ANALISIS SERANGAN RECON MENGGUNAKAN CNN PADA JARINGAN IOT

SKRIPSI

Diajukan Sebagai Syarat Untuk Menyelesaikan Pendidikan Program Sarjana (S-1)
Pada Program Studi Informatika

Oleh :
KRISNA RIZKI PRATAMA
NIM : 2102020105

PROGRAM STUDI INFORMATIKA
FAKULTAS ILMU TEKNIK
UNIVERSITAS BINA INSAN
2024/2025

HALAMAN PENGESAHAN SKRIPSI

ANALISIS SERANGAN N MENGGUNAKAN CNN PADA JARINGAN IOT

Oleh :

KRISNA RIZKI PRATAMA

NIM : 2102020105

Lubuklinggau, Januari 2025

Pembimbing I	**Pembimbing II**
(Dr. Susanto, M.Kom)	**(Andri Anto Tri Susilo, M.Kom)**

Mengesahkan,

Dekan Fakultas Ilmu Teknik

Universitas Bina Insan

(Dr. Rudi Kurniawan, St., M.Kom)

HALAMAN PERSETUJUAN TIM PENGUJI SKRIPSI

Pada hari Sabtu tanggal 25 bu ri tahun 2025 telah dilaksanakan sidang Skripsi oleh Program Studi Informatika, Fakultas Ilmu Teknik Bina Insan.

Nama : Krisna Rizki Pratama

NIM : 2102020105

Judul Skripsi : Analisis Serangan Recon Menggunakan CNN pada Jaringan IoT

Komisi Penguji

1) Ketua : Dr. Susanto, M.Kom (.....)

2) Skretaris : Antri Anto Trisusilo, M.Kom (.....)

3) Anggota : Elmayati, M.Kom (.....)

Mengetahui,

Kepala Program Studi Informatika

Fakultas Ilmu Teknik

Universitas Bina Insan

(Budi Santoso, M.Kom)

HALAMAN MOTTO DAN PERSEMBAHAN

MOTTO:

- ✧ *Tidak ada hasil besar tanpa perjuangan yang besar.*
- ✧ *Langkah kecil yang konsisten akan membawa pada tujuan besar.*
- ✧ Ketika kamu merasa ingin menyerah, ingatlah alasan mengapa kamu memulai.

PERSEMBAHAN KEPADA:

- ❖ *Ayah dan Mama tercinta, yang telah banyak mendukungku dan memberikan do'a untuk keberhasilanku.*
- ❖ *Adik-adikku tercinta.*
- ❖ *Diriku sendiri, yang di tengah lelah dan sakit tetap berjuang menyelesaikan skripsi ini. Meski harus melewati hari-hari berat dengan pengobatan sindrom nefrotik, aku tetap bertahan, menolak menyerah pada keadaan. Semoga semua perjuangan ini tidak sia-sia, dan kelak ada cahaya yang menuntun menuju masa depan yang lebih baik.*
- ❖ *Teman-teman seperjuanganku.*
- ❖ *Almamaterku.*

HALAMAN PERNYATAAN

Saya yang bertanda tangan di bawah ini :

Nama Mahasiswa : Krisna Rizki Pratama
NIM : 2102020105
Program Studi : Informatika
Fakultas : Ilmu Teknik

Menyatakan dengan sesungguhnya bahwa penelitian dan penulisan Skripsi yang saya susun sebagai persyaratan untuk memperoleh gelar Sarjana (S-1) Universitas Bina Insan, merupakan hasil kerja saya sendiri dan tidak menyuruh orang lain yang mengerjakannya. Ada pun bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasilkarya orang lain dan telah saya tuliskan sumbernya secara jelas sesuai dengan norma, kaidah dan etika penulisan ilmiah.

Jika dikemudian hari ternyata terbukti bahwa penelitian dan tugas akhir ini bukan hasil kerja saya sendiri atau plagiat dalam bagian-bagian tertentu, maka saya bersedia dikenakan sanksi sesuai dengan peraturan perundangan yang berlaku.

Lubuklinggau, Januari 2025
Penulis,

(Materai 10.000)

Krisna Rizki Pratama
NIM 2102020105

ABSTRACT

The security of Internet of Th networks is one of the primary challenges in today's digital era, especially with sing number of connected IoT devices. One significant but often overlooked t connaissance (Recon) attacks, which aim to gather network information to facilitate subsequent attacks. This research aims to develop a Recon attack detection model using a combination of Autoencoder and Convolutional Neural Network (CNN).

The Autoencoder is utilized for data extraction from PCAP (Packet Capture) files, producing a simplified and informative numerical representation. The extracted data is then stored in CSV format and used as input for the CNN model, which is trained to detect suspicious patterns indicative of attacks. This study employs a dataset from the UNB Repository for training and evaluation processes. The results indicate that the developed model achieves high accuracy, exceeding 98%, with an Area Under Curve (AUC) value of 1.00 on both the ROC Curve and Precision-Recall Curve. Evaluations using k-fold cross-validation also demonstrate the model's stability, with an average accuracy of 98% for both training and validation data.

This study demonstrates that the combination of Autoencoder and CNN is a reliable solution for detecting Recon attacks on IoT networks. These findings contribute theoretically to the field of network security based on deep learning and offer a practical approach to enhancing IoT network security.

Keywords: IoT, Network Security, Recon Attack, Autoencoder, CNN

ABSTRAK

Keamanan jaringan pada Int[illegible]nings (IoT) merupakan salah satu tantangan utama di era digital saat ini, sei[illegible] meningkatnya jumlah perangkat IoT yang terhubung. Salah satu ancaman si[illegible]amun sering kali diabaikan adalah serangan Reconnaissance (Recon), yang bertujuan mengumpulkan informasi jaringan untuk memfasilitasi serangan lebih lanjut. Penelitian ini bertujuan mengembangkan model deteksi serangan Recon menggunakan kombinasi Autoencoder dan Convolutional Neural Network (CNN).

Autoencoder digunakan untuk ekstraksi data dari file PCAP (Packet Capture), menghasilkan representasi numerik yang lebih sederhana dan informatif. Data hasil ekstraksi ini kemudian disimpan dalam bentuk CSV dan digunakan sebagai input untuk model CNN, yang dilatih untuk mendeteksi pola-pola mencurigakan yang mengindikasikan serangan. Penelitian ini menggunakan dataset dari UNB Repository untuk proses pelatihan dan evaluasi. Hasil penelitian menunjukkan bahwa model yang dikembangkan mampu mencapai akurasi tinggi, yaitu di atas 98%, dengan nilai Area Under Curve (AUC) sebesar 1.00 pada ROC Curve dan Precision-Recall Curve. Evaluasi menggunakan k-fold cross-validation juga menunjukkan stabilitas model, dengan akurasi rata-rata pada data training dan validasi tetap di angka 98%.

Penelitian ini membuktikan bahwa kombinasi Autoencoder dan CNN adalah solusi yang andal dalam mendeteksi serangan Recon pada jaringan IoT. Hasil ini memberikan kontribusi teoritis di bidang keamanan jaringan berbasis deep learning serta menawarkan pendekatan praktis untuk meningkatkan keamanan jaringan IoT.

Kata Kunci: IoT, Keamanan Jaringan, Serangan Recon, Autoencoder, CNN

KATA PENGANTAR

Alhamdulillah puji dan syukur penulis ucapkan kepada Allah SWT atas segala rahmat dan karunia-Nya yang telah memberikan kekuatan dan kesempatan, sehingga penulis dapat menyelesaikan proposal skripsi ini dengan maksimal, Untuk diajukan sebagai syarat menyelesaikan pendidikan program Sarjana (S-1) Pada Program Studi Informatika Fakultas Ilmu Teknik Universitas Bina Insan. Sholawat beserta salam semoga tetap tercurahkan kepada bagi Nabi Muhammad SAW, keluarga, sahabat, serta umatnya hingga akhir zaman. Selama proses penulisan dan penyusunan proposal skripsi ini, penulis telah berusaha sebaikbaiknya untuk dapat menyelesaikan proposal skripsi ini baik tepat pada waktunya. Penulis menyadari bahwa proposal skripsi ini tentunya masih jauh dari sempurna dan mungkin terdapat kesalahan baik sengaja maupun tidak sengaja. Oleh karena itu, kritik dan saran yang membangun tentunya sangat diharapkan dari berbagai pihak.

Penulis mengucapkan banyak terima kasih kepada pihak-pihak yang telah membantu selama proses penyelesaian skripsi ini diantaranya yaitu:

1. Bapak/Ibuku yang telah banyak memberikan dukungan dan bantuannya dalam penulisan Proposal Skripsi ini.
2. Bapak Dr. H. Sardiyo, M.M. selaku Rektor Universitas Bina Insan.
3. Bapak Dr. Muhamad Akbar, S.T., M.IT selaku Wakil Rektor I Universitas Bina Insan.
4. Bapak Wakhid Nur Mukhlis, M.Pd., M.M selaku Wakil Rektor II Universitas Bina Insan.
5. Bapak Dr. Rudi Kurniawan, St., M.Kom selaku Dekan Fakultas Ilmu Teknik Universitas Bina Insan yang telah banyak memberikan bimbingan dan arah dalam penulisan skripsi ini.

6. Bapak Budi Santoso, M.Kom selaku Kepala Program Studi Informatika Fakultas Ilmu Teknik Univ Bina Insan. yang telah banyak memberikan bimbingan dan arah dalam skripsi ini.
7. Bapak Dr. Susanto, M.K ku Pembimbing I yang telah banyak memberikan bimbingan dan arah dalam penulisan Skripsi ini.
8. Bapak Antri Anto Tri Susilo, M.Kom selaku Pembimbing II yang telah banyak memberikan bimbingan dan arah dalam penulisan Skripsi ini.
9. Ibu Elmayati, M.Kom selaku Penguji yang telah banyak memberikan bimbingan dan arah dalam penulisan Skripsi ini.
10. Seluruh Staf Dosen dan Karyawan Universitas Bina Insan Lubuklinggau yang telah banyak memberikan ilmu pengetahuan dan bimbingan kepada penulis.
11. Seluruh individu yang pernah hadir dan menjadi bagian dalam perjalanan hidup penulis, yang telah memberikan motivasi serta semangat kepada penulis.

Akhir kata semoga penelitian ini dapat bermanfaat bagi untuk penelitian selanjutnya.

Lubuklinggau, Desember 2024

Penulis

DAFTAR RIWAYAT HIDUP

Biodata

Nama	: Krisna Rizki Pratama
Tempat / Tanggal Lahir	: Lubuklinggau, 27 April 2003
Jenis Kelamin	: Laki-Laki
Agama	: Islam
Alamat	: Jl Selamat, RT 02, Kelurahan Sukajadi, Kecamatan Lubuklinggau Barat I

Pendidikan

- SD : SD NEGERI 14 LUBUKLINGGAU
- SMP/MTS Sederajat : MTS NEGERI 1 LUBUKLINGGAU
- SMA/SMK Sederajat : SMK MUHAMMADIYAH LUBUKLINGGAU

Pengalaman Organisasi dan Pelatihan

1. Lembaga Dakwah Kampus (LDK)
2. Lembaga Pelatihan Komputer

Prestasi Akademik dan Non-Akademik

No	Prestasi Akademik dan Non-Akademik	Tahun
1.	Salah Satu Peserta Terbaik MSIB Batch 6 di Dicoding Indonesia	2024

DAFTAR ISI

JUDUL .....
HALAMAN PENGESAHAN i
HALAMAN PERSETUJUAN ii
HALAMAN MOTTO DAN PERSEMBAHAN iii
HALAMAN PERNYATAAN iv
ABSTRACT v
ABSTRAK vi
KATA PENGANTAR vii
DAFTAR RIWAYAT HIDUP ix
DAFTAR ISI x
DAFTAR TABEL xii
DAFTAR GAMBAR xiii
DAFTAR LAMPIRAN xv
BAB I PENDAHULUAN 1
1.1 Latar Belakang Penelitian 1
1.2 Identifikasi Masalah 2
1.3 Rumusan Masalah 3
1.4 Batasan Masalah 3
1.5 Tujuan dan Manfaat 4
1.5.1 Tujuan Penelitian 4
1.5.2 Manfaat Penelitian: 4
BAB II KAJIAN PUSTAKA 5
2.1 Literatur 5
2.1.1 Analisis 5
2.1.2 *Internet of Things* (*IoT*) 5
2.1.3 *Reconnaissance* (*Recon*) 6
2.1.4 *Deep Learning* 7
2.1.5 *Convolutional Neural Networks* (*CNN*) 8
2.1.6 *Autoencoder* 14
2.1.7 Python 15

2.2 Penelitian Relevan 16
2.3 Kerangka Berpikir 18
BAB III METODOLOGI PE N 20
3.1 Analisa Sistem 20
3.1.1 Analisa Sistem yang Berjalan 20
3.1.2 Alternatif Pemecahan Masalah 21
3.1.3 Metode Analisa 21
3.2 Teknik Pemilihan Informan 27
3.2.1 Teknik Pengumpulan Data 28
3.2.2 Teknik Analisa Data 30
3.3 Tempat dan Waktu Penelitian 31
3.3.1 Tempat Penelitian 31
3.3.2 Waktu Penelitian 31
3.4 Alat dan Bahan 31
3.4.1 Alat 31
3.4.2 Bahan 32
BAB IV HASIL PENELITIAN DAN PEMBAHASAN 33
4.1 Gambaran Umum 33
4.1.1 Gambaran Umum 33
4.1.2 Struktur Organisasi 33
4.2 Hasil 34
4.3 Pembahasan 34
4.3.1 Penerapan Metode Analisa dan Validitas Data 34
4.3.2 Pengujian Hasil Analisa 36
BAB V KESIMPULAN DAN SARAN 77
5.1 Kesimpulan 77
5.2 Saran 78
DAFTAR PUSTAKA 79
LAMPIRAN 83

DAFTAR TABEL

Tabel 2.1. Penelitian Relavan 16
Tabel 3.2. Waktu Penelitian 31
Tabel 4.3. Sample Data Hasil Autoencoder 35

DAFTAR GAMBAR

Gambar 2.1. Proyeksi jumlah perangkat IoT di seluruh dunia berdasarkan laporan IoT Analytics (2024) [3]..... 6
Gambar 2.2. Arsitektur CNN [12] 9
Gambar 2.3. Proses konvolusi [13] 10
Gambar 2.4. Pooling Layer [13] 11
Gambar 2.5. jaringan saraf standar (A) dan jaringan setelah menerapkan dropout (B) [14]..... 12
Gambar 2.6. Confusion Matrix [15] 13
Gambar 2.7. Kerangka Berpikir 19
Gambar 3.8. Diagram Alur Penelitian 20
Gambar 3.9. Contoh Tampilan Data Serangan dalam Format PCAP yang Dibuka dengan Wireshark..... 29
Gambar 3.10. Contoh Tampilan Data Serangan dalam Format PCAP yang Dibuka dengan Wireshark 29
Gambar 4.11. Struktur Organisasi Universitas Bina Insan 33
Gambar 4.12 Model Loss and Model Accuracy 2 Feature Binnary Classification 36
Gambar 4.13 Confussion Matrix 2 Feature Binnary Classification 37
Gambar 4.14 Classification Report 2 Feature Binnary Classification..... 38
Gambar 4.15 FPR dan FNR 2 Feature Binnary Classification..... 39
Gambar 4.16 ROC Curve 2 Feature Binnary Classification 40
Gambar 4.17 ROC 2 Feature Binnary Classification 41
Gambar 4.18 Precission Recall Curve 2 feature Binnary Classification 42
Gambar 4.19 Model Performance Matrix 2 feature Binnary Classification..... 43
Gambar 4.20 Training and Validation Accuracy For Each Fold..... 43
Gambar 4.21 Model Loss and Model Accuracy 2 Feature Multi Classification . 45
Gambar 4.22 Confussion Matrix 2 Feature Multi Classification 46
Gambar 4.23 Classification Report 2 Feature Multi Classification..... 48
Gambar 4.24 FNR dan FPR 2 Feature Multi Classification..... 49
Gambar 4.25 ROC Curve 2 Feature Multi Classification 50
Gambar 4.26 ROC 2 Feature Multi CLassification..... 52
Gambar 4.27 Precision Recall Curve 2 Feature Multi Classification..... 53
Gambar 4.28 Model Performance Matrix 2 Feature Multi Classification..... 54
Gambar 4.29 Training and Validation Accuracy For Each Fold..... 55
Gambar 4.30 Model Loss and Model Accuracy 3 Feature Binnary Classification 56
Gambar 4.31 Confussion Matrix 3 Feature Binnary Classification 57
Gambar 4.32 Classification Report 3 Feature Binnary Classification..... 58
Gambar 4.33 FPR dan FNR 3 Feature Binnary Classification..... 58

Gambar 4.34 ROC Curve 3 Feature Binnary Classification 59
Gambar 4.35 ROC 3 Feature Binnary Classification 60
Gambar 4.36 Precision Recall Curve 3 Feature Binnary Classification..... 61
Gambar 4.37 Model Performance Matrix 3 Feature Binnary Classification..... 62
Gambar 4.38 Training and Validation Accuracy For Each Fold 3 Feature Binnary Classification..... 63
Gambar 4.39 Model Loss and Model Accuracy 3 Feature Multi Classification . 64
Gambar 4.40 Confussion Matrix 3 Feature Multi Classification 65
Gambar 4.41 Classification Report 3 Feature Multi Classification..... 67
Gambar 4.42 FPR dan FNR 3 Feature Multi CLassification 69
Gambar 4.43 ROC Curve 3 Feature Multi Classification 70
Gambar 4.44 ROC 3 Feature Multi Classification 72
Gambar 4.45 Precision Recall Curve 3 Feature Multi Classification..... 73
Gambar 4.46 Model Performance Matrix 3 Feature Multi Classification..... 75
Gambar 4.47 Training and Validation Accuracy For Each Fold 3 Feature Multi Classification..... 76

DAFTAR LAMPIRAN

Lampiran 1. Lembar pengesah 83
Lampiran 2. Lembar bimbinga l p1 85
Lampiran 3. Lembar bimbinga l p2 86
Lampiran 4. Lembar perbaikan seminar proposal skripsi 87
Lampiran 5. Lembar bimbingan skripsi p1 88
Lampiran 6. Lembar bimbingan skripsi p2 89
Lampiran 7. Lembar perbaikan skripsi..... 90

BAB I
PENDAHULUAN

1.1 Latar Belakang Penelitian

Keamanan siber telah menjadi salah satu fokus utama di era digital, terutama dalam konteks *Internet of Things (IoT)* yang terus berkembang pesat [1]. IoT adalah teknologi yang memungkinkan berbagai perangkat, seperti komputer, ponsel pintar, televisi pintar, perangkat rumah tangga, hingga sensor industri, saling terhubung dan berkomunikasi. Perangkat-perangkat ini dapat bertukar data secara otomatis melalui infrastruktur jaringan yang ada, seperti *Internet*, untuk mendukung beragam aplikasi, mulai dari *smart home* hingga *otomasi industri* [2]. Meskipun memberikan banyak manfaat, perkembangan IoT juga memperluas permukaan serangan yang rentan terhadap ancaman keamanan. Dengan lebih dari 40 miliar perangkat IoT yang diperkirakan aktif pada tahun 2030, memastikan keamanan jaringan IoT menjadi salah satu tantangan yang sangat mendesak untuk diatasi [3].

Salah satu ancaman signifikan terhadap jaringan IoT adalah serangan *Reconnaissance (Recon),* yang meskipun sering kali diabaikan, memiliki dampak besar terhadap keamanan siber. Serangan Recon dilakukan untuk mengumpulkan informasi penting dari jaringan, seperti alamat IP, konfigurasi perangkat, hingga celah keamanan yang dapat dieksploitasi lebih lanjut. Informasi yang berhasil dikumpulkan sering digunakan untuk merencanakan serangan yang lebih destruktif, seperti *Distributed Denial of Service* (DDoS), pengambilalihan perangkat, atau penyebaran malware [4]. Dalam lingkungan IoT, serangan Recon menjadi semakin sulit dideteksi karena karakteristik jaringan yang sangat dinamis dan volume data yang besar. Pendekatan tradisional yang berbasis aturan atau tanda tangan sering kali gagal mengenali serangan semacam ini, sehingga diperlukan solusi yang lebih adaptif dan berbasis pembelajaran [5].

Salah satu pendekatan yang dapat digunakan untuk menghadapi tantangan ini adalah metode deep learning, khususnya Convolutional Neural

Network (CNN). CNN merupakan algoritma deep learning yang dirancang untuk mengenali pola ko dalam data, sehingga sangat cocok untuk mendeteksi serangan jari sulit dikenali oleh metode konvensional. [6]. Dalam penelitian ini, NN diusulkan untuk mendeteksi serangan Recon dengan memanfaatkan data serangan yang telah diolah menjadi format numerik menggunakan *autoencoder*. *Autoencoder* digunakan untuk ekstraksi data, sekaligus menonjolkan fitur-fitur utama yang relevan untuk proses deteksi. Dengan memanfaatkan *autoencoder*, data mentah dari jaringan IoT dapat diubah menjadi representasi yang lebih padat dan informatif, yang kemudian diolah oleh CNN untuk mendeteksi pola-pola mencurigakan yang mengindikasikan aktivitas Recon [7].

Hasil dari penelitian ini diharapkan mampu menunjukkan bahwa model CNN yang dikembangkan dapat mendeteksi serangan Recon dengan tingkat akurasi yang tinggi. Melalui evaluasi menggunakan metrik seperti Confusion Matrix, Classification Report, ROC Curve, Precision-Recall Curve, Model Loss, Model Accuracy, dan Cross Validation, diharapkan model ini tidak hanya memiliki akurasi yang baik, tetapi juga mampu mendeteksi serangan dengan tingkat presisi dan sensitivitas yang optimal.

Selain itu, penelitian ini diharapkan memberikan kontribusi yang signifikan di bidang keamanan siber. Secara teoritis, penelitian ini dapat memperkaya literatur tentang penerapan deep learning, khususnya CNN dan autoencoder, dalam mendeteksi serangan jaringan. Secara praktis, penelitian ini diharapkan menawarkan solusi inovatif yang dapat digunakan untuk meningkatkan keamanan jaringan IoT, sekaligus memberikan dampak positif bagi pengembangan teknologi di masa depan.

1.2 Identifikasi Masalah

Berdasarkan latar belakang diatas, maka dapat diidentifikasi masalah sebagai berikut:

1) Jumlah perangkat IoT yang terus meningkat hingga diperkirakan mencapai 40 miliar pada tahun 2030 menimbulkan tantangan besar dalam menjaga keamanan jaringan.

2) Serangan Recon, yang bertujuan mengumpulkan informasi jaringan untuk digunakan da ngan lanjutan, menjadi ancaman signifikan namun sering diaba
3) Volume data IoT ar dan kompleks sulit ditangani oleh pendekatan konvensional, sementara sistem Deteksi Intrusi (IDS) tradisional berbasis tanda tangan *(signature-based)* dan anomali *(anomaly-based)* memiliki keterbatasan dalam mendeteksi serangan baru *(zero-day attacks)*.

1.3 Rumusan Masalah

Dari hasil identifikasi masalah diatas, maka dapat dirumuskan suatu permasalahan, yaitu:

1) Bagaimana cara mengembangkan model CNN untuk mendeteksi serangan Recon pada jaringan IoT secara efektif?
2) Bagaimana memanfaatkan data serangan Recon yang telah dikonversi menggunakan *autoencoder* untuk meningkatkan akurasi deteksi?
3) Bagaimana cara merancang dan mengimplementasikan model CNN yang mampu mengolah data IoT yang besar dan kompleks untuk mendeteksi serangan Recon secara akurat?

1.4 Batasan Masalah

Untuk membatasi ruang lingkup permasalahan yang diambil, maka perlu diberikan batasan-batasan masalah yang jelas agar nantinya tidak keluar dari pembahasan. Adapun Batasan masalah tersebut sebagai berikut:

1) Fokus penelitian hanya pada deteksi serangan Recon di jaringan IoT, tanpa mengklasifikasikan jenis serangan lainnya.
2) Penelitian ini tidak membahas langkah pencegahan serangan atau sistem mitigasi setelah serangan terdeteksi.
3) Penelitian ini menggunakan pendekatan *Deep Learning* (DL) berbasis CNN tanpa membandingkan langsung dengan metode *machine learning* lainnya.

1.5 Tujuan dan Manfaat

1.5.1 Tujuan Penelitia

1) Merancang dan gun model *Convolutional Neural Network* (CNN) untuk m serangan Recon pada jaringan IoT.
2) Menerapkan teknik *Autoencoder* untuk meningkatkan efisiensi dan efektivitas dalam proses ekstraksi fitur pada data.
3) Menghasilkan model deteksi serangan yang dapat diadaptasikan untuk sistem keamanan jaringan IoT dengan kebutuhan *real-time*

1.5.2 Manfaat Penelitian:

1) Memberikan solusi berbasis *Deep Learning* yang dapat diimplementasikan untuk meningkatkan keamanan jaringan IoT dari ancaman serangan siber, khususnya serangan Recon yang sering kali sulit terdeteksi oleh metode tradisional.
2) Mengoptimalkan penggunaan teknik *Autoencoder* dalam ekstraksi fitur dan pengurangan dimensi data serangan, yang memungkinkan model untuk lebih akurat dalam mengenali pola serangan dengan sumber daya komputasi yang lebih efisien.
3) Memberikan kontribusi dalam pengembangan sistem keamanan jaringan IoT yang lebih cerdas dan adaptif, serta dapat digunakan oleh organisasi atau penyedia layanan untuk melindungi infrastruktur IoT mereka dari risiko serangan yang semakin kompleks.

BAB II
KAJIAN PUSTAKA

2.1 Literatur

2.1.1 Analisis

Analisis, menurut Kamus Besar Bahasa Indonesia (KBBI), didefinisikan sebagai penyelidikan terhadap suatu peristiwa untuk memahami keadaan yang sebenarnya. Proses ini bertujuan untuk mengungkap fakta-fakta yang relevan, mengidentifikasi penyebab, serta memberikan gambaran yang lebih jelas mengenai situasi yang sedang diteliti. Dalam konteks penelitian, analisis menjadi langkah penting untuk memastikan bahwa semua informasi yang diperoleh dapat diolah dan digunakan sebagai dasar dalam pengambilan keputusan.

Analisis biasanya dilakukan pada tahap akhir suatu aktivitas sebagai bentuk evaluasi terhadap keseluruhan proses yang telah berlangsung. Proses ini bertujuan untuk mengidentifikasi masalah-masalah yang mungkin muncul selama pelaksanaan kegiatan, baik itu berupa hambatan, kesenjangan, atau faktor lain yang memengaruhi hasil yang diharapkan [8]. Dengan melakukan analisis, peneliti dapat memahami apa saja yang menjadi kekurangan, kelebihan, atau aspek yang perlu diperbaiki.

