PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) DENGAN ALGORITMA *BACKPROPAGATION* DALAM MEMPREDIKSI KELULUSAN TEPAT WAKTU MAHASISWA

(SKRIPSI)

Oleh

BAGUS STEFANUS SIHOTANG

NPM 1915061051

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

BANDARLAMPUNG

2026

PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) DENGAN ALGORITMA *BACKPROPAGATION* DALAM MEMPREDIKSI KELULUSAN TEPAT WAKTU MAHASISWA

Oleh

BAGUS STEFANUS SIHOTANG

Skripsi

Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK

Pada

Jurusan Teknik Elektro
Fakultas Teknik Universitas Lampung

FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDARLAMPUNG
2026

ABSTRAK

PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) DENGAN ALGORITMA *BACKPROPAGATION* DALAM MEMPREDIKSI KELULUSAN TEPAT WAKTU MAHASISWA

Oleh

BAGUS STEFANUS SIHOTANG

Lembaga pendidikan berperan dalam menghasilkan sumber daya yang berkualitas dari lulusannya, kualitas dan akreditasi menjadi hal yang penting dalam menilai kualitas lembaga. Kelulusan Tepat Waktu (KTW) adalah ketepatan lulus seseorang dalam menyelesaikan masa studinya. KTW dalam Strata-1 di Indonesia adalah 4 tahun atau kurang. KTW merupakan indikator penting bagi mutu pendidikan, berkontribusi pada Inidikator Kinerja Utama dan standar akreditasi institusi. Upaya mengurangi keterlambatan kelulusan sering kali terlambat, sehingga penelitian ini bertujuan untuk membangun model prediktif yang dapat membantu mengidentifikasi dini mahasiswa berisiko tidak lulus tepat waktu. Menggunakan *Artificial Neural Network* (ANN) dengan algoritma *backpropagation* dengan *input* jalur penerimaan, nama prodi, fakultas, dan IPK pada 13.755 data S-1 angkatan 2018 – 2020 dari biro akademik Universitas Lampung. Hasil menunjukan bahwa model ANN dapat diimplementasikan dengan akurasi 50,67%, F1-Score 15,03%, dan *Matthews Correlation Coefficient* (MCC) 0.08 dengan nilai *input* kontribusi paling besar adalah fakultas.

Kata kunci: Statistik kelulusan, IPK, *Deep Learning,* Prediksi Mahasiswa, Pendidikan, Jaringan Syaraf Tiruan atau ANN.

ABSTRACT

IMPLEMENTATION OF ARTIFICIAL NEURAL NETWORK (ANN) WITH BACKPROPAGATION ALGORITHM TO PREDICT STUDENT ON-TIME GRADUATION

By

BAGUS STEFANUS SIHOTANG

Educational institutions play a crucial role in producing high-quality graduates, and their quality and accreditation are important measures of institutional performance. On-time Graduation (OTG) refers to an individual completing their study program within the prescribed period. For bachelor's programs in Indonesia, OTG is defined as four years or less. OTG is an important quality indicator for higher education, contributing to institutional Kei PerFomance Indicators and accreditation standards. Efforts to reduce delayed graduation are often implemented too late, therefore this study aims to develop a predictive model to support early indetification of students at risk of not graduating on time. Using an Artificial Neural Network (ANN) trained with backpropagation algorithm with input features: admission pathways, program name, faculty, and cummulative GPA (in Indonesia: IPK), applied to 13,755 bachelor's records (cohort 2018 - 2020) obtained from Academic Office of Universitas Lampung. The results show the ANN can be implemented with an accuracy of 50.67%, an F1-Score of 15.03%, and Matthews Correlation Coefficient (MCC) of 0.08, with faculty identified as the most important input.

Keywords: Graduation statistics, GPA, Deep Learning, Student Prediction, Education, Artificial Neural Network or ANN.

Judul Skripsi	: **PENERAPAN *ARTIFICIAL NEURAL NETWORK* (ANN) DENGAN ALGORITMA *BACKPROPAGATION* DALAM MEMPREDIKSI KELULUSAN TEPAT WAKTU MAHASISWA**
Nama Mahasiswa	: Bagus Stefanus Sihotang
Nomor Pokok Mahasiswa	: 1915061051
Program Studi	: Teknik Informatika
Fakultas	: Teknik

MENYETUJUI,

1. Komisi Pembimbing

Pembimbing Utama	Pembimbing Pendamping
Dr. Ir.M. Komarudin, S.T., M.T.	**Ir.Titin Yulianti, S.T., M.Eng.**
NIP. 196812071997031006	NIP. 198807092019032015

2. Mengetahui

Ketua Jurusan Teknik Elektro	Ketua Program Studi Teknik Informatika
Herlinawati, S.T., M.T	**Yessi Mulyani, S.T., M.T.**
NIP. 197103141999032001	NIP. 197312262000122001

MENGESAHKAN

1. Tim Penguji

Ketua : **Dr. Ir.M. Komarudin, S.T., M.T.** .....

Sekretaris : **Ir.Titin Yulianti, S.T., M.Eng.** .....

Penguji : **Yessi Mulyani, S.T., M.T.** .....

2. Dekan Fakultas Teknik

Dr. H. Ahmad Herison, S.T., M.T.

NIP. 196910302000031001

Tanggal Lulus Ujian Skripsi : **2 Februari 2026**

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini menyatakan dengan sesungguhnya bahwa skripsi yang berjudul "Penerapan *Artificial Neural Network* (ANN) dengan Algoritma *Backpropagation* dalam Memprediksi Kelulusan Tepat Waktu Mahasiswa" merupakan hasil karya saya sendiri dan bukan merupakan hasil plagiarisme, baik sebagian maupun seluruhnya. Seluruh proses penelitian, analisis, serta penulisan dalam skripsi ini dilakukan berdasarkan pemikiran dan usaha saya sendiri dengan tetap berpedoman pada kaidah, norma, dan etika penulisan karya ilmiah yang berlaku. Apabila dalam penulisan terdapat kutipan atau pendapat orang lain, semuanya telah dicantumkan secara jelas dan lengkap sesuai dengan sumber aslinya dalam Daftar Pustaka. Apabila di kemudian hari terbukti bahwa pernyataan ini tidak benar, maka saya bersedia menerima sanksi akademik sesuai dengan ketentuan yang berlaku.

