BAB 2
LANDASAN TEORI

2.1 *Phishing*

Phishing merupakan salah satu bentuk serangan rekayasa sosial yang bertujuan untuk memperoleh informasi sensitif dari korban, seperti kata sandi, data pribadi, informasi akun, maupun informasi finansial. Serangan ini dilakukan dengan cara menyamar sebagai pihak yang dipercaya oleh korban, misalnya institusi keuangan, layanan digital, perusahaan, atau organisasi resmi [1]. Dalam praktiknya, *phishing* tidak hanya memanfaatkan kelemahan teknis, tetapi juga memanfaatkan faktor psikologis manusia agar korban terdorong untuk mengikuti instruksi penyerang [6]. Serangan *phishing* umumnya dilakukan dengan membuat pesan, halaman web, atau tautan yang tampak sah. Korban diarahkan untuk membuka tautan, mengisi formulir, memperbarui akun, atau memasukkan kredensial pada halaman palsu. Oleh karena itu, *phishing* menjadi ancaman yang berbahaya karena korban sering kali tidak menyadari bahwa pesan atau tautan yang diterima sebenarnya berasal dari pihak yang tidak sah.

2.2 *Email phishing*

Email phishing merupakan email palsu yang dibuat seolah-olah berasal dari pihak resmi atau tepercaya [7]. Email ini biasanya berisi instruksi agar penerima membuka lampiran, menekan tautan tertentu, memperbarui informasi akun, atau memasukkan data sensitif pada halaman web palsu. *Email phishing* sering menggunakan gaya bahasa yang meyakinkan, nama organisasi resmi, serta unsur urgensi agar korban segera mengikuti instruksi tanpa memeriksa keaslian pesan.
Karakteristik *email phishing* dapat terlihat dari beberapa komponen, seperti alamat pengirim, *subject*, isi pesan, penggunaan kata-kata mencurigakan, keberadaan tautan, serta *domain* yang digunakan [8]. *Email phishing* juga dapat memanfaatkan kalimat yang mengandung tekanan psikologis, seperti ancaman penutupan akun, permintaan verifikasi segera, pemberitahuan keamanan palsu, atau hadiah yang harus segera diklaim.

2.3 *Link phishing*

Link phishing merupakan tautan yang digunakan untuk mengarahkan korban ke halaman web palsu atau berbahaya [3]. *Link phishing* sering dibuat menyerupai URL resmi agar terlihat meyakinkan. Teknik yang umum digunakan antara lain penggunaan domain tiruan, *typosquatting*, tanda hubung, angka, *subdomain* panjang, karakter khusus, atau *path* yang mengandung kata seperti *login*, *verify*, *secure*, *account*, dan *update*.

URL *phishing* dapat sulit dikenali karena sebagian URL *legitimate* juga memiliki struktur yang kompleks [9]. Misalnya, URL resmi dapat memiliki path seperti */login*, */account*, atau */security*. Oleh karena itu, deteksi *link phishing* tidak cukup hanya melihat satu kata tertentu, tetapi perlu mempertimbangkan struktur URL secara menyeluruh, seperti *domain*, *subdomain*, panjang URL, penggunaan HTTPS, karakter khusus, dan pola *token* di dalam URL.

2.4 Klasifikasi

Klasifikasi merupakan salah satu teknik dalam *supervised learning* yang bertujuan untuk mengelompokkan data ke dalam kategori tertentu berdasarkan pola atau karakteristik yang dimiliki [10]. Pada klasifikasi biner, data dikelompokkan ke dalam dua kelas. Dalam penelitian ini, klasifikasi dilakukan untuk membedakan data *legitimate* dan *phishing*.

Secara umum, proses klasifikasi terdiri dari tahap *training* dan *testing*. Pada tahap *training*, model mempelajari pola dari data berlabel. Setelah model selesai dilatih, tahap *testing* dilakukan untuk mengevaluasi kemampuan model dalam memprediksi data baru yang tidak digunakan selama proses *training*.

Secara matematis, fungsi klasifikasi dapat dituliskan sebagai berikut:

$$f : X \rightarrow Y \tag{2.1}$$

Pada persamaan tersebut, X merupakan ruang fitur atau data masukan, sedangkan Y merupakan label kelas. Dalam penelitian ini, label kelas terdiri dari dua nilai, yaitu 0 untuk *legitimate* dan 1 untuk *phishing*.

2.5 BERT

Bidirectional Encoder Representations from Transformers atau BERT merupakan model bahasa berbasis *Transformer* yang dirancang untuk memahami konteks kata secara dua arah [5]. Kemampuan *bidirectional* membuat BERT dapat mempertimbangkan konteks kiri dan kanan secara bersamaan ketika menghasilkan representasi teks.

BERT menerima *input* dalam bentuk *token*. Pada proses tokenisasi, teks diubah menjadi urutan *token* yang dapat diproses oleh model. *Input* BERT umumnya diawali dengan *token* khusus [CLS] dan diakhiri dengan *token* [SEP]. *Token* [CLS] sering digunakan sebagai representasi keseluruhan *input* dalam tugas klasifikasi teks.

Arsitektur BERT memanfaatkan mekanisme *self-attention* untuk mempelajari hubungan antar-*token* dalam sebuah urutan teks [11]. Dalam konteks deteksi *phishing*, kemampuan ini penting karena model perlu memahami pola bahasa, instruksi mencurigakan, kata-kata yang menunjukkan urgensi, serta struktur URL yang dapat menjadi indikator *phishing*.