Hasil dari proses analisis juga berfungsi sebagai panduan penting dalam perencanaan kegiatan selanjutnya. Dengan mengetahui akar masalah dari kegiatan sebelumnya, langkah-langkah berikutnya diharapkan dapat dilakukan dengan lebih efektif dan efisien. Oleh karena itu, analisis tidak hanya sekadar mencari tahu apa yang salah, tetapi juga berupaya mengembangkan solusi dan strategi yang lebih baik untuk mencapai hasil optimal.

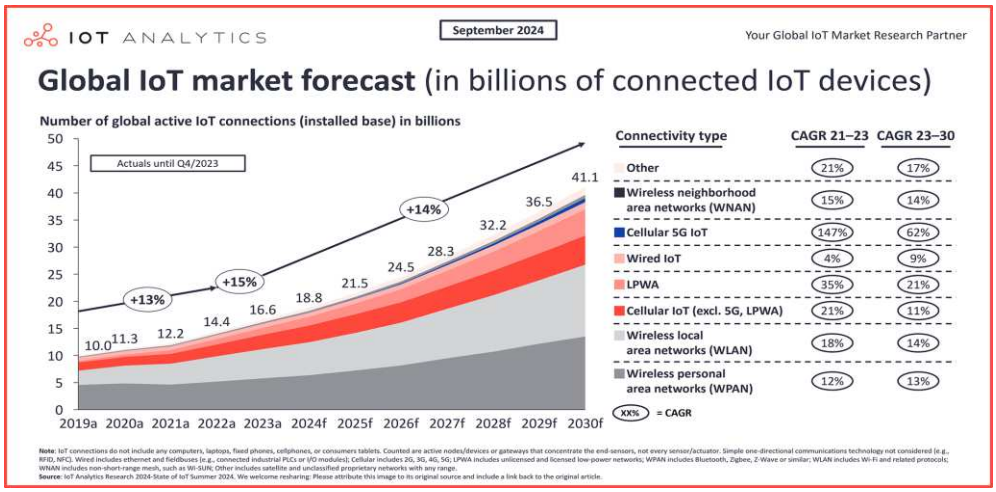
2.1.2 Internet of Things (IoT)

Internet of Things (IoT) adalah teknologi yang memungkinkan berbagai perangkat seperti komputer, ponsel pintar, tablet, televisi pintar, dan perangkat rumah tangga yang dilengkapi dengan sensor, aktuator, serta

perangkat lunak untuk saling terhubung melalui jaringan internet. IoT mendukung pertukaran da[illegible]ra *real-time* antara perangkat, menciptakan ekosistem yang cerdas da[illegible]erintegrasi [2].

Menurut laporan *Io*[illegible]*s* (2024), jumlah perangkat IoT secara global diperkirakan akan mencapai 40 miliar pada tahun 2030 sebagaimana ditunjukkan pada **Gambar 2.1** [3]. Angka ini menunjukkan pertumbuhan signifikan dalam konektivitas perangkat IoT di berbagai sektor. Namun, keberadaan perangkat IoT dalam jumlah besar juga membawa tantangan, terutama dalam hal keamanan. Perangkat IoT sering kali memiliki tingkat keamanan yang rendah, sehingga menjadi sasaran empuk bagi berbagai jenis serangan siber, termasuk serangan *Reconnaissance* (Recon).

Oleh karena itu, meningkatkan keamanan jaringan IoT menjadi prioritas utama dalam menghadapi ancaman tersebut. Sebagai elemen penting dari ekosistem digital modern, memahami ancaman terhadap IoT menjadi langkah awal dalam menciptakan jaringan yang lebih aman dan tangguh.



***Gambar 2.1**. Proyeksi jumlah perangkat IoT di seluruh dunia berdasarkan laporan IoT Analytics (2024) [3]*

2.1.3 Reconnaissance (Recon)

Reconnaissance (Recon) attack merupakan jenis serangan siber yang bertujuan untuk mengumpulkan informasi jaringan secara ilegal. Informasi yang dicari meliputi alamat IP, konfigurasi perangkat, serta celah keamanan yang dapat dimanfaatkan untuk melancarkan serangan lebih lanjut. Serangan

Recon sering kali menjadi langkah awal dalam proses serangan siber, karena informasi yang diperole[illegible]nakan untuk mendukung jenis serangan lainnya, seperti *Distribu[illegible]al of Service* (DDoS) atau penyebaran malware [4].

Reconnaissance dapat dibagi menjadi dua kategori:

1) *Active Reconnaissance*

   Dimana penyerang secara langsung berinteraksi dengan sistem target untuk mengumpulkan informasi, misalnya melalui *port scanning, ping sweep, network mapping*, atau *banner grabbing.*

2) *Passive Reconnaissance*

   Dimana pengumpulan informasi tanpa interaksi langsung dengan target, misalnya melalui pencarian di media sosial, DNS lookup, analisis metadata, atau pencarian di internet.

   Beberapa teknik yang digunakan dalam *reconnaissance* meliputi *OSINT, social engineering, footprinting, scanning, dumpster diving, physical reconnaissance*, *dan packet sniffing*. Alat yang sering digunakan adalah *Nmap, Wireshark, Shodan, Maltego, dan Recon-ng*.

2.1.4 Deep Learning

Deep Learning adalah cabang dari *Artificial Intelligence* (AI) dan *Machine Learning* (ML) yang menggunakan jaringan saraf tiruan serta algoritma yang meniru cara kerja otak manusia, memungkinkan pembelajaran dari data dalam jumlah besar [9].

Deep Learning mengidentifikasi pola yang kompleks secara otomatis, memungkinkan model untuk belajar dari data tanpa memerlukan pengkodean eksplisit untuk setiap aturan atau fitur. Dengan pendekatan ini, *Deep Learning* dapat memanfaatkan kekuatan data untuk menghasilkan prediksi dan klasifikasi yang sangat akurat, bahkan dalam situasi dengan pola yang sangat rumit atau tidak jelas [10].

Deep Learning bekerja dengan menggunakan arsitektur yang terdiri dari banyak lapisan *neuron* buatan, yang dikenal sebagai *deep neural networks*. Lapisan-lapisan ini bekerja secara hierarkis, di mana setiap lapisan

bertanggung jawab untuk mengekstraksi fitur tertentu dari data. Teknologi ini telah mengubah berbagai , termasuk pengenalan wajah, pengenalan suara, pengolahan bahasa LP), deteksi objek, dan bahkan kendaraan otonom. Keunggulan ini n *Deep Learning* sebagai teknologi utama dalam pengembangan solusi berbasis AI di dunia modern.

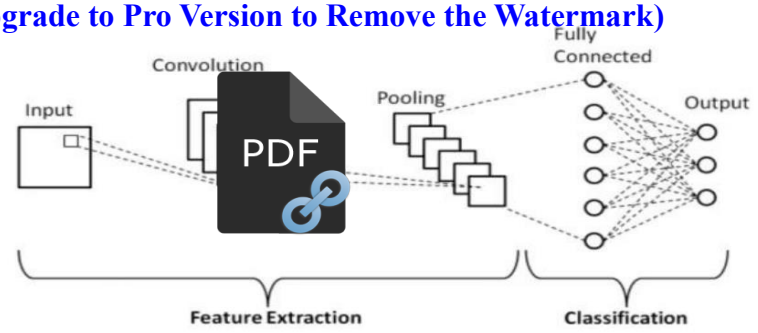
Dengan adanya kemampuan untuk menangani data yang tidak terstruktur seperti gambar, teks, dan audio, *Deep Learning* memanfaatkan algoritma seperti *convolutional neural networks* (CNNs), *recurrent neural networks* (RNNs), dan *transformers*. Semua ini dimungkinkan berkat peningkatan daya komputasi, ketersediaan data dalam jumlah besar, serta algoritma optimasi yang canggih.

2.1.5 Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) merupakan salah satu jenis arsitektur *Deep Learning* yang dirancang untuk mengenali pola pada data berbentuk grid, seperti gambar [11]. Namun, penerapan CNN kini telah meluas ke berbagai bidang, termasuk pengolahan data numerik, analisis teks, dan keamanan jaringan [10].

Secara garis besar, CNN memiliki prinsip kerja yang mirip dengan *neural network* pada umumnya, yang terdiri dari *neuron* dengan *weight*, *bias*, dan *activation function*. Operasi konvolusi pada CNN dilakukan dengan menggeser kernel konvolusi (filter) berukuran tertentu melintasi data, menghasilkan representasi fitur baru dari hasil perkalian antara filter dan sebagian data tersebut.

Seperti yang ditunjukkan pada **Gambar 2.2**, arsitektur CNN secara umum terbagi menjadi dua bagian utama, yaitu *Feature Extraction Layer* dan *Classification Layer*.

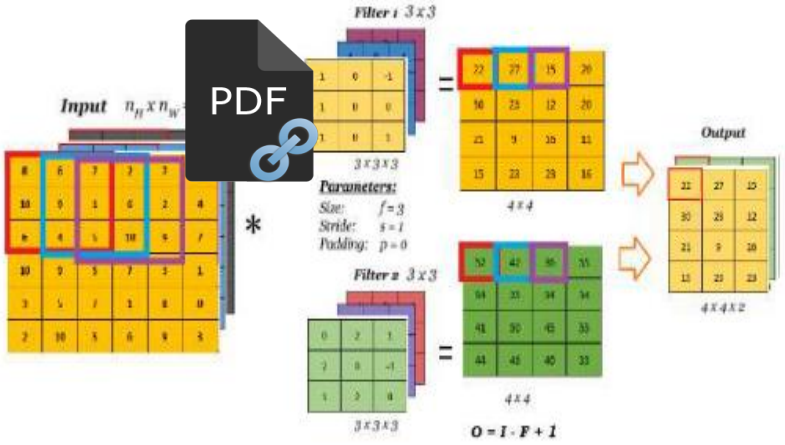


***Gambar 2.2**. Arsitektur CNN* [12]

1) *Convolutional Layer*

Convolutional Layer berfungsi sebagai lapisan ekstraksi fitur *(feature extraction layer)* yang terdiri dari beberapa sub-lapisan. Setiap sub-lapisan memiliki *neuron* yang saling terhubung dengan lapisan sebelumnya. Proses konvolusi dilakukan menggunakan filter dengan dimensi tertentu, mencakup tinggi, lebar, dan kedalaman. Filter tersebut diinisialisasi dengan nilai awal yang akan terus diperbarui selama proses pelatihan. Operasi konvolusi dilakukan dengan menggeser filter melintasi matriks input, disertai proses perkalian dan penjumlahan, menghasilkan matriks baru yang dikenal sebagai *feature map*.

Umumnya, beberapa jenis filter atau kernel digunakan secara bersamaan, seperti filter 1, filter 2, dan seterusnya, untuk menangkap berbagai pola pada data. Dimensi keluaran (output) matriks, baik untuk baris maupun kolom, dapat dihitung menggunakan rumus $O = I - F + 1$ [13].



***Gambar 2.3**. Proses konvolusi* [13]

Keterangan :

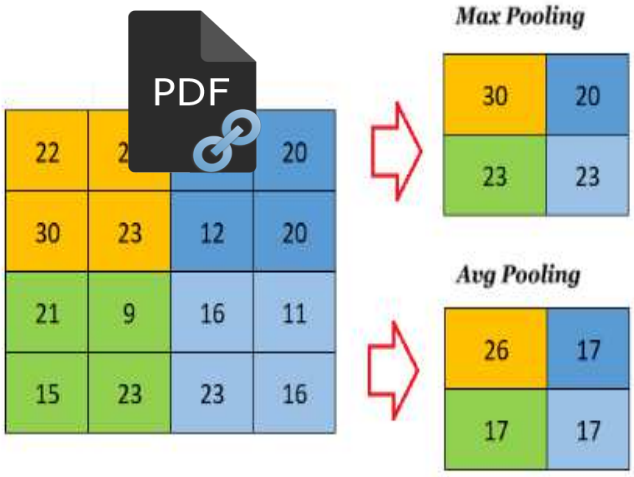
O: Ukuran Output

I: Ukuran Input

F: Ukuran Filter

2) Pooling Layer

Pooling layer berfungsi untuk mengurangi dimensi representasi yang dihasilkan selama proses ekstraksi fitur, sehingga dapat menurunkan kompleksitas komputasi model. Dengan cara ini, *pooling layer* menjadi sangat efektif dalam menyederhanakan jumlah dimensi data sambil tetap mempertahankan fitur-fitur penting. Dua metode pooling yang umum digunakan adalah *Max Pooling*, yang memilih nilai maksimum, dan *Average Pooling*, yang menghitung nilai rata-rata .



***Gambar 2.4**. Pooling Layer* [13]

3) Fungsi Aktivasi

Fungsi aktivasi adalah komponen yang menentukan apakah output dari suatu *neuron* akan diaktifkan dan diteruskan ke *neuron* berikutnya. *Neuron* akan menghasilkan *output* jika nilainya melewati ambang batas tertentu. Beberapa fungsi aktivasi yang sering digunakan meliputi *Sigmoid, Softmax*, dan *Rectified Linear Unit (ReLU).*

4) Optimasi

Optimasi berperan penting dalam menentukan bobot optimal model untuk meningkatkan akurasi prediksi. Selama proses pelatihan, bobot atau parameter akan terus diperbarui agar model dapat membuat prediksi yang lebih akurat. Karena itu, optimasi menjadi komponen penting dalam *neural network*. Beberapa algoritma optimasi yang sering digunakan adalah *Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp), dan Adaptive Momentum Estimation (Adam).*

5) Batch Normalization

Batch Normalization adalah teknik regularisasi yang bertujuan untuk menormalkan aktivasi input sebelum diteruskan ke lapisan berikutnya dalam jaringan. Proses ini melibatkan pengurangan rata-rata

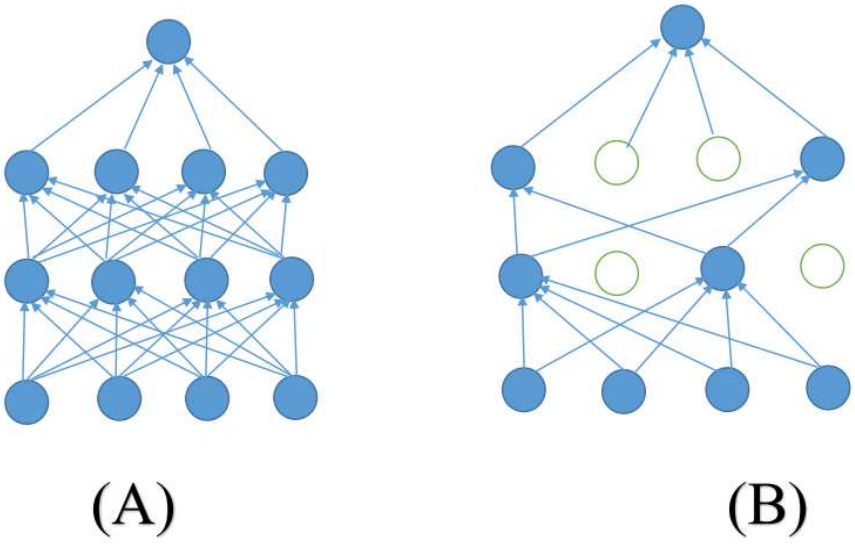
batch dari setiap aktivasi dan pembagian dengan standar deviasinya. Dengan memasukk*h Normalization* ke dalam model, pelatihan dapat berlangsung l secara signifikan.

6) Flatten

Pada tahap *flatten*, matriks berukuran n×n yang dihasilkan dari *pooling layer* akan diubah menjadi matriks vektor satu dimensi berukuran n×1. Vektor ini kemudian digunakan sebagai *input* pada tahap berikutnya, yaitu klasifikasi.

7) Dropout

Dropout adalah teknik regularisasi yang secara acak "menonaktifkan" sejumlah *neuron* selama proses pelatihan. Teknik ini bertujuan untuk mengurangi risiko *overfitting* dan mempercepat pelatihan, terutama pada jaringan yang memiliki banyak lapisan atau *neuron*.



***Gambar 2.5**. jaringan saraf standar (A) dan jaringan setelah menerapkan dropout (B)* [14]

8) Fully-Connected Layer (Dense)

Lapisan *Fully-Connected* adalah lapisan di mana setiap *neuron* dari lapisan sebelumnya terhubung sepenuhnya ke *neuron* pada lapisan berikutnya, serupa dengan prinsip jaringan saraf tiruan. Sebelum

dihubungkan ke lapisan *Fully-Connected*, semua aktivitas dari lapisan sebelumnya harus d menjadi bentuk vektor satu dimensi. Fungsi utama lapisan *Fu cted* adalah untuk melakukan proses klasifikasi.

9) Evaluasi Performa Model

Kualitas sebuah model ditentukan berdasarkan kemampuannya dalam memprediksi data secara akurat. Beberapa metrik evaluasi performa yang umum digunakan meliputi nilai loss, akurasi, dan f1-score yang dihitung berdasarkan confusion matrix.

a. Focal Loss

Focal Loss digunakan untuk menangani masalah ketidakseimbangan kelas yang ekstrem dalam pelatihan model, seperti dalam kasus rasio kelas 1:1000. Rumus untuk menghitung *Focal Loss* adalah sebagai berikut:

$$FL(p_t) = \alpha(1 - p_t)^{\gamma} \log(p_t)$$

b. Confusion Matrix

Confusion matrix adalah matriks yang menunjukkan perbandingan antara hasil prediksi model dengan data sebenarnya.

		Actual values	
		+	-
Predicted values	+	True positive(TP)	False positive(FP)
	-	False negative(FN)	True negative (TN)

Gambar 2.6. *Confusion Matrix* [15]

TN *(True Negative)* adalah jumlah data negatif yang berhasil diprediksi dengan b N *(False Negative)* adalah jumlah data positif yang keliru diprediksi negatif, TP *(True Positive)* adalah jumlah data positif yang d dengan benar, dan FP *(False Positive)* adalah jumlah data negatif yang salah diprediksi sebagai positif.

Nilai akurasi dapat dihitung menggunakan rumus berikut:

$$Akurasi = \frac{Jumlah\ prediksi\ benar}{Total\ data} = \frac{TN + TP}{TN + TP + FN + FP}$$

2.1.6 Autoencoder

Autoencoder adalah salah satu teknik pembelajaran mesin tanpa pengawasan (*unsupervised learning*) yang berbasis jaringan saraf tiruan. Algoritma ini dirancang untuk merekonstruksi *output* yang mendekati *input* aslinya. Secara struktural, *autoencoder* terdiri dari tiga lapisan utama, yaitu lapisan *input*, lapisan *output* (dengan jumlah dimensi yang sama dengan *input*), dan lapisan tersembunyi *(hidden layer)* yang biasanya memiliki dimensi lebih kecil daripada lapisan *input*. Struktur ini memungkinkan *autoencoder* untuk menyederhanakan data menjadi representasi berdimensi rendah tanpa kehilangan informasi penting. Keunggulan utama *autoencoder* terletak pada kemampuannya untuk menemukan pola dalam data dengan lebih efektif dibandingkan metode tradisional seperti *Principal Component Analysis* (PCA). Hal ini dimungkinkan melalui transformasi *non-linear* yang dilakukan selama proses *encoding* dan *decoding*. Pada tahap *encoding*, data *input* diubah menjadi representasi berdimensi rendah, sedangkan tahap *decoding* merekonstruksi data ke bentuk semula. Proses ini menggunakan algoritma *backpropagation* dengan target *output* yang sama seperti *input*. *Autoencoder* juga mampu menangkap hubungan *non-linear* yang kompleks dalam data, sehingga sangat cocok untuk mengolah dataset yang rumit. Lapisan tersembunyi *(encoded layer)* yang dihasilkan selama proses ini sering dimanfaatkan sebagai fitur utama untuk keperluan klasifikasi [7].

Autoencoder digunakan untuk ekstraksi data dengan cara menonjolkan fitur-fitur utama yang rele merepresentasikan data dalam bentuk yang lebih padat. Proses ekst bertujuan untuk meningkatkan efisiensi pemrosesan data sekaligu tahankan informasi penting yang relevan. Dengan merepresentasikan data secara lebih sederhana, performa sistem deteksi atau analisis dapat ditingkatkan [16].

2.1.7 Python

Python adalah bahasa pemrograman tingkat tinggi yang pertama kali dikembangkan oleh Guido van Rossum pada akhir 1980-an dan resmi diluncurkan pada tahun 1991. Bahasa ini dikenal karena sintaksnya yang sederhana dan mudah dipahami, sehingga sangat populer di kalangan pemula maupun pengembang berpengalaman. Selain itu, Python bersifat dinamis, memungkinkan pengembang untuk menulis kode dengan cepat sekaligus memberikan fleksibilitas tinggi dalam memodifikasi skrip.

Python sering digunakan dalam berbagai bidang, seperti pengembangan perangkat lunak, kecerdasan buatan (AI), pengembangan web, *machine learning*, dan analisis data. Keberadaan pustaka-pustaka unggulan seperti *NumPy* untuk komputasi numerik dan *Pandas* untuk analisis data, menjadikan Python alat yang efisien untuk menyelesaikan berbagai tugas dengan cepat dan mudah [17].

Salah satu kekuatan Python adalah kemampuannya dalam menghasilkan visualisasi data. Dengan bantuan modul bawaan seperti *Matplotlib* dan *NumPy*, Python mampu menghasilkan grafik dua variabel, seperti *scatter plot, line chart, bar chart*, hingga *pie chart*. *Matplotlib* juga mendukung berbagai modifikasi visual, seperti penambahan judul dan label. Sementara itu, *NumPy* menjadi alat penting dalam analisis data dan pemodelan matematika. Modul ini mempermudah operasi numerik berbasis array, mencakup operasi aritmatika, trigonometri, fungsi matematika, hingga analisis statistik. Kombinasi *Matplotlib* dan *NumPy* membuat pemrosesan serta visualisasi data numerik menjadi lebih cepat dan efisien dalam lingkungan Python [18].

Selain pustaka-pustaka tersebut, Python mendukung berbagai paradigma pemrograman, seperti p[illegible]man fungsional, berorientasi objek, dan prosedural. Bahasa ini ju[illegible]ki fitur manajemen memori otomatis dan mendukung *multithreadi*[illegible] menjadikannya lebih fleksibel untuk berbagai jenis aplikasi. Popularitas Python juga didukung oleh komunitas yang sangat aktif, menyediakan berbagai tutorial, dokumentasi, dan pustaka tambahan yang terus memperkaya ekosistem Python. Hal inilah yang membuat Python tetap menjadi pilihan utama bagi pengembang dan peneliti di berbagai bidang.

2.2 Penelitian Relevan

Penelitian relavan terdahulu yang dilakukan oleh para peneliti sebelumnya dapat dilihat pada table 1.

***Tabel 2.1**. Penelitian Relavan*

No	Peneliti	Judul	Metode	Hasil
1.	Alani, Mohammed M [19]	Detection of Reconnaissance Attacks on IoT Devices Using Deep Neural Networks BT - Advances in Nature-Inspired Cyber Security and Resilience	Deep Neural Network (DNN)	Akurasi 98%
2.	Mohammed M. Alani, Ernesto Damiani [20].	XRecon: An Explainbale IoT Reconnaissance Attack Detection System Based on Ensemble Learning	Ensemble Learning	Akurasi yang didapatkan rata-rata sebesar 99,57%. Dimana Akurasi model: 99,57% pada dataset IoT-ID dan 99,51% pada dataset TON_IoT.

3.	Hung Nguyen Viet, Linh Le Thi Trang, Quan Nguyen Van , Shone Nathan [21].	Using deep learning model for network scann tion	Naive Bayes, KNN, SVM, Decision Tree, Random Forest, MLP With Feature Extraction, DBN	Naive bayes akurasi : 93.89, KNN akurasi : 91.73, SVM akurasi : 98.11, Decision Tree akurasi : 99.50, Random Forest akurasi : 85.06, MLP With Feature Extraction akurasi : 99.44, DBN akurasi : 99.64
4.	Citra Lestari Mindara, Arief Zulianto, Hadi Prasetyo Utomo Tuti Hatati, Gema Parasti Mindara [22].	Deteksi Intrusi Untuk Klasifikasi Serangan Jaringan Dengan Penerapan Algoritma Convolutional Neural Network.	CNN, Random Forest, Random Tree, Naive Beyes dan J48.	Model 1D-CNN berhasil mencapai akurasi sempurna sebesar 100% dalam mengklasifikasikan kelas Benign, DDoS, dan **Portscan**, mengungguli kinerja model lain seperti Random Forest, Random Tree, Naive Bayes, dan J48. Namun, untuk kelas Brute Force dan Infiltration, hasilnya sedikit lebih rendah dibandingkan model-model tersebut, tetapi tetap menunjukkan kinerja yang baik secara keseluruhan.

5.	Deris Stiawan, Susanto, Abdi Bimantara, Mohd Yazid Idris, and Rahmat Budiarto [23].	IoT Botnet Attack Detection Using Deep Autoencoder And Neural Networks.	Autoencoder, Artificial Neural Network (ANN).	Penelitian ini berhasil mengembangkan sistem deteksi serangan IoT botnet berbasis deep Autoencoder untuk reduksi dimensi dan ANN untuk klasifikasi, dengan hasil kinerja yang sangat baik yaitu akurasi mencapai 99.72%, precision 99,82%, sensitivity 99.82%, specificity 99.31% dan f1-score 99.82%.

2.3 Kerangka Berpikir

Kerangka berpikir merupakan dasar untuk menentukan alur dari suatu penelitian. Dengan adanya kerangka berfikir, maka penelitian dapat tersusun dengan rapi dan mudah untuk dipahami. Adapun kerangka berpikir penelitian ini dapat dilihat pada **Gambar 2.7**.

