



SKRIPSI

Judul:

Pengembangan Sistem Pendekripsi Situs Phishing Berbasis
Web Menggunakan Metode Long Short Term
Memory

Disusun oleh:

JERRY RUSLIM
NIM. 535200031

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TARUMANAGARA
2025

Pengesahan

Nama : JERRY RUSLIM
NIM : 535200031
Program Studi : TEKNIK INFORMATIKA
Judul Skripsi : Pengembangan Sistem Pendekripsi Situs Phishing Berbasis Web Menggunakan Metode Long Short Term Memory
Title : Development of a Web Based Phishing Site Detection System Using the Long Short Term Memory Method

Skripsi ini telah dipertahankan di hadapan Dewan Pengaji Program Studi TEKNIK INFORMATIKA Fakultas Teknologi Informasi Universitas Tarumanagara pada tanggal 03-Desember-2024.

Tim Pengaji:

1. DYAH ERNY HERWINDIATI, Ir., M.Si, Dr., Prof.
2. JANSON HENDRYLI, S. Kom. M.Kom.

Yang bersangkutan dinyatakan: **LULUS.**

Pembimbing:
VINY CHRISTANTI MAWARDI, S.Kom.,
M.Kom.
NIK/NIP: 10805002



Pembimbing Pendamping:
MANATAP DOLOK LAURO, S.Kom.,
M.M.S.I.
NIK/NIP: 10813003



Jakarta, 03-Desember-2024

Ketua Program Studi



VINY CHRISTANTI MAWARDI, S.Kom., M.Kom.

Persetujuan

Nama : JERRY RUSLIM
NIM : 535200031
Program Studi : TEKNIK INFORMATIKA
Judul : Pengembangan Sistem Pendekripsi Situs Phishing Berbasis Web Menggunakan Metode Long Short Term Memory

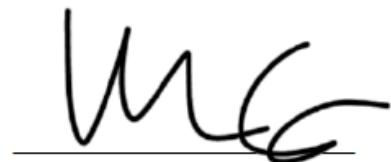
Skripsi ini disetujui untuk diuji

Jakarta, 16-November-2024

Pembimbing:
VINY CHRISTANTI MAWARDI, S.Kom.,
M.Kom.
NIK/NIP: 10805002



Pembimbing Pendamping:
MANATAP DOLOK LAURO, S.Kom.,
M.M.S.I.
NIK/NIP: 10813003



Pernyataan

Nama : JERRY RUSLIM
NIM : 535200031
Program Studi : TEKNIK INFORMATIKA
Judul : Pengembangan Sistem Pendeteksi Situs Phishing Berbasis Web Menggunakan Metode Long Short Term Memory

Dengan ini menyatakan bahwa skripsi ini merupakan hasil kerja saya sendiri di bawah bimbingan Tim Pembimbing dan bukan hasil plagiasi dan/atau kegiatan curang lainnya.

Jika saya melanggar pernyataan ini, maka saya bersedia dikenakan sanksi sesuai aturan yang berlaku di Universitas Tarumanagara.

Demikian pernyataan ini saya buat dengan sebenarnya, untuk dipergunakan sebagaimana mestinya.

Jakarta, 16-November-2024
Yang menyatakan



JERRY RUSLIM
NIM. 535200031

ABSTRAK

Jerry Ruslim, NPM: 535200031. Pengembangan Sistem Pendekripsi Situs Phishing Berbasis Web Menggunakan Metode Long Short Term Memory. Skripsi, Jakarta: Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, November 2024.

Perkembangan era digital yang sangat pesat diikuti juga dengan peningkatan yang signifikan dalam ancaman siber. Phishing adalah salah satu ancaman siber yang paling sering mendapatkan sorotan. Dengan mengeksplorasi teknik rekayasa psikologis, pelaku phishing dapat melancarkan aksinya untuk mengelabui individu dalam mengirimkan informasi pribadi yang bersifat berharga dan konfidensial. Salah satu teknik yang bisa digunakan dalam mendekripsi ancaman phishing ini adalah dengan penggunaan machine learning. Metode yang digunakan dalam penelitian ini adalah kombinasi antara LSTM (Long Short Term Memory) untuk pemrosesan URL dan jaringan dense untuk pemrosesan fitur-fitur yang diekstrak dari URL, seperti panjang URL, tingkat entropi, dan umur domain. Dataset pelatihan model menggunakan data yang diperoleh dari Kaggle serta validasi model menggunakan dataset Ebbu2017 dan data yang diperoleh dari PhishTank. Program yang dibangun berbasis website yang dapat menerima input URL dari pengguna dan menampilkan hasil prediksi dari model yang sudah dilatih. Output dari program ini berupa safe atau phishing. Dari hasil pengujian model, diperoleh akurasi sebesar 99,95% dengan precision 99,96%, recall 99,94%, dan F-1 Score sebesar 99,95%.

Kata Kunci: ancaman siber, phishing, Long Short Term Memory, URL, website

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmatnya, sehingga penulisan skripsi dengan judul “Pengembangan Sistem Pendekripsi Situs Phishing Berbasis Web Menggunakan Metode Long Short Term Memory” dapat diselesaikan dengan tepat waktu.

Penulisan buku skripsi ini dapat diselesaikan dengan baik atas bantuan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan kali ini saya hendak memberikan ucapan terima kasih kepada:

1. Ibu Viny Christanti Mawardi, S.Kom., M.Kom., selaku Ketua Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara sekaligus pembimbing utama dalam penyusunan dari proposal hingga buku skripsi ini dapat diselesaikan dengan baik.
2. Bapak Manatap Dolok Lauro, S.Kom., M.M.S.I., selaku Sekretaris Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Tarumanagara sekaligus pembimbing pendamping yang ikut memberikan arahan, bimbingan, serta saran dalam penyusunan buku skripsi.
3. Ibu Prof. Dr. Ir. Dyah Erny Herwindiati, M.Si., selaku Dekan Fakultas Teknologi Informasi Universitas Tarumanagara yang secara tidak langsung turut membantu dalam penyusunan Buku Skripsi ini.

4. Ibu Ir. Jeanny Pragantha, M.Eng. selaku Koordinator Skripsi yang selalu memberikan pengarahan mengenai informasi administrasi dari awal penulisan proposal skripsi hingga buku skripsi ini selesai dikerjakan
5. Seluruh dosen yang turut memberikan edukasi dalam proses perkuliahan yang bermanfaat untuk penyusunan buku skripsi.
6. Orang tua yang turut memberikan dukungan dalam penyelesaian buku skripsi.
7. Seluruh teman-teman dan rekan mahasiswa yang telah memberikan bantuan dan dukungan dalam penulisan buku skripsi.

Saya selaku penulis menyadari bahwa penulisan buku skripsi ini masih jauh dari kata sempurna dan masih banyak kesalahan yang perlu diperbaiki. Oleh karena itu, kritik dan saran dalam segala bentuk sangat diharapkan agar penulisan kedepannya bisa lebih bagus dan bermanfaat bagi pembaca.

Jakarta, 18 November 2024

Penulis,



Jerry Ruslim

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN SKRIPSI	ii
LEMBAR PERSETUJUAN SIDANG SKRIPSI	iii
LEMBAR PERNYATAAN.....	iv
ABSTRAK.....	v
KATA PENGANTAR	vi
DAFTAR TABEL	xi
DAFTAR GAMBAR.....	xii
DAFTAR LAMPIRAN	Error! Bookmark not defined.
BAB I PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Rancangan	4
1.3. Komponen Rancangan.....	5
1.4. Spesifikasi Rancangan.....	6
1.5. Kegunaan Rancangan	7
1.6. Rancangan yang Sudah Dibuat	8
BAB II LANDASAN TEORETIK	11

2.1. Sistem yang Dirancang	11
2.2. Landasan Teori	12
2.2.1. Phishing	12
2.2.2. Aplikasi Web.....	15
2.2.3. URL (Uniform Resource Locator)	16
2.2.4. <i>Deep Learning</i>	17
2.2.5. <i>Confusion Matrix</i>	28
2.2.6. Optimizer.....	30
2.2.7. Fungsi Aktivasi.....	31
2.2.8. Deep Neural Networks (DNN).....	33
BAB III RANCANGAN dan PEMBUATAN	34
3.1. Rancangan Sistem	34
3.1.1. Tahap Perencanaan.....	39
3.1.2. Tahap Analisis.....	40
3.1.3. Tahap Perancangan.....	40
3.2. Pembuatan Sistem.....	47
3.2.1. Pembuatan Modul Pelatihan dan Evaluasi	47

3.2.2. Pembuatan Antarmuka	48
BAB IV PENGUJIAN.....	49
4.1. Metode Pengujian	49
4.2. Hasil Pengujian	50
4.2.1. Pengujian Terhadap Modul.....	50
4.2.2. Pengujian Model LSTM	52
4.2.3. Pengujian Terhadap Output Program.....	55
4.3. Pembahasan	58
4.3.1. Pembahasan Pengujian Modul	58
4.3.2. Pembahasan Pengujian Model LSTM.....	58
4.3.3. Pembahasan Pengujian Output Program.....	60
BAB V KESIMPULAN DAN SARAN	62
5.1. Kesimpulan	62
5.2. Saran.....	63
DAFTAR PUSTAKA	65
LAMPIRAN 1.....	75
DAFTAR RIWAYAT HIDUP.....	146

DAFTAR TABEL

Tabel 4.1. Pengujian hyperparameter tuning berdasarkan panjang URL dan jumlah epoch.....	52
Tabel 4.2. Pengujian hyperparameter tuning berdasarkan optimizer dan fungsi aktivasi.....	53
Tabel 4.3. Hasil klasifikasi model.....	54
Tabel 4.4. Perbandingan akurasi model berdasarkan arsitektur.....	54
Tabel 4.5. Perbandingan akurasi dataset Ebbu 2017.....	55
Tabel 4.6. Hasil pengujian output program.....	56
Tabel 4.7. Perbandingan pengujian dengan program tersedia.	57
Tabel L1.1. Contoh perhitungan tokenisasi.....	75
Tabel L1.2. Contoh perhitungan embedding.....	76
Tabel L2.1. Contoh daga Kaggle fitur 1-2.....	86
Tabel L2.2. Contoh daga Kaggle fitur 3-4.....	87
Tabel L2.3. Contoh daga Kaggle fitur 5-8.....	88
Tabel L2.4. Contoh daga Kaggle fitur 9-10.....	89
Tabel L2.5. Contoh daga Kaggle fitur 11-12.....	90
Tabel L2.6. Contoh daga Kaggle fitur 13-14.....	91
Tabel L2.7. Contoh daga Kaggle fitur 15.....	92

DAFTAR GAMBAR

Gambar 1.1. Temuan upaya phishing tahun 2022 dan 2023	2
Gambar 1.2. Contoh upaya phishing melalui Whatsapp.....	3
Gambar 2.1. Contoh situs web sah yang menjadi target phishing.....	13
Gambar 2.2. Contoh situs web phishing.....	14
Gambar 2.3. Langkah pelancaran aksi phishing.....	15
Gambar 2.4. Strsuktur sebuah URL.....	18
Gambar 2.4. Diagram Venn tentang konsep dan klasifikasi machine learning.....	19
Gambar 2.5. Ilustrasi jaringan RNN.....	20
Gambar 2.6. Ilustrasi sel LSTM.....	22
Gambar 2.7. Struktur sel LSTM.....	23
Gambar 2.8. Cell state LSTM.....	23
Gambar 2.9. Forget gate LSTM.....	24
Gambar 2.10. Input gate LSTM.....	25
Gambar 2.11. Perubahan cell state.....	27
Gambar 2.12. Output Gate LSTM.....	28
Gambar 2.13. Struktur Confusion Matrix.....	30
Gambar 3.1. Rancangan arsitektur model pendekripsi URL phishing.....	37
Gambar 3.2. Flowchart tahap pelatihan model	39

Gambar 3.3. Flowchart tahap pengujian model	40
Gambar 3.4. Rancangan Diagram Hirarki.....	41
Gambar 3.5. Rancangan State Transition Diagram.....	42
Gambar 3.6. Rancangan modul utama.....	43
Gambar 3.7. Rancangan modul input.....	43
Gambar 3.8. Rancangan modul output.....	44
Gambar 3.9. Rancangan modul bantuan.....	45
Gambar L3.1. Hasil pengujian modul utama.....	86
Gambar L3.2. Hasil pengujian modul input.....	87
Gambar L3.3. Hasil pengujian modul output (safe)	87
Gambar L3.4. Hasil pengujian modul output (phishing)	88
Gambar L3.5. Hasil pengujian modul bantuan.....	88
Gambar L4.1. Distribusi fitur panjang URL.....	89
Gambar L4.2. Distribusi fitur ip address sebagai nama domain.....	90
Gambar L4.3. Distribusi fitur tingkat entropi URL.....	91
Gambar L4.4. Distribusi fitur karakter punycode.....	92
Gambar L4.5. Distribusi fitur rasio huruf dan angka.....	93
Gambar L4.6. Distribusi fitur banyak titik dalam URL.....	93
Gambar L4.7. Distribusi fitur banyak simbol at dalam URL.....	94
Gambar L4.8. Distribusi fitur banyak simbol dash dalam URL.....	95

Gambar L4.9. Distribusi fitur banyak TLD dalam URL.....	96
Gambar L4.10. Distribusi fitur domain mengandung angka.....	97
Gambar L4.11. Distribusi fitur banyak subdomain dalam URL.....	98
Gambar L4.12. Distribusi fitur tingkat entropi karakter non-alfanumerik.....	99
Gambar L4.13. Distribusi fitur URL mengandung tautan.....	100
Gambar L4.14. Distribusi fitur umur domain.....	101
Gambar L5.1. Confusion matrix model 1 LSTM.....	102
Gambar L5.2. Grafik loss model 1 LSTM.....	103
Gambar L5.3. Confusion matrix model 2 LSTM.....	104
Gambar L5.4. Grafik loss model 2 LSTM.....	105
Gambar L5.5. Confusion matrix model 3 LSTM.....	106
Gambar L5.6. Grafik loss model 3 LSTM.....	107
Gambar L5.7. Confusion matrix model 4 LSTM.....	108
Gambar L5.8. Grafik loss model 4 LSTM.....	109
Gambar L5.9. Confusion matrix model 5 LSTM.....	110
Gambar L5.10. Grafik loss model 5 LSTM.....	111
Gambar L5.11. Confusion matrix model 6 LSTM.....	112
Gambar L5.12. Grafik loss model 6 LSTM.....	113
Gambar L6.1. Confusion matrix arsitektur LSTM only.....	114
Gambar L6.2. Grafik loss arsitektur LSTM only.....	115

Gambar L6.3. Confusion matrix arsitektur dense only.....	116
Gambar L6.4. Grafik loss arsitektur dense only.....	117
Gambar L7.1. Confusion matrix pengujian dataset Ebbu2017.....	118
Gambar L8.1. Confusion matrix model A.....	126
Gambar L8.2. Grafik loss model A.....	127
Gambar L8.1. Confusion matrix model B.....	128
Gambar L8.2. Grafik loss model B.....	129
Gambar L8.1. Confusion matrix model C.....	130
Gambar L8.2. Grafik loss model C.....	131
Gambar L8.1. Confusion matrix model D.....	132
Gambar L8.2. Grafik loss model D.....	133
Gambar L8.1. Confusion matrix model E.....	134
Gambar L8.2. Grafik loss model E.....	135
Gambar L8.1. Confusion matrix model F.....	136
Gambar L8.2. Grafik loss model F.....	137
Gambar L8.1. Confusion matrix model G.....	138
Gambar L8.2. Grafik loss model G.....	139
Gambar L8.1. Confusion matrix model H.....	140
Gambar L8.2. Grafik loss model H.....	141
Gambar L8.1. Confusion matrix model I.....	142

Gambar L8.2. Grafik loss model I.....	143
Gambar L9.1. Hasil pengecekan plagiarism.....	144
Gambar L10.1. Link source code GitHub.....	145

DAFTAR LAMPIRAN

LAMPIRAN 1 Contoh Perhitungan.....	75
LAMPIRAN 2 Contoh Data Kaggle.....	86
LAMPIRAN 3 Hasil pengujian modul.....	93
LAMPIRAN 4 Distribusi fitur dataset.....	96
LAMPIRAN 5 Hasil pengujian hyperparameter tuning kedua.....	109
LAMPIRAN 6 Hasil pengujian model berdasarkan arsitektur.....	121
LAMPIRAN 7 Hasil pengujian dataset Ebbu2017.....	125
LAMPIRAN 8 Hasil pengujian hyperparameter tuning pertama.....	126
LAMPIRAN 9 Hasil pengecekan plagiarism Turnitin	144
LAMPIRAN 10 Link source code GitHub	145

BAB I

PENDAHULUAN

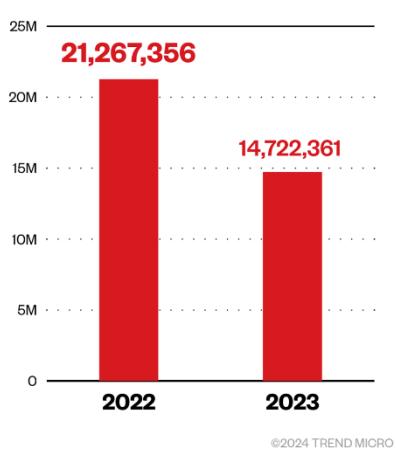
1.1. Latar Belakang

Perkembangan era digital yang sangat pesat telah mempermudah berbagai aktivitas manusia seperti penggunaan teknologi internet untuk mempermudah akses informasi, melakukan transaksi keuangan, dan memungkinkan komunikasi jarak jauh dalam waktu yang singkat. Namun, kemajuan ini juga diikuti dengan peningkatan signifikan dalam ancaman siber. Selain mengakibatkan kerugian terhadap individu, ancaman siber juga dapat menimbulkan kerugian dalam skala yang lebih besar terhadap negara [1].

Berbagai cara bisa dilakukan oleh pelaku serangan siber untuk memperoleh keinginannya. Beberapa contoh dari serangan siber yang sering ditemukan seperti pemerasan, dimana pelaku melakukan tindakan pemaksaan atau pengancaman untuk memperoleh uang. Peretasan, mengeksplorasi kerentanan dari keamanan sistem untuk memperoleh informasi yang bersifat rahasia. Serangan *denial of service (DOS)*, adalah suatu tindakan yang memaksa server untuk mati dengan cara membanjirinya dengan permintaan yang tidak sah [2].

Salah satu ancaman siber yang paling sering mendapatkan sorotan adalah phishing. Phishing adalah salah satu tindakan kriminal untuk mencoba mendapatkan informasi penting atau informasi rahasia dari pengguna dengan mengadopsi

berbagai jenis metode yang baru dan kreatif. Phishing merupakan ancaman serius dalam dunia siber yang telah ada selama beberapa dekade dan terus menjadi masalah besar hingga saat ini [22].

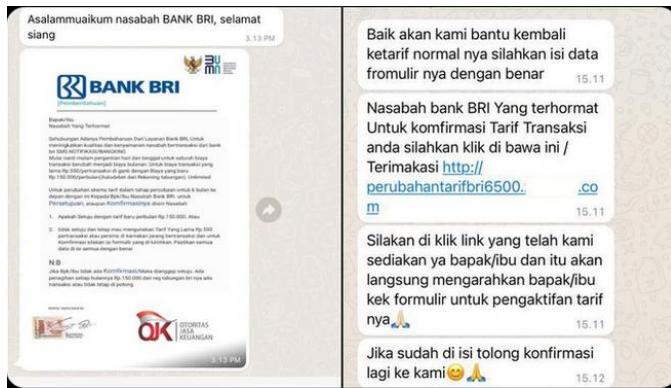


Gambar 1.1. Temuan upaya phishing tahun 2022 dan 2023 [49].

Ancaman ini tidak hanya menimbulkan kerugian finansial, tetapi juga mengancam privasi dan keamanan data, yang bisa dilakukan dengan berbagai teknik seperti manipulasi tautan, pemalsuan situs web, dan rekayasa sosial [3]. Namun, pendekatan yang paling umum ditemukan adalah dengan cara membuat situs web palsu yang meniru situs aslinya [4].

Upaya phishing biasanya memberikan tawaran yang terlalu bagus untuk menjadi kenyataan dan memancing korban untuk mengklik tautan dengan janji adanya hadiah atau tawaran special yang akan diberikan. Selain itu, phishing juga bisa dikenali dengan beberapa ciri lainnya seperti penggunaan bahasa yang

mendesak untuk menciptakan perasaan urgensi, meminta informasi pribadi atau keuangan, kesalahan dalam pengetikan, dan penyapaan yang umum seperti “Pelanggan yang Terhormat”, dimana perusahaan yang sah biasanya akan langsung menyapa menggunakan nama [5].



Gambar 1.2. Contoh upaya phishing melalui Whatsapp [48].

Melihat upaya phishing yang semakin marak, mengakibatkan orang-orang menjadi kurang percaya diri, khawatir, dan merasa kurang puas terhadap teknologi saat ini yang dirasa tidak mampu melindungi masyarakat dari serangan phishing ini [6]. Oleh karena itu, upaya untuk meningkatkan keamanan siber menjadi sangat penting dalam melindungi masyarakat dari risiko ancaman siber yang terus berkembang. Salah satu metode yang umum digunakan oleh browser modern saat ini untuk mendeteksi situs phishing adalah metode *blacklisting*. Namun metode ini terbukti kurang efektif dalam melindungi pengguna dari situs-situs phishing yang

baru [6] dan dapat dengan mudah diakali oleh pelaku phishing dengan cara mengubah beberapa karakter dalam URL [8].

Penelitian ini bertujuan untuk mengembangkan sebuah aplikasi website untuk mendeteksi upaya phishing menggunakan teknik Long Short-Term Memory (LSTM). LSTM merupakan sebuah jenis Recurrent Neural Network (RNN) yang terkenal efektif dalam memproses dan menganalisis data urutan waktu, termasuk teks. LSTM dipilih sebagai metode pendekripsi situs phishing karena kemampuannya memahami dan memproses data dengan dependensi jangka panjang [9]. LSTM memiliki kemampuan pembelajaran yang kuat, dapat secara otomatis mempelajari karakterisasi data tanpa ekstraksi fitur kompleks secara manual, dan memiliki potensi yang kuat dalam menghadapi data masif yang kompleks dan berdimensi tinggi [10].

1.2. Rumusan Rancangan

Berdasarkan beberapa permasalahan yang telah dipaparkan dalam latar belakang, proposal ini mengajukan penggunaan metode LSTM (Long Short Term Memory) dalam perancangan sistem pendekripsi situs phishing berbasis web. Dataset yang digunakan untuk pembangunan model berupa URL dari dari situs-situs yang telah dikonfirmasi sebagai situs phishing yang diperoleh dari platform penyedia dataset, Kaggle, dan sebuah situs komunitas anti-phishing, bernama

PhishTank. Dataset yang telah diperoleh akan melalui tahap ekstraksi fitur sebelum dijadikan *input* untuk model LSTM yang akan dirancang. Hasil akhir rancangan berupa situs web yang menerima *input* berupa URL dan *output* berupa hasil prediksi, yaitu phishing atau non-phishing.

1.3. Komponen Rancangan

Berikut merupakan beberapa komponen rancangan dari sistem pendekripsi situs phishing berbasis web yang akan dirancang.

1. Data

Kumpulan data latih dan data uji yang digunakan dalam perancangan ini diambil dari situs Kaggle dan PhishTank sebanyak 22000 data campuran antara situs phishing dan situs sah. Data yang diperoleh dari Kaggle sudah dilengkapi dengan fitur dan label, sedangkan data yang diperoleh dari PhishTank akan melalui tahap ekstraksi fitur secara manual untuk menyamakan format data.

2. Model LSTM

Model LSTM bisa digunakan sebagai algoritma pendekripsi phishing karena keunggulannya dalam memproses data dalam jumlah besar dan mampu mempelajari dan menganalisis pola-pola yang ada dalam suatu data sekuensial. Dengan demikian, LSTM mampu menemukan fitur-fitur yang mengindikasikan suatu URL phishing.

3. Website

Rancangan aplikasi pendeteksi URL phishing akan diintegrasikan dalam bentuk aplikasi web untuk memudahkan akses bagi pengguna dan bisa digunakan oleh beberapa pengguna sekaligus. Rancangan antarmuka dari aplikasi web mengizinkan pengguna untuk melakukan *input* URL yang ingin divalidasi dan mengembalikan hasil prediksi kepada pengguna.

4. Python

Rancangan aplikasi web dan model LSTM akan menggunakan bahasa pemrograman Python. Python memungkinkan penggunaan *framework* Flask untuk perancangan aplikasi web dan memberikan akses ke *library* seperti Tensorflow dan Keras yang akan memudahkan pembangunan dan pelatihan model. Selain itu, penggunaan Python juga dapat memudahkan proses integrasi model dengan aplikasi web.

1.4. Spesifikasi Rancangan

Aplikasi pendeteksi situs phishing memiliki beberapa komponen didalamnya. Berikut beberapa modul yang akan dimiliki oleh program yang dirancang.

1. Modul Utama

Modul utama merupakan halaman pertama yang dilihat pengguna ketika masuk ke aplikasi

2. Modul *Input*

Modul *input* sebagai tempat pengguna memasukkan URL dari situs yang ingin dilakukan pengecekan.

3. Modul *Output*

Modul *output* menampilkan hasil prediksi phishing atau non-phishing dari URL yang *diinput* pengguna.

4. Modul Bantuan

Modul panduan bertujuan untuk menyediakan panduan penggunaan aplikasi.

1.5. Kegunaan Rancangan

Aplikasi ini dirancang dengan harapan dapat memberikan informasi yang akurat dan terpercaya tentang keamanan situs web yang dikunjungi pengguna. Dengan adanya informasi yang jelas dan tersedia secara *real-time*, diharapkan pengguna dapat meningkatkan kewaspadaan terhadap situs web yang mencurigakan. Dengan demikian, akan secara langsung meningkatkan keamanan pengguna terhadap kejahatan siber dan mengurangi resiko terjebak dalam upaya phishing yang bertujuan untuk memperoleh data pribadi pengguna.

1.6. Rancangan yang Sudah Dibuat

Berikut ini adalah beberapa rancangan sistem pendekripsi situs phishing yang telah dibuat dengan berbagai teknik dan metode.

1. Abdullateef O. Balogun memperkenalkan model pembelajaran meta berbasis *functional tree (FT)* untuk mendekripsi situs web phishing [11], penelitian ini menunjukkan bahwa penggunaan *functional tree* memiliki kinerja yang superior dibandingkan beberapa model klasifikasi dan model hibrid dari studi sebelumnya. Penelitian ini menggunakan *functional tree* yang digabungkan dengan beberapa *meta-learner* seperti *bagging*, *boosting*, dan *rotation forest*. Hasilnya menunjukkan bahwa kombinasi *functional tree* dan *meta-learner* dapat diandalkan, terutama kombinasi dari dengan akurasi prediksi yang mencapai 99%.
2. Xi Xiao dan Wentao Xiao mengusulkan pendekatan baru dalam membangun sistem pendekripsi situs phishing menggunakan kombinasi *self-attention CNN* dan *multi-head self-attention* [12]. Demi keseimbangan pada dataset pelatihan model, GAN digunakan untuk menghasilkan URL phishing. Model yang diusulkan menunjukkan hasil prediksi yang lebih baik menggunakan URL yang dihasilkan GAN dibandingkan URL dari dunia nyata dengan akurasi sebesar 92.05%. Rancangan model ini masih bisa dikembangkan dengan cara

mempertimbangkan konten HTML, sehingga dapat memperkaya fitur yang digunakan.

3. Suleiman Y. Yerima dan Mohammed K. Alzaylaee mengusulkan model *deep learning* berbasis 1D CNN untuk mendeteksi situs phishing [13]. Model ini dievaluasi secara ekstensif menggunakan dataset yang terdiri dari 4.898 sampel situs phishing dan 6.157 sampel situs yang sah. Hasilnya menunjukkan bahwa model berbasis CNN yang diusulkan memiliki performa yang lebih baik dibandingkan beberapa metode klasifikasi *machine learning* yang populer ketika dievaluasi pada dataset yang sama dengan akurasi mencapai 97.3%. Untuk penelitian selanjutnya, proses pelatihan model dapat ditingkatkan dengan mengotomatiskan pencarian dan pemilihan parameter kunci seperti jumlah filter dan panjang filter untuk mencapai kinerja CNN yang lebih optimal.
4. Sami Smadi dan Nauman Aslam memperkenalkan model pengklasifikasi email sah atau phishing menggunakan pendekatan hybrid dengan cara menggabungkan fitur-fitur yang diperoleh dari judul dan konten email [14]. Dari beberapa algoritma yang diuji, *random forest* memiliki akurasi yang paling tinggi ketika dievaluasi menggunakan 23 fitur dengan tingkat akurasi sebesar 98.87%. Studi ini menekankan bahwa proses ekstraksi fitur dalam tahap pemrosesan awal memiliki peranan penting untuk mencapai akurasi yang tinggi. Sistem klasifikasi email phishing ini dapat dikembangkan dengan mengintegrasikan fitur

tambahan yang lebih terperinci dalam proses deteksi agar bisa beradaptasi terhadap jenis serangan phishing baru.

5. Kang Leng Chiew mengusulkan pendekatan heuristik dalam mendeteksi situs phishing dengan menggunakan gambar logo yang ditampilkan sebagai alat verifikasi identitas dari suatu situs web [15]. Metode ini melibatkan dua proses utama antara lain, ekstraksi logo menggunakan algoritma *decision tree* dan proses verifikasi identitas melalui pencarian google image. Dengan tingkat *true positive rate* sebesar 99.8% dan *true negative rate* sebesar 87%, dapat disimpulkan bahwa elemen grafis seperti logo dapat diandalkan dalam mendeteksi situs phishing.
6. Kevin Marcello Jonathan membandingkan kinerja dari algoritma Naïve Bayes dan C4.5 dalam mendeteksi URL phishing [16]. Pengujian yang dilakukan menggunakan lebih dari 100.000 data yang dikoleksi dari berbagai sumber dan kemudian diekstraksi hingga diperoleh 18 fitur. Hasil dari penelitian ini menunjukkan bahwa algoritma C4.5 memiliki akurasi yang lebih baik dibandingkan Naïve Bayes dengan akurasi sebesar 87.11%.

BAB II

LANDASAN TEORETIK

2.1. Sistem yang Dirancang

Program yang dirancang merupakan aplikasi web yang digunakan untuk mendeteksi situs phishing secara otomatis dan real-time menggunakan model berbasis LSTM. Program akan menerima *input* berupa URL dari pengguna. Selanjutnya, sistem akan melakukan preprocessing terhadap url yang diinput pengguna seperti normalisasi dan tokenisasi, dilanjutkan dengan ekstraksi fitur-fitur yang dianggap relevan dalam medeteksi situs phishing. Hasil tokenisasi URL dan nilai dari fitur akan diumpan ke dalam model yang sudah dilatih, hingga akhirnya ditampilkan hasil klasifikasi URL kepada pengguna.

Dalam tahap pelatihan model, URL yang sudah ditokenisasi akan dijadikan input ke dalam jaringan LSTM. URL merupakan data sekuensial yang memiliki struktur yang berurutan, sehingga LSTM dipercaya dapat melakukan analisa terhadap pola dari setiap karakter dalam data latih [9]. Fitur-fitur yang diekstraksi berupa data tabular yang sering sekali memiliki hubungan yang non-linear sehingga penggunaan jaringan dense layer merupakan opsi yang tepat untuk pemrosesan fitur ini. Selanjutnya, hasil dari kedua jaringan ini akan digabungkan untuk dilakukan tahap klasifikasi. Setelah selesai dengan tahap pelatihan, dilanjutkan dengan tahap evaluasi model dengan melakukan hyperparameter tuning, seperti perubahan

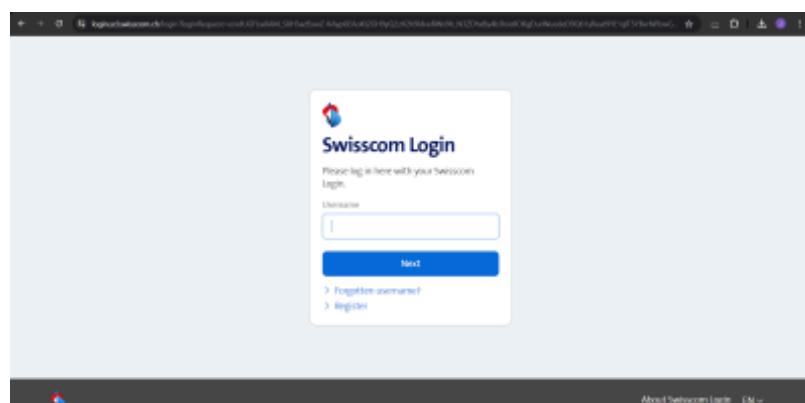
jumlah unit LSTM, optimizer, dan fungsi aktivasi, serta menggunakan bantuan confusion matrix untuk menganalisa performa model.

2.2. Landasan Teori

2.2.1. Phishing

Phishing adalah suatu ancaman terhadap keamanan yang berbahaya dan tidak bisa diremehkan. Dengan cara mengeksplorasi teknik rekayasa psikologis dan sosial yang canggih, pelaku dapat mengelabui individu dalam mengeklik tautan situs berbahaya dan mengirimkan informasi sensitif yang berharga dan konfidensial, seperti data pribadi atau perusahaan serta kredensial akun yang kemudian digunakan untuk mengakibatkan kerugian yang mendalam bagi korban [17].

Gambar 2.1 dan Gambar 2.2 menunjukkan contoh situs web sah dan hasil tiruan yang digunakan untuk upaya phishing.



Gambar 2.1. Contoh situs web sah yang menjadi target phishing.

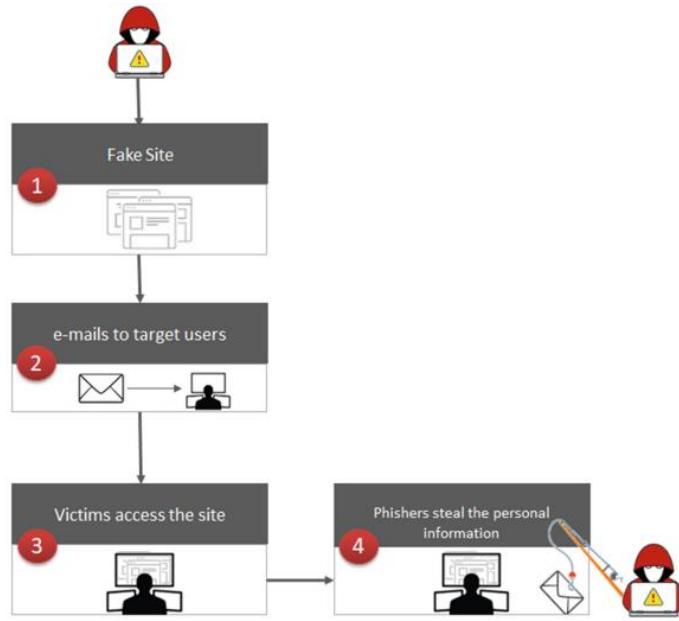


Gambar 2.2. Contoh situs web phishing.