2.6 Multilingual BERT

Multilingual BERT atau mBERT merupakan varian BERT yang dilatih pada data multibahasa [5]. Model ini mampu memproses teks dari berbagai bahasa, termasuk bahasa Inggris dan bahasa Indonesia. Pada penelitian ini, model yang digunakan adalah `bert-base-multilingual-cased`. Penggunaan model berbasis BERT untuk deteksi *phishing* telah diterapkan pada penelitian sebelumnya karena model ini mampu mempelajari representasi teks secara kontekstual [12].

Pemilihan mBERT dilakukan karena *email phishing* dan contoh *input* manual dapat mengandung lebih dari satu bahasa. *Email phishing* dapat ditulis dalam bahasa Inggris, bahasa Indonesia, atau campuran keduanya. Dengan menggunakan mBERT, model diharapkan memiliki kemampuan yang lebih fleksibel dalam memproses variasi bahasa pada data email dan URL.

2.7 *Hybrid* mBERT

Hybrid mBERT mengacu pada penggabungan representasi teks dari mBERT dengan fitur numerik yang diekstraksi dari email dan URL. Dengan pendekatan

ini, model tidak hanya mempelajari konteks teks, tetapi juga mempertimbangkan karakteristik struktural yang dapat menjadi indikator *phishing*.

Pada model BERT biasa, masukan model hanya berupa teks yang telah ditokenisasi. Sementara itu, pada model *hybrid* mBERT, terdapat dua jalur pemrosesan. Jalur pertama memproses teks menggunakan mBERT, sedangkan jalur kedua memproses fitur numerik menggunakan *feature layer*. Keluaran dari kedua jalur tersebut kemudian digabungkan sebelum masuk ke *classification layer*.

Secara umum, representasi teks dari mBERT dapat dituliskan sebagai berikut:

$$h_{text} = \text{mBERT}(x) \tag{2.2}$$

Pada persamaan tersebut, x merupakan *input* teks yang telah melalui tokenisasi, sedangkan h_{text} merupakan representasi teks yang dihasilkan oleh mBERT. Fitur numerik dinyatakan sebagai vektor v dan diproses melalui *feature layer* sebagai berikut:

$$h_{feature} = f(W_f v + b_f) \tag{2.3}$$

Pada persamaan tersebut, W_f merupakan bobot pada *feature layer*, b_f merupakan bias, dan f merupakan fungsi aktivasi non-linear. Representasi teks dan fitur numerik kemudian digabungkan menggunakan *concatenation*:

$$h_{hybrid} = [h_{text}; h_{feature}] \tag{2.4}$$

Representasi gabungan tersebut digunakan sebagai masukan ke *classification layer*:

$$z = W_c h_{hybrid} + b_c \tag{2.5}$$

Pada persamaan tersebut, W_c merupakan bobot *classification layer*, b_c merupakan bias, dan z merupakan logit prediksi. Logit tersebut kemudian digunakan untuk menentukan kelas *legitimate* atau *phishing*.

Pendekatan *hybrid* mBERT pada penelitian ini menggabungkan dua jenis representasi, yaitu representasi teks dan representasi fitur numerik. Representasi teks digunakan untuk menangkap informasi kontekstual dan pola bahasa yang

terdapat pada email atau URL, sedangkan fitur numerik digunakan untuk merepresentasikan karakteristik yang dapat dihitung secara langsung dari struktur data.

Representasi teks diperoleh dari model `bert-base-multilingual-cased`. Teks yang telah melalui proses tokenisasi dimasukkan ke dalam mBERT untuk menghasilkan representasi kontekstual. Representasi tersebut dapat memuat informasi mengenai hubungan antar-token, pola bahasa, kemunculan kata tertentu, dan urutan karakter yang terdapat pada masukan.

Fitur numerik diproses melalui jalur yang berbeda menggunakan *feature layer*. Jalur fitur numerik bertujuan mengubah sekumpulan nilai numerik menjadi representasi yang dapat digabungkan dengan keluaran mBERT. Keluaran dari jalur teks dan jalur fitur numerik kemudian digabungkan menggunakan operasi *concatenation*. Hasil penggabungan tersebut diteruskan ke *dropout layer* dan *classification layer* untuk menghasilkan prediksi kelas *legitimate* atau *phishing*. Dengan demikian, istilah *hybrid* pada penelitian ini tidak berarti menggabungkan dataset email dan dataset URL menjadi satu dataset. Istilah tersebut mengacu pada penggabungan dua bentuk informasi dalam satu model, yaitu representasi teks dari mBERT dan representasi fitur numerik.

2.8 Analisis Tekstual dan Analisis Struktural

Analisis tekstual merupakan proses untuk mempelajari informasi yang terkandung dalam bentuk teks. Pada penelitian ini, analisis tekstual dilakukan menggunakan mBERT. Model tersebut memproses token-token dari teks email atau URL untuk menghasilkan representasi kontekstual. Melalui analisis tekstual, model dapat mempelajari pola bahasa, hubungan antarkata, kemunculan kata mencurigakan, serta pola token yang berpotensi menjadi indikator *phishing*.