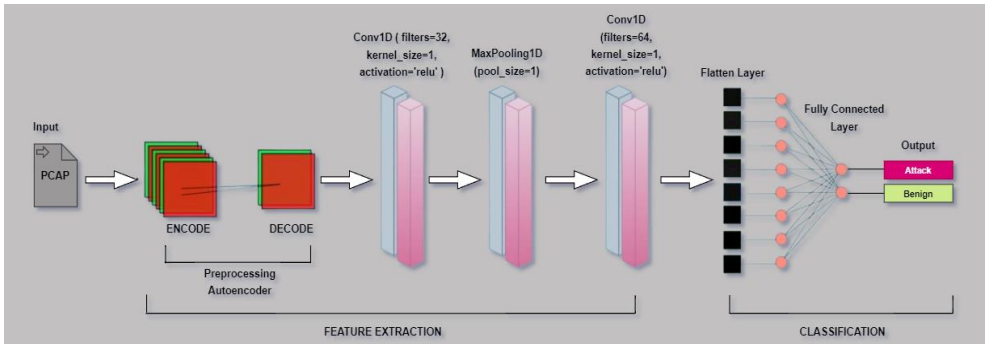
***Gambar 2.7**. Kerangka Berpikir*

BAB III
METODOLOGI PENELITIAN

3.1 Analisa Sistem

Analisa sistem dilakukan untuk memahami bagaimana sistem yang akan dirancang dapat bekerja sesuai dengan kebutuhan. Dalam penelitian ini, sistem yang dirancang bertujuan untuk mendeteksi serangan Recon pada jaringan IoT menggunakan model Convolutional Neural Network (CNN). Langkah-langkah analisa sistem meliputi:

1) Mengidentifikasi data masukan berupa file .pcap yang berisi data lalu lintas jaringan.
2) Melakukan preprocessing data untuk mengonversi file .pcap menjadi format numerik dan CSV agar dapat digunakan sebagai input model.
3) Mendesain arsitektur model CNN yang mampu mendeteksi pola serangan dalam data jaringan.
4) Menganalisis keluaran model untuk menentukan apakah serangan terdeteksi atau tidak.



***Gambar 3.8**. Diagram Alur Penelitian*

3.1.1 Analisa Sistem yang Berjalan

Sistem yang berjalan saat ini pada jaringan IoT sering kali bergantung pada metode tradisional, seperti penggunaan aturan berbasis signature dalam sistem Intrusion Detection System (IDS). Pendekatan ini memiliki kelemahan:

1) Tidak mampu mendeteksi serangan baru atau varian serangan yang belum terdefinisi
2) Membutuhkan p an dan pembaruan aturan secara manual.
3) Keterbatasan per t lalu lintas jaringan sangat besar.

3.1.2 Alternatif Pemecahan Masalah

Untuk mengatasi kelemahan ini, penelitian ini mengusulkan penggunaan model CNN yang mampu mendeteksi pola serangan secara otomatis berdasarkan data lalu lintas jaringan.

3.1.3 Metode Analisa

Penelitian ini berfokus pada analisis serangan Recon pada jaringan *Internet of Things* (IoT) dengan menggunakan kombinasi *Autoencoder* dan *Convolutional Neural Network* (CNN). Serangan Recon *(Reconnaissance)* adalah langkah awal yang dilakukan oleh penyerang untuk mengumpulkan informasi tentang jaringan sebelum melakukan eksploitasi lebih lanjut. Oleh karena itu, mendeteksi serangan ini secara dini sangat penting untuk meningkatkan keamanan jaringan IoT.

Metode yang digunakan dalam penelitian ini terdiri dari dua tahap utama: *preprocessing* data menggunakan *Autoencoder* dan analisis pola serangan menggunakan CNN. Berikut adalah penjelasan rinci mengenai kedua metode tersebut.

1) Autoencoder untuk Preprocessing Data

   Autoencoder adalah algoritma *deep learning* yang digunakan untuk mereduksi dimensi data tanpa kehilangan informasi penting. Dalam penelitian ini, *Autoencoder* berfungsi untuk mengolah data serangan Recon yang kompleks menjadi representasi fitur yang lebih sederhana dan mengkonversi data Pcap menjadi CSV, sehingga mempermudah analisis lebih lanjut oleh CNN.

   Arsitektur *Autoencoder* terdiri dari dua komponen utama:

   1) *Encoder*: Berfungsi untuk mereduksi dimensi data input (X) menjadi representasi tersembunyi (Y).

2) *Decoder*: Berfungsi untuk merekonstruksi data dari representasi tersembunyi (njadi data yang mendekati data asli (X')

Rumus Autoencode ebagai berikut:

1) Fungsi encodi $Y = f(X) = s(WX + bx)$
2) Fungsi decoding : $X' = g(Y) = s(W'Y + by)$

Keterangan :

X: Data input.

Y: Representasi tersembunyi (encoded representation).

X': Data hasil rekonstruksi.

W': Bobot encoding dan decoding.

bx, by: Bias encoding dan decoding.

s: Fungsi aktivasi, seperti sigmoid atau ReLU.

Proses pelatihan *Autoencoder* bertujuan untuk meminimalkan error rekonstruksi antara X dan X' menggunakan fungsi loss *Mean Squared Error* (MSE):

$$\mathcal{L}_{Autoencode} = \frac{1}{n}\sum_{i=1}^{n}(X_i - X_i')^2$$

Dengan menggunakan *Autoencoder*, data mentah serangan Recon yang kompleks diolah menjadi representasi yang lebih sederhana dan informatif. Representasi ini kemudian digunakan sebagai input untuk tahap analisis selanjutnya menggunakan CNN.

2) Convolutional Neural Network (CNN) untuk Analisis Serangan

Setelah data diproses oleh *Autoencoder*, representasi fitur yang dihasilkan digunakan sebagai input untuk *Convolutional Neural Network* (CNN). CNN dirancang untuk mengenali pola kompleks dari data yang telah dipreproses dan digunakan untuk mendeteksi apakah data tersebut mengindikasikan adanya serangan Recon atau tidak.

Komponen utama CNN meliputi:

1) Lapisan Konvolusi (Convolution Layer)

Mengeks itur lokal dari data input menggunakan kernel (filter).

Operasi konv atakan sebagai:

$$z_{i,j} = \sum_{m=1}^{M} \sum_{n=1}^{N} X_{i+m-1,j+n-1} \cdot K_{m,n} + b$$

Keterangan :

$z_{i,j}$: Nilai feature map pada posisi(i, j).

X: Data input.

K: Kernel dengan ukuran $M \times N$.

b: Bias.

Setelah operasi konvolusi, fungsi aktivasi seperti ReLU ($s(x) = max(0, x)$) digunakan untuk menambahkan non-linearitas.

2) Lapisan Pooling

Mengurangi dimensi data *(down-sampling)* sambil mempertahankan fitur penting. Teknik pooling yang digunakan adalah *Max Pooling*, dengan rumus:

$$z_{i,j} = max(X_{m,n}), \qquad m, n \in area\ pooling$$

3) Lapisan Fully Connected

Setelah beberapa lapisan konvolusi dan pooling, data diratakan menjadi vektor 1 dimensi dan diteruskan ke lapisan *fully connected* untuk menghasilkan output prediksi.

Tugas deteksi serangan Recon adalah klasifikasi biner (serangan atau tidak). Oleh karena itu, model CNN dilatih menggunakan fungsi loss Binary Cross-Entropy (BCE):

$$\mathcal{L}_{CNN} = -\frac{1}{m}\sum_{i=1}^{m}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)]$$

Keterangan :

y_i: Label se a.

$\hat{y}_i$: Probabili si.

m: Jumlah d

Pengujian performa model dilakukan menggunakan berbagai alat evaluasi untuk menilai kemampuan model dalam mendeteksi serangan Recon pada jaringan IoT. Adapun metode pengujian yang digunakan meliputi:

1) Confusion Matrix

Confusion Matrix digunakan untuk memvisualisasikan hasil prediksi model terhadap data uji dengan membandingkan prediksi yang benar dan salah. Matriks ini terdiri dari empat elemen utama:

a) *True Positive* (TP): Jumlah data serangan yang diprediksi sebagai serangan secara benar.
b) *True Negative* (TN): Jumlah data non-serangan yang diprediksi sebagai non-serangan secara benar.
c) *False Positive* (FP): Jumlah data non-serangan yang salah diprediksi sebagai serangan.
d) *False Negative* (FN): Jumlah data serangan yang salah diprediksi sebagai non-serangan.

2) Classification Report

Classification Report memberikan analisis rinci performa model untuk setiap kelas dalam dataset. Laporan ini mencakup metrik seperti precision, recall, F1-score, dan support (jumlah sampel per kelas). Analisis ini berguna untuk memahami bagaimana model menangani masing-masing kelas, baik serangan Recon maupun kondisi non-serangan, serta memberikan gambaran lebih lengkap tentang kinerja model dalam berbagai aspek.

Beberapa metrik penting yang digunakan adalah sebagai berikut:

a) Akurasi: Mengukur proporsi prediksi yang benar terhadap total dat[illegible]s: $\frac{TP+TN}{TP+TN+FP+FN}$

b) Presisi: [illegible]etepatan prediksi positif. Rumus: $\frac{TP}{TP+FP}$

c) Recall: Mengukur sejauh mana model dapat menangkap data positif. Rumus: $\frac{TP}{TP+FN}$

d) Specificity: Menilai ketepatan prediksi negatif. Rumus: $\frac{TN}{TN+FP}$

e) F1-Score: Harmonic mean dari presisi dan recall. Rumus: $2 \times \frac{Precision \times Recall}{Precision + Recall}$

3) ROC Curve

Receiver Operating Characteristic (ROC) Curve digunakan untuk mengevaluasi kemampuan model dalam membedakan antara kelas positif dan negatif pada berbagai nilai ambang (threshold). Kurva ROC menggambarkan hubungan antara *True Positive Rate* (TPR) dan *False Positive Rate* (FPR), yang dihitung menggunakan rumus berikut:

a) True Positive Rate (TPR):

$$TPR = \frac{TP}{TP + FN}$$

b) False Positive Rate (FPR):

$$FPR = \frac{FP}{FP + TN}$$

Pada ROC Curve, sumbu Y merepresentasikan TPR, sementara sumbu X merepresentasikan FPR. Kurva yang lebih dekat dengan sudut kiri atas grafik menunjukkan performa model yang lebih baik.

Area Under the Curve (AUC) adalah skor yang dihitung dari luas di bawah ROC Curve. AUC memberikan ukuran keseluruhan performa model, dengan nilai berkisar antara 0 dan 1. Nilai AUC

yang mendekati 1 menunjukkan bahwa model memiliki performa yang sangat b[illegible]m mendeteksi kelas target dengan akurasi tinggi.

4) Precision-Rec[illegible]

Precision-Recall Curve mengevaluasi performa model dalam menangani ketidakseimbangan kelas, yang umum terjadi dalam dataset serangan jaringan. Kurva ini menggambarkan hubungan antara *precision* (ketepatan prediksi positif) dan *recall* (cakupan prediksi positif) pada berbagai *threshold*. *Precision* dan *recall* dihitung menggunakan rumus yang sama seperti pada *Classification Report*. Evaluasi ini penting untuk memastikan bahwa model tetap memiliki performa yang baik meskipun data antar kelas tidak seimbang. Kurva ini sangat berguna terutama ketika dataset memiliki lebih banyak data negatif dibandingkan data positif. *Precision-Recall Curve* dibuat dengan memplot *precision* pada sumbu y dan *recall* pada sumbu x untuk setiap nilai *threshold* yang diambil dari probabilitas prediksi model

5) Model Loss Plot

Model Loss Plot digunakan untuk memantau nilai *error* yang dihasilkan selama proses pelatihan dan validasi. Grafik ini membantu mengidentifikasi apakah model mengalami *overfitting* (loss validasi lebih tinggi dibandingkan loss pelatihan) atau *underfitting* (loss pelatihan dan validasi sama-sama tinggi). Dengan memplot nilai loss setiap epoch, peneliti dapat mengamati tren pembelajaran model. Penurunan nilai loss yang konsisten selama pelatihan menunjukkan bahwa model sedang belajar dengan baik dari data. Nilai loss dihitung berdasarkan fungsi loss yang digunakan selama pelatihan, seperti *categorical cross-entropy* untuk klasifikasi, dengan rumus: di mana adalah label aktual, adalah probabilitas yang diprediksi oleh model, adalah jumlah sampel, dan adalah jumlah kelas.

6) Model Accuracy Plot

Model A... Plot memberikan gambaran tentang performa mo... memprediksi data selama pelatihan dan validasi. Graf... nunjukkan seberapa baik model mampu menggeneralisasi dari data pelatihan ke data validasi. Performa yang konsisten antara akurasi pelatihan dan validasi menjadi indikator penting dari model yang stabil dan tidak terlalu spesifik pada data pelatihan. *Model Accuracy Plot* membantu dalam mengidentifikasi potensi *overfitting* atau *underfitting* berdasarkan pola perubahan akurasi selama pelatihan

7) Cross Validation

Cross Validation adalah teknik evaluasi model yang digunakan untuk mengukur performa dan generalisasi model pada data yang berbeda dari data pelatihan. Teknik ini membagi dataset menjadi beberapa subset atau folds, di mana proses pelatihan dilakukan pada beberapa subset sementara satu subset digunakan untuk pengujian. Proses ini diulangi hingga setiap subset menjadi bagian dari data pengujian, dan hasil akhirnya diambil rata-rata untuk mendapatkan metrik performa yang lebih akurat.

3.2 Teknik Pemilihan Informan

Pada subbab ini dijelaskan bagaimana pemilihan informan dilakukan secara sistematis untuk mendukung penelitian yang dilakukan. Teknik yang digunakan meliputi:

1) Populasi

Populasi dalam penelitian ini mencakup seluruh data atau individu yang memenuhi kriteria tertentu yang relevan dengan tujuan penelitian. Dalam konteks penelitian ini, populasi terdiri dari data serangan jaringan yang diambil dari file pcap yang telah dikonversi menjadi format numerik.

2) Sample

Sampel merup agian dari populasi yang dipilih untuk dianalisis lebih lanj an sampel dilakukan secara representatif agar hasil peneliti menggambarkan karakteristik populasi secara keseluruhan. Dalam penelitian ini, sampel yang digunakan adalah data serangan dan data normal yang telah diproses.

3) Teknik Sampling

Teknik sampling yang digunakan dalam penelitian ini adalah purposive sampling. Teknik ini dipilih karena penelitian hanya membutuhkan data yang memenuhi kriteria tertentu, seperti data yang telah di-encode menjadi numerik dan memiliki label serangan atau normal.

3.2.1 Teknik Pengumpulan Data

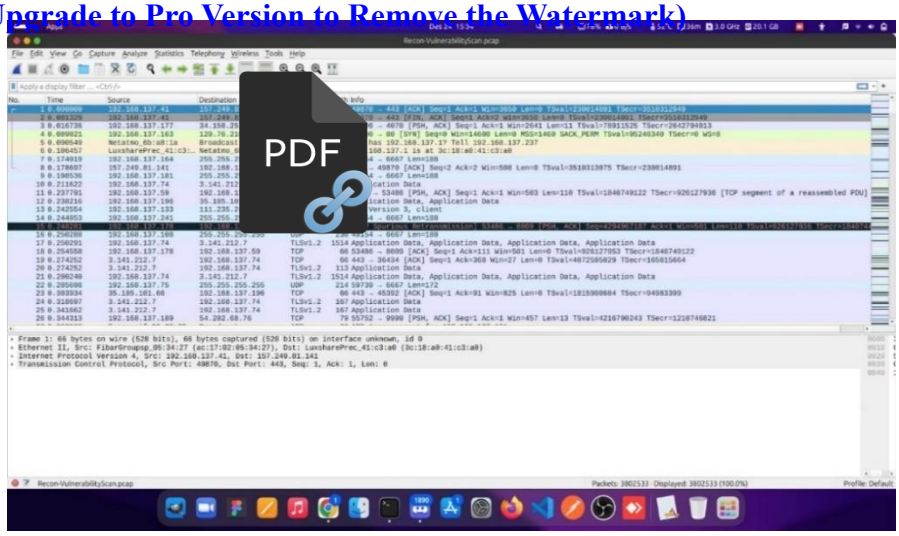
Dalam penelitian ini, data yang digunakan sepenuhnya berasal dari dataset publik yang tersedia di UNB Repository, yang dapat diakses melalui http://205.174.165.80/IOTDataset/CIC_IOT_Dataset2023/Dataset/.

Dataset terdiri dari dua kategori utama:

1) Data Serangan Recon

Dataset ini mencakup jenis serangan *seperti OS Scan, Ping Sweep, Host Discovery, Port Scan*, dan *Vulnerability Scan*. Data ini didapat dalam bentuk file PCAP. Dengan rincian sebagai berikut :

a. Os Scan : 992241 data
b. Ping Sweep : 22943 data
c. Host Discovery : 1371112 data
d. Port Scan : 831856 data
e. Vulnerability Scan : 3802533 data
f. Semua data Serangan : 7020685 data

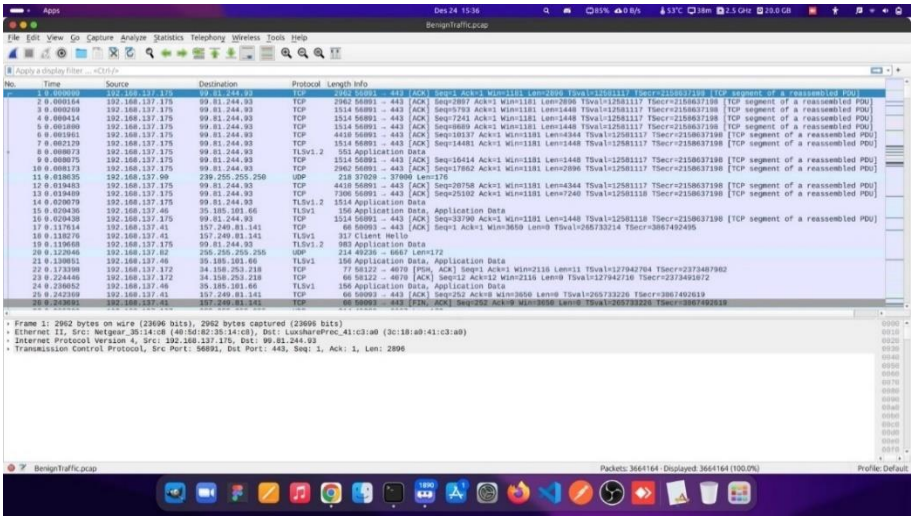


***Gambar 3.9**. Contoh Tampilan Data Serangan dalam Format PCAP yang Dibuka dengan Wireshark*

2) Data Normal

Dataset yang berisi lalu lintas jaringan normal tanpa serangan. Terdapat beberapa dataset, diantaranya:

a. Benign Traffic : 3664164 data
b. Benign Traffic1 : 2988642 data
c. Benign Traffic2 : 3138002 data
d. Benign Traffic3 : 1311897 data
e. Semua data Benign : 11102705 data



***Gambar 3.10**. Contoh Tampilan Data Serangan dalam Format PCAP yang Dibuka dengan Wireshark*

3.2.2 Teknik Analisa Data

Proses pengolahan d g baik sangat penting untuk meningkatkan akurasi dan performa mo mendeteksi serangan jaringan. Berikut ini adalah langkah-langkah y ukan peneliti dalam mengolah data.

1) Pembacaan Data PCAP

   Data lalu lintas jaringan yang berupa file PCAP dibaca menggunakan pustaka *Scapy*, yang memungkinkan ekstraksi informasi penting seperti timestamp, source IP, destination IP, protocol, dan packet length

2) Preprocessing Data Menggunakan Autoencoder

   Data yang telah diekstraksi selanjutnya diproses menggunakan *Autoencoder*, yang berfungsi untuk mengurangi dimensi dan mengekstrak fitur yang lebih relevan. *Autoencoder* membantu dalam menemukan representasi yang lebih efisien dari data asli dengan mengurangi informasi yang tidak penting dan meningkatkan kualitas fitur yang berguna untuk deteksi serangan jaringan. Teknik ini memungkinkan model untuk fokus pada fitur utama dan meningkatkan akurasi deteksi dengan mengurangi noise dan redundansi dalam data.

3) Penyimpanan Data dalam Format CSV

   Setelah proses ekstraksi dan pengurangan dimensi dengan *Autoencoder*, data yang telah diproses disimpan dalam format CSV. Format ini mempermudah pengelolaan data untuk tahap-tahap selanjutnya dalam pelatihan model deteksi serangan.

4) Pemberian Label

   Setelah data disimpan dalam file CSV, langkah selanjutnya adalah pemberian label untuk membedakan data benign dan data serangan. Data benign, yang merujuk pada lalu lintas jaringan normal, diberi label 0, sedangkan data serangan, yang merujuk pada lalu lintas jaringan yang mencurigakan, diberi label 1. Pemberian label ini sangat penting karena model deteksi serangan jaringan membutuhkan data yang telah dilabeli untuk dapat membedakan antara lalu lintas jaringan yang aman

dan yang berbahaya. Proses pemberian label ini memungkinkan model untuk belajar pola-p ngan yang ada dan mengklasifikasikan data secara akurat selam n.

3.3 Tempat dan Waktu Peneli

3.3.1 Tempat Penelitian

Penelitian ini dilakukan di Laboratorium Universitas Bina Insan dengan menggunakan perangkat komputer yang disediakan oleh pihak universitas. Seluruh proses penelitian, mulai dari pengolahan data, pembuatan model, hingga analisis hasil, dilakukan secara digital dengan memanfaatkan dataset publik yang tersedia secara daring.

3.3.2 Waktu Penelitian

Penelitian ini dilakukan dari bulan November 2024 sampai dengan bulan Desember 2024

***Tabel 3.2**. Waktu Penelitian*

No	Jenis Kegiatan	Waktu kegiatan											
		November				Desember				Januari			
		1	2	3	4	1	2	3	4	1	2	3	4
1	Pengajuan Judul												
2	Pengumpulan Data												
3	Pembuatan Proposal												
4	Ujian Proposal												
5	Pembuatan Skripsi												
6	Ujian Skripsi												

3.4 Alat dan Bahan

Dalam penelitian ini, peneliti memerlukan beberapa alat dan bahan utama untuk mendukung proses pelaksanaan penelitian. Adapun alat dan bahan yang digunakan dijelaskan sebagai berikut:

3.4.1 Alat

1) Laptop

Penelitian ini dilakukan menggunakan laptop dengan spesifikasi:

Processor : AMD Ryzen 5 5500u

RAM : 8GB

Penyimpan 512GB

Sistem Op Dual Boot (Windows untuk penulisan dokumen dan Ubuntu untuk eksekusi program).

2) Akses Internet

3) Software Pendukung

   1) Python sebagai bahasa pemrograman utama, dengan pustaka pendukung

   2) Jupyter Notebook untuk eksperimen dan dokumentasi proses pemrograman.

   3) Microsoft Word untuk menyusun dokumen proposal dan laporan penelitian.

3.4.2 Bahan

1) Dataset Publik

2) Penelitian Terkait

Peneliti memanfaatkan berbagai artikel jurnal, buku, dan referensi ilmiah lainnya sebagai bahan kajian literatur dan pendukung teoritis dalam menyusun metodologi penelitian.

BAB IV

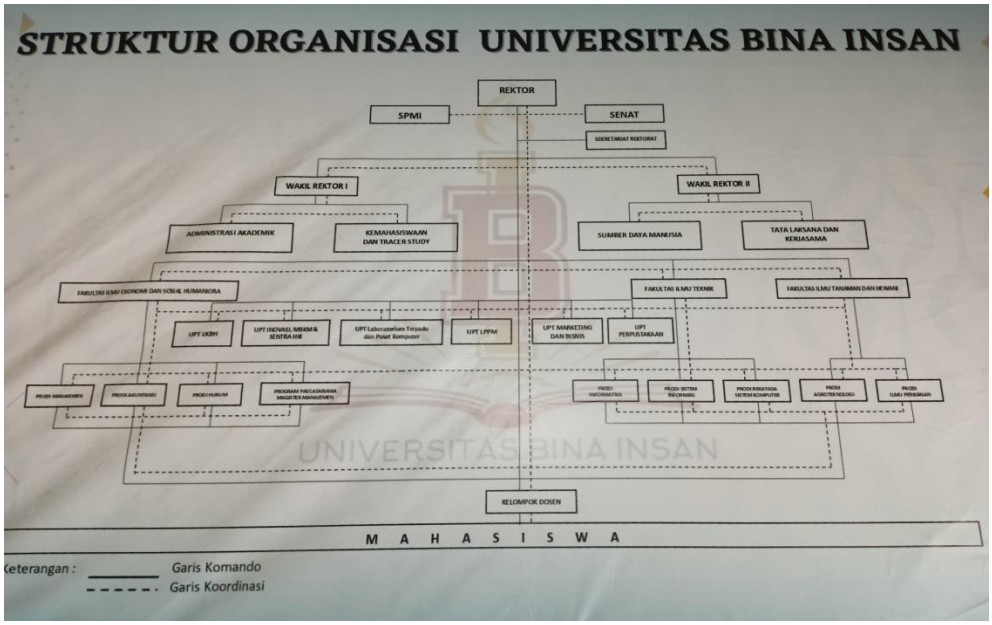
HASIL PENE[illegible]N DAN PEMBAHASAN

4.1 Gambaran Umum

4.1.1 Gambaran Umum

Universitas Bina Insan, Kota Lubuklinggau, merupakan perguruan tinggi swasta yang berada di bawah naungan Yayasan Pendidikan Dwi Tunggal Kota Palembang. Universitas ini didirikan pada tanggal 20 Maret 2019 melalui penggabungan STIE dan STMIK Mura Lubuklinggau. Saat ini, Universitas Bina Insan memiliki tiga fakultas, yaitu Fakultas Ilmu Teknik, Fakultas Ilmu Ekonomi dan Sosial Humaniora, serta Fakultas Ilmu Tanaman dan Hewani, dengan 8 program studi jenjang S1 dan satu program pascasarjana. Universitas ini juga dikenal sebagai perguruan tinggi swasta terbaik di Kota Lubuklinggau, menduduki peringkat pertama di kota tersebut, peringkat ke-775 di Indonesia, dan peringkat ke-12.915 di dunia menurut Webometrics 2024 [24].

4.1.2 Struktur Organisasi



***Gambar 4.11**. Struktur Organisasi Universitas Bina Insan*

4.2 Hasil

Penelitian ini meng[illegible]n temuan signifikan yang mendukung pengembangan sistem de[illegible]ngan Recon berbasis *deep learning*. Data serangan jaringan yang d[illegible]ri file *pcap* berhasil diolah dan fitur-fitur relevan diekstraksi menggunakan *Autoencoder*, dengan menonjolkan elemen-elemen penting seperti protokol, ukuran paket, dan sumber data jaringan. Model CNN yang dikembangkan mampu mengidentifikasi pola-pola kompleks dalam data jaringan dengan tingkat akurasi yang sangat baik, serta efektif dalam membedakan antara data normal dan data serangan berdasarkan fitur yang telah diproses. Evaluasi kinerja model menggunakan metrik akurasi, *precision*, *recall*, dan *F1-score* menunjukkan performa yang stabil dengan tingkat error yang sangat rendah. Penelitian ini memberikan kontribusi penting dalam bidang keamanan siber, khususnya dalam deteksi serangan Recon pada jaringan IoT, dan menambah literatur mengenai penerapan deep learning dalam konteks keamanan jaringan. Secara praktis, hasil penelitian ini memberikan solusi yang dapat meningkatkan keamanan jaringan IoT, dengan harapan dapat menjadi dasar untuk pengembangan sistem deteksi serangan yang lebih efektif di masa depan.