Bandar Lampung, 23 Februari 2026

Penulis,

Bagus Stefanus Sihotang

NPM. 1915061051

RIWAYAT HIDUP

Penulis lahir di Bekasi pada 25 Mei 2001 sebagai anak bungsu dari tiga bersaudara dari pasangan Wanton Sihotang dan Mardeliana Hutagaol. Pendidikan dasar ditempuh di TK Cemara dan SDN Setia Mekar 03, dilanjutkan ke SMPN 1 Tambun Selatan dan SMAN 1 Tambun Utara. Setelah menyelesaikan SMA pada 2019, penulis melanjutkan ke Program Studi Teknik Informatika, Universitas Lampung, dengan tugas akhir yang berjudul *"Penerapan Artificial Neural Network (ANN) dengan Algoritma Backpropagation dalam Memprediksi Kelulusan Tepat Waktu Mahasiswa."*

Pengalaman kerja dan pengembangan keterampilan menjadi bagian penting dari perjalanan akademik penulis. Pada akhir 2021 hingga awal 2022 penulis menjalani magang sebagai *Frontdesk Officer* di PT TASPEN KC Bekasi (29 Desember 2021–17 Februari 2022), yang menumbuhkan kemampuan komunikasi dan kedisiplinan kerja. Pada Februari 2022 penulis mengikuti program Studi Independen di Alterra Academy dengan fokus *Golang Backend* selama satu semester, kemudian melanjutkan pembelajaran melalui beasiswa kelas di Codepolitan dan pelatihan *Alibaba Cloud Developer* pada periode Januari–Maret 2024. Kegiatan-kegiatan tersebut dimanfaatkan untuk memperdalam kemampuan pemrograman, arsitektur layanan *backend*, dan pemahaman mengenai teknologi *cloud*.

Di luar kegiatan formal, penulis memiliki ketertarikan kuat pada dunia coding, khususnya aspek yang berkaitan dengan pengembangan game. Keinginan untuk menciptakan game sendiri menjadi motivasi utama dalam memilih proyek-proyek pembelajaran dan pelatihan yang diikuti. Penulis dikenal sebagai pribadi yang cepat belajar dan antusias mencoba hal baru; setiap kesempatan pelatihan atau proyek dipandang sebagai sarana untuk menambah kompetensi teknis sekaligus menyiapkan diri menghadapi tantangan industri perangkat lunak. Dengan kombinasi pengalaman praktis dan komitmen untuk terus berkembang, penulis bertekad menerjemahkan kemampuan yang dimiliki menjadi karya nyata di bidang *game development* dan riset teknologi.

PERSEMBAHAN

Puji syukur saya panjatkan ke hadirat Tuhan Yesus Kristus yang telah memberikan rahmat, berkat, dan kasih sayang-Nya sehingga saya dapat menyelesaikan skripsi ini. Skripsi ini saya persembahkan kepada orang-orang yang saya cintai dan hormati, yaitu:

Bapak dan Ibu

Ibu saya tercinta yang selalu berjuang gigih demi keluarga dan terus memberi dukungan tanpa henti. Kepada beliau saya mengucapkan terima kasih atas cinta, doa, dan pengorbanan yang tiada tara. Ayah saya tercinta yang telah mendahului saya; semoga beliau selalu meridhai kelulusanku dan selalu berbahagia di sisi Tuhan. Keterlambatan saya dalam menyelesaikan studi ini merupakan penyesalan tersendiri, namun saya berharap kelulusan ini menjadi kebanggaan bagi keluarga kami.

Kepala Program Studi dan Dosen Pembimbing

Kepada Bapak/Ibu Kepala Program Studi Teknik Sipil dan dosen pembimbing saya, saya ucapkan terima kasih atas bimbingan, ilmu, dan arahan yang telah diberikan dengan penuh kesabaran. Tanpa dukungan serta arahan dari Bapak/Ibu, penyusunan skripsi ini tidak akan terselesaikan dengan baik. Saya sangat menghargai setiap waktu dan perhatian yang telah mereka curahkan.

Teman-teman Seperjuangan

Terima kasih saya haturkan kepada teman-teman seperjuangan yang selalu hadir mendukung saya dalam suka maupun duka selama masa-masa akhir perkuliahan ini. Kebersamaan dan semangat mereka telah menjadi kekuatan bagi saya dalam menyelesaikan studi serta skripsi ini.

MOTTO

"If somebody offers you an amazing opportunity but you are not sure you can do it, say yes — then learn how to do it later!"

(Richard Branson)

"Songon na adong, marsiapsip tu jolo"

(Pepatah Batak)

SANWACANA

Puji syukur ke hadirat Tuhan Yang Maha Esa karena dengan limpahan rahmat-Nya penulis dapat menyelesaikan penelitian yang berjudul "Penerapan *Artificial Neural Network* (ANN) dengan Algoritma *Backpropagation* dalam Memprediksi Kelulusan Tepat Waktu Mahasiswa". Penelitian ini melibatkan pengumpulan data, pemodelan, pelatihan jaringan, dan implementasi perangkat lunak; proses tersebut tidak lepas dari bimbingan akademik dan dukungan teknis dari banyak pihak. Penulis mengucapkan terima kasih yang sebesar-besarnya kepada:

1. Kedua orang tua dan kakak – kakak penulis yang selalu memberikan dukungan dan doa habis – habisan.
2. Ibu Prof. Dr. Ir. Lusmeilia Afriani, D.E.A., I.P.M, selaku Rektor Universitas Lampung.
3. Bapak Dr. H. Ahmad Herison, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Lampung.
4. Ibu Herlinawati, S.T.,M.T., selaku Ketua Jurusan Teknik Elektro, Universitas Lampung.
5. Ibu Yessi Mulyani, S.T.,M.T., selaku Ketua Program Studi Teknik Informatika, Universitas Lampung sekaligus penguji saya dalam menyelesaikan tugas akhir saya.
6. Bapak Dr. Ir.M. Komarudin, S.T., M.T., selaku pembimbing utama yang telah memberikan dukungan, arahan dalam menyelesaikan penelitian ini.
7. Ibu Ir.Titin Yulianti, S.T., M.Eng., selaku pembimbing pendamping yang juga turut memberikan dukungan serta arahan dalam menyelesaikan penelitian ini.
8. Seluruh *civitas* akademik Jurusan Teknik Elektro;

9. Keluarga seperjuangan saya, Mahasiswa Teknik Elektro angkatan 2019 Universitas Lampung yang selalu membimbing selama perkuliahan.
10. Sahabat – sahabat saya di titik penghunjung masa studi saya: Caesar, Anggi, Sarah, Ang, Opet, Kunyang, Alwan, Komring, Ilham dan yang lainnya yang tidak bisa saya tuliskan satu persatu tanpa mengurangi rasa hormat.

Semoga laporan penelitian ini dapat berguna dan bermanfaat bagi kita semua. Oleh karena itu, saran serta kritik yang membangun sangat diharapkan dari pembaca.

Bandar Lampung, 23 Februari 2026
Penulis,

Bagus Stefanus Sihotang
NPM. 1915061051

DAFTAR ISI

Halaman

DAFTAR TABEL

DAFTAR GAMBAR

I. PENDAHULUAN

1.1. Latar Belakang

Salah satu indikator yang membuat SDM itu berkualitas dan berkompeten di dunia perkuliahan adalah tingkat kelulusan mahasiswa tepat waktu dan pencapaian Indeks Prestasi Kumulatif (IPK) yang tinggi. Kelulusan tepat waktu dan IPK biasa dijadikan sebagai tolak ukur dalam menilai bagus tidaknya suatu instansi perguruan tinggi. Menurut penelitian dari Departemen Pendidikan Nasional, peningkatan kualitas pendidikan tinggi sangat bergantung pada berbagai faktor, termasuk kinerja akademik mahasiswa dan sistem pendukung akademik yang tersedia [1].

