TUGAS AKHIR - TF 181801

PERANCANGAN *MACHINE LEARNING* SISTEM DETEKSI INTRUSI PADA SISTEM KONTROL INDUSTRI DENGAN MODEL *NEURAL NETWORK*

MOCHAMMAD HAIDAR DZAKALAKSANA
NRP. 02311740000116

Dosen Pembimbing:
Dr. Bambang Lelono Widjiantoro, S.T., M.T.
Moh. Kamalul Wafi, S.T., M.Sc.DIC

DEPARTEMEN TEKNIK FISIKA
Fakultas Teknologi Industri dan Rekayasa Sistem
Institut Teknologi Sepuluh Nopember
Surabaya 2022

Halaman ini sengaja dikosongkan

FINAL PROJECT - TF 181801

DESIGN OF MACHINE LEARNING INTRUSION DETECTION SYSTEM IN INDUSTRIAL CONTROL SYSTEM WITH NEURAL NETWORK MODEL

MOCHAMMAD HAIDAR DZAKALAKSANA
NRP. 02311740000116

Supervisors:
Dr. Bambang Lelono Widjiantoro, S.T., M.T.
Moh. Kamalul Wafi, S.T., M.Sc.DIC

DEPARTMENT OF ENGINEERING PHYSICS
Faculty of Industrial Technology and System Engineering
Institut Teknologi Sepuluh Nopember
Surabaya 2022

Halaman ini sengaja dikosongkan

PERNYATAAN BEBAS PLAGIASI

Saya yang bertanda tangan di bawah ini.

Nama	: Mochammad Haidar Dzakalaksana
NRP	: 02311740000116
Departemen / Prodi	: Teknik Fisika / S1 Teknik Fisika
Fakultas	: Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)
Perguruan Tinggi	: Institut Teknologi Sepuluh Nopember

Dengan ini menyatakan bahwa Tugas Akhir dengan judul "PERANCANGAN *MACHINE LEARNING* SISTEM DETEKSI INTRUSI PADA SISTEM KONTROL INDUSTRI DENGAN MODEL *NEURAL NETWORK*" adalah benar karya saya sendiri dan bukan plagiat dari karya orang lain. Apabila di kemudian hari terbukti terdapat plagiat pada Tugas Akhir ini, maka saya bersedia menerima sanksi sesuai ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sebenarnya-benarnya.

Surabaya, 9 Februari 2022

Yang membuat pernyataan,

Mochammad Haidar Dzakalaksana

NRP. 02311740000116

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

TUGAS AKHIR

PERANCANGAN *MACHINE LEARNING* SISTEM DETEKSI INTRUSI PADA SISTEM KONTROL INDUSTRI DENGAN MODEL *NEURAL NETWORK*

Oleh:

Mochammad Haidar Dzakalaksana

NRP. 02311740000116

Surabaya,

Menyetujui,	**Menyetujui,**
Pembimbing I	**Pembimbing II**
Dr. Bambang L. Widjiantoro, S.T., M.T.	**Moh. Kamalul Wafi, S.T., M.Sc.DIC.**
NIP. 19690507 1995121 001	**NUP. 9900008686**

Mengetahui,

Kepala Departemen

Teknik Fisika FTIRS-ITS

KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI INSTITUT TEKNOLOGI SEPULUH NOPEMBER FAKULTAS TEKNOLOGI INDUSTRI DAN REKAYASA SISTEM DEPARTEMEN TEKNIK FISIKA

Digitally signed by Dr. Suyanto, S.T., M.T.
Date: 2022.02.15 09:38:37 +07'00'

Dr. Suyanto, S.T., M.T.

NIP. 197111319951002

Halaman ini sengaja dikosongkan

LEMBAR PENGESAHAN

PERANCANGAN *MACHINE LEARNING* SISTEM DETEKSI INTRUSI PADA SISTEM KONTROL INDUSTRI DENGAN MODEL *NEURAL NETWORK*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat

Memperoleh Gelar Sarjana Teknik

pada

Progam Studi S-1 Departemen Teknik Fisika

Fakultas Teknologi Industri & Rekayasa Sistem (FTIRS)

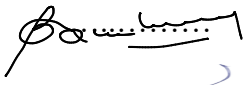
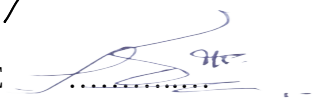
Institut Teknologi Sepuluh Nopember

Oleh:

MOCHAMMAD HAIDAR DZAKALAKSANA

NRP. 02311740000116

Disetujui oleh Tim Penguji Tugas Akhir:

1. Dr. Bambang Lelono W., S.T., M.T.  (Pembimbing I)
2. Moh. Kamalul Wafi, S.T., M.Sc.DIC  (Pembimbing II)
3. Dr. Katherin Indriawati, S.T., M.T.  (Ketua Penguji)
4. Dyah Sawitri, S.T., M.T.  (Penguji I)

SURABAYA

2022

Halaman ini sengaja dikosongkan

PERANCANGAN *MACHINE LEARNING* SISTEM DETEKSI INTRUSI PADA SISTEM KONTROL INDUSTRI DENGAN MODEL *NEURAL NETWORK*

Nama	**:**	**Mochammad Haidar Dzakalaksana**
NRP	**:**	**02311740000116**
Departemen	**:**	**Teknik Fisika FTIRS - ITS**
Dosen Pembimbing	**:**	**Dr. Bambang Lelono Widjiantoro, S.T, M.T**
		Moh Kamalul Wafi, S.T., M.Sc.DIC

ABSTRAK

Sistem kontrol industri adalah istilah umum yang mencakup beberapa jenis sistem kontrol dan komponen terikat yang digunakan untuk kontrol proses industri. Dengan peningkatannya integrasi dengan computer, sistem kontrol industri menjadi lebih terbuka. Oleh karena itu, penulis merancang sebuah *machine learning* sistem deteksi intrusi ini agar dapat mendeteksi serangan masuk pada sistem kontrol industri. Pada tugas akhir ini mengimplementasikan algoritma *binary classification* dalam merancang sebuah sistem deteksi intrusi dengan *python* sebagai *software* pendukungnya. *Dataset* yang diambil berasal dari *dataset power system attacks* di Missisipi State University. Pembuatan model machine learning menggunakan *multi layer perceptron* dengan algoritma pembelajaran *backpropagation*. Hasil dari tugas akhir ini dapat menerapkan *neural networks* sebagai cara untuk mendeteksi serangan sebuah serangan. Setelah dilakukan pelatihan dengan variasi 5 hidden layer didapatkan nilai keakuratan terbesar sebesar 72,3 %. Dengan nilai rata-rata *precision* sebesar 0,71, *recall* sebesar 0,72, dan *f1-score* sebesar 0,60.

Kata Kunci: Sistem kontrol industri, sistem deteksi intrusi, *machine learning, binary classification, multi layer perceptron, backpropagation, neural networks.*

Halaman ini sengaja dikosongkan

DESIGN OF MACHINE LEARNING INTRUSION DETECTION SYSTEM IN INDUSTRIAL CONTROL SYSTEM WITH NEURAL NETWORK MODEL

Name	**:**	**Mochammad Haidar Dzakalaksana**
NRP	**:**	**02311740000116**
Department	**:**	**Engineering Physics FTIRS - ITS**
Supervisors	**:**	**Dr. Bambang Lelono Widjiantoro, S.T, M.T**
		Moh Kamalul Wafi, S.T., M.Sc.DIC

ABSTRACT

Industrial control system is a general term covering several types of control systems and their bonded components used for industrial process control. With increased integration with computers, industrial control systems are becoming more open. Therefore, the author designed a machine learning intrusion detection system in order to detect incoming attacks on industrial control systems . In this final project, implement a binary classification in designing an intrusion detection system with python as software the supporting dataset taken comes from the dataset power system attacks at Mississippi State University. Making machine learning models using multi layer perceptron learning algorithm backpropagation. The results of this final project can apply neural networks as a way to detect an attack. After training with 5 hidden layer variations, the greatest accuracy value is 72.3%. With an average precision of 0.71, recall of 0.72, and an f1-score of 0.60.

Keywords: Industrial control systems, intrusion detection systems, machine learning, binary classification, multi layer perceptron, backpropagation, neural networks.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang senantiasa melimpahkan rahmat dan hidayah-Nya, sehingga penulisan skripsi ini dapat diselesaikan. Skripsi ini ditulis sebagai salah satu syarat untuk menyelesaikan program pendidikan Strata Satu di Institut Teknologi Speuluh Nopember Surabaya.

Peneliti menyadari bahwa skripsi ini dapat diselesaikan berkat dukungan dan bantuan dari berbagai pihak. Peneliti menyampaikan ucapan terima kasih kepada :

1. Ibu Linna Muzayanti dan Bapak Moch. Padang Dirgantara selaku kedua orang tua penulis, serta Amelinda Batari dan Fadiah Novemia Madarina selaku kakak dan adik penulis yang telah memberi semangat dan doa kepada penulis.
2. Bapak Dr. Bambang Lelono Widjiantoro, S.T., M.T. dan Bapak Moh Kamalul Wafi, S.T., M.Sc.DIC selaku Dosen Pembimbing yang telah membantu penulis dalam menyelesaikan tugas akhir ini.
3. Ibu Dyah Sawitri, S.T., M.T. dan Ibu Dr. Katherin Indriawati, S.T., M.T. selaku dosen penguji Tugas Akhir saya yang telah membantu penulis dalam menguji dan memperbaiki laporan Tugas Akhir.
4. Bapak Dr. Ir. Syamsul Arifin, M.T. selaku Kepala Laboratorium Sistem Tanam dan Siber Fisik yang telah memberikan fasilitas, ilmu, dan petunjuk kepada penulis.
5. Ibu Dr.-Ing Doty Dewi Risanti, S.T., M.T. Selaku dosen wali yang telah banyak membantu dalam hal akademik selama penulis kuliah di Teknik Fisika ITS.
6. Seluruh dosen, karyawan, dan civitas akademika ITS yang telah memberikan kesempatan dan bantuan kepada penulis.
7. Teman-teman Tiksna Falcata yang membantu saya dalam berjuang menghadapi dunia perkuliahan ini.
8. Teman-teman Himpunan Mahasiswa Teknik Fisika yang telah membantu banyak penulis dalam mengembangkan soft skill maupun hard skill di dalam lingkungan Teknik Fisika ITS.

Serta pihak-pihak lain yang tidak dapat disebutkan satu-persatu. Semoga laporan tugas akhir ini dapat dipergunakan dengan sebaik-baiknya.

Surabaya, 9 Februari 2022

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

HALAMAN JUDUL i
COVER PAGE iii
PERNYATAAN BEBAS PLAGIASI v
LEMBAR PENGESAHAN vii
LEMBAR PENGESAHAN ix
ABSTRAK xi
ABSTRACT xiii
KATA PENGANTAR xv
DAFTAR ISI xvii
DAFTAR GAMBAR xix
DAFTAR TABEL xxi
BAB I PENDAHULUAN 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 2
1.3 Tujuan 2
1.4 Batasan Masalah 2
1.5 Sistematika Laporan 3
BAB II TINJAUAN PUSTAKA DAN SASAR TEORI 5
2.1 Sistem Kontrol Industri 5
2.2 Sistem Deteksi Intrusi 5
2.3 *Neural Networks* 6
2.4 *Multi Layer Perceptron* (*Feed forward neural networks*) 7
2.5 Mode Pelatihan *Backpropagation* 8
2.6 Supervised Machine Learning 8
2.7 *Tensor Flow* 9
2.8 Pengujian dan Pelatihan 10
2.9 *Binary Classification* 10
2.10 *Confussion Matrix* 11
2.11 Perfomansi Metrik 12
BAB III METODOLOGI PENELITIAN 15
3.1 Perumusan Masalah 15
3.2 Penetapan Tujuan 16
3.3 Studi literatur 16
3.4 Perancangan Sistem Deteksi Intrusi 16
3.5 Pengumpulan dan Eksplorasi Data 17
3.6 Pembuatan Model *Machine Learning* 19
3.7 Pelatihan Model *Machine Learning* 20

3.8 Evaluasi Model *Machine Learning* 21
3.9 Analisa Data 21
3.10 Penarikan Kesimpulan 22
3.11 Penyusunan Laporan 22
BAB IV HASIL DAN PEMBAHASAN 23
4.1 Hasil Pelatihan Model *Machine Learning* 23
4.2 Pengujian Deteksi Intrusi menggunakan *Machine Learning* 35
4.3 Analisa Performansi Model 39
BAB V KESIMPULAN DAN SARAN 43
5.1 Kesimpulan 43
5.2 Saran 43
DAFTAR PUSTAKA 45
LAMPIRAN xlvii
BIODATA PENULIS xlix

DAFTAR GAMBAR

Gambar 2. 1 Sistem kontrol industri (Stouffer & Falco, 2006).....5
Gambar 2. 2 Arsitektur Sistem Deteksi Intrusi (Bachar et al., 2020) 6
Gambar 2. 3 *neural networks* (IBM Cloud Education, 2020).....7
Gambar 2. 4 Arsitektur *Multi layer perceptron* (LeNail, 2019).....7
Gambar 2. 5 Arsitektur *supervised machine learning*.....9
Gambar 2. 6 Ilustrasi *binary classification* 10
Gambar 2. 7 Fungsi sgn (Gotama, 2020) 11
Gambar 2. 8 *Confusion matrix* (Nugroho, 2019) 12
Gambar 2. 9 *Confusion matrix* menggambarkan nilai *accuracy* (Nugroho, 2019).....12
Gambar 2. 10 *Confusion matrix* menggambarkan nilai *precision* (Nugroho, 2019).....13
Gambar 2. 11 *Confusion matrix* menggambarkan nilai *recall* (Nugroho, 2019) 13
Gambar 3. 1 Diagram Alir Metodologi Penelitian.....15
Gambar 3. 2 Diagram perancangan model *machine learning* (Pan et al., 2015) 16
Gambar 3. 3 Arsitektur jaringan sistem kontrol industri (Pan et al., 2015) 17
Gambar 3. 4 *Feedforward neural network*.....19
Gambar 3. 5 Diagram blok pelatihan model 20
Gambar 4. 1 Grafik *binary accuracy* dengan 8 *hidden layer*.....23
Gambar 4. 2 Grafik *binary crossentropy* dengan 8 *hidden layer*.....24
Gambar 4. 3 Grafik *loss function* terendah dengan *8 hidden layer*.....25
Gambar 4. 4 Grafik *binary accuracy* dengan 16 *hidden layer*.....26
Gambar 4. 5 Grafik *binary crossentropy* dengan 16 *hidden layer*.....26
Gambar 4. 6 Hasil *loss function* terendah dengan 16 *hidden layer*.....27
Gambar 4. 7 Grafik *binary accuracy* dengan 32 *hidden layer*.....28
Gambar 4. 8 Grafik *binary crossentropy* dengan 32 *hidden layer*.....29
Gambar 4. 9 Hasil *loss function* terendah dengan 32 *hidden layer*.....30
Gambar 4. 10 Grafik *binary accuracy* dengan 64 *hidden layer*.....31
Gambar 4. 11 Grafik *binary crossentropy* dengan 64 *hidden layer*.....31
Gambar 4. 12 Hasil *loss function* terendah dengan 64 *hidden layer*.....32
Gambar 4. 13 Grafik *binary accuracy* dengan 128 *hidden layer*.....33
Gambar 4. 14 Grafik *binary crossentropy* dengan 128 *hidden layer*.....34
Gambar 4. 15 Hasil *loss function* terendah dengan 128 *hidden layer*.....35
Gambar 4. 16 Hasil *confusion matrix* dengan 8 *hidden layer*.....36

Gambar 4. 17 Hasil *confusion matrix* dengan 16 *hidden layer* 36
Gambar 4. 18 Hasil *confusion matrix* dengan 32 *hidden layer* 37
Gambar 4. 19 Hasil *confusion matrix* dengan 64 *hidden layer* 38
Gambar 4. 20 Hasil *confusion matrix* dengan 128 *hidden layer* 38

DAFTAR TABEL

Tabel 3. 1 Parameter Input pada *dataset* (Pan et al., 2015) 18
Tabel 4. 1 Tabel persentase hasil nilai akhir *binary accuracy* dengan *8 hidden layer* 23
Tabel 4. 2 Tabel hasil nilai akhir *binary crossentropy* dengan *8 hidden layer* 24
Tabel 4. 3 Tabel persentase hasil nilai akhir *binary accuracy* dengan 16 *hidden layer* 26
Tabel 4. 4 Tabel hasil nilai akhir *binary crossentropy* dengan 16 *hidden layer* 27
Tabel 4. 5 Tabel persentase hasil nilai akhir *binary accuracy* dengan 32 *hidden layer* 28
Tabel 4. 6 Tabel hasil nilai akhir *binary crossentropy* dengan 32 *hidden layer* 29
Tabel 4. 7 Tabel persentase hasil nilai akhir *binary accuracy* dengan 64 *hidden layer* 31
Tabel 4. 8 Tabel hasil nilai akhir *binary crossentropy* dengan 64 *hidden layer* 32
Tabel 4. 9 Tabel persentase hasil nilai akhir *binary accuracy* dengan 128 h*idden layer* ... 33
Tabel 4. 10 Tabel hasil nilai akhir *binary crossentropy* dengan 128 *hidden layer* 34
Tabel 4. 11 Performansi *nilai precision, recall, dan f1-score* dengan 8 *hidden layer* 39
Tabel 4. 12 Performansi *nilai precision, recall, dan f1-score* dengan 16 *hidden layer* 39
Tabel 4. 13 Performansi *nilai precision, recall, dan f1-score* dengan 32 *hidden layer* 39
Tabel 4. 14 Performansi *nilai precision, recall, dan f1-score* dengan 64 *hidden layer* 40
Tabel 4. 15 Performansi *nilai precision, recall, dan f1-score* dengan 128 *hidden layer* 40

Halaman ini sengaja dikosongkan

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sistem kontrol industri adalah istilah umum yang mencakup beberapa jenis sistem kontrol dan komponen terkait yang digunakan untuk kontrol proses industri. Sistem kontrol industri bertanggung jawab atas akuisisi data waktu nyata, pemantauan sistem, dan kontrol otomatis serta manajemen proses industri. Dengan peningkatan integrasi dengan komputer dan teknologi Internet, sistem kontrol industri menjadi lebih cerdas dan lebih terbuka (Stouffer & Falco, 2006).

Sistem deteksi intrusi adalah sistem untuk mendeteksi serangan pada jaringan. Umumnya, deteksi serangan dilakukan dengan memvalidasi pola lalu lintas jaringan dengan pola serangan yang diketahui atau dengan mengetahui pola lalu lintas dari sebuah jaringan yang *abnormal* (anomali). Dalam rangka merancang sebuah sistem deteksi intrusi yang efisien maka dapat menggunakan metode deteksi anomali, seorang analis mengandalkan pengalaman untuk menyesuaikan ukuran statistik deteksi intrusi dan untuk deteksi kesalahan. Seorang analis harus menganalisis terlebih dahulu dan mengklasifikasikan model serangan, kekurangan sebuah sistem, dan merancang sebuah aturan dan pola sesuai dengan serangan yang ada (Ellis, 2017).

Dengan semakin berkembangnya kegiatan yang dilakukan seseorang di dunia maya dan serangan terhadap sebuah jaringan sistem komputer semakin meningkat tahun ke tahun, yang berarti bahwa data yang dianalisis sangat besar dan dapat menjadi sebuah masalah bagi seorang analis paket data dalam menklasifikasikan data dan membentuk model skenario dari yang dikumpulkan. Sehingga timbul kecurigaan bahwa sistem pendeteksi serangan yang digunakan tidak memungkinkan untuk mendeteksi serangan berbahaya yang dilakukan dengan teknik baru. Masalah ini mendorong perlunya sistem yang dapat membantu analis dalam proses analisis data dan dapat menemukan serangan yang tidak dapat ditemukan oleh analis atau sensor (Sprengers & van Haaster, 2016).