4.3 Pembahasan

4.3.1 Penerapan Metode Analisa dan Validitas Data

Penelitian ini memanfaatkan *Autoencoder* sebagai metode utama untuk mengekstraksi fitur penting dari data mentah yang diperoleh dari jaringan IoT. *Autoencoder* berperan signifikan dalam menyaring informasi relevan sekaligus menonjolkan fitur-fitur utama yang diperlukan untuk mendeteksi serangan Recon secara efektif.

Proses analisis dimulai dengan membaca file PCAP (*Packet Capture*) menggunakan pustaka *Scapy*. Proses ini memungkinkan peneliti untuk mengekstraksi informasi utama dari data jaringan, seperti protokol, ukuran paket, alamat sumber (*source*), dan tujuan (*destination*). Data mentah ini kemudian melalui tahap pra-pemrosesan untuk memastikan relevansi dan kemudahan analisis.

Pada tahap pra-pemrosesan, beberapa kolom, seperti *timestamp*, *source*, dan *destination*, dihapus dataset. Penghapusan ini dilakukan karena kolom-kolom tersebut lai string yang tidak dapat langsung digunakan dalam pelatih berbasis pembelajaran mesin. Selain itu, menghilangkan informasi yang kurang relevan ini juga membantu menyederhanakan data dan mengurangi kompleksitas analisis.

Setelah data disederhanakan, langkah berikutnya adalah normalisasi menggunakan StandardScaler. Normalisasi ini bertujuan untuk menyelaraskan skala setiap fitur dalam dataset, sehingga memastikan bahwa proses pelatihan *Autoencoder* berjalan dengan lebih efisien dan menghasilkan model yang lebih akurat.

Proses pelatihan *Autoencoder* menggunakan fungsi loss *Mean Squared Error* (*MSE*), yang dirancang untuk meminimalkan perbedaan antara data asli dan data hasil rekonstruksi. Setelah pelatihan selesai, bagian *encoder* dari model digunakan untuk menghasilkan representasi fitur yang lebih padat. Representasi ini disimpan dalam format CSV untuk digunakan pada tahap analisis berikutnya.

Dengan pendekatan ini, penelitian memastikan bahwa hanya fitur-fitur yang benar-benar relevan yang dipertahankan dalam dataset, sehingga dapat meningkatkan akurasi dan keandalan deteksi serangan di tahap-tahap analisis berikutnya. Metode ini membuktikan efektivitas Autoencoder dalam menyaring informasi penting dari data jaringan IoT yang kompleks.

Tabel 4.3. Sample Data Hasil Autoencoder

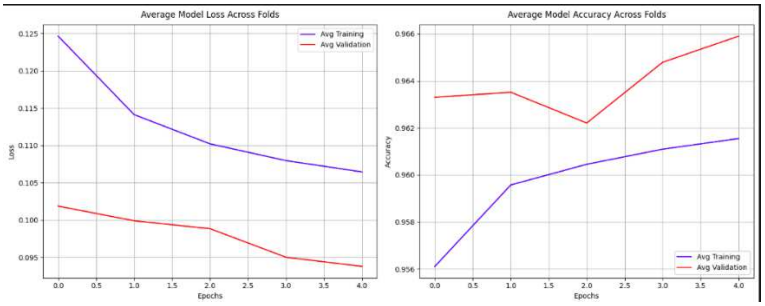
	Feature_0	Feature_1	Feature_2
0	5.167843	2.337601	3.217580
1	5.167843	2.337601	3.217580
2	5.137548	2.238563	3.216436
3	5.145810	2.265573	3.216748
4	0.287781	12.082898	5.835472

4.3.2 Pengujian Hasil Analisa

4.3.2.1 Dua Feat nary Classification

1) Mod n Akurasi

Ga 12 menampilkan dua grafik yang menggambarkan performa model selama pelatihan dengan *cross-validation*. Grafik kiri menunjukkan bahwa *loss* pada *training* dan *validation* menurun secara bertahap, menandakan model semakin baik dalam meminimalkan kesalahan tanpa *overfitting*. Grafik kanan menunjukkan peningkatan akurasi pada kedua *dataset*, dengan akurasi *validation* tetap lebih tinggi dari *training*, mengindikasikan generalisasi yang baik. Model menunjukkan peningkatan performa yang stabil dengan keseimbangan yang baik antara *training* dan *validation*.



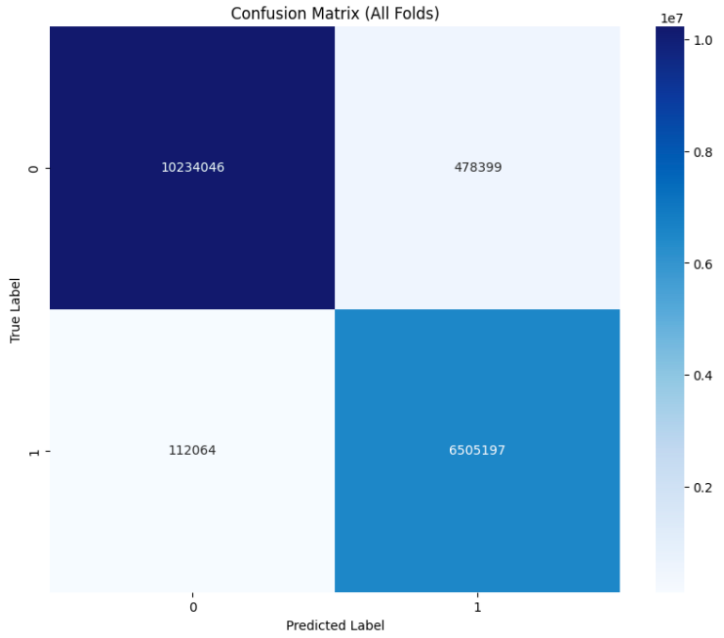
Gambar 4.12 *Model Loss and Model Accuracy 2 Feature Binnary Classification*

2) Confussion Matrix

Gambar 4.13 menampilkan *confusion matrix* yang menunjukkan performa model klasifikasi biner. Model memiliki **10.234.046** *true negatives* (label 0 diklasifikasi dengan benar) dan **6.505.197** *true positives* (label 1 diklasifikasi dengan benar), yang menunjukkan akurasi tinggi.

Kesalahan klasifikasi relatif kecil, dengan **478.399** *false positives* (label 0 salah diklasifikasikan sebagai 1)

dan **112.064** *false negatives* (label 1 salah diklasifikasikan sebagai 0). Dengan jumlah kesalahan yang dibandingkan total data, model menunjukkan performa yang baik dalam membedakan dua kelas.



Gambar 4.13 *Confussion Matrix 2 Feature Binnary Classification*

3) Classification Report

Gambar 4.14 menampilkan *classification report* yang merangkum performa model klasifikasi biner setelah diuji pada seluruh fold. Untuk kelas *Normal/Benign*, model mencapai *precision* sebesar 0.99, *recall* 0.96, dan *f1-score* 0.97 dari total 10.712.445 sampel. Sementara itu, untuk kelas *Attack*, *precision* sebesar 0.93, *recall* 0.98, dan *f1-score* 0.96 dari 6.617.261 sampel.

Model mencapai akurasi 0.97. Nilai *macro average* dan *weighted average* untuk *precision*, *recall*, dan *f1-score* semuanya berada di angka 0.96 hingga 0.97,

menunjukkan keseimbangan performa model di kedua kelas ini mengindikasikan bahwa model memiliki kema ng baik dalam mendeteksi baik serangan maup si normal dengan tingkat kesalahan yang rendah.

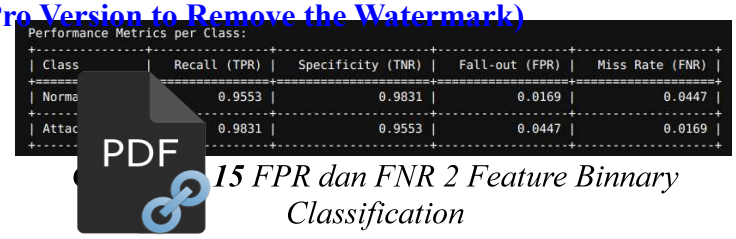
Classification Report (All Folds):				
	precision	recall	f1-score	support
Normal/Benign	0.99	0.96	0.97	10712445
Attack	0.93	0.98	0.96	6617261
accuracy			0.97	17329706
macro avg	0.96	0.97	0.96	17329706
weighted avg	0.97	0.97	0.97	17329706

Gambar 4.14 *Classification Report 2 Feature Binnary Classification*

4) FPR (False Positif Rate) dan FNR (False Negatif Rate)

Gambar 4.15 menampilkan metrik kinerja untuk dua kelas, yaitu Normal/*Benign* dan *Attack*, dengan fokus pada *False Positive Rate* (*FPR*) dan *False Negative Rate* (*FNR*). Untuk kelas *Normal/Benign*, *FPR* sebesar 0.0169 menunjukkan bahwa 1.69% data normal salah diklasifikasikan sebagai serangan, sementara *FNR* sebesar 0.0447 mengindikasikan bahwa 4.47% data normal tidak terdeteksi. Pada kelas *Attack*, *FPR* sebesar 0.0447 berarti 4.47% data serangan salah diklasifikasikan sebagai normal, sedangkan *FNR* sebesar 0.0169 menunjukkan bahwa 1.69% data serangan tidak terdeteksi.

Model ini menunjukkan performa yang baik dalam membedakan kedua kelas, meskipun masih ada ruang untuk perbaikan, terutama dalam mengurangi *FNR* untuk kelas Normal/*Benign* dan *FPR* untuk kelas *Attack*.

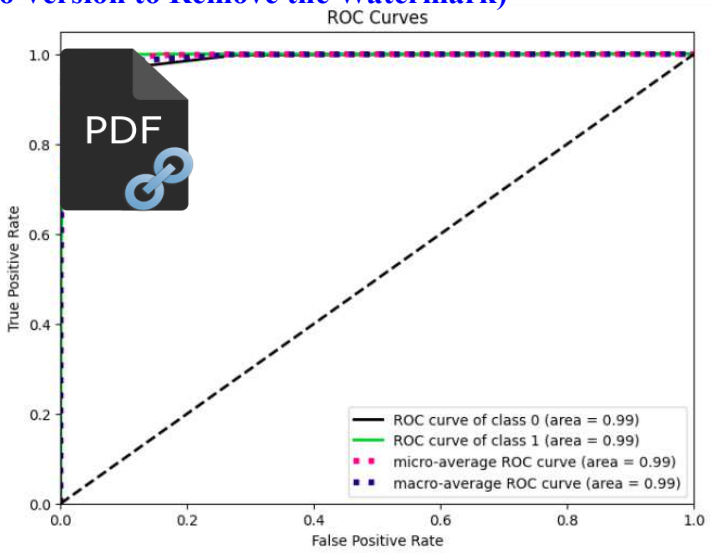


***15** FPR dan FNR 2 Feature Binnary Classification*

5) ROC Curve (Receiver Operating Characteristic Curve)

Gambar 4.16 menampilkan *ROC* Curve (*Receiver Operating Characteristic Curve*) untuk model klasifikasi, yang menunjukkan performa model dalam membedakan dua kelas, yaitu kelas Normal/*Benign* (0) dan kelas *Attack* (1). Area di bawah kurva (*AUC*) untuk kedua kelas adalah 0.99, yang menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam memisahkan kedua kelas dengan akurasi yang tinggi. Kurva *micro-average* dan *macro-average* juga memiliki *AUC* sebesar 0.99, mengindikasikan konsistensi performa model secara keseluruhan.

Dengan nilai *AUC* yang mendekati 1, model ini menunjukkan kinerja yang sangat baik dalam membedakan kedua kelas, dengan tingkat *True Positive Rate* (*TPR*) yang tinggi dan *False Positive Rate* (*FPR*) yang rendah. Hal ini mencerminkan keandalan model dalam melakukan klasifikasi dengan presisi yang sangat baik.

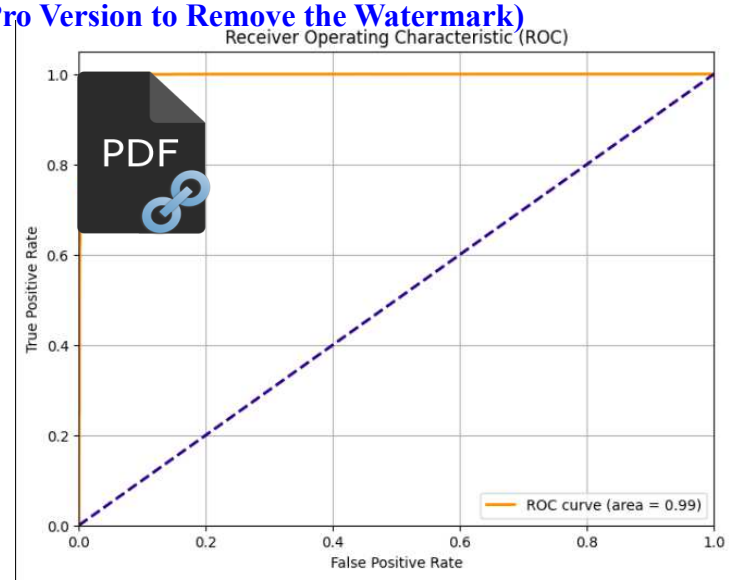


Gambar 4.16 *ROC Curve 2 Feature Binnary Classification*

6) ROC (Receiver Operating Characteristic)

Gambar 4.17 menampilkan *ROC* (*Receiver Operating Characteristic*) yang menggambarkan performa model klasifikasi. Area di bawah kurva (*AUC*) sebesar 0.99 menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan antara kelas *Attack* dan Normal/*Benign*. Nilai *AUC* yang mendekati 1 mengindikasikan bahwa model mencapai *True Positive Rate* (*TPR*) yang tinggi sambil menjaga *False Positive Rate* (*FPR*) tetap rendah.

Kurva *ROC* ini mencerminkan keandalan dan presisi model yang sangat kuat dalam melakukan klasifikasi, dengan performa yang hampir sempurna dalam memisahkan kedua kelas.

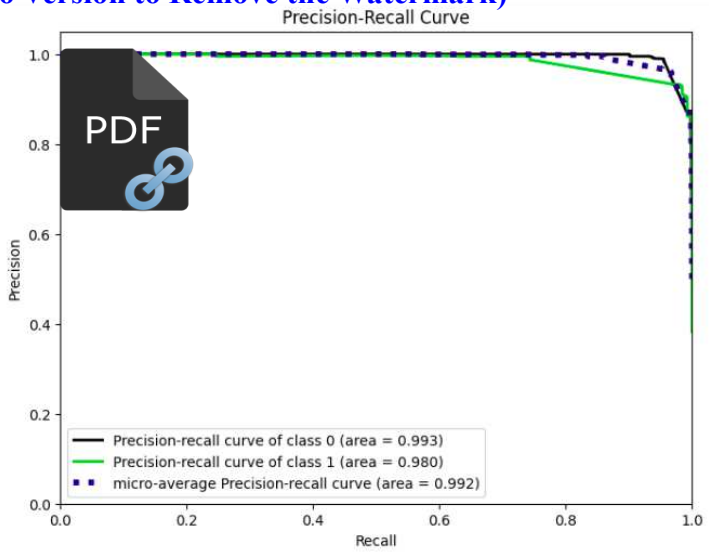


Gambar 4.17 *ROC 2 Feature Binnary Classification*

7) Precision Recall Curve

Gambar 4.18 menampilkan *Precision-Recall-Curve* (*PRC*) yang menggambarkan performa model klasifikasi dalam hal presisi dan *recall*. Area di bawah kurva (*AUC*) untuk kelas 0 (Normal/*Benign*) adalah 0.993, sementara untuk kelas 1 (*Attack*) adalah 0.980, menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam memprediksi kedua kelas dengan presisi dan *recall* yang tinggi. Kurva *micro-average* juga memiliki *AUC* sebesar 0.992, yang mencerminkan konsistensi performa model secara keseluruhan.

Dengan nilai *AUC* yang mendekati 1, model ini menunjukkan kinerja yang sangat kuat dalam menyeimbangkan presisi dan *recall*, yang mengindikasikan bahwa model dapat mengidentifikasi sebagian besar *instance positif* dengan akurasi yang tinggi. Hal ini mencerminkan keandalan model dalam melakukan klasifikasi dengan performa yang sangat baik.

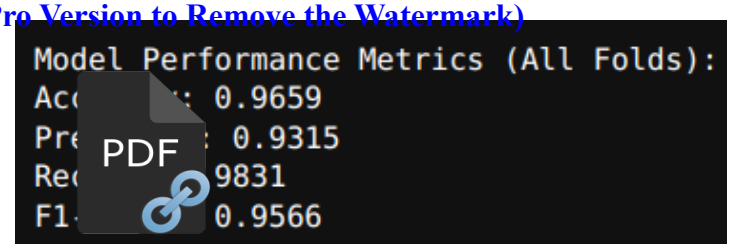


Gambar 4.18 *Precission Recall Curve 2 feature Binnary Classification*

8) Model Performance Matrix

Gambar 4.19 menampilkan *Model Performance Matrix* yang mencakup akurasi, presisi, *recall*, dan *F1-score*. Model ini mencapai akurasi sebesar 0.9659, yang menunjukkan bahwa model mampu mengklasifikasikan data dengan benar hampir 96.6% dari total data. Presisi sebesar 0.9315 mengindikasikan bahwa model memiliki tingkat kesalahan yang rendah dalam memprediksi kelas positif. *Recall* sebesar 0.9831 menunjukkan bahwa model dapat mengidentifikasi sebagian besar *instance positif* dengan baik.

F1-score sebesar 0.9566 mencerminkan keseimbangan yang sangat baik antara presisi dan *recall*, yang menandakan bahwa model tidak hanya akurat tetapi juga konsisten dalam memprediksi kedua kelas. Metrik ini menunjukkan bahwa model memiliki performa yang sangat kuat dan dapat diandalkan dalam melakukan tugas klasifikasi.



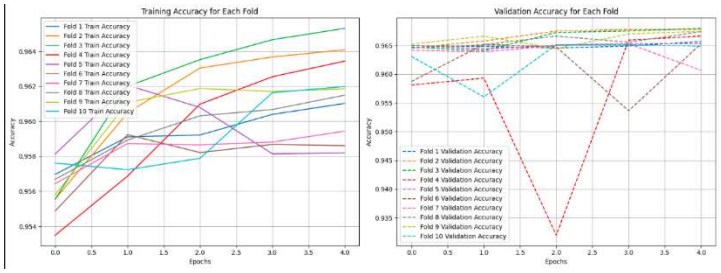
Gambar 4.19 *Model Performance Matrix 2 feature Binnary Classification*

9) Cross Validation

Gambar 4.20 menampilkan akurasi pelatihan dan validasi untuk setiap *fold* dalam proses *cross-validation*. Akurasi pelatihan untuk setiap *fold* menunjukkan konsistensi yang baik, dengan nilai berkisar antara 0.954 hingga 0.964-an, yang mengindikasikan bahwa model mampu mempelajari data pelatihan dengan efektif di setiap *fold*.

Sementara itu, akurasi validasi untuk setiap *fold* juga menunjukkan performa yang stabil, dengan nilai yang relatif tinggi dan konsisten di seluruh *fold*. Hal ini mencerminkan kemampuan model untuk melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

Grafik ini menunjukkan bahwa model memiliki performa yang konsisten dan dapat diandalkan di setiap *fold*, baik dalam hal pembelajaran dari data pelatihan maupun kemampuan generalisasi terhadap data validasi.



Gambar 4.20 *Training and Validation Accuracy For Each Fold*

4.3.2.2 Dua Feature Multi Classification

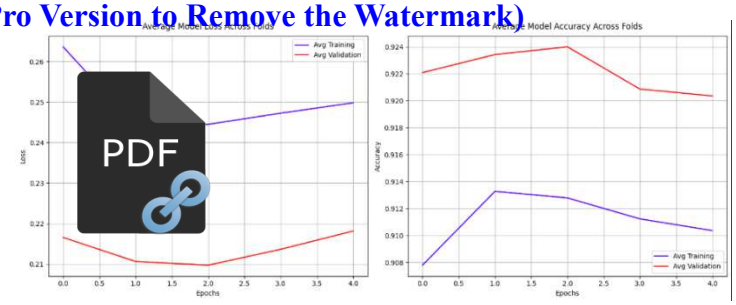
1) Mod dan Akurasi

4.21 menampilkan dua grafik yang menu performa model dalam hal *loss* dan akurasi selama pelatihan menggunakan cross-validation.

Grafik di sebelah kiri menggambarkan perubahan nilai *loss* pada data *training* dan *validation* di setiap *epoch*. *Loss* pada *training* (garis biru) awalnya menurun, tetapi kemudian meningkat kembali setelah epoch ke-2, sementara *loss* pada *validation* (garis merah) menunjukkan tren yang lebih stabil dengan sedikit peningkatan di akhir *epoch*. Pola ini dapat mengindikasikan potensi *overfitting*, di mana model mulai kehilangan kemampuan generalisasi setelah beberapa *epoch*.

Grafik di sebelah kanan menunjukkan perubahan akurasi model. Akurasi pada *validation* (garis merah) awalnya meningkat tetapi kemudian sedikit menurun setelah *epoch* ke-2. Akurasi *training* (garis biru) juga mengalami peningkatan, namun tetap lebih rendah dibandingkan akurasi *validation*. Perbedaan ini menunjukkan bahwa model mungkin belum sepenuhnya belajar secara optimal dari data *training* dan perlu penyesuaian lebih lanjut.

Secara keseluruhan, pola *loss* dan akurasi ini mengindikasikan bahwa model berpotensi mengalami *overfitting* setelah beberapa *epoch*, sehingga perlu dilakukan *fine-tuning* seperti penyesuaian jumlah *epoch*, regulasi tambahan, atau optimasi parameter lainnya.



Gambar 4.21 *Model Loss and Model Accuracy 2 Feature Multi Classification*

2) Confussion Matrix

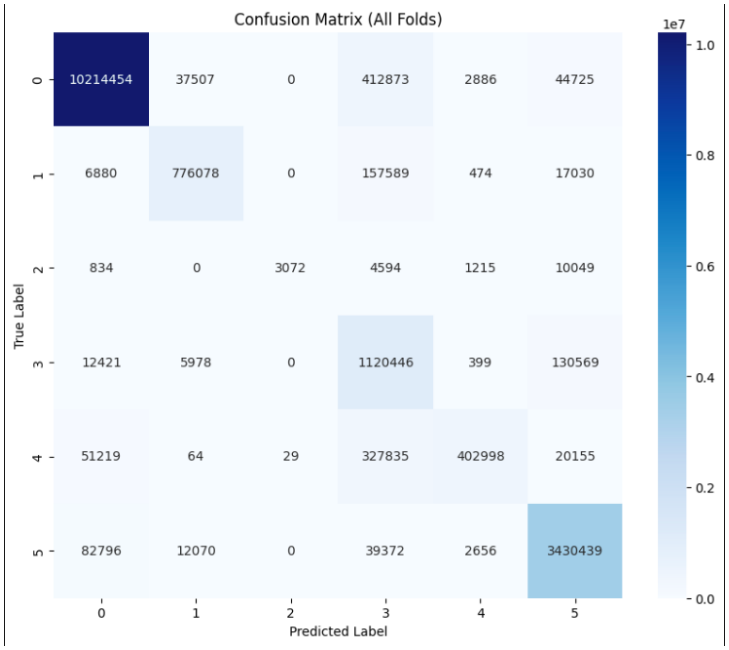
Gambar 4.22 merupakan *confusion matrix* yang menunjukkan performa model klasifikasi *multi*-kelas setelah diuji pada seluruh *fold* dalam *cross-validation*.

Dari *confusion matrix* ini, terlihat bahwa model memiliki performa cukup baik dalam mengklasifikasikan berbagai jenis aktivitas jaringan. Kelas *Benign*/Normal (0) memiliki jumlah prediksi yang benar paling tinggi, dengan 10.214.454 sampel yang diklasifikasikan dengan benar. Namun, terdapat juga sejumlah kesalahan klasifikasi, seperti 412.873 sampel normal yang diklasifikasikan sebagai *Host Discovery* (3).

Beberapa serangan seperti *OS Scan* (1) dan *Host Discovery* (3) juga memiliki tingkat prediksi yang cukup akurat, dengan masing-masing 776.078 dan 1.120.446 sampel diklasifikasikan dengan benar. Namun, kelas seperti *Port Scan* (4) dan *Vulnerability Scan* (5) masih mengalami beberapa kesalahan klasifikasi, terutama dengan adanya sejumlah data yang diklasifikasikan ke kelas lain.

Model menunjukkan performa yang baik tetapi masih mengalami beberapa kesalahan klasifikasi, terutama dalam membedakan beberapa jenis serangan

yang mirip. Hal ini menunjukkan bahwa model mungkin perlu [illegible]ikan lebih lanjut, seperti fine-tuning para[illegible]u peningkatan fitur dam penambahan juml[illegible]meningkatkan akurasi klasifikasi di antara kelas-kelas yang lebih sulit dibedakan.



Gambar 4.22 *Confussion Matrix 2 Feature Multi Classification*

3) Classification Report

Gambar 4.23 menampilkan *Classification Report* yang merangkum performa model untuk setiap kelas berdasarkan metrik *precision*, *recall*, dan *f1-score*. Untuk kelas Normal/*Benign*, model menunjukkan performa yang sangat baik dengan *precision* 0.99, *recall* 0.95, dan *f1-score* 0.97. Hal ini mencerminkan kemampuan model dalam mengidentifikasi *instance* normal dengan akurasi tinggi, yang mungkin didukung oleh jumlah data yang besar (support = 10.712.445).

Untuk kelas *OS Scan*, model memiliki *precision* 0.93 dan 0.81, dengan *f1-score* 0.87. Performa ini masi meskipun ada ruang untuk peningkatan, terut n mendeteksi instance *OS Scan*. Jumlah data untuk kelas ini (support = 958.051) relatif lebih kecil dibandingkan kelas Normal/*Benign*, yang mungkin memengaruhi kemampuan model untuk mempelajari pola dengan lebih baik.

Kelas *Ping Sweep* memiliki *precision* yang tinggi (0.99) tetapi *recall* yang sangat rendah (0.16), *dengan f1-score* hanya 0.27. Performa yang buruk ini kemungkinan besar disebabkan oleh jumlah data yang sangat sedikit (support = 19.764). Ketika data suatu kelas sangat sedikit, model kesulitan mempelajari pola yang representatif, sehingga sering melewatkan *instance* yang seharusnya dikenali (*recall* rendah).