Berikut adalah beberapa jenis phishing yang umum ditemukan :

1. *Spear Phishing*

Spear phishing adalah salah satu jenis serangan phishing yang populer dimana penyerang memberikan informasi yang secara spesifik hanya ditujukan kepada beberapa orang terpilih saja [18]. *Spear phishing* biasanya memerlukan waktu persiapan yang lebih lama dibandingkan jenis phishing lainnya, karena melibatkan riset yang mendalam dari korban yang ditargetkan [19]. Hal ini dilakukan demi merancang email dan situs web yang relevan terhadap preferensi dari korban, sehingga dapat meningkatkan peluang keberhasilan dalam mengelabui korban untuk melakukan apapun yang diinginkan pelaku [20].



Gambar 2.3. Langkah pelancaran aksi phishing [24].

2. Whaling

Whaling adalah metode phishing yang mirip dengan *spear phishing*. Satu-satunya hal yang membedakan *whaling* dengan *spear phishing* adalah korban *whaling* yang biasanya ditargetkan terhadap individu dengan pangkat yang tinggi di suatu perusahaan yang memiliki akses istimewa terhadap informasi perusahaan [21]. Tujuan dari pelaku *whaling* sama dengan pelaku phishing pada umumnya, yaitu membujuk korban untuk mengunduh program *malware* yang memberikan akses kepada pelaku untuk memonitor aktivitas korban dan memperoleh data perusahaan [22].

3. *Vishing*

Vishing adalah suatu jenis rekayasa sosial melalui suara atau percakapan, dimana penyerang menggunakan saluran komunikasi, seperti panggilan telepon untuk mengelabui korbannya [50]. Tindakan *vishing* biasanya mengandalkan kepercayaan diri pelaku dan mengatasnamakan dirinya sebagai pihak atau entitas yang sah demi memperoleh kepercayaan korban [23]. Selanjutnya, pelaku akan berusaha untuk memperoleh informasi pribadi dari korban atau membujuk korban untuk melakukan pengiriman uang dalam jumlah besar [24].

2.2.2. Aplikasi Web

Aplikasi web adalah perangkat lunak yang dirancang untuk dijalankan dalam lingkungan berbasis web yang memungkinkan fungsi pemrosesan informasi dari jarak jauh melalui *browser* dan dijalankan di server web, server aplikasi, atau server basis data. Berbeda dengan situs web sederhana pada umumnya yang hanya memiliki tautan navigasi, aplikasi web menawarkan fungsi yang lebih kompleks dan terperinci sesuai dengan kebutuhan pengguna [25]. Meningkatnya penggunaan Internet telah mempercepat pengembangan dan penyebaran luas aplikasi web, menjadikan web sebagai platform dominan untuk aplikasi perangkat lunak baru [26].

Aplikasi web memiliki beberapa perbedaan dari cara pengembangan, pemeliharaan, dan penggunaan dibandingkan dengan sistem perangkat lunak tradisional yang

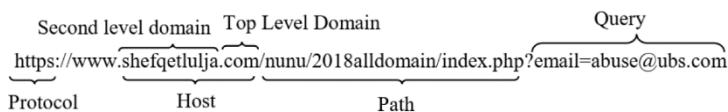
memerlukan instalasi secara lokal pada komputer atau server tertentu. Berikut beberapa keuntungan yang dimiliki aplikasi web [27]:

1. Akses Global, pemanfaatan internet dan web dapat menghilangkan jarak fisik yang memungkinkan akses terhadap informasi secara global dengan cepat dan mudah.
2. Mendukung penggunaan *multi-user* secara simultan, berbeda dengan aplikasi *desktop* tradisional yang biasanya hanya bisa digunakan oleh satu orang pada satu waktu, aplikasi web memungkinkan program untuk digunakan beberapa orang secara bersamaan.
3. Kompatibilitas, aplikasi web memiliki kompatibilitas dengan perangkat apapun yang memiliki akses internet.
4. Efektivitas biaya, aplikasi web memiliki banyak komponen seperti *web browser*, *database*, dan layanan *hosting* yang tersedia secara gratis, sehingga dapat mengurangi biaya pemeliharaan.

2.2.3. URL (Uniform Resource Locator)

URL adalah penanda unik dari suatu halaman yang digunakan untuk mengakses suatu sumber daya di internet. URL merupakan mekanisme utama yang digunakan seluruh browser untuk melihat atau mengambil sumber daya yang dipublikasikan seperti halaman situs, gambar, dan lainnya [56]. URL dapat disebut sebagai data sekuensial karena sifatnya yang memiliki pola atau komponen yang

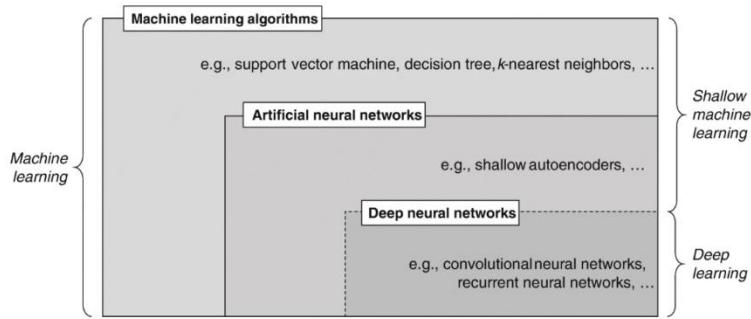
terstruktur. Dari sebuah URL dapat diperoleh beberapa fitur seperti panjang URL, banyak titik dalam URL, banyak subdomain, protokol yang digunakan, simbol apa saja dalam URL, dan sebagainya [56]. Struktur dari sebuah URL dapat dilihat pada Gambar 2.4.



Gambar 2.4. Struktur sebuah URL[54].

2.2.4. Deep Learning

Deep learning adalah salah satu cabang dari *machine learning*. Algoritma ini sering digunakan dalam menyelesaikan permasalahan data, seperti klasifikasi, analisis, dan pengelompokan data. *Deep learning* umumnya terdiri dari jaringan yang lebih banyak dan kompleks, sehingga memiliki performa yang jauh mengungguli metode *machine learning* tradisional [28]. Dengan memanfaatkan struktur jaringan didalamnya, *deep learning* sangat baik digunakan dalam memproses data dalam jumlah banyak dan data berdimensi tinggi, seperti foto, video, teks, dan suara [29]. Namun, beberapa penelitian menunjukkan bahwa *machine learning* tradisional dapat menjadi lebih efektif dalam memproses data dengan fitur yang lebih sedikit [30]. Konsep *deep learning* sebagai salah satu algoritma *machine learning* dapat dilihat pada Gambar 2.5.

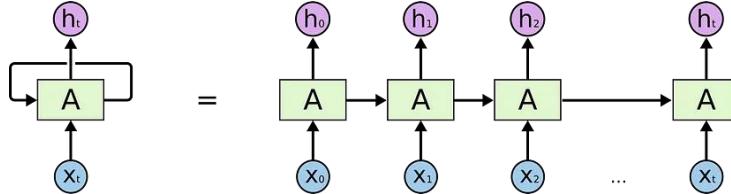


Gambar 2.5. Diagram Venn tentang konsep dan klasifikasi *machine learning* [42].

2.2.3.1. RNN (Recurrent Neural Network)

Recurrent Neural Network (RNN) merupakan salah satu tipe arsitektur *machine learning* yang sering kali digunakan sebagai metode pendekripsi pola pada data sekuensial, seperti suara, teks, tulisan, dan video. Hal yang membedakan RNN dengan arsitektur *machine learning* lainnya adalah siklus dari jaringan RNN yang mengirimkan informasi kembali ke dirinya sendiri [31]. Melalui pengulangan ini, model dapat menyimpan informasi dari *input* sebelumnya dan memungkinkannya untuk menemukan korelasi antar peristiwa di dalam data.

Gambaran dari cara kerja arsitektur RNN dapat dilihat pada Gambar 2.6.



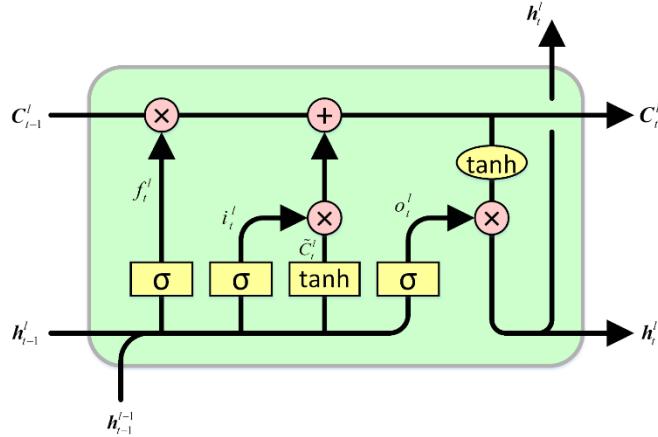
Gambar 2.6. Ilustrasi jaringan RNN [43].

Namun, RNN memiliki dua permasalahan yang cukup krusial, yaitu *vanishing gradient* dan *exploding gradient* [32]. *Vanishing gradient* terjadi ketika gradien dari *loss function* menjadi semakin kecil dalam setiap langkah waktu. Akibatnya, jaringan tidak dapat mempelajari informasi yang memiliki ketergantungan jangka panjang. Sebaliknya, *exploding gradient* terjadi akibat nilai gradien yang terus membesar akibat bobot yang terlalu besar. Hal ini akan berdampak terhadap perubahan pada parameter model yang mengakibatkannya untuk mendapatkan pembaruan yang sangat besar dan menimbulkan ketidakstabilan. Dengan alasan ini, telah ditemukan beberapa modifikasi dari RNN yang didesain khusus untuk menghadapi permasalahan ini, namun tetap mempertahankan sifat utama dari RNN, contohnya *Long Short Term Memory (LSTM)*, *Gated Recurrent Unit (GRU)* [33], dan *Bidirectional RNN (BRNN)* [34].

2.2.3.2. LSTM (*Long Short Term Memory*)

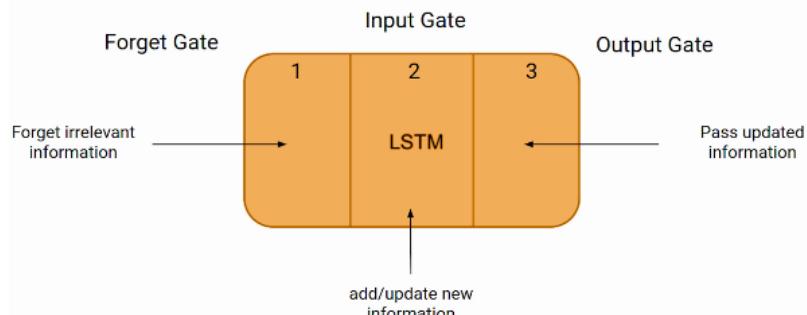
Long short term memory (LSTM) adalah salah satu variasi dari *recurrent neural network* (RNN). RNN dikenal dengan kemampuannya yang sangat baik dalam menangkap informasi dari data sekuensial. Namun, RNN memiliki kekurangan dalam perihal ingatan yang mengakibatkan turunnya kemampuan klasifikasi dalam jangka waktu yang lebih lama [35]. LSTM hadir dengan tujuan mengatasi permasalahan ini [36].

LSTM yang pertama kali diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997 [37], merupakan jenis *Recurrent Neural Network* yang telah dimodifikasi dengan cara menambahkan sel memori. Modifikasi ini memungkinkan LSTM untuk menyimpan informasi dalam jangka waktu yang lama [38], sehingga memiliki keunggulan dalam mengekstraksi fitur, dan mengingatnya dalam periode yang lebih lama, serta mengabaikan informasi yang tidak relevan, sehingga menjadi jauh lebih efektif dalam memproses dan mengklasifikasikan data sekuensial [39]. Ilustrasi dari sebuah sel LSTM dapat dilihat pada Gambar 2.7.



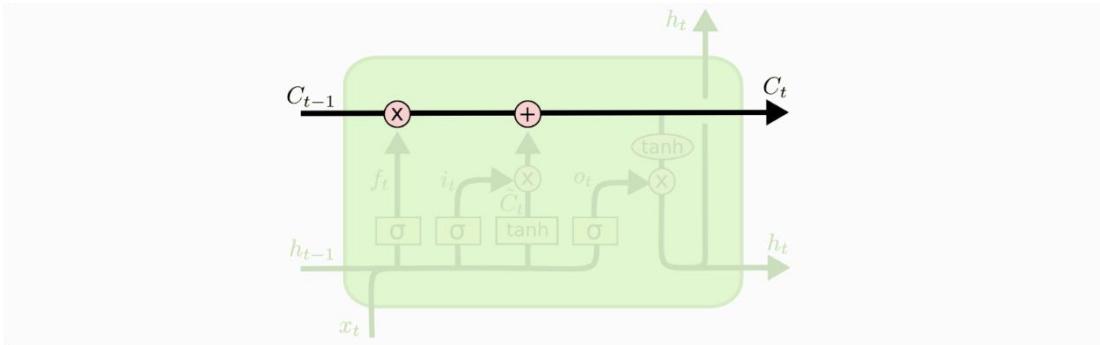
Gambar 2.7. Ilustrasi sel LSTM [44].

Arsitektur LSTM terdiri dari 3 bagian yang disebut gerbang atau *gate*, antara lain, *input gate*, *forget gate*, dan *output gate*. Struktur sel LSTM dapat dilihat pada Gambar 2.8.



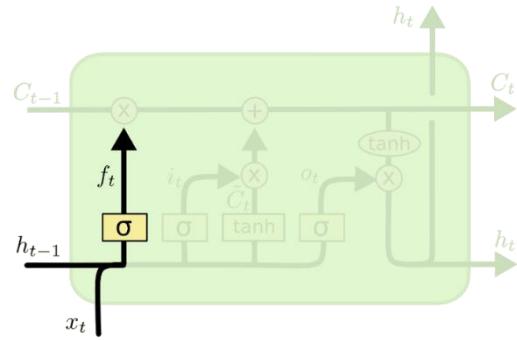
Gambar 2.8. Struktur sel LSTM [45].

Selain tiga gerbang ini, LSTM juga memiliki komponen *cell state* yang memiliki peranan krusial dalam algoritma LSTM yang bertindak sebagai penyimpan memori sepanjang rangkaian jaringan [40]. *Cell state* dalam sel LSTM dapat dilihat pada Gambar 2.9.



Gambar 2.9. Cell state LSTM [46].

Dengan adanya *forget gate*, algoritma LSTM dapat memutuskan informasi apa yang diperlukan dan berapa banyak informasi dari *cell state* sebelumnya yang dihapus. *Forget gate* menerima 2 *input* yaitu x_t (*input* pada waktu tertentu) dan h_{t-1} (*output* sel sebelumnya) yang akan dikalikan dengan matriks bobot dilanjutkan dengan penambahan bias [41]. Fungsi aktivasi sigmoid digunakan untuk memastikan *output* dari *forget gate* bernilai 0 untuk informasi yang akan dilupakan atau 1 untuk informasi yang akan disimpan. Alur *forget gate* dalam sel LSTM dapat dilihat pada Gambar 2.10. Perhitungan pada *forget gate* dapat dilihat pada persamaan 2.1 [46].



Gambar 2.10. *Forget gate* LSTM [46].

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.1)$$

Keterangan :

f_t = forget gate

σ = fungsi sigmoid

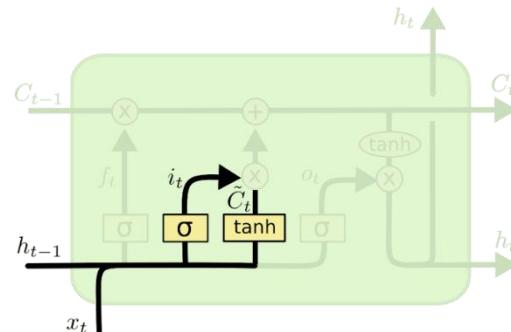
W_f = bobot forget gate

h_{t-1} = output pada timestep t-1

x_t = input pada timestep t

b_f = bias forget gate

Setelah melewati forget gate, tahap berikutnya adalah *input* gate yang berfungsi untuk menyaring dan menyimpan informasi baru yang relevan. *Input* gate menggunakan *input* yang sama dengan forget gate yaitu x_t dan h_{t-1} dan memiliki memiliki dua unit yang menggunakan fungsi aktivasi yang berbeda [41]. Unit pertama menggunakan fungsi aktivasi sigmoid dan unit kedua menggunakan fungsi aktivasi tanh. Hasil dari kedua unit ini akan dikalikan dan nantinya digunakan sebagai acuan perubahan memori pada cell state. Tahap *input* gate dalam sel LSTM dapat dilihat pada Gambar 2.11. Perhitungan pada *input* gate dapat dilihat pada persamaan 2.2 [46].



Gambar 2.11. *Input* gate LSTM [46].

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.2)$$

Keterangan :

i_t = *input* gate

σ = fungsi sigmoid

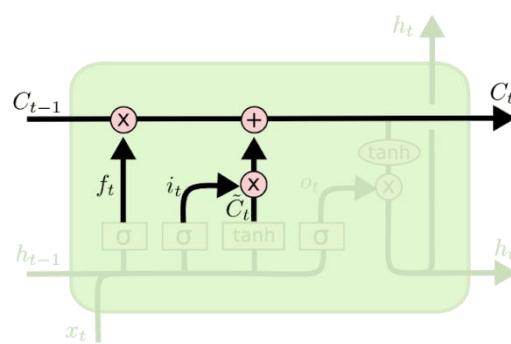
W_i = bobot *input gate*

h_{t-1} = *output* pada *timestep* t-1

x_t = *input* pada *timestep* t

b_i = bias *input gate*

Hasil perhitungan yang diperoleh dari *forget gate* dan *input gate* menjadi komponen yang akan menentukan perubahan dalam *cell state*. Nilai *cell state* dari sel sebelumnya akan dikalikan dengan hasil *forget gate* dan ditambahkan dengan hasil *input gate* untuk menghasilkan nilai *cell state* baru yang akan menjadi salah satu *output* dari sel ini [41]. Ilustrasi perhitungan *cell state* dapat dilihat pada Gambar 2.12. Hasil kalkulasi perubahan pada *cell state* dapat dilihat pada persamaan 2.3 dan 2.4 [46].



Gambar 2.12. Perubahan *cell state* [46].

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.3)$$

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t \quad (2.4)$$

Keterangan :

C'_t = kandidat cell state

\tanh = fungsi tanh

W_C = bobot kandidat cell state

h_{t-1} = output pada timestep t-1

x_t = input pada timestep t

b_C = bias kandidat cell state

C_t = cell state pada timestep t

f_t = forget gate

C_{t-1} = cell state pada timestep t-1

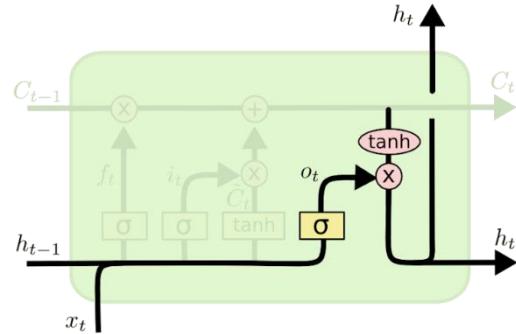
i_t = input gate

Tahap terakhir adalah menentukan *output* dari satu siklus LSTM melalui *output gate*. *Output gate* berfungsi untuk menggabungkan semua informasi yang telah diperoleh demi menghasilkan prediksi yang akurat, serta meneruskan informasi yang terbaru ke sel berikutnya. Perhitungan pada tahap ini menggunakan *output cell state* sebagai *input* ke dalam perhitungan tanh, diikuti dengan perkalian

dengan hasil perhitungan sigmoid [41]. Tahapan *output gate* dapat dilihat pada

Gambar 2.13. Perhitungan hasil akhir dari *output gate* dapat dilihat pada persamaan

2.5 dan 2.6 [46].



Gambar 2.13. *Output Gate* LSTM [46].

$$h_t = o_t \cdot \tanh(C_t) \quad (2.5)$$

Keterangan :

o_t = *output gate*

σ = fungsi sigmoid

W_o = bobot *output gate*

h_{t-1} = *output* pada timestep t-1

x_t = *input* pada timestep t

b_o = bias *output gate*

h_t = *output* pada timestep t

\tanh = fungsi \tanh

C_t = cell state pada *timestep* t

LSTM merupakan arsitektur yang dapat mengekstrak ketergantungan jangka pendek dan jangka panjang dalam semua data sekuensial yang sangat berguna untuk pemrosesan teks. Hal ini dikarenakan LSTM dapat memproses data sekuensial seperti kalimat, paragraf, ataupun URL yang memungkinkan model untuk memahami konteks yang dimaksud dalam sebuah urutan teks [60].

2.2.5. Confusion Matrix

Confusion Matrix adalah alat yang biasanya digunakan dalam *machine learning* untuk mengukur akurasi sebuah model berdasarkan perbandingan nilai hasil prediksi dengan nilai aktual. Matriks ini terdiri dari 4 elemen antara lain [51] :

1. *True Negative (TN)*, yaitu jumlah nilai negatif yang secara tepat diklasifikasi sebagai hasil negatif.
2. *False Negative (FN)*, yaitu jumlah nilai positif yang salah diklasifikasi sebagai hasil negatif.
3. *True Positive (TP)*, yaitu jumlah nilai positif yang secara tepat diklasifikasi sebagai hasil positif.

4. *False Positive (FP)*, yaitu jumlah nilai negatif yang salah diklasifikasi sebagai hasil positif.

		Predicted Class	
		TP	FN
Actual Class	TP		
	FP		TN

Gambar 2.14. Struktur *Confusion Matrix* [47].

Dengan elemen-elemen diatas, confusion matrix dapat digunakan untuk memperoleh beberapa hasil evaluasi, antara lain [44] :

1. *Accuracy* (akurasi), rasio dari jumlah prediksi yang tepat dibandingkan dengan keseluruhan data uji. Untuk mengkalkulasi nilai akurasi, digunakan persamaan 2.7.

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FN+FP} \quad (2.7)$$

2. *Precision* (presisi), sebagian dari nilai yang diambil yang dianggap relevan. *Precision* meliputi semua nilai yang diperoleh dan hanya menggunakan hasil teratas yang dikeluarkan oleh sistem. Untuk mengkalkulasi nilai *precision*, digunakan persamaan 2.8.

$$Precision = \frac{TP}{TP+FP} \quad (2.8)$$

3. *Recall*, sebagian dari nilai relevan yang dijumlahkan dengan nilai yang diambil dan kemudian dibagi dengan nilai relevan . *Recall* juga dikenal sebagai sensitivitas dalam klasifikasi biner. Untuk mengkalkulasi nilai *recall*, digunakan persamaan 2.9.

$$Recall = \frac{TP}{TP+FN} \quad (2.9)$$

4. *F-1 score*, dapat didefinisikan sebagai hasil rata-rata tertimbang dari *precision* dan *recall*. *F-1 score* memiliki nilai tertinggi 1 dan nilai terendah 0 [44]. Fungsi dari *F-1 score* adalah kemampuannya untuk mendeteksi kesalahan yang tidak terdistribusi secara merata [47]. Untuk mengkalkulasi nilai *F-1 score*, digunakan persamaan 2.10 [44].

$$F1 Score = 2 * \frac{Precision*Recall}{Precision+Recall} \quad (2.10)$$

2.2.6. Optimizer

Algoritma optimasi adalah salah satu faktor yang berperan penting dalam machine learning. Algoritma optimasi yang tepat memungkinkan pembelajaran mesin untuk meminimalisir fungsi kerugian melalui perhitungan gradien [58].

2.2.6.1. Adaptive Moment Estimation (Adam)

Adam adalah salah satu algoritma aktivasi yang paling sering digunakan dalam machine learning. Hal ini dikarenakan Adam merupakan teknik optimasi berbasis gradien tingkat pertama yang dapat memperbarui bobot di dalam jaringan secara iteratif berdasarkan data pelatihan [58]. Adam memiliki keunggulan dalam implementasinya yang mudah, efisiensi terhadap daya komputasi, dan memerlukan memori yang rendah. Namun, dengan kecepatannya, Adam masih kurang ideal untuk diimplementasikan dalam aplikasi real-time.

2.2.6.2. Stochastic Gradient Descent (SGD)

Stochastic gradient descent (SGD) juga merupakan salah satu algoritma optimasi yang sering ditemukan dalam algoritma machine learning. SGD bekerja dengan cara memulai dari suatu titik acak dan memperbarui bobot untuk setiap pasangan data pelatihan [58]. Namun SGD memiliki kelemahan yaitu osilasi gradien yang tinggi yang dapat menyebabkan proses pelatihan menjadi semakin lambat.

2.2.7. Fungsi Aktivasi

Fungsi aktivasi adalah komponen yang penting dalam *neural networks* yang menentukan bagaimana output dari setiap neuron akan diproses dan diteruskan ke neuron berikutnya yang memungkinkan jaringan untuk mempelajari dan memahami pola dalam data yang kompleks [59]. Tanpa adanya fungsi aktivasi, model yang

dirancang tidak akan mampu untuk menyelesaikan tugas-tugas seperti pengenalan pola dalam gambar, teks, suara, atau video.

2.2.7.1. Sigmoid

Fungsi aktivasi Sigmoid adalah fungsi aktivasi yang paling sering digunakan dikarenakan sifatnya yang non-linear. Fugnsi sigmoid akan mengubah nilai apapun ke dalam rentang nilai 0 hingga 1 dan dapat diturunkan secara terus menerus, Namun fungsi sigmoid akan mengeluarkan nilai dari setiap neuron dengan tanda yang sama [59].

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.11)$$

2.2.7.2. Tanh

Fungsi aktivasi Tanh mirip dengan fungsi sigmoid. Perbedaan dari keduanya adalah fungsi Tanh dapat menghasilkan tanda keluaran yang berbeda yang neuron sebelumnya dikarenakan fungsi Tanh memiliki simetri terhadap titik asal. Fungsi Tanh juga dapat diturunkan secara terus menerus dan memiliki nilai keluaran dalam rentang -1 hingga 1 [59].

$$f(x) = 2 \cdot \text{sigmoid}(2x) - 1 \quad (2.12)$$

2.2.7.3. Relu

Fungsi ReLu merupakan fungsi aktivasi yang paling sering digunakan dalam neural networks. Fungsi ReLu tidak mengaktifkan semua neuron secara bersamaan yang membuatnya lebih efisien dan unggul dibandingkan fungsi aktivasi lainnya [59].

$$f(x) = \max(0, x) \quad (2.13)$$

2.2.8. Deep Neural Networks (DNN)

Deep neural networks (DNN) adalah model penbelajaran mesin dengan arsitekturnya yang terdiri dari beberapa neuron yang saling terhubung [61]. DNN sering sekali digunakan dalam pemrosesan data tabular atau data terstruktur karena fleksibilitasnya dalam mengatasi masalah ketidakseimbangan kelas dan data. Hal ini membuat DNN memiliki performa yang lebih bagus dalam pemrosesan data tabular dibandingkan beberapa metode tradisional.

Salah satu komponen utama dalam DNN adalah jaringan dense yang sering sekali menjadi komponen inti dalam suatu jaringan DNN. Dense layer sendiri adalah lapisan neuron di dalam DNN yang saling terhubung dengan lapisan sebelumnya, menjadikannya solusi yang tepat untuk tugas klasifikasi yang melibatkan data tabular.

BAB III

RANCANGAN dan PEMBUATAN

3.1. Rancangan Sistem

Program yang dirancang merupakan sebuah aplikasi web yang bertujuan untuk mendeteksi situs phishing menggunakan bantuan *machine learning* [55]. Aplikasi ini menerima *input* berupa URL yang ingin diperiksa keamanannya oleh pengguna. Setelah menerima *input*, akan dilanjutkan pemrosesan menggunakan model LSTM yang sudah dilatih. Setelah proses evaluasi dilakukan oleh model, tahap terakhir yang dilakukan adalah menampilkan hasil prediksi dari model kepada pengguna.

Proses pelatihan model terdiri dari beberapa tahapan penting. Pertama, URL akan melalui tahap pra-pemrosesan berupa normalisasi untuk membersihkan URL menjadi lebih sederhana dan konsisten sehingga lebih mudah diproses oleh model seperti penghapusan protocol “<https://>” dari URL. Selanjutnya, akan dilakukan tokenisasi untuk mengubah setiap karakter yang ada dalam URL menjadi token yang lebih mudah dipahami model. URL yang sudah dinormalisasi dan tokenisasi akan diubah menjadi representasi numerik dengan teknik embedding sehingga lebih mudah dipahami model. URL yang telah melalui tahap pra-pemrosesan ini lah yang akan dijadikan input ke dalam jaringan Long Short Term Memory (LSTM).

Kedua, URL yang diberikan akan melalui tahapan analisis untuk mengekstraksi fitur-fitur yang merupakan karakteristik dari URL yang dapat mengindikasikan potensi phishing [52]. Berikut adalah beberapa fitur yang digunakan dalam pembuatan model.

1. Panjang URL

URL yang panjang dapat dimanfaatkan oleh pelaku phishing untuk menyembunyikan bagian yang meragukan dalam alamat URL. Untuk mengekstraksi fitur ini, cukup dilakukan perhitungan terhadap jumlah karakter dalam URL.

2. Penggunaan IP address sebagai nama domain

Pelaku phishing sering sekali menggunakan alamat IP sebagai alternative nama domain, sehingga alamat situs yang dimulai dengan alamat IP harus diwaspadai. Untuk memeriksa apakah URL menggunakan IP address sebagai nama domain, perlu dilakukan pengecekan terhadap bagian awal dari URL apakah dimulai dengan angka dan memiliki format menyerupai IP address.

3. Tingkat entropi URL

Kalkulasi dilakukan menggunakan rumus Shannon Entropy. Entropi yang tinggi menandakan penggunaan karakter secara acak dalam sebuah URL yang membuatnya terlihat mencurigakan. Shannon entropi dapat dikalkulasikan menggunakan rumus [57]:

$$H(x) = \sum_{i=0}^n p(x_i) \log_b p(x_i) \quad (3.1)$$

4. Mengandung karakter punycode

Karakter punycode adalah karakter Unicode yang diubah kedalam format ASCII.

Karakter punycode bisa digunakan untuk meniru nama domain dari url asli dengan menggunakan karakter yang mirip. Untuk mencari punycode, dilakukan dengan memeriksa prefix “xn--” di dalam URL. Contoh : perubahan karakter ä menjadi xn--4ca.

5. Rasio perbandingan huruf dan angka dalam URL.

URL sah biasanya mengandung huruf yang lebih banyak dibandingkan angka sehingga URL dengan rasio angka yang lebih besar perlu diwaspadai. Rasio ini dihitung dengan membagikan jumlah angka dan jumlah huruf dalam suatu URL.

6. Banyak titik (.) dalam URL

URL yang mengandung banyak titik (.) bisa menandakan subdomain yang banyak sehingga bias dijadikan fitur penting dalam deteksi URL phishing.

7. Banyak simbol at (@) dalam URL

Simbol at (@) biasanya digunakan sebagai pemisah kredensial dengan nama domain dari URL, sehingga jarang ada URL sah yang menggunakan symbol at.

8. Banyak simbol dash (-) dalam URL

Simbol dash (-) jarang sekali digunakan dalam URL sah dan sering digunakan pelaku phishing untuk mengelabui pengguna agar merasa sedang mengakses situs sah.

9. Banyak TLD (Top Level Domain) dalam URL

Banyak TLD dalam sebuah URL dapat dijadikan salah satu fitur yang relevan dalam pendekripsi situs phishing. Beberapa website phishing bisa menggunakan TLD yang tidak umum atau menggunakan lebih dari 1 TLD untuk mengelabui pengguna. Contoh TLD seperti “.com”, “.org”, “.net”.

10. Domain mengandung angka

Situs yang sah jarang sekali memiliki angka di dalam nama domain, sehingga URL yang mengandung angka dalam nama domain bias dianggap mencurigakan.

11. Banyak subdomain dalam URL

Pelaku phishing bisa menggunakan subdomain yang banyak untuk mengelabui korbannya dan memalingkan pandangannya dari domain utama. Untuk menghitung banyak subdomain, pertama dilakukan pemisahan URL terhadap domain utama dan TLD. Kemudian, menghitung berapa banyak segmen yang dipisahkan oleh titik(.) .

12. Entropi karakter non- alfanumerik

Untuk menhitung entropi dari karakter non-alfanumerik, digunakan rumus Shannon entropi yang sama dengan perhitungan fitur entropi URL. Namun, pada fitur ini hanya menggunakan karakter non-alfanumerik dalam kalkulasi.

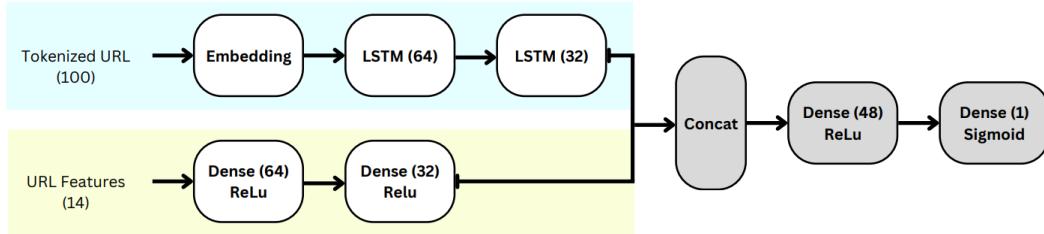
13. Subdirectory URL mengandung tautan

Tautan dalam subdirectory URL dapat dimanfaatkan untuk membingungkan korban dan membuatnya sulit untuk mendeteksi domain yang asli. Untuk mengekstrak fitur ini, cukup dilakukan pengecekan apakah URL mengandung simbol #. Simbol ini biasanya digunakan untuk mengarahkan pengguna ke halaman yang lain.

14. Umur domain

Situs web phishing biasanya hanya hidup selama periode waktu yang singkat sehingga situs dengan umur yang pendek perlu diwaspadai. Untuk mengekstraksi fitur ini dilakukan pengecekan ke database WHOIS, cukup dengan mengimpor library dari WHOIS/.

Fitur-fitur yang sudah diperoleh akan diumpan ke dalam semua jaringan dense layer untuk diolah. Selanjutnya, hasil perhitungan dari semua jaringan LSTM dan dense layer akan digabungkan dan melalui dense layer terakhir untuk dilakukan tahap klasifikasi [53]. Arsitektur dari model yang dirancang dapat dilihat pada Gambar 3.1.



Gambar 3.1. Rancangan arsitektur model pendekripsi URL phishing.

3.1.1. Tahap Perencanaan

Tahap awal yang perlu dilakukan dalam perancangan ini adalah pengumpulan data. Dalam proposal ini, data yang dikumpulkan berupa tautan URL dari beberapa situs, baik situs phishing maupun situs sah. Dataset akan diambil dari, Kaggle suatu platform penyedia dataset yang memberikan akses secara gratis dan PhishTank, sebuah situs komunitas anti-phishing. Selanjutnya, akan dilakukan pra-pemrosesan dan labelisasi terhadap data yang sudah dikumpulkan, yaitu phishing atau sah.