Metode *data mining* digunakan untuk mendeteksi sebuah intrusi layak untuk menjadi solusi untuk masalah pertumbuhan data dalam jumlah yang banyak karena mempunyai keunggulan dalam memproses log sistem atau mengaudit data dalam jumlah besar dan metode data mining dapat membantu mengintegrasikan kedua teknik deteksi intrusi (anomali dan penyalahgunaan). Untuk menemukan pola rutin dalam dataset yang besar maka doperlukan *data mining* dan sanggup untuk memberikan solusi untuk masalah penghapusan

data pada sekelompok data yang besar, sehingga memudahkan analis dalam mengenali data (Salo et al., 2018).

Jaringan syaraf tiruan (JST) atau *neural networks* dapat diterapkan dalam memecahkan sebuah masalah di sebuah jaringan komputer, terlebih pada sebuah sistem deteksi intrusi (IDS). Sistem deteksi intrusi bukanlah sebuah bidang baru dalam dunia komputer, namun belum dimanfaatkan secara komprehensif dalam sebuah jaringan komputer komersial. Memang, sistem komputer belum sempurna untuk memecahkan masalah (Chamou et al., 2019).

Pada kasus tugas akhir yang saya kerjakan, saya mengambil metode klasifikasi biner dengan dua jenis kelas data. Klasifikasi biner adalah salah satu teknik dalam mencari sebuah hasil *output* yang memiliki dua jenis data yang akan diuji menggunakan sebuah *confusion matrix* dengan analisa performansi *metric* (Caelen, 2017). Metode tersebut dilakukan agar kita dapat mengetahui seberapa bagus model yang kita buat.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah pada penelitian ini adalah:

a) Bagaimana merancang sebuah *machine learning* sistem pendeteksi intrusi pada sistem kontrol industri?
b) Bagaimana keakuratan sistem pendeteksi intrusi dalam mencegah serangan yang masuk ke sistem kontrol industri?

1.3 Tujuan

Tujuan dari penelitian ini adalah :

a) Merancang sistem pendeteksi intrusi pada sistem kontrol industri dengan menggunakan algoritma *machine learning*.
b) Menganalisa keakuratan sistem pendeteksi intrusi dalam mencegah serangan yang masuk ke sistem kontrol indutri.

1.4 Batasan Masalah

Batasan Masalah pada Tugas Akhir ini adalah sebagai berikut :

a) Pada tahap pengumpulan data berasal dari *dataset traffic* dcs dari Missisippi State University dan Oak Ridge National Laboratory pada tahun 2014.
b) *Software* yang dirancang dan dibangun hanya memiliki fungsi untuk menjadi sistem deteksi intrusi ketika mendapat serangan siber pada sistem kontrol industri

c) Algoritma dirancang menggunakan bahasa pemrograman python merupakan *binary classification.*
d) Machine learning yang dipakai dalam memodelkan menggunakan *multi layer perceptron (feedforward neural networks)* algoritma pembelajaran *backpropagation*.
e) Pendeteksi intrusi hanya berupa deteksi adanya serangan atau tidak.

1.5 Sistematika Laporan

Laporan tugas akhir ini terdiri dari lima bab dan dilengkapi dengan daftar Pustaka serta lampiran yang berisi cetak biru desain dan *source code* yang digunakan. Secara garis besar, sistematika dari penulisan tugas akhir ini adalah sebagai berikut :

a. BAB I PENDAHULUAN

   Pada bab I ini terdiri latar belakang, rumusan masalah, tujuan penelitian, perumusan masalah, dan sistematika laporan.

b. BAB II TINJAUAN PUSTAKA DAN DASAR TEORI

   Pada bab II berisi tentang dasar teori yang berkaitan dengan penelitian seperti Sistem kontrol industri, Sistem Deteksi Intrusi,

c. BAB III METODOLOGI PENELITIAN

   Pada bab II berisi mengenai rancangan penelitian yang dilakukakan, metode dan Langkah Langkah dalam penelitian.

d. BAB IV ANALISA DATA DAN PEMBAHASAN

   Pada bab IV berisi Analisa hasil perancangan

e. BAB V PENUTUP

   Pada bab V berisi kesimpulan dari penelitian yang telah dilakukan , serta saran sebagai penunjang dalam pengembangan akhir selanjutnya.

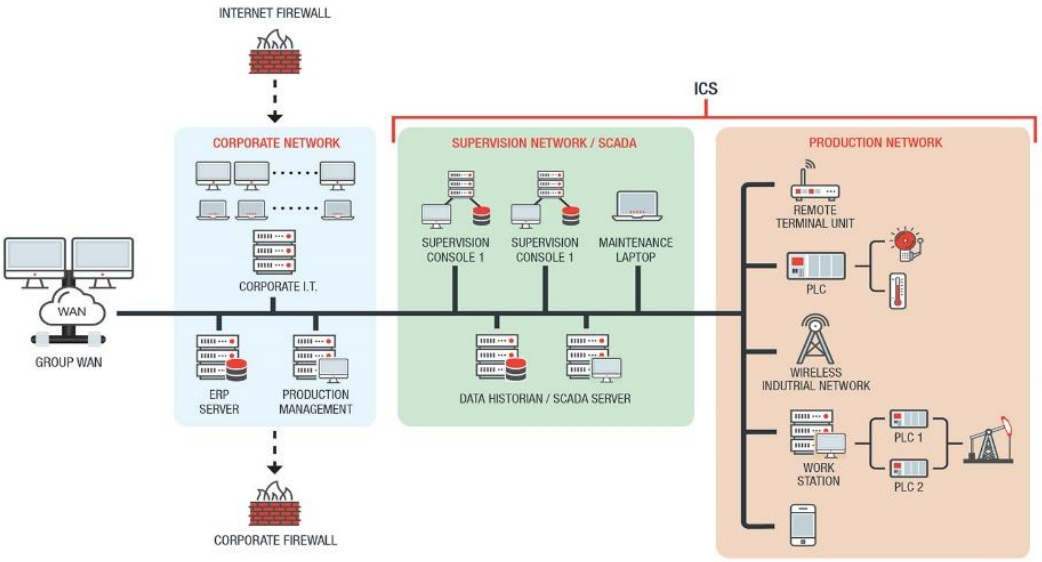
Halaman ini sengaja dikosongkan

BAB II
TINJAUAN PUSTAKA DAN SASAR TEORI

2.1 Sistem Kontrol Industri

Sistem kontrol industri adalah istilah umum yang mencakup beberapa jenis sistem kontrol, termasuk sistem kontrol pengawasan dan akuisisi data, *distributed control system* (DCS), dan konfigurasi sistem kontrol lainnya seperti *Programmable Logic Controllers* (PLC) sering ditemukan di sektor industri dan infrastruktur kritis (Stouffer & Falco, 2006).

Sistem memiliki persyaratan kinerja dan keandalan yang berbeda, dan juga menggunakan sistem operasi dan aplikasi yang mungkin dianggap tidak konvensional dalam lingkungan jaringan sistem informasi yang khas. Perlindungan keamanan harus diterapkan dengan cara menjaga integritas sistem selama operasi normal serta selama serangan terjadi.



Gambar 2. 1 Sistem kontrol industri (Stouffer & Falco, 2006)

2.2 Sistem Deteksi Intrusi

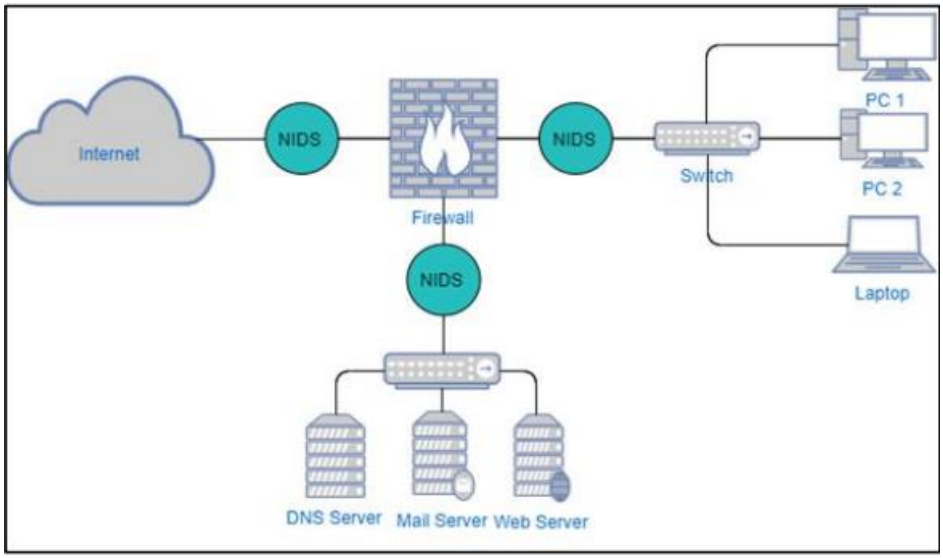
Sistem Deteksi Intrusi memungkinkan kita untuk melindungi infrastruktur jaringan komputer dari aktivitas jahat apa pun. Aktivitas ini sering menargetkan integritas, kerahasiaan, dan ketersediaan data di jaringan (Bachar et al., 2020). Sistem Deteksi intrusi adalah kontrol keamanan reaktif yang berusaha mengidentifikasi secara otomatis pelanggaran kebijakan keamanan sistem yang dipantau. Untuk melakukan tugasnya, sistem deteksi intrusi dapat menggunakan sumber data yang berbeda dari lingkungan yang dipantau. Kemudian, melalui metode deteksi, Sistem Deteksi Intrusi mendeteksi adanya

gangguan dan memunculkan peringatan. Sistem Deteksi Intrusi dapat diklasifikasikan tergantung pada jenis sumber data dan metode deteksi.

Sistem Deteksi Intrusi diklasifikasikan menurut beberapa kriteria seperti jenis IDS dan metode klasifikasi yang terkait yaitu sebagai berikut.

N-IDS (Network Intrusion Detection System): NIDS mengamankan seluruh jaringan. Lokasinya di jaringan mempengaruhi tingkat *false alarm*. Misalnya, jika NIDS terletak di bagian hulu *Firewall*, maka *false alarm* yang dihasilkan lebih sedikit. Karena lalu lintas sudah disaring oleh *firewall*. Jika tidak, jika ditempatkan di hilir Firewall. Ini menghasilkan lebih banyak *false alarm*.

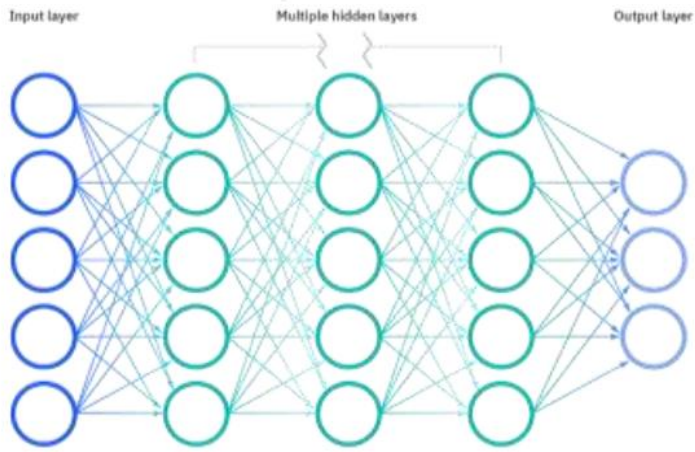
H-IDS (Host Intrusion Detection System): Ini mengamankan mesin sendiri tanpa menggunakan sistem lain.



Gambar 2. 2 Arsitektur Sistem Deteksi Intrusi (Bachar et al., 2020)

2.3 *Neural Networks*

Neural networks adalah adalah sebuah model yang menerapkan prinsip bagaimana neuron pada otak manusia bekerja (Hassoun, 1995). Neural networks terdiri dari 3 lapis yaitu lapis input (*input layer*), lapis output (*output layer*), dan lapis tersembunyi (*hidden layer*). Dalam setiap lapis terdapat neuron. Neuron pada otak manusia dapat menghubungkan dengan *neuron* lainnya. Tiap neuron menerima input dan menjalankan operasi dengan sebuah *weight* dan mentotalkannya. Hasil dari operasi tersebut akan menjadi parameter fungsi aktivasi yang akan menjadi output dari neuron tersebut. Neural networks diilustrasikan sesuai dengan gambar 2.3.

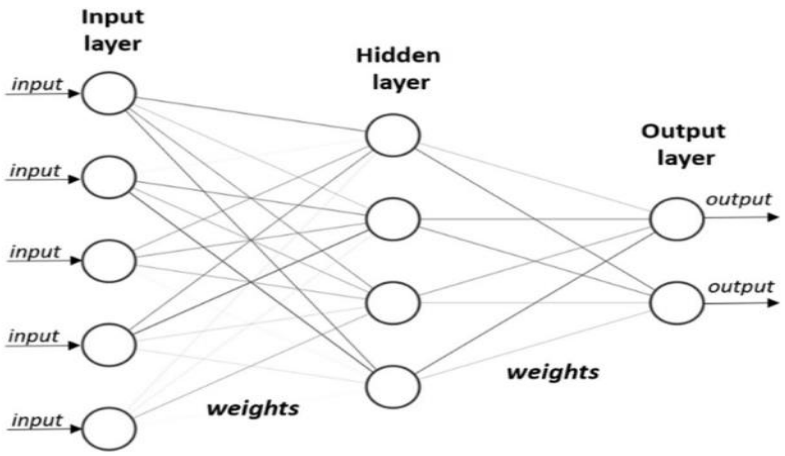


Gambar 2. 3 *neural networks* (IBM Cloud Education, 2020)

Banyak sekali perbedaan arsitektur *neural networks* yang telah diaplikasikan ke dalam sebuah sistem deteksi intrusi. Arsitektur yang paling sering digunakan untuk diaplikasiakan ke dalam sistem deteksi intrusi adalah *feedforward neural networks (multi layer perceptron)* (van Veen & Leijnen, 2016). *Multi layer perceptron* (MLP) adalah model simple komputasi yang mengikuti cara kerja otak manusia. MLP tediri dari 3 *fully connected layers* dari sebuah *perceptron.*

2.4 *Multi Layer Perceptron* (*Feed forward neural networks*)

Multi Layer Perceptron (MLP) adalah salah satu contoh arsitektur *neural networks*. Pada MLP ini neuron saling berkaitan satu sama lain dengan menjalankan operasi dengan menggunakan weight.. Pada multi layer perceptron terdapat tiga layers yaitu *input layer*, *hidden layer*, dan *output layer*.



Gambar 2. 4 Arsitektur *Multi layer perceptron* (LeNail, 2019)

Input layer menerima sebuah *input* yang kita berikan tanpa melakukan sebuah operasi apapun. Kemudian nilai sebuah input diberikan ke *hidden layer*. *Hidden layer* berfungsi untuk memproses sebuah *input* dan dilakukan perhitungan nilai aktivasi pada tiap neuron. Lalu hasilnya diberikan kepada *layer* berikutnya. *Output dari input layer* diterima sebagai hasil dari *input hidden layer* (Mitchell & McGraw-Hill, 1997). Arsitektur dari sebuah multi layer perceptron diilustrasikan seperti pada gambar 2.4

Pada lapisan *input* (x) dan lapisan tersembunyi terdapat neuron yang disebut bias selalu menunjukkan nilai 1 Nilai yang tersimpan pada neuron di layer tersembunyi (z) sesuai dengan persamaa (2.1). Dimana (b) merupakan nilai bias, dan (w) merupakan nilai *weights*

$$z = \left(\sum_{i=1}^{n} x_i * w_{i1}\right) + b_{i1} \tag{2.1}$$

Nilai z digunakan sebagai parameter fungsi aktivasi ($f_{act}(z)$), sehingga *output* (y) dari neuron menjadi persamaan (2.2).

$$y = f_{act}(z) \tag{2.2}$$

Jenis fungsi aktivasi ada berbagai macam seperti *leaky Rel.U, Sigmoid, tanh, swish, linear, dan RelU.*

≥

2.5 Mode Pelatihan *Backpropagation*

Backpropagation merupakan salah satu metode pelatihan neural networks untuk mendapatkan regresi dari suatu fungsi. Setelah pelatihan selesai dilakukan, makanakan didapatkan nilai bobot. Nilai tersebut akan diimplementasikan pada *feedfordward neural networks* untuk memprediksi output suatu fungsi. Algoritma *backprogation* umumnya digunakan dalam melatih *Multi Layer Perceptron* (Holyoak, 1987). Backprogation adalah metode *gradient-based optimization* yang diterapkan pada sebuah *artificial neural networks*.

Konsep dari sebuah *backpropagation* adalah ketika sebuah model diberikan pasang *input* (x) dan *output* yang diinginkan (y) sebagai sebuah training data. Untuk meminimalkan loss, algoritma *backropagation* menggunakan sebuah prinsip *gradient descent* (Gotama, 2020).

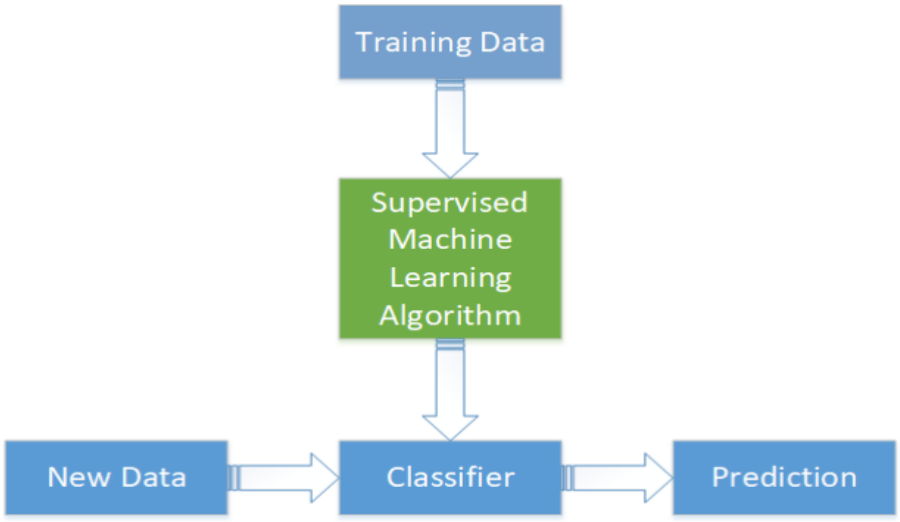
2.6 Supervised Machine Learning

Supervised Machine Learning menggunakan komputer untuk mensimulasikan pembelajaran manusia dan memungkinkan mereka untuk mengidentifikasi dan memperoleh pengetahuan dari dunia nyata dan meningkatkan kinerja pada beberapa tugas berdasarkan

pengetahuan baru (Portugal et al, 2018). *Supervised machine learning* bekerja ketika seseorang memiliki Variabel *input* (x) dan variabel *output* (y) menggunakan algoritma untuk mempelajari fungsi pemetaan dari *input* ke *output*, dengan rumus $y = f(x)$. Di dalamnya, semua data diberi label dan algoritma akan memelajara hasil fungsi pemetaan untuk memprediksi *output* dari data *input* (Brownlee, 2016) Hal ini dimaksudkan untuk memperkirakan fungsi pemetaan sebaik mungkin sehingga ketika seseorang nantinya memiliki data input baru (x), ia dapat memprediksi variabel output (y) dari data tersebut.