Salah satu permasalahan yang sedang dihadapi oleh banyak perguruan tinggi saat ini adalah rendahnya persentase mahasiswa yang lulus tepat waktu dan memiliki IPK (Indeks Prestasi Kumulatif) di atas 3.00. Berdasarkakn data dari Kementrian Riset, Teknologi dan Pendidikan Tinggi, dari data yang diberikan untuk tahun 2021 hanya sekitar 49% mahasiswa yang mampu menyelesaikan studi mereka tepat waktu yaitu 4 tahun lebih, dan sekitar 4% dari seluruh mahasiswa yang ada di Indoensia memiliki IPK 3.00 atau kurang (data didapatkan dari data perhitungan kasar). Kondisi ini memerlikan sistem yang dapat membantu pihak akademik dalam memantau dan meningkatkan kinerja akademik yang dikelolanya [2].

Terdapat peluang besar untuk mengembangkan alat yang dapat membantu dalam memprediksi dan meningkatkan kinerja akademik mahasiswa. Model *deep learning* mampu memberikan prediksi yang akurat dalam berbagai domain, termasuk pendidikan. Implementasi teknologi ini dalam sistem pendidikan tinggi dapat menjadi solusi efektif untuk mengatasi permasalahan rendahnya tingkat kelulusan tepat waktu dan pencapaian IPK tinggi.

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, dapat dirumuskan masalah penelitian, yaitu:

1. Faktor apa yang berpengaruh dalam kelulusan tepat waktu mahasiswa?
2. Bagaimana penerapan ANN dengan *backpropagation* untuk memprediksi kemungkinan mahasiswa lulus tepat waktu?
3. Bagaimana kinjerja model prediksi ANN yang menggunakan *backpropagation* dalam memproyeksikan mahasiswa yang akan lulus tepat waktu?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah diatas maka tujuan penelitian ini adalah untuk mengetahui sebagai berikut:

1. Mengidentifikasi fitur – fitur yang paling berpengaruh dalam prediksi kelulusan mahasiswa dengan menggunakan *Mutual Information.*
2. Mengimplementasikan metode ANN dengan *backpropagation* untuk prediksi kelulusan tepat waktu mahasiswa.
3. Melakukan eksplorasi skenario konfigurasi untuk ANN kelulusan tepat waktu.

1.4. Batasan Masalah

Batasan masalah pada penelitian ini adalah sebagai berikut:

1. Hanya menggunakan data dari satu sumber berisi angkatan 2018 – 2023 dengan 11 atribut yang diperoleh dari akhir 2024.
2. Hasil prediksi tidak dapat divalidasi secara langsung karena label aktual ketepatan waktu kelulusan mahasiswa angkatan 2021–2023 belum tersedia.

1.5. Sistematika Penulisan

Adapun sistematika penulisan laporan penelitian ini adalah sebagai berikut:

1) PENDAHULUAN

Permasalahan Indikator Kinerja Utama (IKU) yang menjadi tolak ukur mutu dalam perguruan tinggi, khususnya terkait persentase kelulusan mahasiswa tepat waktu. Penelitian akan berfokus pada penerapan sistem dalam memprediksi mahasiswa yang berpotensi lulus tidak tepat waktu menggunakan AI berbasis *deep learning*.

2) TINJAUAN PUSTAKA

Membahas konsep – konsep dasar yang mendukung penelitian, dimulai dari metode yang akan digunakan sebagai dasar dari penelitian, dan alat – alat pendukung penelitian seperti bahasa pemrograman yang digunakan.

3) METODOLOGI PENELITIAN

Menjelaskan tentang tahapan penelitian yang mengikuti metodologi CRISP-DM. Tahapan yang dibahas meliputi *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, dan *Evaluation*. Setiap tahap dijelaskan dengan alur kerja dan tujuan yang jelas untuk memberikan pemahaman mengenai langkah – langkah yang dilakukan dalam mengembangkan model prediksi kelulusan. Selain itu, dijelaskan pula alat dan bahan yang digunakan dalam penelitian, serta jadwal yang ditargetkan untuk penyelesaian setiap tahapan penelitian hingga mencapai hasil yang diinginkan.

4) HASIL DAN PEMBAHASAN

Tahapan – tahapan CRISP-DM dalam penelitian. Proses dimulai dari perencanaan sistem, penyiapan data yang relevan untuk penelitian, pengelolaan data, hingga pengembangan sistem yang mencakup pelatihan model kecerdasan buatan (AI), evaluasi dan pengaplikasian model.

5) KESIMPULAN DAN SARAN

Kesimpulan disampaikan berdasarkan pencapaian tujuan yang telah ditetapkan, diikuti dengan saran untuk pengembangan lebih lanjut yang dapat dilakukan guna meningkatkan kinerja model atau memperluas cakupan penelitian. Saran tersebut mencakup perbaikan metode pemodelan, pemanfaatan fitur tambahan, atau pengembangan sistem yang lebih komprehensif lainnya.

II. TINJAUAN PUSTAKA

2.1. Indikator Kinerja Utama (IKU)

IKU adalah bagian dari kebijakan yang dirancang untuk meningkatkan mutu pendidikan tinggi di Indonesia melalui evaluasi berbasis capaian. IKU bertujuan memastikan perguruan tinggi menghasilkan lulusan yang berdaya saing, serta menciptakan lingkungan yang mendukung pengembangan dosen dan mahasiswa. Indikator Kinerja Utama Perguruan Tinggi Negeri yang baru ini didasari oleh berbagai landasan hukum, dikeluarkanlah oleh Menteri Pendidikan dan Kebudayaan melalui Keputusan Menteri Pendidikan dan Kebudayaan Nomor 754/P/2020 merupakan ukuran kinerja baru bagi perguruan tinggi untuk mewujudkan perguruan tinggi yang adaptif dengan berbasis luaran lebih konkret. Kebijakan tersebut juga menjadi alat ukur untuk mengakselerasi implementasi Merdeka Belajar-Kampus Merdeka [3].

IKU menetapkan 8 indikator yang menjadi fokus utama transformasi pendidikan tinggi, yang antara lain [3]:

Tabel 2.1 Indikator Kinerja Utama

IKU	Deskripsi	Pemenuhan oleh PTN
IKU 1	Lulusan Mendapatkan Pekerjaan yang Layak	Memastikan lulusan mendapatkan pekerjaan dengan gaji layak, melanjutkan pendidikan, atau menjadi wirausahawan.
IKU 2	Mahasiswa Mendapat Pengalaman di Luar Kampus	Melibatkan mahasiswa dalam magang, penelitian, proyek sosial, atau pertukaran pelajar.
IKU 3	Dosen Berkegiatan di Luar Kampus	Dosen melaksanakan tridarma di industri, universitas lain, atau lembaga internasional.
IKU 4	Praktisi Mengajar di Kampus	Mengundang praktisi dari industri untuk mengajar atau menjadi pembimbing.
IKU 5	Hasil Kerja Dosen Digunakan oleh Masyarakat	Karya dosen diterapkan dalam masyarakat atau diakui secara internasional.

