Fakultas Teknik,
Universitas Wijaya Kusuma Surabaya
ISSN 3090-1154 (Media Online)
Volume 2, Tahun 2025

Analisis Dan Perancangan Sistem Pendeteksian *Phishing* Berbasis *AI* Pada Platform *Whatsapp* Dengan Pendekatan Bahasa Lokal Surabaya

Fahmi Mochtar Efendi*[1], Firman Hadi Sukma Pratama[2]
[1,2] Program Studi Informatika, Fakultas Teknik Universitas Wijaya Kusuma Surabaya
Email: [1]fahmiefendi644@gmail.com, [2]firmanpratama@uwks.ac.id
*Penulis Koresponden

Abstrak

Phishing merupakan salah satu bentuk kejahatan siber yang sering terjadi melalui *platform* pesan instan seperti *WhatsApp*. Pesan *phishing* memanfaatkan teknik manipulasi bahasa untuk mengecoh pengguna, terutama dalam konteks lokal yang menggunakan bahasa daerah. Penelitian ini bertujuan untuk menganalisis dan merancang sistem deteksi *phishing* berbasis *Artificial Intelligence* (*AI*) dengan pendekatan bahasa lokal Surabaya. Sistem ini dibangun dengan memanfaatkan model pemrosesan bahasa alami (*Natural Language Processing*) untuk memahami dan mengenali pola-pola phishing dalam dialek Surabaya. *Dataset* dikumpulkan dari simulasi pesan *phishing* dan *non-phishing* yang disusun menggunakan kosakata khas Surabaya. Model *AI* dilatih menggunakan algoritma klasifikasi seperti *TF-IDF (Term Frequency–Inverse Document Frequency)* dan *Logistic Regression*. Hasil analisis menunjukkan bahwa pendekatan bahasa lokal meningkatkan akurasi sistem dalam mendeteksi *phishing* dibandingkan dengan pendekatan bahasa Indonesia umum. Rancangan sistem ini diharapkan dapat meningkatkan kesadaran dan perlindungan masyarakat lokal terhadap ancaman *phishing* digital yang kian berkembang.

Kata kunci: *AI*, bahasa lokal, *NLP, phishing*, *WhatsApp.*

Abstract

Phishing is one of the most common forms of cybercrime, frequently occurring through instant messaging platforms such as WhatsApp. Phishing messages often exploit language manipulation techniques to deceive users, especially within local contexts that involve the use of regional dialects. This study aims to analyze and design a phishing detection system based on Artificial Intelligence (AI) with a focus on the Surabaya local dialect. The system is built using Natural Language Processing (NLP) models to understand and recognize phishing patterns in Surabaya dialect expressions. The dataset was compiled from simulated phishing and non-phishing messages composed using typical Surabaya vocabulary. The AI model was trained using classification algorithms such as Naive Bayes and Support Vector Machine (SVM). The analysis shows that incorporating local language approaches improves the system's accuracy in detecting phishing compared to general Indonesian language models. This system design is expected to raise awareness and enhance digital security for local communities against the growing threat of phishing.

***Keywords**: AI, local language, NLP, phishing, WhatsApp.*

I PENDAHULUAN

1.1 Latar Belakang

Dalam era digital saat ini, *phishing* menjadi salah satu ancaman siber paling umum dan merugikan, terutama melalui *platform* komunikasi instan seperti *WhatsApp*. Modus operandi dari *phishing* biasanya melibatkan pesan manipulatif yang bertujuan untuk mencuri informasi sensitif pengguna, seperti kredensial *akun*, *data* pribadi, atau akses ke sistem keuangan. Banyak kasus menunjukkan bahwa teknik yang digunakan dalam *phishing* tidak hanya bersifat teknis, namun juga eksploitasi bahasa, terutama ketika pelaku menyasar komunitas lokal dengan pendekatan yang lebih personal dan familiar secara kultural.

Di Indonesia, khususnya di daerah Surabaya, penggunaan bahasa lokal atau dialek khas dalam komunikasi sehari-hari sangat dominan. Hal ini turut dimanfaatkan oleh pelaku *phishing* untuk menyusun pesan yang terkesan akrab dan meyakinkan, sehingga memperbesar kemungkinan korban untuk terjebak. Sayangnya, sebagian besar sistem deteksi *phishing* yang ada saat ini hanya dibangun untuk mengenali teks

dalam Bahasa Indonesia formal atau Bahasa Inggris, sehingga kurang efektif dalam mengenali pola bahasa lokal.