Supervised learning dapat dikelompokkan lebih lanjut ke dalam regresi dan klasifikasi. Masalah regresi adalah ketika variabel output adalah nilai nyata, seperti dolar atau berat. Sedangkan masalah klasifikasi adalah ketika variabel keluaran berupa kategori, seperti 'merah atau biru' dan 'ya atau tidak' (Brownlee, 2016). Gambar 2.5 menunjukkan diagram prosedur pembuatan model untuk klasifikasi data.

Algoritma *supervised learning* (klasifikasi) lebih disukai daripada algoritma *unsupervised learning* (pengelompokan), karena pengetahuan sebelumnya dari *data record label class* memudahkan pemilihan fitur/atribut, dan menghasilkan akurasi prediksi/klasifikasi yang lebih baik.



Gambar 2. 5 Arsitektur *supervised machine learning*

2.7 *Tensor Flow*

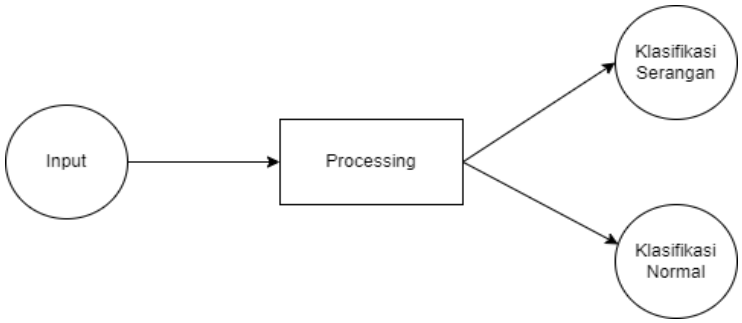
Tensorflow adalah *open-source software library* yang dikembangkan oleh *Google* untuk komputasi numerik, yang sekarang banyak digunakan oleh banyak perusahaan besar. *Tensor flow* menyediakan antarmuka untuk mengekspresikan algoritma *machine learning* dan aplikasi untuk menjalankan algoritma ini (Abadi et al., 2016).

2.8 Pengujian dan Pelatihan

Dalam sebuah machine learning, terdapat dua parameter yang penting yaitu *training* dan *testing*. *Training* atau pelatihan merupakah proses pembentukan sebuah model dan *testing* atau pengujian merupakan proses menguji kinerja sebuah model pembelajaran. Dataset merupakan kumpulan data atau sampel data dalam bentuk statistik. Dalam mengolah sebuah dataset diperlukan dua parameter *training* dan *testing* agar dapat mengevaluasi sebuah model *machine learning*. Rasio pembagian dataset umumnya memiliki rasio (90% : 10%), (80% : 20%), (70% : 30%), atau (50% : 50%) (Gotama, 2020).

2.9 *Binary Classification*

Binary classification adalah sebuah metode klasifikasi data menjadi dua kelas. Umumnya binary classification digunakan untuk memprediksi dua keadaan (Kumari & Srivastava, 2017). Contohnya adalah serangan atau tidak ada serangan.



Gambar 2. 6 Ilustrasi *binary classification*

Gambar 2.6 merupakan sebuah ilustrasi bagaimana klasifikasi biner tersebut. Dapat dilihat pada gambar diatas dimana hasil dari *input* tersebut akan diolah oleh sebuah model yang nantinya akan memperoleh 2 jenis *output*. Setelah hasil didapatkan maka kita dapat mengevaluasi sebuah model tersebut. Secara *neural networks* klasifikasi dimodelkan secara *blackbox* (model matematis secara spesifik tidak diketahui) namun secara konseptual klasifikasi didefinisikan sesuai dengan persamaan (2.3) yaitu memilih label paling optimal dari sekumpulan label C diberikan suatu sampel data (Gotama, 2020).

$$\hat{y_i} = \arg\max p(y_i \mid x_i, w) \tag{2.3}$$

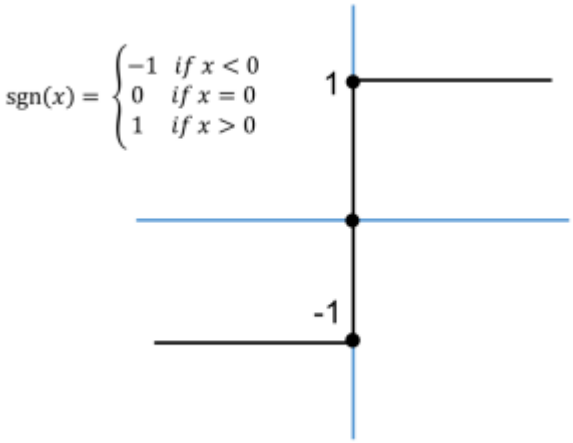
Persamaan (2.4) dibawah merupakan persamaan model sederhana untuk *binary classification*. Dimana suatu data direpresentasikan sebagai fitur vector x, dan terdapat bias b, serta weight w. Klasifikasi dilaksanakan dengan melewatkan data pada sebuah fungsi yang memiliki parameter. Fungsi tersebut menghitung bobot pada setiap vektor dan mengkalikan dengan *dot product*.

$$f(x) = x \,.\, w + b \tag{2.4}$$

Kemudian dari persamaan (2.4), kita dapat menulis Kembali dengan persamaan (2.5) yang dimana x merupakan elemen ke-i dari sebuah vektor x. Misal nilai $f(x) > threshold$ maka di masukkan pada kelas pertama dan jika $f(x) \leq threshold$ maka dimasukkan pada kelas kedua.

$$f(x) = x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + b \quad (2.5)$$

Threshold disini menjadi sebuah pembatas antar kelas pertama dan kelas kedua (*decision boundary*). Pada umunya threshold menggunakan sebuah fungsi sgn yang diilustrikan sesuai gambar 2.7.



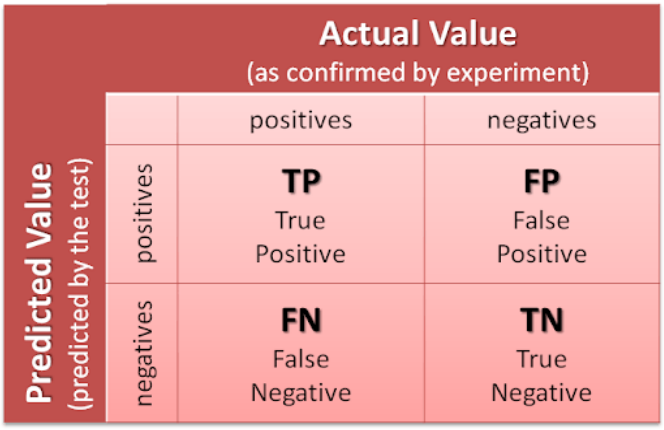
Gambar 2. 7 Fungsi sgn (Gotama, 2020)

Untuk mengubah nilai fungsi menjadi [-1,1] sebagai output maka dirumuskan sesuai dengan persamaan (2.6).

$$output = sgn(f(x)) \quad (2.6)$$

2.10 *Confussion Matrix*

Confusion matrix merupakan perbandingan hasil klasifikasi model dengan klasifikasi yang sesungguhnya. *Confusion Matrix* biasanya berupa tabel matriks $n\,x\,n$ yang menunjukkan kinerja dari klasifikasi pada serangkaian data yang diuji nilai sesungguhnya. *Confusion matrix* digunakan untuk *machine learning* untuk mengeavaluasi dan menvisualisasikan kebiasaan dari sebuah model (Caelen, 2017). Gambar 2.8 Merupakan gambar *confusion matrix* dengan nilai aktual dan prediksi.

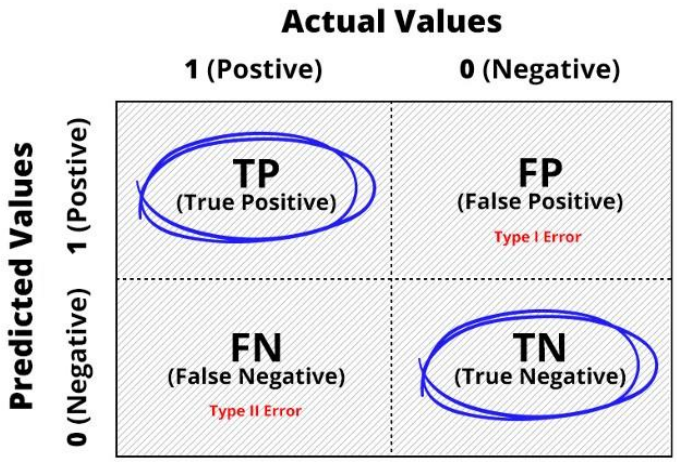


Gambar 2. 8 *Confusion matrix* (Nugroho, 2019)

True positives (TP) merupakan data positif dan diprediksi benar. *True negatives* (TN) merupakan data negative yang diprediksi benar. *False positive* (FP) atau biasa disebut eror tipe 1 merupakan data negatif yang diprediksi sebagai data positif. Dan *false negative* (FN) atau biasa disebut eror tipe 2 merupakan data positif namun diprediksi sebagai data negatif.

2.11 Perfomansi Metrik

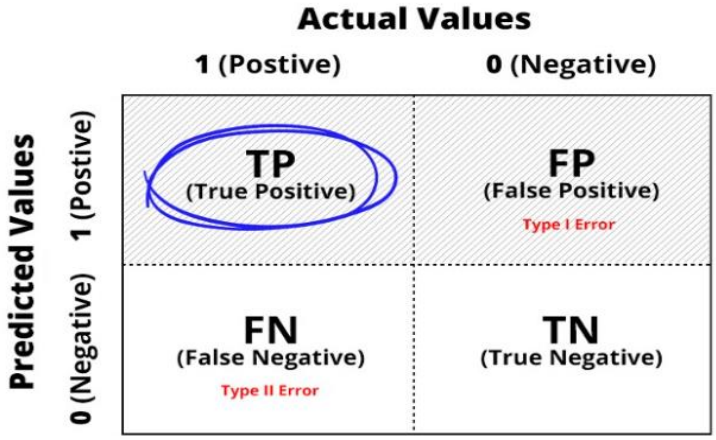
Dalam performansi metrik terdapat 4 parameter yang paling sering digunakan yaitu *accuracy*, *precision*, *recall*, dan *f1 score*. Cara lebih detail dalam mengevaluasi hasil klasifikasi adalah mengukur nilai *precision* dan *recall* (Janardhanan & Sabika, 2015). *Accuracy* (A) merupakan nilai kekauratan model yang dapat mengklasifikan dengan benar. *Accuracy* didefinisikan sebagai perbandingan nilai prediksi benar (positif dan negatif) dengan kesuluruhan data. Dirumuskan dengan persamaan (2.7).



Gambar 2. 9 *Confusion matrix* menggambarkan nilai *accuracy* (Nugroho, 2019)

$$A = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.7}$$

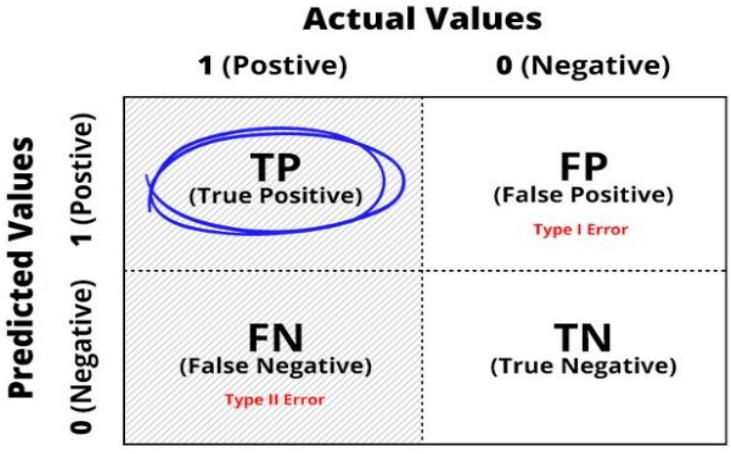
Precision (P) dideskripsikan sebagai rasio dari jumlah *true positive* (TP) dibagi dengan jumlah *true positive* (TP) dan *false positives* (FP). Dan dirumuskan sesuai persamaan (2.8).



Gambar 2. 10 *Confusion matrix* menggambarkan nilai *precision* (Nugroho, 2019)

$$P = \frac{TP}{(TP + FP)} x\ 100\% \tag{2.8}$$

Recall (R) dideskripsikan sebagai jumlah *true positives* (TP) dibagi tengan jumlah *true positives* (TP) dan *false negatives* (FN). Dan dirumuskan sesuai dengan persamaan (2.9).



Gambar 2. 11 *Confusion matrix* menggambarkan nilai *recall* (Nugroho, 2019)

$$R = \frac{TP}{(TP + FN)} x\ 100\% \tag{2.9}$$

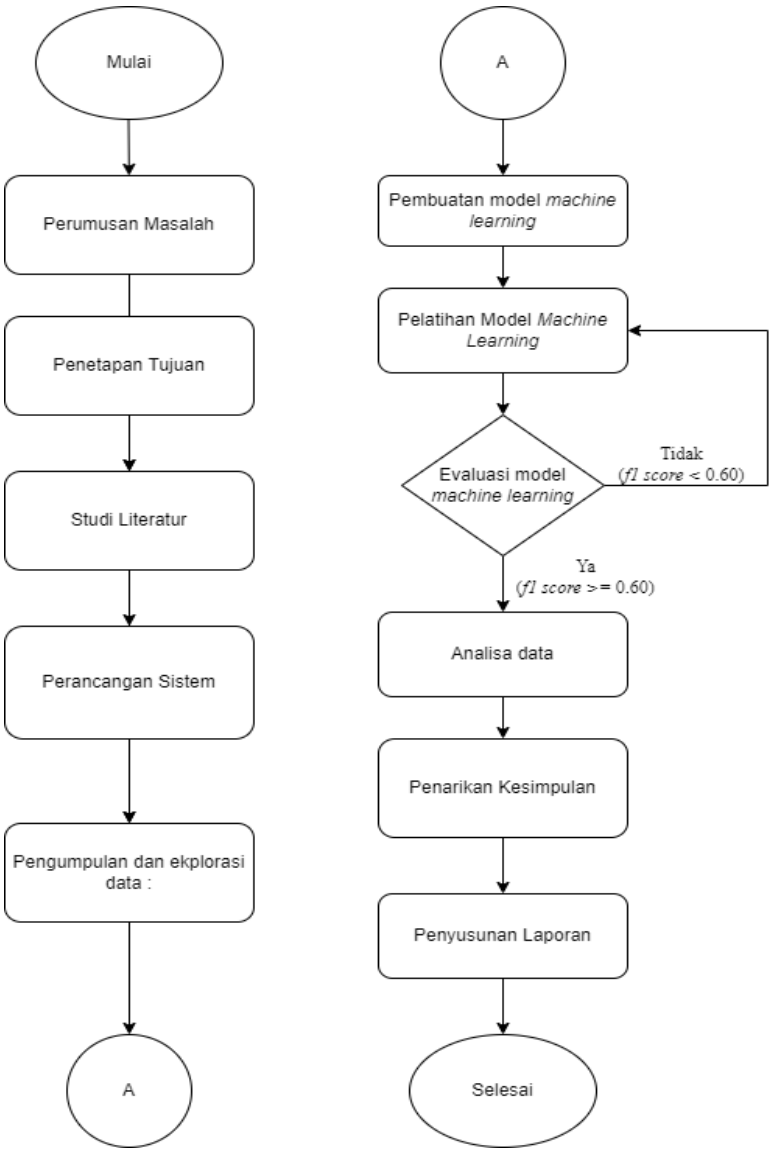
F1 score (F) dideskripsikan sebagai rata-rata harmonik dari nilai *precision* (P) dan *recall* (R) dibagi jumlah dari *precission* (P) dan *recall* (R). Dirumuskan oleh persamaan (2.10).

$$F = \frac{2.P.R}{(P + R)} \tag{2.10}$$

Halaman ini sengaja dikosongkan

BAB III
METODOLOGI PENELITIAN

Tahapan penelitian Tugas Akhir ini, secara umum dapat digambarkan dalam *flowchart* seperti gambar 3.1 :



Gambar 3. 1 Diagram Alir Metodologi Penelitian

3.1 Perumusan Masalah

Perumusan masalah pada penelitian ini adalah bagaimana merancang sistem pendeteksi intrusi pada sistem kontrol industri dengan menggunakan algoritma *machine learning* dan berapa akurasi sistem pendeteksi intrusi dalam mencegah serangan yang masuk ke sistem kontrol industri.

3.2 Penetapan Tujuan

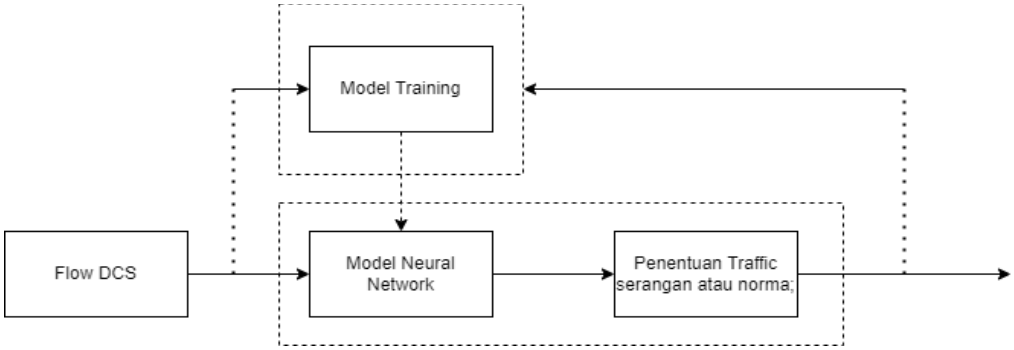
Penetapan ditujuan dilakukakn setelah mengetahui latar belakang dan rumusan masalah yang ada. Tujuan dari penelitian ini adalah merancang sistem pendeteksi intrusi pada sistem kontrol industri dengan menggunakan algoritma *machine learning* dan menganalisa keakuratan sistem pendeteksi intrusi dalam mencegah serangan yang masuk ke sistem kontrol indutri.

3.3 Studi literatur

Tahap awal pada pengerjaan tugas akhir ini dimulai dengan adanya studi literatur sebagai upaya pemahaman terhadap materi yang menunjang tugas akhir mengenai "Perancangan dan Analisis Sistem Deteksi Intrusi pada Sistem Kontrol Industri". Studi literatur ini dilakukan dengan mencari dan mempelajari informasi dari jurnal mengenai sistem deteksi intrusi dan sistem kontrol industri.

3.4 Perancangan Sistem Deteksi Intrusi

Sistem deteksi intrusi yang saya gunakan disini untuk mencegah adanya serangan pada sistem kontrol industri. Untuk *software* yang digunakan dalam merancang sistem deteksi intrusi adalah dengan menggunakan aplikasi *python* dan Bahasa pemrograman yang mendukung dalam tugas akhir ini. Untuk algoritma yang digunakan adalah algoritma *Binary Classification*. *Binary Classification* merupakan salah satu contoh algoritma klasifikasi yang dimana hasil nya berupa dua kelas data. Pada kasus ini menggunakan label serangan didefinisikan sebagai angka 0 dan label normal didefinisikan sebagai angka 1.