Analisis struktural merupakan proses untuk mempelajari karakteristik data yang dapat dihitung secara langsung. Karakteristik tersebut tidak berfokus pada makna bahasa, tetapi pada bentuk atau susunan data. Contohnya adalah panjang domain, jumlah URL, keberadaan angka, penggunaan tanda hubung, jumlah subdomain, jumlah karakter khusus, dan nilai *entropy*.

Pada skenario email, analisis tekstual digunakan untuk mempelajari isi dan komponen teks email, sedangkan analisis struktural digunakan untuk mempelajari karakteristik pengirim, subjek, isi pesan, dan keberadaan URL. Pada skenario URL, analisis tekstual digunakan untuk mempelajari pola token pada URL, sedangkan

analisis struktural digunakan untuk menganalisis susunan domain, subdomain, karakter khusus, angka, protokol, dan nilai *entropy*.

Kedua analisis tersebut saling melengkapi. Analisis tekstual memberikan informasi mengenai konteks dan pola token, sedangkan analisis struktural memberikan informasi kuantitatif mengenai bentuk data. Penggabungan keduanya digunakan agar model tidak hanya bergantung pada isi teks, tetapi juga mempertimbangkan karakteristik struktural yang dapat menjadi indikator *phishing*.

2.9 Fitur Teks dan Fitur Numerik pada Skenario Email dan URL

Penelitian ini menggunakan fitur teks dan fitur numerik pada dua skenario klasifikasi yang terpisah, yaitu klasifikasi email dan klasifikasi URL.

Pada skenario email, fitur teks dibentuk dari komponen *sender*, *receiver*, *subject*, *body*, URL, domain pengirim, domain penerima, serta kata-kata mencurigakan yang terdapat pada email. Seluruh komponen tersebut disusun menjadi teks terstruktur sebelum diproses menggunakan *tokenizer* mBERT.

Fitur numerik email digunakan untuk merepresentasikan karakteristik yang dapat dihitung dari komponen email, seperti karakteristik domain pengirim, jumlah URL, panjang subjek, jumlah kata mencurigakan, panjang isi pesan, jumlah tanda seru, serta rasio huruf kapital. Fitur numerik membantu model mengenali pola struktural yang mungkin tidak dapat dipahami secara langsung hanya melalui representasi teks.

Pada skenario URL, fitur teks berupa URL yang telah dinormalisasi dan diproses sebagai urutan token menggunakan mBERT. Representasi teks URL digunakan untuk mempelajari pola token, susunan kata, nama domain, subdomain, dan bagian *path* yang terdapat dalam URL.

Fitur numerik URL digunakan untuk merepresentasikan struktur URL secara kuantitatif. Fitur tersebut mencakup panjang URL, panjang domain, penggunaan HTTPS, keberadaan alamat IP, karakter @, tanda hubung, jumlah subdomain, jumlah angka, jumlah karakter khusus, jumlah kata sensitif, nilai *domain entropy*, dan nilai *URL entropy*. Dengan demikian, model URL memanfaatkan informasi kontekstual dari teks URL sekaligus informasi struktural dari fitur numeriknya.

2.10 Fitur Numerik

Fitur numerik merupakan fitur yang dinyatakan dalam bentuk angka dan diperoleh dari karakteristik suatu data. Fitur ini digunakan untuk merepresentasikan informasi yang dapat diukur secara kuantitatif, seperti panjang teks, jumlah karakter tertentu, jumlah kata, jumlah simbol, atau nilai statistik tertentu. Dalam proses klasifikasi, fitur numerik dapat membantu menggambarkan pola yang terdapat pada data secara lebih terstruktur.

Fitur numerik berbeda dengan fitur tekstual karena tidak merepresentasikan makna bahasa secara langsung, melainkan merepresentasikan ciri-ciri terukur dari data. Sebagai contoh, pada data berbentuk teks, fitur numerik dapat berupa panjang kalimat, jumlah kata, jumlah huruf kapital, jumlah tanda baca, atau jumlah kata tertentu. Pada data berbentuk URL, fitur numerik dapat berupa panjang URL, panjang *domain*, jumlah *subdomain*, jumlah angka, jumlah karakter khusus, penggunaan HTTPS, keberadaan *IP address*, dan nilai *entropy*.

Dalam konteks deteksi *phishing* URL, fitur numerik dapat merepresentasikan karakteristik struktural URL seperti panjang URL, *domain*, *subdomain*, dan pola karakter tertentu [3]. Karakteristik tersebut dapat digunakan untuk membedakan URL *legitimate* dan *URL phishing* karena *URL phishing* sering memiliki pola tertentu, seperti penggunaan domain tiruan, tanda hubung, angka, subdomain berlebih, atau kata-kata sensitif.

Dengan demikian, fitur numerik berperan sebagai representasi kuantitatif dari karakteristik data. Penggunaan fitur numerik memungkinkan proses klasifikasi mempertimbangkan aspek-aspek struktural yang dapat dihitung dan dibandingkan secara langsung.

2.11 Pemrosesan Fitur Numerik melalui *Feature Layer*

Fitur numerik tidak langsung digabungkan dengan keluaran mBERT karena nilai fitur memiliki karakteristik dan skala yang berbeda dari representasi teks. Oleh karena itu, fitur numerik terlebih dahulu dinormalisasi menggunakan StandardScaler. Proses normalisasi membuat fitur memiliki rata-rata mendekati 0 dan standar deviasi mendekati 1 sehingga tidak ada fitur dengan rentang nilai besar yang mendominasi proses pembelajaran.