IKU	Deskripsi	Pemenuhan oleh PTN
IKU 6	Program Studi Bermitra dengan Mitra Kelas Dunia	Membentuk kerja sama internasional dalam kurikulum atau pengajaran.
IKU 7	Kelas yang Kolaboratif dan Partisipatif	Mengembangkan pembelajaran berbasis proyek atau pemecahan masalah.
IKU 8	Program Studi Berstandar Internasional	Mendapat akreditasi internasional atau pengakuan di tingkat global.

2.2. Perguruan Tinggi

Pendidikan adalah usaha terencana dalam proses belajar agar peserta didik dapat mengembangkan potensi dalam berbagai aspek untuk keterampilan yang diperlukan dikedepannya [4]. Menurut *United Nations Educational, Scientific and Cultural Organization* (UNESCO), perguruan tinggi adalah insitusi pendidikan yang berfokus pada pengembangan pengetahuan, keterampilan, dan kemampuan melalui pengajaran, penelitian, serta pengabdian masyarakat, dengan begitu perguruan tinggi memiliki peran penting dalam menudukung pencapaian Tujuan Pembangunan Berkelanjutan (*Sustainable Development Goals/*SDGs) [5].

Proses dari pendidikan di Indonesia terbagi menjadi berbagai macam, salah satunya adalah perguruan tinggi. Perguruan tinggi adalah bagian ketiga dari jenjang pendidikan formal yang ada setelah pendidikan mengengah yang mencakup program – program pendidikan seperti diploma, sarjana, magister, dan masih banyak lagi. Pihak pengampu atau pengelola dalam pendidikan tinggi di Indonesia adalah perguruan tinggi, atau bisa diartikan perguruan tinggi adalah lembaga pendidikan formal yang bertujuan menghasilkan sumber daya manusia yang berkualitas melalui pendidikan, penelitian, dan pengabdian kepada masyarakat [4].

Sistem pendidikan tinggi di Indonesia diatur dalam Undang-Undang No. 20 Tahun 2003 tentang Sistem Pendidikan Nasional, yang menetapkan perguruan tinggi sebagai penyelenggara pendidikan formal di tingkat tertinggi. Perguruan tinggi di Indonesia dikelompokkan menjadi Perguruan Tinggi Negeri (PTN), Perguruan Tinggi Swasta (PTS), dan Perguruan Tinggi Keagamaan [4]. Pemerintah melalui Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi (Kemendikbudristek)

terus mendorong peningkatan mutu pendidikan tinggi dengan program seperti Kampus Merdeka.

PTN sendiri dibagi menjadi 3 macam dalam kategori pengelolaannya, yaitu Perguruan Tinggi Negeri Badan Hukum (PTN-BH), Perguruan Tinggi Negeri Badan Layanan Umum (PTN-BLU), dan Perguruan Tinggi Negeri Satuan Kerja (PTN-Satker). Pembagian ini mengacu pada kerangka regulasi yang mengatur otonomi, pengelolaan keuangan, serta tanggung jawab administrasi dari masing – masing kategori perguruan tinggi. Melalui peraturan menteri pendidikan dan kebudayaan no. 88 tahun 2014 dan sudah dievaluasi di peraturan menteri pendidikan dan kebudayaan no. 4 tahun 2020, terdapat beberapa syarat dan prasyarat untuk membuat sebuah universitas dan dibagi menjadi beberapa tahapan dalam universitas. Sehingga perguruan – perguruan tinggi yang ingin memaksimalkan pengelolaan instansinya berjuang memaksimalkan kualitas dan membuat seluruh perguruan tinggi di Indonesia berlomba untuk menduduki sebagai posisi terbaik dalam peringkat nasional maupun internasional [6][7].

Perguruan tinggi memainkan peran strategis dalam membangun generasi terdidik yang mampu menjawab tantangan global dan mendukung pembangunan nasional. Secara global, perguruan tinggi menjadi pusat inovasi, pengembangan teknologi, dan pemberdayaan masyarakat. Laporan *World Bank* menyebutkan bahwa investasi dalam pendidikan tinggi memberikan kontribusi signifikan terhadap pertumbuhan ekonomi melalui peningkatan kualitas Sumber Data Manusia (SDM) dan pengembangan pengetahuan [8].

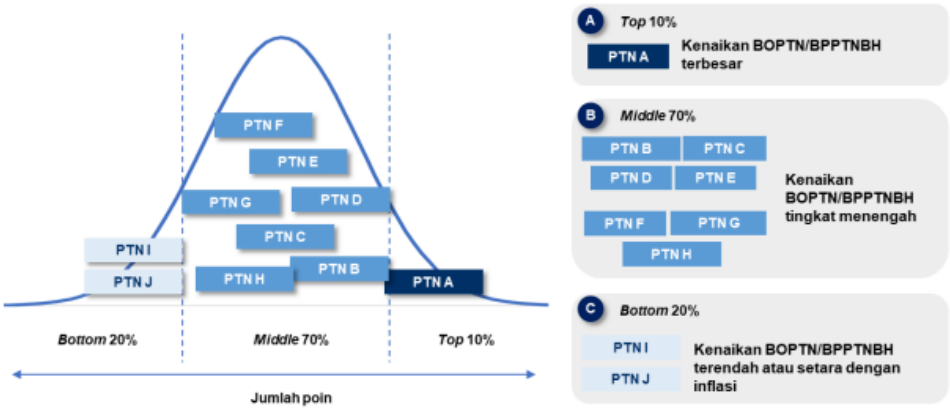
Indikator Kinerja Utama (IKU) memiliki peran yang sangat penting bagi perguruan tinggi, karena menjadi alat evaluasi utama untuk menilai keberhasilan institusi dalam menjalankan tugas dan tanggung jawabnya. IKU mendorong perguruan tinggi untuk mencapai target – target strategis, seperti peningkatan kualitas lulusan, efisiensi kelulusan tepat waktu, relevansi pendidikan dengan kebutuhan industri, dan kontribusi penelitian. Bagi perguruan tinggi, pencapaian IKU tidak hanya mencerminkan reputasi institusi, tetapi juga berpengaruh langsung terhadap alokasi dana dan insentif dari pemerintah.

Penilaian IKU pada perguruan tinggi dilakukan berdasarkan indikator – indikator spesifik yang telah ditetapkan oleh Kementerian Pendidikan, Kebudayaan, Riset, dan Teknologi. Data kuantitatif seperti jumlah lulusan tepat waktu, daya serap lulusan di dunia kerja, jumlah kerja sama dengan mitra industri, serta jumlah publikasi ilmiah menjadi komponen utama dalam penghitungan nilai IKU. Hasil penilaian ini menjadi dasar untuk mengelompokkan perguruan tinggi ke dalam sistem liga.

Target "Standar Emas" adalah target untuk setiap IKU yang ditetapkan sebagai tolak ukur keunggulan, setiap jenis PTN memiliki standar emas yang berbeda, ini diatur oleh peraturan, keputusan, surat edaran, atau pedoman terpisah. Perguruan tinggi yang mampu memenuhi atau melampaui standar ini dianggap telah mencapai tingkat kinerja yang optimal. Target emas ini berfungsi sebagai pedoman bagi perguruan tinggi untuk terus meningkatkan kualitas layanan dan kontribusi akademik mereka, baik di tingkat nasional maupun internasional.