Gambar 3. 2 Diagram perancangan model *machine learning* (Pan et al., 2015)

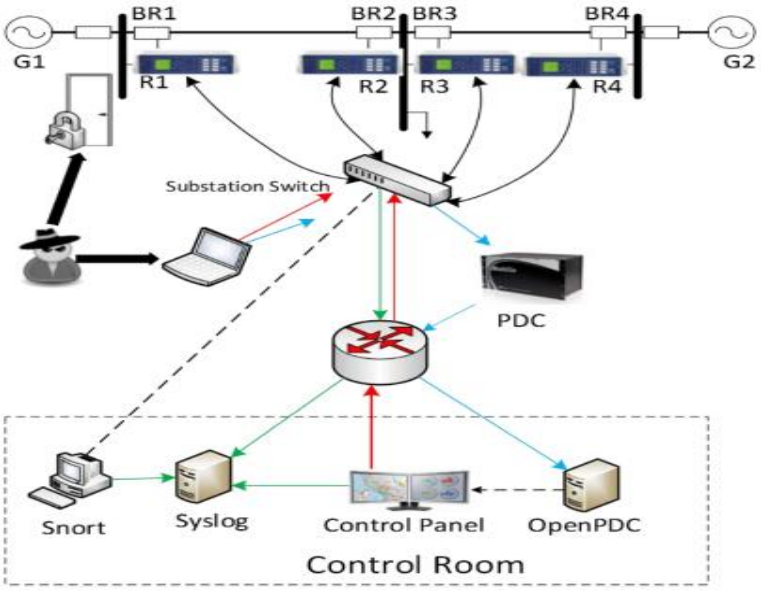
Pelatihan model *machine learning* diilustrasikan seperti pada gambar 3.2 dimana penulis mengambil data melalui *flow dcs* untuk melakukan *model training* dengan model *neural network*. Setelah itu, model dapat tentunkan *traffic* apa yang terjadi setelah data diolah. Sesuai dengan gambar 3.2 diatas input berasal dari *dataset flow dcs* pada *power*

system attack dengan output berupa penentuan kondisi berupa *traffic serangan* atau *traffic normal*.

3.5 Pengumpulan dan Eksplorasi Data

Pada tahap pengumpulan data berasal dari *dataset traffic* dcs dari Missisippi State University dan Oak Ridge National Laboratory pada tahun 2014. *Dataset* digunakan untuk melatih (*training*) model *neural network. Dataset* yang didapat dalam bentuk csv dengan 5 jenis traffic yang diberikan. Dimana dari 5 jenis traffic tersebut, kita membagi menjadi 41 skeanario.

Dataset diberikan dalam bentuk klasifikasi biner (traffic serangan dan *traffic* bukan serangan) dan 41 kelas (41 skenario yang dibuat). Dataset diterima penulis dalam bentuk *black box testing.* Menurut sumber *dataset* tersebut didapatkan pada sebuah arsitektur jaringan sistem kontrol industri seperti pada gambar 3.3.



Gambar 3. 3 Arsitektur jaringan sistem kontrol industri (Pan et al., 2015)

Gambar 3.3 menunjukkan konfigurasi kerangka sistem tenaga yang digunakan dalam menghasilkan skenario ini. Dalam diagram jaringan kami memiliki beberapa komponen, pertama, G1 dan G2 adalah pembangkit listrik. R1 hingga R4 adalah *Intelligent Electronic Devices* (IED) yang dapat mengaktifkan atau menonaktifkan *breaker*. *Breakers* ini diberi label BR1 hingga BR4. Kami juga memiliki dua baris. Jalur Satu membentang dari *breaker* satu (BR1) ke *breaker* dua (BR2) dan jalur dua membentang dari *breaker* tiga (BR3) ke pemutus empat (BR4). Setiap IED secara otomatis mengontrol satu *breakers*. R1 mengontrol BR1, R2 mengontrol BR2 dan sesuai dengan itu. IED menggunakan skema perlindungan

jarak yang membuat pemutus arus pada kesalahan yang terdeteksi apakah benar-benar valid atau palsu karena mereka tidak memiliki validasi *internal* untuk mendeteksi perbedaannya. Operator juga dapat secara manual mengeluarkan perintah ke IED R1 hingga R4 untuk secara manual membuat pemutus BR1 melalui BR4 trip. Pengabaian manual digunakan saat melakukan pemeliharaan pada saluran atau komponen sistem lainnya.

128 fitur dijelaskan dalam tabel 3.1 dibawah ini. Ada 29 jenis pengukuran dari masing-masing *Phasor Measurement Unit* (PMU). PMU adalah perangkat yang digunakan untuk mengukur gelombang listrik pada sebuah jaringan listrik, menggunakan sumber waktu yang sama untuk sinkronisasi. Dalam sistem kami ada 4 PMU yang mengukur 29 fitur untuk total 116 kolom pengukuran PMU. Indeks setiap kolom berupa "R#-Signal Reference" yang menunjukkan jenis pengukuran dari PMU yang ditentukan oleh "R#".

Tabel 3. 1 Parameter Input pada *dataset* (Pan et al., 2015)

Parameter	Deskripsi
PA1:VH – PA3:VH	Fasa A-C Sudut Fasa Tegangan
PM1:V – PM3: V	Fasa A-C Magnitudo Fasa Tegangan
PA4 : IH – PA6:IH	Fasa A-C Sudut Fasa Arus
PM4: I – PM6:I	Fasa A-C Magnitudo Fasa Arus
PA7 :VH – PA9: VH	Pos. -Neg. -Zero Sudut Fasa Tegangan
PM7: V – PM9 : V	Pos. -Neg. -Zero Magnitudo Fasa Tegangan
PA10 : VH - PA12:VH	Pos. -Neg. -Zero Sudut Fasa Arus
PM10: V – PM12: V	Pos. -Neg. -Zero Magnitudo Fasa Arus
F	Frekuensi *Relay*
DF	Frekuensi Delta *Relay*
PA:Z	Impendasi *Relay*
PA:ZH	Sudut Impedansi *Relay*
S	*Status Flag Relay*

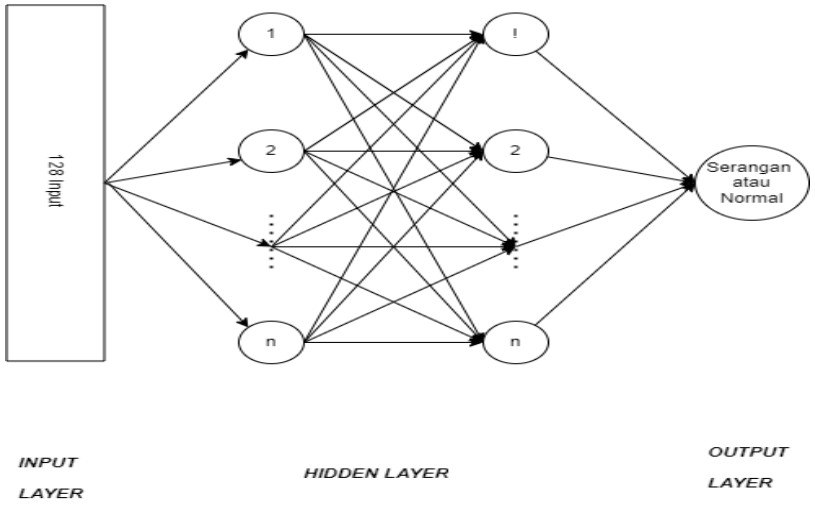
Referensi sinyal dan deskripsi terkait tercantum di bawah ini. Misalnya, R1-PA1:VH berarti sudut fasa tegangan Fase A yang diukur dengan PMU R1. Setelah kolom pengukuran PMU, ada 12 kolom untuk *log panel* kontrol, peringatan *Snort* dan *log* relai dari 4 PMU/relai (relai dan PMU terintegrasi bersama). Kolom terakhir adalah penanda. Tiga digit pertama di sebelah kanan adalah kondisi beban (dalam Megawatt). Tiga digit lainnya di sebelah kirinya adalah lokasi kesalahan, misalnya, "085" berarti kesalahan pada 85% saluran transmisi yang ditentukan oleh deskripsi skenario. Namun, untuk hal-hal yang tidak melibatkan kesalahan,

misalnya “pemeliharaan saluran”, digit ini akan diatur ke 000. Satu atau dua digit paling kiri menunjukkan nomor skenario.

Eksplorasi data digunakan untuk memilah data yang berada diluar jangkauan agar dapat dilakukan pelatihan. Dalam tahapan ini dilakukan pembersihan data dalam satu baris yang memiliki nilai *infinity* dan *not a number* (nan). Dan juga penghapusan satu kolom yang memiliki jenis data yang sama.

3.6 Pembuatan Model *Machine Learning*

Pembuatan model machine learning menggunakan *framework tensorflow*. *Tensorflow* digunakan untuk memudahkan pembuatan model *machine learning*. Pada penelitian kali ini model yang digunakan berjenis *multi layer perceptron (feedforward neural networks)* yang diilustrasikan seperti pada gambar 3.4. Dimana n merupakan jumlah *nodes* pada setiap *hidden layer* .

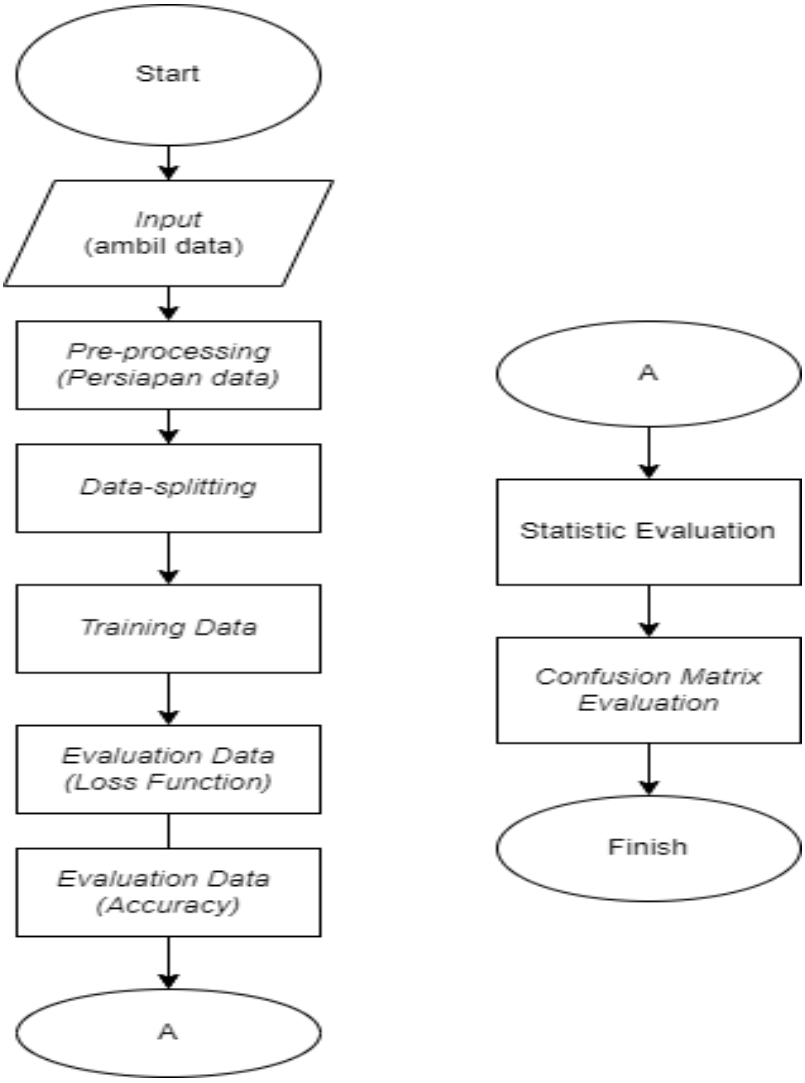


Gambar 3. 4 *Feedforward neural network*

Jumlah *input layer* didapatkan berdasarkan dataset 128 *input nodes* dan setelah itu dilakukan eksplorasi data pada langkah sebelumnya, *hidden layer* pada penelitian ini dirancang dengan 8, 16, 32, 64, dan 128 *hidden layer* dengan fungsi aktivasi *leaky relu* dan *tanh.* Serta *output layer* yang digunakan berjumlah 1 *node boolean* dengan fungsi aktivasi *sigmoid* yang dapat mendeteksi sebuah paket *traffic* pada sistem kontrol industri berupa kondisi *traffic normal* atau *traffic attack.*

3.7 Pelatihan Model *Machine Learning*

Pelatihan model *machine learning* menggunakan *backpropagation*. *Backpropagation* digunakan untuk mendapatkan nilai output dari sebuah model.



Gambar 3. 5 Diagram blok pelatihan model

Pada gambar 3.5 diatas terdapat beberapa proses dalam merancang sebuah sistem yang dideskripsikan sebagai berikut :

a. Input

Memasukkan atau mengambil data *traffic* sistem pada *dataset* yang tersedia. Pada proses ini data dimasukkan ke dalam program untuk dilakukan pelatihan dan evaluasi.

b. Pre processing

Mempersiapkan data agar proses dapat dilakukan dengan baik. Pada tahap ini dilakukan pemilahan data yang mencakup proses pembuangan data diantaranya memeriksa data yang

terdapat inkonsistensi didalamnya, membuang duplikasi data, dan memperbaiki kesalahan pada data.

c. Data-splitting

Membagi persentase jumlah antara data *test* dan data *training*. Pada tahap ini penulis menggunakan rasio (70% : 30%) (Gotama, 2020).

d. *Training Data*

Mengolah data dengan menggunakan *binary classification* dengan susunan 128-8-8-1 (128 *Layer Input*, 8 *Hidden Layer*, 8 *Hidden Layer*, dan 1 *Output)*. Dan berupa variasi hidden layer yaitu 16, 32, 64, dan 128 *hidden layer*. menggunakan algoritma pelatihan *backpropagation.* 128 *nodes* pada *input layers* berdasarkan jumlah kolom (karakteristik *dataset*) pada *dataset* yang digunakan.

e. Evaluation Data (Loss Function)

Mengevaluasi data *loss function* dan memvisualisasikan berupa gambar grafik. Grafik *Loss function* pada training dan evaluasi yang membentuk konvergen menunjukkan pelatihan yang tidak *overfitting* atau *underfitting*. Pada proses ini merupakan *binary crossentropy.*

f. *Evaluation Data (Binary Accuracy)*

Mengevaluasi data *binary accuracy* dan mwmvisualisasikan berupa gambar grafik. Grafik *Binary accuracy* pada training dan evaluasi yang membentuk konvergen menunjukkan pelatihan yang tidak *overfitting* atau *underfitting.*

g. Statistic Evaluation

Menghitung secara statistik sebuah model dengan nilai *precision, recall, dan f1 score*.

h. Confusion Matrix Evaluation

Mengevaluasi model yang kita rancang menggunakan sebuah matriks kebingungan.

3.8 Evaluasi Model *Machine Learning*

Evaluasi model *machine learning* digunakan model dengan *loss function* (*binary crossentropy*) terendah dan akurasi tertinggi (*binary accuracy*). Dilakukan evaluasi menggunakan data test hingga didapatkan model yang tidak overfitting dan underfitting. Dilakukan juga Analisa *precision, recall, f1-score* (Niyaz et al., 2016.).

3.9 Analisa Data

Setelah dilakukan serangkaian langkah tersebut kemudian dilakukan pengujian pada dataset yang tidak termasuk *data train* dan *data test* dianalisis untuk mengetahui performa dari model dengan menggunakan *confusion matrix* dan performansi metrik.

3.10 Penarikan Kesimpulan

Penarikan kesimpulan didapat melakukan pelatihan dan menganalisa kesesuain hasil penelitian dengan standar nilai yang ada pada jurnal.

3.11 Penyusunan Laporan

Penyusunan laporan tugas akhir ini dikerjakan sesuai format yang disediakan oleh Departemen Teknik Fisika. Dengan tujuan untuk memberikan hasil kerja yang telah dilakukan peneliti. Serta memberikan referensi dan saran yang akan diberikan kepada penelitian selanjutnya.

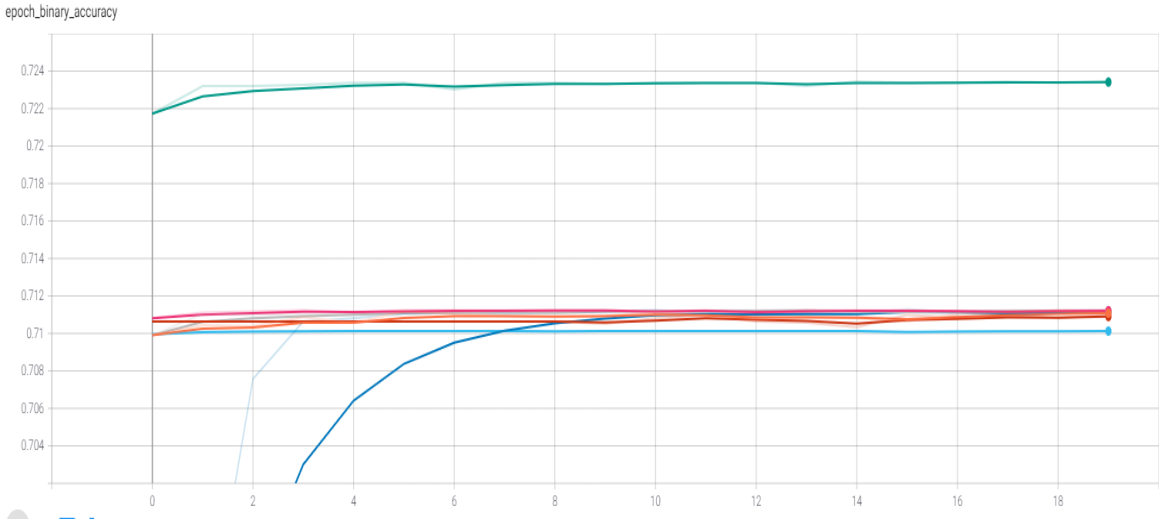
BAB IV
HASIL DAN PEMBAHASAN

4.1 Hasil Pelatihan Model *Machine Learning*

Pelatihan model *machine learning* dilakukan dengan variasi 5 jumlah *hidden layer* yaitu 8, 16, 32, 64, 128. Dilakukan 7 kali percobaan pada setiap *hidden layer*,.

4.1.1 Pelatihan Model *Machine Learning* dengan 8 *hidden layer*

Pelatihan pertama dilakukan 7 kali percobaan dengan 8 *hidden layer*. Didapatkan nilai *binary accuracy* tertinggi pada percobaan 7 dengan nilai 72,34%. Grafik *binary accuracy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.1 dimana sumbu x merupakan jumlah *epoch* pelatihan dan sumbu y merupakan nilai dari *binary accuracy*. Nilai akhir *binary accuracy* dari tiap percobaan ditunjukkan pada table 4.1.