Setelah dinormalisasi, vektor fitur numerik dimasukkan ke dalam *feature layer*. *Feature layer* mengubah 13 fitur numerik menjadi representasi laten yang dapat

dipelajari bersama dengan representasi teks. Secara umum, proses tersebut dapat dinyatakan sebagai:

$$h_{\text{feature}} = f(W_f v + b_f) \tag{2.6}$$

Pada persamaan tersebut, v merupakan vektor fitur numerik, W_f merupakan bobot pada *feature layer*, b_f merupakan bias, dan f merupakan fungsi aktivasi non-linear. *Feature layer* mengubah 13 fitur numerik menjadi representasi laten yang dapat dipelajari bersama dengan representasi teks.

Pada jalur teks, mBERT menghasilkan representasi teks h_{text}. Keluaran dari jalur teks kemudian digabungkan dengan keluaran *feature layer* menggunakan operasi *concatenation*. Representasi gabungan tersebut diteruskan ke *dropout layer* untuk mengurangi risiko *overfitting*, kemudian masuk ke *classification layer*:

$$h_{\text{hybrid}} = [h_{\text{text}}; h_{\text{feature}}] \tag{2.7}$$

Representasi gabungan tersebut diteruskan ke *dropout layer* untuk mengurangi risiko *overfitting*, kemudian masuk ke *classification layer*. Lapisan klasifikasi menghasilkan dua nilai *logit* yang merepresentasikan kelas *legitimate* dan *phishing*. Dalam arsitektur model, *feature layer* merupakan jalur kedua yang berjalan sejajar dengan jalur mBERT. Jalur mBERT memproses teks, sedangkan jalur *feature layer* memproses fitur numerik. Kedua jalur baru digabungkan sebelum tahap klasifikasi akhir.

2.12 *Entropy*

Entropy merupakan ukuran yang dapat digunakan untuk menggambarkan tingkat ketidakteraturan atau variasi karakter dalam sebuah *string* [13]. Dalam konteks deteksi *phishing*, *entropy* dapat digunakan untuk menganalisis *domain* atau URL. *Domain* atau URL yang memiliki pola karakter acak dapat memiliki nilai *entropy* yang lebih tinggi.

Secara umum, *entropy* dapat dirumuskan sebagai berikut:

$$H(X) = -\sum_{i=1}^{n} p(x_i) \log_2 p(x_i) \tag{2.8}$$

Pada persamaan tersebut, $H(X)$ merupakan nilai *entropy*, $p(x_i)$ merupakan probabilitas kemunculan karakter ke-i, dan n merupakan jumlah karakter unik. Semakin tinggi nilai *entropy*, semakin besar variasi karakter pada *string* tersebut.

2.13 *Data Augmentation*

Data augmentation merupakan teknik untuk menambah variasi data latih dengan membuat contoh data tambahan yang masih sesuai dengan karakteristik kelas tertentu [14]. Teknik ini digunakan untuk memperkaya data *training* agar model dapat mengenali pola yang lebih bervariasi.

2.14 *Data Leakage*

Pencegahan *data leakage* penting dilakukan dengan memastikan proses transformasi seperti normalisasi hanya di-*fit* pada data *training* [15]. Kondisi ini dapat menyebabkan hasil evaluasi menjadi tidak valid karena model secara tidak langsung memperoleh informasi dari data yang seharusnya hanya digunakan untuk pengujian.

Pencegahan *data leakage* penting karena evaluasi model harus mencerminkan kemampuan model terhadap data yang benar-benar belum pernah dilihat. Jika *data leakage* terjadi, nilai *accuracy*, *precision*, *recall*, dan *F1-score* dapat terlihat tinggi tetapi tidak mencerminkan kemampuan generalisasi model yang sebenarnya.

2.15 StandardScaler

StandardScaler merupakan metode normalisasi fitur numerik yang mengubah nilai fitur agar memiliki rata-rata 0 dan standar deviasi 1 [15]. Normalisasi ini penting karena fitur numerik dapat memiliki skala yang berbeda-beda. Misalnya, panjang URL dapat bernilai ratusan, sedangkan fitur indikator seperti `has_https` hanya bernilai 0 atau 1.

Secara matematis, *standardization* dapat dirumuskan sebagai berikut:

$$z = \frac{x - \mu}{\sigma} \tag{2.9}$$

Pada persamaan tersebut, z merupakan nilai hasil standardisasi, x merupakan nilai fitur asli, μ merupakan rata-rata fitur pada data *training*, dan σ merupakan standar

deviasi fitur pada data *training*.

2.16 *Early Stopping*

Early stopping merupakan teknik untuk menghentikan *training* secara otomatis apabila performa model pada *validation set* tidak lagi membaik [16]. Teknik ini digunakan untuk mengurangi risiko *overfitting*, yaitu kondisi ketika model terlalu menyesuaikan diri terhadap data *training* sehingga performanya menurun pada data baru.

Penggunaan *early stopping* membuat model yang digunakan untuk evaluasi akhir bukan selalu model pada *epoch* terakhir. Model yang dipilih adalah *checkpoint* dengan performa *validation* terbaik. Dengan demikian, *early stopping* membantu menjaga agar model tidak terlalu menyesuaikan diri terhadap data *training*.