Kelas *Host Discovery* memiliki *recall* yang tinggi (0.88) tetapi *precision* yang lebih rendah (0.54), dengan *f1-score* 0.67. Ini menunjukkan bahwa model sering salah mengklasifikasikan *instance* lain sebagai *Host Discovery*. Meskipun jumlah datanya cukup besar (support = 1.269.813), ketidakseimbangan data atau kompleksitas pola dalam kelas ini mungkin menjadi penyebabnya.

Untuk kelas *Port Scan*, *precision* tinggi (0.98) tetapi *recall* rendah (0.50), dengan *f1-score* 0.66. Performa ini mengindikasikan bahwa model akurat dalam memprediksi *Port Scan* ketika dilakukan, tetapi sering melewatkan *instance* yang sebenarnya. Jumlah data yang relatif kecil (support = 802.300) mungkin menjadi

penyebab *recall* yang rendah, karena model tidak mem kup contoh untuk belajar dengan baik.

nerability Scan menunjukkan performa yang ik dengan *precision* 0.94, *recall* 0.96, dan *f1-score* 0.95. Hal ini mencerminkan kemampuan model dalam mendeteksi *instance Vulnerability Scan* dengan akurasi tinggi, yang mungkin didukung oleh jumlah data yang cukup besar (support = 3.567.333).

Laporan ini menunjukkan bahwa model memiliki performa yang baik, terutama untuk kelas dengan data yang melimpah seperti Normal/*Benign* dan *Vulnerability Scan*. Namun, untuk kelas dengan data yang sedikit seperti *Ping Sweep* dan *Port Scan*, performa model cenderung lebih rendah, terutama dalam hal *recall*. Hal disebabkan oleh ketidakseimbangan data, di mana kelas minoritas tidak memiliki cukup contoh untuk dipelajari oleh model.

Classification Report (All Folds):

	precision	recall	f1-score	support
Normal/Benign	0.99	0.95	0.97	10712445
OS Scan	0.93	0.81	0.87	958051
Ping Sweep	0.99	0.16	0.27	19764
Host Discovery	0.54	0.88	0.67	1269813
Port Scan	0.98	0.50	0.66	802300
Vulnerability Scan	0.94	0.96	0.95	3567333
accuracy			0.92	17329706
macro avg	0.90	0.71	0.73	17329706
weighted avg	0.94	0.92	0.92	17329706

Gambar 4.23 *Classification Report 2 Feature Multi Classification*

4) FPR (False Positif Rate) dan FNR (False Negatif Rate)

Gambar 4.24 menampilkan metrik kinerja untuk setiap kelas dengan fokus pada *False Positive Rate* (*FPR*) dan *False Negative Rate* (*FNR*). Untuk kelas Normal/*Benign*, *FPR* sebesar 0.0233 menunjukkan

bahwa 2.33% data normal salah diklasifikasikan sebagai seran sementara *FNR* sebesar 0.0465 meng an bahwa 4.65% data normal tidak terde

Pada kelas *OS Scan*, *FPR* sangat rendah (0.0034), tetapi *FNR* cukup tinggi (0.1899), yang berarti model sering melewatkan *instance OS Scan*. Untuk kelas *Ping Sweep*, *FPR* 0.0000 menunjukkan tidak ada kesalahan positif, tetapi *FNR* sangat tinggi (0.8446), menandakan bahwa model kesulitan mendeteksi *instance Ping Sweep*.

Kelas *Host Discovery* memiliki *FPR* 0.0587 dan *FNR* 0.1176, menunjukkan beberapa kesalahan dalam mengidentifikasi *instance* ini. Untuk kelas *Port Scan*, *FPR* sangat rendah (0.0005), tetapi *FNR* cukup tinggi (0.4977), mengindikasikan bahwa model sering melewatkan instance Port Scan.

Terakhir, kelas *Vulnerability Scan* memiliki *FPR* 0.0162 dan *FNR* 0.0384, yang menunjukkan performa yang sangat baik dalam mendeteksi *instance* ini.

Secara keseluruhan, model ini memiliki *FPR* yang rendah untuk sebagian besar kelas, tetapi *FNR* yang tinggi untuk kelas minoritas seperti Ping Sweep dan Port Scan, karena ketidakseimbangan data.

Performance Metrics per Class:

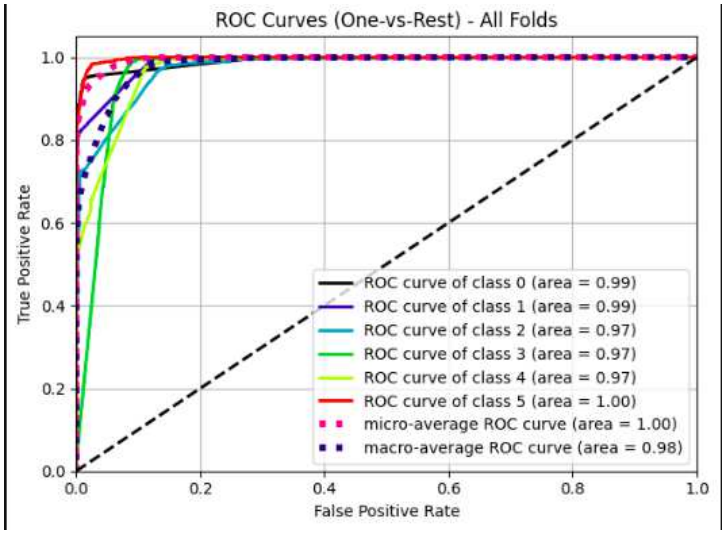
Class	Recall (TPR)	Specificity (TNR)	Fall-out (FPR)	Miss Rate (FNR)
Normal/Benign	0.9535	0.9767	0.0233	0.0465
OS Scan	0.8101	0.9966	0.0034	0.1899
Ping Sweep	0.1554	1.0000	0.0000	0.8446
Host Discovery	0.8824	0.9413	0.0587	0.1176
Port Scan	0.5023	0.9995	0.0005	0.4977
Vulnerability Scan	0.9616	0.9838	0.0162	0.0384

Gambar 4.24 *FNR dan FPR 2 Feature Multi Classification*

5) ROC Curve (Receiver Operating Characteristic Curve)

Gambar 4.25 menampilkan *ROC Cruve* (*Receiver Operating Characteristic Curve*) untuk model klasifikasi dengan pendekatan *One-vs-Rest* di semua *fold*. Area di bawah kurva (*AUC*) untuk setiap kelas sangat tinggi, dengan nilai 0.99 untuk kelas 0 (Normal/*Benign*) dan 1 (*Os Scan*), 0.97 untuk kelas 2 (*Ping Sweep*), 3 (*Host Discovery*), dan 4 (*Port Scan*), serta 1.00 untuk kelas 5 (*Vulnerability Scan*). Kurva *micro-average* dan *macro-average* juga menunjukkan performa yang sangat baik, dengan *AUC* masing-masing 1.00 dan 0.98.

Nilai *AUC* yang mendekati 1 menunjukkan bahwa model memiliki kemampuan yang sangat kuat dalam membedakan setiap kelas dengan *True Positive Rate* (*TPR*) yang tinggi dan *False Positive Rate* (*FPR*) yang rendah. Secara keseluruhan, kurva *ROC* ini mencerminkan keandalan model dalam melakukan klasifikasi multi-kelas dengan presisi yang sangat baik.

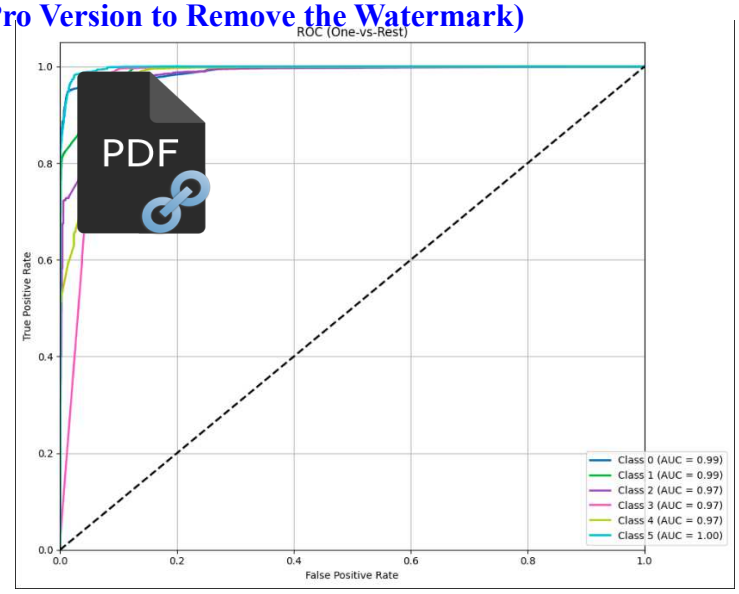


Gambar 4.25 *ROC Curve 2 Feature Multi Classification*

6) ROC (Receiver Operating Characteristic)

4.26 menampilkan *ROC* (*Receiver Operating Characteristic*) untuk model klasifikasi dengan pendekatan *One-vs-Rest*. Nilai *AUC* (*Area Under Curve*) untuk setiap kelas sangat tinggi, yaitu 0.99 untuk kelas 0 (Normal/*Benign*) dan 1 (*Os Scan*), 0.97 untuk kelas 2 (*Ping Sweep*), 3 (*Host Discovery*), dan 4 (*Port Scan*), serta 1.00 untuk kelas 5 (*Vulnerability Scan*).

Nilai *AUC* yang mendekati 1 menunjukkan bahwa model memiliki kemampuan yang sangat kuat dalam membedakan setiap kelas dari kelas lainnya, dengan *True Positive Rate* (*TPR*) yang tinggi dan *False Positive Rate* (*FPR*) yang rendah. Kurva *ROC* ini mencerminkan keandalan model dalam melakukan klasifikasi multi-kelas dengan presisi yang sangat baik. Meskipun beberapa kelas mungkin memiliki *recall* yang rendah (seperti kelas 2/*Ping Sweep*), *AUC* yang tinggi menunjukkan bahwa model secara *intrinsik* mampu membedakan kelas-kelas ini dengan baik.



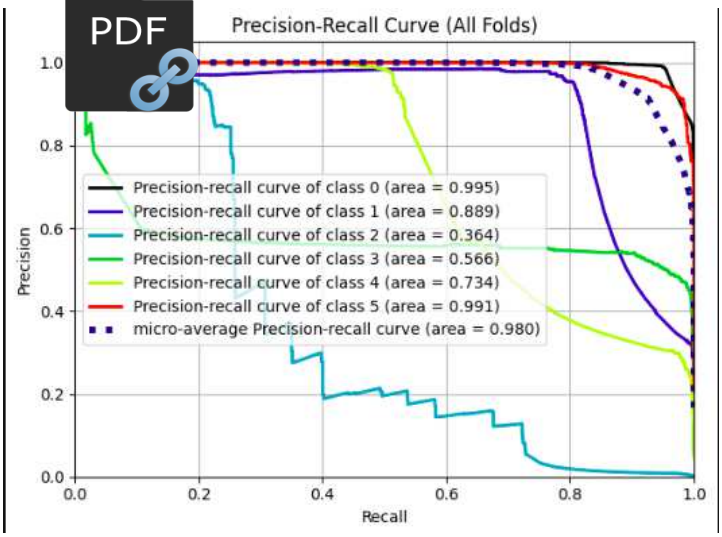
Gambar 4.26 *ROC 2 Feature Multi CLassification*

7) Precision Recall Curve

Gambar 4.27 menampilkan *Precision-Recall-Curve* (*PRC*) untuk model klasifikasi di semua *fold*. Area di bawah kurva (*AUC*) bervariasi untuk setiap kelas: 0.995 untuk kelas 0 (Normal/*Benign*), 0.889 untuk kelas 1 (*Os Scan*), 0.364 untuk kelas 2 (*Ping Sweep*), 0.566 untuk kelas 3 (*Host Discovery*), 0.734 untuk kelas 4 (*Port Scan*), dan 0.991 untuk kelas 5 (*Vulnerability Scan*). Kurva *micro-average* juga menunjukkan performa yang sangat baik dengan *AUC* 0.980.

Nilai *AUC* yang tinggi untuk kelas 0 (Nomal/*Benign*) dan 5 (*Vulnerability Scan*) menunjukkan bahwa model sangat baik dalam memprediksi kelas-kelas tersebut dengan presisi dan *recall* yang tinggi. Namun, untuk kelas 2 (*Ping Sweep*), *AUC* yang rendah (0.364) mencerminkan kesulitan model dalam mendeteksi *instance* dari kelas ini, karena jumlah data yang sedikit. Secara keseluruhan, kurva ini menunjukkan bahwa model memiliki performa yang

kuat untuk sebagian besar kelas, meskipun ada tantangan dalam deteksi kelas minoritas seperti *Ping Sweep*.



Gambar 4.27 *Precision Recall Curve 2 Feature Multi Classification*

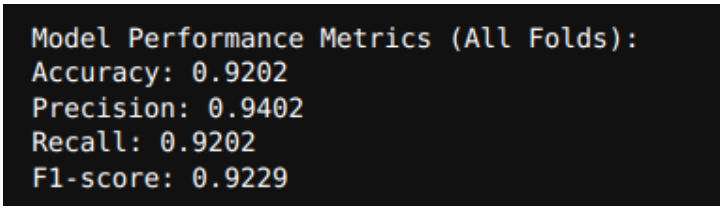
8) Model Performance Matrix

Gambar 4.28 menampilkan *Model Performance Matrix* di semua *fold* untuk 6 kelas, yaitu Normal (kelas 0) dan 5 jenis serangan (kelas 1-5). Model mencapai akurasi sebesar 0.9202, yang menunjukkan bahwa model mampu mengklasifikasikan data dengan benar sekitar 92.02% dari total data. *Precision* sebesar 0.9402 mengindikasikan bahwa model memiliki tingkat kesalahan yang rendah dalam memprediksi kelas positif, baik untuk kelas Normal maupun serangan.

Recall sebesar 0.9202 menunjukkan bahwa model dapat mengidentifikasi sebagian besar *instance* positif dengan baik, termasuk *instance* Normal dan serangan. *F1-score* sebesar 0.9229 mencerminkan keseimbangan yang baik antara *precision* dan *recall*, yang menandakan bahwa model tidak hanya akurat tetapi juga konsisten

dalam memprediksi kedua jenis kelas (Normal dan seran

eluruhan, metrik ini menunjukkan bahwa mode iki performa yang kuat dan dapat diandalkan dalam membedakan antara aktivitas Normal dan berbagai jenis serangan. Namun, untuk kelas serangan tertentu (seperti yang terlihat di laporan klasifikasi sebelumnya), mungkin masih ada ruang untuk perbaikan, terutama jika data untuk kelas serangan tersebut tidak seimbang.

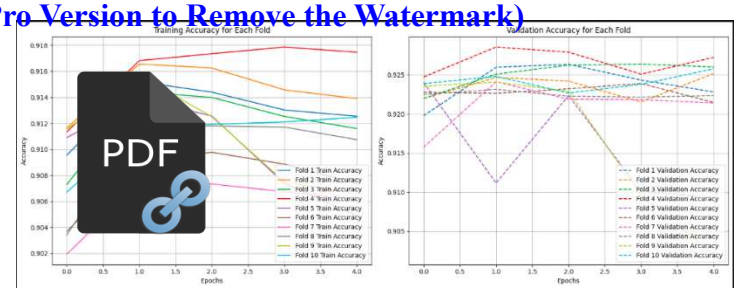


Gambar 4.28 *Model Performance Matrix 2 Feature Multi Classification*

9) Cross Validation

Gambar 4.29 menampilkan akurasi pelatihan dan validasi untuk setiap *fold* dalam proses *cross-validation*. Akurasi pelatihan untuk setiap *fold* berkisar antara 0.905 hingga 0.918, menunjukkan konsistensi yang baik dalam pembelajaran model dari data pelatihan. Sementara itu, akurasi validasi untuk setiap *fold* juga stabil, dengan nilai berkisar antara 0.905 hingga 0.925, yang mencerminkan kemampuan model untuk melakukan generalisasi dengan baik terhadap data yang belum pernah dilihat sebelumnya.

Secara keseluruhan, model menujukkan performa yang cukup konsisten di setiap *fold*, tanpa indikasi overfitting yang signifikan.



Gambar 4.29 *Training and Validation Accuracy For Each Fold*

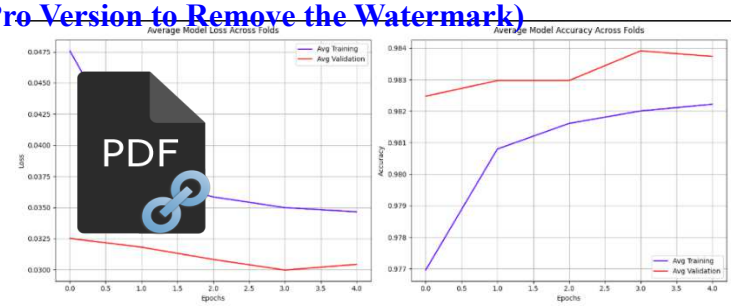
4.3.2.3 Tiga Feature Binnary Classification

1) Model Loss dan Akurasi

Gambar 4.30 menampilkan rata-rata *loss* dan akurasi model di seluruh *fold* selama proses pelatihan dan validasi. Pada grafik *loss*, terlihat bahwa *loss* pelatihan dan validasi menurun secara stabil seiring dengan bertambahnya *epoch*, menunjukkan bahwa model semakin baik dalam meminimalkan kesalahan. *Loss* pelatihan berkisar antara 0.0350 hingga 0.0475, sementara *loss* validasi juga menunjukkan tren penurunan yang serupa.

Pada grafik akurasi, akurasi pelatihan dan validasi meningkat secara konsisten, dengan akurasi pelatihan mencapai sekitar 0.982 dan akurasi validasi mendekati 0.984 setelah 5 *epoch*. Hal ini mencerminkan bahwa model tidak hanya belajar dengan baik dari data pelatihan tetapi juga mampu melakukan generalisasi yang baik terhadap data validasi.

Grafik ini menunjukkan bahwa model memiliki performa yang stabil dan konsisten, dengan kemampuan yang baik dalam meminimalkan *loss* dan meningkatkan akurasi selama proses pelatihan dan validasi.

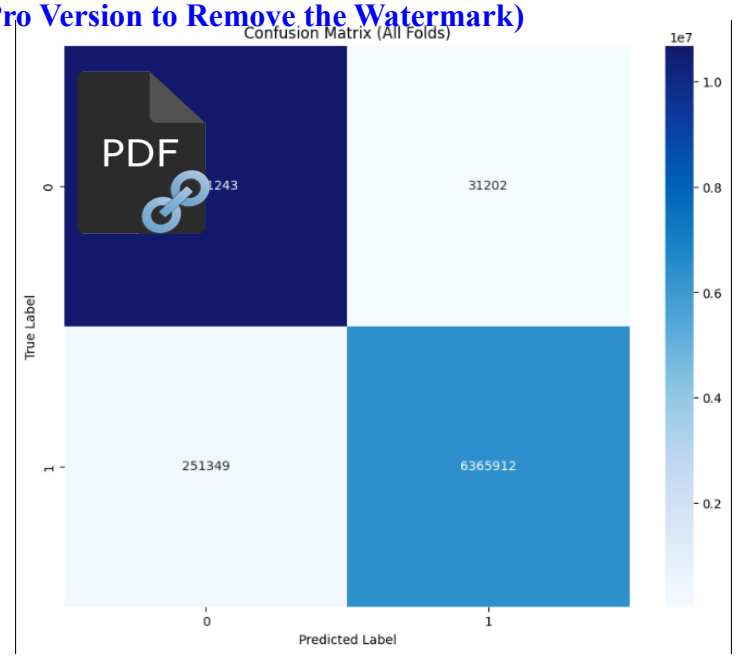


Gambar 4.30 *Model Loss and Model Accuracy 3 Feature Binnary Classification*

2) Confussion Matrix

Gambar 4.31 menampilkan *confusion matrix* yang menunjukkan performa model klasifikasi biner. Model memiliki **10.681.243** *true negatives* (label 0 diklasifikasi dengan benar) dan **6.365.912** *true positives* (label 1 diklasifikasi dengan benar), yang menunjukkan akurasi tinggi.

Kesalahan klasifikasi relatif kecil, dengan **31.202** *false positives* (label 0 salah diklasifikasikan sebagai 1) dan **251.349** *false negatives* (label 1 salah diklasifikasikan sebagai 0). Dengan jumlah kesalahan yang rendah dibandingkan total data, model menunjukkan performa yang baik dalam membedakan dua kelas.

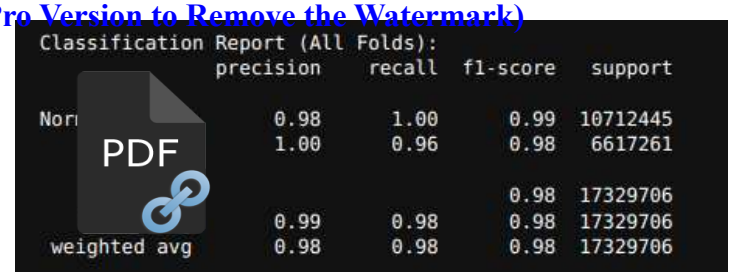


Gambar 4.31 *Confussion Matrix 3 Feature Binnary Classification*

3) Classification Report

Gambar 4.32 menampilkan *classification report* yang merangkum performa model dalam mengklasifikasikan dua kelas, yaitu Normal/*Benign* dan *Attack*. Model menunjukkan precision sebesar 98% untuk kelas Normal/*Benign* dan 100% untuk kelas *Attack*, sementara *recall* mencapai 100% untuk Normal/*Benign* dan 96% untuk *Attack*. Nilai *f1-score* untuk kedua kelas juga tinggi, masing-masing 99% dan 98%, yang menandakan keseimbangan antara *precision* dan *recall*.

Model memiliki akurasi 98%, dengan *macro average f1-score* sebesar 98%. Hal ini menunjukkan bahwa model dapat mengidentifikasi kedua kelas dengan sangat baik, meskipun masih terdapat beberapa instance serangan yang salah diklasifikasikan sebagai kondisi normal.

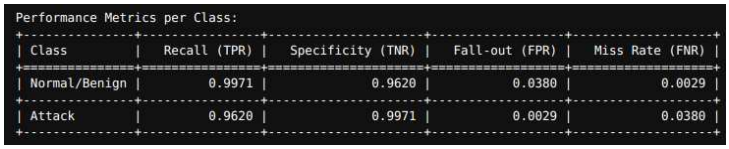


Gambar 4.32 *Classification Report 3 Feature Binnary Classification*

4) FPR (False Positive Rate) dan FNR (False Negatif Rate)

Gambar 4.33 menampilkan metrik *False Positive Rate* (*FPR*) dan *False Negative Rate* (*FNR*) untuk kedua kelas. Nilai *FPR* untuk kelas Normal/*Benign* adalah 0.0380, yang berarti sekitar 3.8% dari *instance* sebenarnya normal salah diklasifikasikan sebagai serangan. Sementara itu, kelas *Attack* memiliki *FPR* yang jauh lebih kecil, yaitu 0.0029, menunjukkan bahwa hampir tidak ada instance serangan yang salah diklasifikasikan sebagai normal.

Sebaliknya, *FNR* untuk kelas Normal/*Benign* hanya 0.0029, yang berarti sangat sedikit instance normal yang terdeteksi sebagai serangan. Namun, untuk kelas *Attack*, *FNR* lebih tinggi, yaitu 0.0380, yang menunjukkan bahwa sekitar 3.8% serangan salah diklasifikasikan sebagai normal. Metrik ini menunjukkan bahwa model lebih cenderung salah mengklasifikasikan serangan sebagai kondisi normal daripada sebaliknya.

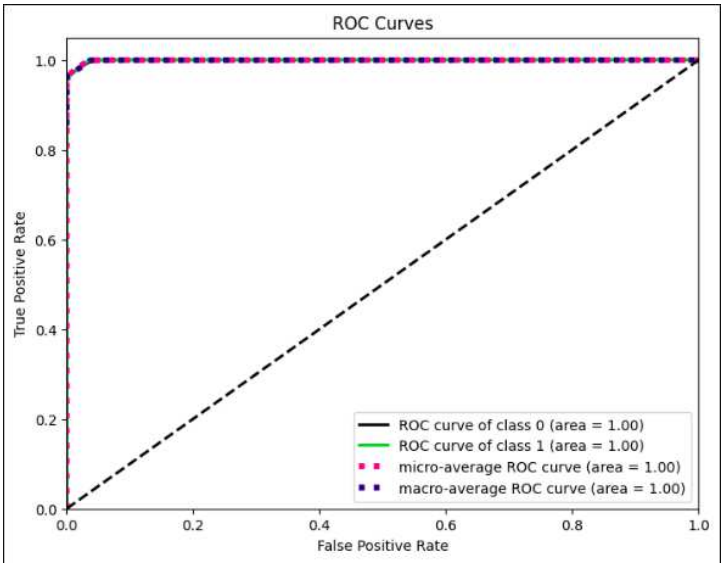


Gambar 4.33 *FPR dan FNR 3 Feature Binnary Classification*

5) ROC Curve (Receiver Operating Characteristic Curve)

Gambar 4.34 menampilkan *ROC Curve* (*Receiver Operating Characteristic Curve*) untuk model klasifikasi yang menunjukkan seberapa baik model dalam membedakan dua kelas, yaitu kelas 0 (Normal/*Benign*) dan kelas 1 (*Attack*). Area di bawah kurva (*AUC*) untuk kedua kelas adalah 1.00, yang menandakan bahwa model memiliki performa sempurna dalam memisahkan kedua kelas tanpa kesalahan.

Kurva *micro-average* dan *macro-average* juga memiliki *AUC* sebesar 1.00, menunjukkan bahwa model mempertahankan konsistensi performa di semua kelas. Dengan nilai *AUC* yang mencapai 1, model ini memiliki *True Positive Rate* (*TPR*) yang sangat tinggi dan *False Positive Rate* (*FPR*) yang sangat rendah, yang berarti model dapat melakukan klasifikasi dengan akurasi dan keandalan yang sangat tinggi.