Rancangan arsitektur LSTM akan dibuat menggunakan Google Collab, bahasa pemrograman yang digunakan adalah Phyton, serta menggunakan *library* TensorFlow. Setelah selesai dengan perancangan arsitektur, dilakukan tahap uji coba untuk menemukan model dengan akurasi paling tinggi.

3.1.2. Tahap Analisis

Dalam tahap analisis, dilakukan analisis terhadap dataset dan metode yang digunakan serta perangkat yang digunakan selama perencangan aplikasi. Dataset yang digunakan diambil dari Kaggle dan PhishTank yang akan dibagi untuk tahap pelatihan dan pengujian. Metode yang digunakan untuk proses prediksi adalah LSTM (*Long Short Term Memory*). Berikut spesifikasi dari perangkat keras yang digunakan dalam perancangan aplikasi ini :

1. Laptop dengan processor AMD Ryzen 5 4600H
2. RAM 16 GB
3. SSD 239 GB

Spesifikasi dari perangkat lunak yang digunakan adalah sebagai berikut :

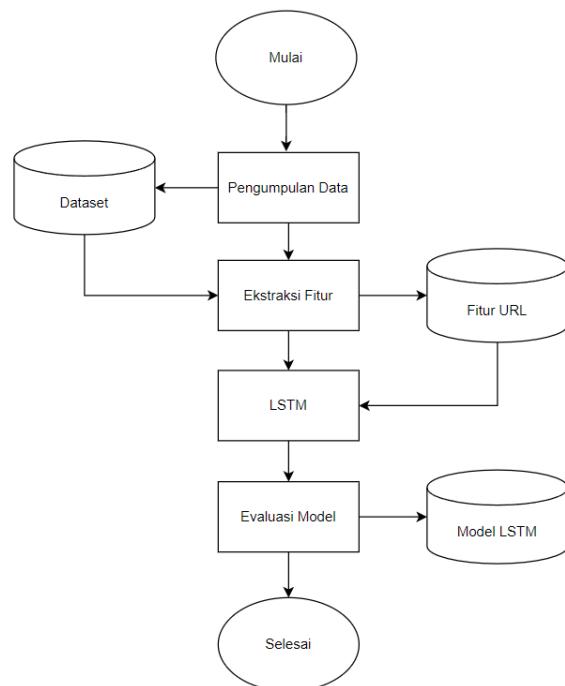
1. Sistem operasi Windows 11
2. Python versi 3.8.5
3. Visual Studio Code versi 1.92.0

3.1.3. Tahap Perancangan

Dalam tahap perancangan, akan dijelaskan menggunakan diagram rancangan modul pelatihan, diagram rancangan modul pengujian, diagram hirarki, *State Transition Diagram* (STD), dan rancangan tampilan *user interface*. Dengan demikian, diharapkan dapat memberikan gambaran yang lebih jelas mengenai program aplikasi yang akan dirancang.

3.1.3.1. Rancangan Modul Pelatihan

Rancangan modul pelatihan menggunakan algoritma LSTM melibatkan proses pengumpulan data, ekstraksi fitur, hingga inisialisasi model LSTM. Proses ini terus diulangi hingga ditemukan kombinasi parameter pada model LSTM yang memiliki akurasi tertinggi.

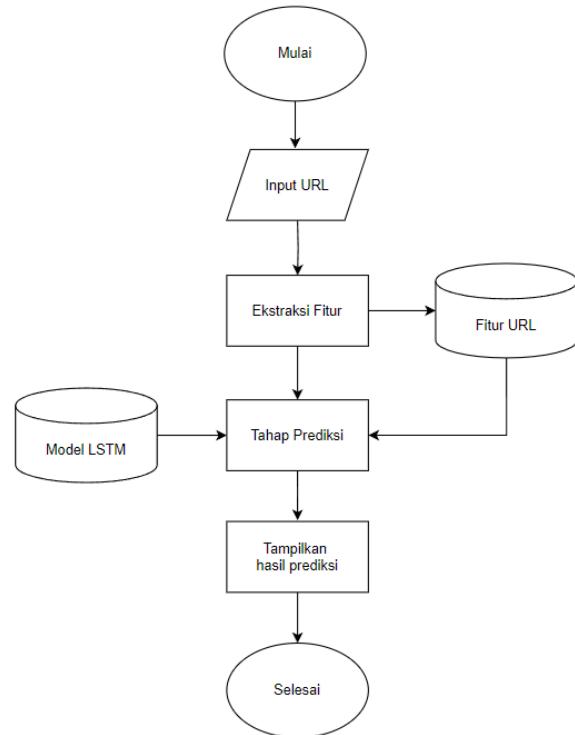


Gambar 3.2. *Flowchart* tahap pelatihan model.

3.1.3.2. Rancangan Modul Pengujian

Rancangan modul pengujian melibatkan penggunaan data yang belum digunakan dalam tahap pelatihan model dan evaluasi model sebagai *input*. URL yang

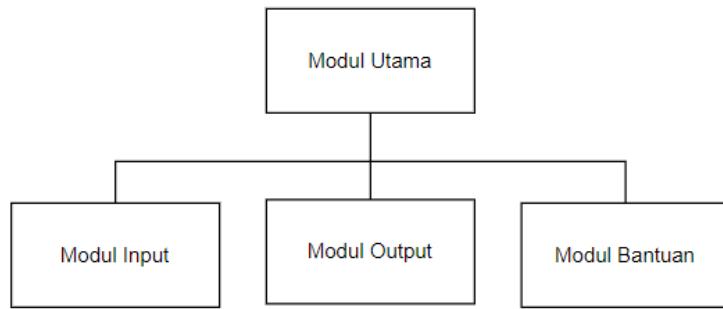
diinput tetap akan melalui tahap ekstraksi fitur sebelum dilakukan prediksi menggunakan model LSTM hasil pelatihan.



Gambar 3.3. *Flowchart* tahap pengujian model.

3.1.3.3. Rancangan Diagram Hirarki

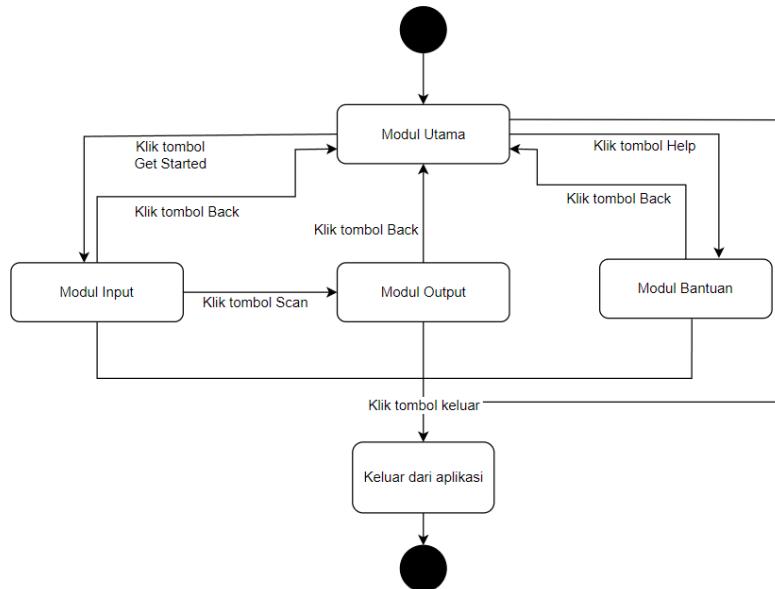
Diagram hirarki memberikan informasi mengenai fitur dan modul apa saja yang dimiliki program yang akan dirancang. Program aplikasi ini akan memiliki empat modul, yaitu modul utama, modul *input*, modul *result*, dan modul bantuan.



Gambar 3.4. Rancangan Diagram Hirarki.

3.1.3.4. Rancangan State Transition Diagram

State Transition Diagram berfungsi untuk menjelaskan mengenai hubungan dan transisi dari masing-masing modul yang ada dalam suatu sistem. Dengan adanya diagram ini dapat memberikan gambaran mengenai proses yang terjadi dalam suatu sistem.



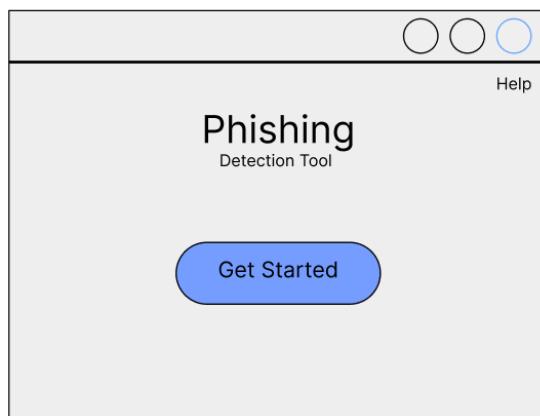
Gambar 3.5. Rancangan *State Transition Diagram*.

3.1.3.5. Rancangan Antarmuka

Rancangan antarmuka bertujuan untuk dijadikan pedoman dalam perancangan sistem dan memberikan pengguna gambaran mengenai tampilan dari program aplikasi. Berikut rancangan antarmuka dari aplikasi.

1. Rancangan Modul Utama

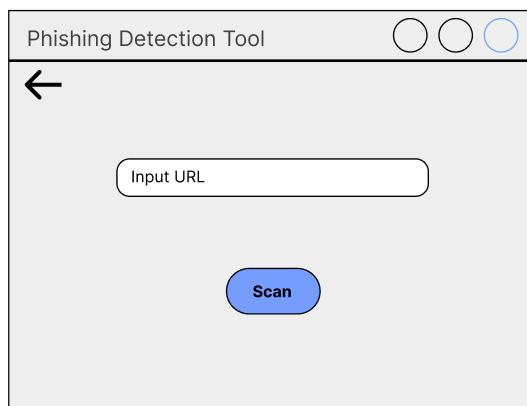
Modul utama merupakan tampilan yang akan dilihat pengguna ketika pertama kali membuka aplikasi. Modul ini berisi daftar menu apa saja yang disediakan dalam aplikasi ini. Rancangan modul utama dapat dilihat di Gambar 3.6.



Gambar 3.6. Rancangan modul utama.

2. Rancangan Modul *Input*

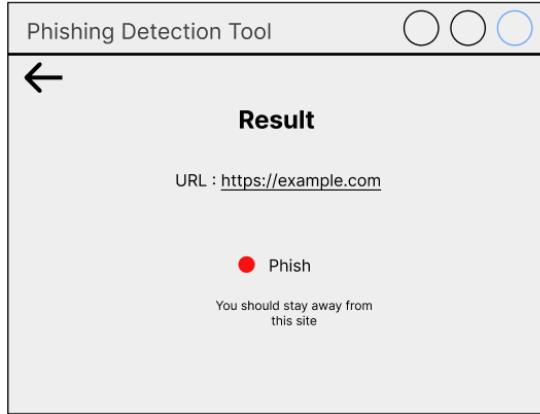
Modul ini merupakan tempat pengguna melakukan *input* tautan URL yang ingin dilakukan pengecekan. Rancangan modul *input* dapat dilihat di Gambar 3.7.



Gambar 3.7. Rancangan modul *input*.

3. Rancangan Modul *Output*

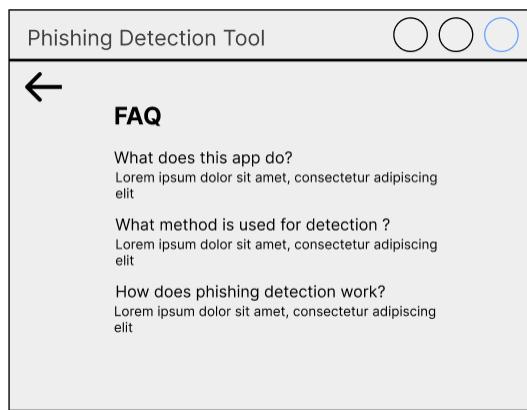
Modul *output* menunjukkan hasil yang diperoleh oleh rancangan sistem setelah data *input* telah diproses dan diklasifikasikan. Rancangan modul *output* dapat dilihat di Gambar 3.8.



Gambar 3.8. Rancangan modul *output*.

4. Rancangan Modul Bantuan

Modul bantuan berfungsi untuk menjawab pertanyaan yang mungkin dimiliki pengguna, sekaligus memberikan panduan terhadap cara pemakaian program aplikasi. Rancangan modul bantuan dapat dilihat di Gambar 3.9.



Gambar 3.9. Rancangan modul bantuan

3.2. Pembuatan Sistem

Pembuatan sistem pendekripsi URL phishing melibatkan penggunaan perangkat keras dan perangkat lunak. Perangkat keras yang digunakan dalam pembuatan sistem aplikasi ini adalah:

1. ASUS Vivobook
2. Laptop dengan processor AMD Ryzen 5 4600H
3. RAM 16 GB
4. SSD 239 GB

Perangkat lunak yang digunakan dalam pembuatan sistem aplikasi adalah :

1. Sistem Operasi Windows 10
2. Visual Studio Code
3. Google Collab Notebook
4. Python 3.8.5
5. Tensorflow 2.11.0

3.2.1. Pembuatan Modul Pelatihan dan Evaluasi

Pembangunan model pelatihan menggunakan Google Collab Notebook dan bahasa pemrograman Python. Model yang dibangun berbasis LSTM untuk proses analisis URL dan beberapa jaringan dense yang berfungsi untuk mempelajari fitur-fitur non-sekuensial yang sudah diekstrak. Hasil dari kedua tahap pemrosesan ini digabungkan untuk melalui tahap klasifikasi akhir. Tahap evaluasi model dilakukan

dengan confusion matrix dan grafik loss untuk memperoleh tingkat akurasi dari model yang dibuat.

3.2.2. Pembuatan Antarmuka

Pembuatan antarmuka dilakukan menggunakan HTML, CSS, dan JavaScript dan dibangun menggunakan aplikasi Visual Studio Code. HTML digunakan untuk merancang kerangka dan elemen-elemen dalam situs web, CSS untuk membuat visual dari situs web lebih menarik, dan JavaScript yang memungkinkan interaksi antar modul berjalan lancar.

BAB IV

PENGUJIAN

4.1. Metode Pengujian

Pengujian sistem aplikasi dilakukan untuk memastikan seluruh fungsi dan fitur dalam aplikasi berfungsi sesuai dengan harapan dan telah memenuhi kriteria yang telah ditetapkan. Dengan melakukan pengujian yang menyeluruh, diharapkan dapat mengidentifikasi *bugs* atau *error* yang perlu diatasi sebelum aplikasi dirilis untuk digunakan publik, demi menjaga kenyamanan pengguna.

Metode pengujian yang digunakan adalah metode *black box testing*. Metode ini adalah salah satu metode pengujian perangkat lunak yang cukup umum digunakan. Metode *black box testing* merupakan metode pengujian yang berfokus ke fungsionalitas, dimana pengujian hanya dilakukan terhadap input dan output dari aplikasi tanpa memperhatikan coding di dalam aplikasi, sehingga penguji tidak memerlukan pemahaman yang dalam mengenai struktur kode atau logika pemrograman di dalam aplikasi. Dengan menggunakan metode pengujian ini, dapat diperoleh penilaian aplikasi berdasarkan sudut pandang pengguna.

Dalam tahapan pengujian ini, akan menggunakan data dari 2 kelas yang berbeda dan telah dilabelisasi, yaitu sah dan phishing. Data ini akan dijadikan input kedalam aplikasi untuk dilakukan evaluasi terhadap keamanan situs. Hasil output yang diharapkan dari pengujian aplikasi ini adalah hasil prediksi yang akurat untuk

keseluruhan data uji, yang mencerminkan kemampuan aplikasi untuk mengklasifikasikan URL dengan tepat.

4.2. Hasil Pengujian

Tahap pengujian aplikasi website pendekripsi URL phishing, dibagi menjadi 3 bagian, yaitu, pengujian terhadap modul, pengujian model, dan pengujian data.

4.2.1. Pengujian Terhadap Modul

Pertama pengujian modul untuk memastikan semua fungsi dan fitur di dalam program berjalan dengan lancar dan sesuai dengan yang diharapkan. Dalam aplikasi website pendekripsi situs phishing, terdapat 4 modul, yaitu modul utama, modul input, modul output, dan modul bantuan.

4.2.1.1. Pengujian Modul Utama

Modul Utama merupakan tampilan pertama yang akan dilihat pengguna ketika membuka program aplikasi. Pada modul ini terdapat 2 tombol yang mengarahkan ke halaman input dan halaman bantuan. Selain itu, modul utama juga menjelaskan secara singkat tahapan yang perlu dilakukan untuk menggunakan program aplikasi.

Hasil pengujian terhadap modul utama dapat dilihat pada Gambar L4.1 Lampiran 3.

4.2.1.2. Pengujian Modul Input

Modul Input merupakan tempat pengguna memasukkan URL yang ingin divalidasi. Pada modul ini terdapat 1 buat text box yang berfungsi sebagai tempat mengetikkan URL. Selanjutnya, terdapat tombol scan yang akan mengecek

keamanan dari URL menggunakan model yang sudah dilatih dan mengarahkan pengguna ke halaman output. Pada modul ini, juga terdapat tombol yang bisa mengarahkan pengguna ke modul bantuan. Hasil pengujian terhadap modul input dapat dilihat pada Gambar L4.2 Lampiran 3.

4.2.1.3. Pengujian Modul Output

Modul Output merupakan halaman dimana pengguna dapat melihat hasil prediksi dari URL yang dimasukkan. Pada modul ini, juga terdapat tombol yang mengarahkan pengguna kembali ke modul input apabila ingin melakukan validasi terhadap URL lainnya dan tombol untuk keluar ke halaman utama. Selain itu, modul output juga memiliki tombol yang bisa mengarahkan pengguna ke modul bantuan. Hasil pengujian terhadap modul output dapat dilihat pada Gambar L4.3 dan Gambar L4.4 Lampiran 3.

4.2.1.4. Pengujian Modul Bantuan

Modul Bantuan berisikan jawaban terhadap beberapa pertanyaan yang mungkin dimiliki oleh pengguna. Dengan adanya modul ini diharapkan bisa membantu pengguna untuk menyadari kegunaan dari aplikasi ini dan cara pemakaiannya. Pada modul ini terdapat 1 tombol yang mengarahkan ke modul utama. Hasil pengujian terhadap modul bantuan dapat dilihat pada Gambar L4.5 Lampiran 3.

4.2.2. Pengujian Model LSTM

Pelatihan dan pengujian model LSTM dilakukan dengan dataset yang diperoleh dari Kaggle. Contoh data pengujian model dapat diihat dapat Lampiran 2. Dari dataset yang telah diperoleh, dapat diperoleh kesimpulan bahwa beberapa fitur seperti panjang URL dan tingkat entropi memiliki relevansi yang tinggi terhadap keamanan situs. Distribusi dari setiap fitur dalam dataset dapat dilihat pada Lampiran 4.

Tahap pengujian diikuti dengan hyperparameter tuning untuk mencari kombinasi model dengan akurasi tertinggi. Pengujian hyperparameter tuning pertama dilakukan dengan mengubah epoch dan maksimal panjang karakter yang digunakan untuk pemrosesan *string* URL. Confusion matix dan grafik loss untuk setiap model dapat dilihat pada Lampiran 8. Hasil pengujian dari 6 variasi model dapat dilihat pada Tabel 4.1.

Tabel 4.1. Pengujian hyperparameter tuning berdasarkan panjang URL dan jumlah epoch.

Model	Epoch	Maksimal Panjang URL	Akurasi Latih	Loss Latih	Akurasi Validasi	Loss Validasi
Model A	40	80	99,95%	0,23%	99,93%	0,25%
Model B	60	80	99,96%	0,13%	99,97%	0,13%
Model C	80	80	99,97%	0,08%	99,94%	0,01%
Model D	40	100	99,95%	0,01%	99,93%	0,01%
Model E	60	100	99,94%	0,20%	99,95%	0,21%
Model F	80	100	99,98%	0,07%	99,98%	0,07%
Model G	40	150	99,98%	0,08%	99,97%	0,03%
Model H	60	150	99,94%	0,18%	99,98%	0,06%
Model I	80	150	99,98%	0,05%	99,95%	0,01%

Confusion matix dan grafik loss untuk setiap model dapat dilihat pada Lampiran 5. Hasil pengujian dari 6 variasi model dapat dilihat pada Tabel 4.2. Semua pengujian model menggunakan struktur jaringan yang sama dengan menggabungkan LSTM dan dense layer.

Tabel 4.2. Pengujian hyperparameter tuning berdasarkan optimizer dan fungsi aktivasi.

Model	Optimizer	Fungsi Aktivasi	Akurasi Latih	Loss Latih	Akurasi Validasi	Loss Validasi
Model 1	Adam	Relu	99,94%	0,20%	99,95%	0,21%
Model 2	SGD	Tanh	96,90%	10,12%	97,00%	9,58%
Model 3	Adam	Sigmoid	99,79%	0,59%	98,63%	6,86%
Model 4	SGD	Relu	92,76%	19,11%	93,47%	17,92%
Model 5	Adam	Tanh	99,69%	0,96%	98,04%	9,97%
Model 6	SGD	Sigmoid	97,01%	9,98%	97,00%	10,21%

Dari tabel diatas dapat dilihat bahwa kombinasi optomizer Adam dengan penggunaan aktivasi Relu dalam dense layer memiliki akurasi tertinggi ketika dilatih menggunakan dataset yang sama yaitu sebesar 99.95%. Nilai akurasi, presisi, recall, dan F-1 Score dari setiap model dapat dilihat pada Tabel 4.3.

Tabel 4.3. Hasil klasifikasi model.

Model	Akurasi	Presisi	Recall	F-1 Score
Model 1	0,9995	0,9996	0,9994	0,9995
Model 2	0,96995	0,9539	0,9878	0,9705
Model 3	0,98635	0,9837	0,9891	0,9864
Model 4	0,93465	0,9154	0,9582	0,9363
Model 5	0,98045	0,9769	0,9842	0,9805

Selain melakukan pengujian model dengan teknik hyperparameter tuning, telah dilakukan pengujian untuk mengukur performa model ketika hanya menggunakan LSTM untuk pemrosesan URL dan ketika hanya menggunakan dense layer untuk pemrosesan fitur URL. Pengujian ini bertujuan untuk mengevaluasi seberapa besar kontribusi dari setiap komponen terhadap akurasi dari keseluruhan model. Hasil pelatihan dan validasi dari setiap arsitektur model dapat dilihat pada Lampiran 6. Perbandingan akurasi untuk setiap arsitektur dapat dilihat pada Tabel 4.3.

Tabel 4.3. Perbandingan akurasi model berdasarkan arsitektur.

Model	Dataset	Akurasi Validasi
LSTM + Dense	Kaggle	99,95%
LSTM	Kaggle	95,67%
Dense	Kaggle	99,75%
LSTM + Dense	Ebbu2017	52,03%
LSTM	Ebbu2017	50,35%
Dense	Ebbu2017	50,53%

Selain itu dilakukan juga perbandingan terhadap model pertama ketika divalidasi menggunakan dataset Ebbu2017. Dataset Ebbu2017 diproses tanpa menggunakan fitur umur domain, dikarenakan semua URL dalam dataset tersebut tidak lagi aktif sehingga umur dari domain tidak bisa didapatkan dan diisi dengan nilai -1. Perbandingan pelatihan dan pengujian dari kedua dataset dapat dilihat pada Tabel 4.4.

Tabel 4.4. Perbandingan akurasi dataset Ebbu2017.

Model	Dataset	Jumlah Data	Label Sah	Label Phishing	Akurasi Validasi
Model 1	Kaggle	20.000	10.035	9.965	99,95%
Model 1	Ebbu2017	73.575	36.400	37.175	52,03%

Dari perbandingan validasi menggunakan dataset Ebbu2017, diperoleh akurasi yang sangat rendah untuk klasifikasi URL sah. Kesimpulan yang bisa didapatkan adalah fitur umur dari domain dapat berpengaruh dalam akurasi pendekripsi URL phishing. Confusion matrix dari pengujian model menggunakan dataset Ebbu2017 dapat dilihat pada Lampiran 7.

4.2.3. Pengujian Terhadap Output Program

Pengujian output program dilakukan dengan cara memasukkan beberapa URL ke dalam hasil program yang dijalankan dan kemudian dilakukan pendekripsi. Data yang digunakan pada pengujian ini menggunakan data yang telah dilabelisasi

ke dalam 2 kelas, yaitu phishing dan sah (legitimate). URL sah dikumpulkan sendiri, menggunakan beberapa URL yang umum untuk digunakan, sedangkan URL phishing dikumpulkan dari PhishTank. Berikut beberapa contoh data dan hasil pengujian output program.

Tabel 4.5. Hasil pengujian output program.

No.	URL	Label	Hasil
1.	https://www.amazon.com	Legitimate	Legitimate
2.	https://www.wikipedia.com	Legitimate	Legitimate
3.	https://www.linkedin.com	Legitimate	Legitimate
4.	https://www.instagram.com	Legitimate	Legitimate
5.	https://www.kemdikbud.co.id	Legitimate	Legitimate
6.	https://www.bca.co.id	Legitimate	Legitimate
7.	https://evisa.imigrasi.go.id	Legitimate	Legitimate
8.	https://www.cnnindonesia.com	Legitimate	Legitimate
9.	https://www.pssi.org	Legitimate	Legitimate
10.	https://untar.ac.id/	Legitimate	Legitimate
11.	https://magenta312505.studio.site	Phishing	Legitimate
12.	https://x21002.cn/index.php	Phishing	Phishing
13.	https://lmlcas.com/	Phishing	Legitimate
14.	https://tinyurl.com/29d2vzfk	Phishing	Phishing
15.	http://trezarsuite-suite.com	Phishing	Phishing
16.	https://yumegeurio.com/	Phishing	Legitimate
17.	https://sunhairrr.com/	Phishing	Legitimate
18.	https://blue518017.studio.site/	Phishing	Legitimate
19.	https://ing-es-ayuda.com	Phishing	Phishing
20.	https://blocksdappmain.vercel.app/syncwallets	Phishing	Phishing

Dalam pengujian output program, dilakukan juga pengujian untuk membandingkan perfoma antara program yang dirancang dengan program yang sudah ada. Perbandingan ini dilakukan terhadap situs Indonesia Cyber Crime

Combat Center menggunakan beberapa URL phishing yang diperoleh dari PhishTank.

Tabel 4.6. Perbandingan pengujian dengan program tersedia.

No.	URL	Program yang dirancang	Indonesia Cyber Crime Combat Center
1.	https://begin-trizor.github.io/en-us/	Phishing	Legitimate
2.	https://lidl-vins.shop/	Phishing	Legitimate
3.	https://p.updateinfof.top/us/	Phishing	Legitimate
4.	https://logistique-de-livraison.com/index.php	Phishing	Legitimate
5.	https://recodatii.com/s/s/swpass/	Phishing	Legitimate
6.	http://www.jollyolesoul.com/nie/	Phishing	Legitimate
7.	https://tabamweg.sviluppo.host/old/	Phishing	Legitimate
8.	https://icloud.apple-cond.cvequ.cn/accountin/	Phishing	Phishing
9.	https://trustwallt.bbk.net.au/eng/	Phishing	Legitimate
10.	https://tabamweg.sviluppo.host/old/	Phishing	Legitimate
11.	https://supports-client.com/login.php	Phishing	Legitimate
12.	https://docusign-docu.vercel.app/	Phishing	Phishing
13.	https://mobiib-facebook.click/	Phishing	Legitimate
14.	https://berbagiewallet.vercel.app/Com1	Phishing	Legitimate
15.	https://correo-para-gu.help/	Phishing	Legitimate
16.	https://estafeta-postal.cc/	Phishing	Legitimate
17.	https://vrl-sg0623.top/	Phishing	Legitimate
18.	https://www.inpost-c7345.top/	Phishing	Legitimate
19.	https://zajel-eg.top/	Phishing	Legitimate
20.	https://80-76-51-55.crapid.com/	Phishing	Legitimate

4.3. Pembahasan

4.3.1. Pembahasan Pengujian Modul

Hasil pengujian modul menggunakan metode black box testing menunjukkan bahwa setiap fitur di dalam program aplikasi berjalan dengan baik dan sesuai dengan yang diharapkan. Pada pengujian ini, seluruh fitur dalam aplikasi sudah diuji, mulai dari tombol-tombol yang mengarahkan pengguna ke halaman lain di program aplikasi, tombol scan untuk mengecek keamanan URL, hingga fungsionalitas textbox tempat pengguna memasukkan URL yang ingin divalidasi. Perpindahan antar halaman juga cukup responsif sehingga tidak memerlukan waktu pemuatan yang lama. Secara keseluruhan, pada pengujian model tidak ditemukan error atau bugs yang dapat mengganggu pengalaman pengguna.

4.3.2. Pembahasan Pengujian Model LSTM

Hasil pengujian model LSTM menggunakan teknik hyperparameter tuning untuk mencari kombinasi terbaik dengan akurasi tertinggi. Pengujian hyperparameter pertama menghasilkan 9 kombinasi model dengan jumlah epoch pelatihan dan panjang maksimal URL untuk pemrosesan menggunakan LSTM. Dari pengujian ini dapat dilihat bahwa pengujian dengan maksimal panjang URL antara 100 sampai 150 dengan jumlah epoch pelatihan sebanyak 60 sampai 80 memiliki performa yang lebih bagus.

Pada pengujian hyperparameter tuning kedua, diperoleh 6 buah model dengan kombinasi optimizer dan fungsi aktivasi yang berbeda. Detail dari kombinasi optimizer dan fungsi aktivasi untuk setiap model dapat dilihat pada Tabel 4.2. Dari hasil pengujian menggunakan epoch yang sama, dapat dilihat bahwa model dengan kombinas optimizer adam dann fungsi aktivasi Relu memiliki performa paling bagus dibandingkan model lainnya dengan akurasi sebesar 99,95%, presisi sebesar 0,9996, recall sebesar 0,9994 dan F-1 Score sebesar 0,9995 ketika dilakukan validasi menggunakan data yang diperoleh dari Kaggle.

Selain pengujian menggunakan teknik hyperparameter tuning, dilakukan juga pengujian menggunakan arsitektur yang berbeda. Pengujian ini dilakukan untuk membandingkan akurasi dari model yang dirancang dengan arsitektur model yang hanya menggunakan LSTM untuk pemrosesan URL dan arsitektur model yang hanya menggunakan jaringan dense untuk pemrosesan fitur-fitur dari URL. Pengujian ini dilakukan untuk mengevaluasi relevansi dari masing-masing komponen dalam arsitektur model terhadap akurasi dari model itu sendiri. Dari percobaan ini diperoleh akurasi terbesar ketika model dilatih menggunakan gabungan dari kedua komponen tersebut.

Selanjutnya, dilakukan juga pengujian model 1 yang menggunakan optimizer Adam dan fungsi aktivasi ReLu, terhadap dataset Ebbu2017 yang diperoleh dari salah satu referensi penelitian. Dataset Ebbu dilatih tanpa menggunakan fitur umur

domain dikarenakan URL yang disediakan pada dataset ini tidak disediakan nilai dari fitur tersebut sehingga perlu dilakukan ekstraksi secara manual. Namun, dikarenakan seluruh URL yang disediakan dalam dataset merupakan URL yang tidak lagi aktif, maka tidak dapat dilakukan ekstraksi terhadap umur dari domain, sehingga semua validasi menggunakan dataset ini diberikan nilai -1 untuk fitur umur domain. Perbandingan dari model 1 ketika dilatih dan diuji menggunakan 2 jenis dataset dengan pengulangan sebanyak 60 epoch, menunjukkan bahwa pengujian menggunakan dataset yang diperoleh dari kaggle memiliki akurasi yang jauh lebih baik dibandingkan pengujian dengan dataset Ebbu2017. Dari pengujian ini dapat disimpulkan bahwa penggunaan fitur umur domain dapat mempengaruhi akurasi dari model.

4.3.3. Pembahasan Pengujian Output Program

Hasil pengujian terhadap output program menunjukkan persentasi kebenaran sebesar 75%. Dari pengujian ini bisa dilihat bahwa program berhasil mendeteksi semua URL sah secara tepat. Namun, ketika digunakan untuk mengvalidasi URL phishing, hanya diperoleh akurasi sebesar 50%. Melihat dari hasil pelatihan dan pengujian model, fitur yang memiliki peran paling besar dalam memprediksi URL phishing adalah fitur entropi. Hasil pengujian output program yang kurang memuaskan dapat disebabkan oleh banyaknya data pelatihan yang dilabelisasi sebagai phishing memiliki tingkat entropi yang tinggi dan kurangnya data

URL phishing yang memiliki tingkat entropi rendah, sehingga dalam pengujian output program menggunakan URL memiliki jumlah karakter sedikit dan entropi rendah gagal diprediksi secara akurat.

Pada pengujian ini juga dilakukan perbandingan akurasi dengan salah satu program yang sudah tersedia yaitu Indonesia Cyber Crime Combat Center menggunakan dataset yang diperoleh dari PhishTank dimana dapat dilihat bahwa program yang dirancang memiliki tingkat akurasi yang jauh mengungguli aplikasi lainnya, sehingga dapat disimpulkan bahwa program yang dirancang dapat menjadi opsi yang signifikan dalam mendeteksi situs phishing.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil perancangan dan pengujian model machine learning pendekripsi URL Phishing menggunakan arsitektur gabungan LSTM dan dense layer, dapat disimpulkan bahwa model hasil rancangan ini mampu menganalisa dan mengenali pola di dalam URL serta beberapa fitur tambahan seperti panjang URL, tingkat entropi dan umur domain untuk mendekripsi ancaman phishing dari URL.

Hasil pengujian pertama menunjukkan bahwa model yang dirancang memiliki performa yang lebih baik apabila dilatih menggunakan maksimal panjang URL sepanjang 100 sampai 150 karakter dan jumlah pelatihan sebanyak 60 sampai 80 epoch, dengan akurasi tertinggi 99,98% pada kombinasi pelatihan sebanyak 60 epoch dengan panjang URL sebanyak 150 karakter dan 80 epoch dengan panjang url sebanyak 100 karakter.

Hasil pengujian ketiga, menunjukkan performa yang cukup bagus ketika dilatih menggunakan kombinasi optimizer Adam dan fungsi aktivasi Relu dalam pemrosesan fitur dengan dense layer dengan akurasi validasi sebesar 99.95%, presisi sebesar 0,9996, recall sebesar 0,9994, dan F-1 Score sebesar 0,9995 pada dataset yang diperoleh dari Kaggle. Namun, terlihat penurunan performa menjadi ketika model divalidasi menggunakan dataset Ebbu2017. Penurunan dalam akurasi

ini dapat disebabkan oleh fitur umur domain dari dataset Ebbu2017 yang tidak bisa diekstrak.

