Pambudi. "UMMIBOT sebagai Media Layanan Informasi Penerimaan Mahasiswa Baru Universitas Muhammadiyah Sukabumi", Jurnal Informatika Terpadu, 2023 Publication	1%
2	epub.imandiri.id Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	docplayer.info Internet Source	1%
5	Dewi Arumsari, Kharisma, Ulfi Saidata Aesyi. "Sistem Chatbot Layanan Informasi Mahasiswa Menggunakan Algoritma Long Short-Term Memory", INDONESIAN JOURNAL ON DATA SCIENCE, 2025 Publication	1%
6	pt.slideshare.net Internet Source	1%

Lampiran 3

Source Code app.py

```
from flask import Flask, request, jsonify

import json

import torch

from sentence_transformers import SentenceTransformer, util



app = Flask(__name__)



# ☑ Load model hasil fine-tuning

model = SentenceTransformer('model_finetuned_unsada')



# ☑ Load dataset FAQ

with open('train_data.json', 'r', encoding='utf-8') as f:

    faq_data = json.load(f)
```

```
# ☑ Siapkan daftar pertanyaan dan jawaban

all_questions = []

all_answers = []



for item in faq_data:

    question = item['question']

    answer = item['answer']



    # Tambahkan pertanyaan utama

    all_questions.append(question)

    all_answers.append(answer)



    # Tambahkan variasi (jika ada)

    for variation in item.get('variations', []):

        all_questions.append(variation)

        all_answers.append(answer)
```

```
# ☑ Encode semua pertanyaan (sekali saja)
question_embeddings = model.encode(all_questions, convert_to_tensor=True, normalize_embeddings=True)

@app.route('/chat', methods=['POST'])
def chat():
    data = request.get_json()
    user_input = data.get("message", "").strip()

    if not user_input:
        return jsonify({"response": "Pesan kosong tidak dapat diproses."})

    # ☑ Encode input pengguna
    input_embedding = model.encode(user_input, convert_to_tensor=True, normalize_embeddings=True)

    # ☑ Hitung similarity
```

```
            similarity_scores     =     util.cos_sim(input_embedding,
question_embeddings)[0]
    best_idx = torch.argmax(similarity_scores).item()
    best_score = similarity_scores[best_idx].item()


    # ☑ Threshold relevansi jawaban
    threshold = 0.60
    if best_score >= threshold:
        answer = all_answers[best_idx]
    else:
        answer = "Maaf, saya tidak memahami pertanyaan Anda. Silakan
ulangi dengan pertanyaan yang lebih spesifik."


    return jsonify({"response": answer})


if __name__ == '__main__':
    app.run(debug=True, host='0.0.0.0', port=5000)
```