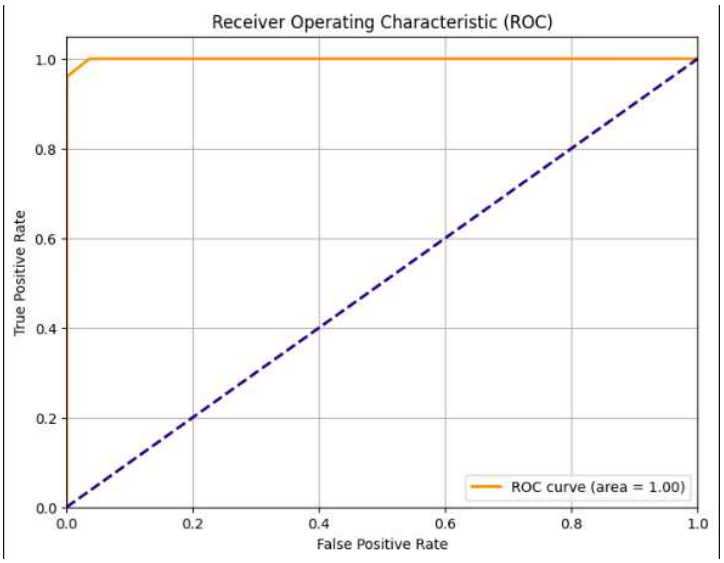


Gambar 4.34 *ROC Curve 3 Feature Binnary Classification*

6) ROC (Receiver Operating Characteristic)

Gambar 4.35 menampilkan *ROC* (*Receiver Operating Characteristic*) yang menggambarkan performa model klasifikasi dalam membedakan dua kelas. Garis horizontal di bagian atas menunjukkan bahwa model memiliki tingkat *True Positive Rate* (*TPR*) yang sangat tinggi, sementara garis diagonal putus-putus berfungsi sebagai garis acuan yang menunjukkan performa acak.

Area di bawah kurva (*AUC*) memiliki nilai 1.00, yang berarti model mampu mengklasifikasikan data dengan sempurna tanpa kesalahan. Dengan nilai *AUC* yang mencapai 1, model ini menunjukkan kinerja yang sangat baik, dengan *False Positive Rate* (*FPR*) yang sangat rendah dan kemampuan tinggi dalam mengenali data dengan benar.



Gambar 4.35 *ROC 3 Feature Binnary Classification*

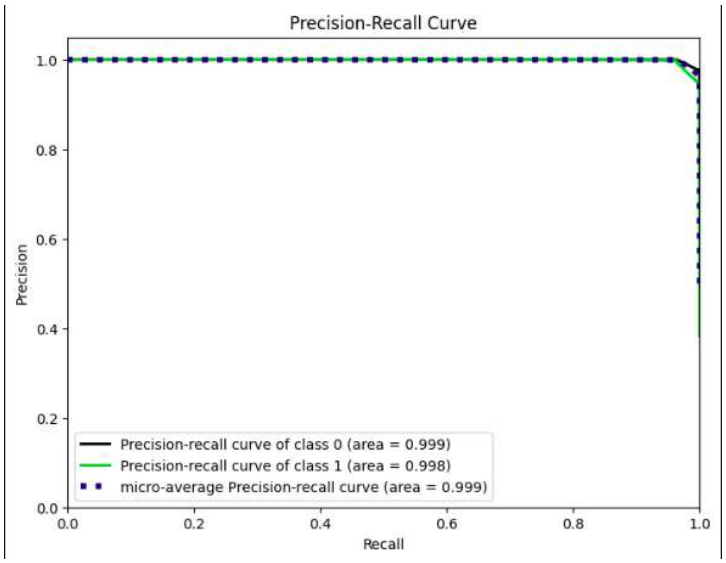
7) Precision Recall Curve

Gambar 4.36 menampilkan *Precision-Recall-Curve* yang digunakan untuk mengevaluasi performa model klasifikasi. *Precision* menunjukkan seberapa

akurat prediksi positif model, sedangkan *Recall* menu n seberapa baik model dalam menangkap semu positif.

ng ditampilkan hampir mendekati nilai *Precision* = 1 sepanjang grafik, yang menandakan bahwa model memiliki performa yang sangat baik. Area di bawah kurva (*AUC*) untuk kelas 0 (Normal/*Benign*) adalah 0.999, untuk kelas 1 (*Attack*) adalah 0.998, dan untuk *micro-average* adalah 0.999. Nilai *AUC* yang tinggi ini menunjukkan bahwa model mampu mengidentifikasi kelas dengan tingkat akurasi yang hampir sempurna.

Dengan *Precision* dan *Recall* yang tinggi, model ini sangat andal dalam melakukan klasifikasi dan meminimalkan kesalahan dalam mendeteksi kelas yang benar.



Gambar 4.36 *Precision Recall Curve 3 Feature Binnary Classification*

8) Model Performance Matrix

Gambar 4.37 menunjukkan *Model Performance Matrix* dalam membedakan antara aktivitas normal dan

serangan. Dengan akurasi sebesar 98.37%, model ini mam gklasifikasikan hampir semua data dengan bena *n* yang mencapai 99.51% menunjukkan bahw semua aktivitas yang diklasifikasikan sebagai serangan memang benar-benar serangan, dan semua aktivitas normal yang diklasifikasikan sebagai normal juga memang benar-benar normal. Sementara itu, *recall* sebesar 96.20% menandakan bahwa model dapat mendeteksi sebagian besar serangan yang terjadi, dengan hanya sedikit yang terlewat. Dengan *F1-score* sebesar 97.83%, model ini menunjukkan keseimbangan yang sangat baik antara *precision* dan *recall*, menjadikannya sangat andal dalam mengenali aktivitas normal dan mendeteksi serangan dengan tingkat kesalahan yang sangat rendah.

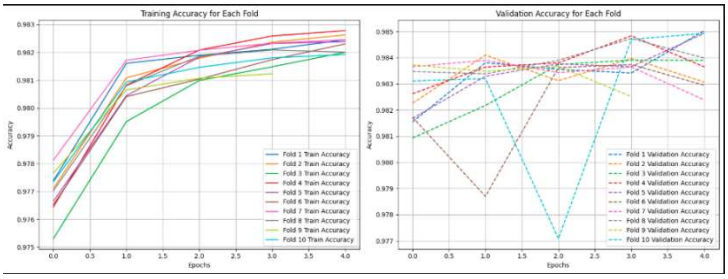
```
Model Performance Metrics (All Folds):
Accuracy: 0.9837
Precision: 0.9951
Recall: 0.9620
F1-score: 0.9783
```

Gambar 4.37 *Model Performance Matrix 3 Feature Binnary Classification*

9) Cross Validation

Gambar 4.38 menampilkan grafik akurasi selama pelatihan dan validasi untuk setiap *fold* dalam proses *cross-validation*. Grafik di sebelah kiri menunjukkan peningkatan akurasi pelatihan seiring bertambahnya *epoch*, di mana semua *fold* mengalami tren naik yang konsisten dan mendekati nilai maksimal sekitar 0.983. Hal ini menunjukkan bahwa model berhasil belajar dari data dengan baik selama pelatihan.

Sementara itu, grafik di sebelah kanan meng[illegible]rkan akurasi validasi untuk setiap *fold*. Mesk[illegible]ra umum nilai akurasi validasi tetap tinggi di s[illegible]982 hingga 0.985, terdapat beberapa fluktuasi di beberapa *fold*, yang mungkin mengindikasikan adanya variasi dalam distribusi data validasi. Namun, secara keseluruhan, model menunjukkan performa yang stabil di seluruh *fold*, dengan akurasi validasi yang tetap tinggi.



Gambar 4.38 *Training and Validation Accuracy For Each Fold 3 Feature Binnary Classification*

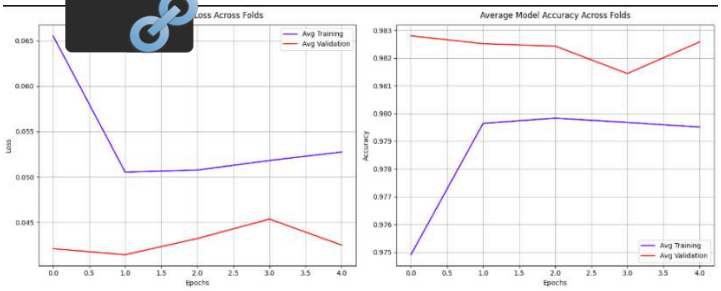
4.3.2.4 Tiga Feature Multi Classification

1) Model Loss dan Akurasi

Gambar 4.39 menampilkan rata-rata *loss* dan akurasi model selama pelatihan dan validasi di seluruh *fold*. Grafik di sebelah kiri menunjukkan tren *loss*, di mana *loss* pelatihan (garis biru) menurun tajam pada epoch pertama, kemudian cenderung stabil dengan sedikit peningkatan. Sementara itu, *loss* validasi (garis merah) tetap relatif rendah namun mengalami sedikit fluktuasi, yang dapat mengindikasikan adanya variasi dalam data validasi.

Grafik di sebelah kanan menunjukkan tren akurasi model. Akurasi pelatihan (garis biru) meningkat dengan cepat pada awal pelatihan, lalu stabil mendekati 0.98. Akurasi validasi (garis merah) tetap tinggi di sekitar

0.982 hingga 0.983, menunjukkan bahwa model mem nkan performa yang baik tanpa tanda-tanda *overf*



Gambar 4.39 *Model Loss and Model Accuracy 3 Feature Multi Classification*

2) Confussion Matrix

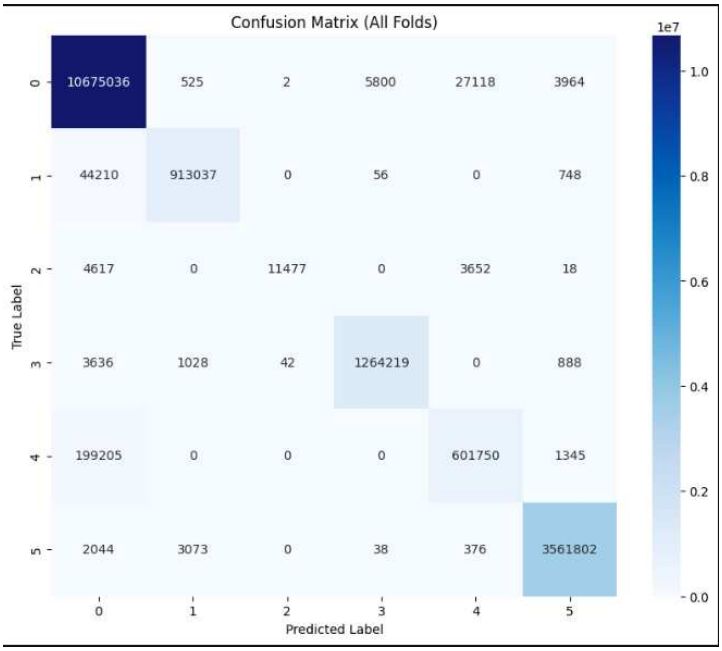
Gambar 4.40 menampilkan *confusion matrix* untuk model klasifikasi dengan 6 kelas, yaitu *Normal/Benign* (0), *OS Scan* (1), *Ping Sweep* (2), *Host Discovery* (3), *Port Scan* (4), *dan Vulnerability Scan* (5). Model menunjukkan performa yang baik dengan sebagian besar prediksi berada di diagonal utama, yang menandakan klasifikasi yang akurat untuk masing-masing kategori.

Kelas *Normal/Benign* (0) memiliki jumlah prediksi benar yang dominan, meskipun masih terdapat beberapa kesalahan klasifikasi, terutama ke kelas *OS Scan* (1) dan *Port Scan* (4). Sementara itu, kelas *OS Scan* (1) sebagian besar diklasifikasikan dengan benar, tetapi ada beberapa instance yang dikira sebagai Normal/*Benign* (0). Untuk kelas *Ping Sweep* (2), model menunjukkan performa yang cukup baik, meskipun terdapat beberapa kesalahan klasifikasi ke kelas Normal/*Benign* (0) dan *Port Scan* (4).

Kelas *Host Discovery* (3) memiliki jumlah prediksi benar yang tinggi dengan hanya sedikit kesalahan klasifikasi. Begitu pula dengan kelas *Port Scan* (4) yang

diklasifikasikan dengan cukup baik, meskipun terdapat bebe *instance* yang diprediksi sebagai Norm (0). Sedangkan untuk kelas *Vuln Scan* (5), model juga mampu mengklasifikasikan dengan baik, dengan hanya sedikit kesalahan ke kelas lain.

Secara keseluruhan, model memiliki kemampuan yang sangat baik dalam membedakan antara kelas-kelas yang ada, dengan hanya sedikit misclassifications yang terjadi.



Gambar 4.40 *Confussion Matrix 3 Feature Multi Classification*

3) Classification Report

Gambar 4.41 menampilkan *classification report* dari model yang menunjukkan performa yang sangat baik dalam mendeteksi berbagai jenis serangan serta trafik normal/*benign*. Model mencapai akurasi keseluruhan sebesar 0.98, yang menunjukkan bahwa model mampu mengklasifikasikan sebagian besar

sampel dengan benar. Selain itu, nilai rata-rata *macro F1-s* besar 0.92 dan *weighted F1-score* sebesar 0.98 inkan keseimbangan yang baik antara presi *all* dalam seluruh kelas.

Pada kelas Normal/*Benign*, model menunjukkan *recall* yang sangat tinggi (1.00), yang berarti hampir semua sampel dari kelas ini berhasil dikenali dengan benar. Presisi sebesar 0.98 menunjukkan bahwa hanya sedikit prediksi yang salah diklasifikasikan sebagai normal padahal sebenarnya merupakan serangan. Hal ini bisa disebabkan oleh jumlah data yang sangat besar pada kelas ini (10.7 juta sampel), yang memberikan model cukup banyak contoh untuk mengenali pola dengan baik.

Untuk kelas serangan, model juga menunjukkan performa yang tinggi secara keseluruhan. Kelas *OS Scan* memiliki presisi 0.99 dan *recall* 0.95, menunjukkan bahwa model cukup akurat dalam mendeteksi serangan ini, meskipun ada beberapa kasus yang tidak terdeteksi. Hal yang sama juga terlihat pada *Port Scan*, dengan presisi 0.95 dan *recall* 0.75, yang menunjukkan bahwa masih ada beberapa *instance* serangan yang tidak terklasifikasi dengan benar.

Namun, pada kelas *Ping Sweep*, hanya memiliki *recall* sebesar 0.58. Artinya, model kesulitan dalam mendeteksi serangan *Ping Sweep*, dan banyak *instance* dari kelas ini yang salah diklasifikasikan ke kelas lain. Hal ini disebabkan oleh jumlah data yang jauh lebih sedikit dibandingkan kelas lainnya (hanya 19.764 sampel), sehingga model tidak mendapatkan cukup banyak variasi untuk mengenali pola dengan baik. Meskipun memiliki presisi 1.00, yang berarti saat model

memprediksi *Ping Sweep*, prediksi tersebut hampir selal[illegible] rendahnya recall menunjukkan bahwa mod[illegible]ering melewatkan deteksi serangan ini darip[illegible]klasifikasikannya dengan benar.

Sebaliknya, kelas *Vulnerability Scan* menunjukkan performa terbaik, dengan presisi dan *recall* 1.00, yang berarti model mampu mengenali semua *instance* dari kelas ini dengan sempurna tanpa salah klasifikasi. Hal ini bisa disebabkan oleh jumlah data yang cukup besar (3.56 juta sampel), yang memungkinkan model belajar pola serangan dengan lebih baik.

Secara keseluruhan, model memiliki performa yang sangat baik dalam mengklasifikasikan berbagai jenis serangan dan trafik normal, tetapi masih menghadapi tantangan dalam mendeteksi serangan dengan jumlah sampel yang lebih sedikit.

Classification Report (All Folds):

	precision	recall	f1-score	support
Normal/Benign	0.98	1.00	0.99	10712445
OS Scan	0.99	0.95	0.97	958051
Ping Sweep	1.00	0.58	0.73	19764
Host Discovery	1.00	1.00	1.00	1269813
Port Scan	0.95	0.75	0.84	802300
Vulnerability Scan	1.00	1.00	1.00	3567333
accuracy			0.98	17329706
macro avg	0.99	0.88	0.92	17329706
weighted avg	0.98	0.98	0.98	17329706

Gambar 4.41 *Classification Report 3 Feature Multi Classification*

4) FPR (False Positive Rate) dan FNR (False Negatif Rate)

Gambar 4.42 Menampilkan *False Positive Rate* (*FPR*) dan *False Negative Rate* (*FNR*) menunjukkan bagaimana model menangani kesalahan klasifikasi dari dua perspektif, kesalahan dalam mendeteksi serangan (*FNR*) dan kesalahan dalam mengklasifikasikan normal sebagai serangan (*FPR*).

Kelas Normal/*Benign* memiliki *FPR* sebesar 0.038[illegible] berarti sekitar 3.83% dari sampel serangan salah[illegible]kasikan sebagai trafik normal. Sementara itu, [illegible]sar 0.0035, menunjukkan bahwa hanya 0.35% dari sampel normal yang salah diklasifikasikan sebagai serangan. Ini menunjukkan bahwa model sangat baik dalam mengenali trafik normal dengan sedikit kesalahan deteksi.

Pada kelas serangan, *OS Scan* memiliki *FPR* yang sangat rendah (0.0003), menunjukkan bahwa hampir tidak ada sampel dari kelas lain yang salah diklasifikasikan sebagai *OS Scan*. Namun, *FNR* sebesar 0.0470 menunjukkan bahwa 4.7% dari serangan *OS Scan* gagal terdeteksi dan diklasifikasikan ke kelas lain.

Masalah utama terlihat pada *Ping Sweep*, di mana *FNR* mencapai 0.4193, artinya 41.93% dari serangan *Ping Sweep* tidak terdeteksi dengan benar dan diklasifikasikan sebagai kelas lain. Namun, *FPR* sebesar 0.0000 menunjukkan bahwa tidak ada kelas lain yang salah diklasifikasikan sebagai Ping Sweep. Ini sejalan dengan observasi sebelumnya bahwa model kesulitan mendeteksi *Ping Sweep* karena jumlah sampel yang sedikit, tetapi saat model memprediksi *Ping Sweep*, prediksinya hampir selalu benar.

Kelas *Port Scan* juga memiliki *FNR* yang cukup tinggi (0.2500), artinya 25% dari serangan *Port Scan* tidak terdeteksi dengan benar. Namun, *FPR* sebesar 0.0019 menunjukkan bahwa hanya 0.19% dari kelas lain yang salah diklasifikasikan sebagai *Port Scan.*

Sebaliknya, kelas *Vulnerability Scan* memiliki *FPR* yang sangat rendah (0.0005) dan *FNR* yang juga sangat

kecil (0.0016), yang berarti model hampir tidak pernah salah m mendeteksi atau mengklasifikasikan seran

seluruhan, model memiliki *FPR* yang sangat rendah untuk semua kelas serangan, yang berarti model jarang salah dalam menganggap trafik normal sebagai serangan. Namun, *FNR* yang tinggi pada kelas *Ping Sweep* dan *Port Scan* menunjukkan bahwa model masih sering melewatkan deteksi pada kelas ini, disebabkan karena jumlah sampel yang lebih sedikit dibandingkan kelas lain..

Performance Metrics per Class:

Class	Recall (TPR)	Specificity (TNR)	Fall-out (FPR)	Miss Rate (FNR)
Normal/Benign	0.9965	0.9617	0.0383	0.0035
OS Scan	0.9530	0.9997	0.0003	0.0470
Ping Sweep	0.5807	1.0000	0.0000	0.4193
Host Discovery	0.9956	0.9996	0.0004	0.0044
Port Scan	0.7500	0.9981	0.0019	0.2500
Vulnerability Scan	0.9984	0.9995	0.0005	0.0016

Gambar 4.42 *FPR dan FNR 3 Feature Multi CLassification*

5) ROC Curve (Receiver Operating Characteristic Curve)

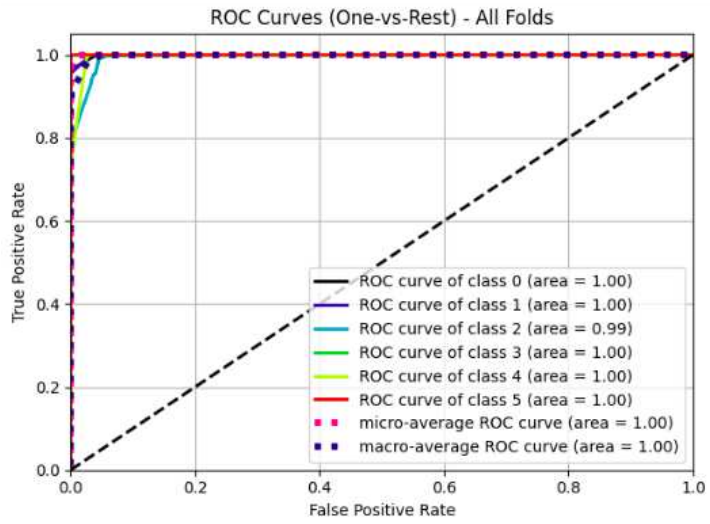
Gambar 4.43 menampilkan grafik *ROC Curve (One-vs-Rest) - All Folds* yang menunjukkan performa model dalam membedakan antara kelas serangan dan trafik normal berdasarkan *True Positive Rate* (*TPR*) dan *False Positive Rate* (*FPR*).

Semua kelas memiliki kurva *ROC* yang sangat dekat dengan pojok kiri atas (0,1), yang menunjukkan bahwa model mampu mendeteksi hampir semua serangan dengan tingkat kesalahan yang sangat rendah. *Area Under Curve* (*AUC*) untuk sebagian besar kelas adalah 1.00, kecuali kelas *Ping Sweep*, yang memiliki *AUC* sebesar 0.99. Ini konsisten dengan hasil sebelumnya, di mana model mengalami sedikit kesulitan dalam

mendeteksi serangan *Ping Sweep* akibat jumlah data yang edikit.

rage dan *macro-average ROC curve* juga mem 1.00, yang menunjukkan bahwa secara keseluruhan, model sangat baik dalam membedakan antara serangan dan trafik normal. Garis diagonal putus-putus di tengah grafik merepresentasikan model random classifier, dan karena semua kurva ROC jauh di atas garis ini, model jauh lebih baik dibandingkan dengan tebakan acak.

Meskipun hasilnya sangat baik, ketidakseimbangan data antar kelas tetap menjadi faktor yang dapat mempengaruhi generalisasi model, terutama dalam kasus *Ping Sweep* yang memiliki *AUC* sedikit lebih rendah.



Gambar 4.43 *ROC Curve 3 Feature Multi Classification*

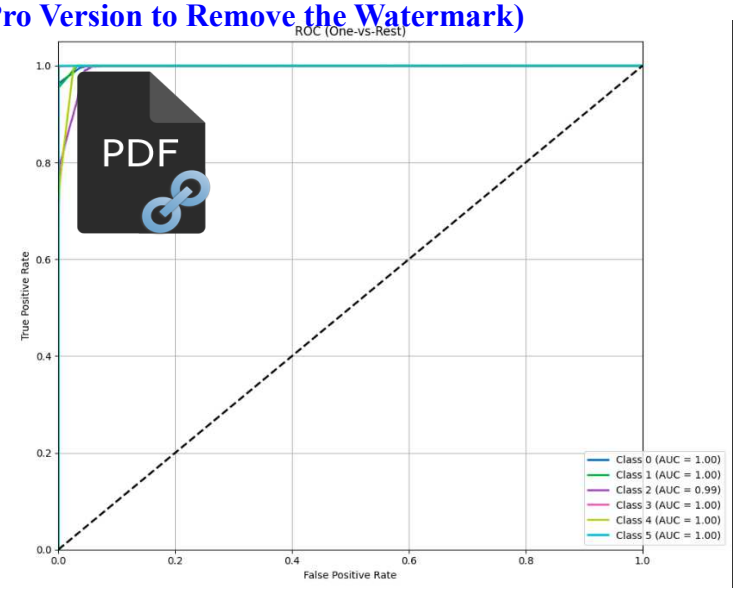
6) ROC (Receiver Operating Characteristic)

Gambar 4.44 menunjukkan *ROC* (*Receiver Operating Characteristic*) untuk skenario *One-vs-Rest*, di mana setiap kelas dibandingkan dengan semua kelas lainnya. Hampir semua kelas memiliki nilai *AUC* (*Area*

Under Curve) sebesar 1.00, kecuali kelas *Ping Sweep* yang [illegible]iki AUC sebesar 0.99. Hal ini menunjukkan bahw[illegible] memiliki kemampuan yang sangat baik dalam[illegible]dakan setiap kelas. Untuk meningkatkan performa model dalam mengenali kelas *Ping Sweep*, diperlukan peningkatan jumlah sampel agar model dapat lebih mengenali pola serangan tersebut dengan lebih akurat.

Meskipun kelas *Ping Sweep* memiliki nilai *recall* sebesar 0.58, nilai *AUC*-nya tetap tinggi. Hal ini disebabkan karena *AUC* mengukur keseimbangan antara *True Positive Rate* (*TPR*) dan *False Positive Rate* (*FPR*) di berbagai *threshold*. *Recall* yang rendah menunjukkan bahwa model gagal mendeteksi sebagian sampel dari kelas *Ping Sweep*, tetapi pada saat yang sama, model hampir tidak melakukan kesalahan dalam mengklasifikasikan sampel dari kelas lain sebagai *Ping Sweep*.

Kurva *ROC* yang mendekati sudut kiri atas grafik mengindikasikan bahwa model memiliki *False Positive Rate* yang sangat rendah. Hal ini sesuai dengan tabel *FPR/FNR* sebelumnya, di mana hampir semua kelas memiliki *FPR* mendekati nol. Dengan kata lain, model sangat jarang salah dalam mengklasifikasikan sampel dari kelas lain sebagai kelas tertentu.



Gambar 4.44 *ROC 3 Feature Multi Classification*

7) Precision Recall Curve

Gambar 4.45 menunjukkan hubungan antara *precision* dan *recall* untuk setiap kelas dalam model yang diuji. Sebagian besar kelas memiliki performa yang sangat baik, dengan area di bawah kurva (*AUC-PR*) mendekati 1.00, yang menunjukkan bahwa model mampu mempertahankan *precision* yang tinggi meskipun *recall* meningkat.