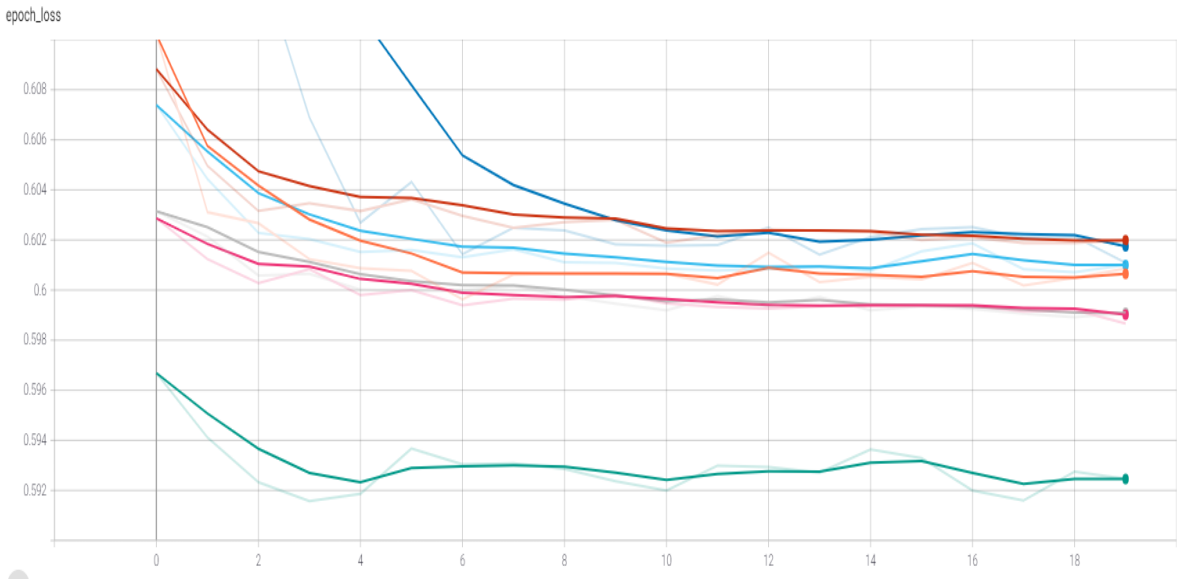


Gambar 4. 1 Grafik *binary accuracy* dengan 8 *hidden layer*

Tabel 4. 1 Tabel persentase hasil nilai akhir *binary accuracy* dengan *8 hidden layer*

Nomor	Nama	*Binary Accuracy*
1	Percobaan 1	71,11 %
2	Percobaan 2	71,10 %
3	Percobaan 3	71, 12 %
4	Percobaan 4	71,12 %
5	Percobaan 5	71,12 %
6	Percobaan 6	71,01 %
7	Percobaan 7	72, 34 %

Setelah dilakukan pelatihan pertama dalam mencari nilai *binary accuracy*, kemudian mencari nilai losses terendah pada percobaan 7 dengan nilai 0,5925. Grafik *binary crossentropy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.2 dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *binary crossentropy*. dan dan nilai akhir *binary crossentropy* dari tiap pelatihan ditunjukkan pada table 4.2.

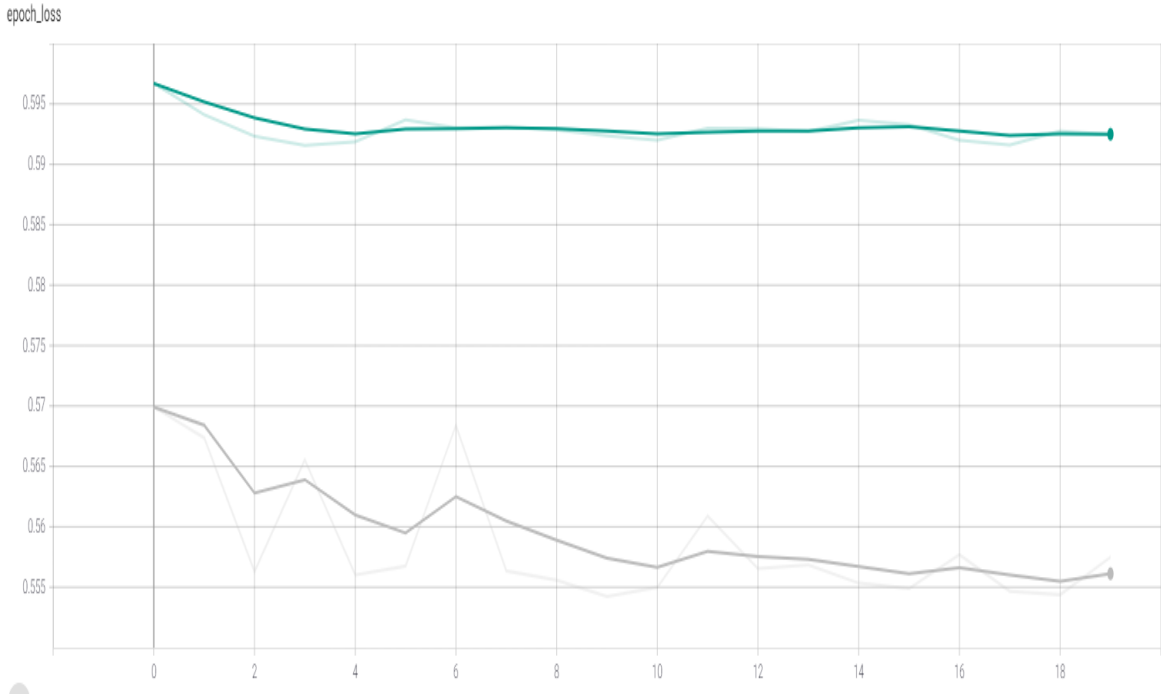


Gambar 4. 2 Grafik *binary crossentropy* dengan 8 *hidden layer*

Tabel 4. 2 Tabel hasil nilai akhir *binary crossentropy* dengan *8 hidden layer*

Nomor	Nama	*Binary Crossentropy*
1	Percobaan 1	0.6009
2	Percobaan 2	0.602
3	Percobaan 3	0.5987
4	Percobaan 4	0.5991
5	Percobaan 5	0.6017
6	Percobaan 6	0.6010
7	Percobaan 7	0.5925

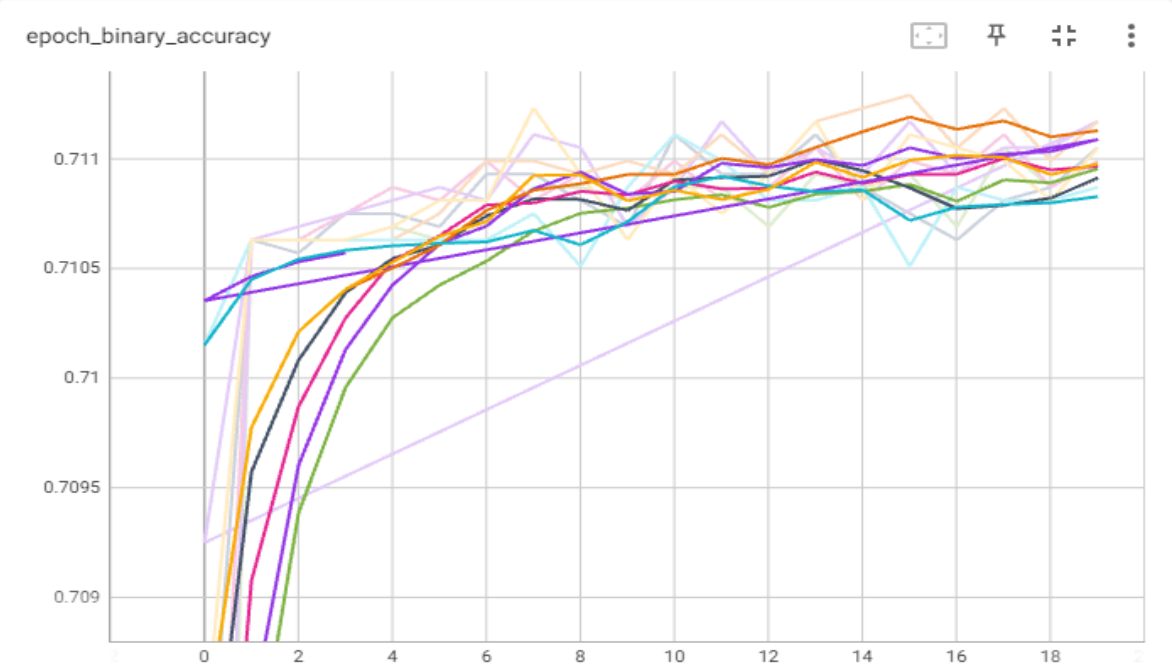
Gambar 4.3 menunjukkan *loss function* pelatihan menggunakan data validasi (garis tipis) dan *training* (garis tebal) dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *loss function*. Seperti terlihat pada gambar *loss function* pelatihan menunjukkan hasil yang konvergen (tidak *overfitting*).



Gambar 4. 3 Grafik *loss function* terendah dengan *8 hidden layer*

4.1.2 Pelatihan Model Machine Learning dengan 16 Hidden Layer

Pelatihan kedua dilakukan selama 7 kali percobaan dengan 16 *hidden layer*. Didapatkan nilai *binary accuracy* tertinggi pada percobaan 7 dengan nilai 71,12%. Grafik *binary accuracy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.4 dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *binary accuracy*. Nilai akhir *binary accuracy* dari tiap percobaan ditunjukkan pada tabel 4.3

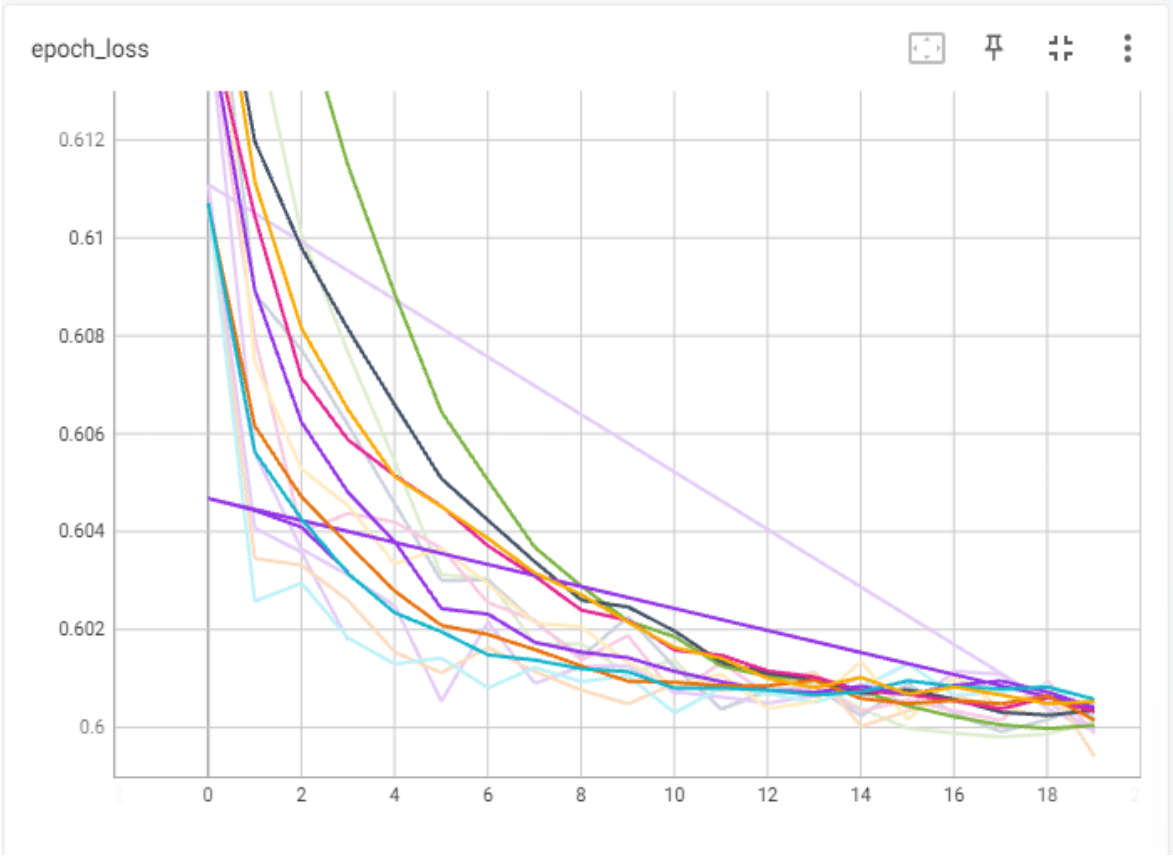


Gambar 4. 4 Grafik *binary accuracy* dengan 16 *hidden layer*

Tabel 4. 3 Tabel persentase hasil nilai akhir *binary accuracy* dengan 16 *hidden layer*

Nomor	Nama	*Binary Accuracy*
1	Percobaan 1	71,11 %
2	Percobaan 2	71,11 %
3	Percobaan 3	71,10 %
4	Percobaan 4	71,06 %
5	Percobaan 5	71,12 %
6	Percobaan 6	71,09 %
7	Percobaan 7	72,11 %

Setelah dilakukan pelatihan kedua dalam mencari nilai *binary accuracy*, kemudian mencari nilai *losses* terendah yaitu dengan nilai 0,5994. Grafik *binary crossentropy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.5 dimana sumbu x merupakan jumlah *epoch* pelatihan dan sumbu y merupakan nilai dari *binary crossentropy*. Nilai akhir *binary crossentropy* dari tiap percobaan ditunjukkan pada table 4.4.

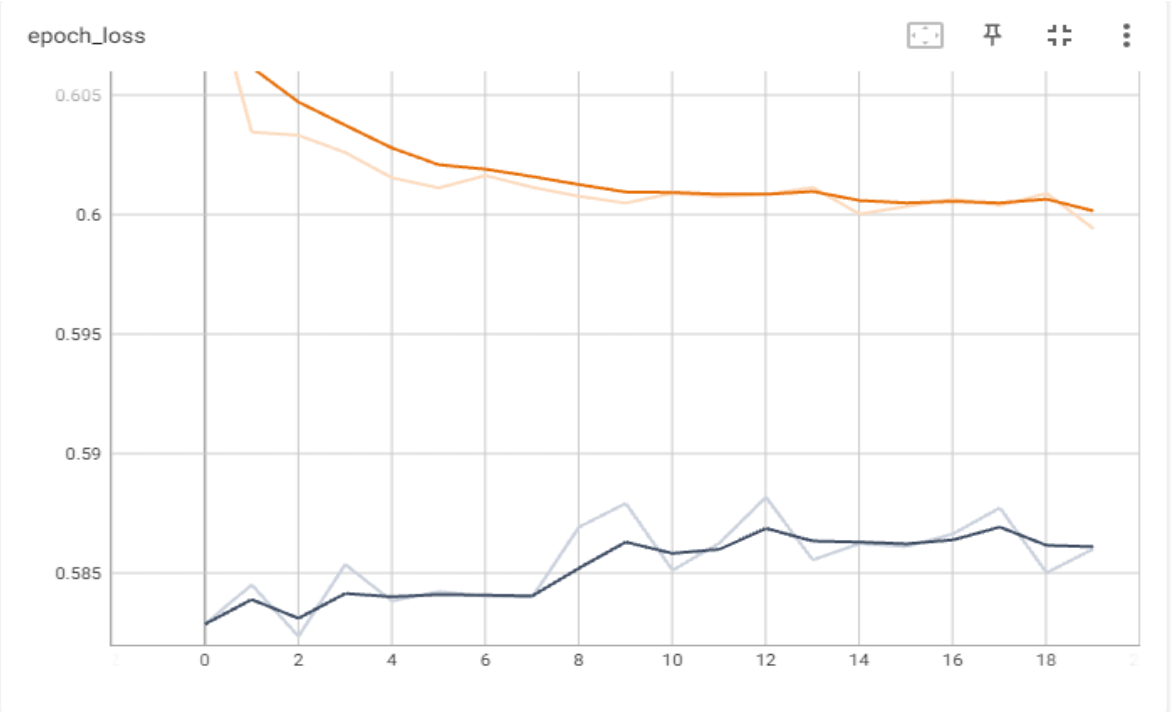


Gambar 4. 5 Grafik *binary crossentropy* dengan 16 *hidden layer*

Tabel 4. 4 Tabel hasil nilai akhir *binary crossentropy* dengan 16 *hidden layer*

Nomor	Nama	*Binary Crossentropy*
1	Percobaan 1	0.6001
2	Percobaan 2	0.6005
3	Percobaan 3	0.5999
4	Percobaan 4	0.6018
5	Percobaan 5	0.5994
6	Percobaan 6	0.6002
7	Percobaan 7	0.6006

Gambar 4.6 menunjukkan *loss function* pelatihan menggunakan data validasi (garis tipis) dan *training* (garis tebal) dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *loss function*. Seperti terlihat pada gambar *loss function* pelatihan menunjukkan hasil yang konvergen (tidak *overfitting*).

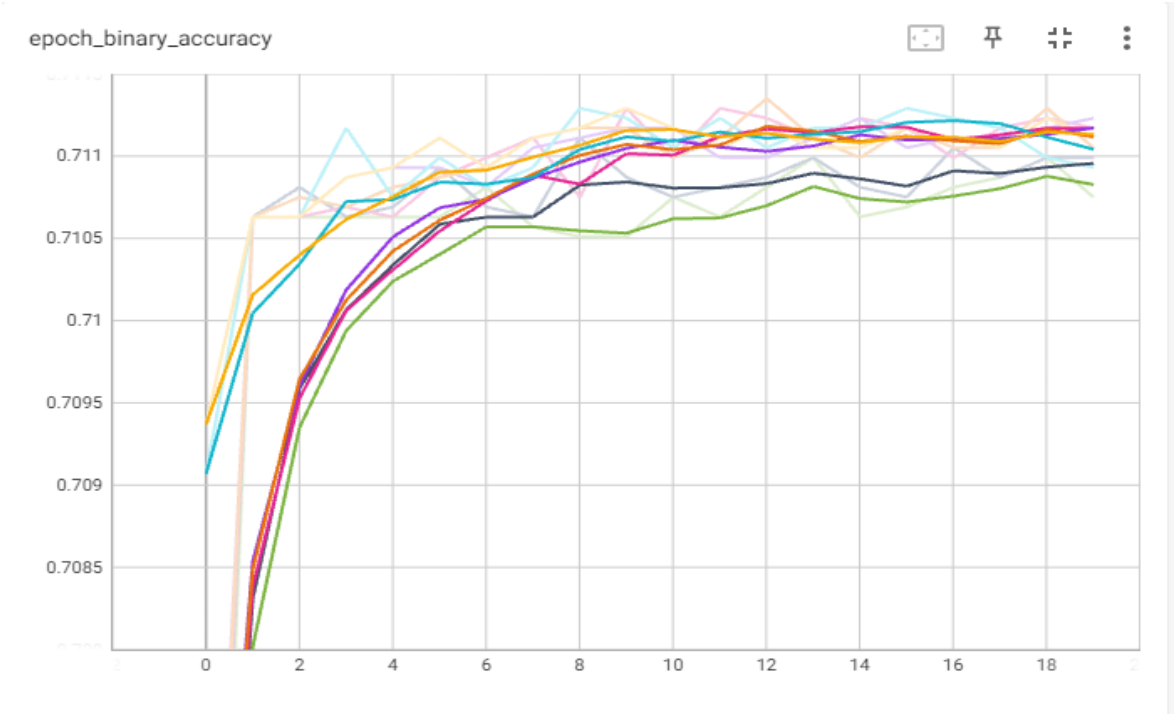


Gambar 4. 6 Hasil *loss function* terendah dengan 16 *hidden layer*

4.1.3 Pelatihan Model Machine Learning dengan 32 Hidden Layer

Pelatihan ketiga dilakukan selama 7 kali percobaan dengan 32 *hidden layer*. Didapatkan nilai *binary accuracy* tertinggi pada percobaan 7 dengan nilai 71,12%. Grafik *binary accuracy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.7 dimana sumbu *x*

merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *binary accuracy*. Nilai akhir *binary accuracy* dari tiap percobaan ditunjukkan pada tabel 4.5