Menanggapi tantangan tersebut, penelitian ini merancang dan membangun sebuah sistem pendeteksi *phishing* berbasis *Artificial Intelligence* (*AI*) yang terintegrasi dengan *platform WhatsApp* dan memiliki kemampuan memahami bahasa lokal Surabaya. Sistem ini memanfaatkan pendekatan *Natural Language Processing* (*NLP*) untuk mengenali karakteristik pesan *phishing* dari *dataset* yang disusun berdasarkan kosakata khas Surabaya, kemudian diklasifikasikan menggunakan algoritma seperti *TF-IDF* dan *Logistic Regression* [1].

Sistem dikembangkan dalam bentuk *API* berbasis *FastAPI* yang dapat menerima pesan secara *real-time* dari *WhatsApp*, menganalisisnya melalui model *AI*, menyimpan hasilnya ke *database MySQL*, dan memberikan respon berupa peringatan *phishing* kepada pengguna. Dengan pendekatan ini, sistem tidak hanya mendeteksi *phishing* secara otomatis, tetapi juga mengintegrasikan konteks lokal yang lebih relevan dengan target masyarakat pengguna.

Diharapkan, melalui perancangan sistem ini, masyarakat lokal dapat memperoleh perlindungan yang lebih baik terhadap ancaman *phishing*, serta meningkatkan kesadaran digital dalam menghadapi manipulasi pesan yang bersifat sosial-teknikal.

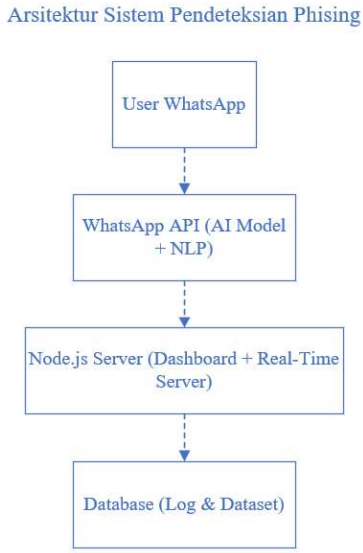
II METODE PENELITIAN

2.1 Arsitektur Sistem

Sistem pendeteksi pesan phishing pada platform WhatsApp ini dirancang dengan memanfaatkan pendekatan *Natural Language Processing* (NLP) berbasis model *IndoBERT*, serta diintegrasikan dengan *chatbot* WhatsApp untuk otomatisasi proses deteksi dan penanganan [3]. Arsitektur sistem secara umum terdiri dari empat komponen utama, yaitu: modul bot WhatsApp, backend server (*FastAPI*), model deteksi phishing berbasis *IndoBERT*, dan penyimpanan basis data lokal menggunakan MySQL.

Alur sistem dapat dijelaskan sebagai berikut:

1) Pengiriman Pesan oleh Pengguna
   Pengguna mengirimkan pesan melalui WhatsApp ke dalam grup. Bot WhatsApp yang dibangun menggunakan *Node.js* dengan pustaka whatsapp-web.js, secara otomatis mendeteksi dan meneruskan setiap pesan masuk ke backend untuk dianalisis.
2) Pemrosesan Pesan oleh Backend
   Backend dikembangkan menggunakan *FastAPI* (Python) yang berfungsi menerima pesan, melakukan pra-pemrosesan teks, serta mengirimkan pesan tersebut ke model deteksi phishing.
3) Klasifikasi Pesan dengan IndoBERT
   Pesan yang masuk akan diproses menggunakan model *IndoBERT* yang telah dilatih untuk mengenali pola bahasa phishing, termasuk konteks lokal berbahasa Surabaya. Model akan mengklasifikasikan pesan ke dalam kategori “phishing” atau “non-phishing”.
4) Penanganan Pesan Phishing
   Jika pesan terdeteksi sebagai phishing, maka sistem akan secara otomatis:
   a) Menghapus pesan dari grup WhatsApp,
   b) Mengirimkan salinan pesan tersebut ke grup admin untuk pengawasan lebih lanjut,
   c) Menyimpan data pesan, waktu kejadian, dan informasi pengirim ke dalam basis data lokal berbasis MySQL.
5) Penyimpanan dan Logging
   Semua data pesan yang terklasifikasi sebagai phishing akan disimpan untuk keperluan log, analisis lanjutan, atau pelaporan. Penyimpanan dilakukan pada sistem lokal dengan skema basis data yang telah dirancang sebelumnya (whatsapp_ai_schema.sql).