Poin dari IKU akan diukut menggunakan kurva yang dibagi menggunakan sistem liga. PTN akan dibagi menjadi 3 liga, yakni liga PTN Badan Hukum, liga PTN BLU, dan liga PTN Satker.

Setelah penghitungan poin dan skor dilakukan, semua pencapaian kinerja PTN akan diukur menggunakan kurva yang dibagi menggunakan system liga. PTN agan dibagi menjadi 3 liga yakni Liga PTN Badan hukum, Liga PTN BLU, dan Liga PTN Satker. Masing-masing liga akan dinilai menggunakan kurva kompetitif.



Gambar 2.1 Implikasi kurva terhadap penetapan kenaikan BOPTN

Liga ini terdiri dari tiga kategori utama: Top 10%, Middle 70%, dan Bottom 20%. Perguruan tinggi di kategori Top 10% adalah yang memiliki kinerja terbaik dan menjadi prioritas untuk mendapatkan insentif tambahan atau Bantuan Operasional Perguruan Tinggi Negeri (BOPTN). Sistem ini dirancang untuk mendorong persaingan sehat antar perguruan tinggi sekaligus memberikan pembinaan bagi institusi yang berada di kategori bawah. Selain kenaikan BOPTN/BPPTNBH berdasarkan klasifikasi liga, PTN juga dapat menerima bonus berdasarkan keunggulan dalam 8 kategori IKU seperti skor nominal terbaik, pertumbuhan terbaik, nominal kualitas lulusan terbaik dan lainnya[3].

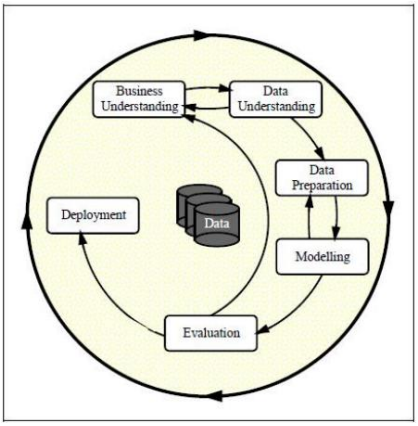
Kelulusan Tepat Waktu (KTW) adalah keberhasilan mahasiswa menyelesaikan studi sesuai dengan durasi yang telah ditentukan oleh kurikulum. KTW mencerminkan kemampuan mahasiswa untuk memenuhi semua persyaratan akademik tanpa mengalami penundaan. Mahasiswa menyelesaikan studinya sesuai rencana akademik tanpa mengorbankan kualitas pembelajaran. Konsep ini relevan terutama untuk mahasiswa yang sepenuhnya berfokus pada perkuliahan, sehingga faktor eksternal seperti magang atau kegiatan non-akademik lainnya dapat dikelola secara fleksibel tanpa menghambat KTW.

Dalam poin – poin IKU, tidak termuat secara jelas tentang mahasiswa yang harus lulus tepat waktu. Meskipun demikian, KTW mendukung pencapaian beberapa poin IKU, antara lain :

1. IKU 1 : Lulusan yang menyelesaikan studi tepat waktu dianggap memiliki daya saing lebih tinggi karena menunjukkan kemampuan manajemen waktu, disiplin, dan komitmen terhadap target.
2. IKU 2 : Pengalaman belajar di luar kampus memiliki potensi untuk memperpanjang waktu studi. Hal ini disebabkan oleh ketidaksesuaian antara jadwal akademik dan program luar kampus atau beban tambahan yang perlu diselesaikan setelah mengikuti program tersebut. Kelulusan tepat waktu mendukung poin ini karena perguruan tinggi yang baik memiliki integrasi yang baik seperti pengakuat kredit untuk program luar kampus.

2.3. *CRISP-DM*

Cross Industry Standard Process for Data Mining (CRISP-DM) merupakan salah satu standar yang lumayan umum digunakan dalam melakukan suatu pemrosesan data mining, versi pertama dipublikasikan oleh Pete Chapman, dkk pada tahun 1999. Metodologi ini ditujukan untuk membuat proyek data mining besar dapat diandalkan, diulang, mudah dikelola dan lebih cepat [9].



Gambar 2.2 Fase Model Referensi CRISP-DM [10]

CRISP-DM dari versi 1.0 hingga 2.0 menjabarkan bahwa didalamnya terdapat 6 tahapan, pada Gambar 2.2 dijelaskan tahapan – tahapan dalam menggunakan CRISP-DM.

1. Tahapan *business understanding,* tahapan untuk memahami objektivitas dan kebutuhan apa saja dari kacamata bisnis, lalu merangkum datanya menjadi data yang siap diubah menjadi pengkategorian *data mining*, sehingga dapat melancarkan dalam pembuatan rencana objektivitas yang ingin diraih.
2. Tahapan *data understanding,* data yang sudah didapat, dicerna kembali, membuat subset data, sehingga data diidentifikasi keseluruhan informasi yang bisa didpat. Hasil dari tahapan ini bisa dikembalikan kembali ke dalam tahap *business understanding* untuk mencari perspektif bisnis dari hasil indentifikasi data yang diperoleh.
3. Tahap *data preparation,* adalah tahap untuk mempersiapkan data mentah yang telah dikumpulkan agar siap digunakan untuk dimasukan ke dalam sistem pengolahan data, tahap ini akan mengolah data seperti membersihkan, mengatur

format, pembuatan fitur baru dan sebagainya. Tahap ini mempunyai ikatan dengan *requirement input system* sehingga data bisa terbaca dengan jelas.

4. Tahapan *modelling.* Di tahapan terjadi kegiatan pemilihan dan penerapan model – model yang sesuai untuk di gunakan dalam proyek. Biasanya dalam menggunakan suatu teknik diperlukan parameter yang sudah sesuai, ini membuat dataset harus dikembangkan dahulu, atau kembali ke tahap sebelumnya yaitu *data understanding* untuk membuat parameter yang diperlukan menjadi sesuai.
5. Tahap *evaluation*. Tahapan ini akan memaksimalkan kualitas yang tinggi dalam perspektif bisnis. Sebelum masuk ke tahapan akihr, sangatlah penting untuk menyimak dan memahami model yang sudah dibuat dan meninjau kembali agar proyeknya dapat mencapai titik yang diinginkan dalam perspektif bisnis. Kuncinya adalah menemukan masalah dalam bisnis yang tidak bisa teratasi secara memadai. Jika teridentifikasi masalah bisnis yang penting dan tidak bisa teratasi maka kembali ketahapan awal.
6. Tahapan *deployment*. Pencpitaan sebuah model secara umum bukanlah hasil dari suatu proyek. Walaupun tujuan dari sebuah model adalah untuk meningkatkan kualitas informasi dari datanya, namun informasi yang diperoleh harus diatur dan dipresentasikan dengan cara yang customer dapat pahami. Sangatlah mungkin jika model yang akan digunakan harus melibatkan *"live" model*. Sebagai contoh sebuah *web*site yang memperlihatkan skor dari *marketing databases*, jika diberikan informasi data yang secara *real time* untuk memberi informasi angka saat ini, terkadang lebih baik untuk memberikan *report* secara per periode untuk memberikan hasilnya, karena *customer* tidak semuanya adalah data analisis yang dapat mengurai dan menganalisis informasi dari sebuah data sederhana [10].