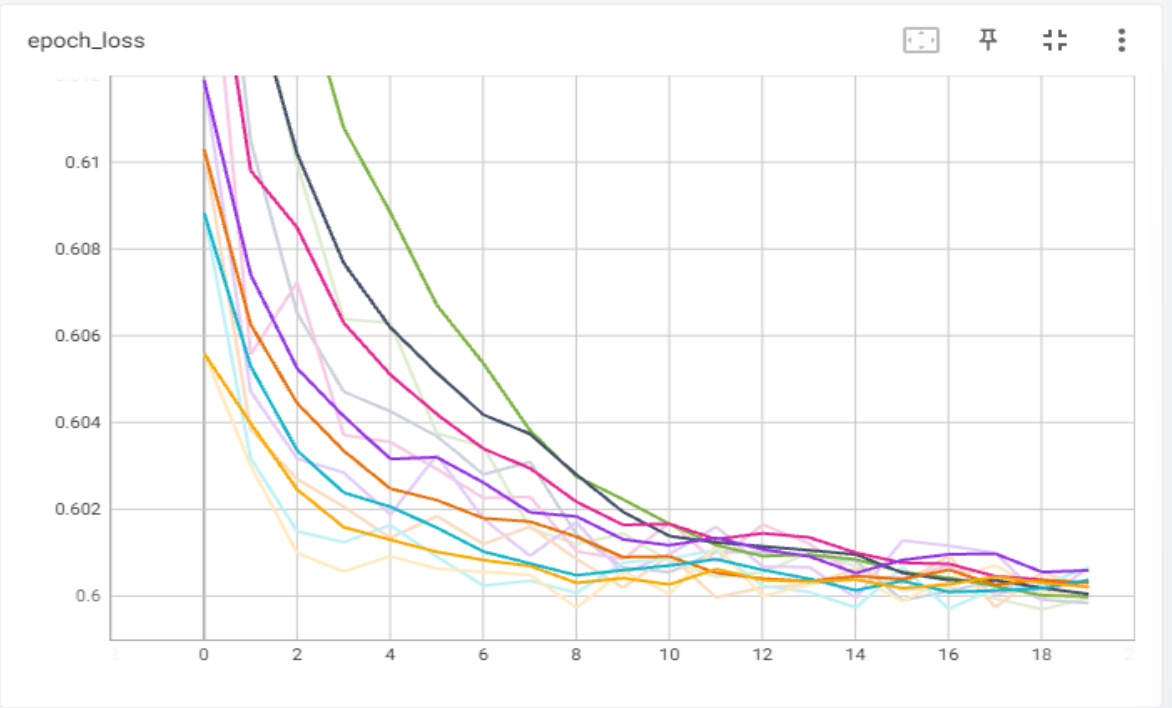


Gambar 4. 7 Grafik *binary accuracy* dengan 32 *hidden layer*

Tabel 4. 5 Tabel persentase hasil nilai akhir *binary accuracy* dengan 32 *hidden layer*

Nomor	Nama	*Binary Accuracy*
1	Percobaan 1	71,08 %
2	Percobaan 2	71,10 %
3	Percobaan 3	71,12 %
4	Percobaan 4	71,12 %
5	Percobaan 5	71,11 %
6	Percobaan 6	71,09 %
7	Percobaan 7	72,11 %

Setelah dilakukan pelatihan kedua dalam mencari nilai *binary accuracy*, kemudian mencari nilai *losses* terendah yaitu dengan nilai 0,5998. Grafik *binary crossentropy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.8 dimana sumbu x merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *binary crossentropy*. Nilai akhir *binary crossentropy* dari tiap percobaan ditunjukkan pada table 4.6.

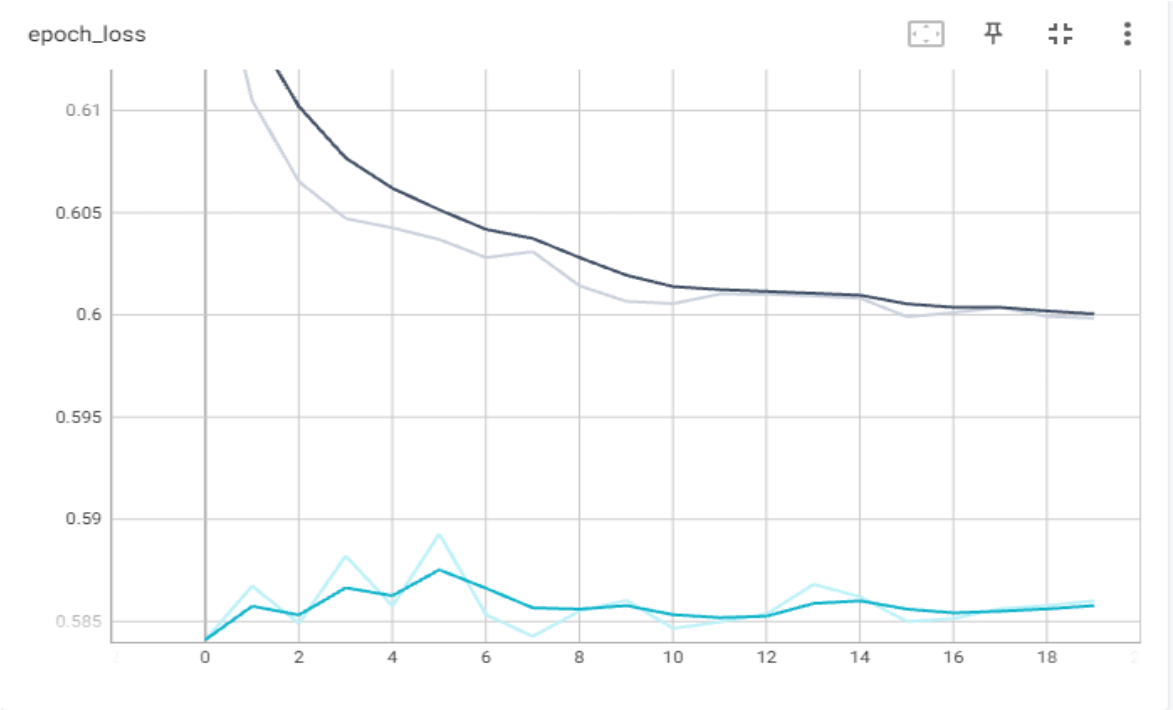


Gambar 4. 8 Grafik *binary crossentropy* dengan 32 *hidden layer*

Tabel 4. 6 Tabel hasil nilai akhir *binary crossentropy* dengan 32 *hidden layer*

Nomor	Nama	*Binary Crossentropy*
1	Percobaan 1	0.6000
2	Percobaan 2	0.5998
3	Percobaan 3	0.5999
4	Percobaan 4	0.6007
5	Percobaan 5	0.6002
6	Percobaan 6	0.6004
7	Percobaan 7	0.6002

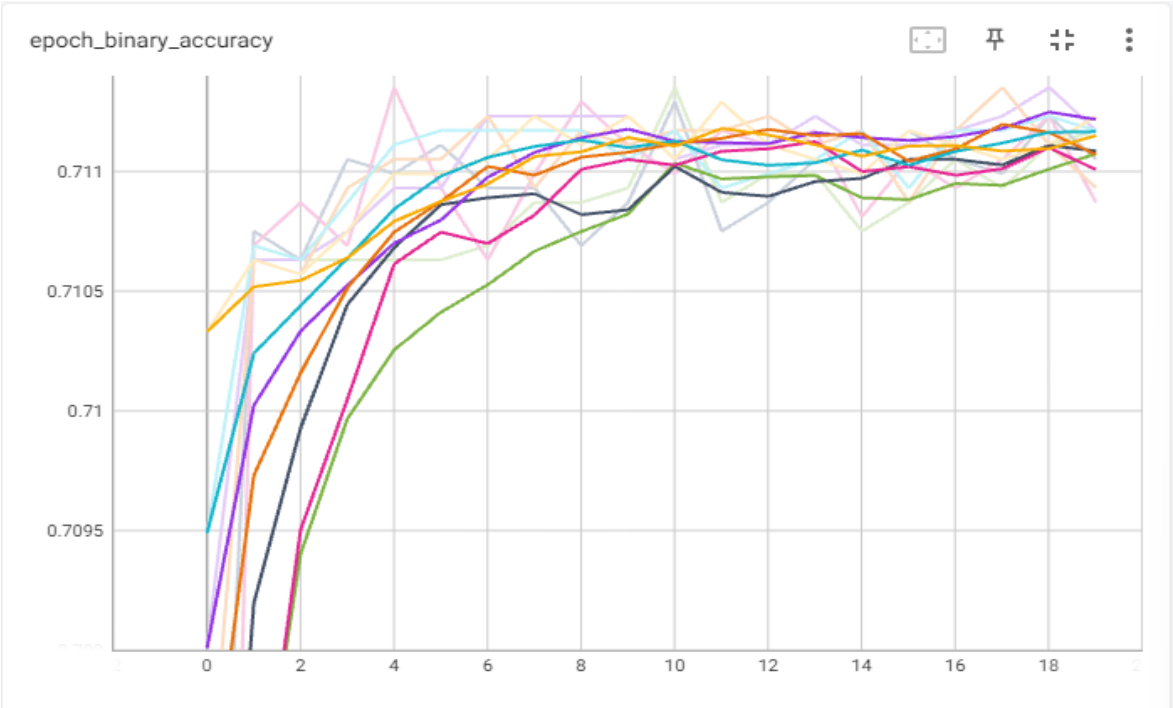
`Gambar 4.9 menunjukkan *loss function* pelatihan menggunakan data validasi (garis tipis) dan *training* (garis tebal) dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *loss function*. Seperti terlihat pada gambar *loss function* pelatihan menunjukkan hasil yang konvergen (tidak *overfitting*)



Gambar 4. 9 Hasil *loss function* terendah dengan 32 *hidden layer*

4.1.4 Pelatihan Model Machine Learning dengan 64 Hidden Layer

Pelatihan keempat dilakukan selama 7 kali percobaan dengan 64 *hidden layer*. Didapatkan nilai *binary accuracy* tertinggi pada percobaan 7 dengan nilai 71,12%. Grafik *binary accuracy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.10 dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *binary accuracy*. Nilai akhir *binary accuracy* dari tiap percobaan ditunjukkan pada tabel 4.7

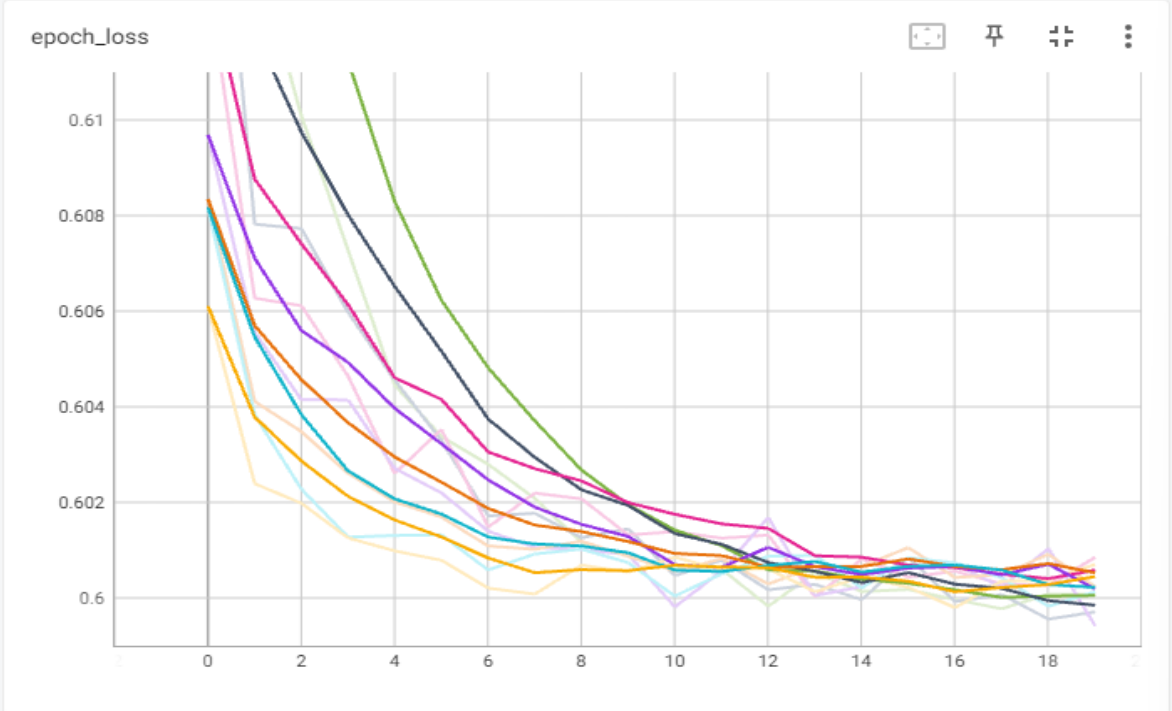


Gambar 4. 10 Grafik *binary accuracy* dengan 64 *hidden layer*

Tabel 4. 7 Tabel persentase hasil nilai akhir *binary accuracy* dengan 64 *hidden layer*

Nomor	Nama	*Binary Accuracy*
1	Percobaan 1	71,12 %
2	Percobaan 2	71,11 %
3	Percobaan 3	71,09 %
4	Percobaan 4	71,12 %
5	Percobaan 5	71,09 %
6	Percobaan 6	71,12 %
7	Percobaan 7	72,12 %

Setelah dilakukan pelatihan kedua dalam mencari nilai *binary accuracy*, kemudian mencari nilai *losses* terendah yaitu dengan nilai 0,5994. Grafik *binary crossentropy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.11 dimana sumbu x merupakan jumlah *epoch* pelatihan dan sumbu y merupakan nilai dari *binary crossentropy*. Nilai akhir *binary crossentropy* dari tiap percobaan ditunjukkan pada tabel 4.8.

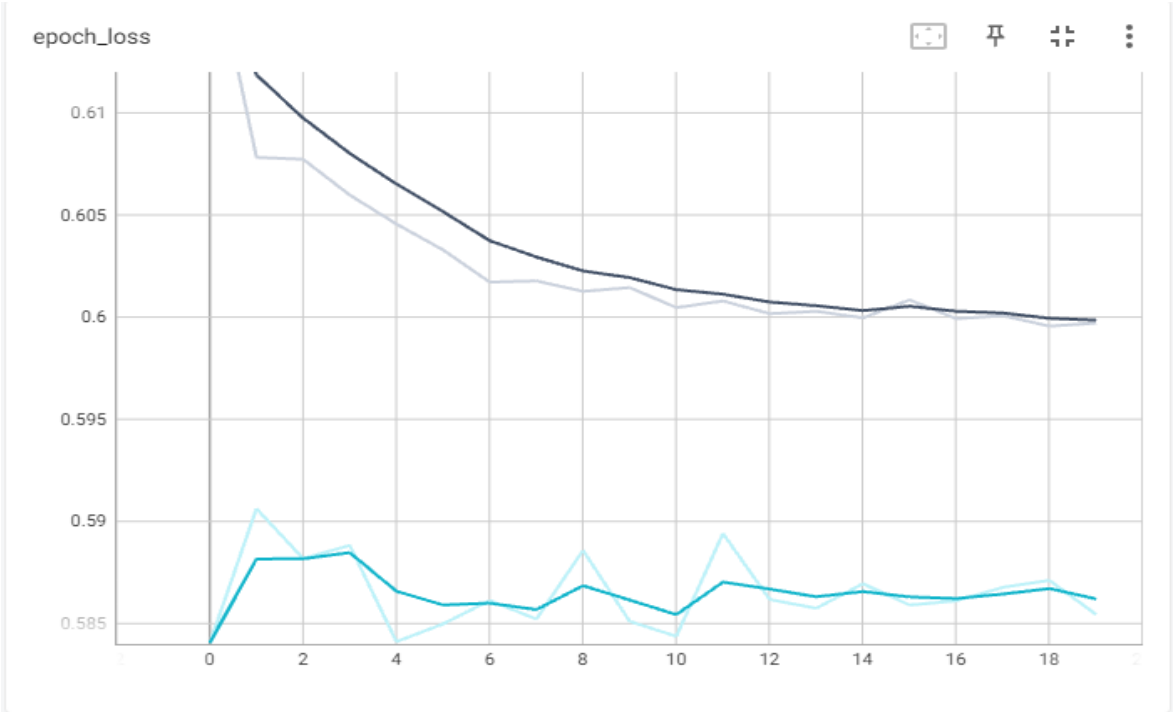


Gambar 4. 11 Grafik *binary crossentropy* dengan 64 *hidden layer*

Tabel 4. 8 Tabel hasil nilai akhir *binary crossentropy* dengan 64 *hidden layer*

Nomor	Nama	*Binary Crossentropy*
1	Percobaan 1	0.6001
2	Percobaan 2	0.5997
3	Percobaan 3	0.6008
4	Percobaan 4	0.5994
5	Percobaan 5	0.6002
6	Percobaan 6	0.6001
7	Percobaan 7	0.6007

Gambar 4.12 menunjukkan *loss function* pelatihan menggunakan data validasi (garis tipis) dan *training* (garis tebal) dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *loss function*. Seperti terlihat pada gambar *loss function* pelatihan menunjukkan hasil yang konvergen (tidak *overfitting*)

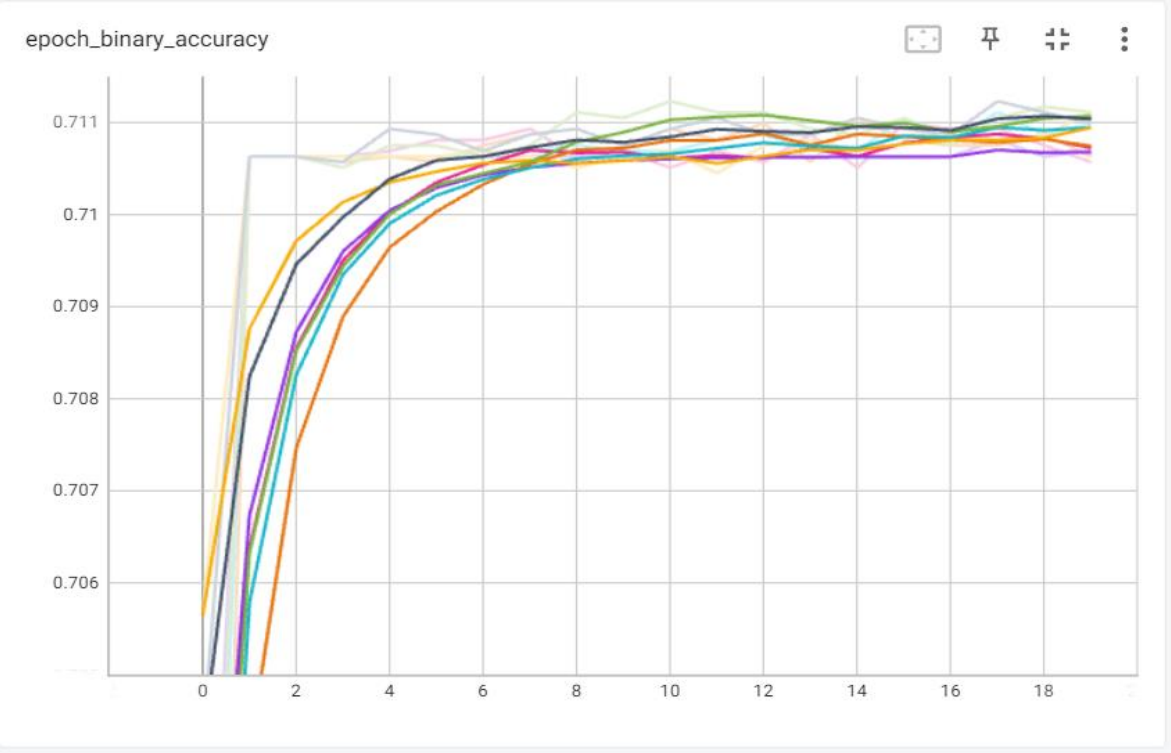


Gambar 4. 12 Hasil *loss function* terendah dengan 64 *hidden layer*

4.1.5 Pelatihan Model Machine Learning dengan 128 Hidden Layer

Selanjutnya dilakukan pelatihan kelima dengan 128 *hidden layer*. Pelatihan kedua dilakukan 7 kali percobaan dengan 128 *hidden layer*. Didapatkan nilai *binary accuracy* tertinggi pada percobaan 7 dengan nilai 71,10%. Grafik *binary accuracy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.13 dimana sumbu *x* merupakan jumlah *epoch*

pelatihan dan sumbu *y* merupakan nilai dari *binary accuracy*. Nilai akhir *binary accuracy* dari tiap percobaan ditunjukkan pada tabel 4.9