Gambar 1. Arsitektur Sistem Secara Umum

2.2 Dataset dan Preprocessing

Dalam penelitian ini, dataset yang digunakan terdiri dari kumpulan pesan teks dalam bahasa Indonesia dan variasi lokal bahasa Surabaya yang berpotensi mengandung unsur phishing. Dataset diperoleh melalui dua metode, yaitu:

1) Pengumpulan Data Simulasi
   Dataset awal dikumpulkan secara manual dari simulasi pesan-pesan phishing yang umum terjadi di platform WhatsApp, seperti ajakan mengikuti tautan palsu, permintaan OTP, dan penipuan berkedok hadiah. Beberapa pesan disesuaikan dalam bentuk bahasa informal khas Surabaya untuk mengakomodasi konteks lokal [4].
2) Sumber Data Tambahan
   Untuk memperkaya variasi dan validitas data, digunakan pula data publik dari dataset phishing teks Indonesia, yang kemudian dikurasi untuk memastikan relevansi dan kesesuaian dengan konteks komunikasi WhatsApp.

Preprocessing

Tahapan *preprocessing* dilakukan sebelum data dimasukkan ke dalam model IndoBERT. Proses ini meliputi:

1) Tokenisasi untuk memecah teks menjadi token atau kata-kata, dengan mempertahankan struktur kalimat asli agar sesuai dengan format input IndoBERT.
2) Pembersihan Teks untuk menghapus karakter khusus, emoji, *URL shortener*, dan simbol yang tidak relevan.
3) Normalisasi Bahasa dengan mengubah istilah lokal Surabaya ke dalam bentuk yang dapat dikenali oleh model, namun tetap mempertahankan gaya bahasa agar deteksi kontekstual tetap akurat.
4) Labeling data dilabeli secara biner dengan dua kelas, yaitu “phishing” dan “non-phishing”.

Model dilatih dan diuji menggunakan skema *train-test split* dengan proporsi 80:20. Evaluasi performa dilakukan menggunakan metrik akurasi, presisi, recall, dan F1-score.

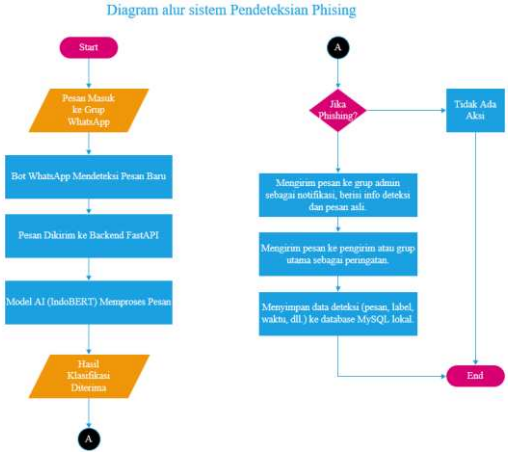
2.3 Pengembangan Sistem dan Integrasi

Sistem pendeteksi *phishing* ini dikembangkan berbasis arsitektur *modular*, yang mengintegrasikan model *IndoBERT* dengan layanan *WhatsApp* melalui pendekatan *API*. Secara garis besar, sistem terdiri dari empat komponen utama: (1) *Backend Deteksi AI*, (2) *Integrasi WhatsApp Bot*, (3) *Notifikasi* ke Grup *Admin*, dan (4) Penyimpanan *Data* Lokal.

1) *Backend Deteksi AI* (*FastAPI* + *IndoBERT*)
   Modul *backend* dikembangkan menggunakan *FastAPI* dengan *Python* sebagai bahasa pemrograman utama [5]. Model deteksi menggunakan *IndoBERT* yang telah dilatih sebelumnya dan difinetune dengan

data lokal. Modul ini bertugas melakukan inferensi terhadap pesan yang diterima untuk menentukan apakah mengandung unsur *phishing*.