2.4. *Javascript*

Javascript adalah bahasa pemrograman yang dikembangkan pertama kali oleh Brendan Eich pada 1995. *Javascipt* memiliki karakterisistik sebagai bahasa yang berbasis prototipe dinamis dan mendukung gaya pemrograman berorientasi objek yang imperatif dan fungsional.

Javascript adalah bahasa pemrograman yang berjalan di sisi klien (*client-side*) di dalam browser, sehingga memungkinkan pengembang untuk membuat aplikasi *web* yang *responsive* dan interaktif. Selain itu, dengan adanya Node.js, Javascript dapat digunakan untuk di sisi server juga (*server-side*), ini memperluas kemampuannya dalam pengembangan aplikasi *web* secara penuh (*full-stack*). Keunggulan *Javascript* terletak pada kemampuannya untuk dapat berinteraksi dengan DOM (*Document Object Model*) yang memungkinkan pengubahan konten, struktur, dan gaya halaman *web* secara langsung tanpa perlu memuat ulang halaman *web*.

Seiring dengan perkembangannya, *Javascript* kini memiliki ekosistem yang luas, dengan banyaknya pustaka dan kerangka kerja (*framework*) seperti React, Angular, Vue.js dan masih banyak lagi memudahkan pengembangan aplikasi *web* menjadi lebih mudah. Popularitas *Javascript* terus meningkat karena sifatnya yang mudah dipelajari. Kompabilitas dengan berbagai platform, dan dukungannya terhadap berbagai standar *web* [11][12].

2.5. Tensorflow

TensorFlow adalah kerangka kerja yang bersifat terbuka (*open-source*) yang dipublikasi oleh Google di November 2015. TensorFlow memudahkan pengembang dalam membuat model pembelajaran mesin untuk berbagai perangkat seperti *desktop*, seluler, *web*, maupun *cloud.* TensorFlow menawarkan fleksibilitas yang sangat baik, membuat pengguna dapat membuat model dari skala kecil hingga besar dengan mudah. Salah satu keunggulan dari TensorFlow yaitu penggunaan bahasa dalam membangunnya. Tensorflow sendiri terbagi menjadi 2 bahasa yaitu Python dan Javascript.

Dengan penggunaan Python dan Javascript, TensorFlow mendapatkan bantuan dari pustaka – pustaka lain yang dapat mendukung dalam pembuatan model ML (*Machine Learning*) yang baik. Bahkan TensorFlow memberikan perspektif lain dari memproses ML, seperti yang ditawarkan dari TensorFlow.js yang menggunakan bahasa *javascript* dalam pembuatan model ML. seperti yang kita ketahui bahwa *javascript* bekerja di sisi klien (HTML, CSS, Javascript) sehingga membuat penggunaan memori dalam fungsi kerja ML bekerja di sisi klien [13][14].

2.6. *Machine Learning* (ML)

Machine *Learning* atau pembelajaran mesin adalah teknik yang memungkinkan mesin untuk belajar dari data dan pengalaman, tanpa harus diprogram secara eksplisit untuk setiap tugas. Tujuan utamanya adalah membuat model yang dapat menggeneralisasi pola-pola dalam data, sehingga mesin dapat melakukan prediksi atau mengambil keputusan terhadap data baru yang belum pernah dilihat sebelumnya [15]. Secara definisi, *machine learning* adalah ilmu studi yang membahas tentang algoritma dan model statistik yang digunakan komputer untuk dapat menjalankan tugasnya secara otomatis tanpa perintah secara eksplisit [16].

Cara kerja dari ML dengan melatih sebuah algoritma pada suatu set data untuk mencapai hasil yang diharapkan seperti menganalisa apakah ada pola yang mempengaruhi hasil. ML adalah proses pengoptimalan agar model yang dihasilkan adalah model yang paling baik sehingga dapat memprediksi respon dengan berdasarkan sampel data pelatihan.

Dalam memprediksi sebuah hasil, data pelatihan haruslah data yang berkualitas tinggi, dan semakin banyak data sampel pelatihan yang ada, maka semakin bagus juga model untuk memprediksi hasil yang di hasilkan, algoritma penyesuaian model selama latihan ini disebut juga dengan "proses penyesuaian". Namun jika hasilnya tidak sesuai dengan hasil yang diharapkan, model dapat dilatih berulang kali sehingga menghasilkan respon yang akurat [17].

Deep Learning adalah bidang dalam pembelajaran mesin yang menggunakan pendekatan yang lebih kompleks, data yang diterima akan diekstrak fiturnya dengan beberapa lapisan untuk dapat mengidentifikasi aspek yang relevan dengan data *input*. *Deep learning* dinilai lebih baik dalam prediksi pada data yang tak terstruktur [18].

Deep *Learning* vs Machine *Learning*: Deep *learning* adalah sub-bidang dari machine *learning*, tetapi dengan fokus yang lebih pada model yang kompleks, terutama yang memiliki banyak lapisan (*hidden layers*). Sementara *machine learning* lebih tradisional dan lebih sering menggunakan teknik seperti decision trees atau SVM dengan fitur yang dipilih secara manual, deep *learning* lebih fokus

pada arsitektur jaringan saraf tiruan (ANN) yang dapat "belajar" fitur dari data secara otomatis melalui proses *backpropagation*.

Salah satu perbedaan mendasar adalah penggunaan banyak lapisan (*deep*) pada jaringan saraf. Ini memberikan kemampuan bagi deep *learning* untuk menangkap pola yang lebih kompleks, yang sulit dilakukan oleh model machine *learning* dengan satu hidden *layer*. Deep *learning* mempelajari representasi hierarkis dari data. Ini berarti lapisan pertama mungkin menangkap fitur dasar, seperti tepi pada gambar, sementara lapisan yang lebih dalam dapat menangkap fitur yang lebih abstrak dan kompleks [19].

Supervised Learning (pembelajaran yang diawasi), teknik ini adalah pendekatan yang paling umum digunakan,ini adalah teknik dalam membuat model *machine learning* yang menggunakan data masukan yang berlabel untuk menentukan *output* yang sudah diketahui seperti mengenali gambar buah, dan model dilatih menggunakan data dari *output* yang diketahui. Jadi model dilatih dari pasangan masukan dan keluaran, sehingga untuk menentukan keluaran dari masukan yang belum mempunyai pasangan keluaran dapat diprediksi dengan akurat [17][20].

Unsupervised learning atau model yang tidak terawasi adalah model yang selama pembelajaran tidak menggunakan data berlabel (data tidak berstruktur) untuk mencari pola yang ada. Tidak seperti *supervised learning* yang masukannya diberikan label dan memiliki nilai keluaran yang sudah ditentukan, disini, model harus mencari tahu sendiri dari apa saja masukan yang diberikan. misalnya diberikan gambar apel dan pisan, model harus mengengstrak nilai – nilai yang ada didalamnya sehingga dapat membedakan gambar pisang dan gambar apel. Jadi mesin tidak tahu keluaran sama sekali [17][20].