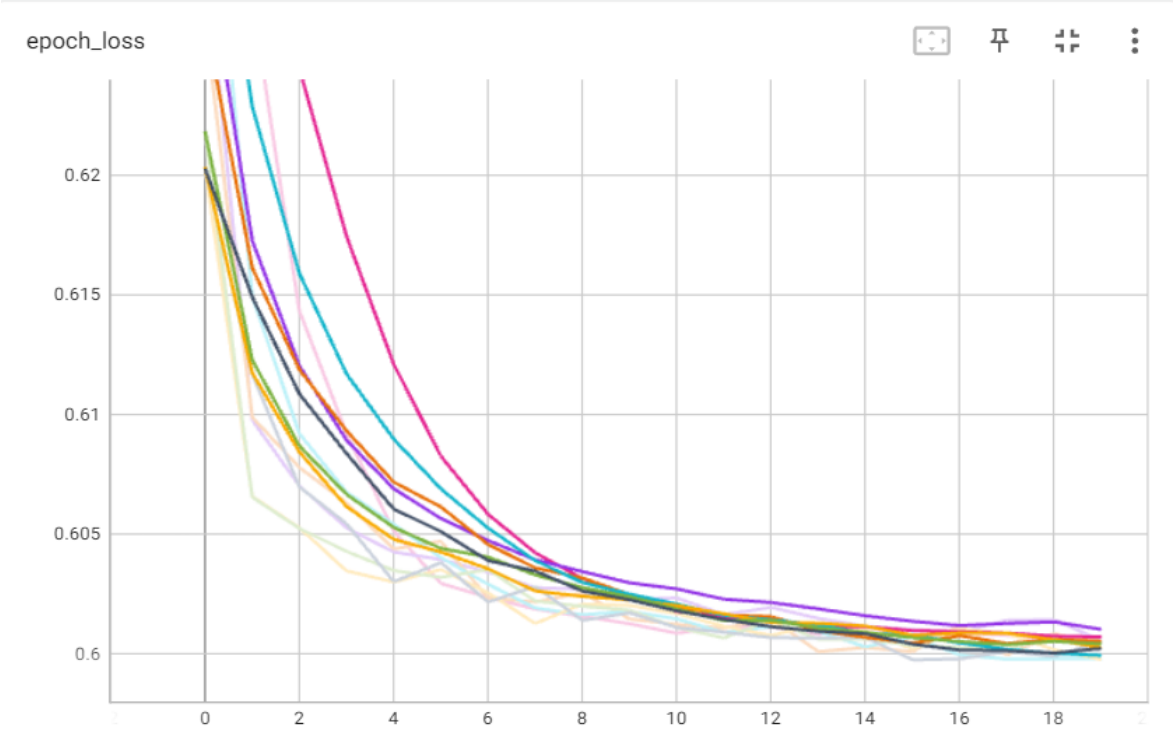


Gambar 4. 13 Grafik *binary accuracy* dengan 128 *hidden layer*

Tabel 4. 9 Tabel persentase hasil nilai akhir *binary accuracy* dengan 128 h*idden layer*

Nomor	Nama	*Binary Accuracy*
1	Percobaan 1	71,06 %
2	Percobaan 2	71,07 %
3	Percobaan 3	71,06 %
4	Percobaan 4	71,10 %
5	Percobaan 5	71,11 %
6	Percobaan 6	71,11 %
7	Percobaan 7	71,10 %

Setelah dilakukan pelatihan kedua dalam mencari nilai *binary accuracy*, kemudian mencari nilai *losses* terendah yaitu dengan nilai 0,5998%. Grafik *binary crossentropy* beberapa hasil pelatihan model ditunjukkan pada gambar 4.14 dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *binary crossentropy*. Nilai akhir *binary crossentropy* dari tiap percobaan ditunjukkan pada table 4.10

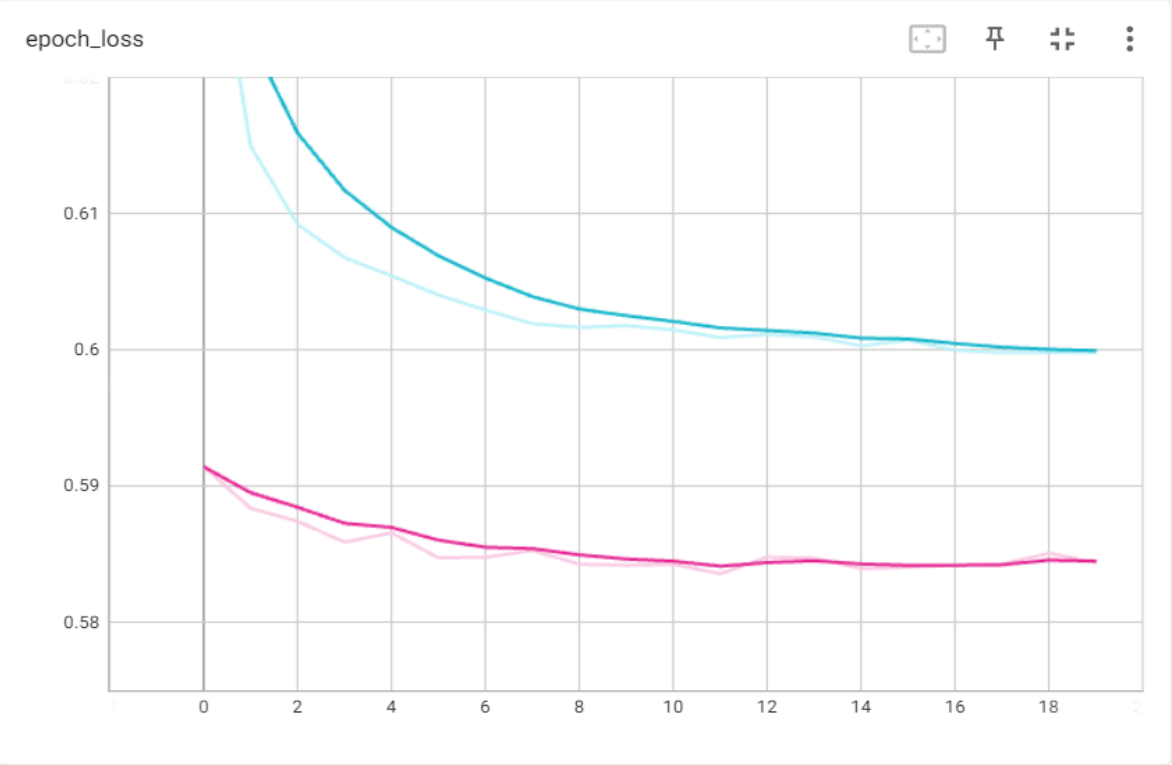


Gambar 4. 14 Grafik *binary crossentropy* dengan 128 *hidden layer*

Tabel 4. 10 Tabel hasil nilai akhir *binary crossentropy* dengan 128 *hidden layer*

Nomor	Nama	*Binary Crossentropy*
1	Percobaan 1	0.6006
2	Percobaan 2	0.6005
3	Percobaan 3	0.6004
4	Percobaan 4	0.5998
5	Percobaan 5	0.5998
6	Percobaan 6	0.6003
7	Percobaan 7	0.6005

Gambar 4.15 menunjukkan *loss function* pelatihan menggunakan data validasi (garis tipis) dan *training* (garis tebal) dimana sumbu *x* merupakan jumlah *epoch* pelatihan dan sumbu *y* merupakan nilai dari *loss function*. Seperti terlihat pada gambar *loss function* pelatihan menunjukkan hasil yang konvergen (tidak *overfitting*)



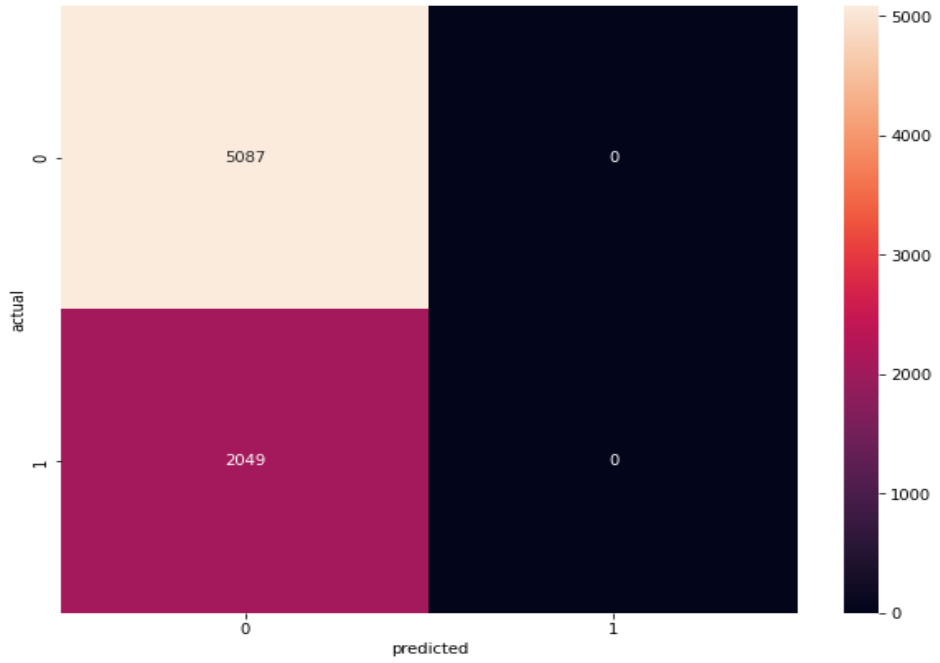
Gambar 4. 15 Hasil *loss function* terendah dengan 128 *hidden layer*

4.2 Pengujian Deteksi Intrusi menggunakan *Machine Learning*

Pengujian deteksi intrusi menggunakan *machine learning* dilakukan dengan variasi 5 jumlah *hidden layer* yaitu 8, 16, 32, 64, 128.

4.2.1 Pengujian Deteksi Intrusi dengan 8 *Hidden Layer*

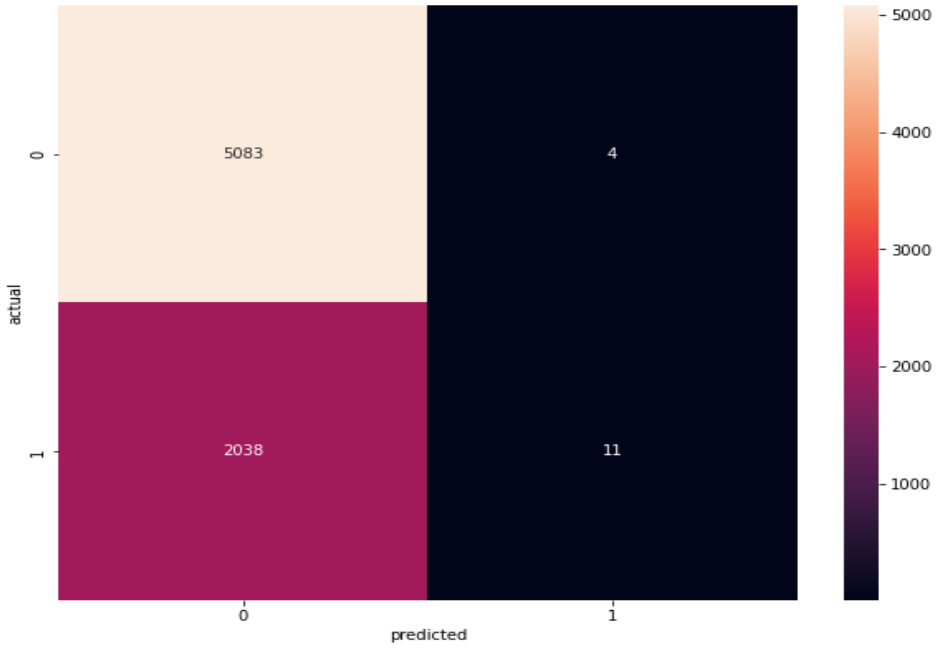
Pengujian pertama dengan 8 *hidden layer* menggunakan 7005 sampel data *traffic* pada sistem kontrol industri didapatkan dari 5087 sampel data traffic normal diprediksi normal (*true positive*), 2049 sampel data traffic attack dideteksi normal (*false negative*), 0 sampel data traffic normal dideteksi sebagai serangan (*true negative*), dan 0 sampel data traffic attack dideteksi sebagai serangan (*false negative*). Gambar 4.16 menunjukkan hasil *confussion matrix* dengan 8 *hidden layer*. *Confussion matrix* dengan ukuran $n \, x \, n$ yang menunjukkan prediksi (*predicted*} dan kenyataan (*actual*) klasifikasi, dimana n adalah angka dari berbagai kelas (Ramsay et al., 2011).



Gambar 4. 16 Hasil *confusion matrix* dengan 8 *hidden layer*

4.2.2 Pengujian Deteksi Intrusi dengan 16 *Hidden Layer*

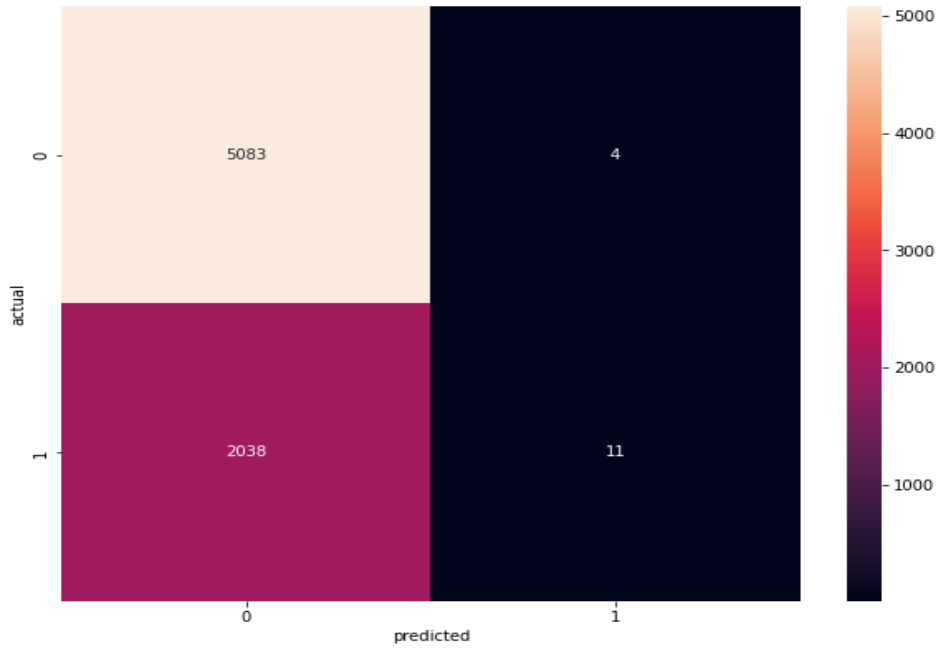
Pengujian kelima dengan 16 *hidden layer* menggunakan 7005 sampel data *traffic* pada sistem kontrol industri didapatkan dari 5083 sampel data traffic normal diprediksi normal (*true positive*), 2038 sampel data traffic attack dideteksi normal (*false negative*), 4 sampel data traffic normal dideteksi sebagai serangan (*true negative*), dan 11 sampel data traffic attack dideteksi sebagai serangan (*false negative*). Gambar 4.17 menunjukkan hasil *confussion matrix* dengan 16 *hidden layer*.



Gambar 4. 17 Hasil *confusion matrix* dengan 16 *hidden layer*

4.2.3 Pengujian Deteksi Intrusi dengan 32 *Hidden Layer*

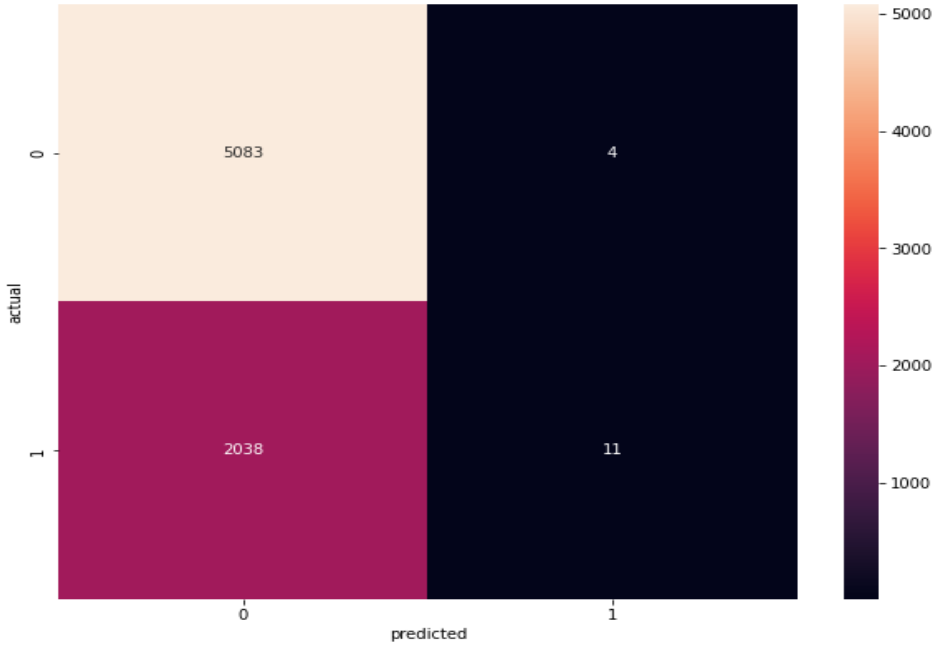
Pengujian kelima dengan 32 *hidden layer* menggunakan 7005 sampel data *traffic* pada sistem kontrol industri didapatkan dari 5083 sampel data traffic normal diprediksi normal (*true positive*), 2038 sampel data traffic attack dideteksi normal (*false negative*), 4 sampel data traffic normal dideteksi sebagai serangan (*true negative*), dan 11 sampel data traffic attack dideteksi sebagai serangan (*false negative*). Gambar 4.18 menunjukkan hasil *confussion matrix* dengan 32 *hidden layer*.