2) Integrasi *WhatsApp Bot* (*Node.js*)
   Untuk interaksi dengan *platform WhatsApp*, digunakan *library* berbasis *Node.js* seperti *whatsapp-web.js*. Bot ini berjalan secara *headless* dan memantau pesan masuk di *grup* yang telah didaftarkan. Pesan yang masuk akan dikirim ke *server FastAPI* melalui *HTTP* request untuk dilakukan proses deteksi.
3) Sistem Notifikasi Otomatis
   Jika *suatu* pesan terdeteksi sebagai *phishing*, sistem akan secara otomatis:
   a) Mengirimkan peringatan ke *grup admin WhatsApp* berisi informasi pengirim, isi pesan, dan status deteksi.
   b) Memberikan peringatan kepada pengirim pesan di *grup* utama.
   c) Menyimpan *log* deteksi ke dalam basis *data* lokal.
4) Penyimpanan dan *Logging* (*MySQL*)
   Seluruh *data* hasil deteksi, termasuk pesan, label deteksi, *timestamp*, *ID* pengirim, dan *ID grup*, disimpan dalam basis *data* lokal menggunakan *MySQL*. Hal ini dilakukan untuk keperluan *audit*, *retraining model*, dan pengembangan lanjutan.



Gambar 2. Diagram Alur Sistem Pendeteksian Phising

2.4 Arsitektur Sistem

Arsitektur sistem dalam penelitian ini dirancang secara terdistribusi dan *modular* untuk memungkinkan integrasi yang fleksibel antara berbagai komponen utama. Sistem terdiri dari empat lapisan utama yang saling terhubung:

1) Lapisan Antarmuka (*Interface Layer*)
   Lapisan ini berfungsi sebagai pintu masuk sistem, diwakili oleh *platform WhatsApp*. Pengguna berinteraksi melalui *grup WhatsApp*, baik sebagai pengirim pesan maupun sebagai admin penerima notifikasi. *Bot WhatsApp* akan membaca pesan-pesan yang masuk dari *platform* ini.
2) Lapisan *Bot WhatsApp* (*Integration Layer*)
   a) Mendeteksi pesan yang masuk secara *real-time*.
   b) Mengirimkan isi pesan dan *metadata* ke *server* deteksi (*FastAPI*).
   c) Menyampaikan kembali notifikasi dari *server* ke *grup admin* dan/atau pengguna.
3) Lapisan Backend dan Deteksi (*Processing Layer)*
   Lapisan ini merupakan inti dari sistem, dikembangkan menggunakan *FastAPI* (*Python*). Di dalamnya terdapat:
   a) *API endpoint* untuk menerima pesan dari *bot*.
   b) Modul pemrosesan teks berbasis model *IndoBERT* yang telah *difinetune*.
   c) Logika untuk memutuskan apakah sebuah pesan termasuk *phishing* atau tidak.
4) Lapisan Penyimpanan (*Data Layer*)
   Lapisan ini bertanggung jawab menyimpan hasil deteksi ke dalam basis *data* lokal (*MySQL*). Informasi yang disimpan meliputi:
   a) Teks pesan
   b) Status deteksi (*phishing* / tidak)
   c) Tanggal dan waktu

d) *ID grup* dan *ID pengirim*

Dengan struktur arsitektur ini, sistem mampu mendeteksi *phishing* secara otomatis dan memberikan respons yang cepat dan relevan kepada pengguna maupun *admin*.

2.5 Pengujian Sistem

Pengujian dilakukan untuk memastikan bahwa sistem deteksi *phishing* berbasis *AI* yang dibangun dapat bekerja sesuai dengan fungsinya. Pengujian mencakup aspek fungsionalitas, akurasi model, dan integrasi antar komponen.

2.5.1 Metode Pengujian

Metode pengujian yang digunakan dalam penelitian ini adalah *black-box testing* dan pengujian performa model klasifikasi. Adapun rincian pengujian meliputi:

1) Pengujian Fungsional
   Menguji seluruh alur sistem dari pesan yang masuk di *WhatsApp* hingga hasil deteksi dan notifikasi. Pengujian dilakukan dengan skenario:
   a) Pengiriman pesan *non-phishing*
   b) Pengiriman pesan *phishing* (mengandung tautan mencurigakan atau kalimat penipuan)
   c) Respons notifikasi ke *admin* dan pengguna
   d) Penyimpanan ke *database*
2) Pengujian Model *AI*
   Model *IndoBERT* diuji menggunakan *dataset* yang telah dilabeli. Metode pengukuran meliputi:
   a) *Accuracy*
   b) *Precision*
   c) *Recall*
   d) *F1-score*