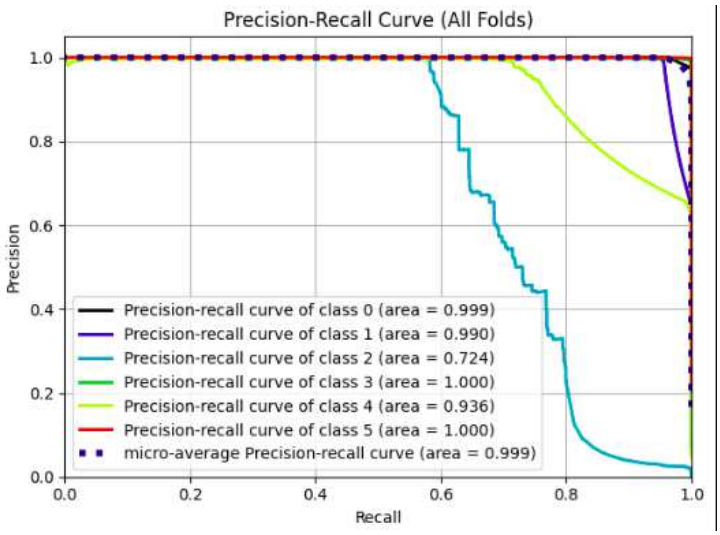
Namun, terdapat perbedaan yang cukup signifikan untuk kelas *Ping Sweep*, yang memiliki *AUC-PR* sebesar 0.724. Kurva yang lebih menurun dibandingkan kelas lain menunjukkan bahwa *precision* model pada kelas ini cepat menurun seiring bertambahnya *recall*. Hal ini menandakan bahwa meskipun model dapat mengidentifikasi beberapa *instance* dari kelas ini, tingkat kepercayaannya dalam prediksi tersebut lebih rendah dibandingkan dengan kelas lainnya.

Kelas lain, seperti Normal/*Benign*, *OS Scan*, dan *Host Discovery*, memiliki kurva Precision-Recall yang

sangat baik, dengan *precision* yang tetap tinggi mesk*call* meningkat. Hal ini sesuai dengan hasil evalu umnya, di mana model memiliki tingkat *Fals e Rate* yang rendah dan mampu mengklasifikasikan sebagian besar sampel dengan benar.

Penurunan performa pada kelas *Ping Sweep* disebabkan oleh jumlah data yang lebih sedikit dibandingkan kelas lainnya. Dengan jumlah sampel yang terbatas, model tidak dapat belajar pola yang cukup kuat untuk membedakan kelas ini dengan akurat. Hal ini menyebabkan *precision* menurun saat model mencoba mengidentifikasi lebih banyak *instance* dari kelas tersebut, yang terlihat dari turunnya kurva pada nilai recall yang lebih tinggi.

Secara keseluruhan, kurva ini mengonfirmasi bahwa model bekerja dengan sangat baik dalam mendeteksi sebagian besar kelas, tetapi masih memiliki keterbatasan dalam mengenali kelas dengan jumlah data yang lebih sedikit seperti *Ping Sweep.*



Gambar 4.45 *Precision Recall Curve 3 Feature Multi Classification*

8) Model Performance Matrix

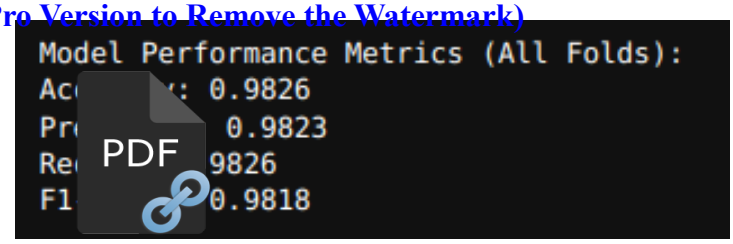
4.46 menampilkan *Model Performance Matr* menunjukkan performa yang sangat baik deng *curacy* sebesar 0.9826, *precision* 0.9823, *recall* 0.9826, dan *F1-score* 0.9818. Nilai *accuracy* yang tinggi mengindikasikan bahwa model mampu mengklasifikasikan sebagian besar sampel dengan benar.

Precision yang juga tinggi menunjukkan bahwa ketika model memprediksi suatu sampel sebagai serangan, kemungkinan besar prediksi tersebut benar. Hal ini sejalan dengan hasil sebelumnya yang menunjukkan bahwa sebagian besar kelas memiliki *False Positive Rate* (*FPR*) yang sangat rendah.

Recall yang tinggi mengindikasikan bahwa model berhasil menangkap sebagian besar sampel serangan dengan baik, meskipun ada sedikit penurunan *recall* pada kelas *Ping Sweep*. Hal ini berarti sebagian kecil *instance* dari kelas tersebut masih terlewat oleh model.

F1-score yang hampir setara dengan *precision* dan *recall* menunjukkan keseimbangan yang baik antara keduanya. Dengan kata lain, model tidak hanya mampu mengidentifikasi serangan dengan baik, tetapi juga minim dalam kesalahan prediksi.

Meskipun secara keseluruhan performa model sangat baik, evaluasi terhadap masing-masing kelas menunjukkan adanya kesulitan dalam mendeteksi kelas dengan jumlah data yang lebih sedikit, seperti *Ping Sweep*. Hal ini dapat menyebabkan *recall* yang lebih rendah pada kelas tersebut, meskipun dampaknya terhadap performa keseluruhan model relatif kecil.



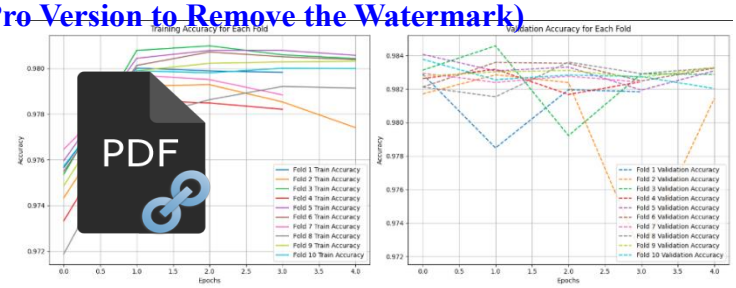
Gambar 4.46 *Model Performance Matrix 3 Feature Multi Classification*

9) Cross Validation

Gambar 4.47 menunjukkan akurasi selama proses pelatihan dan validasi untuk setiap *fold* dalam *cross-validation*. Pada grafik sebelah kiri, terlihat bahwa akurasi pelatihan meningkat dengan cepat dalam beberapa *epoch* pertama sebelum akhirnya stabil mendekati angka 0.98. Hal ini menunjukkan bahwa model mampu belajar dengan baik dari data latih tanpa mengalami *overfitting*.

Sementara itu, grafik di sebelah kanan memperlihatkan variasi akurasi validasi untuk masing-masing *fold*. Meskipun terdapat sedikit fluktuasi antar *fold*, akurasi validasi tetap berada dalam kisaran yang tinggi, berkisar antara 0.978 hingga 0.984. Fluktuasi ini menunjukkan adanya sedikit variasi performa model tergantung pada data yang digunakan dalam masing-masing *fold*, tetapi secara keseluruhan model tetap stabil dan konsisten.

Perbedaan kecil antara akurasi pelatihan dan validasi menunjukkan bahwa model tidak mengalami *overfitting*. Variasi antar *fold* dalam akurasi validasi bisa disebabkan oleh distribusi data yang tidak sepenuhnya merata atau adanya kelas yang lebih sulit untuk diprediksi dibandingkan yang lain.



Gambar 4.47 *Training and Validation Accuracy For Each Fold 3 Feature Multi Classification*

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini berhasil membuktikan efektivitas penggunaan *Autoencoder* dan *Convolutional Neural Network* (*CNN*) dalam mendeteksi serangan *Recon* pada jaringan *IoT* dengan berbagai konfigurasi fitur dan klasifikasi. *Autoencoder* digunakan untuk mengekstraksi data mentah yang berasal dari file *PCAP*, menghasilkan representasi *numerik* yang lebih sederhana dan *informatif*. Proses ini memungkinkan pengurangan kompleksitas data tanpa kehilangan informasi yang relevan dalam mendeteksi pola serangan.

Evaluasi model dilakukan pada beberapa skenario eksperimen. Untuk klasifikasi biner dengan dua fitur, model mencapai akurasi sebesar 96.59%, *precision* 93.15%, *recall* 98.31%, dan *F1-score* 95.66%. Sementara itu, pada klasifikasi multi-kelas dengan dua fitur, performa model mengalami sedikit penurunan dengan akurasi sebesar 92.02%, *precision* 94.02%, *recall* 92.02%, dan *F1-score* 92.29%.

Ketika jumlah fitur yang digunakan ditingkatkan menjadi tiga, model klasifikasi biner menunjukkan peningkatan performa dengan akurasi mencapai 98.37%, *precision* 99.51%, *recall* 96.20%, dan *F1-score* 97.83%. Sementara itu, pada klasifikasi multi-kelas dengan tiga fitur, model menunjukkan performa terbaik dengan akurasi 98.26%, *precision* 98.23%, *recall* 98.26%, dan *F1-score* 98.18%.

Selain itu, hasil evaluasi menggunakan *Precision-Recall Curve* dan *ROC Curve* menunjukkan bahwa model memiliki kemampuan yang sangat baik dalam membedakan antara serangan dan data normal. Dengan *AUC* yang mendekati 1.00 pada semua skenario, model mampu memisahkan kelas dengan sangat baik. Hasil pengujian dengan metode *cross-validation* juga menunjukkan bahwa performa model tetap stabil, dengan akurasi *training* dan validasi yang konsisten.

Secara keseluruhan, penelitian ini membuktikan bahwa kombinasi *Autoencoder* dan *CNN* adalah pendekatan yang efektif untuk mendeteksi

serangan *Recon* pada jaringan *IoT*. Hasil penelitian ini tidak hanya memberikan kontribusi t dalam bidang keamanan jaringan berbasis *deep learning*, tetapi juga rkan solusi praktis yang dapat diterapkan dalam meningkatkan kea a jaringan IoT.

5.2 Saran

Penelitian ini menyarankan agar model yang telah dikembangkan diuji lebih lanjut pada jaringan IoT nyata untuk mengukur performanya dalam lingkungan produksi. Implementasi langsung pada jaringan nyata akan memberikan wawasan tambahan mengenai efektivitas model dalam mendeteksi serangan pada skenario dunia nyata, serta mengidentifikasi potensi tantangan dan perbaikan yang diperlukan untuk pengembangan lebih lanjut.

DAFTAR PUSTAKA

[1] E. Susanto, Lady Antira, , E. Stanzah, and A. A. Majid, "Manajemen Keamanan Era Digital," *J. Bus. Entrep.*, vol. 11, no. 1, p. 23, 2023, doi: 10.46273/jobe.v11i1.365.

[2] H. Fitriawan, D. Despa, and I. Kustiani, "Potensi Internet of Things (IoT) dan Ragam Sensor untuk Layanan Kesehatan," *J. Profesi Ins. Univ. Lampung*, vol. 1, no. 1, pp. 1–4, 2020, doi: 10.23960/jpi.v1n1.10.

[3] Satyajit Sinha, "State of IoT 2024: Number of connected IoT devices growing 13% to 18.8 billion globally," IoT Analytics. Accessed: Dec. 08, 2024. [Online]. Available: https://iot-analytics.com/number-connected-iot-devices/

[4] Nuroji, "Penerapan Intrusion Detection andPrevention System (IDPS) padaJaringan komputer sebagaipencegahan serangan Port-Scanning," *J. Data Sci. Inf. Syst.*, vol. 1, no. 2, pp. 41–49, 2023, [Online]. Available: https://doi.org/10.58602/dimis.v1i2.35

[5] R. Rizal, N. Widiyasono, and S. Yuliyanti, "Kecerdasan Buatan untuk Klasifikasi Serangan Siber pada Internet of Things Network Traffic," *JUMANJI (Jurnal Masy. Inform. Unjani)*, vol. 7, no. 2, pp. 61–71, 2023, [Online]. Available: https://jumanji.unjani.ac.id/index.php/jumanji/article/view/325

[6] R. P. Simanjuntak and R. R. M. Sijabat, "Meningkatkan Keamanan Siber dalam Lingkungan Internet of Things (IoT) dengan Menggunakan Sistem Deteksi Intrusi Berbasis Pembelajaran Mesin," *DIKE J. Ilmu Multidisiplin*, vol. 2, no. 2, pp. 62–68, 2024, doi: 10.69688/dike.v2i2.106.

[7] Y. N. Kunang, S. Nurmaini, D. Stiawan, A. Zarkasi, and F. Jasmir, "Automatic Features Extraction Using Autoencoder in Intrusion Detection System," *Proc. 2018 Int. Conf. Electr. Eng. Comput. Sci. ICECOS 2018*,

no. December, pp. 219–224, 2019, doi: 10.1109/ICECOS.2018.8605181.

[8] I. Magdalena, T. Sundari, S. Namilah, D. Ayu Amalia, and U. Muhammadiyah Tangera[illegible]isis Bahan Ajar," *J. Pendidik. dan Ilmu Sos.*, vol. 2, no. 2, pp. 31[illegible]20, [Online]. Available: https://ejournal.stitpn.ac.id/index.php/nusantara

[9] F. Ertam, "Data classification with deep learning using tensorflow," *2nd Int. Conf. Comput. Sci. Eng. UBMK 2017*, pp. 755–758, 2017, doi: 10.1109/UBMK.2017.8093521.

[10] K. Singh, R. Scholar, A. Mahajan, and V. Mansotra, "1D-CNN based Model for Classification and Analysis of Network Attacks," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 11, pp. 604–613, 2021, doi: 10.14569/IJACSA.2021.0121169.

[11] K. Azmi, S. Defit, and S. Sumijan, "Implementasi Convolutional Neural Network (CNN) Untuk Klasifikasi Batik Tanah Liat Sumatera Barat," *J. Unitek*, vol. 16, no. 1, pp. 28–40, 2023, doi: 10.52072/unitek.v16i1.504.

[12] F. A. Febriyanti, "Image Processing Dengan Metode Convolutional Neural Network (Cnn) Untuk Deteksi Penyakit Kulit Pada Manusia," *Kohesi J. Sains Dan Teknol.*, vol. 3, no. 10, pp. 21–30, 2024, [Online]. Available: https://ejournal.warunayama.org/kohesi

[13] I. Bakti and M. Firdaus, "Klasifikasi File Gambar Hasil X-Ray Paru -Paru Dengan Arsitektur Convolution Neural Network (CNN)," *Jifotech (Journal Inf. Technol.*, vol. 3, no. 1, pp. 26–34, 2023.

[14] I. Salehin and D. K. Kang, "A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain," *Electron.*, vol. 12, no. 14, 2023, doi: 10.3390/electronics12143106.

[15] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Comput. Oper. Res.*, vol. 152, no. April 2022, p. 106131, 2023, doi:

10.1016/j.cor.2022.106131.

[16] S. Susanto, M. A. S. Ari[illegible] O. L. Wijaya, "IoT Botnet Detection Using Autoencoders and [illegible]Trees," *J. Sisfokom (Sistem Inf. dan Komputer)*, vol. 12, no. 3, pp. [illegible]–334, 2023, doi: 10.32736/sisfokom.v12i3.1693.

[17] Angelina M. T. I. Sambi Ua *et al.*, "Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-Paru," *J. Publ. Tek. Inform.*, vol. 2, no. 2, pp. 88–99, 2023, doi: 10.55606/jupti.v2i2.1742.

[18] J. Sardi, H. Hamdani, and V. B. Pramuja, "Aplikasi Pengukuran Berat dan Tinggi Badan Anak Balita Menggunakan Metode Radbms Berbasis Python," *JTEIN J. Tek. Elektro Indones.*, vol. 2, no. 1, pp. 71–79, 2021, doi: 10.24036/jtein.v2i1.130.

[19] M. M. Alani, "Detection of Reconnaissance Attacks on IoT Devices Using Deep Neural Networks BT - Advances in Nature-Inspired Cyber Security and Resilience," S. K. Shandilya, N. Wagner, V. B. Gupta, and A. K. Nagar, Eds., Cham: Springer International Publishing, 2022, pp. 9–27. doi: 10.1007/978-3-030-90708-2_2.

[20] M. M. Alani and E. Damiani, "XRecon: An Explainbale IoT Reconnaissance Attack Detection System Based on Ensemble Learning," *Sensors*, vol. 23, no. 11, pp. 1–26, 2023, doi: 10.3390/s23115298.

[21] H. N. Viet, L. L. T. Trang, Q. Nguyen Van, and S. Nathan, "Using deep learning model for network scanning detection," *ACM Int. Conf. Proceeding Ser.*, pp. 117–121, 2018, doi: 10.1145/3233347.3233379.

[22] C. L. Mindara, A. Zulianto, H. P. Utomo, T. Hatati, and D. Sudrajat, "Convolutional Neural Network Deteksi Intrusi Untuk Klasifikasi Serangan Jaringan Dengan Penerapan Algoritma Convolutional Neural Network," *J. ICT Inf. Commun. Technol.*, vol. 23, no. 2, pp. 517–522, 2023, [Online]. Available: https://ejournal.ikmi.ac.id/index.php/jict-ikmi/article/view/128

[23] D. Stiawan, Susanto, A. Bimantara, M. Y. Idris, and R. Budiarto, "IoT botnet attack detection u[illegible]p autoencoder and artificial neural networks," *KSII Trans. I[illegible]f. Syst.*, vol. 17, no. 5, pp. 1310–1338, 2023, doi: 10.3837/tiis.2[illegible]1.

[24] Webometrics, "Ranking Web Of Universities," Webometrics. Accessed: Jan. 27, 2025. [Online]. Available: https://www.webometrics.info/en/detalles/NoRoR25193

LAMPIRAN

Lampiran 1. *Lembar pengesah*

UNIVERSITAS BINA INSAN

Jalan Jendral Besar H.M. Soeharto KM.13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

Formulir Pengajuan Judul Skripsi
Program Studi Informatika

Nama	: Krisna Rizki Pratama
NIM	: 2102020105
Alamat	: JL Selamat, RT 02, Kel Sukajadi, Kec Lubuklinggau Barat I
No.Hp	: 082281258929

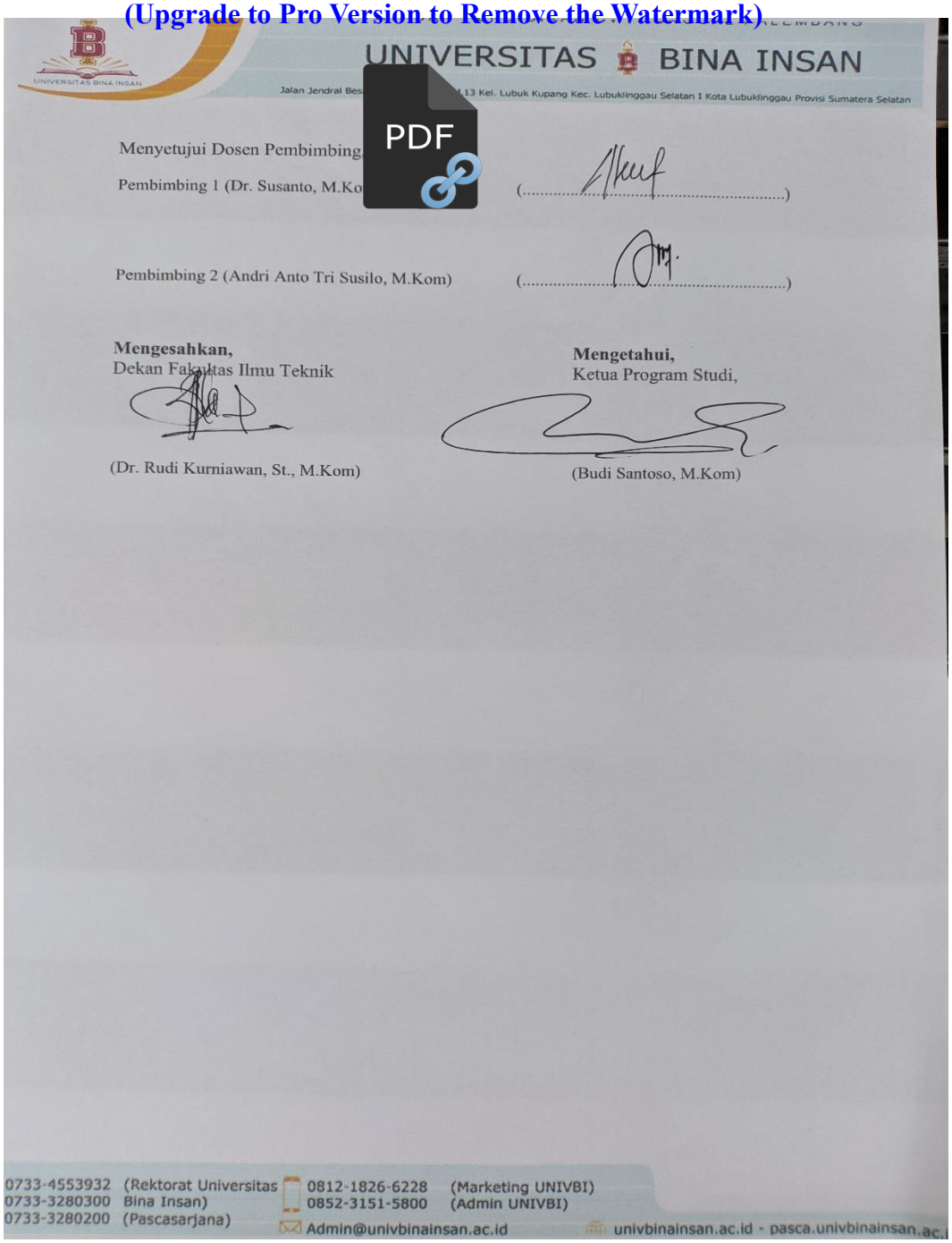
Rumusan Masalah 1	: Apakah penerapan Progressive Web Apps (PWAs) secara signifikan meningkatkan performa aplikasi web dalam hal metrik LCP (Largest Contentful Paint), FID (First Input Delay), dan CLS (Cumulative Layout Shift)?
Judul 1	: "Pengembangan dan Perbandingan Performa Aplikasi Web dengan dan tanpa Progressive Web Apps (PWAs) Menggunakan Metrik LCP, FID, dan CLS."
Rumusan Masalah 2	: Bagaimana penerapan teknologi Progressive Web App (PWA) dan integrasi RESTful API menggunakan Hapi.js dapat meningkatkan aksesibilitas, pengalaman pengguna, serta efisiensi pengelolaan data perpustakaan secara real-time di Universitas Bina Insan, sekaligus menggantikan peran aplikasi native dalam memberikan kemudahan akses layanan, baik secara online maupun offline?
Judul 2	: "Pengembangan Sistem Informasi Perpustakaan Berbasis Progressive Web App (PWA) dengan RESTful API Menggunakan Hapi.js (Studi Kasus: Universitas Bina Insan)."
Rumusan Masalah 3	: Bagaimana mengembangkan model Convolutional Neural Network (CNN) yang mampu mendeteksi adanya serangan Recon pada data jaringan dengan menggabungkan berbagai jenis serangan Recon dalam satu model deteksi?
Judul 3	: "Analisis Serangan Recon Menggunakan CNN pada Jaringan IoT."

Diusulkan Judul Nomor :1(satu)/ 2(Dua)/ 3(Tiga)*

Lubuklinggau, 21 November 2024
Mahasiswa yang mengusulkan,

(Krisna Rizki Pratama)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac



UNIVERSITAS BINA INSAN

Jalan Jendral Bes .13 Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

Menyetujui Dosen Pembimbing

Pembimbing 1 (Dr. Susanto, M.Ko (.....)

Pembimbing 2 (Andri Anto Tri Susilo, M.Kom) (.....)

Mengesahkan,
Dekan Fakultas Ilmu Teknik

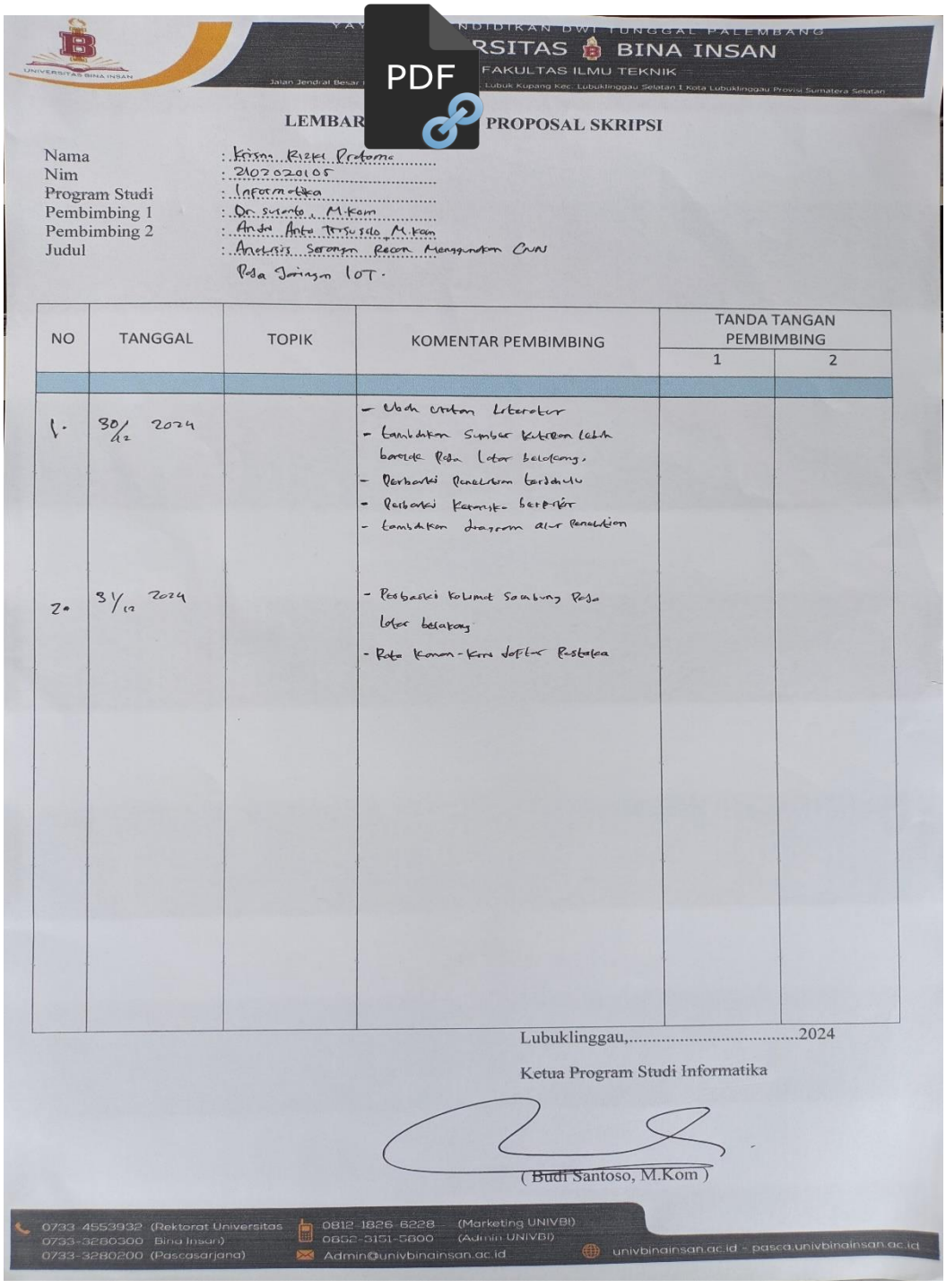
(Dr. Rudi Kurniawan, St., M.Kom)

Mengetahui,
Ketua Program Studi,

(Budi Santoso, M.Kom)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.i

***Lampiran 2**. Lembar bimbingan proposal p1*



YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar ... Lubuk Kupang Kec. Lubuklinggau Selatan 1 Kota Lubuklinggau Provinsi Sumatera Selatan

LEMBAR ... PROPOSAL SKRIPSI

Nama : Krisna Rizki Pratama
Nim : 2102020105
Program Studi : Informatika
Pembimbing 1 : Dr. Suranto, M.Kom
Pembimbing 2 : Andri Anto Tristianto M.Kom
Judul : Analisis Serangan Recon Menggunakan CVSS Pada Jaringan IoT.