2.17 *Confusion Matrix*

Confusion matrix merupakan metode evaluasi yang digunakan untuk membandingkan hasil prediksi model dengan label aktual [17]. Pada klasifikasi biner, *confusion matrix* terdiri dari empat komponen utama, yaitu *true positive*, *true negative*, *false positive*, dan *false negative*.

True Positive merupakan data *phishing* yang berhasil diprediksi sebagai *phishing*, sedangkan *True Negative* merupakan data *legitimate* yang berhasil diprediksi sebagai *legitimate*.

False positive merupakan data *legitimate* yang salah diprediksi sebagai *phishing*. *False negative* merupakan data *phishing* yang salah diprediksi sebagai *legitimate*. Dalam konteks deteksi *phishing*, *false negative* perlu diperhatikan karena berarti email atau URL *phishing* tidak terdeteksi oleh model.

Secara umum, *confusion matrix* dapat dituliskan sebagai berikut:

$$CM = \begin{bmatrix} TN & FP \\ FN & TP \end{bmatrix} \tag{2.10}$$

2.18 *Accuracy*

Accuracy merupakan metrik evaluasi yang digunakan untuk mengukur proporsi prediksi benar terhadap seluruh data yang diuji [17]. Metrik ini memberikan

gambaran umum mengenai seberapa banyak data yang berhasil diklasifikasikan dengan benar oleh model.
Secara matematis, *accuracy* dirumuskan sebagai berikut:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.11}$$

Accuracy digunakan untuk melihat performa model secara keseluruhan. Namun, *accuracy* tidak digunakan sebagai satu-satunya metrik karena pada kasus keamanan, kesalahan pada kelas phishing memiliki dampak yang lebih serius. Metrik selain *accuracy* tersebut yaitu, *precision*, *recall*, dan *F1-score*.

2.19 *Precision*

Precision merupakan metrik evaluasi yang digunakan untuk mengukur seberapa banyak data yang diprediksi sebagai *phishing* benar-benar merupakan *phishing* [17]. Metrik ini penting untuk melihat ketepatan model ketika memberikan prediksi positif.
Secara matematis, *precision* dirumuskan sebagai berikut:

$$precision = \frac{TP}{TP + FP} \tag{2.12}$$

Nilai *precision* yang tinggi menunjukkan bahwa model jarang salah mengklasifikasikan data *legitimate* sebagai *phishing*. Dalam konteks deteksi *phishing*, *precision* penting agar sistem tidak terlalu sering memberikan peringatan palsu terhadap email atau URL *legitimate*.

2.20 *Recall*

Recall merupakan metrik evaluasi yang digunakan untuk mengukur kemampuan model dalam menemukan seluruh data *phishing* yang sebenarnya [17]. Metrik ini sangat penting dalam deteksi *phishing* karena *phishing* yang tidak terdeteksi dapat membahayakan pengguna.
Secara matematis, *recall* dirumuskan sebagai berikut:

$$recall = \frac{TP}{TP + FN} \tag{2.13}$$

Nilai *recall* yang tinggi menunjukkan bahwa model mampu mendeteksi sebagian besar data *phishing*.

2.21 *F1-score*

F1-score merupakan rata-rata harmonik antara *precision* dan *recall* [17]. Metrik ini digunakan untuk melihat keseimbangan antara ketepatan prediksi positif dan kemampuan model dalam menemukan seluruh data positif.
Secara matematis, *F1-score* dirumuskan sebagai berikut:

$$F1\text{-}score = 2 \times \frac{precision \times recall}{precision + recall} \tag{2.14}$$

F1-score digunakan dalam penelitian ini karena deteksi phishing membutuhkan keseimbangan antara *precision* dan *recall*. Model yang baik tidak hanya harus tepat ketika memprediksi *phishing*, tetapi juga harus mampu menemukan sebanyak mungkin data *phishing* yang sebenarnya.

2.22 Penelitian Terdahulu

Penelitian terdahulu merupakan kajian terhadap penelitian-penelitian sebelumnya yang memiliki keterkaitan dengan topik, metode, objek, atau permasalahan yang dibahas dalam penelitian ini. Kajian ini digunakan untuk memahami perkembangan pendekatan yang telah digunakan oleh peneliti sebelumnya, mengetahui metode yang relevan, serta menemukan ruang penelitian yang masih dapat dikembangkan. Dalam penelitian ini, penelitian terdahulu digunakan sebagai dasar untuk melihat penerapan model berbasis BERT, *Transformer*, fitur numerik, dan pendekatan *hybrid* dalam mendeteksi *email phishing*, *URL phishing*, maupun halaman web *phishing*.
Penelitian terdahulu yang dibahas pada bagian ini dipilih berdasarkan keterkaitannya dengan deteksi *phishing* menggunakan pendekatan *machine learning*, *deep learning*, BERT, *Transformer*, fitur URL, fitur HTML, dan integrasi beberapa jenis fitur. Beberapa penelitian terdahulu juga menekankan pentingnya pemilihan dataset dan analisis fitur dalam deteksi *phishing email*. Champa dkk. membahas penggunaan curated dataset dan analisis fitur untuk deteksi *phishing email* menggunakan *machine learning* [18].

Ringkasan penelitian terdahulu ditunjukkan pada Tabel 2.1.