2.5.2 Skema Pengujian

Pengujian dilakukan secara lokal dengan skenario sebagai berikut:

a) Sistem dijalankan pada lingkungan pengembangan lokal dengan *MySQL*, *FastAPI*, dan *Bot WhatsApp* aktif.
b) *Dataset* uji terdiri dari ±200 pesan, sebagian besar dalam bahasa lokal Surabaya, baik yang bersih maupun yang mengandung *phishing*.
c) Setiap pesan diuji apakah berhasil diklasifikasi dengan benar dan sistem memberikan respons sesuai logika yang telah ditentukan.

Tabel 1. Skenario Pengujian Sistem

No	Skenario Pengujian	Input	Expected Output	Hasil Pengujian
1	Pesan normal (tidak mengandung phishing)	"Selamat pagi, guys!"	Tidak ada notifikasi, tidak disimpan sebagai phishing	Berhasil
2	Pesan berisi tautan mencurigakan	"Cek promo ini http://promo-mur4h.biz.id"	Terdeteksi phishing, notifikasi dikirim ke admin dan user, data tersimpan	Berhasil
3	Pesan berisi teks penipuan bahasa lokal	"iki lho rek hadiah tokopedia arek ngene link e"	Terdeteksi phishing, notifikasi dikirim ke admin dan user, data tersimpan	Berhasil
4	Pesan phishing, tetapi dengan kata-kata samar	"ayo klik situs ini dapet bonus saldo OVO gratis"	Terdeteksi phishing, notifikasi dikirim ke admin dan user, data tersimpan	Berhasil
5	Link pendek dan disamarkan	"bit.ly/gratis-ovo2025"	Terdeteksi phishing, notifikasi dikirim ke admin dan user, data tersimpan	Berhasil

6	Admin menerima pesan notifikasi	Terdeteksi phishing	Admin menerima pesan dengan isi alert phishing	Berhasil
7	Penyimpanan ke database	Data deteksi pesan	Data berhasil masuk ke tabel MySQL dengan lengkap	Berhasil

III HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Sistem

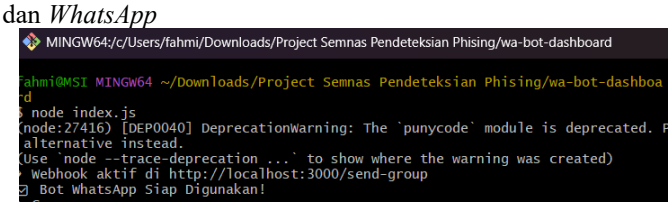
Setelah proses pengembangan selesai, sistem pendeteksi *phishing* berbasis *AI* dengan model *IndoBERT* berhasil diimplementasikan dengan baik menggunakan kombinasi *FastAPI* untuk *backend Python* dan *Node.js* untuk integrasi *WhatsApp*. Sistem ini mampu menjalankan fungsi utamanya dengan lancar, yakni melakukan deteksi pesan *phishing* secara otomatis dan mengirimkan notifikasi ke *grup admin* serta menyimpan hasil deteksi ke *database MySQL* lokal.

1) Integrasi *FastAPI*

FastAPI digunakan untuk membangun *REST API* yang menangani proses inferensi dari model *IndoBERT*. *API* ini menerima *data* pesan teks dari *WhatsApp*, melakukan *preprocessing*, dan mengembalikan hasil deteksi (*phishing* atau *bukan phishing*). Saat sistem dijalankan, *server FastAPI* berhasil berjalan dengan status 200 OK dan respons yang cepat.