2.7. *Data Preprocessing*

Pemrosesan awal yang biasa dilakukan untuk ML melibatkan rekayasa data dan rekayasa fitur. Rekayasa data adalah proses mengubah data mentah menjadi data yang siap pakai. Rekayasa data digunakan untuk menyiapkan fitur yang dibutuhkan oleh model ML. Data – data ini dibagi menjadi 3, yaitu

1. Data mentah, data yang diambil dari sumbernya, dalam konteks ML, data mentah berarti data tersebut belum disiapkan secara khusus untuk model ML.
2. Data yang sudah disiapkan, kumpulan data yang dari sumbernya sudah disiapkan, yang berarti sudah memilih apa saja yang akan dipakai dalam model ML, sudah diurai, digabungkan, dan dimasukkan ke dalam satu kesatuan kumpulan data. Kumpulan data ini sudah disiapkan perbarisnya, artinya data sudah disaring sehingga tidak ada data yang tidak relevan, data yang cacat, dan sudah tidak ada lagi data yang tidak valid.
3. Fitur yang direkayasa, kumpulan data dengan fitur yang akan dimasukkan kedalam model ML sudah diubah, sehingga model bisa memahami isi dari setiap data. Kebanyakan model ML lebih memahami angka dibandingkan kata – kata, sehingga data sudah dikategorikan dan diubah menjadi angka. Seperti *true false* dalam suatu data diubah menjadi 1 dan 0 [21].

2.7.1. Integrasi Data

Tujuan utama dari pengintegrasian data ini adalah menciptakan data yang terpadu dan holistik dari data – data yang sebelumnya kurang lengkap, sehingga memungkinkan analisa data yang baik sehingga pemahaman tentang data akan lebih baik dari data mentah yang didapatkan dari basis data sebelumnya. Integrasi yang dilakukan dengan baik memberikan manfaat yang besar, seperti mengurangi redudansi dan inkonsistensi dalam kumpulan data yang didapat. Hal ini dapat meningkatkan akurasi dan kecepatan proses dalam pemrosesan data. Namun *heterogenis semantic* dan struktur data menimbulkan tantangan besar dalam integrasi data, seperti mengidentifikasi esensi dari masalah yang teridentifikasi, atribut yang berkolerasi mapun data yang terduplikasi [22].

2.7.2. *Cleaning Data*

Pembersihan data melibatkan perbaikan masalah sistematis atau memperbaiki kesalahan data yang berantakan. Pembersihan data yang paling baik adalah yang dapat mengidentifikasi masalah yang mendalam, melibatkan identifikasi dan penanganan data yang munkin salah. Ada banyak asalan mengapa data bisa

dikatakan salah, seperti kesalahan ketik, rusak, duplikasi, dan sebagainya. Dengan begitu, data yang dinyatakan rusak dapat dilakukan tindakan lanjut seperti penghapusan baris atau kolom, atau mungkin penggantian dengan nilai baru.

Dalam perbaikan data ada beberapa operasi standar yang dilakukan agar data yang didapat dapat digunakan dalam pengolahan data dengan baik, khususnya *machine learning*. Berikut adalah beberapa tahapan dalam mempersiapkan data dalam pengolahan data.

1. **Identifikasi kolom untuk mencari nilai duplikasi atau data yang tidak memiliki varians.**

Dilangkah ini memiliki tujuan untuk menemukan kolom – kolom dalam dataset yang tidak memberikan informasi yang berguna dalam pembangunan model. Seperti kolom yang hanya memiliki nilai yang sama untuk semua baris atau biasa disebut *zero-variance predictors* atau *predictor* dengan varians nol sehingga kolom ini tidak berguna dalam kemampuan prediktif suatu model. Kolom yang hanya memiliki satu nilai untuk semua baris disebut sebagai "*zero-variance predictor*s" atau prediktor dengan varians nol. Karena tidak ada variasi dalam nilai-nilai tersebut, kolom ini tidak akan berkontribusi pada kemampuan prediktif model.

Kolom yang hanya memiliki nilai yang sedikit, misalnya dalam suatu kolom hanya terdapat 3 nilai yang berbeda yaitu 2,4,9. Meskipun kolom ini tidak memiliki varians nol, tapi dalam membangun model yang prediktif terlebih untuk model yang lebih akurat, lebih baik untuk tidak memasukan nilai – nilai ini kedalam fitur dalam pembuatan model.

2. **Identifikasi baris data duplikat.**

Langkah ini ditujukan untuk menemukan dan menghapus baris – baris dalam dataset yang memiliki data identik. Baris yang memiliki nilai duplikat dapat menyebabkan bias dalam evaluasi model, terutama jika baris yang sama muncul pada set pelatihan dan set pengujian, sehingga dapat memberikan hasil yang terlalu optimis dalam memberikan hasil prediksi mengingat data yang sama. Langkah ini berfokus pada menemukan dan menghapus baris-baris dalam dataset yang memiliki data identik. Baris duplikat dapat menyebabkan bias dalam evaluasi model, terutama jika baris yang sama muncul di set pelatihan dan set pengujian.

Ini dapat memberikan hasil yang terlalu optimis karena model mungkin "mengingat" data yang sama.

3. Penggunaan statistik untuk menentukan data normal dan identifikasi *outlier.*

Dalam proses pembersihan data, identifikasi nilai *outlier* sangat penting karena dapat mendistorsi hasil analisis data. Metode statistik yang dilakukan dalam tahapan ini adalah menghitung nilai rata – rata dan simpangan baku untuk mendeteksi nilai *outlier*.

Outlier mengacu pada data yang jauh berbeda atau menyimpang dari mayoritas nilai dalam sebuah dataset. *Outlier* bisa terjadi secara alami, bisa berdasarkan karena kesalahan pengukuran atau karena faktor lain yang tidak terduga. *Outlier* penting untuk diidentifikasi karena dapat mempengaruhi data, seperti mengubah nilai rata – rata sehingga memberikan representasi yang tidak akurat dari data, atau juga dapat mempengaruhi dalam pengumpulan data atau sesuatu yang berbeda dalam fenomena yang dipelajari.

Nilai *outlier* sering didefinisikan sebagai data yang berada diluar 3 standar deviasi dari nilai rata – rata dataset. Sehingga proses ini melibatkan beberapa langkah yaitu:

a. Menghitung rata – rata dan simpangan baku. Rata – rata didapat dari semua data dalam dataset. Rumusnya adalah menjumlahkan semua nilai dan membaginya dengan jumlah data. Lalu untuk simpangan baku adalah pengukuran seberapa jauh setiap data menyimpang dari nilai rata – rata yang didapat. Data yang lebih dekat ke rata – rata memiliki simpangan baku yang lebih rendah. Untuk rumusnya adalah akar dari variansi, yaitu rata – rata dari kuadrat selisih setiap niali dari rata – rata.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \qquad (2.1)$$

Keterangan variabel:
σ = simpangan baku
N = jumlah data
x_i = nilai individu dalam dataset

μ = rata-rata dari dataset

b. Menentukan batas *cut-off* , atau batas ambang. Dengan metode ini batas yang dihitung dengan rumus :

$$Batas\ Bawah = \mu - (3 \times \sigma) \quad (2.2)$$

$$Batas\ Atas = \mu + (3 \times \sigma) \quad (2.3)$$

c. Melakukan tindakan lanjut yaitu menghapusnya untuk memperbaiki aurasi model dalam prediksi.