Pada pengujian keempat dilakukan perbandingan antara aplikasi yang dirancang dengan program Indonesia Cyber Crime Combat Center yang menunjukkan bahwa program yang dirancang sangat unggul dalam akurasi pengujian menggunakan data PhishTank. Hal ini membuktikan bahwa program yang dirancang dapat menjadi opsi yang signifikan dalam mendeteksi situs phishing

Pengujian modul antarmuka pada aplikasi menggunakan metode black box testing menunjukkan bahwa seluruh fitur dan tombol di dalam aplikasi berjalan lancar dan sesuai dengan harapan tanpa ada penemuan eror atau bug yang bisa mengganggu pengalaman pengguna. Namun, pengujian terhadap output dari program menunjukkan akurasi dalam mendeteksi URL phishing hanya sekitar 50% yang dinilai kurang memuaskan, meskipun pengujian dalam mendeteksi URL sah diperoleh akurasi sebesar 100%. Dari pengujian ini dapat disimpulkan bahwa variasi dataset yang digunakan dalam tahap pelatihan model sangatlah berpengaruh terhadap hasil prediksi, melihat dari rata-rata data pelatihan yang memiliki tingkat entropi tinggi untuk URL phishing.

5.2. Saran

Saran yang dapat diberikan untuk peningkatan akurasi dan penelitian berikutnya, antara lain :

1. Mengumpulkan dataset URL phishing yang lebih beragam untuk memastikan semua fitur yang digunakan dapat ditemukan dalam dataset URL.
2. Mengeksplorasi beberapa arsitektur lainnya yang lebih modern seperti transformer yang dikenal dalam kemampuannya untuk mengenali pola yang kompleks.
3. Menambahkan beberapa fitur lainnya yang relevan dan dapat membantu mengenali URL phishing, atau bahkan melakukan scraping untuk menemukan kejanggalan dalam situs.

DAFTAR PUSTAKA

- [1] Santoso, Sugeng, "Memperkuat Pertahanan Siber Guna Meningkatkan Ketahanan Nasional," *Jurnal Lemhannas RI*, vol. 6, pp. 43-48, 2018.
- [2] Harjinder Singh Lallie and Lynsay A. Shepherd and Jason R.C. Nurse and Arnau Erola and Gregory Epiphaniou and Carsten Maple and Xavier Bellekens, "Cyber security in the age of COVID-19: A timeline and analysis of cyber-crime and cyber-attacks during the pandemic," *Computers & Security*, vol. 105, p. 102248, 2021.
- [3] Catal, Cagatay and Giray, G{"o}rkem and Tekinerdogan, Bedir and Kumar, Sandeep and Shukla, Suyash, "Applications of deep learning for phishing detection: a systematic literature review," *Knowledge and Information Systems*, vol. 64, pp. 1457-1500, 2022.
- [4] Shahrivari, Vahid and Darabi, Mohammad Mahdi and Izadi, Mohammad, "Phishing detection using machine learning techniques," *arXiv preprint arXiv:2009.11116*, 2020.
- [5] "7 Tips for How to Spot Email Phishing," Cofense, 6 June 2023. [Online]. Available: <https://cofense.com/knowledge-center/how-to-spot-phishing/>. [Accessed 31 August 2024].

- [6] Carroll, Fiona, "How Good Are We at Detecting a Phishing Attack? Investigating the Evolving Phishing Attack Email and Why It Continues to Successfully Deceive Society," *SN Computer Science*, vol. 3, no. 2, p. 170, 2022.
- [7] Sheng, Steve and Wardman, Brad and Warner, Gary and Cranor, Lorrie and Hong, Jason and Zhang, Chengshan, "An empirical analysis of phishing blacklists," 2009.
- [8] Y. Su, "Research on website phishing detection based on LSTM RNN," vol. 1, pp. 284-288, 2020.
- [9] Roy, Sanjiban Sekhar and Awad, Ali Ismail and Amare, Lamesgen Adugnaw and Erkihun, Mabrie Tesfaye and Anas, Mohd, "Multimodel phishing url detection using lstm, bidirectional lstm, and gru models," *Future Internet*, vol. 14, p. 340, 2022.
- [10] Alshingiti, Zainab and Alaquel, Rabeah and Al-Muhtadi, Jalal and Haq, Qazi Emad UI and Saleem, Kashif and Faheem, Muhammad Hamza, "A deep learning-based phishing detection system using CNN, LSTM, and LSTM-CNN," *Electronics*, vol. 12, p. 232, 2023.
- [11] Balogun, Abdullateef O and Adewole, "Improving the phishing website detection using empirical analysis of Function Tree and its variants," *Heliyon*, vol. 7, 2021.

- [12] Xiao, Xi and Xiao, Wentao and Zhang, Dianyan and Zhang, Bin and Hu, Guangwu and Li, Qing and Xia, Shutao, "Phishing websites detection via CNN and multi-head self-attention on imbalanced datasets," *Computers & Security*, vol. 108, p. 102372, 2021.
- [13] Yerima, Suleiman Y and Alzaylaee, Mohammed K, "High Accuracy Phishing Detection Based on Convolutional Neural Networks," pp. 1-6, 2020.
- [14] Smadi, Sami and Aslam, Nauman and Zhang, Li and Alasem, Rafe and Hossain, M Alamgir, "Detection of phishing emails using data mining algorithms," pp. 1-8, 2015.
- [15] "Utilisation of website logo for phishing detection," *Computers & Security*, vol. 54, pp. 16-26, 2015.
- [16] Jonathan, Kevin Marcello and Mulyawan, Bagus and Perdana, Novario Jaya, "PERBANDINGAN KINERJA ALGORITMA NAIVE BAYES DAN C4. 5 UNTUK MENDETEKSI PENGELABUAN UNIFORM RESOURCE LOCATOR (PHISHING URL)," *Jurnal Ilmu Komputer dan Sistem Informasi*, vol. 8, pp. 116-120, 2020.
- [17] Wen, Zikai Alex and Lin, Zhiqiu and Chen, Rowena and Andersen, Erik, "What.Hack: Engaging Anti-Phishing Training Through a Role-playing Phishing Simulation Game," pp. 1-12, 2019.
- [18] Sumner, Alex and Yuan, Xiaohong, "Mitigating phishing attacks: an overview,"

pp. 72-77, 2019.

- [19] S. Purkait, "Phishing counter measures and their effectiveness--literature review," *Information Management & Computer Security*, vol. 20, pp. 382-420, 2012.
- [20] Caputo, Deanna D and Pfleeger, Shari Lawrence and Freeman, Jesse D and Johnson, M Eric, "Going spear phishing: Exploring embedded training and awareness," *IEEE security & privacy*, vol. 12, pp. 28--38, 2013.
- [21] Shankar, Akarshita and Shetty, Ramesh and Nath, B, "A review on phishing attacks," *International Journal of Applied Engineering Research*, vol. 14, p. 5, 2019.
- [22] R. Alabdani, "Phishing attacks survey: Types, vectors, and technical approaches," *Future internet*, vol. 12, p. 168, 2020.
- [23] Yeboah-Boateng, Ezer Osei and Amanor, Priscilla Mateko, "Phishing, SMiShing & Vishing: an assessment of threats against mobile devices," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 5, pp. 297-307, 2014.
- [24] Guarda, Teresa and Augusto, Maria Fernanda and Lopes, Isabel, "The art of phishing," pp. 683-690, 2019.
- [25] Bruno, Vince and Tam, Audrey and Thom, James, "CHARACTERISTICS OF WEB

APPLICATIONS THAT AFFECT USABILITY: A REVIEW," pp. 1-4, 2005.

- [26] Finifter, Matthew, "Exploring the relationship between web application development tools and security," 2011.
- [27] N. Dordevic, "Evaluation of the usability of Web-based applications," *Vojnotehnicki glasnik/Military Technical Courier*, vol. 65, pp. 785-802, 2017.
- [28] Chauhan, Nitin Kumar and Singh, Krishna, "A review on conventional machine learning vs deep learning," pp. 347-352, 2018.
- [29] Bartlett, Peter L and Montanari, Andrea and Rakhlin, Alexander, "Deep learning: a statistical viewpoint," *Acta numerica*, vol. 30, pp. 87-201, 2021.
- [30] Wang, Pin and Fan, En and Wang, Peng, "Comparative analysis of image classification algorithms based on traditional machine learning and deep learning," *Pattern recognition letters*, vol. 141, pp. 61-67, 2021.
- [31] Schmidt, Robin M, "Recurrent neural networks (rnns): A gentle introduction and overview," *arXiv preprint arXiv:1912.05911*, 2019.
- [32] R. Pascanu, "On the difficulty of training recurrent neural networks," *arXiv preprint arXiv:1211.5063*, 2013.
- [33] S. Zargar, "Introduction to sequence learning models: RNN, LSTM, GRU," *Department of Mechanical and Aerospace Engineering, North Carolina State University*, 2021.

- [34] Schuster, Mike and Paliwal, Kuldip K, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, pp. 2673-2681, 1997.
- [35] DiPietro, Robert and Hager, Gregory D, "Deep learning: RNNs and LSTM," pp. 503-519, 2020.
- [36] Zhao, Jingyu and Huang, Feiqing and Lv, Jia and Duan, Yanjie and Qin, Zhen and Li, Guodong and Tian, Guangjian, "Do RNN and LSTM have long memory?," pp. 11365-11375, 2020.
- [37] S. Hochreiter, "Long Short-term Memory," *Neural Computation MIT-Press*, 1997.
- [38] Masri, F and Saepudin, D and Adytia, D, "Forecasting of Sea Level Time Series using Deep Learning RNN, LSTM, and BiLSTM, Case Study in Jakarta Bay, Indonesia," *e-Proceeding of Engineering*, vol. 7, pp. 8544-8551, 2020.
- [39] Staudemeyer, Ralf C and Morris, Eric Rothstein, "Understanding LSTM--a tutorial into long short-term memory recurrent neural networks," *arXiv preprint arXiv:1909.09586*, 2019.
- [40] Yu, Yong and Si, Xiaosheng and Hu, Changhua and Zhang, Jianxun, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, pp. 1235-1270, 2019.

- [41] Vennerod, Christian Bakke and Kjaerran, Adrian and Bugge, Erling Stray, "Long short-term memory RNN," *arXiv preprint arXiv:2105.06756*, 2021.
- [42] Janiesch, Christian and Zschech, Patrick and Heinrich, Kai, "Machine learning and deep learning," *Electronic Markets*, vol. 31, pp. 685-695, 2021.
- [43] V. Jitani, "Recurrent Neural Network: Maths in 5 Minutes," Medium, 21 September 2020. [Online]. Available:
<https://vidishajitani.medium.com/recurrent-neural-network-maths-69214e4d69e1>. [Accessed 31 August 2024].
- [44] Handayani, Felisia and Mustikasari, Metty, "Sentiment Analysis Of Electric Cars Using Recurrent Neural Network Method In Indonesian Tweets," *Jurnal Ilmiah KURSOR*, vol. 10, 2020.
- [45] S. Saxena, "What is LSTM? Introduction to Long Short-Term Memory," Analytics Vidhya, 23 August 2024. [Online]. Available:
<https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>. [Accessed 31 August 2024].
- [46] C. Olah, "Understanding LSTM Networks," 27 August 2015. [Online]. Available:
<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>. [Accessed 31 August 2024].
- [47] Yalman, Yunus and Uyanik, "Prediction of voltage sag relative location with

data-driven algorithms in distribution grid," *Energies*, vol. 15, p. 6641, 2022.

- [48] Alinda Hardiantoro and Sari Hardiyanto, "Ramai soal Penipuan Perubahan

Biaya Administrasi ATM, Ini Kata BRI," Kompas, 22 June 2022. [Online].

Available:

<https://www.kompas.com/tren/read/2022/06/22/160400965/ramai-soal-penipuan-perubahan-biaya-administrasi-atm-ini-kata-bri>. [Accessed 31 August 2024].

- [49] "Worldwide 2023 Email Phishing Statistics and Examples," trendmicro, 20

June 2024. [Online]. Available: <https://www.trendmicro.com/>. [Accessed 31

August 2024].

- [50] Ashfaq, Shaikh and Chandre, "Defending Against Vishing Attacks: A

Comprehensive Review for Prevention and Mitigation Techniques," pp. 411-

422, 2023.

- [51] G. Zeng, "On the confusion matrix in credit scoring and its analytical

properties," *Communications in Statistics-Theory and Methods*, vol. 49, pp.

2080-2093, 2020.

- [52] Gopali, Saroj and Namin, Akbar S and Abri, Faranak and Jones, Keith S, "The

Performance of Sequential Deep Learning Models in Detecting Phishing

Websites Using Contextual Features of URLs," pp. 1064--1066, 2024.

- [53] Nagy, Naya and Aljabri, Malak and Shaahid, Afrah and Ahmed, "Phishing URLs Detection Using Sequential and Parallel ML Techniques: Comparative Analysis," *Sensors*, vol. 23, p. 3467, 2023.
- [55] Sahingoz, Ozgur Koray and Buber, Ebubekir and Demir, Onder and Diri, Banu, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345--357, 2019.
- [56] J. Scarpati, "What is a URL (Uniform Resource Locator)?," TechTarget, [Online]. Available:
<https://www.techtarget.com/searchnetworking/definition/URL>. [Accessed 30 August 2024].
- [57] Patil, Dharmaraj R and Patil, Jayantro B, "Malicious URLs detection using decision tree classifiers and majority voting technique," *Cybernetics and Information Technologies*, vol. 18, pp. 11--29, 2018.
- [58] Hassan, Esraa and Shams, Mahmoud Y and Hikal, Noha A and Elmougy, Samir, "The effect of choosing optimizer algorithms to improve computer vision tasks: a comparative study," *Multimedia Tools and Applications*, vol. 82, pp. 16591--16633, 2023.
- [59] Sharma, Sagar and Sharma, Simone and Athaiya, Anidhya, "Activation functions in neural networks," *Towards Data Sci*, vol. 6, pp. 310--316, 2017.

- [60] Chai, Christine P, "Comparison of text preprocessing methods," *Natural Language Engineering*, vol. 29, pp. 509--553, 2023.
- [61] Borisov, Vadim and Leemann, Tobias and Se{\ss}ler, Kathrin and Haug, Johannes and Pawelczyk, Martin and Kasneci, Gjergji, "Deep neural networks and tabular data: A survey," *IEEE transactions on neural networks and learning systems*, 2022.

LAMPIRAN 1

Contoh Perhitungan LSTM

Contoh perhitungan ini akan menggunakan URL acak sebagai input. URL ini akan melalui tahap preprocessing dimulai dengan normalisasi

URL input : <https://contoh.com>

Hasil normalisasi : contoh.com

Tahap berikutnya adalah pemisahan karakter untuk tokenisasi setiap karakter unik.

['c', 'o', 'n', 't', 'o', 'h', '.', 'c', 'o', 'm']

Karakter unik : ['c', 'o', 'n', 't', 'h', '.']

Mapping karakter ke integer :

Tabel L1.1. Contoh perhitungan tokenisasi

Karakter	Integer
c	1
o	2
n	3
t	4
h	5
.	6
m	7

Hasil tokenisasi ['c', 'o', 'n', 't', 'o', 'h', '.', 'c', 'o', 'm'] :

[1, 2, 3, 4, 2, 5, 6, 1, 2, 7]

Misalnya digunakan vector 2 dimensi untuk setiap karakter :

Karakter	Embedding Vector
c	[0.0, 0.2]
o	[0.2, 0.1]
n	[0.3, 0.2]
t	[0.4, 0.4]
h	[0.5, 0.3]
.	[0.0, 0.0]
m	[0.2, 0.5]

Hasil dari preprocessing ini akan dijadikan sebagai input kedalam perhitungan LSTM

State awal

$$h_{t-1} = [0, 0], C_{t-1} = [0, 0]$$

Bobot W_f dan bias b_f Forget Gate

$$W_f = [[0.5, -0.1, 0.3, 0.2], [0.0, 0.4, -0.2, 0.1]]$$

$$b_f = [0.1, -0.1]$$

Bobot W_i dan bias b_i Input Gate

$$W_i = [[-0.3, 0.2, 0.1, 0.5], [0.6, -0.4, 0.0, -0.2]]$$

$$b_i = [0.05, 0.05]$$

Bobot W_C dan bias b_C Candidate Gate

$$W_C = [[0.1, 0.3, -0.4, 0.2], [-0.2, 0.0, 0.2, 0.1]]$$

$$b_C = [0.0, 0.0]$$

Bobot W_o dan bias b_o Output Gate

$$W_o = [[0.2, -0.1, 0.4, 0.3], [0.5, 0.2, -0.3, -0.1]]$$

$$b_o = [0.0, 0.1]$$

Persamaan yang digunakan dalam perhitungan ini sebagai berikut :

Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Candidate Gate

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Update Cell State

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t$$

Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Hidden State

$$h_t = o_t \cdot \tanh(C_t)$$

Berikut langkah-langkah perhitungan LSTM pada 1 timestep:

x_t = embedding dari karakter ‘c’ : [0.0, 0.2]

Kalkulasi Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

$$[h_{t-1}, x_t] = [0.0, 0.0, 0.0, 0.2]$$

Baris pertama

$$f_t = \sigma([0.5, -0.1, 0.3, 0.2] \cdot [0.0, 0.0, 0.0, 0.2] + 0.1)$$

$$f_t = \sigma(0.5 * 0.0 + (-0.1) * 0.0 + 0.3 * 0.0 + 0.2 * 0.2 + 0.1)$$

$$f_t = \sigma(0 + 0 + 0 + 0.04 + 0.1)$$

$$f_t = \sigma(0.14)$$

$$f_t = 0.535$$

Baris kedua

$$f_t = \sigma([0.0, 0.4, -0.2, 0.1] \cdot [0.0, 0.0, 0.0, 0.2] - 0.1)$$

$$f_t = \sigma(0.0 * 0.0 + 0.4 * 0.0 + (-0.2) * 0.0 + 0.1 * 0.2 - 0.1)$$

$$f_t = \sigma(0 + 0 + 0 + 0.02 - 0.1)$$

$$f_t = \sigma(-0.08)$$

$$f_t = 0.480$$

Hasil forget gate : $f_t = [0.535, 0.480]$

Kalkulasi Input Gate

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

Baris pertama

$$i_t = \sigma([-0.3, 0.2, 0.1, 0.5] \cdot [0.0, 0.0, 0.0, 0.2] + 0.05)$$

$$i_t = \sigma((-0.3) * 0.0 + 0.2 * 0.0 + 0.1 * 0.0 + 0.5 * 0.2 + 0.05)$$

$$i_t = \sigma(0 + 0 + 0 + 0.1 + 0.05)$$

$$i_t = \sigma(0.15)$$

$$i_t = 0.537$$

Baris kedua

$$i_t = \sigma([0.6, -0.4, 0.0, -0.2] \cdot [0.0, 0.0, 0.0, 0.2] + 0.05)$$

$$i_t = \sigma(0.6 * 0.0 + (-0.4) * 0.0 + 0.0 * 0.0 + (-0.2) * 0.2 + 0.05)$$

$$i_t = \sigma(0 + 0 + 0 - 0.04 + 0.05)$$

$$i_t = \sigma(0.01)$$

$$i_t = 0.5025$$

Hasil input gate : $i_t = [0.537, 0.5025]$

Kalkulasi Candidate Gate

$$C'_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Baris pertama

$$C'_t = \tanh([0.1, 0.3, -0.4, 0.2] \cdot [0.0, 0.0, 0.0, 0.2] + 0.0)$$

$$C'_t = \tanh(0.1 * 0.0 + 0.3 * 0.0 + (-0.4) * 0.0 + 0.2 * 0.2 + 0.0)$$

$$C'_t = \tanh(0 + 0 + 0 + 0.04 + 0.0)$$

$$C'_t = \tanh(0.04)$$

$$C'_t = 0.04$$

Baris kedua

$$C'_t = \tanh([-0.2, 0.0, 0.2, 0.1] \cdot [0.0, 0.0, 0.0, 0.2] + 0.0)$$

$$C'_t = \tanh((-0.2) * 0.0 + 0.0 * 0.0 + 0.2 * 0.0 + 0.1 * 0.2 + 0.0)$$

$$C'_t = \tanh(0 + 0 + 0 + 0.02 + 0.0)$$

$$C'_t = \tanh(0.02)$$

$$C'_t = 0.02$$

Hasil candidate gate : $C'_t = [0.04, 0.02]$

Update Cell State

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C'_t$$

Baris pertama

$$C_t = 0.537 * 0.04$$

$$C_t = 0.0215$$

Baris kedua

$$C_t = 0.5025 * 0.02$$

$$C_t = 0.01005$$

Hasil update cell state : $C_t = [0.0215, 0.01005]$

Kalkulasi Output Gate

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

Baris pertama

$$o_t = \sigma([0.2, -0.1, 0.4, 0.3] \cdot [0.0, 0.0, 0.0, 0.2] + 0.0)$$

$$o_t = \sigma(0.2 * 0.0 + (-0.1) * 0.0 + 0.4 * 0.0 + 0.3 * 0.2 + 0.0)$$

$$o_t = \sigma(0 + 0 + 0 + 0.06 + 0.0)$$

$$o_t = \sigma(0.06)$$

$$o_t = 0.515$$

Baris kedua

$$o_t = \sigma([0.5, 0.2, -0.3, -0.1] \cdot [0.0, 0.0, 0.0, 0.2] + 0.1)$$

$$o_t = \sigma(0.5 * 0.0 + 0.2 * 0.0 + (-0.3) * 0.0 + (-0.1) * 0.2 + 0.1)$$

$$o_t = \sigma(0 + 0 + 0 - 0.02 + 0.1)$$

$$o_t = \sigma(0.08)$$

$$o_t = 0.520$$

Hasil output gate : $o_t = [0.515, 0.520]$

Hidden State

$$h_t = o_t \cdot \tanh(C_t)$$

Baris pertama

$$h_t = 0.515 \cdot \tanh(0.0215)$$

$$h_t = 0.515 \cdot 0.0215$$

$$h_t = 0.0111$$

Baris kedua

$$h_t = 0.520 \cdot \tanh(0.01005)$$

$$h_t = 0.520 \cdot 0.0215$$

$$h_t = 0.0052$$

Hasil hidden state : $h_t = [0.0111, 0.0052]$

Setelah diperoleh hasil perhitungan LSTM, dilanjutkan dengan perhitungan menggunakan dense layer dengan menggunakan beberapa fitur dari URL sebagai input. Akan digunakan angka acak yang mewakili nilai dari masing-masing fitur dalam URL.

Bobot dan bias Dense Layer :

$$W_{dense} = [[0.2, -0.3, 0.5], [0.1, 0.4, -0.2]]$$

$$b_{dense} = [0.0, 0.1]$$

Contoh fitur : [1, 3.5, 0]

Kalkulasi dense layer

$$y_{dense} = \text{ReLU}(W_{dense} \cdot x_{features} + b_{dense})$$

Baris pertama

$$y_{dense} = \text{ReLU}([0.2, -0.3, 0.5] \cdot [1, 3.5, 0] + 0.0)$$

$$y_{dense} = \text{ReLU}(0.2 \cdot 1 + (-0.3) \cdot 3.5 + 0.5 \cdot 0 + 0.0)$$

$$y_{dense} = \text{ReLU}(0.2 - 1.05 + 0 + 0.0)$$

$$y_{dense} = \text{ReLU}(-0.85)$$

$$y_{dense} = 0$$

Baris kedua

$$y_{dense} = \text{ReLU}([0.1, 0.4, -0.2] \cdot [1, 3.5, 0] + 0.1)$$

$$y_{dense} = \text{ReLU}(0.1 \cdot 1 + 0.4 \cdot 3.5 + (-0.2) \cdot 0 + 0.1)$$

$$y_{dense} = \text{ReLU}(0.1 + 1.4 + 0 + 0.1)$$

$$y_{dense} = \text{ReLU}(1.6)$$

$$y_{dense} = 1.6$$

Hasil dense layer : $y_{dense} = [0, 1.6]$

Setelah diperoleh hasil perhitungan LSTM dan dense layer, kedua output akan digabungkan sebelum melalui perhitungan dense layer terakhir untuk dilakukan klasifikasi.

Bobot dan bias final dense layer :

$$W_{dense} = [0.3, -0.2, 0.5, 0.1]$$

$$b_{dense} = [-0.1]$$

$$h_{LSTM} = [0.0111, 0.0052]$$

$$y_{dense} = [0, 1.6]$$

Gabungan output : [0.0111, 0.0052, 0, 1.6]

Kalkulasi final dense layer

$$y_{dense} = \text{ReLU}([0.3, -0.2, 0.5, 0.1] \cdot [0.0111, 0.0052, 0, 1.6] - 0.1)$$

$$y_{final} = \text{ReLU}((0.3 \cdot 0.0111) + (-0.2 \cdot 0.0052) + (0.5 \cdot 0) + (0.1 \cdot 1.6) - 0.1)$$

$$y_{final} = \text{ReLU}(0.00333 - 0.00104 + 0 + 0.16 - 0.01)$$

$$y_{final} = \text{ReLU}(0.06229)$$

$$y_{final} = 0.06229$$

Dilanjutkan dengan fungsi Sigmoid untuk tahap klasifikasi

$$\text{Probabilitas} = \sigma(\text{Score}) = \frac{1}{1 + e^{-\text{Score}}}$$

$$\text{Probabilitas} = \frac{1}{1 + e^{-0.06229}}$$

$$\text{Probabilitas} = 0.51625$$

Dengan asumsi threshold yang digunakan 0.5, diperoleh 2 kemungkinan, yaitu :

1. Probabilitas > 0.5 , diklasifikasikan sebagai phishing
2. Probabilitas ≤ 0.5 , diklasifikasikan sebagai non-phishing.

Dalam contoh perhitungan ini, karena probabilitas yang didapatkan adalah 0.51625, maka diperoleh hasil prediksi bahwa situs yang digunakan sebagai input merupakan situs phishing.

LAMPIRAN 2

Contoh Data Kaggle

Tabel L2.1. Contoh data Kaggle fitur 1-2

url	url_length	starts_with_ip
apaceast.cloudguest.central.arubanetworks.com	45	FALSE
quintadonoval.com	17	FALSE
nomadfactory.com	16	FALSE
tvarenasport.com	16	FALSE
widget.cluster.groovehq.com	27	FALSE
haymarketbeer.com	17	FALSE
oslosportslager.no	18	FALSE
cloudshell.me-south-1.aws.amazon.com	36	FALSE
bunkrstatic.b-cdn.net	21	FALSE
yourdailydish.com	17	FALSE
trentemoller.com	16	FALSE
coveto.de	9	FALSE
offertracking.vuclip.com	24	FALSE
fllt.org	8	FALSE
revn.jp	7	FALSE
ldcws.pcpf.panasonic.com	24	FALSE
aarohamresorts.com	18	FALSE
gccbusinessnews.com	19	FALSE
accounting-simplified.com	25	FALSE
ege.sdamgia.ru	14	FALSE
stamp0-fd.prod.athena.msrareservices.com	40	FALSE
rebelmouse.net	14	FALSE
garagedreams.net	16	FALSE
rr5---sn-cu-aigss.googlevideo.com	33	FALSE
beian.gov.cn.iname.damddos.com	30	FALSE
castleintheclouds.org	21	FALSE
www.franchising.com	19	FALSE
portal.rules.comcast.com	24	FALSE
tealcreek.net	13	FALSE
ciec1.org	9	FALSE
tigermuaythai.com	17	FALSE

Tabel L2.2. Contoh data Kaggle fitur 3-4

url	url_entropy	has_punycode
apaceast.cloudguest.central.arubanetworks.com	3.9	FALSE
quintadonoval.com	3.5	FALSE
nomadfactory.com	3.3	FALSE
tvarenasport.com	3.5	FALSE
widget.cluster.groovehq.com	3.9	FALSE
haymarketbeer.com	3.4	FALSE
oslosportslager.no	3.2	FALSE
cloudshell.me-south-1.aws.amazon.com	3.9	FALSE
bunkrstatic.b-cdn.net	3.6	FALSE
yourdailydish.com	3.6	FALSE
trentemoller.com	3.0	FALSE
coveto.de	2.7	FALSE
offertracking.vuclip.com	3.9	FALSE
fllt.org	2.75	FALSE
revn.jp	2.8	FALSE
ldcws.pcpf.panasonic.com	3.5	FALSE
aarohamresorts.com	3.1	FALSE
gccbusinessnews.com	3.3	FALSE
accounting-simplified.com	3.9	FALSE
ege.sdamgia.ru	3.2	FALSE
stamp0-fd.prod.athena.msrareservices.com	3.9	FALSE
rebelmouse.net	3.2	FALSE
garagedreams.net	3.1	FALSE
rr5---sn-cu-aigss.googlevideo.com	3.8	FALSE
beian.gov.cn.iname.damddos.com	3.5	FALSE
castleintheclouds.org	3.8	FALSE
www.franchising.com	3.5	FALSE
portal.rules.comcast.com	3.5	FALSE
tealcreek.net	2.9	FALSE
ciec1.org	2.9	FALSE
tigermuaythai.com	3.6	FALSE

Tabel L2.3. Contoh data Kaggle fitur 5-8

url	digit_letter_ratio	dot_count	at_count
apaceast.cloudguest.central.arubanetworks.com	0.0	4	0
quintadonoval.com	0.0	1	0
nomadfactory.com	0.0	1	0
tvarenasport.com	0.0	1	0
widget.cluster.groovehq.com	0.0	3	0
haymarketbeer.com	0.0	1	0
oslosportslager.no	0.0	1	0
cloudshell.me-south-1.aws.amazon.com	0.03448275862068	4	0
bunkerstatic.b-cdn.net	0.0	2	0
yourdailydish.com	0.0	1	0
trentemoller.com	0.0	1	0
coveto.de	0.0	1	0
offertracking.vuclip.com	0.0	2	0
flit.org	0.0	1	0
revn.jp	0.0	1	0
ldcws.pcpf.panasonic.com	0.0	3	0
aarohamresorts.com	0.0	1	0
gccbusinessnews.com	0.0	1	0
accounting-simplified.com	0.0	1	0
ege.sdamgia.ru	0.0	2	0
stamp0-fd.prod.athena.msrareservices.com	0.02941176470588	4	0
rebelmouse.net	0.0	1	0
garagedreams.net	0.0	1	0
rr5---sn-cu-aigss.googlevideo.com	0.04	2	0
beian.gov.cn.iname.damddos.com	0.0	5	0
castleintheclouds.org	0.0	1	0
www.franchising.com	0.0	2	0
portal.rules.comcast.com	0.0	3	0
tealcreek.net	0.0	1	0
ciec1.org	0.14285714285714	1	0
tigermuaythai.com	0.0	1	0

Tabel L2.4. Contoh data Kaggle fitur 9-10

url	dash_count	tld_count
apaceast.cloudguest.central.arubanetworks.com	0	0
quintadonoval.com	0	0
nomadfactory.com	0	0
tvarenasport.com	0	0
widget.cluster.groovehq.com	0	0
haymarketbeer.com	0	0
oslosportslager.no	0	0
cloudshell.me-south-1.aws.amazon.com	2	0
bunkrstatic.b-cdn.net	1	0
yourdailydish.com	0	0
trentemoller.com	0	0
coveto.de	0	0
offertracking.vuclip.com	0	0
flit.org	0	0
revn.jp	0	0
ldcws.ppcf.panasonic.com	0	0
aarohamresorts.com	0	0
gccbusinessnews.com	0	0
accounting-simplified.com	1	0
ege.sdamgia.ru	0	0
stamp0-fd.prod.athena.msrareservices.com	1	0
rebelmouse.net	0	0
garagedreams.net	0	0
rr5---sn-cu-aigss.googlevideo.com	5	0
beian.gov.cn.iname.damddos.com	0	0
castleintheclouds.org	0	0
www.franchising.com	0	0
portal.rules.comcast.com	0	0
tealcreek.net	0	0
ciec1.org	0	0
tigermuaythai.com	0	0

Tabel L2.5. Contoh data Kaggle fitur 11-12

url	domain_has_digits	subdomain_c
apaceast.cloudguest.central.arubanetworks.com	FALSE	3
quintadonoval.com	FALSE	0
nomadfactory.com	FALSE	0
tvarenasport.com	FALSE	0
widget.cluster.groovehq.com	FALSE	2
haymarketbeer.com	FALSE	0
oslosportslager.no	FALSE	0
cloudshell.me-south-1.aws.amazon.com	FALSE	3
bunkrstatic.b-cdn.net	FALSE	1
yourdailydish.com	FALSE	0
trentemoller.com	FALSE	0
coveto.de	FALSE	0
offertracking.vuclip.com	FALSE	1
flit.org	FALSE	0
revn.jp	FALSE	0
ldcws.pcpf.panasonic.com	FALSE	2
aarohamresorts.com	FALSE	0
gccbusinessnews.com	FALSE	0
accounting-simplified.com	FALSE	0
ege.sdamgia.ru	FALSE	1
stamp0-fd.prod.athena.msrareservices.com	FALSE	3
rebelmouse.net	FALSE	0
garagedreams.net	FALSE	0
rr5---sn-cu-aigss.googlevideo.com	FALSE	1
beian.gov.cn.iname.damddos.com	FALSE	4
castleintheclouds.org	FALSE	0
www.franchising.com	FALSE	1
portal.rules.comcast.com	FALSE	2
tealcreek.net	FALSE	0
ciec1.org	TRUE	0
tigermuaythai.com	FALSE	0

Tabel L2.6. Contoh data Kaggle fitur 13-14

url	nan_char_entropy	has_internal_links
apaceast.cloudguest.central.arubanetworks.com	0.310386941895971	FALSE
quintadonoval.com	0.240438990661784	FALSE
nomadfactory.com	0.25	FALSE
tvarenasport.com	0.25	FALSE
widget.cluster.groovehq.com	0.352213889049145	FALSE
haymarketbeer.com	0.240438990661784	FALSE
oslosportslager.no	0.231662500080128	FALSE
cloudshell.me-south-1.aws.amazon.com	0.352213889049145	FALSE
bunkrstatic.b-cdn.net	0.323077849788453	FALSE
yourdailydish.com	0.240438990661784	FALSE
trentemoller.com	0.25	FALSE
coveto.de	0.352213889049145	FALSE
offertracking.vuclip.com	0.298746875060096	FALSE
fllt.org	0.375	FALSE
revn.jp	0.401050703151086	FALSE
ldcws.pcpf.panasonic.com	0.375	FALSE
aarohamresorts.com	0.231662500080128	FALSE
gccbusinessnews.com	0.223575132286504	FALSE
accounting-simplified.com	0.185754247590989	FALSE
ege.sdamgia.ru	0.401050703151086	FALSE
stamp0-fd.prod.athena.msrareservices.com	0.332192809488736	FALSE
rebelmouse.net	0.271953923004114	FALSE
garagedreams.net	0.25	FALSE
rr5---sn-cu-aigss.googlevideo.com	0.245114795112633	FALSE
beian.gov.cn.iname.damddos.com	0.430827083453526	FALSE
castleintheclouds.org	0.209157972513274	FALSE
www.franchising.com	0.341887106678272	FALSE
portal.rules.comcast.com	0.375	FALSE
tealcreek.net	0.284649209087776	FALSE
ciec1.org	0.352213889049145	FALSE
tigermuaythai.com	0.240438990661784	FALSE

Tabel L2.7. Contoh data Kaggle fitur 15

url	domain_age_days
apaceast.cloudguest.central.arubanetworks.com	8250.0
quintadonoval.com	10106.0
nomadfactory.com	8111.0
tvarenasport.com	5542.0
widget.cluster.groovehq.com	5098.0
haymarketbeer.com	5455.0
oslosportslager.no	9064.0
cloudshell.me-south-1.aws.amazon.com	10904.0
bunkrstatic.b-cdn.net	3058.0
yourdailydish.com	3533.0
trentemoller.com	7450.0
coveto.de	9064.0
offertracking.vuclip.com	7389.0
fllt.org	8993.0
revn.jp	7415.0
ldcws.pcpf.panasonic.com	12528.0
aarohamresorts.com	2111.0
gccbusinessnews.com	3495.0
accounting-simplified.com	4801.0
ege.sdamgia.ru	7757.0
stamp0-fd.prod.athena.msrareservices.com	3987.0
rebelmouse.net	2089.0
garagedreams.net	2801.0
rr5---sn-cu-aigss.googlevideo.com	7777.0
beian.gov.cn.iname.damddos.com	3566.0
castleintheclouds.org	7290.0
www.franchising.com	10634.0
portal.rules.comcast.com	10603.0
tealcreek.net	8979.0
ciec1.org	8833.0
tigermuaythai.com	7131.0

LAMPIRAN 3

Hasil pengujian modul

The screenshot shows the homepage of the SafeClick service. At the top left is the logo 'SafeClick' with a shield icon. At the top right is a 'FAQ' button. Below the header is a main title 'Protect Yourself from Phishing Attacks' with a subtitle explaining the AI-powered real-time analysis. A large blue 'Start Scanning' button is centered. Below this are three cards detailing features: 'Real-time Detection' (ML model analyzes URLs), 'Deep URL Analysis' (examines URL structure, domain age, and suspicious patterns), and 'Comprehensive Protection' (detects various phishing techniques and malicious patterns). At the bottom is a 'How It Works' section with three numbered steps: 1. Enter URL (Paste any suspicious URL you want to analyze), 2. Deep Analysis (Our AI model examines multiple security features), and 3. Get Results (Receive instant feedback on potential threats).