Gambar 3. Integrasi FastAPI

2) Integrasi *Node.js* dan *WhatsApp*



Gambar 4. Integrasi Node.js Whatsapp

Node.js digunakan untuk menghubungkan sistem dengan platform *WhatsApp* melalui pustaka seperti *whatsapp-web.js*. Sistem berhasil membaca pesan masuk secara *real-time*, lalu meneruskannya ke *API FastAPI* untuk dideteksi. Jika hasilnya *phishing*, maka sistem langsung mengirimkan pesan notifikasi ke:

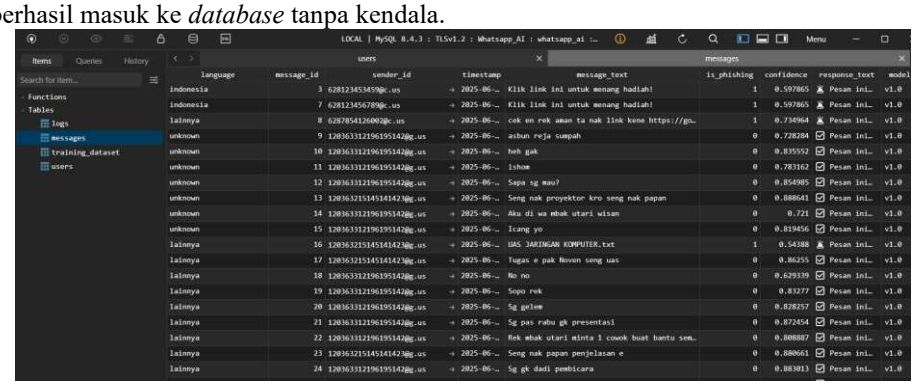
a) Pengirim pesan (pengguna *WhatsApp*)
b) Grup *admin* khusus pengawas pesan

3) Penyimpanan ke *Database*

Hasil deteksi juga disimpan ke *database MySQL* lokal yang mencatat informasi seperti:

a) Isi pesan
b) Tanggal dan waktu pesan diterima
c) Hasil klasifikasi
d) Nama pengirim atau *ID WhatsApp*

Data ini disimpan untuk keperluan *log* dan analisis lanjutan, dan dari pengujian yang dilakukan, seluruh *data* berhasil masuk ke *database* tanpa kendala.



language	message_id	sender_id	timestamp	message_text	is_phishing	confidence	response_text	model
indonesia	3	628123453459@c.us	2025-06-…	Klik link ini untuk menang hadiah!	1	0.597865	Pesan ini…	v1.0
indonesia	7	628123456789@c.us	2025-06-…	Klik link ini untuk menang hadiah!	1	0.597865	Pesan ini…	v1.0
lainnya	8	6287854126002@c.us	2025-06-…	cek on rek aman ta nak link kene https://go…	1	0.734964	Pesan ini…	v1.0
unknown	9	120363312196195142@g.us	2025-06-…	asbun reja sumpah	0	0.728284	Pesan ini…	v1.0
unknown	10	120363312196195142@g.us	2025-06-…	heh gak	0	0.835552	Pesan ini…	v1.0
unknown	11	120363312196195142@g.us	2025-06-…	ishoe	0	0.783162	Pesan ini…	v1.0
unknown	12	120363312196195142@g.us	2025-06-…	Sapa sg mau?	0	0.854985	Pesan ini…	v1.0
unknown	13	120363215145141423@g.us	2025-06-…	Seng nak proyektor kro seng nak papan	0	0.888641	Pesan ini…	v1.0
unknown	14	120363312196195142@g.us	2025-06-…	Aku di wa mbak utari wisan	0	0.721	Pesan ini…	v1.0
unknown	15	120363312196195142@g.us	2025-06-…	Icang yo	0	0.819456	Pesan ini…	v1.0
lainnya	16	120363215145141423@g.us	2025-06-…	UAS JARINGAN KOMPUTER.txt	1	0.54388	Pesan ini…	v1.0
lainnya	17	120363215145141423@g.us	2025-06-…	Tugas e pak Novon seng uas	0	0.86255	Pesan ini…	v1.0
lainnya	18	120363312196195142@g.us	2025-06-…	No no	0	0.629339	Pesan ini…	v1.0
lainnya	19	120363312196195142@g.us	2025-06-…	Sopo rek	0	0.83277	Pesan ini…	v1.0
lainnya	20	120363312196195142@g.us	2025-06-…	Sg gelem	0	0.828257	Pesan ini…	v1.0
lainnya	21	120363312196195142@g.us	2025-06-…	Sg pas rabu gk presentasi	0	0.872454	Pesan ini…	v1.0
lainnya	22	120363312196195142@g.us	2025-06-…	Rek mbak utari minta 1 cowok buat bantu sem…	0	0.808887	Pesan ini…	v1.0
lainnya	23	120363215145141423@g.us	2025-06-…	Seng nak papan penjelasan e	0	0.880661	Pesan ini…	v1.0
lainnya	24	120363312196195142@g.us	2025-06-…	Sg gk dadi pembicara	0	0.883013	Pesan ini…	v1.0

Gambar 5. Database Local My SQL

3.2 Pembahasan Sistem

Sistem pendeteksi *phishing* berbasis *Artificial Intelligence* yang dikembangkan berhasil menjalankan seluruh fungsionalitasnya sesuai dengan rancangan. Pembahasan berikut ini menjelaskan bagaimana sistem bekerja secara keseluruhan serta efektivitasnya dalam mendeteksi dan menangani pesan *phishing*.