Tabel 2.1. Penelitian Terdahulu

No	Penulis	Judul	Metode dan Kinerja	Keterbatasan	*Research Gap*
1	Otieno dkk. (2023)	The Application of the BERT *Transformer* Model for Phishing Email Classification [19]	BERT melalui proses *fine-tuning* untuk melakukan klasifikasi biner antara *email phishing* dan non-*phishing*. Hasil penelitian menunjukkan bahwa representasi kontekstual BERT dapat digunakan untuk membedakan email berdasarkan pola bahasa yang terkandung di dalamnya.	Penelitian berfokus pada representasi teks email menggunakan BERT dan belum menggabungkannya dengan fitur numerik yang secara khusus menggambarkan karakteristik domain pengirim, *subject*, *body*, dan URL. Penelitian tersebut juga tidak membahas klasifikasi URL sebagai skenario tersendiri.	Penelitian ini menggunakan pendekatan *hybrid* mBERT yang menggabungkan representasi teks dengan 13 fitur numerik email. Selain itu, penelitian ini mengembangkan skenario terpisah untuk mendeteksi *link phishing* berdasarkan representasi teks URL dan fitur numerik strukturnya.
2	Songailaite dkk. (2023)	BERT-Based Models for Phishing Detection [12]	DistilBERT, TinyBERT, dan RoBERTa melalui proses *fine-tuning* untuk mendeteksi *email phishing*. Seluruh model memperoleh nilai di atas 0,985 pada metrik *accuracy*, *precision*, *recall*, dan *F1-score*, dengan RoBERTa menghasilkan kinerja terbaik.	Penelitian hanya memanfaatkan informasi teks email melalui model berbasis BERT. Karakteristik struktural atau fitur numerik email belum diproses melalui jalur fitur tersendiri. Penelitian juga tidak mengevaluasi URL *phishing* sebagai objek klasifikasi terpisah.	Penelitian ini menggunakan mBERT untuk menghasilkan representasi teks dan menggabungkannya dengan fitur numerik email melalui *feature layer*. Pendekatan *hybrid* yang sama juga dievaluasi pada skenario klasifikasi URL secara terpisah.

Lanjut pada halaman berikutnya

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 2.1 Penelitian Terdahulu (lanjutan)

No	Penulis	Judul	Metode dan Kinerja	Keterbatasan	*Research Gap*
3	Melendez dkk. (2024)	Comparative Investigation of Traditional Machine-Learning Models and *Transformer* Models for Phishing Email Detection [20]	Penelitian membandingkan model *machine learning* tradisional dengan model berbasis *Transformer*, seperti BERT, DistilBERT, dan RoBERTa, menggunakan metrik *accuracy*, *precision*, *recall*, dan *F1-score*. Model berbasis *Transformer* menunjukkan kemampuan yang lebih baik dalam mempelajari konteks dan pola bahasa pada email.	Penelitian berorientasi pada perbandingan kinerja beberapa model untuk klasifikasi email. Penelitian tersebut belum membangun arsitektur *hybrid* yang memproses representasi teks dan fitur numerik secara bersamaan serta belum mengevaluasi URL *phishing* sebagai skenario terpisah.	Penelitian ini tidak hanya menggunakan representasi kontekstual mBERT, tetapi juga menggabungkannya dengan fitur numerik email sebelum proses klasifikasi. Arsitektur serupa diterapkan secara terpisah pada data URL untuk mengevaluasi kemampuan deteksi *link phishing*.
4	Ibrahim dan Elhafiz (2026)	Phishing Email Detection Using BERT and RoBERTa [21]	BERT dan RoBERTa melalui proses *fine-tuning* menggunakan gabungan Nazario Phishing Corpus dan email asli dari dataset Enron. Hasil pengujian menunjukkan bahwa RoBERTa memberikan kinerja yang lebih menyeluruh dan menghasilkan jumlah *false negative* yang lebih rendah dibandingkan BERT.	Penelitian berfokus pada analisis teks email menggunakan BERT dan RoBERTa. Keluaran model bahasa belum digabungkan dengan fitur numerik email melalui arsitektur *hybrid*. Penelitian juga tidak melakukan klasifikasi URL melalui model tersendiri.	Penelitian ini menggabungkan representasi teks mBERT dengan 13 fitur numerik email yang menggambarkan karakteristik pengirim, *subject*, *body*, dan URL. Selain itu, model *hybrid* juga dievaluasi pada dataset URL melalui eksperimen terpisah.

Lanjut pada halaman berikutnya

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 2.1 Penelitian Terdahulu (lanjutan)