Protect Yourself from Phishing Attacks

Our advanced AI-powered system analyzes URLs in real-time to detect potential phishing threats.

Start Scanning

Real-time Detection
Advanced ML model analyzes URLs instantly for phishing threats

Deep URL Analysis
Examines URL structure, domain age, and suspicious patterns

Comprehensive Protection
Detects various phishing techniques and malicious patterns

How It Works

- Enter URL**
Paste any suspicious URL you want to analyze
- Deep Analysis**
Our AI model examines multiple security features
- Get Results**
Receive instant feedback on potential threats

Gambar L3.1. Hasil pengujian modul utama



Gambar L3.2. Hasil pengujian modul input



Gambar L3.3. Hasil pengujian modul output (safe)

The screenshot shows the SafeClick interface. At the top left is the SafeClick logo. At the top right is a 'FAQ' button. The main title is 'Scan Results'. Below it, the URL 'http://www.grupoly.com/slap/GD/' is listed. A red warning message '⚠ Potential Phishing URL Detected' is displayed. At the bottom are two buttons: 'Scan Another URL.' and 'Back to Home.'

Gambar L3.4. Hasil pengujian modul output (phishing)

The screenshot shows the SafeClick FAQ page. At the top left is the SafeClick logo. At the top right is a 'Back' button. The title 'Frequently Asked Questions' is centered above a question mark icon. Below the title are three expandable sections: 'What is phishing?', 'How does the detection work?', and 'How accurate is the detection?'. Each section contains a brief description.

- What is phishing?**

Phishing is a cybercrime where attackers impersonate legitimate institutions to steal sensitive information. They often use deceptive URLs that look similar to genuine websites.
- How does the detection work?**

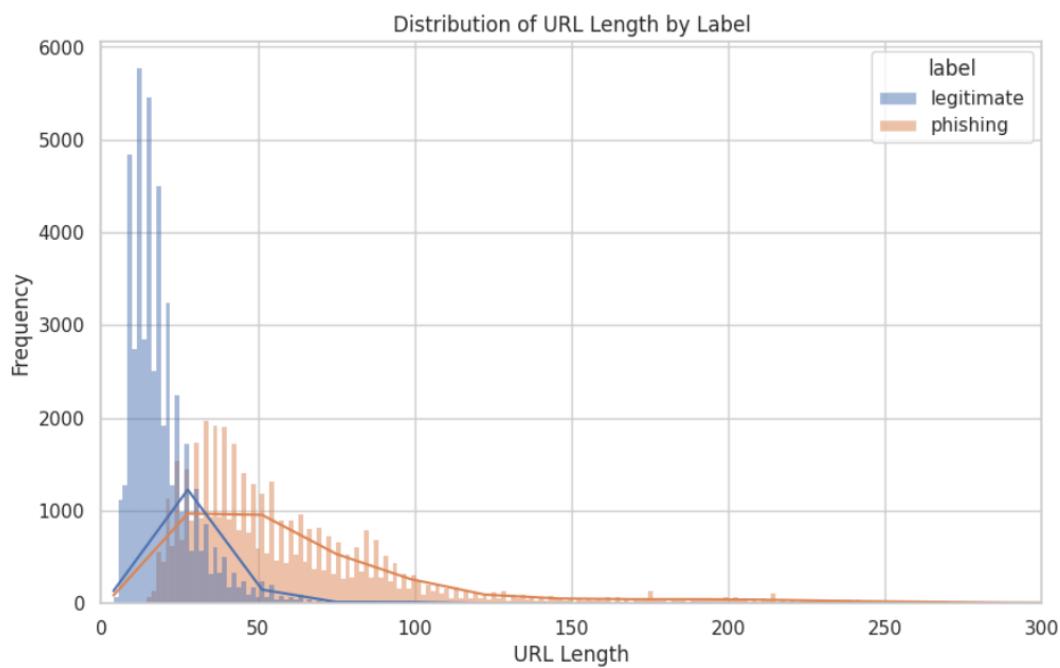
Our system uses machine learning to analyze various URL characteristics including length, entropy, special characters, domain age, and other patterns commonly associated with phishing attempts.
- How accurate is the detection?**

Our model provides a confidence score with each prediction. While highly accurate, we recommend using it as part of a broader security approach and always exercising caution with suspicious URLs.

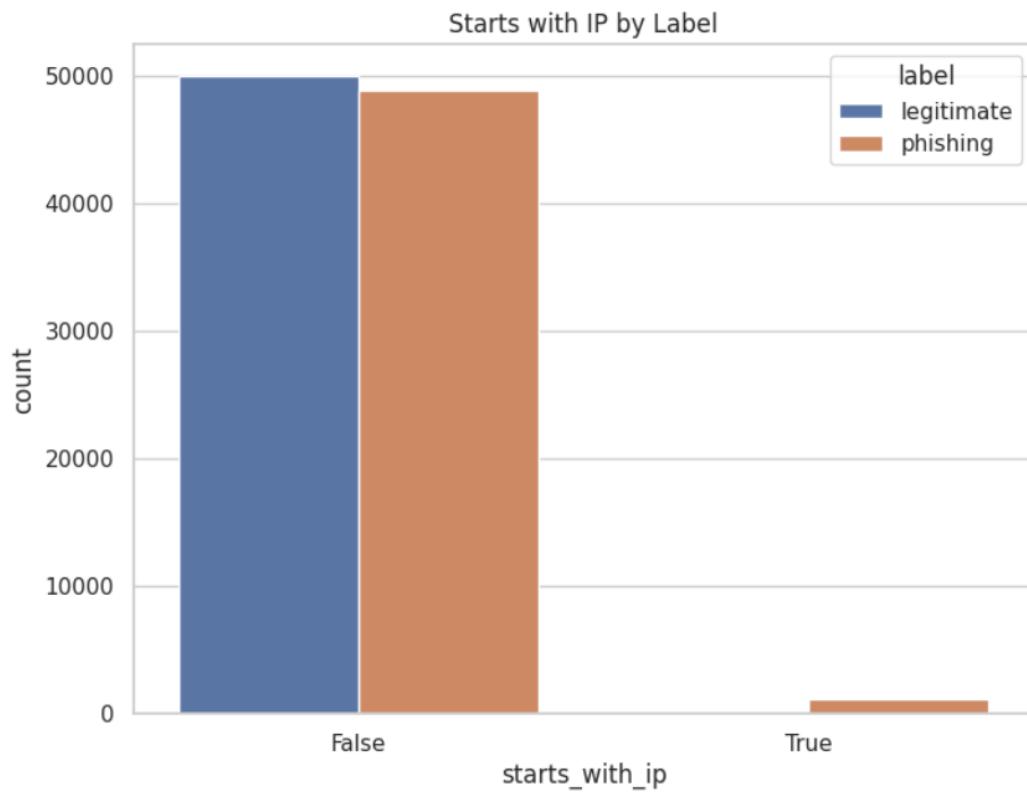
Gambar L3.5. Hasil pengujian modul bantuan

LAMPIRAN 4

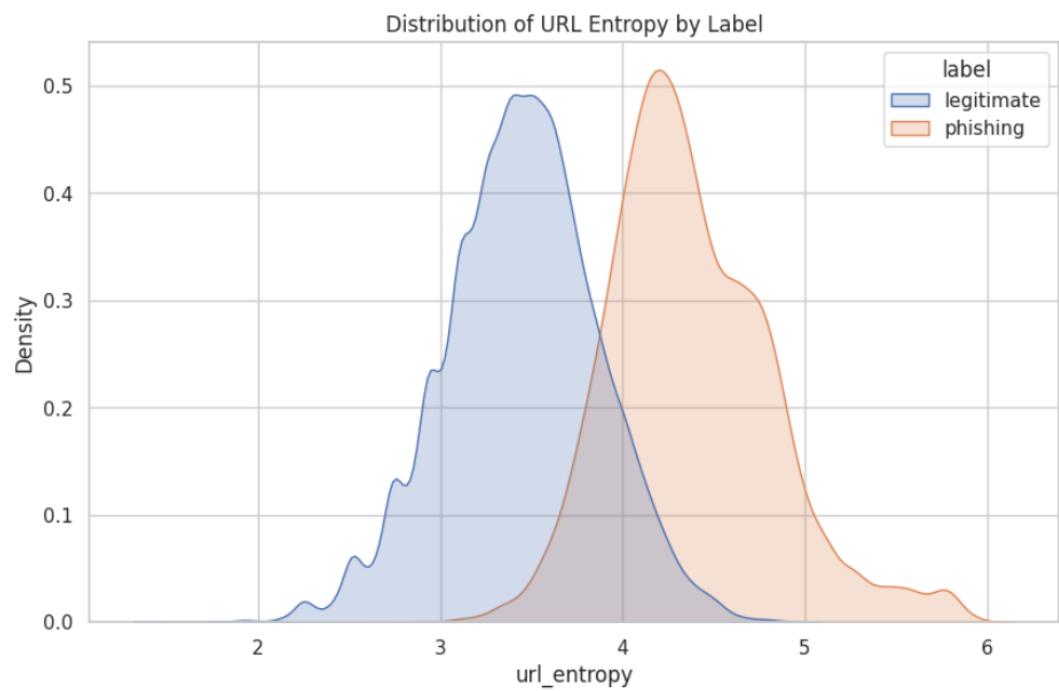
Distribusi fitur dataset



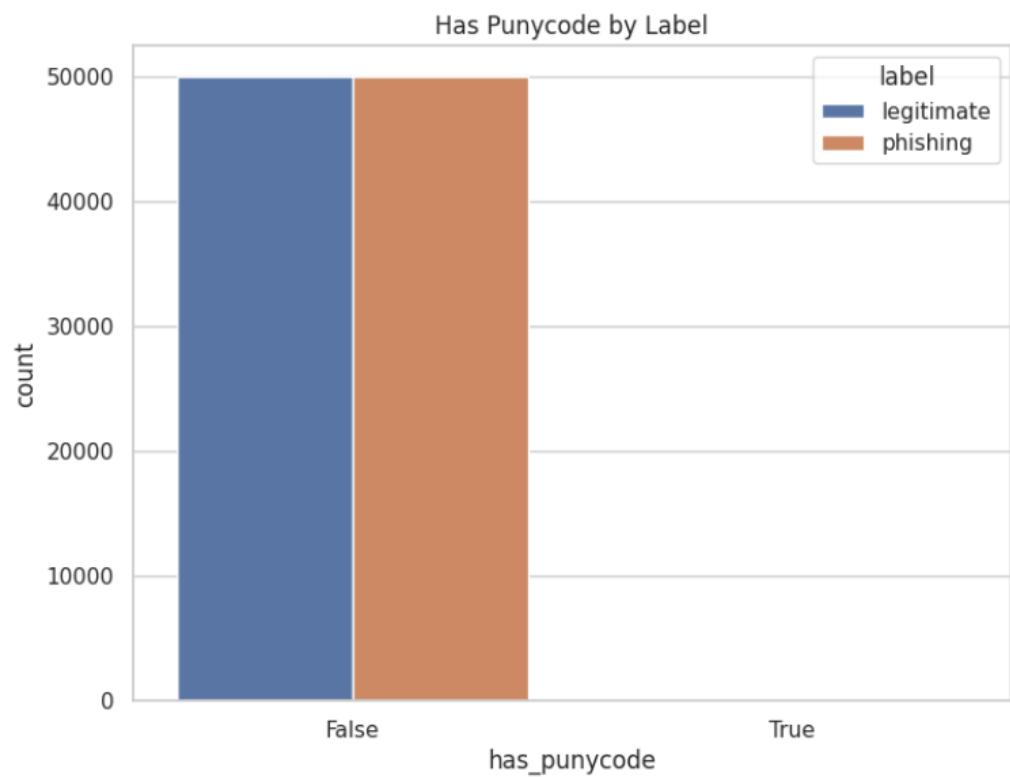
Gambar L4.1. Distribusi fitur panjang URL



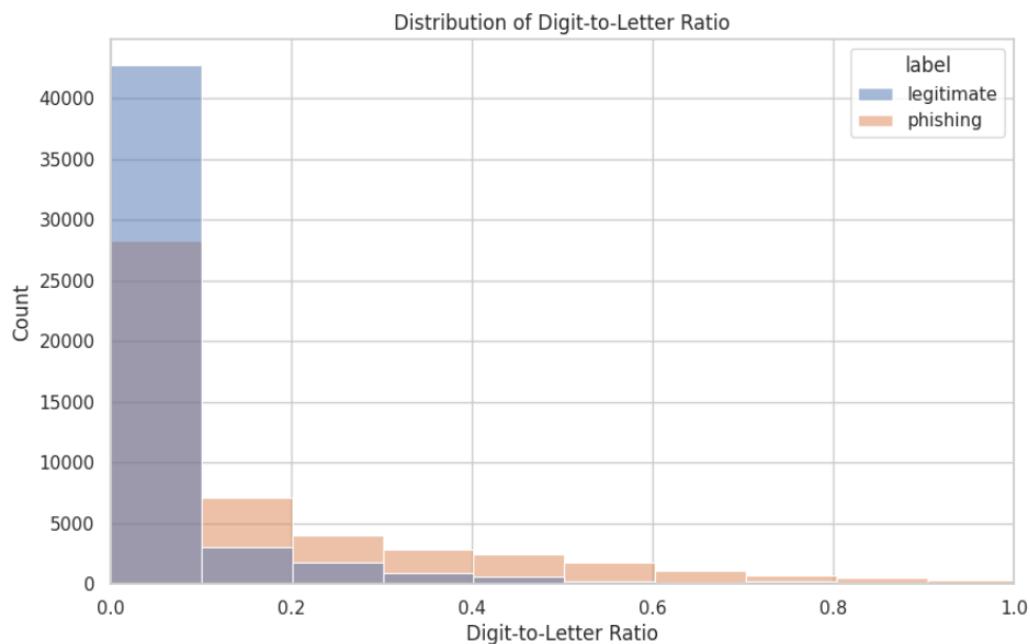
Gambar L4.2. Distribusi fitur ip address sebagai nama domain



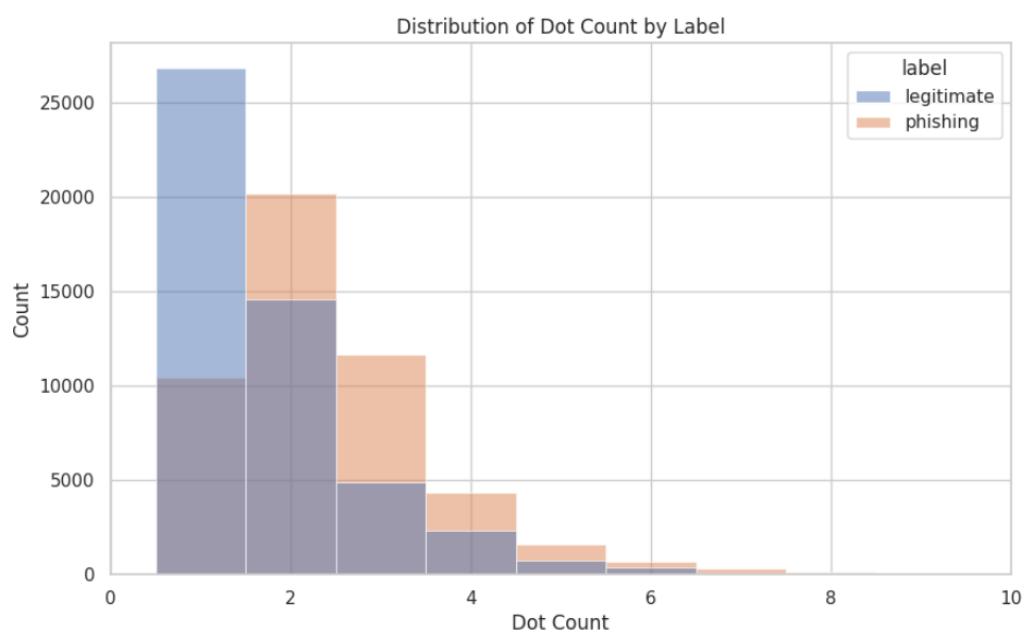
Gambar L4.3. Distribusi fitur tingkat entropi URL



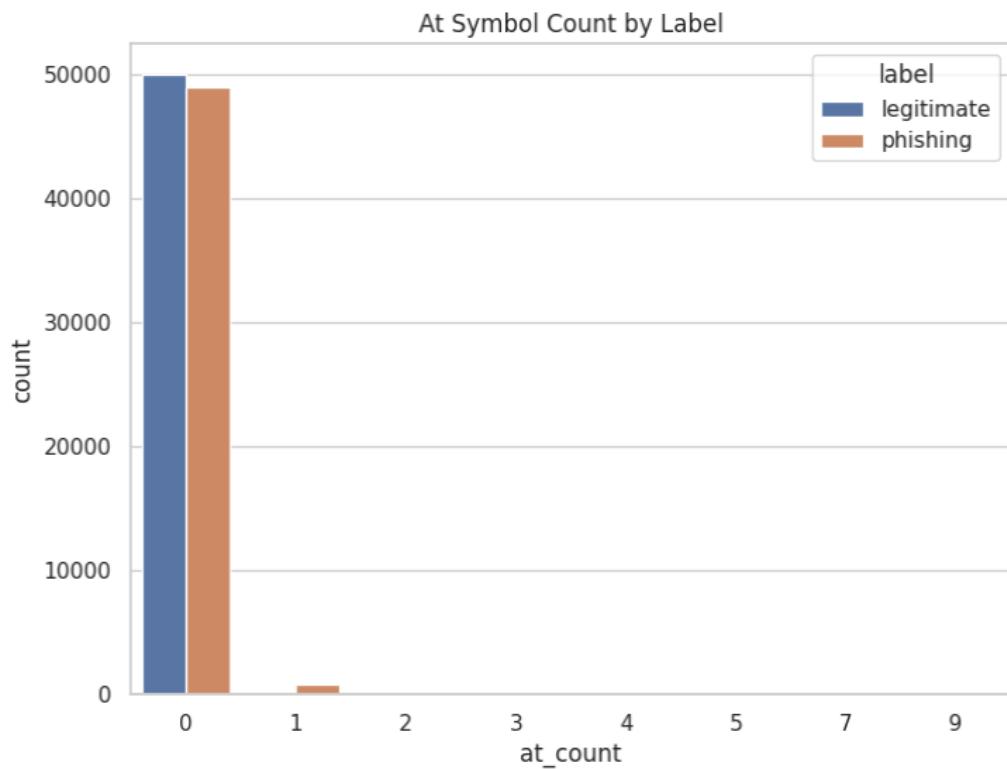
Gambar L4.4. Distribusi fitur karakter punycode



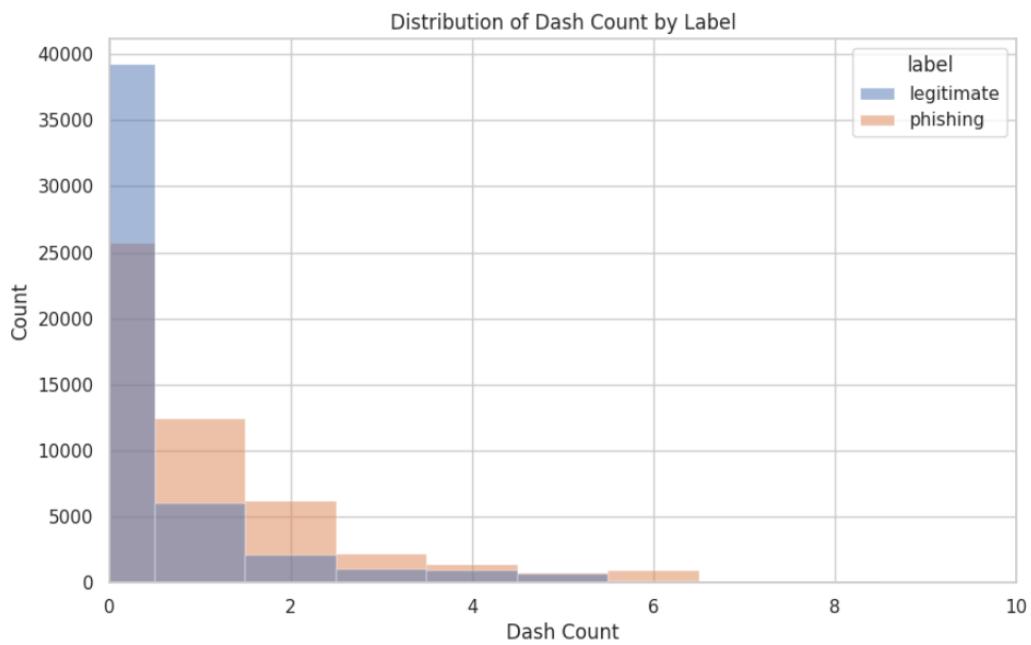
Gambar L4.5. Distribusi fitur rasio huruf dan angka



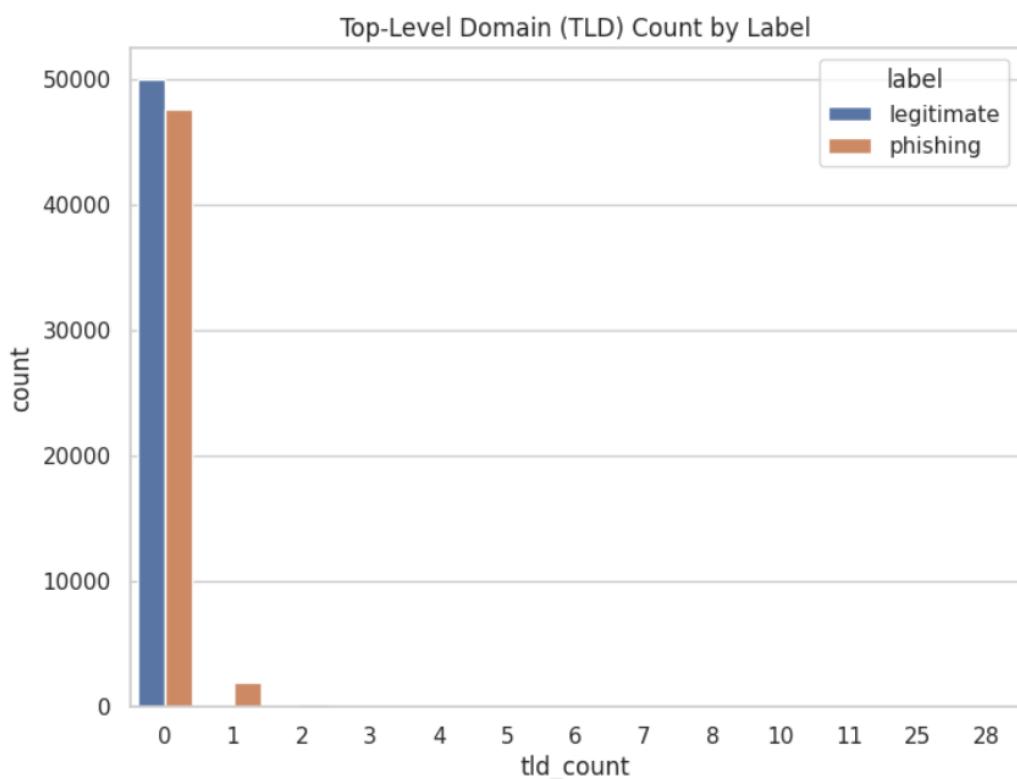
Gambar L4.6. Distribusi fitur banyak titik dalam URL



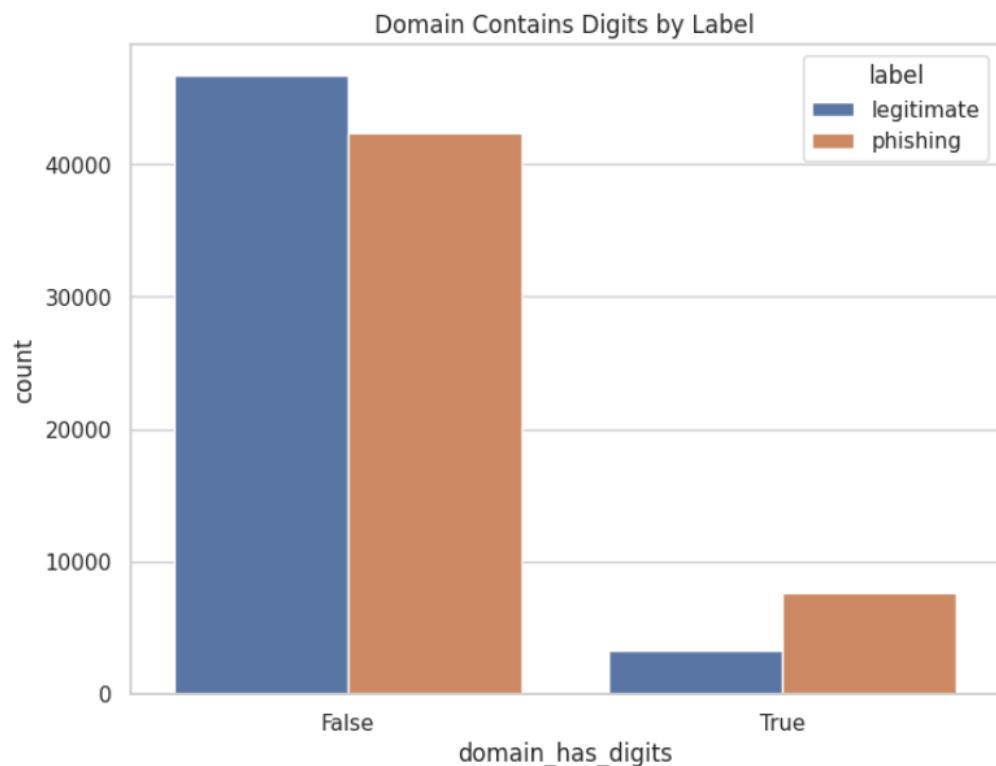
Gambar L4.7. Distribusi fitur banyak simbol at dalam URL



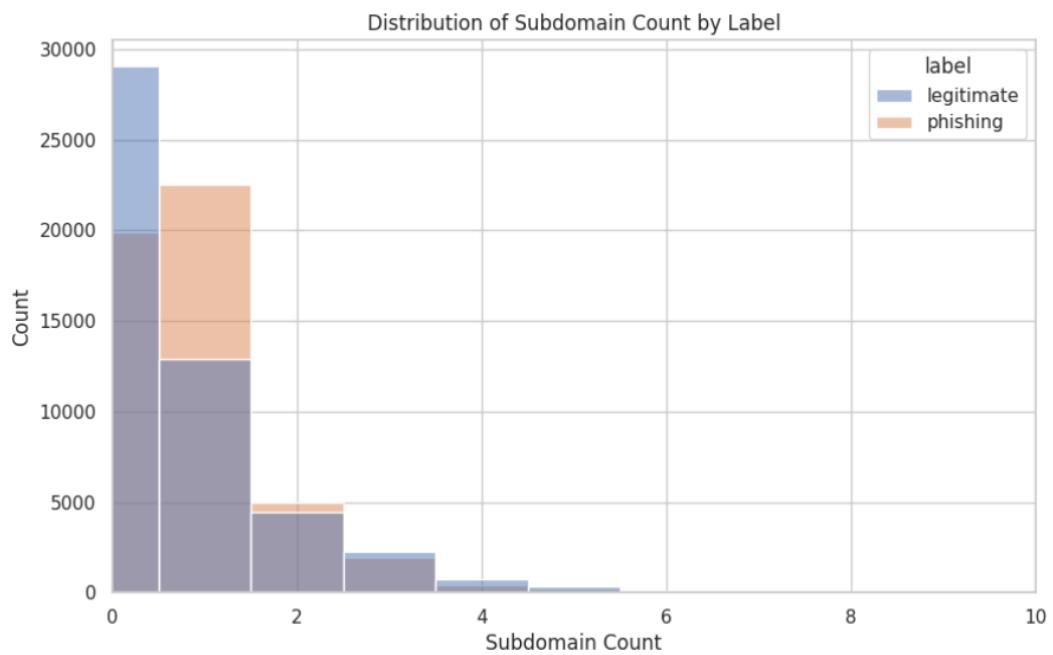
Gambar L4.8. Distribusi fitur banyak simbol dash dalam URL



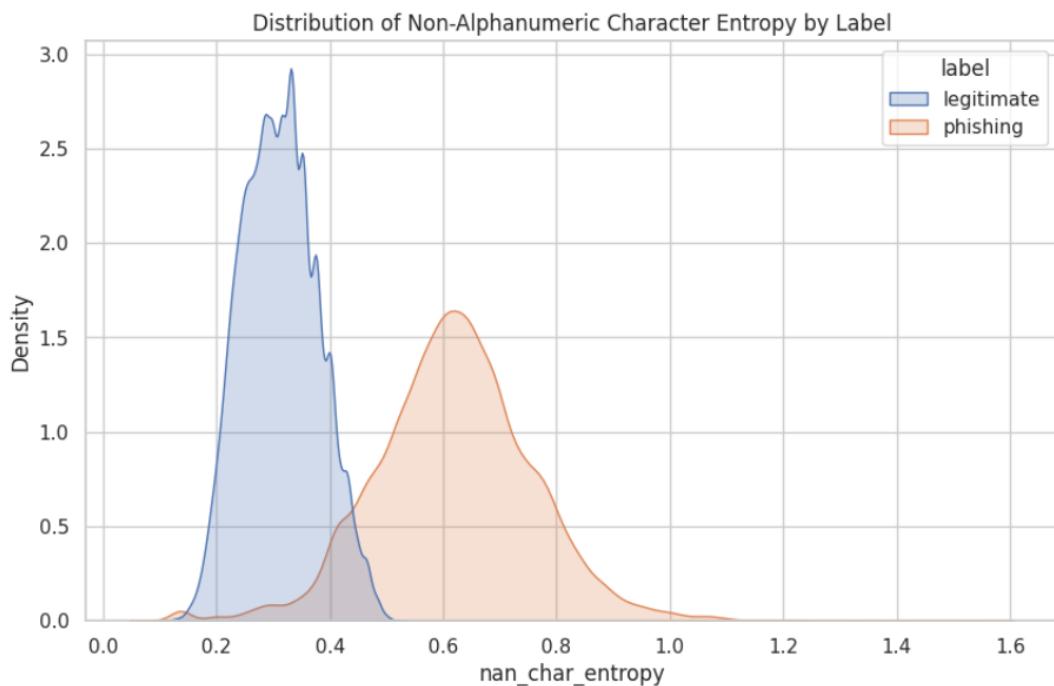
Gambar L4.9. Distribusi fitur banyak TLD dalam URL



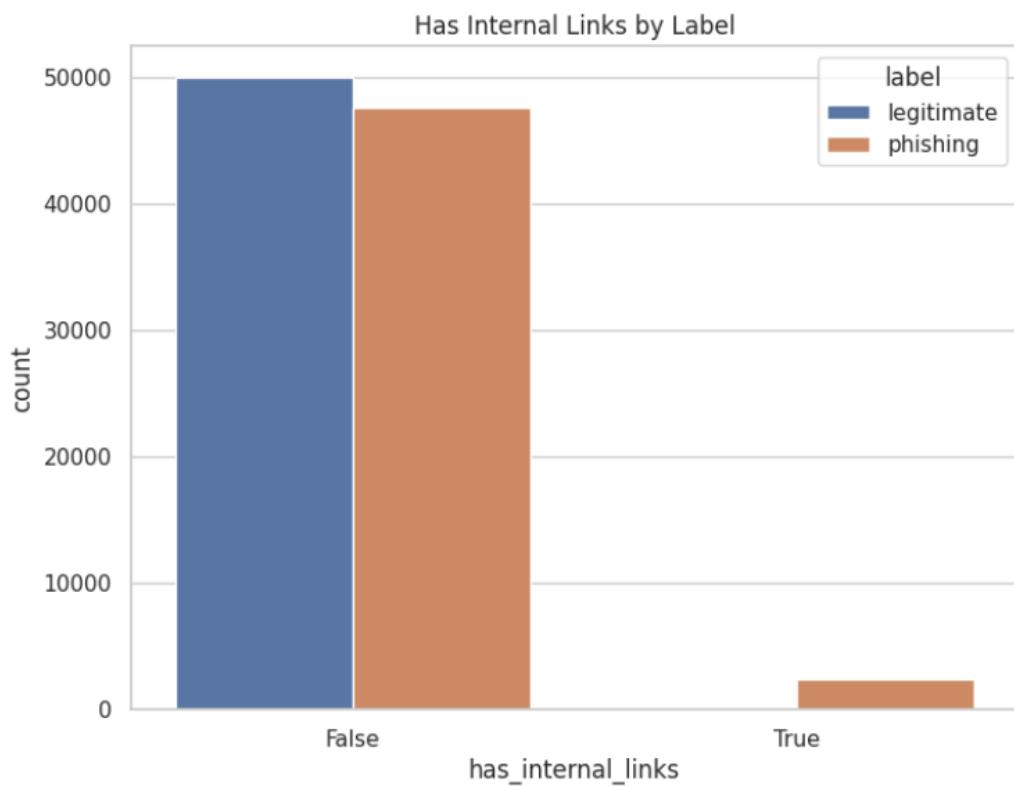
Gambar L4.10. Distribusi fitur domain mengandung angka