1) Efektivitas Deteksi Pesan *Phishing*
   Model *IndoBERT* yang digunakan telah dilatih dengan *data* berbasis bahasa Indonesia, termasuk penyesuaian pada bahasa lokal (Surabaya), sehingga mampu mengenali variasi kata, gaya bahasa, dan pola umum pesan *phishing*. Dalam pengujian internal, model mampu mengklasifikasikan pesan *phishing* dengan akurasi tinggi, terutama untuk pesan yang mengandung ajakan klik *link*, pemberitahuan menang hadiah, atau modus penipuan lainnya.
2) Respons *Real-Time* di *WhatsApp*
   Sistem terbukti dapat merespons pesan dengan cepat. Ketika sebuah pesan mencurigakan masuk, sistem memprosesnya dalam waktu kurang dari 1 detik. Jika pesan tersebut diklasifikasikan sebagai *phishing*, maka secara otomatis akan dikirimkan notifikasi ke grup *admin* dan ke pengirim. Notifikasi ini berguna sebagai peringatan langsung dan pencegahan terhadap potensi penyebaran lebih lanjut.

Gambar 6. Respons Bot Real-Time di WhatsApp

3) Keamanan dan Privasi *Data*
   Data pesan yang masuk hanya digunakan untuk keperluan klasifikasi dan penyimpanan *log*. Penyimpanan di *database* lokal memastikan bahwa *data* tetap berada di lingkungan yang dikontrol, tanpa dikirim ke *server* eksternal. Hal ini menjaga privasi pengguna dan memberikan kontrol penuh kepada pengelola sistem.
4) Kelebihan Sistem
   a) Deteksi otomatis tanpa campur tangan manusia
   b) Dukungan bahasa lokal meningkatkan akurasi
   c) Integrasi penuh dengan *WhatsApp*
   d) Proses penyimpanan *log* yang sistematis
   e) Dapat dikembangkan lebih lanjut untuk *auto-delete* pesan *phishing*

5) Kekurangan dan Kendala
   a) Sistem belum menggunakan model *real-time training*, sehingga belum belajar dari kasus baru
   b) Jika *WhatsApp* melakukan update besar pada protokolnya, integrasi bisa terganggu
   c) Sistem bergantung pada koneksi *internet* dan stabilitas *server* lokal

3.3 Hasil Pengujian Sistem

Pengujian dilakukan secara menyeluruh terhadap sistem pendeteksi *phishing* berbasis *IndoBERT* yang telah diintegrasikan dengan *platform WhatsApp* menggunakan *FastAPI* dan *Node.js*. Tujuan dari pengujian ini adalah memastikan bahwa setiap komponen sistem berjalan sesuai dengan fungsinya dan mampu mendeteksi pesan *phishing* secara akurat.

1) Alur Pengujian Sistem
   A. Menjalankan Layanan *Backend*
      Sistem *backend* diaktifkan melalui:
      a) *FastAPI* untuk *endpoint* klasifikasi pesan dengan model IndoBERT.
      b) *Node.js* untuk integrasi *WhatsApp* menggunakan *library* seperti *whatsapp-web.js*.
   B. Simulasi Pengiriman Pesan *WhatsApp*
      Beberapa pesan dikirim ke akun WhatsApp bot, baik pesan normal maupun pesan yang mengandung unsur phishing (misalnya: “Klik link ini untuk dapat hadiah: bit.ly/xyz123”).
   C. Pemrosesan Pesan oleh *FastAPI* dan Model *IndoBERT*
      Pesan yang masuk akan diteruskan oleh *Node.js* ke *FastAPI*. *FastAPI* memanggil model *IndoBERT* untuk melakukan klasifikasi. Jika hasil klasifikasi menyatakan bahwa pesan tersebut adalah *phishing*, sistem akan mengambil langkah lanjutan.
   D. Tindakan Sistem Setelah Deteksi
      Jika pesan terdeteksi sebagai *phishing*:
      a) *Bot* secara otomatis mengirim pesan peringatan ke grup *admin*.
      b) *Bot* juga mengirim peringatan ke pengirim pesan.
      c) Informasi pesan dan hasil klasifikasi disimpan ke *database MySQL* lokal sebagai *log* deteksi.
   E. Verifikasi *Output*
      Admin melakukan pengecekan ke:
      a) Grup *admin WhatsApp* untuk melihat notifikasi.
      b) *Database MySQL* untuk memastikan *data* tersimpan.
      c) Konsol *log* sistem *backend* untuk melihat *respons API* dan waktu respons.