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
1.	30/12 2024		- Ubah urutan literatur - tambahkan sumber kutipan lebih banyak pada latar belakang. - Perbanyak penelitian terdahulu - Perbaiki kerangka berpikir - tambahkan diagram alur penelitian		
2.	31/12 2024		- Perbaiki kalimat sambung pada latar belakang - Rata kanan-kiri daftar pustaka		

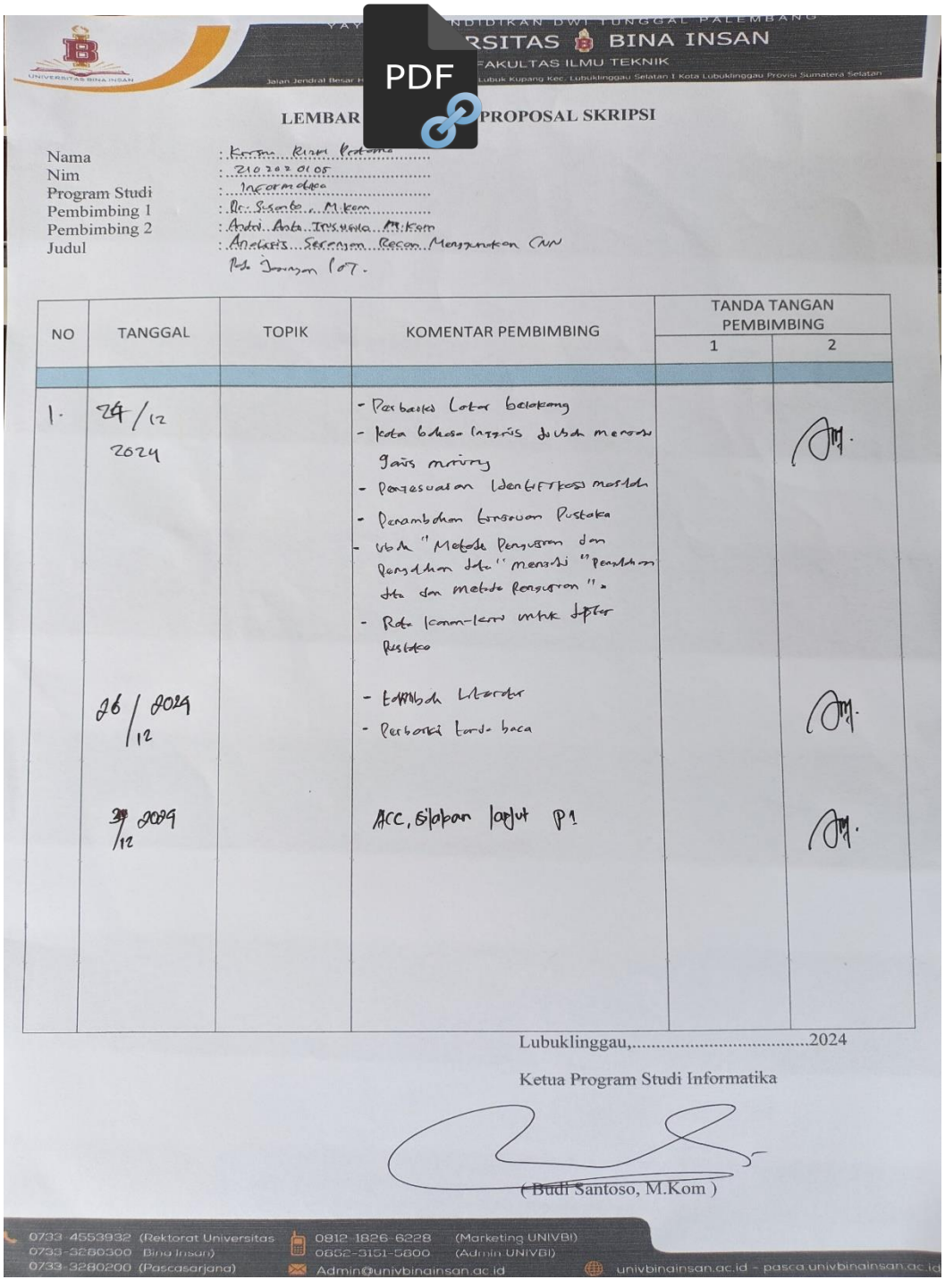
Lubuklinggau,.....2024

Ketua Program Studi Informatika

(Budi Santoso, M.Kom)

0733 4553932 (Rektorat Universitas Bina Insani)
0733-3280300
0733-3280200 (Pascasarjana)
0812 1826 6228 (Marketing UNIVBI)
0852-3151-5600 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.id

***Lampiran 3**. Lembar bimbingan proposal p2*



YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar H ... Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provinsi Sumatera Selatan

LEMBAR ... PROPOSAL SKRIPSI

Nama : Karim Rizki Pratama
Nim : 210202 01105
Program Studi : Informatika
Pembimbing 1 : Ir. Sigit ... M.Kom
Pembimbing 2 : Andi Ardi Irsyana, M.Kom
Judul : Analisis Sentimen Pada Mengunakan CNN Pada Jaringan IoT.

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING 1	TANDA TANGAN PEMBIMBING 2
1.	24/12 2024		- Perbaiki Latar belakang - kata bahasa inggris di ubah menjadi garis miring - Penyesuaian identifikasi masalah - Penambahan kerangka Pustaka - ubah "Metode Pengujian dan Pengolahan data" menjadi "pengolahan data dan metode pengujian" - Ada komen-lain untuk daftar Pustaka		
	26/12 2024		- tambah Literatur - Perbaiki tanda baca		
	30/12 2024		ACC, silakan lanjut P1		

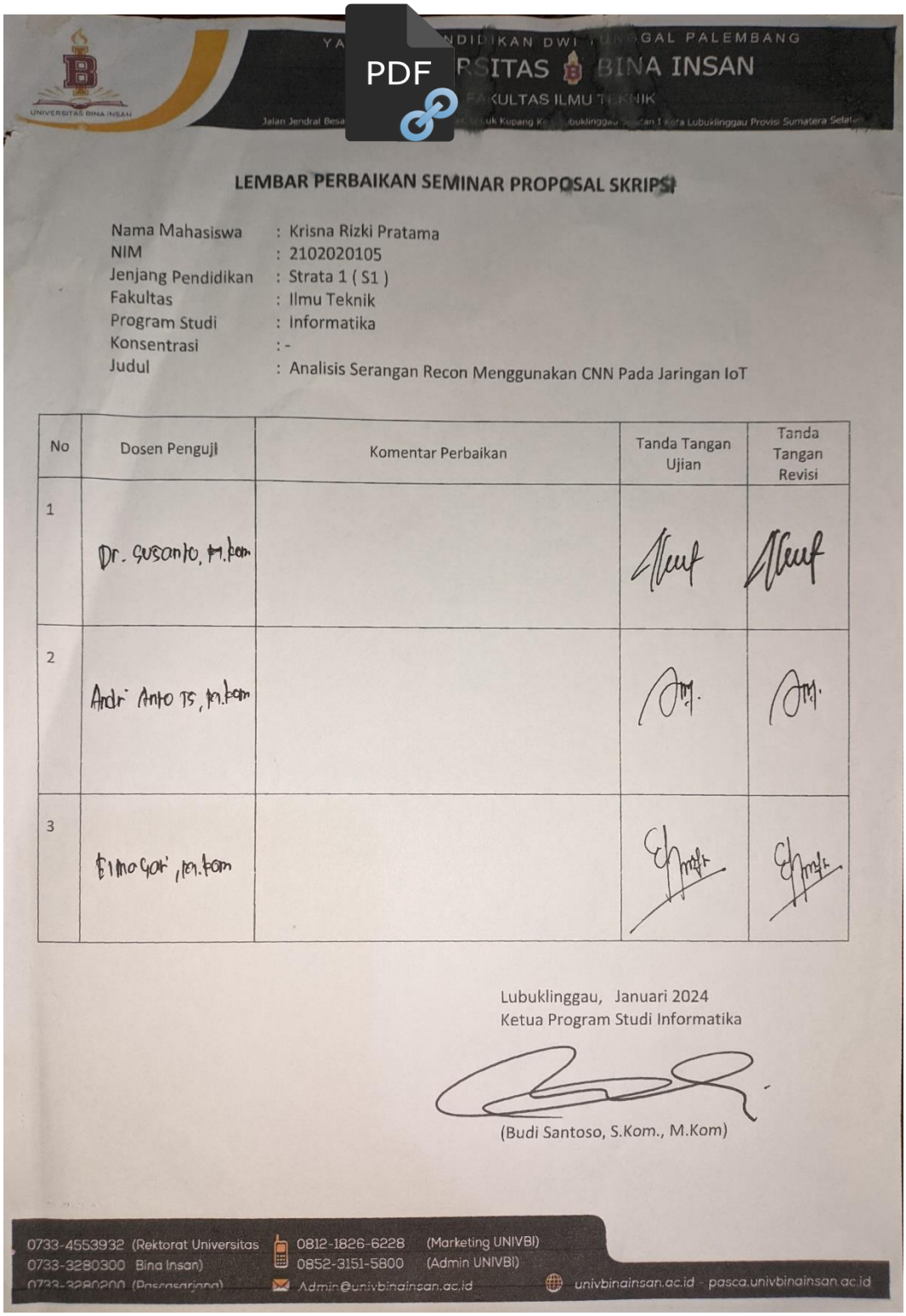
Lubuklinggau,.....2024

Ketua Program Studi Informatika

(Budi Santoso, M.Kom)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.id

Lampiran 4. *Lembar perbaikan seminar proposal skripsi*



UNIVERSITAS BINA INSAN

Jalan Jendral Besa ... Kupang K... Lubuklinggau ... Kota Lubuklinggau Provinsi Sumatera Selatan

LEMBAR PERBAIKAN SEMINAR PROPOSAL SKRIPSI

Nama Mahasiswa	: Krisna Rizki Pratama
NIM	: 2102020105
Jenjang Pendidikan	: Strata 1 (S1)
Fakultas	: Ilmu Teknik
Program Studi	: Informatika
Konsentrasi	: -
Judul	: Analisis Serangan Recon Menggunakan CNN Pada Jaringan IoT

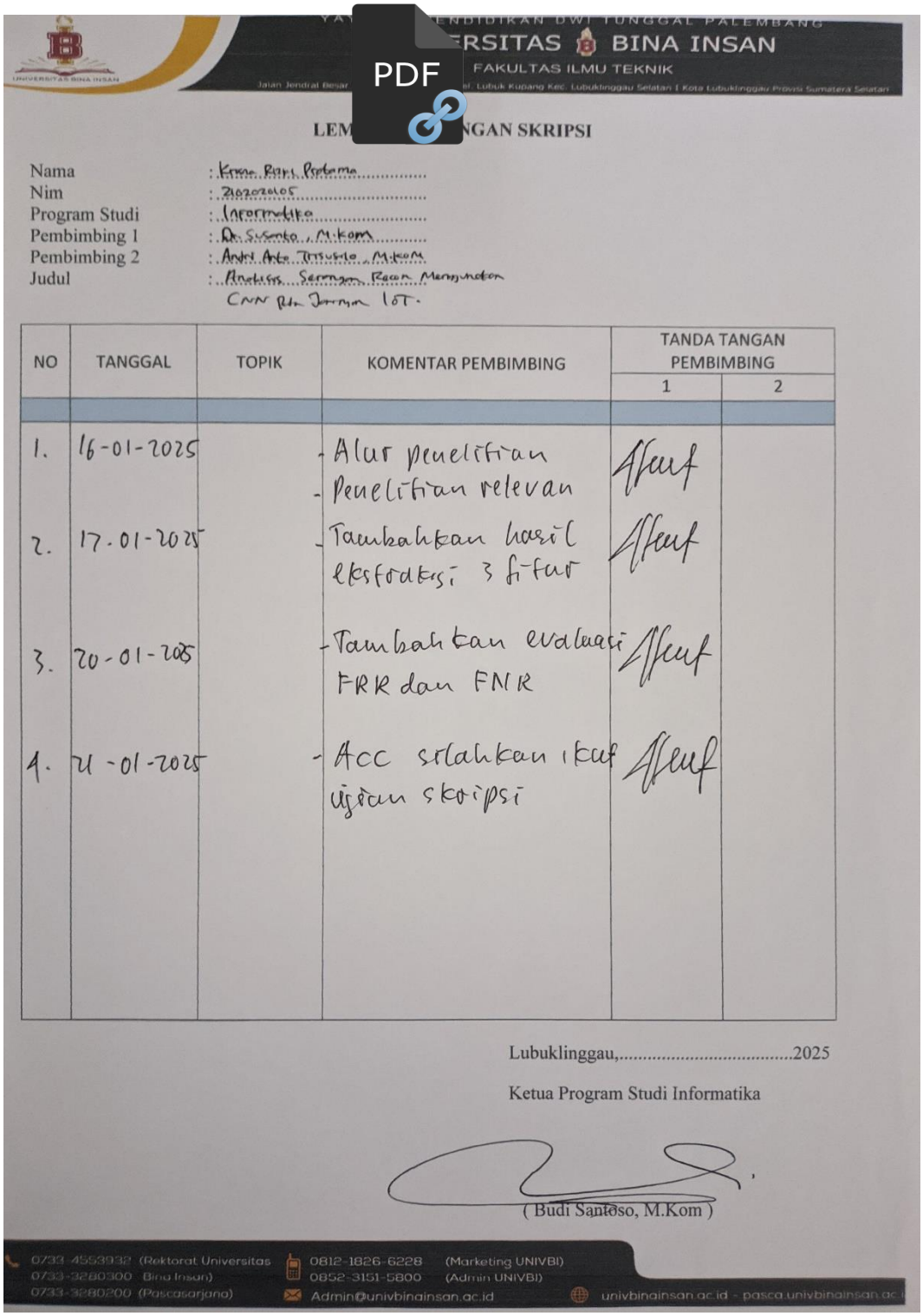
No	Dosen Penguji	Komentar Perbaikan	Tanda Tangan Ujian	Tanda Tangan Revisi
1	Dr. Susanto, M.Kom			
2	Andri Anto TS, M.Kom			
3	Bima Gori, M.Kom			

Lubuklinggau, Januari 2024
Ketua Program Studi Informatika

(Budi Santoso, S.Kom., M.Kom)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.id

***Lampiran 5**. Lembar bimbingan skripsi p1*



YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG

UNIVERSITAS BINA INSAN

FAKULTAS ILMU TEKNIK

Jalan Jendral Besar ... Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provinsi Sumatera Selatan

LEMBAR BIMBINGAN SKRIPSI

Nama : Kresna Bayu Pratama
Nim : 2102026105
Program Studi : Informatika
Pembimbing 1 : Ir. Susanto, M.Kom
Pembimbing 2 : Andri Arta Tristanto, M.Kom
Judul : Analisis Serangan Rasa Menyimpan CNN Pada Jaringan IoT.

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
1.	16-01-2025		- Alur penelitian - Penelitian relevan		
2.	17-01-2025		- Tambahkan hasil ekstraksi 3 fitur		
3.	20-01-205		- Tambahkan evaluasi FRR dan FNR		
4.	21-01-2025		- Acc silahkan ikut ujian skripsi		

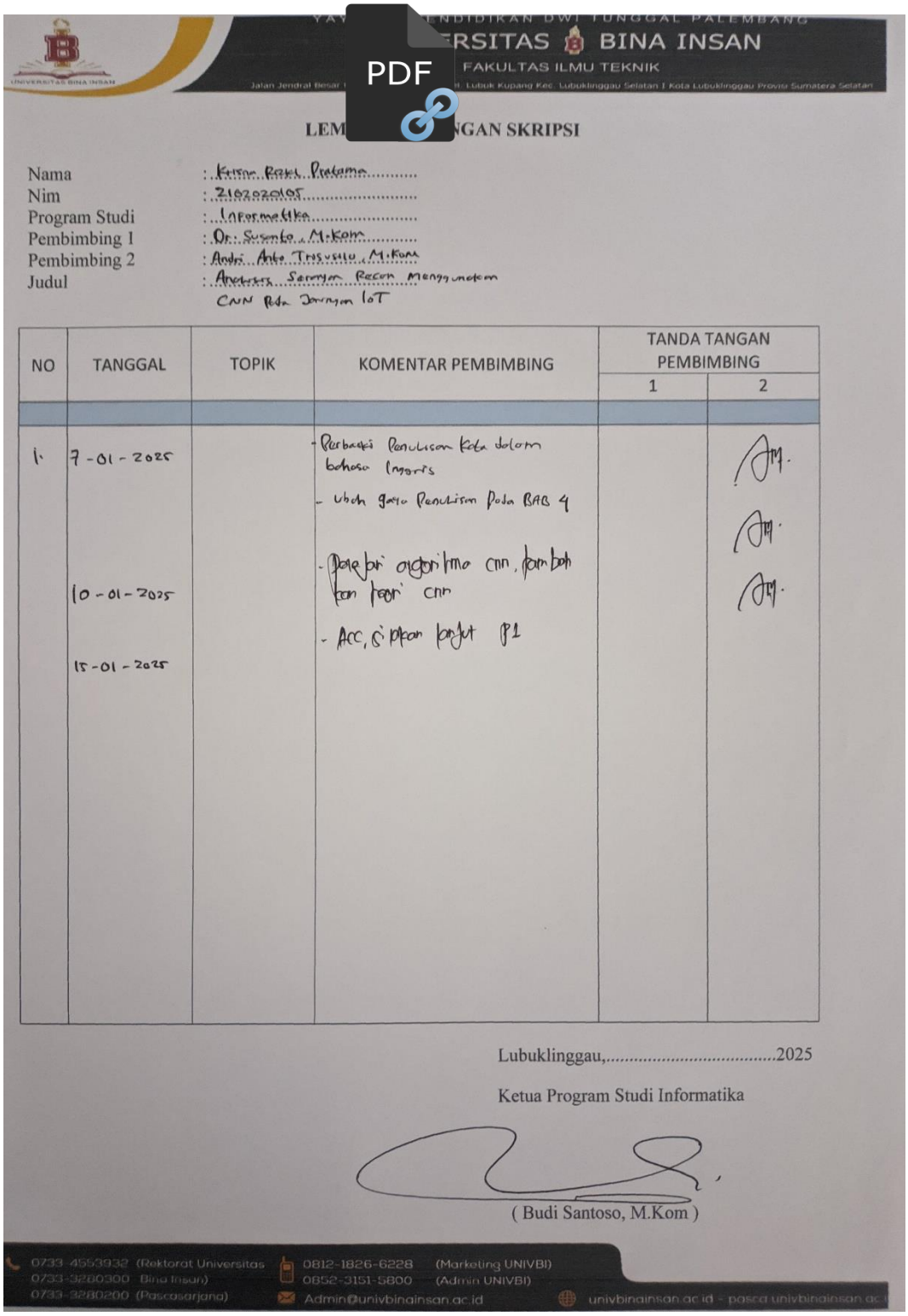
Lubuklinggau,.....2025

Ketua Program Studi Informatika

(Budi Santoso, M.Kom)

0733-4553932 (Rektorat Universitas Bina Insan) 0733-3280300 0733-3280200 (Pascasarjana) 0812-1826-6228 (Marketing UNIVBI) 0852-3151-5800 (Admin UNIVBI) Admin@univbinainsan.ac.id univbinainsan.ac.id - pasca.univbinainsan.ac.i

***Lampiran 6**. Lembar bimbingan skripsi p2*



YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar ... Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provinsi Sumatera Selatan

LEM...NGAN SKRIPSI

Nama : Kiram Rahel Pratama
Nim : 2102020105
Program Studi : Informatika
Pembimbing 1 : Dr. Susanto, M.Kom
Pembimbing 2 : Andri Ante Tristanto, M.Kom
Judul : Analisis Serangan Recon Menggunakan CNN Pada Jaringan IoT

NO	TANGGAL	TOPIK	KOMENTAR PEMBIMBING	TANDA TANGAN PEMBIMBING	
				1	2
1.	7-01-2025		- Perbaiki Penulisan kata dalam bahasa Inggris - Ubah gaya Penulisan Pada BAB 4		
	10-01-2025		- Perjelas algoritma CNN, tambah teori CNN		
	15-01-2025		- ACC, lanjut BAB 1		

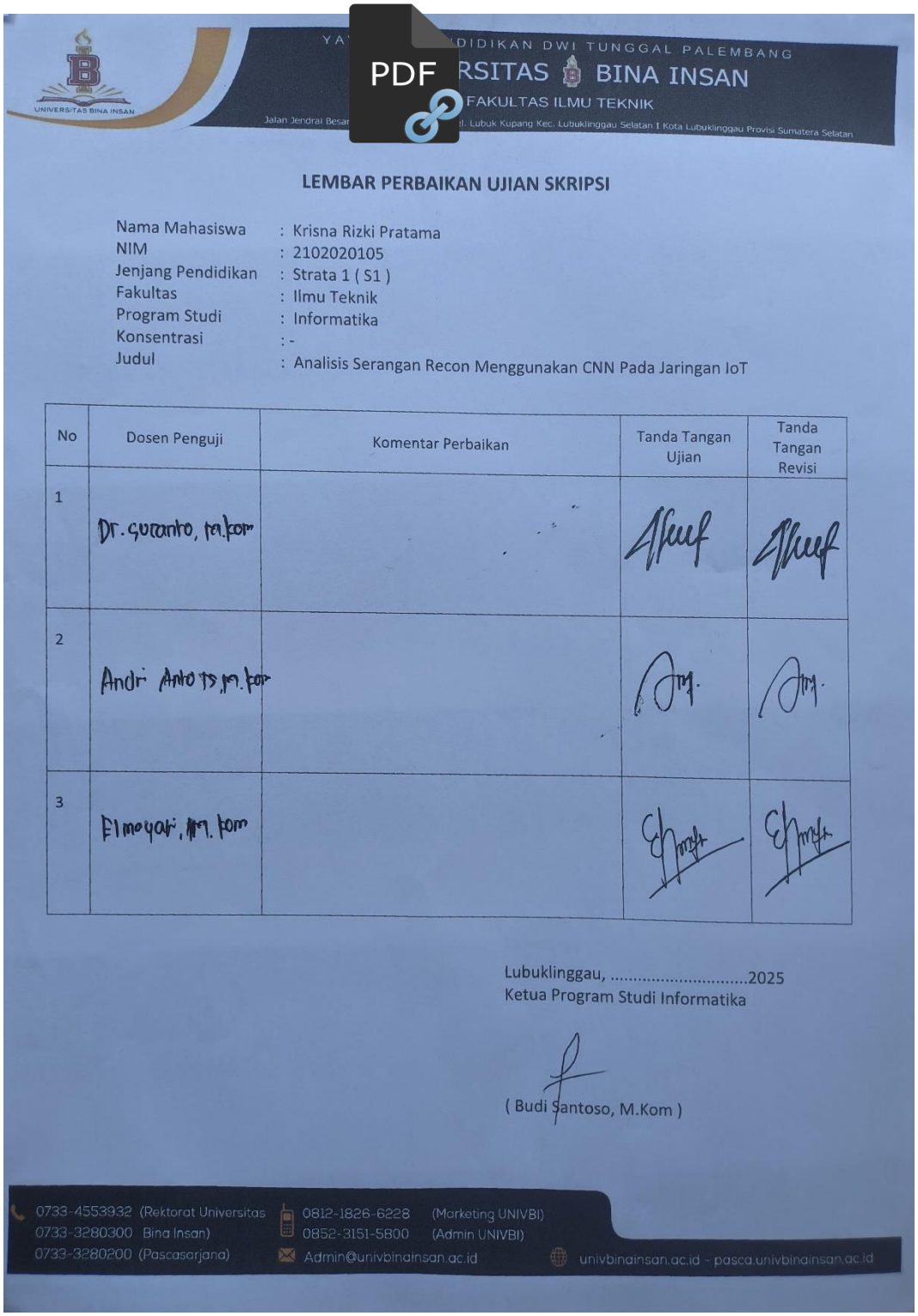
Lubuklinggau,.....2025

Ketua Program Studi Informatika

(Budi Santoso, M.Kom)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.id

***Lampiran** 7. Lembar perbaikan skripsi*



YAYASAN PENDIDIKAN DWI TUNGGAL PALEMBANG
UNIVERSITAS BINA INSAN
FAKULTAS ILMU TEKNIK
Jalan Jendral Besar ... Kel. Lubuk Kupang Kec. Lubuklinggau Selatan I Kota Lubuklinggau Provisi Sumatera Selatan

LEMBAR PERBAIKAN UJIAN SKRIPSI

Nama Mahasiswa	: Krisna Rizki Pratama
NIM	: 2102020105
Jenjang Pendidikan	: Strata 1 (S1)
Fakultas	: Ilmu Teknik
Program Studi	: Informatika
Konsentrasi	: -
Judul	: Analisis Serangan Recon Menggunakan CNN Pada Jaringan IoT

No	Dosen Penguji	Komentar Perbaikan	Tanda Tangan Ujian	Tanda Tangan Revisi
1	Dr. Suranto, M.Kom			
2	Andri Antoni, M.Kom			
3	Elmoyati, M.Kom			

Lubuklinggau, ...2025
Ketua Program Studi Informatika

(Budi Santoso, M.Kom)

0733-4553932 (Rektorat Universitas Bina Insan)
0733-3280300
0733-3280200 (Pascasarjana)
0812-1826-6228 (Marketing UNIVBI)
0852-3151-5800 (Admin UNIVBI)
Admin@univbinainsan.ac.id
univbinainsan.ac.id - pasca.univbinainsan.ac.id