Gambar 4. 18 Hasil *confusion matrix* dengan 32 *hidden layer*

4.2.4 Pengujian Deteksi Intrusi dengan 64 *Hidden Layer*

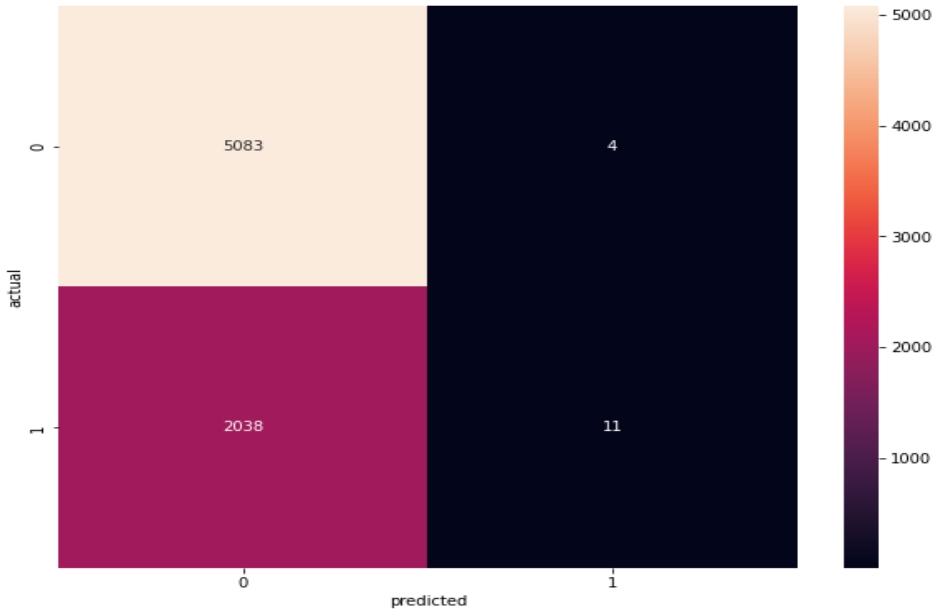
Pengujian kelima dengan 64 *hidden layer* menggunakan 7005 sampel data *traffic* pada sistem kontrol industri didapatkan dari 5083 sampel data traffic normal diprediksi normal (*true positive*), 2038 sampel data traffic attack dideteksi normal (*false negative*), 4 sampel data traffic normal dideteksi sebagai serangan (*true negative*), dan 11 sampel data traffic attack dideteksi sebagai serangan (*false negative*). Gambar 4.19 menunjukkan hasil *confussion matrix* dengan 64 *hidden layer*.



Gambar 4. 19 Hasil *confusion matrix* dengan 64 *hidden layer*

4.2.5 Pengujian Deteksi Intrusi dengan 128 *Hidden Layer*

Pengujian kelima dengan 128 *hidden layer* menggunakan 7005 sampel data *traffic* pada sistem kontrol industri didapatkan dari 5083 sampel data traffic normal diprediksi normal (*true positive*), 2038 sampel data traffic attack dideteksi normal (*false negative*), 4 sampel data traffic normal dideteksi sebagai serangan (*true negative*), dan 11 sampel data traffic attack dideteksi sebagai serangan (*false negative*). Gambar 4.20 menunjukkan hasil *confussion matrix* dengan 128 *hidden layer*.



Gambar 4. 20 Hasil *confusion matrix* dengan 128 *hidden layer*

4.3 Analisa Performansi Model

Setelah dilakukan pelatihan dan pengujian pada model dengan 5 variasi *hidden layer*. Model dilakukan pengolahan (*training*) dengan mencari nilai *precision recall* dan *f1-score* dari data pengujian.

4.3.1 Analisa Performansi Model dengan 8 *Hidden Layer*

Tabel 4. 11 Performansi *nilai precision, recall, dan f1-score* dengan 8 *hidden layer*

Jenis Traffic	Precision	Recall	f1-score
Traffic Normal	0.73	1.00	0.84
Traffic Attack	0.64	0.00	0.01
Average	0.51	0.71	0.60

Setelah dilakukan pelatihan pertama dengan 8 *hidden layer* didapatkan nilai rata-rata sesuai dengan tabel 4.11 dengan nilai rata-rata *precision* sebesar 0,51, rata-rata recall sebesar 0,71, dan rata-rata *f1 score* sebesar 0,60.

4.3.2 Analisa Performansi Model dengan 16 *Hidden Layer*

Tabel 4. 12 Performansi *nilai precision, recall, dan f1-score* dengan 16 *hidden layer*

Jenis Traffic	Precision	Recall	f1-score
Traffic Normal	0.71	1.00	0.83
Traffic Attack	0.73	0.01	0.01
Average	0.72	0.71	0.60

Setelah dilakukan pelatihan pertama dengan 16 *hidden layer* didapatkan nilai rata-rata sesuai dengan tabel 4.12 dengan nilai rata-rata *precision* sebesar 0,72, rata-rata *recall* sebesar 0,71, dan rata-rata *f1 score* sebesar 0,60.

4.3.3 Analisa Performansi Model dengan 32 *Hidden Layer*

Tabel 4. 13 Performansi *nilai precision, recall, dan f1-score* dengan 32 *hidden layer*

Jenis Traffic	Precision	Recall	f1-score
Traffic Normal	0.71	1.00	0.83
Traffic Attack	0.73	0.01	0.01
Average	0.72	0.71	0.60

Setelah dilakukan pelatihan pertama dengan 32 *hidden layer* didapatkan nilai rata-rata sesuai dengan tabel 4.13 dengan nilai rata-rata *precision* sebesar 0,51, rata-rata recall sebesar 0,71, dan rata-rata *f1 score* sebesar 0,60.

4.3.4 Analisa Performansi Model dengan 8 *Hidden Layer*

Tabel 4. 14 Performansi *nilai precision, recall, dan f1-score* dengan 64 *hidden layer*

Jenis Traffic	Precision	Recall	f1-score
Traffic Normal	0.71	1.00	0.83
Traffic Attack	0.72	0.01	0.01
Average	0.72	0.71	0.60

Setelah dilakukan pelatihan pertama dengan 64 *hidden layer* didapatkan nilai rata-rata sesuai dengan tabel 4.14 dengan nilai rata-rata *precision* sebesar 0,72, rata-rata *recall* sebesar 0,71, dan rata-rata *f1 score* sebesar 0,60.

4.3.5 Analisa Performansi Model dengan 128 *Hidden Layer*

Tabel 4. 15 Performansi *nilai precision, recall, dan f1-score* dengan 128 *hidden layer*

Jenis Traffic	Precision	Recall	f1-score
Traffic Normal	0.71	1.00	0.84
Traffic Attack	0.73	0.01	0.01
Average	0.72	0.71	0.60

Setelah dilakukan pelatihan pertama dengan 128 *hidden layer* didapatkan nilai rata-rata sesuai dengan tabel 4.15 dengan nilai rata-rata *precision* sebesar 0,72, rata-rata *recall* sebesar 0,71, dan rata-rata *f1 score* sebesar 0,60.

4.3.6 Hasil Analisa Data Performansi Metrik

Berdasarkan panduan klasifikasi keakuratan dari sebuah *diagnose test* menggunakan AUC (Gorunescu, 2011), didapatkan nilai klasifikasi sebagai berikut :

1. 0.90 – 1.00 : Klasifikasi sempurna;
2. 0.80 – 0.90 : Klasifikasi bagus
3. 0.70 – 0.80 : Klasifikasi cukup bagus
4. 0.60 – 0.70 : Klasifikasi kurang bagus
5. 0.50 - 0.60 : Gagal

Dari hasil lima pelatihan dengan 8, 16, 32, 64, dan 128 *hidden layer* didapatkan nilai klasifikasi diatas dengan rata-rata nilai f1 score berada di klasifikasi kurang bagus. Hal ini

dikaranekan adanya data yang tidak seimbang antara *traffic normal* dan *traffic attack*. Dalam sebuah pemodelan klasifikasi biner (*binary classification)* data yang bagus adalah Ketika dua label kelas data memiliki persentase yang sama yaitu 50% berbanding 50% (Gotama, 2020). Pada tugas akhir ini terjadi ketimpangan data dimana terdapat 7529 *normal data* dan 18284 *attack data*. Pada dataset yang digunakan terlalu banyak jumlah *traffic* berupa *attack data* yang dilatih sehingga model yang penulis buat mendapatkan nilai akurasi yang diklasifikasikan sebagai klasifikasi kurang bagus.

Selain itu, pada eksplorasi data yang dilakukan pada tugas akhir ini, eksplorasi yang dilakukan belum sampai menseleksi fitur-fitur yang tersedia dan penentuan karakteristik data. Hanya sebatas agar data yang dihasilkan tidak divergen saja. Oleh karena itu, pada penelitian selanjutnya diperlukan untuk mengeksplorasi data secara lebih rinci terutama dalam menseleksi fitur-fitur yang tersedia dan dapat menentukan karakteristik data yang ada secara detail.

Hal lain yang memungkinkan nilai f1 score rendah adalah dikarenakan d*ataset* yang digunakan berbentuk *time series* (data yang ada sebelumnya beerkaitan dengan data selanjutnya). Sehingga *neural networks* yang digunakan pada penelitian ini masih belum terlalu baik dalam memodelkan. Oleh karena itu, pada penelitian selanjutnya diharapkan dapat memodelkan menggunakan *time series neural networks (Recurrent Neural Networks)*. Recurrent neural networks menampilkan perbaikan dari *feedforward neural networks* yang memanfaatakan depedensi waktu dan urutan.

Halaman ini sengaja dikosongkan

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Kesimpulan yang didapat pada penelitian ini adalah sebagai berikut :

a) *Machine learning* di sistem deteksi intrusi pada sistem kontrol industri dirancang menggunakan *framework tensorflow* dengan algoritma pembelajaran *backpropagation*. *Neural networks* pada perancangan menggunakan model *feedforward neural networks* dengan susunan neural networks 128 *input parameter* yang tersedia, variasi 5 *hidden layer* yaitu 8, 16, 32, 64, dan 128 *hidden layer*, dan 1 output *Boolean* dengan fungsi aktivasi *sigmoid* dimana hasilnya berupa kondisi *traffic normal* atau *traffic attack*.
b) Keakuratan sistem deteksi intrusi yang dibuat setelah melakukan beberapa kali pelatihan dengan 8, 16, 32, 64, dan 128 *hidden layer* didapatkan keakuratan terbesar sebesar 72,3%, 71,12%, 71,12%, 71,12%, dan 71,10%. Dengan nilai rata-rata *precision* sebesar 0,51, *recall* sebesar 0,71, dan *f1-score* sebesar 0,60 pada 8 *hidden layer* dan nilai rata-rata precision *sebesar* 0,73, *recall* sebesar 0,71, dan *f1 score* sebesar 0,60 pada 16, 32, 64, dan 128 *hidden layer*.

5.2 Saran

Adapun saran pada penelitian kali ini adalah :

a) Penelitian yang dilakukan belum bisa diimplementasi pada plan asli karena *f1 score* nya termasuk dalam kategori klasifikasi kurnag baik.
b) Penelitian yang dilakukan diharapkan dapat menjadi acuan untuk penelitian-penelitian selanjutnya.
c) Eksplorasi data lebih dikembangkan agar dapat mengetahui karakteristik sebuah serangan.
d) Dikarenakan dataset yang ada merupakan *dataset time series* maka perlu digunakan metode *recurrent neural networks* untuk hasil yang lebih baik.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., & Isard, M. (2016). Tensorflow: A system for large-scale machine learning. *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, 265–283.

Alencar, P., & Cowan, D. (n.d.). *The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review Ivens Portugal*.

Anyanwu Matthew, N., & Shiva Sajjan, G. (n.d.). *Comparative Analysis of Serial Decision Tree Classification Algorithms*.

Bachar, A., el Makhfi, N., & Bannay, O. E. L. (2020). *Machine Learning for Network Intrusion Detection Based on SVM Binary Classification Model*.

Brownlee, J. (2016). . *Supervised and unsupervised machine learning algorithms. Machine Learning Mastery*. Http://Machinelearningmastery. Com/Supervised-and-Unsupervised-Machine-Learning-Algorithms.

Caelen, O. (2017). A Bayesian interpretation of the confusion matrix. *Annals of Mathematics and Artificial Intelligence*, *81*(3), 429–450.

Chamou, D., Toupas, P., Ketzaki, E., Papadopoulos, S., Giannoutakis, K. M., Drosou, A., & Tzovaras, D. (2019). Intrusion detection system based on network traffic using deep neural networks. *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 1–6.

Ellis, S. R. (2017). Detecting System Intrusions. *Computer and Information Security Handbook*, 67–107. https://doi.org/10.1016/B978-0-12-803843-7.00005-3

Gorunescu, F. (2011). *Data Mining Concepts, Models and Techniques* (Vol. 12).

Gotama, P. (2020). *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning Edisi 1.4*.

Hassoun, M. H. (1995). *Fundamentals of artificial neural networks*. MIT press.

Holyoak, K. J. (1987). Parallel distributed processing: explorations in the microstructure of cognition. *Science*, *236*, 992–997.

IBM Cloud Education. (2020). *Neural Networks*. Https://Www.Ibm.Com/Cloud/Learn/Neural-Networks.

Janardhanan, P., & Sabika, F. (2015). Effectiveness of support vector machines in medical data mining. *Journal of Communications Software and Systems*, *11*(1), 25–30.

Kumari, R., & Srivastava, S. K. (2017). Machine learning: A review on binary classification. *International Journal of Computer Applications*, *160*(7).

LeNail, A. (2019). NN-SVG: Publication-Ready Neural Network Architecture Schematics. *J. Open Source Softw.*, *4*(33), 747.

Mitchell, T., & McGraw-Hill, M. L. (1997). *Edition*. New York: McGraw-Hill, Inc.

Niyaz, Q., Sun, W., Javaid, A. Y., & Alam, M. (n.d.). *A Deep Learning Approach for Network Intrusion Detection System*.

Nugroho, K. (2019, November). *Confusion Matrix untuk Evaluasi Model pada Supervised Learning*. Https://Ksnugroho.Medium.Com/Confusion-Matrix-Untuk-Evaluasi-Model-Pada-Unsupervised-Machine-Learning-Bc4b1ae9ae3f.

Pan, S., Morris, T., & Adhikari, U. (2015). Developing a hybrid intrusion detection system using data mining for power systems. *IEEE Transactions on Smart Grid*, *6*(6), 3104–3113.

Ramsay, B., Ralescu, A., van der Knaap, E., & Visa, S. (2011). *Confusion Matrix-based Feature Selection. Confusion Matrix-based Feature Selection*. https://www.researchgate.net/publication/220833270

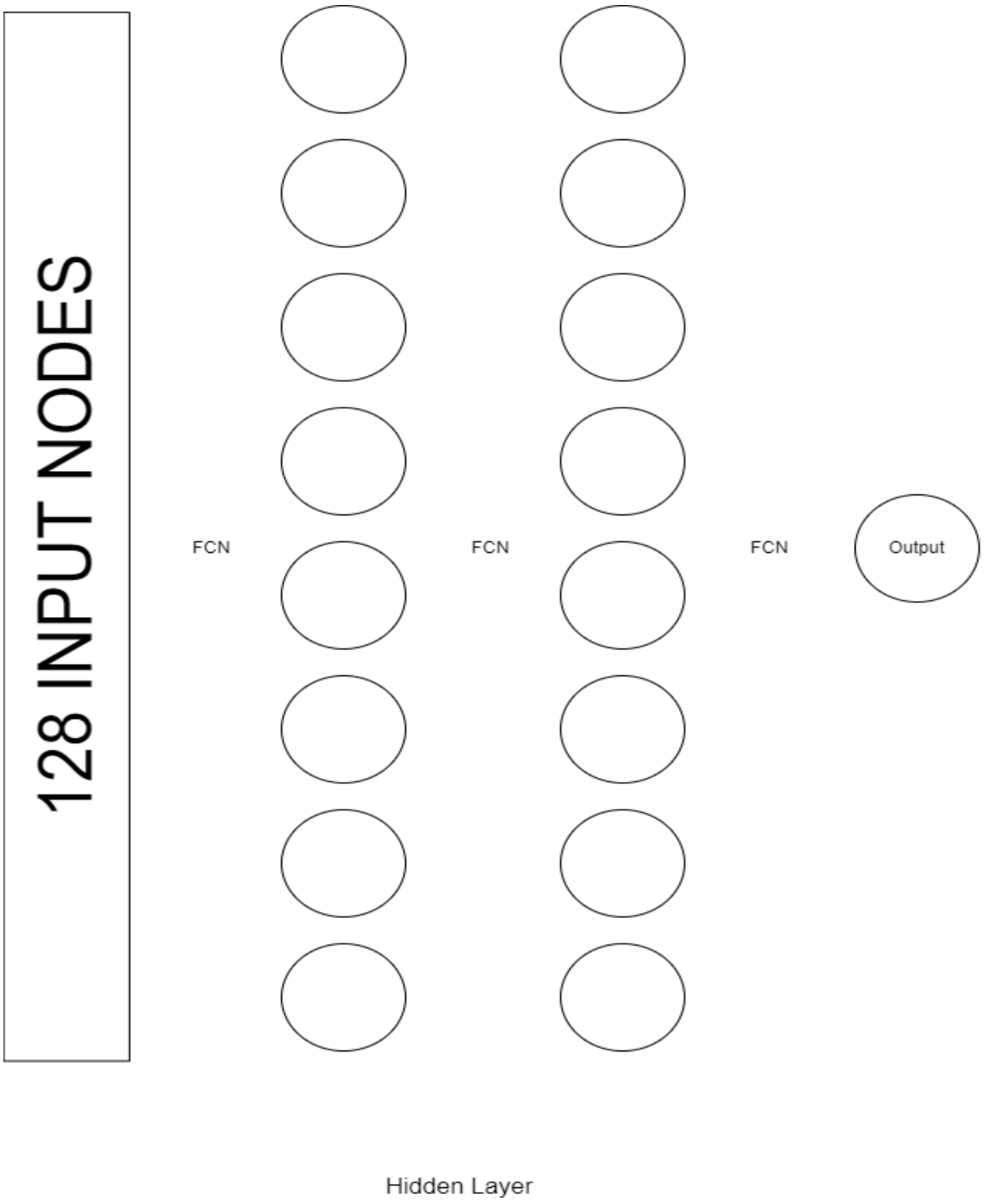
Salo, F., Injadat, M., Nassif, A. B., Shami, A., & Essex, A. (2018). Data mining techniques in intrusion detection systems: A systematic literature review. *IEEE Access*, *6*, 56046–56058.

Sprengers, M., & van Haaster, J. (2016). Organization of #operations. *Cyber Guerilla*, 41–109. https://doi.org/10.1016/B978-0-12-805197-9.00003-6

Stouffer, K., & Falco, J. (2006). *Guide to supervisory control and data acquisition (SCADA) and industrial control systems security*. National institute of standards and technology.

van Veen, F., & Leijnen, S. (2016). The neural network zoo. *The Asimov Institute*.

LAMPIRAN



Ket :

FCN = Fully Connected Nodes

Halaman ini sengaja dikosongkan

BIODATA PENULIS

Mochammad Haidar Dzakalaksana biasa dipanggil Haidar. Penulis dilahirkan di Jakarta 20 Agustus 1999. Penulis merupakan anak kedua dari 3 bersaudara. Penulis bertempat tinggal di Jakarta, Indonesia. Penulis memulai Pendidikan Sekolah Dasar di dua tempat dan wilayah berbeda yaitu SDN Regol 13 Garut kemudian pindah ke Jakarta dan menempuh Pendidikan Sekolah Dasar di SDN Tebet Timur 15 Pagi. Penulis menempuh Pendidikan Sekolah Menengah Pertama (SMP) di SMPN 216 Jakarta. Penulis menempuh Pendidikan Sekolah Menengah Atas (SMA) di SMAN 3 Jakarta. Dan penulis menempuh Pendidikan Sarjana di S1 Teknik Fisika Institut Teknologi Sepuluh Nopember Surabaya. Semasa kuliah penulis aktif berorganisasi sebagai bagian dari Himpunan Mahasiswa Teknik Fisika ITS di Departemen Olahraga dan Kesehatan (saat ini bernama Departemen Minat dan Bakat). Bagi pembaca yang mempunyai kritik, saran, atau ingin berdikusi dengan saya bisa menghubungi saya melalui :

Email : hdzakalaksana@gmail.com