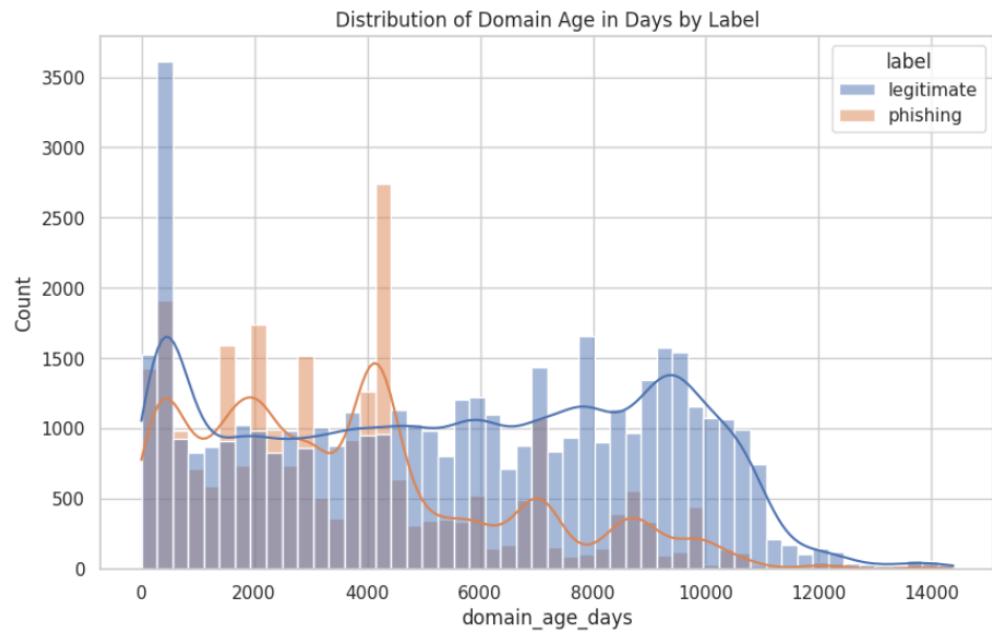
Gambar L4.11. Distribusi fitur banyak subdomain dalam URL



Gambar L4.12. Distribusi fitur tingkat entropi karakter non-alfanumerik



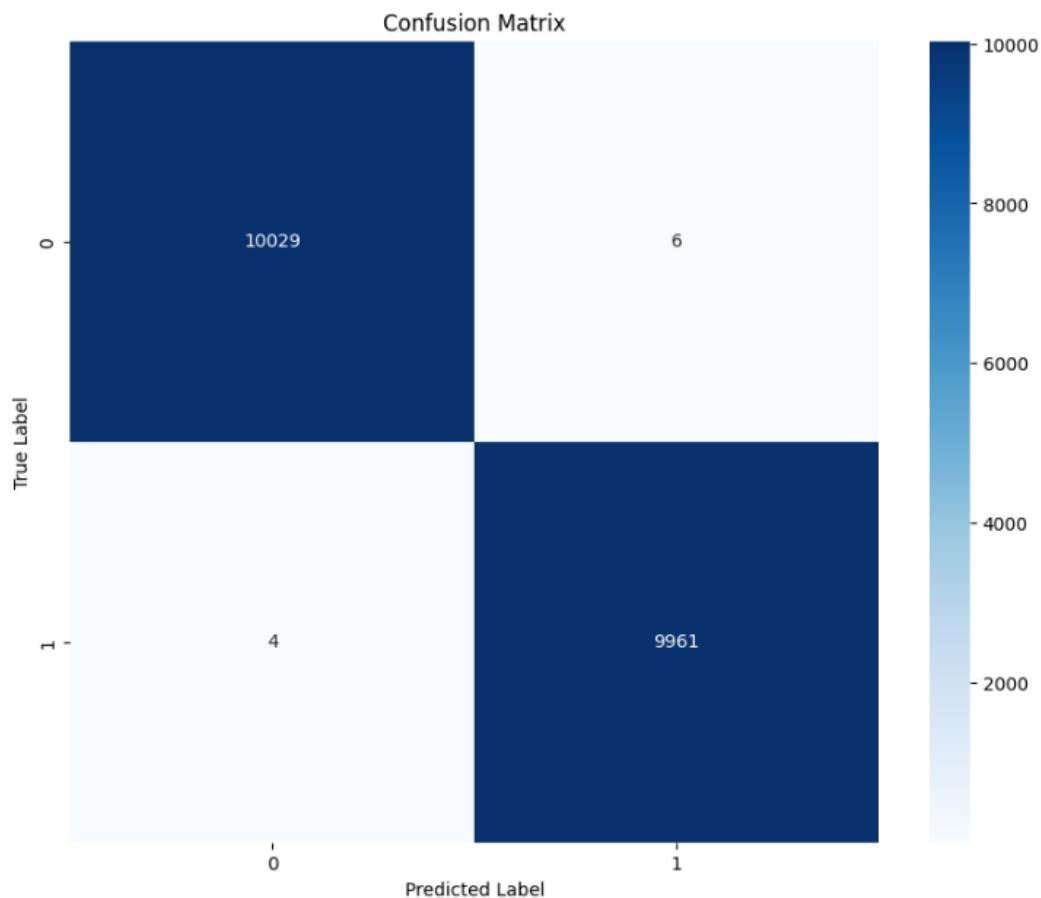
Gambar L4.13. Distribusi fitur URL mengandung tautan



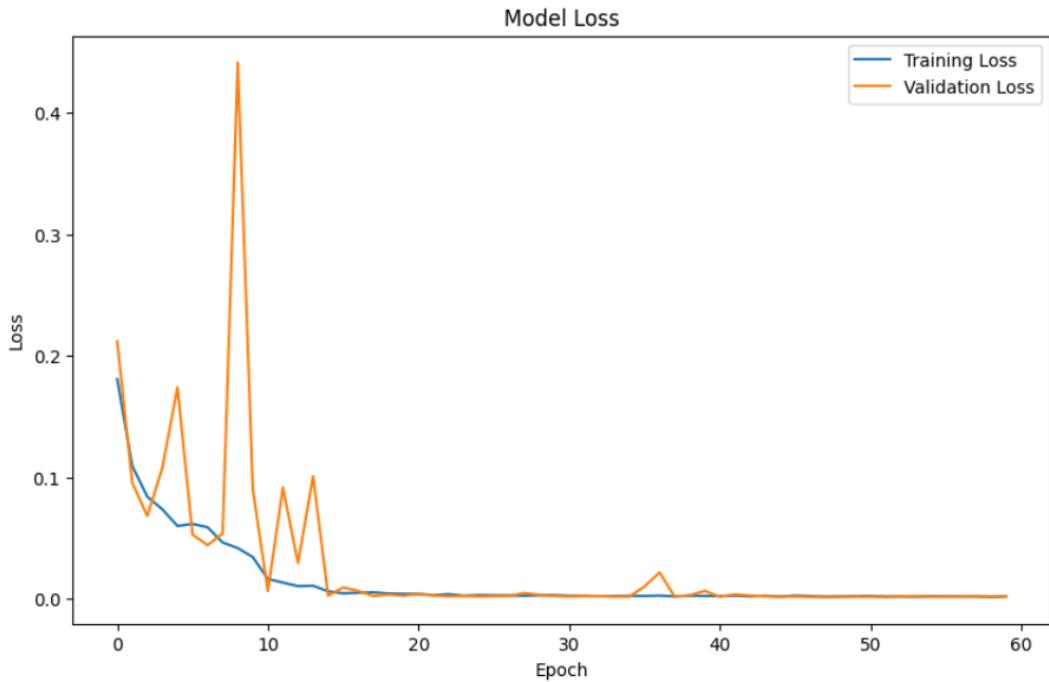
Gambar L4.14. Distribusi fitur umur domain

LAMPIRAN 5

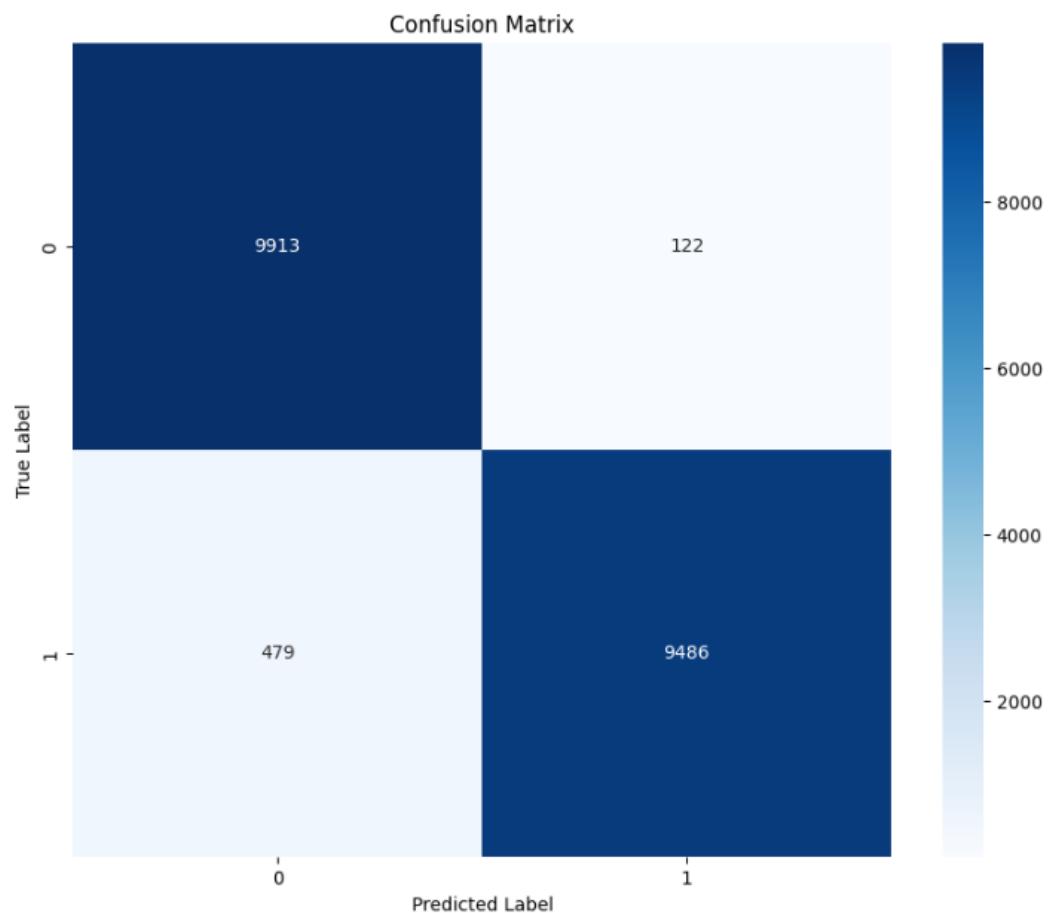
Hasil pengujian hyperparameter tuning kedua



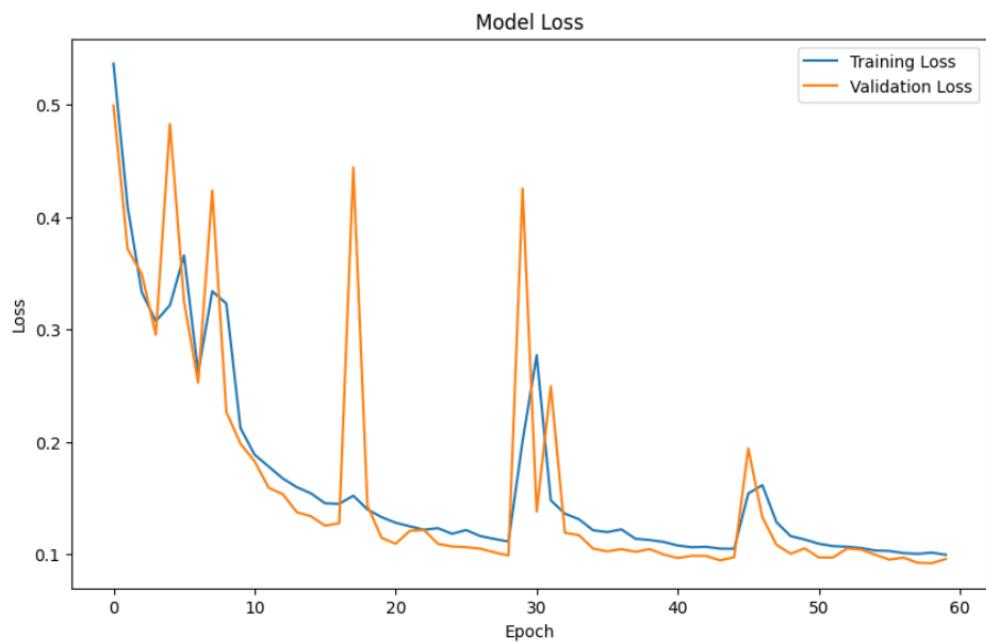
Gambar L5.1. Confusion matrix model 1 LSTM



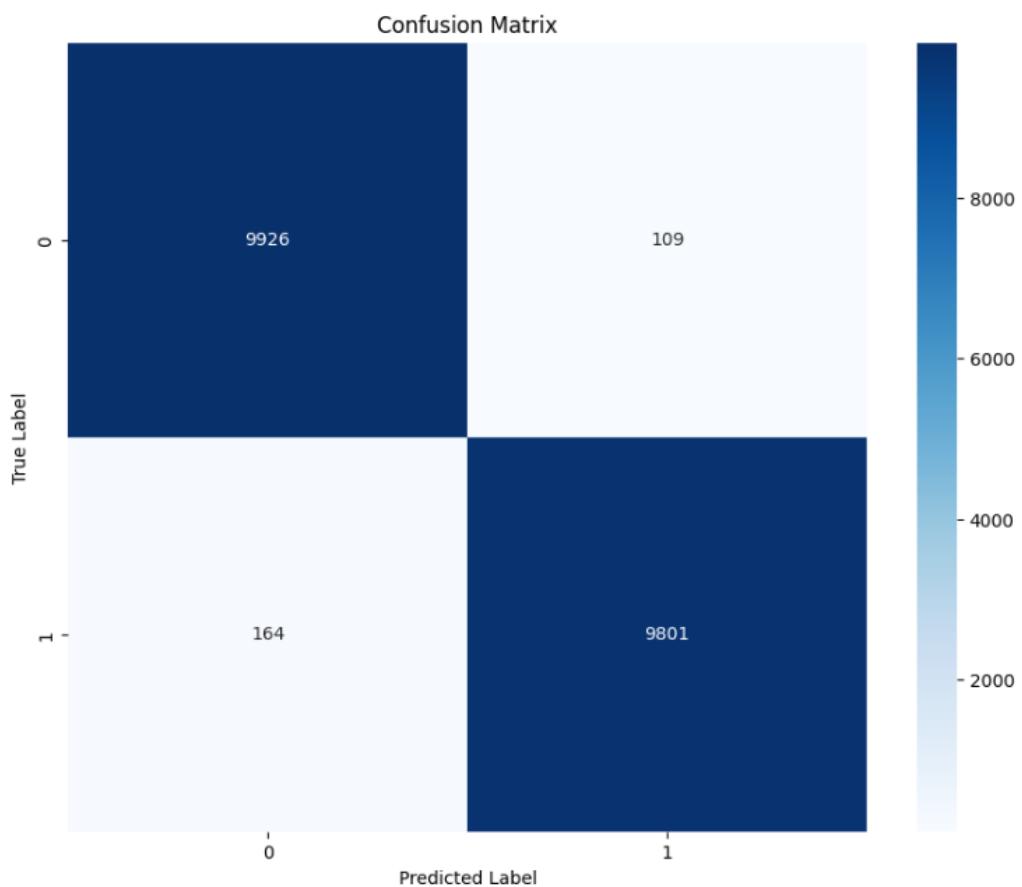
Gambar L5.2. Grafik loss model 1 LSTM



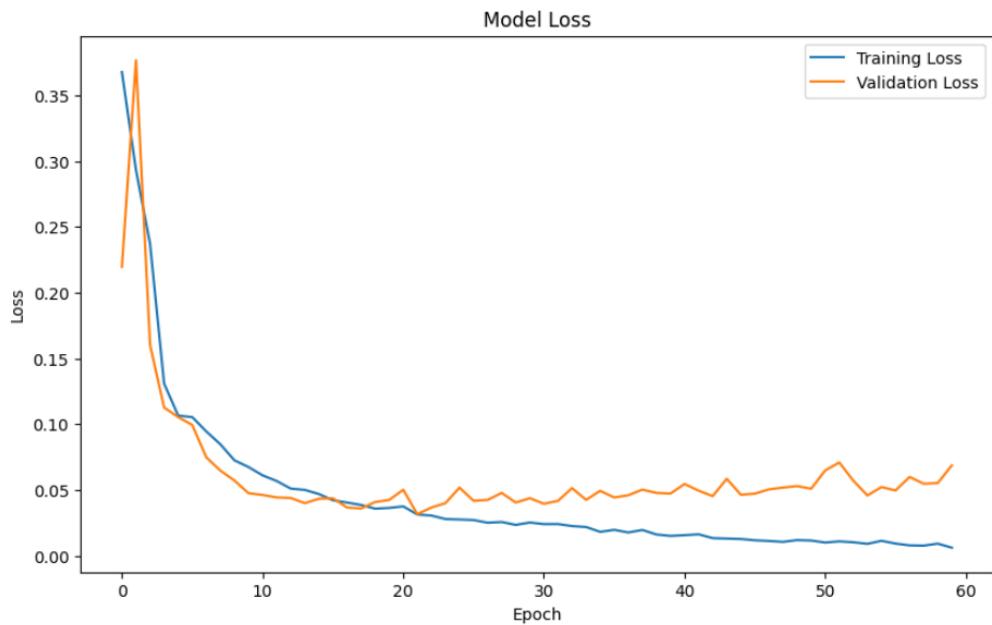
Gambar L5.3. Confusion matrix model 2 LSTM



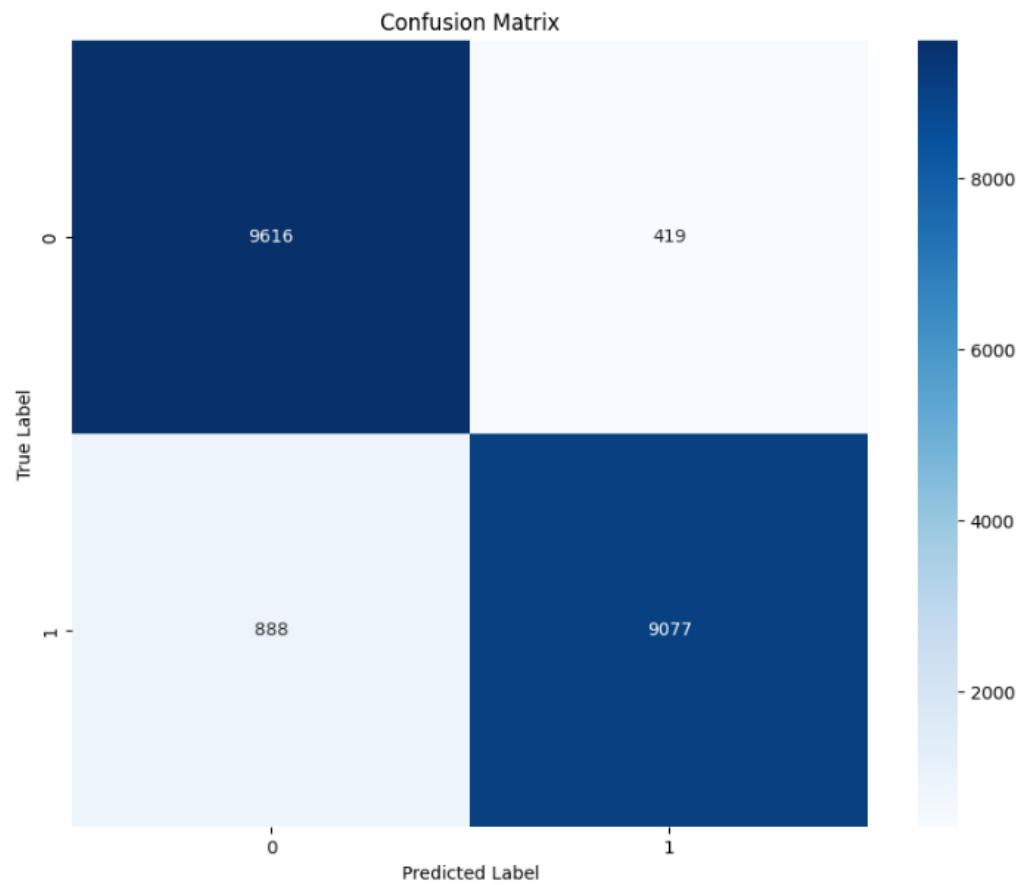
Gambar L5.4. Grafik loss model 2 LSTM



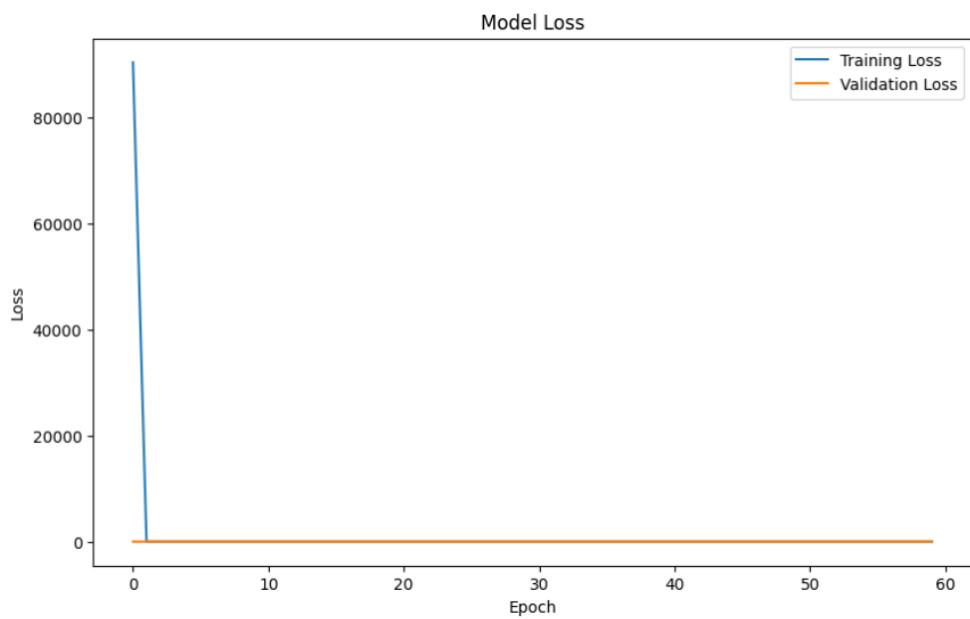
Gambar L5.5. Confusion matrix model 3 LSTM



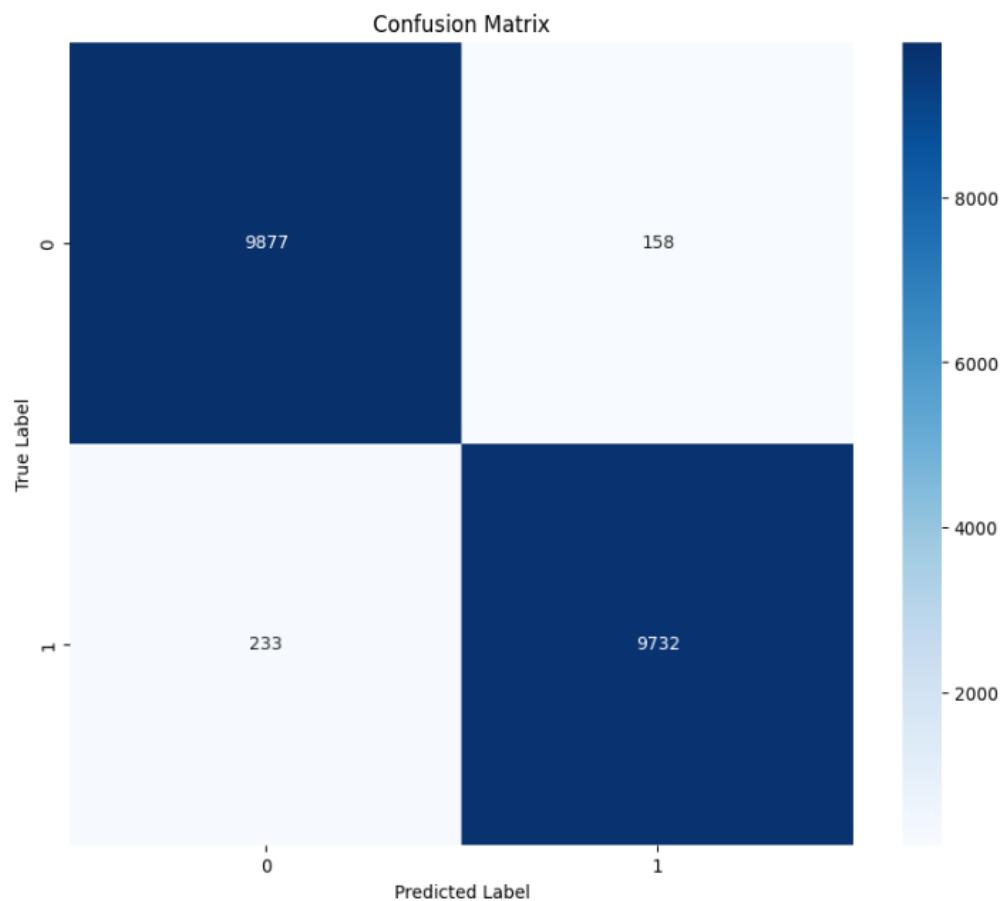
Gambar L5.6. Grafik loss model 3 LSTM



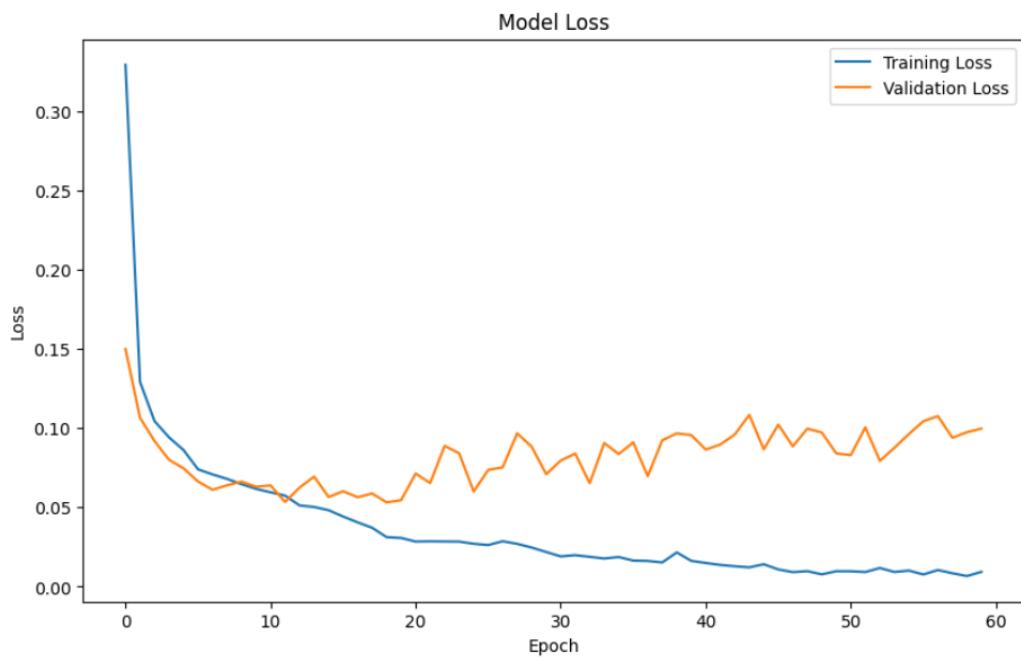
Gambar L5.7. Confusion matrix model 4 LSTM



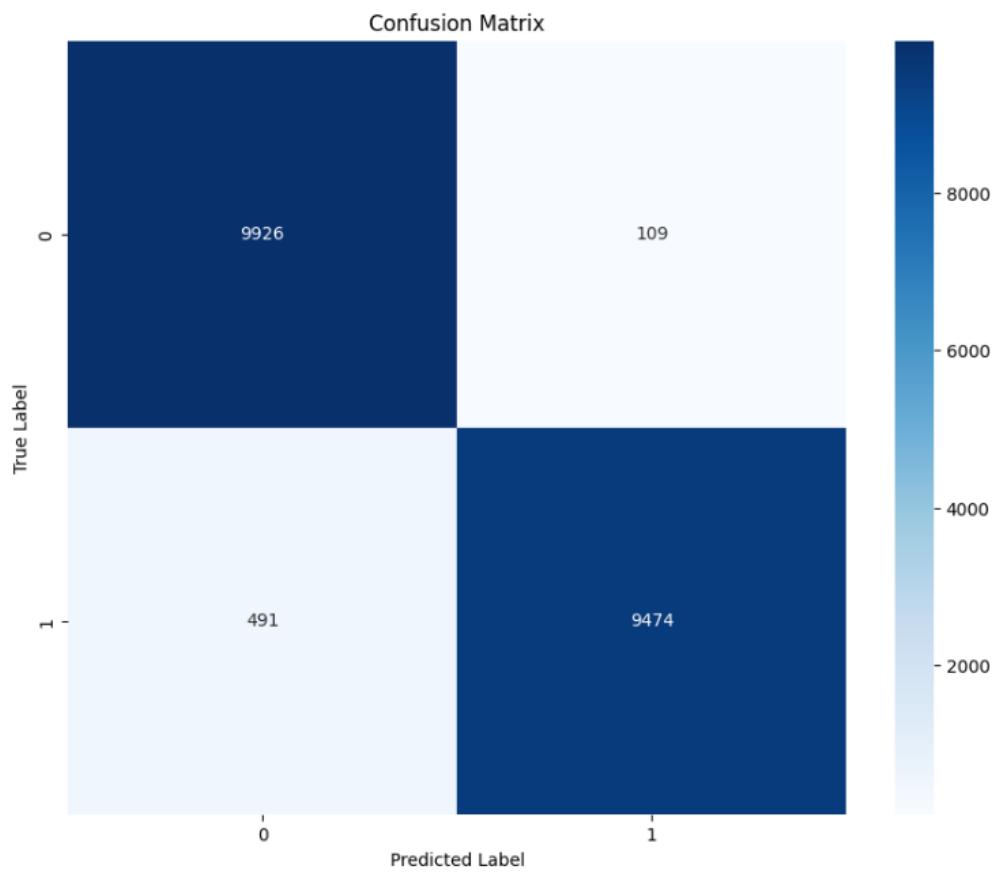
Gambar L5.8. Grafik loss model 4 LSTM



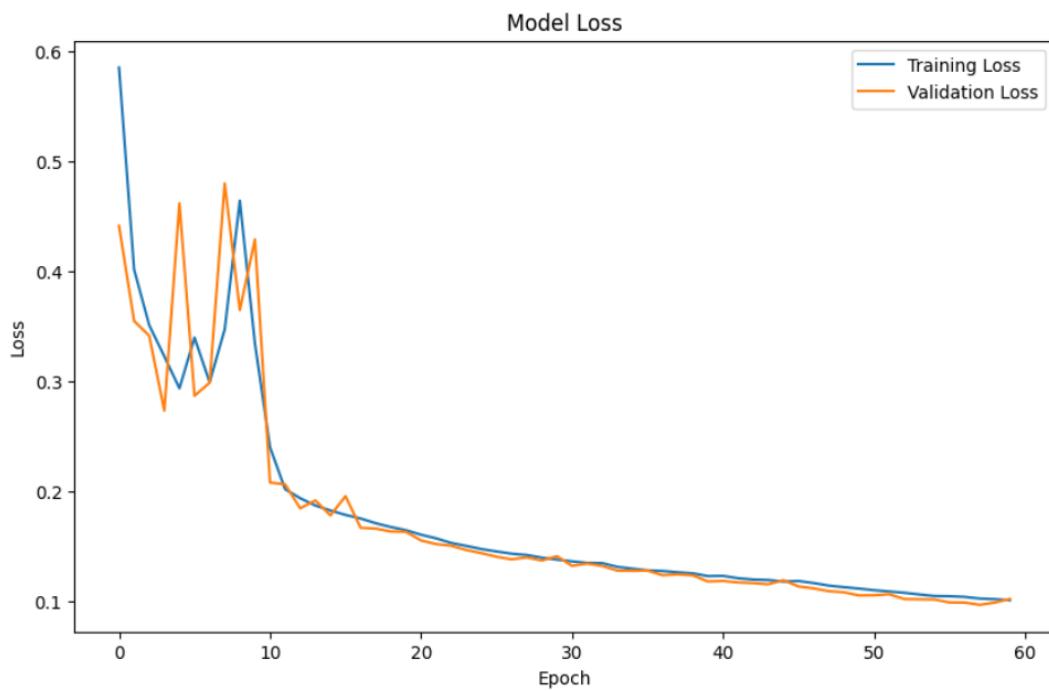
Gambar L5.9. Confusion matrix model 5 LSTM