No	Penulis	Judul	Metode dan Kinerja	Keterbatasan	*Research Gap*
5	Kumar (2024)	Detecting *URL Phishing* Using BERT and DistilBERT Classifiers [9]	BERT dan DistilBERT digunakan untuk melakukan klasifikasi URL *phishing*. Hasil penelitian menunjukkan bahwa DistilBERT memberikan kinerja yang lebih baik berdasarkan metrik *accuracy*, *precision*, *recall*, dan *F1-score*.	Penelitian berfokus pada representasi URL yang dipelajari oleh model *Transformer*. Fitur numerik yang secara eksplisit menggambarkan panjang URL, panjang domain, jumlah subdomain, karakter khusus, dan nilai *entropy* belum digabungkan melalui jalur pemrosesan fitur tersendiri. Penelitian juga tidak membahas klasifikasi email.	Penelitian ini menggabungkan representasi teks URL dari mBERT dengan 13 fitur numerik struktur URL. Selain klasifikasi URL, penelitian ini juga mengevaluasi model *hybrid* pada skenario deteksi *email phishing* secara terpisah.
6	Elsadig dkk. (2022)	Intelligent Deep Machine Learning Cyber Phishing URL Detection Based on BERT Features Extraction [22]	BERT digunakan untuk mengekstraksi representasi teks URL, kemudian hasil ekstraksi diproses menggunakan model *deep learning* berbasis CNN. Model tersebut menunjukkan bahwa kombinasi ekstraksi fitur BERT dan CNN dapat digunakan untuk membedakan URL *phishing* dan URL *legitimate*.	BERT digunakan terutama sebagai pengekstrak fitur dan keluaran tersebut diteruskan ke CNN. Penelitian belum menggabungkan representasi kontekstual BERT dengan sekumpulan fitur numerik URL yang dihitung secara eksplisit. Ruang lingkup penelitian juga terbatas pada URL dan tidak mencakup email.	Penelitian ini menggabungkan representasi mBERT secara langsung dengan fitur numerik URL melalui proses *concatenation* sebelum klasifikasi. Penelitian ini juga menambahkan skenario terpisah untuk mendeteksi *email phishing*.

Lanjut pada halaman berikutnya

UNIVERSITAS MULTIMEDIA NUSANTARA

Tabel 2.1 Penelitian Terdahulu (lanjutan)

No	Penulis	Judul	Metode dan Kinerja	Keterbatasan	*Research Gap*
7	Asiri dkk. (2024)	PhishTransformer: A Novel Approach to Detect Phishing Attacks Using URL Collection and *Transformer* [4]	PhishTransformer menggabungkan CNN dan *Transformer encoder* untuk menganalisis URL yang terdapat di dalam halaman web. Pengujian terhadap 10.000 URL menghasilkan *precision*, *recall*, dan *F1-score* masing-masing sebesar 99%.	Model memerlukan kumpulan URL yang terdapat di dalam konten halaman web, seperti *hyperlink* dan *iframe*. Dengan demikian, masukan penelitian tidak hanya bergantung pada satu teks URL. Penelitian juga tidak membahas karakteristik email atau klasifikasi *email phishing*.	Penelitian ini melakukan klasifikasi berdasarkan teks satu URL dan 13 fitur numerik strukturnya tanpa membutuhkan konten halaman web. Selain itu, penelitian ini mengevaluasi skenario deteksi email menggunakan arsitektur *hybrid* mBERT yang terpisah.
8	Su dan Su (2023)	BERT-Based Approaches to Identifying Malicious URLs [23]	Pendekatan berbasis BERT digunakan untuk mempelajari hubungan semantik dan pola token dalam URL. Model yang dikembangkan memperoleh kinerja tinggi dengan *accuracy* pada kisaran 98–99% pada beberapa skenario pengujian URL berbahaya.	Penelitian membahas kategori URL berbahaya secara lebih umum dan tidak secara khusus berfokus pada klasifikasi biner URL *phishing*. Pendekatan tersebut juga belum menggabungkan representasi BERT dengan fitur numerik struktur URL melalui jalur fitur terpisah dan tidak membahas data email.	Penelitian ini secara khusus mengklasifikasikan URL menjadi kelas *legitimate* dan *phishing* menggunakan kombinasi representasi mBERT dan fitur numerik struktur URL. Penelitian ini juga menambahkan skenario deteksi *email phishing*.
9	Aljofey dkk. (2022)	An Effective Detection Approach for Phishing Websites Using URL and HTML Features [24]	Penelitian menggabungkan urutan karakter URL, informasi *hyperlink*, dan konten tekstual halaman web menggunakan TF-IDF, kemudian melakukan klasifikasi dengan XGBoost. Model dievaluasi pada dataset yang terdiri dari 60.252 halaman web dan menunjukkan bahwa kombinasi fitur tersebut efektif untuk mendeteksi halaman *phishing*.	Proses deteksi membutuhkan konten halaman web dan informasi HTML atau *hyperlink*. Informasi tersebut tidak selalu tersedia ketika sistem hanya menerima masukan berupa teks URL. Pendekatan ini juga tidak menggunakan model bahasa kontekstual seperti mBERT dan tidak membahas klasifikasi email.	Penelitian ini hanya membutuhkan teks URL dan fitur numerik strukturnya sehingga tidak bergantung pada ketersediaan HTML halaman web. Representasi kontekstual mBERT digabungkan dengan fitur numerik, serta diterapkan pula pada skenario klasifikasi email secara terpisah.

Lanjut pada halaman berikutnya

Tabel 2.1 Penelitian Terdahulu (lanjutan)

No	Penulis	Judul	Metode dan Kinerja	Keterbatasan	*Research Gap*
10	Yoon dkk. (2024)	Phishing Webpage Detection via Multi-Modal Integration of HTML DOM Graphs and URL Features Based on Graph Convolutional and *Transformer* Networks [25]	Penelitian menggunakan pendekatan multimodal yang menggabungkan GCN untuk mempelajari struktur HTML DOM, CNN untuk mempelajari karakter URL, dan *Transformer* untuk mempelajari pola kata pada URL. Integrasi beberapa modalitas tersebut memberikan kinerja yang lebih baik dibandingkan penggunaan representasi tunggal dalam mendeteksi halaman *phishing*.	Model membutuhkan struktur HTML DOM beserta URL halaman sehingga proses pengumpulan dan pemrosesan datanya lebih kompleks. Pendekatan tersebut tidak dapat langsung digunakan apabila masukan yang tersedia hanya berupa teks email atau URL tanpa HTML. Penelitian juga tidak membahas klasifikasi *email phishing*.	Penelitian ini menggunakan arsitektur yang lebih sederhana dengan menggabungkan representasi teks mBERT dan fitur numerik tanpa membutuhkan HTML DOM. Pendekatan tersebut dievaluasi pada dua skenario terpisah, yaitu klasifikasi email dan klasifikasi URL.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA

Berdasarkan Tabel 2.1, penelitian terdahulu mengenai deteksi *phishing* dapat dikelompokkan ke dalam beberapa pendekatan utama. Sebagian penelitian berfokus pada klasifikasi *email phishing* menggunakan model berbasis BERT dan *Transformer*, seperti penelitian Otieno dkk., Songailaite dkk., Melendez dkk., serta Ibrahim dan Elhafiz. Hasil penelitian tersebut menunjukkan bahwa model berbasis *Transformer* mampu menghasilkan kinerja yang baik dalam memahami konteks dan pola bahasa pada isi email. Namun, pendekatan tersebut umumnya masih berfokus pada representasi teks dan belum menggabungkan informasi tersebut dengan fitur numerik yang menggambarkan karakteristik domain pengirim, *subject*, *body*, serta keberadaan URL.

Penelitian lain berfokus pada deteksi URL *phishing* dan URL berbahaya menggunakan BERT, DistilBERT, CNN, dan *Transformer*, seperti penelitian Kumar, Elsadig dkk., Asiri dkk., serta Su dan Su. Penelitian-penelitian tersebut menunjukkan bahwa pola token dan struktur URL dapat dipelajari menggunakan model berbasis *Transformer* dengan kinerja yang tinggi. Meskipun demikian, beberapa penelitian belum menggabungkan representasi kontekstual URL dengan fitur numerik yang dihitung secara eksplisit, seperti panjang URL, panjang domain, jumlah subdomain, penggunaan alamat IP, jumlah karakter khusus, kata sensitif, dan nilai *entropy*.

Selain itu, penelitian Aljofey dkk. dan Yoon dkk. menggunakan informasi yang lebih kompleks, seperti fitur HTML, *hyperlink*, struktur HTML DOM, Graph Convolutional Network, CNN, dan *Transformer*. Pendekatan tersebut mampu memanfaatkan informasi struktural halaman web secara lebih lengkap. Namun, metode tersebut membutuhkan konten HTML atau struktur halaman web yang tidak selalu tersedia apabila masukan sistem hanya berupa teks email atau URL. Pendekatan tersebut juga belum secara khusus membahas klasifikasi email dan URL sebagai dua skenario yang dievaluasi menggunakan arsitektur *hybrid* yang sama.

Berdasarkan kinerja dan keterbatasan penelitian terdahulu tersebut, terdapat celah penelitian pada penggabungan representasi kontekstual mBERT dengan fitur numerik untuk dua objek *phishing* yang berbeda. Penelitian sebelumnya cenderung berfokus pada salah satu objek, yaitu email, URL, atau halaman web, serta menggunakan representasi teks atau fitur struktural secara terpisah. Oleh karena itu, penelitian ini menggunakan pendekatan *hybrid* mBERT yang menggabungkan representasi teks dengan fitur numerik melalui dua jalur pemrosesan sebelum proses klasifikasi.

Pada skenario email, representasi teks dari mBERT digabungkan dengan 13 fitur

numerik yang diekstraksi dari *sender*, *subject*, *body*, dan URL. Pada skenario URL, representasi teks URL dari mBERT digabungkan dengan 13 fitur numerik yang menggambarkan struktur URL. Kedua skenario tersebut dikembangkan dan dievaluasi secara terpisah karena dataset email dan dataset URL berasal dari sumber yang berbeda serta tidak memiliki pasangan langsung antara sebuah email dan URL yang terdapat di dalamnya.

Secara konseptual, *email phishing* dan *link phishing* memiliki hubungan yang erat. Email sering digunakan sebagai media rekayasa sosial untuk meyakinkan korban, menciptakan rasa mendesak, atau meminta korban melakukan tindakan tertentu. Tautan yang terdapat di dalam email kemudian dapat mengarahkan korban ke halaman palsu untuk memperoleh informasi sensitif. Namun, tidak semua *email phishing* mengandung tautan karena serangan juga dapat dilakukan melalui lampiran berbahaya, permintaan balasan, permintaan transfer, atau permintaan informasi pribadi secara langsung. Sebaliknya, *link phishing* juga dapat disebarkan melalui pesan singkat, media sosial, aplikasi percakapan, maupun media digital lainnya.

Dengan demikian, kontribusi penelitian ini bukan berupa satu model terpadu yang menganalisis email dan URL dalam satu pasangan data, melainkan penerapan dan evaluasi pendekatan *hybrid* mBERT pada dua skenario klasifikasi yang saling berkaitan secara konseptual. Pendekatan tersebut digunakan untuk mengevaluasi apakah kombinasi representasi teks dan fitur numerik dapat memberikan kinerja yang baik dalam mendeteksi *email phishing* dan *link phishing*.

UMN
UNIVERSITAS
MULTIMEDIA
NUSANTARA