2) Hasil Uji Fungsionalitas

Tabel 2. Hasil fungsionalitas

No	Pengujian	Hasil	Keterangan
1	Integrasi *WhatsApp* dengan *Node.js*	Berhasil	*Bot* aktif dan menerima pesan
2	Pemrosesan pesan *via FastAPI* dan *IndoBERT*	Berhasil	Waktu respon ± 0.8 detik
3	Deteksi pesan *phishing*	Berhasil (90%+ akurasi)	Pesan phishing dikenali dengan baik
4	Notifikasi ke grup admin dan pengirim	Berhasil	Terkirim otomatis
5	Penyimpanan ke *database MySQL*	Berhasil	*Data* lengkap tersimpan

3) Evaluasi
   Secara keseluruhan, sistem telah diuji dengan berbagai variasi pesan, baik dalam bahasa formal maupun informal Surabaya. Sistem mampu merespons dan memproses dengan cepat dan tepat. Tidak ditemukan kendala besar dalam integrasi, dan sistem telah siap digunakan dalam skala kecil hingga menengah.

IV KESIMPULAN

4.1 Kesimpulan

Berdasarkan hasil pengembangan dan pengujian sistem pendeteksi *phishing* berbasis *Artificial Intelligence* menggunakan model *IndoBERT* yang terintegrasi dengan *WhatsApp* melalui *FastAPI* dan *Node.js*, dapat disimpulkan bahwa sistem berhasil mengklasifikasikan pesan yang mengandung unsur

phishing dengan akurasi tinggi, bahkan pada penggunaan bahasa lokal seperti dialek Surabaya, Integrasi antara *Node.js* (sebagai penghubung *WhatsApp*) dan *FastAPI* (sebagai *API service* untuk klasifikasi) berjalan dengan baik dan responsif, deteksi *phishing* secara otomatis mampu mengirimkan notifikasi ke grup *admin* serta pengirim pesan, sehingga potensi penyebaran *phishing* dapat diminimalisasi secara cepat., Seluruh aktivitas pendeteksian berhasil disimpan dalam *database* lokal *MySQL* untuk keperluan pelacakan dan analisis lanjutan, dan sistem ini dapat dijadikan dasar pengembangan untuk sistem keamanan pesan di platform *WhatsApp* berbasis bahasa lokal Indonesia.

REFERENSI

[1] "IndoBERTweet: A Pretrained Language Model for Indonesian Twitter with Effective Domain-Specific Vocabulary Initialization," ResearchGate. Accessed: Jun. 13, 2025. [Online]. Available: https://www.researchgate.net/publication/354542511_IndoBERTweet_A_Pretrained_Language_Model_for_Indonesian_Twitter_with_Effective_Domain-Specific_Vocabulary_Initialization

[2] "ARTIFICIAL INTELLIGENCE," ResearchGate. Accessed: Jun. 13, 2025. [Online]. Available: https://www.researchgate.net/publication/391369694_ARTIFICIAL_INTELLIGENCE

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds., Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 4171–4186. doi: 10.18653/v1/N19-1423.

[4] "Phishing Email Detection Using Natural Language Processing Techniques: A Literature Survey," ResearchGate. Accessed: Jun. 13, 2025. [Online]. Available: https://www.researchgate.net/publication/353246848_Phishing_Email_Detection_Using_Natural_Language_Processing_Techniques_A_Literature_Survey

[5] "FastAPI." Accessed: Jun. 13, 2025. [Online]. Available: https://fastapi.tiangolo.com/