Gambar L5.10. Grafik loss model 5 LSTM



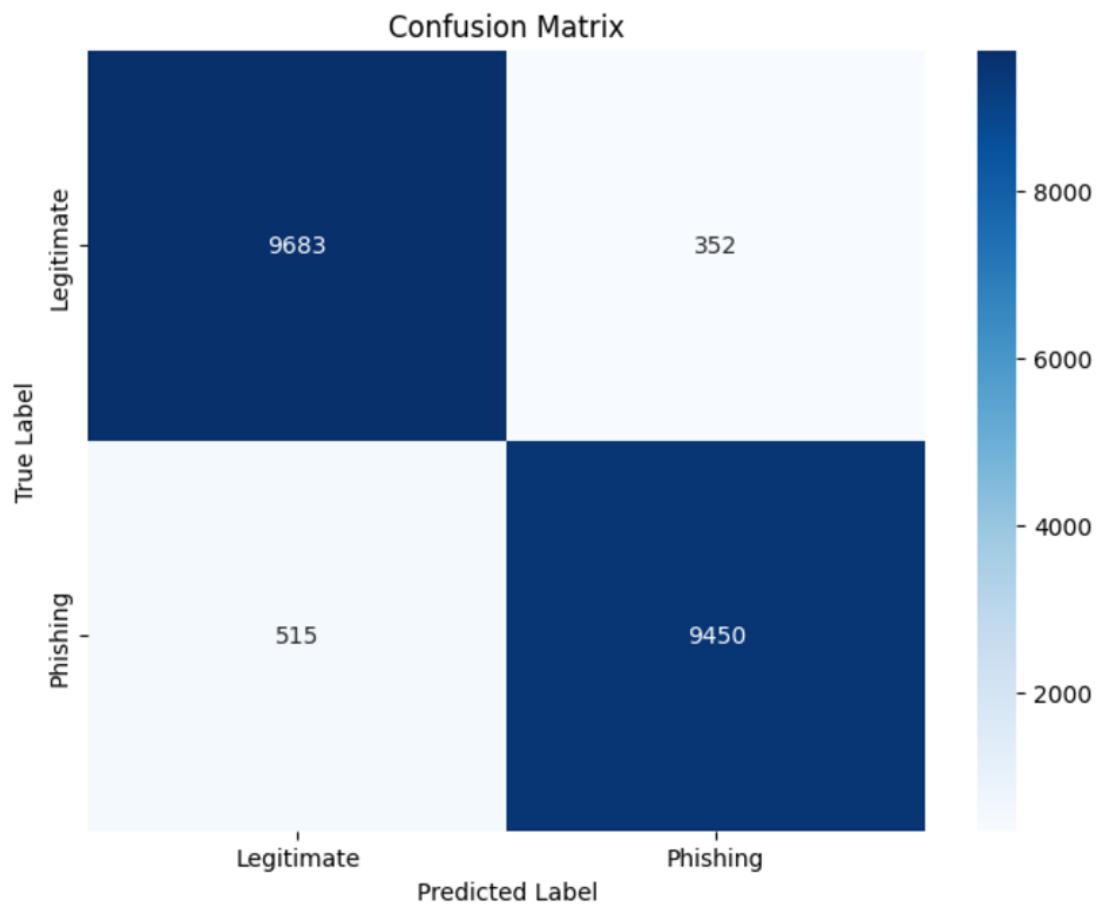
Gambar L5.11. Confusion matrix model 6 LSTM



Gambar L5.12. Grafik loss model 6 LSTM

LAMPIRAN 6

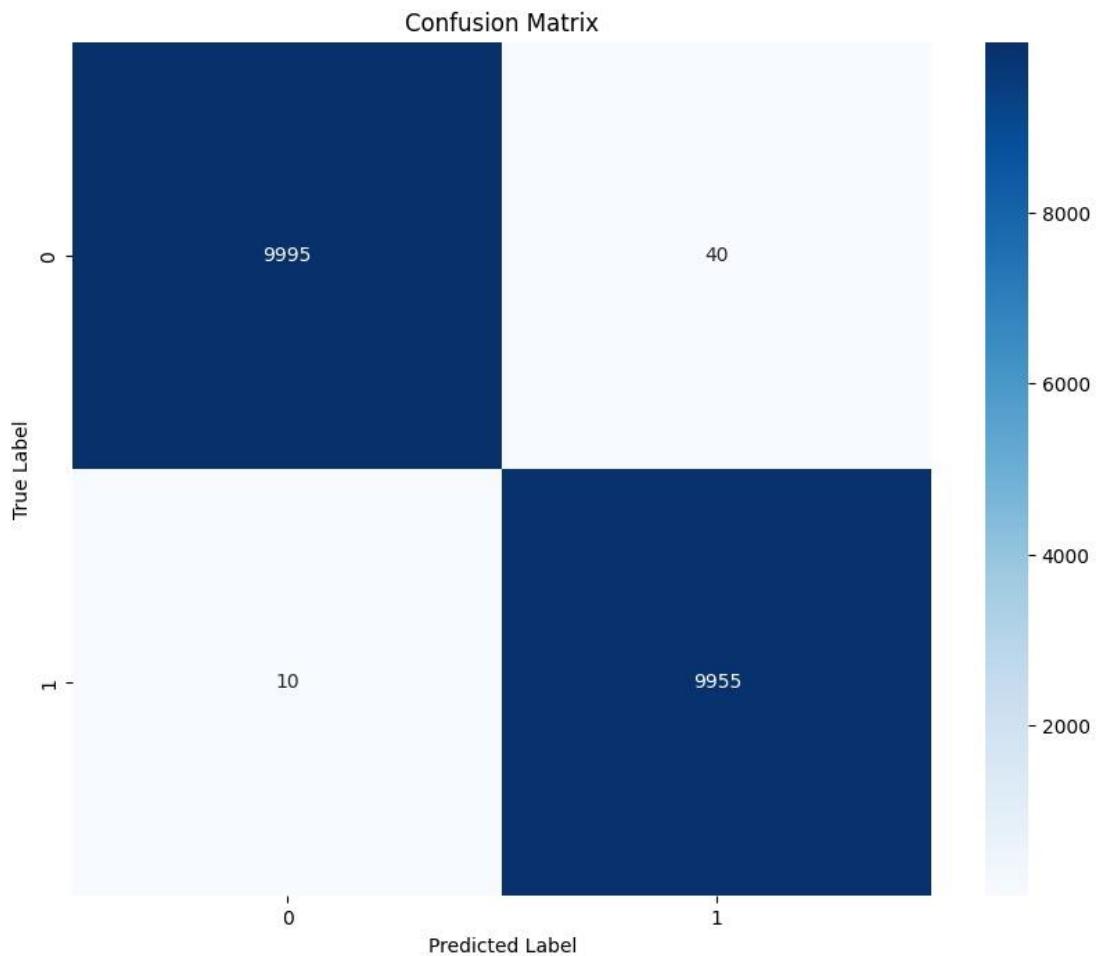
Hasil pengujian model berdasarkan arsitektur



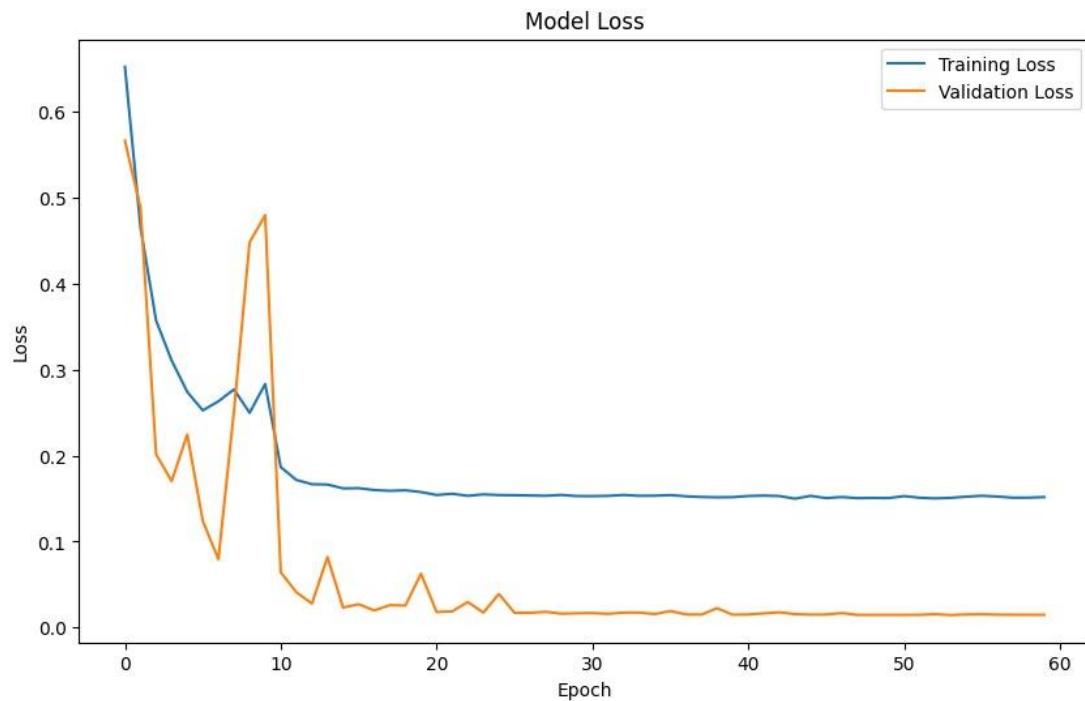
Gambar L6.1. Confusion matrix arsitektur LSTM only



Gambar L6.2. Grafik loss arsitektur LSTM only



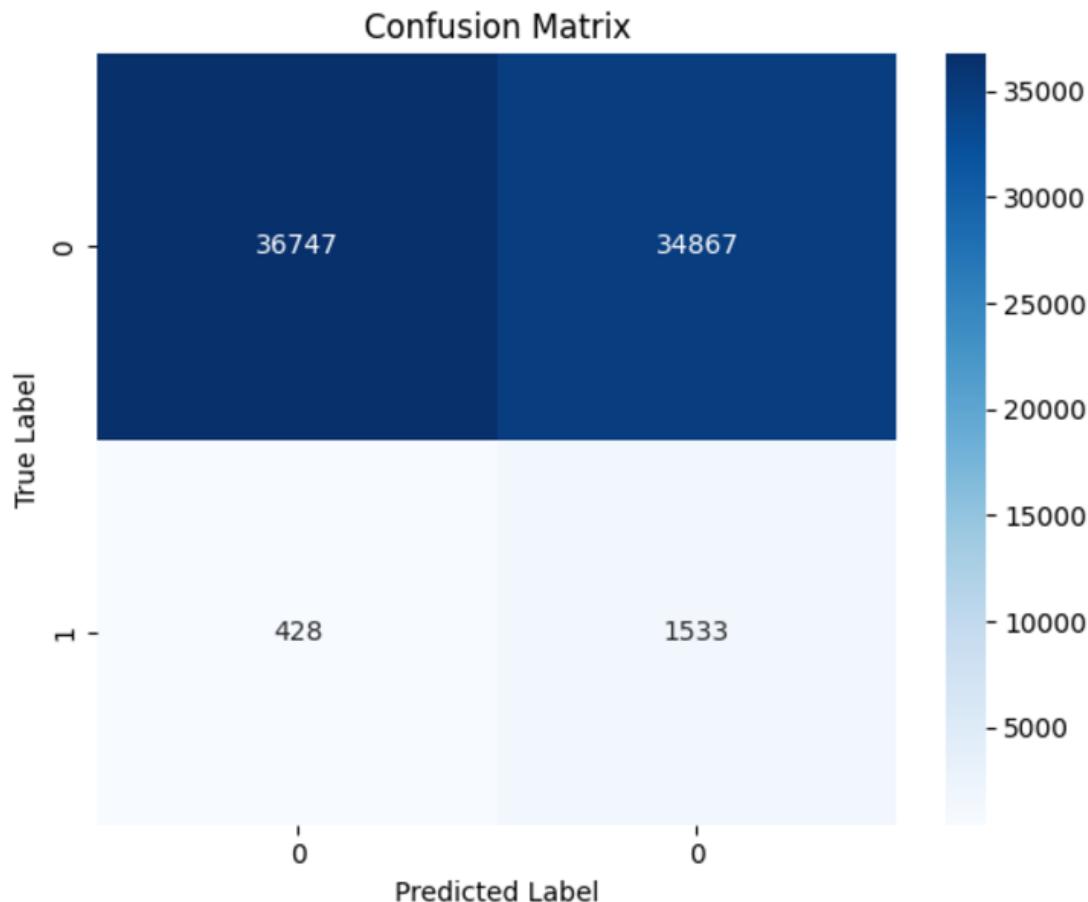
Gambar L6.3. Confusion matrix arsitektur dense only



Gambar L6.4. Grafik loss arsitektur dense only

LAMPIRAN 7

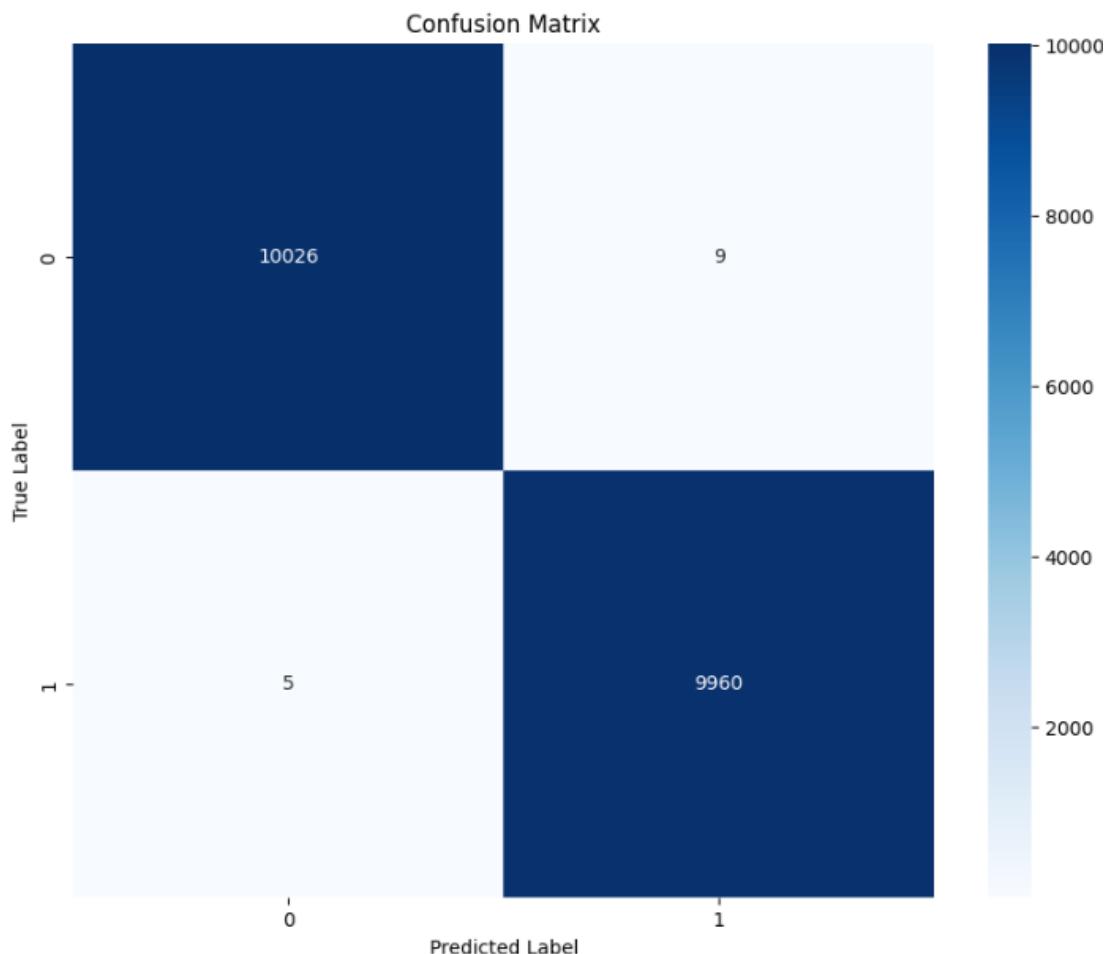
Hasil pengujian dataset Ebbu2017



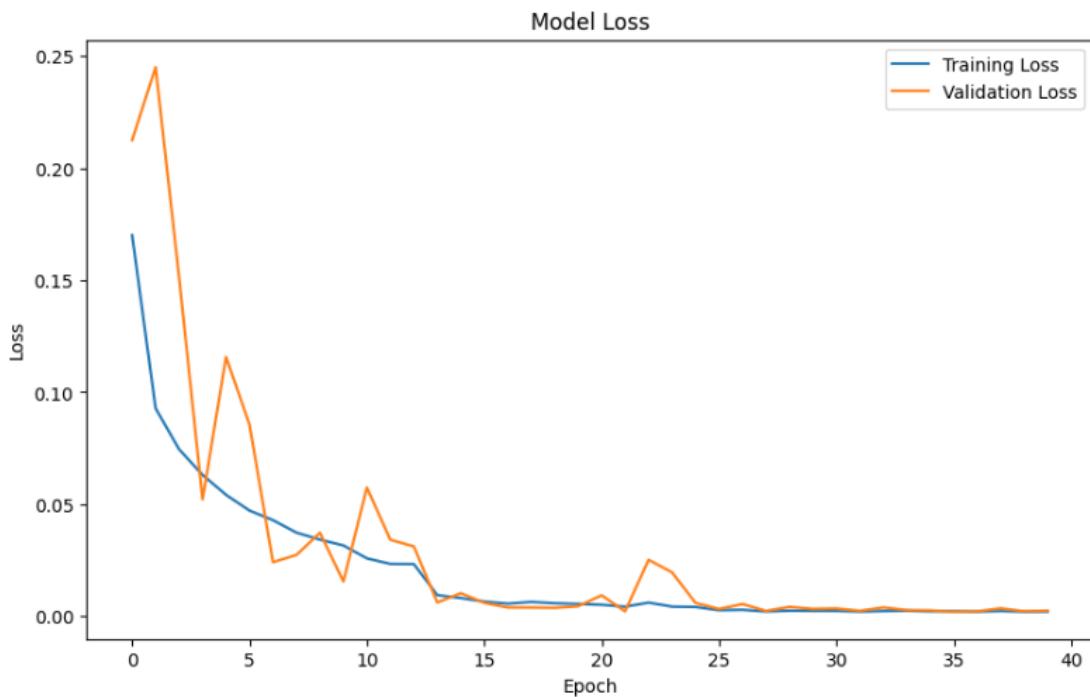
Gambar L7.1. Confusion matrix pengujian dataset Ebbu2017

LAMPIRAN 8

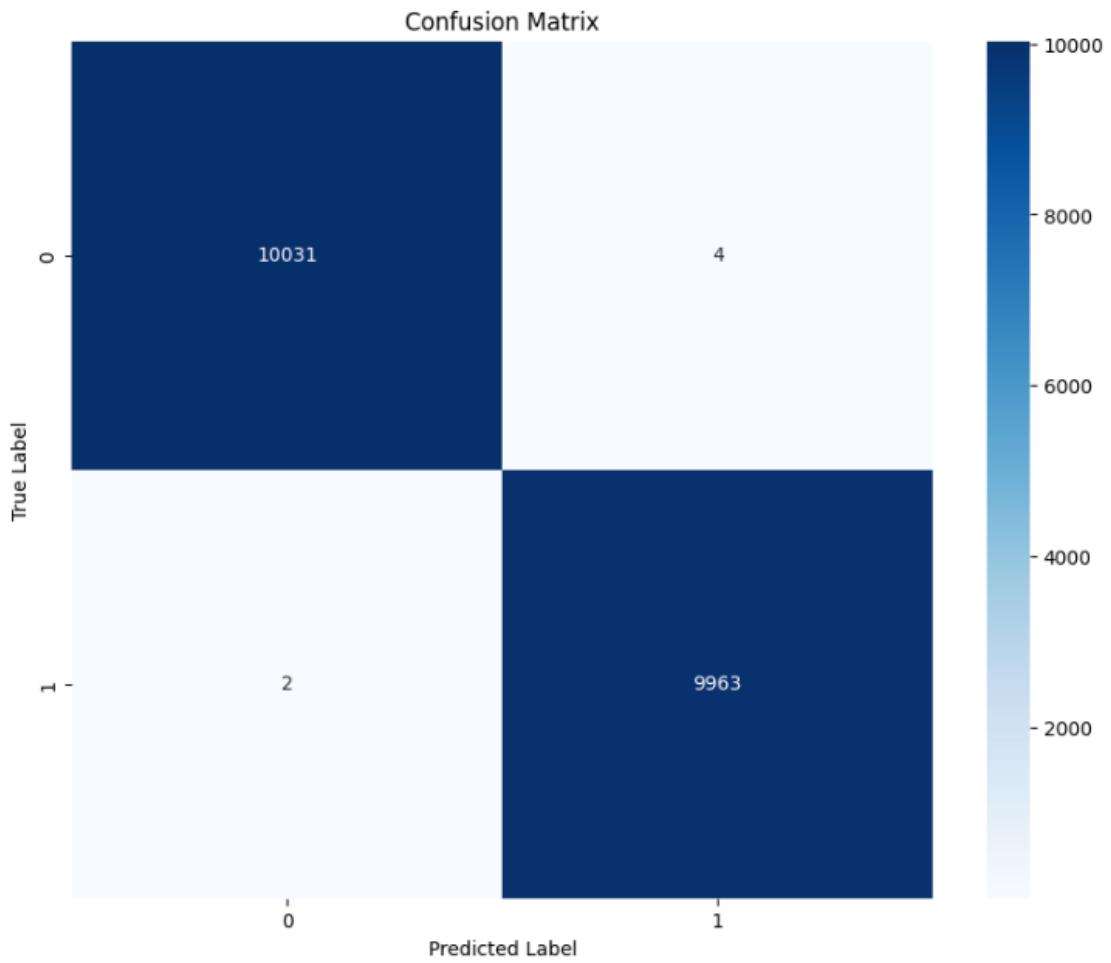
Hasil pengujian hyperparameter tuning pertama

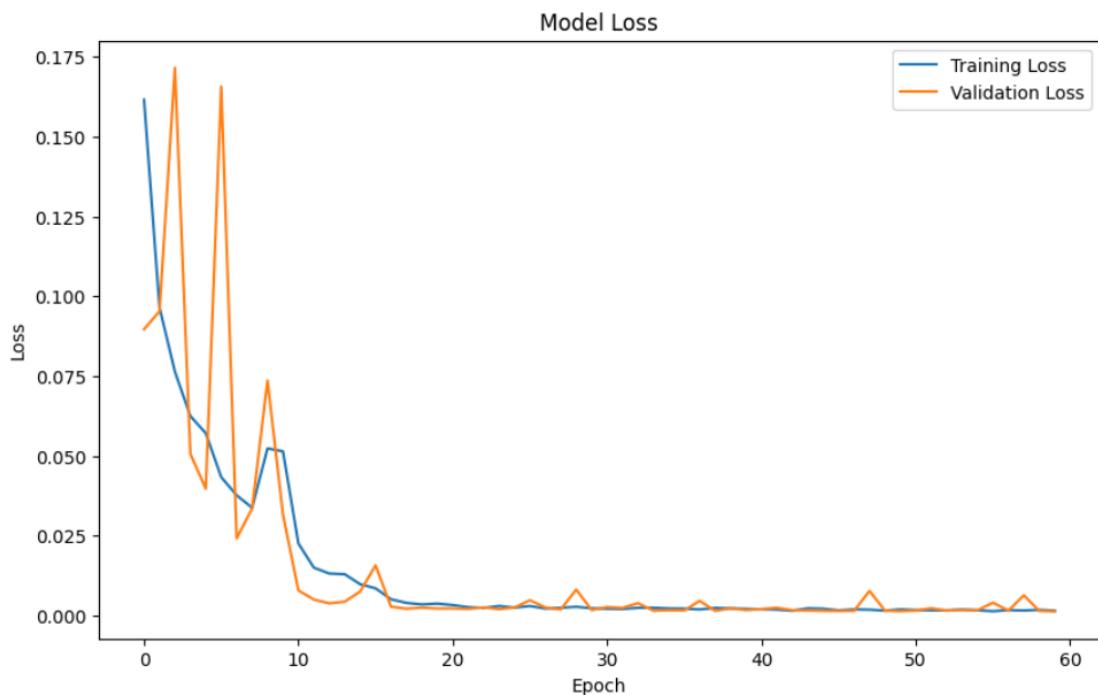


Gambar L8.1. Confusion matrix model A

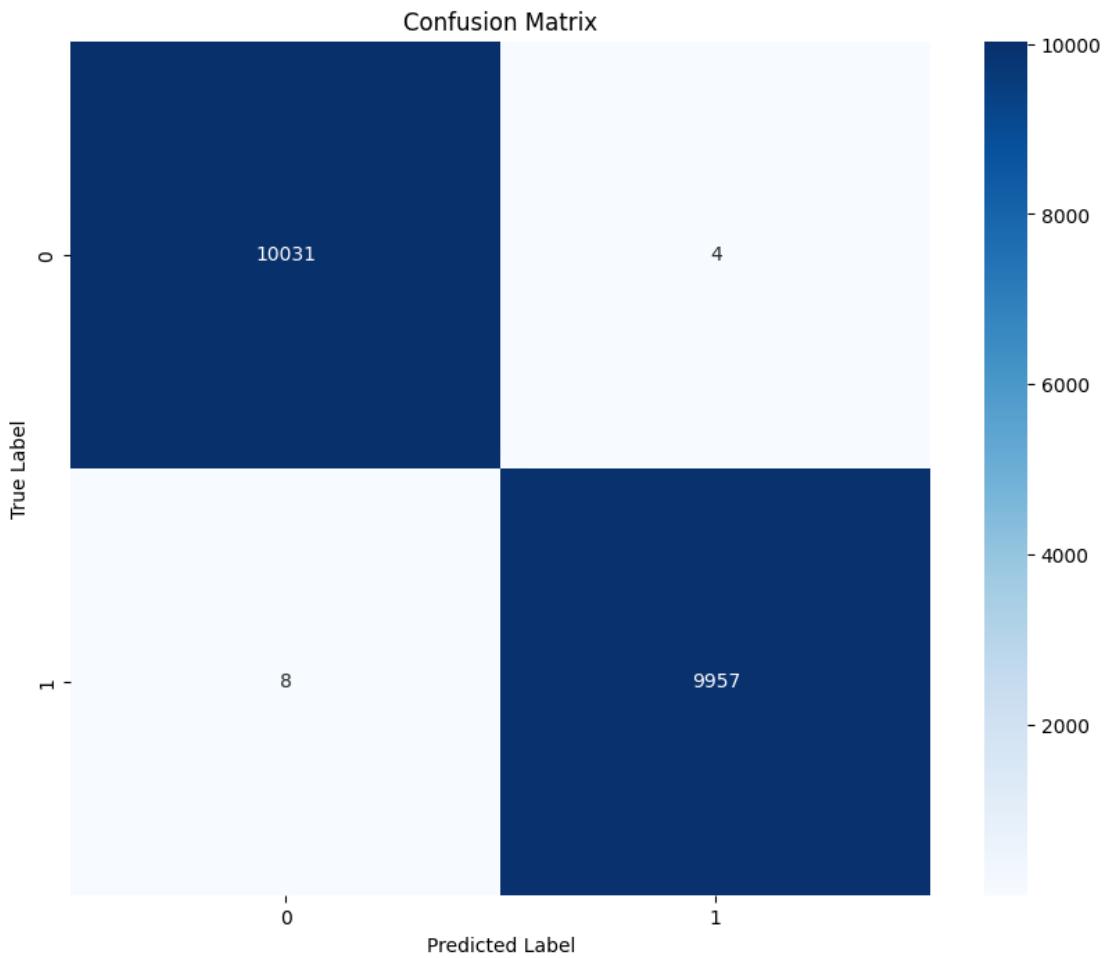


Gambar L8.2. Grafik loss model A

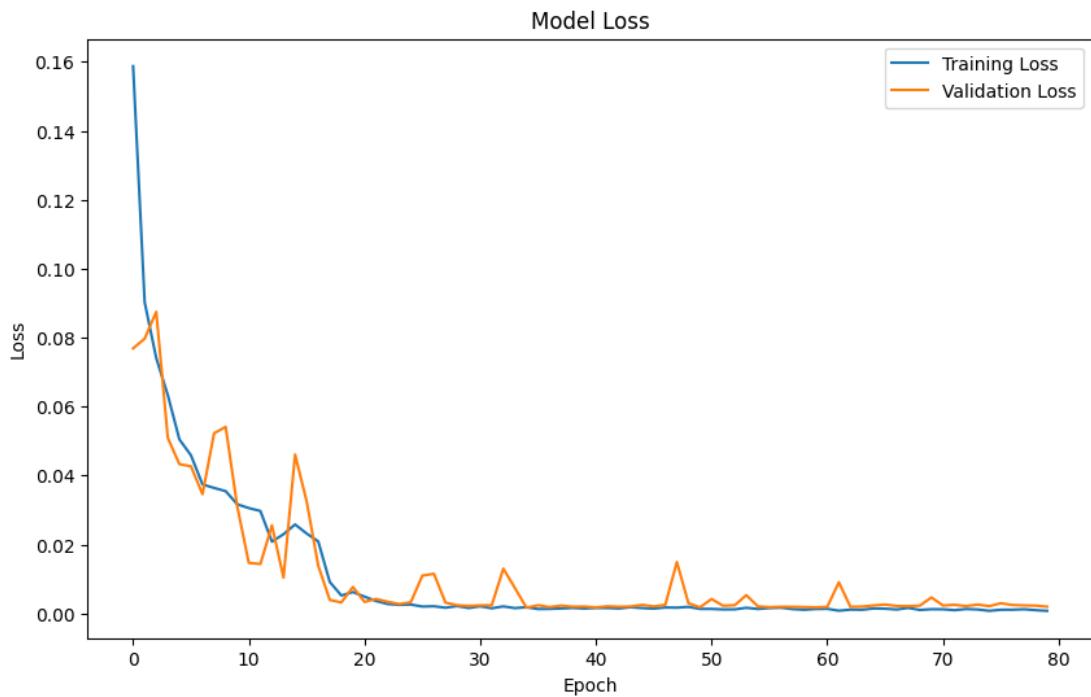




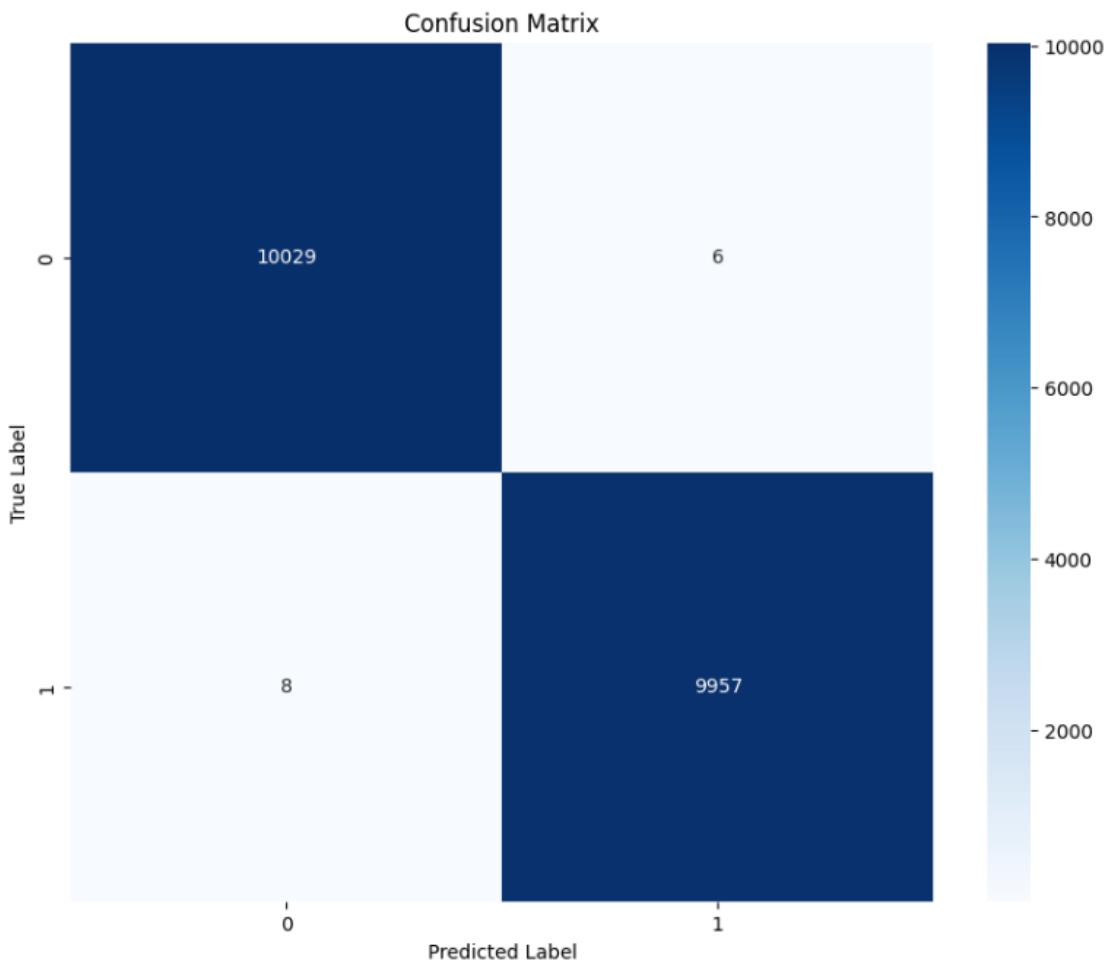
Gambar L8.4. Grafik loss model B



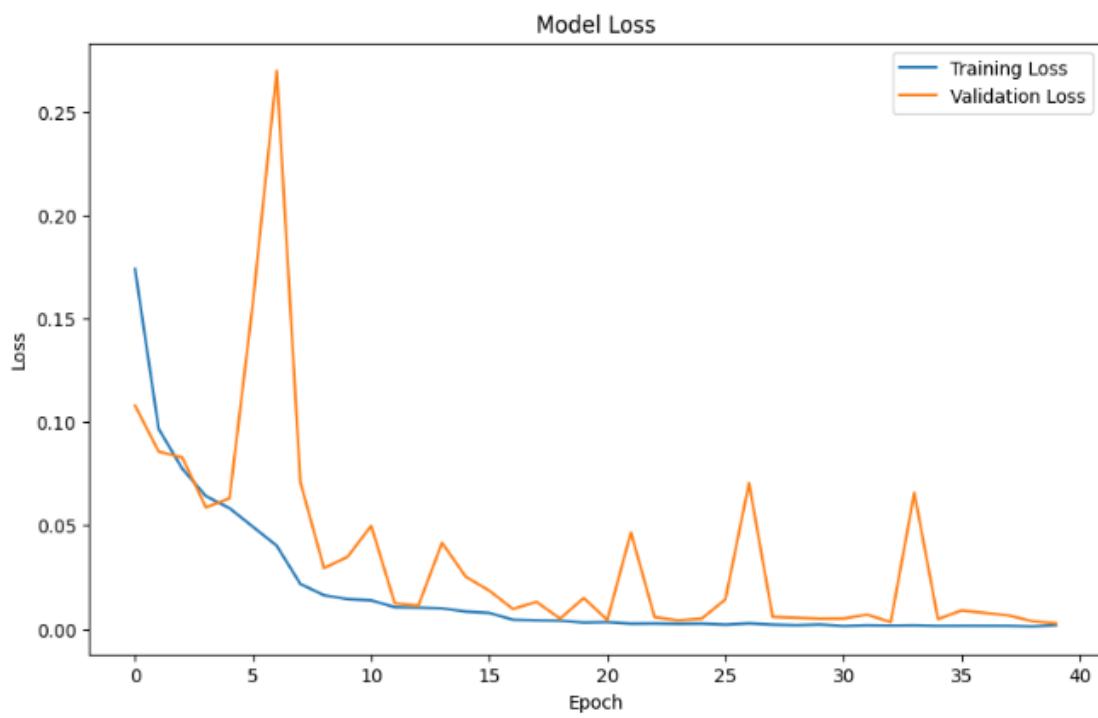
Gambar L8.5. Confusion matrix model C



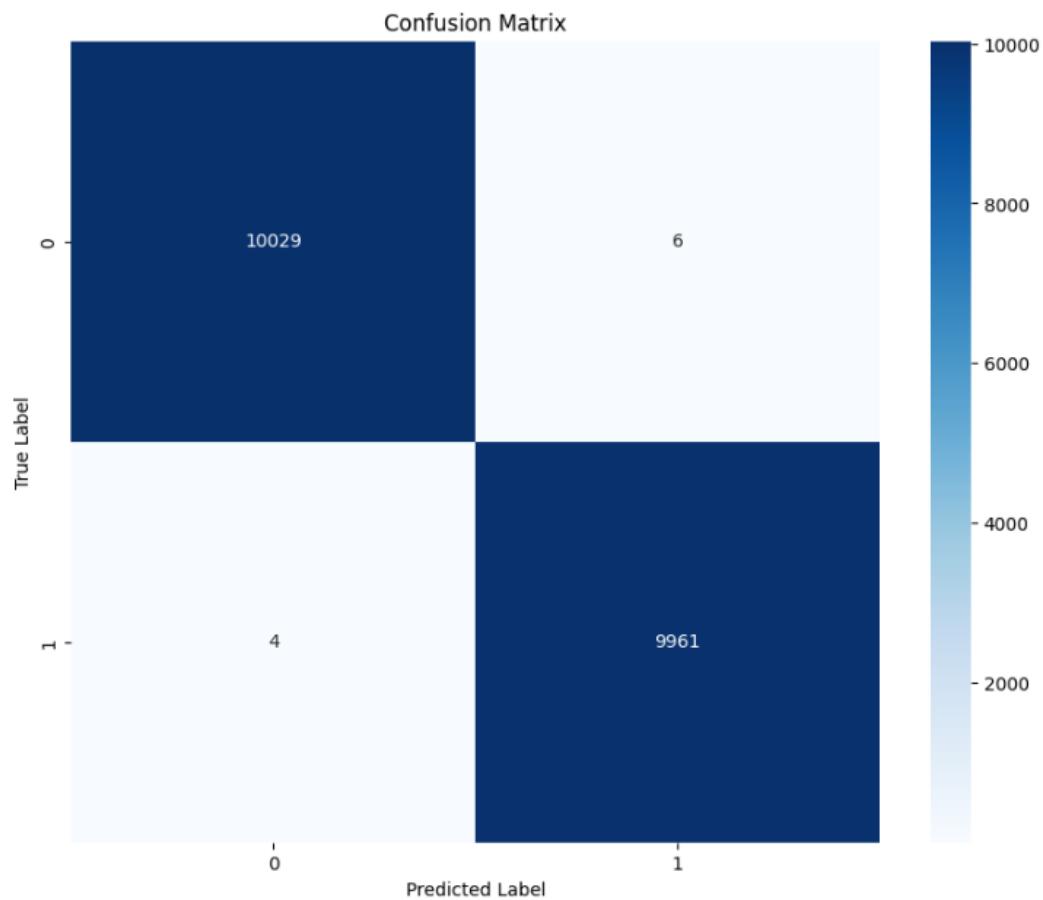
Gambar L8.6. Grafik loss model C



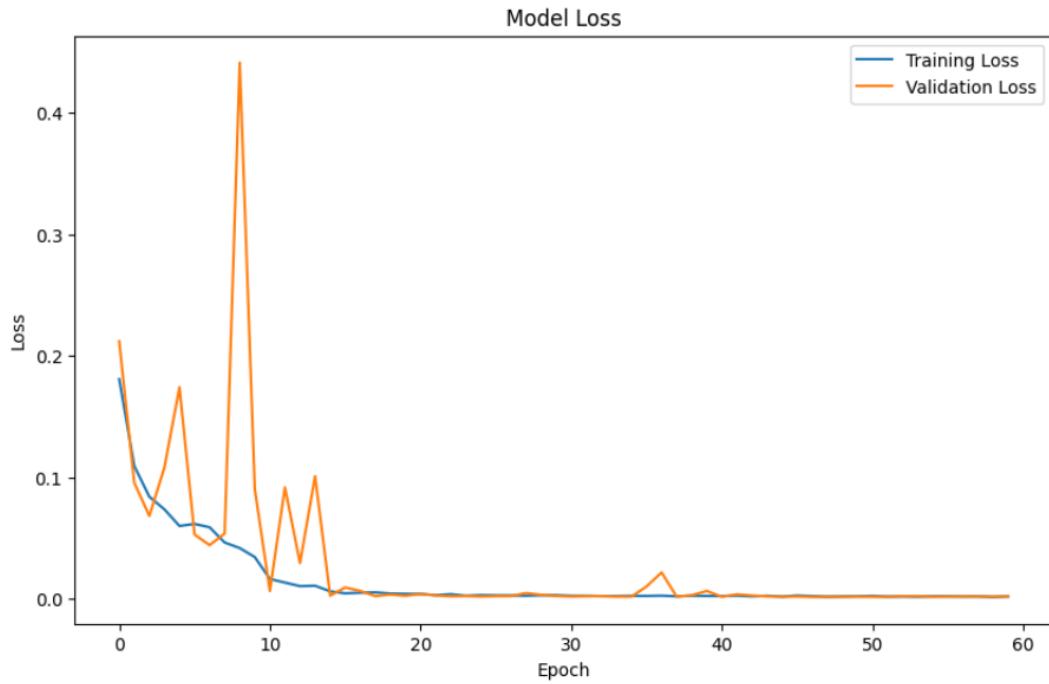
Gambar L8.7. Confusion matrix model D



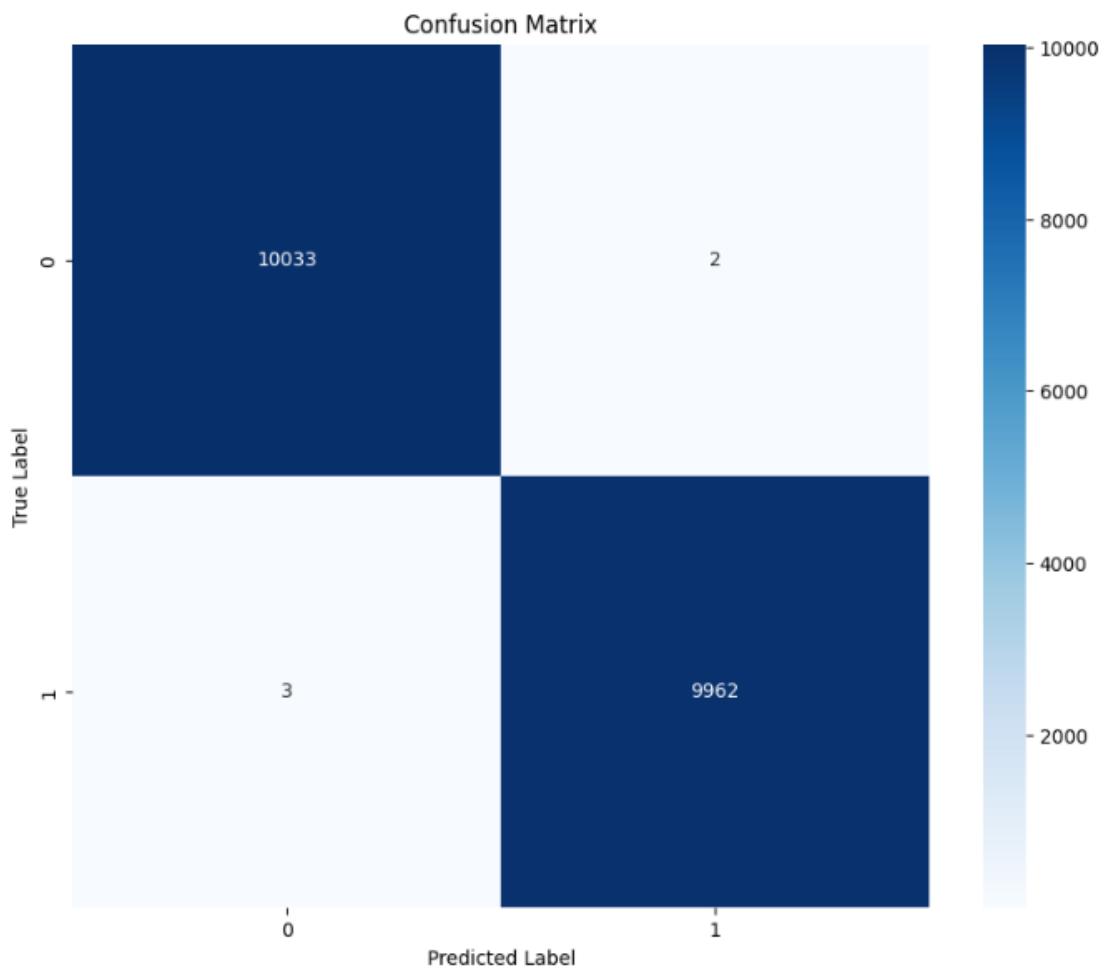
Gambar L8.8. Grafik loss model D



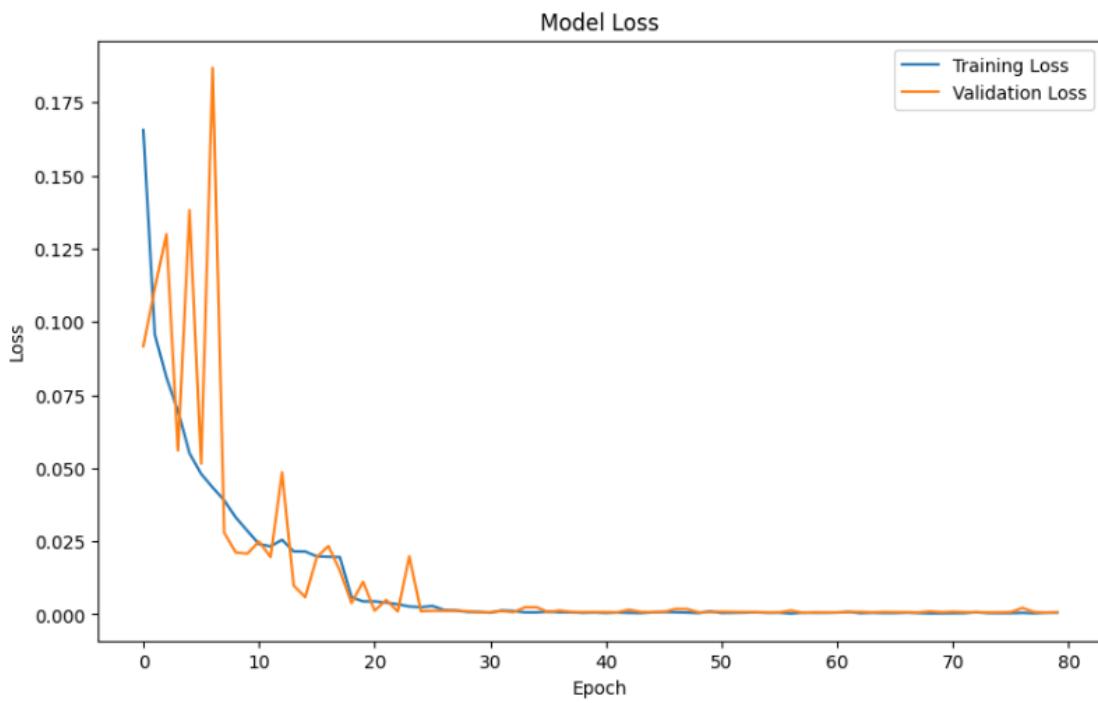
Gambar L8.9. Confusion matrix model E



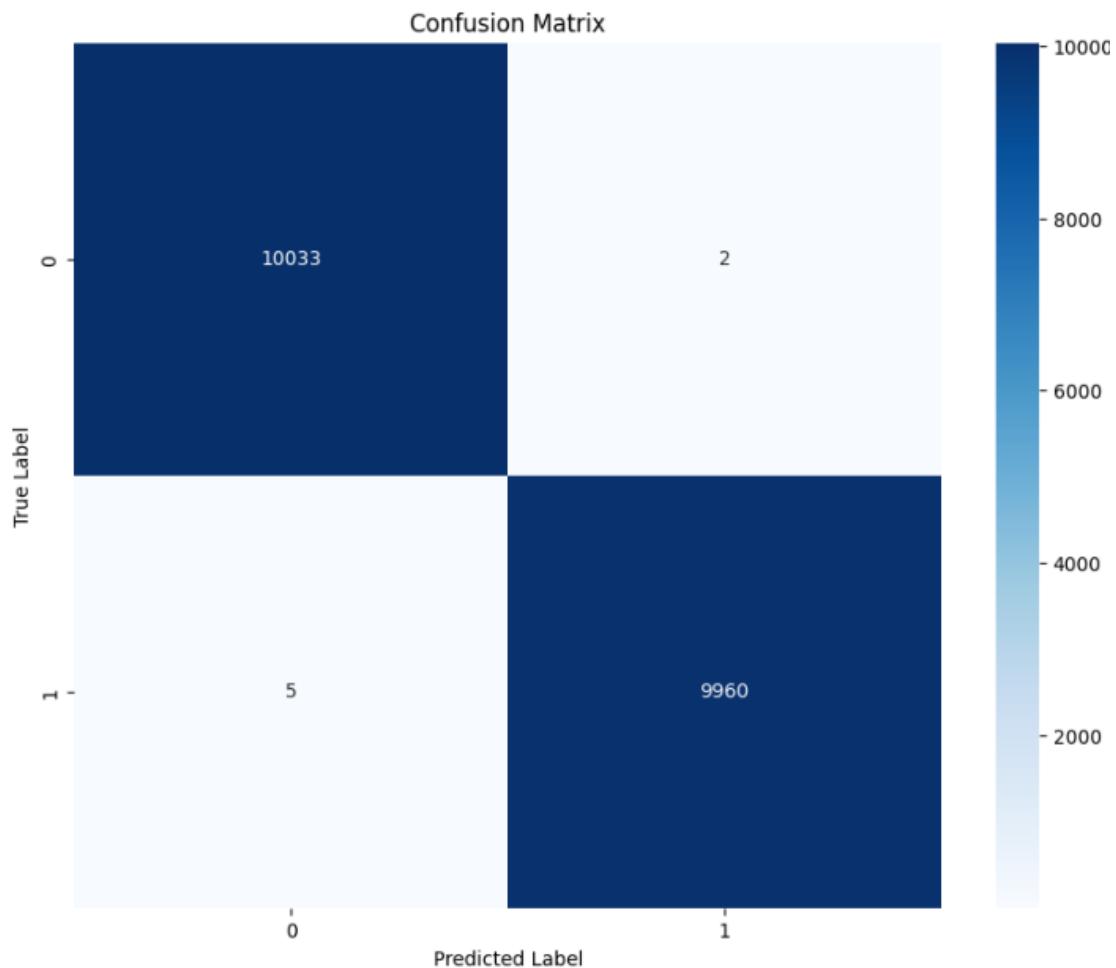
Gambar L8.10. Grafik loss model E



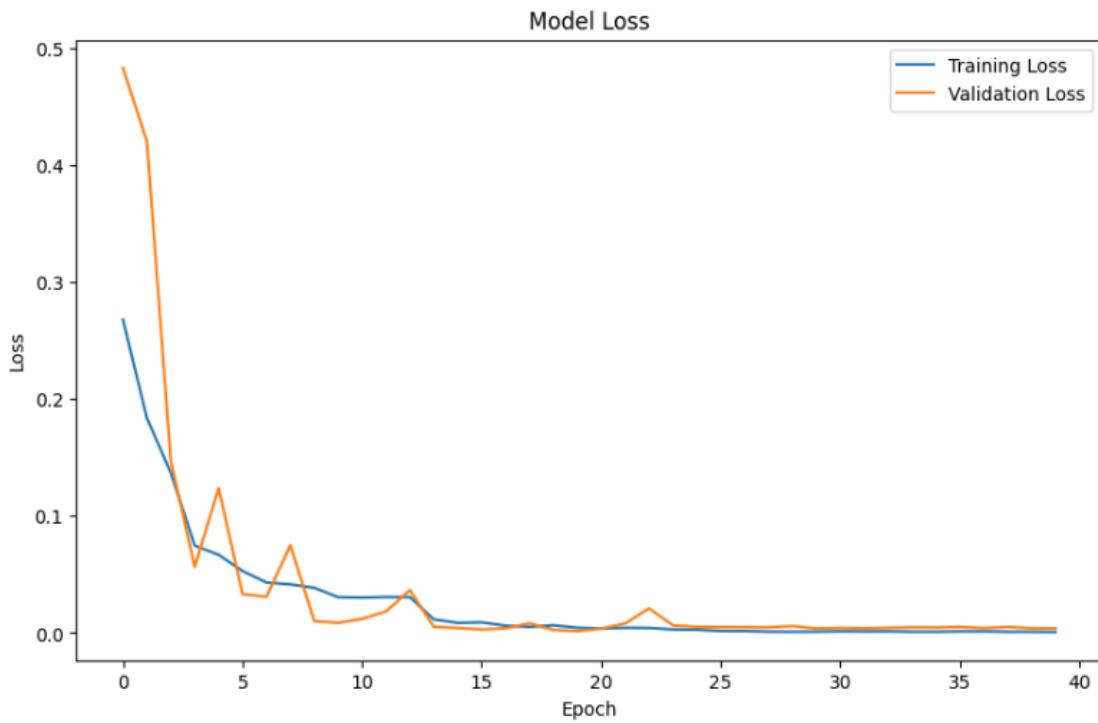
Gambar L8.11. Confusion matrix model F



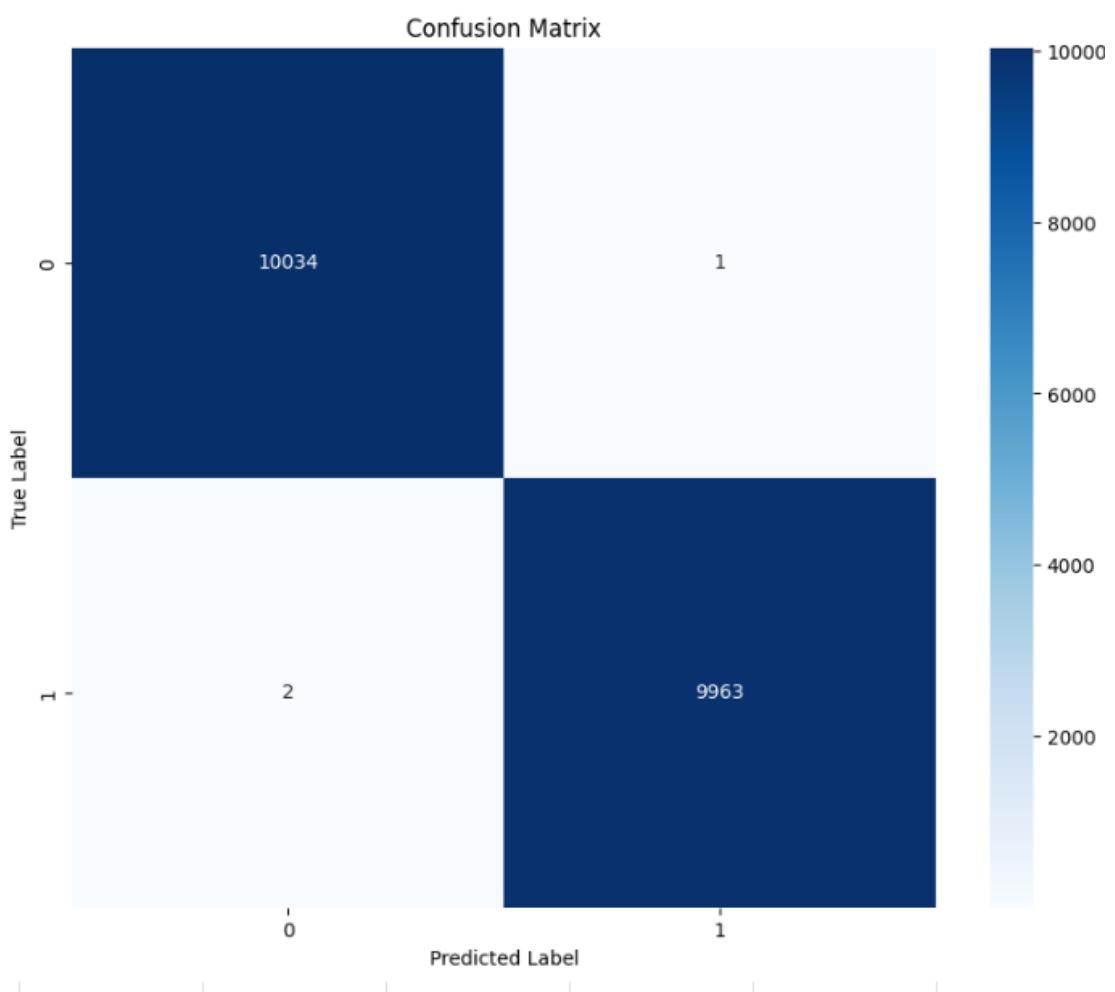
Gambar L8.12. Grafik loss model F



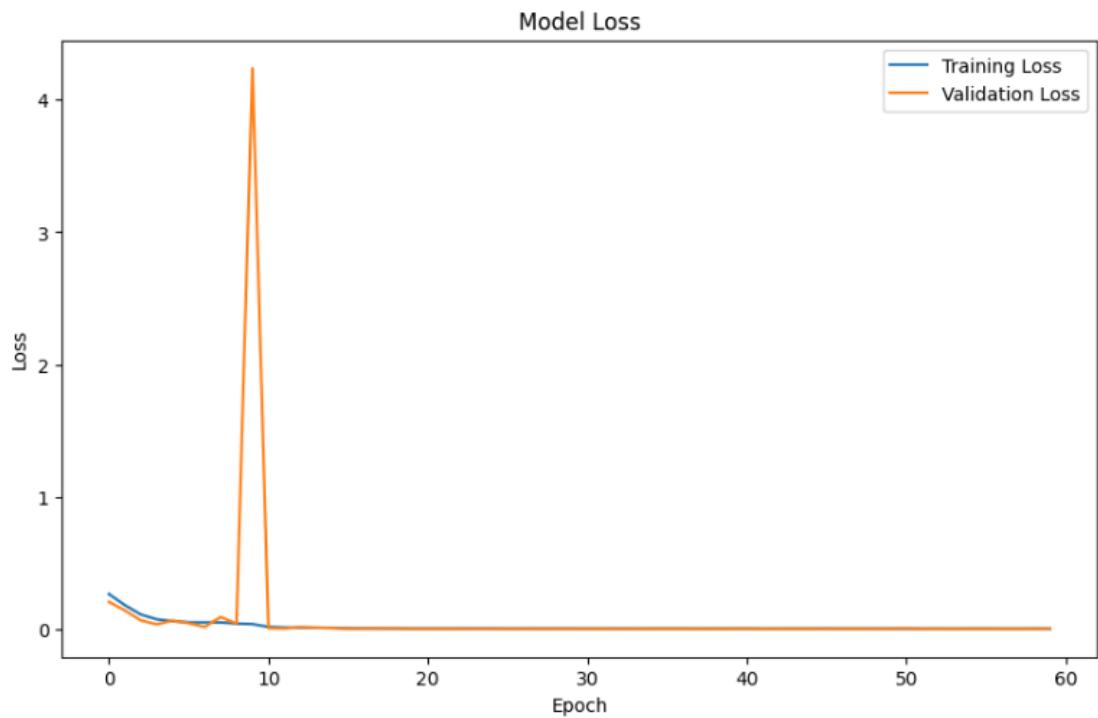
Gambar L8.13. Confusion matrix model G



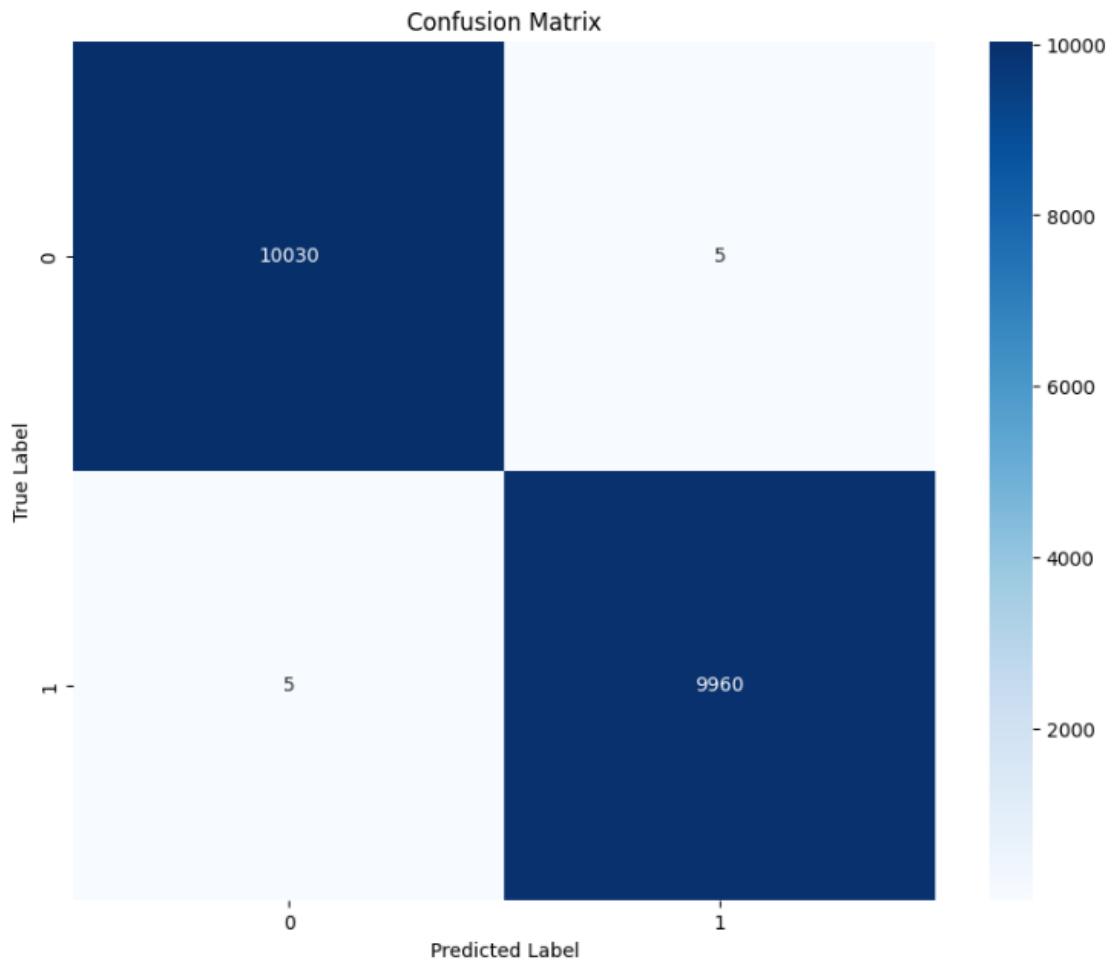
Gambar L8.14. Grafik loss model G



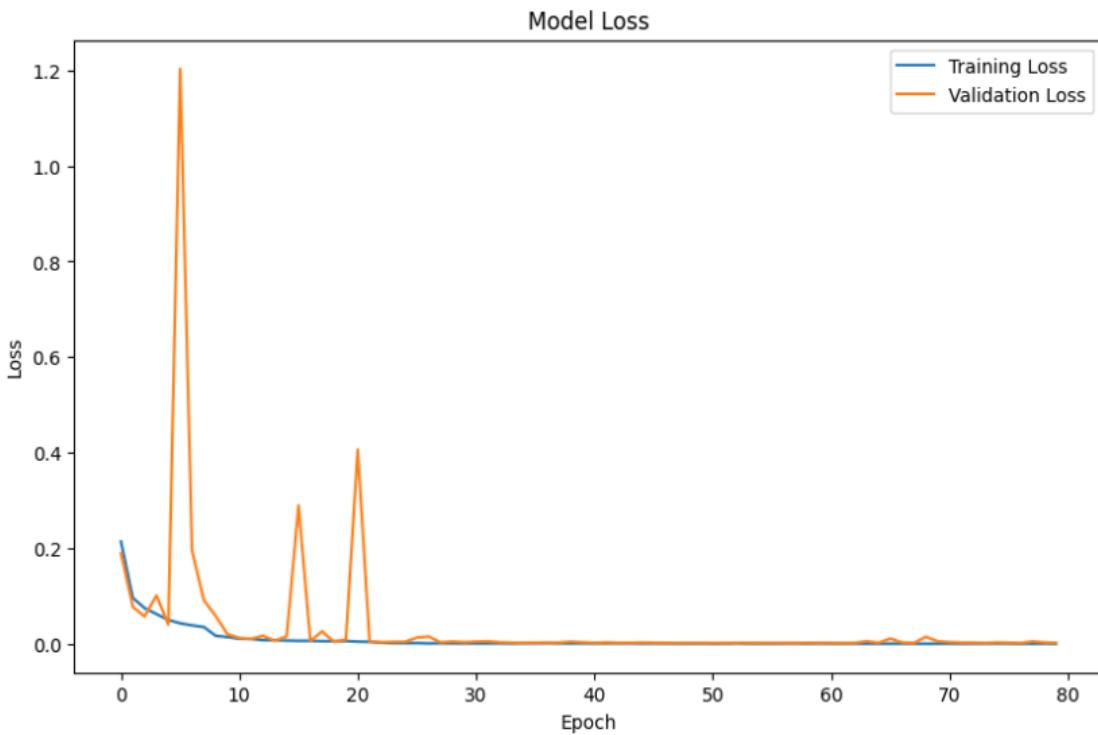
Gambar L8.15. Confusion matrix model H



Gambar L8.16. Grafik loss model H



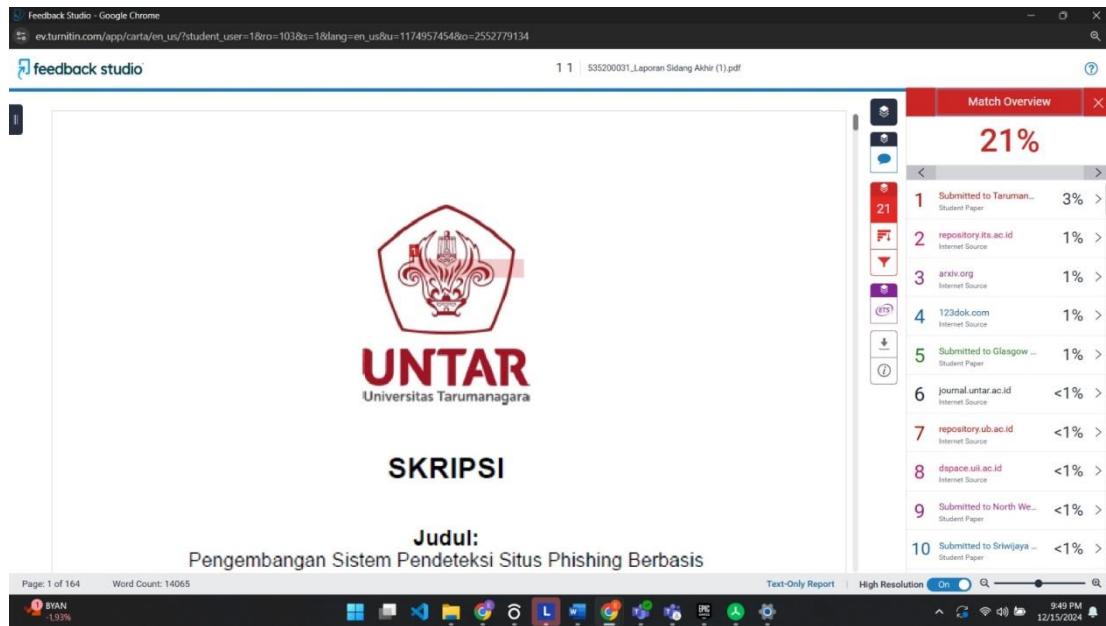
Gambar L8.17. Confusion matrix model I



Gambar L8.18. Grafik loss model I

LAMPIRAN 9

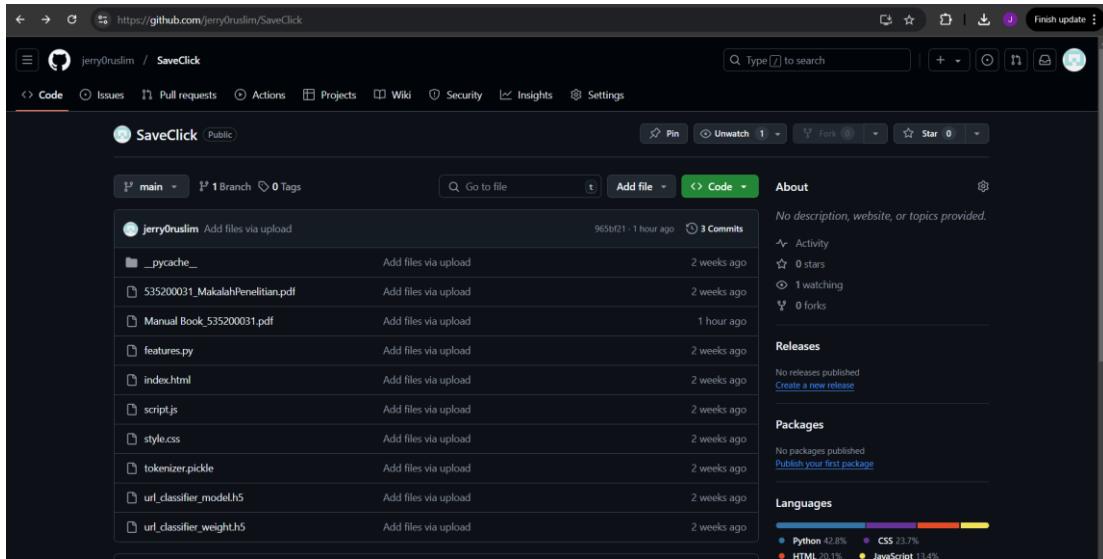
Hasil pengecekan plagiarism Turnitin



Gambar L9.1. Hasil pengecekan plagiarism

LAMPIRAN 10

Link source code GitHub



Gambar L10.1. Link source code GitHub

DAFTAR RIWAYAT HIDUP

Nama : Jerry Ruslim
NPM : 535200031
Tempat/Tanggal Lahir : Medan, 15 Desember 2002
Alamat : Apartemen Puri Orchard, Jalan Daan Mogot, Rawa Buaya, Jakarta Barat, 11740
No. Telepon/HP : 081263983931
Alamat Email : jerryruslimm@gmail.com

Riwayat Pendidikan :

- SD Swasta Chandra Kumalam, Jalan Cemara, Perumahan Cemara Asri Blok O, Kab. Deli Serdang, Sumatera Utara
- SMP Swasta Cinta Budaya, Jalan Willem Iskandar Komp. MMTC Blok Cinta Budaya No.1, Percut Sei Tuan, Deli Serdang, 20371.
- SMA Swasta Cinta Budaya, Jalan Willem Iskandar Komp. MMTC Blok Cinta Budaya No.1, Percut Sei Tuan, Deli Serdang, 20371.
- Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Tarumanagara, 2020-2024.

Riwayat Pekerjaan :

- Digital Product Associate, PT Kawan Lama Sejahtera, Februari 2023 -
- sekarang.