Identifikasi dan penanganan outlier sering dilakukan dalam proses data *preprocessing* untuk memastikan model tidak terpengaruh oleh data yang tidak representatif atau ekstrem. Biasanya dalam pengidentifikasi *outlier* dengan metode IQR (*Interquartile Range*) atau Z-skor. Metode yang sebelumnya adalah metode Z-skor. IQR menggunakan pendekatan mengabaikan data yang ekstrem, sehingga menghitung data yang berada di ambang tengah saja.

4. Identifikasi nilai yang kosong.

Mengidentifikasi nilai yang hilang adalah langkah yang mendasar dalam pembersihan data, karena kesenjangan data yang hilang dapat menyebabkan analisis menjadi bias. Dalam mengangani hal ini terdapat beberapa tahapan yaitu:

a. Inspeksi data untuk menemukan nilai yang *NaN* (*not a number*) atau bukan angka sehingga ini bisa dikatakan menjadi data yang hilang.
b. Menandai nilai yang hilang.
c. Menghitung nilai yang hilang, termasuk persentase nilai yang hilang dalam keseluruhan dataset yang ada, sehingga dapat menetahui seberapa sempurnakah model dalam prediksi dengan data yang tidak lengkap.
d. Representasi visual, bisa menggunakan alat visual seperti perta panas untuk menggambarkan distribusi nilai yang hilang di seluruh data set yang ada, sehingga dapat memahami pola data hilang yang terjadi.

5. Penghitungan nilai yang hilang menggunakan statistik atau model yang dipelajari.

Imputasi dalam penggantian nilai yang hilang dengan nilai yang diestimasukan adalah salah satu tahapan yang penting dalam *preprocessing* data untuk *data*

mining. Dalam pemrosesan data khususnya sebagai fitur untuk model, data sangatlah penting, sehingga data yang hilang tidak boleh dihapus secara cuma – cuma, terdapat beberapa alasan mengapa demikian, seperti dapat mengurangi informasi dan dapat menimbulkan bias. Jika proporsi data yang hilang cukup besar, menghapus baris ini bisa mengurangi kualitas dataset secara signifikan dan menyebabkan model kurang akurat. Data yang hilang tidak selalu terjadi secara acak. Jika data yang hilang cenderung memiliki pola tertentu, menghapusnya dapat menyebabkan bias dalam analisis, yang berarti model bisa menghasilkan kesimpulan yang tidak akurat. Imputasi data, terdapat beberapa metode yaitu:

a. Sederhana. Imputasi dilakukan dengan cara yang sederhana, menggantikan nilai yang hilang dengan ukuran statistik seperti *mean, median* atau modus kolom.
b. Iteratif, yaitu memanfaatkan model yang dapat memprediksi nilai yang hilang berdasarkan data lain yang tersedia dalam kumpulan data. Dalam beberapa kasus, kebanyakan imputasi menggunakan model KNN (*K-Nearest Neighbors*).
c. Multivariat, teknik lanjutan seperti MICE (*Multiple Imputation by Chained Equations*). Dalam hal pendekatan, ini sama seperti imputasi secara iteratif, yang beda adalah imputasi iteratif dilakukan secara berulang (biasanya dilakukan 5 sampai 10×).

Setelah imputasi, dokumentasi dari nilai yang hilang sebelum dan sesudah imputasi sangat penting dalam menjaga transparansi dalam pemrosesan data [23].

2.7.3. *Feature Engineering*

Feature engineering atau rekayasa fitur adalah tahapan penting dalam mempersiapkan data agar data lebih imformatif dan sesuai untuk model *machine learning*. Pada dasarnya rekayasa fitur mencakup banyak tahapan. Yaitu pemilihan fitur (*feature selection*), pembuatan fitur baru (*feature creation*), transformasi data, *handling missing values, scalling* dan normalisasi.

1. *feature selection* atau pemilihan fitur adalah proses mengidentifikasi dan memilih fitur terpenting dari subset data yang ada. Proses ini penting untuk

menghindari model mengalami *overfitting*. Dalam pembelajaran mesin, jika terlalu banyak fitur yang tidak relevan atau berlebihan dapat membingungkan model sehingga model memiliki kinerja yang buruk.

2. Dalam *O'Reilly feature engineering* oleh Alice Zheng menjelaskan bahwa dalam proses ini menyarankan menggunakan teknik seperti uji statistik, keahlian domain, atau metode berbasis model (seperti skor kepentingan fitur dari pohon keputusan) untuk pemilihan fitur *deep learning*. Algoritma seperti Lasso (mengecilkan koefisien fitur yang kurang penting menjadi nol) juga disarankan agar dapat melakukan pemilihan fitur secara otomatis.
3. *feature creation* atau pembuatan fitur baru. Setelah memilih fitur yang relevan, terkadang data yang didapat tidak bisa diterima oleh model, sehingga perlu adanya pengubahan data menjadi data yang bisa diterima oleh mesin. Proses ini melibatkan modifikasi atau pembuatan fitur baru yang lebih mewakili pola dasar dalam data. Pembuatan fitur baru penting karena dapat meningkatkan kinerja model secara signifikan [24].

2.7.4. *Data Transformation*

Data transformasi adalah proses penting dalam penyiapan data khususnya dalam *machine learning*, proses yang terjadi disini adalah menkonversi atau memanipulasi data dari format aslinya menjadi bentuk yang lebih cocok digunakan dalam model pembelajaran mesin. Ada beberapa metode untuk mengubah data mentah agar sesuai dengan *input* mesin.

Encoding categorial: menkonversi data kategorial menjadi data numerik. Jadi data yang sifatnya *NaN* akan diubah menjadi numerik menggunakan kategori. Kategori dibuat atau dibedakan berdasarkan keinginan oleh pembuat. Contoh : Terdapat data nama – nama sekolah, agar mesin dapat menghitung dan menganalisa nama – nama sekolah ini, data bisa diubah berdasarkan daerahnya, atau jenis sekolahnya, atau biayanya. Data – data sekolah tersebut diubah menjadi data angka sehingga mesin dapat mengerti bahwa data – data tersebut berbeda. Pada dasarnya, metode ini memiliki 2 gaya dalam pengubahan data, yaitu:

1. *One hot-encoding* : Teknik transformasi data yang digunakan yang mengubah variabel kategorikal menjadi bentuk bit. Setiap bit mewakili kategori yang mungkin. Jika variabel tidak dapat menjadi bagian dari beberapa kategori sekaligus, maka hanya satu bit dalam kelompok tersebut yang menjadi aktif.

Tabel 2.2 Contoh pengaplikasian *one-hot encoding* pada 3 variabel

Nilai	i_1	i_2	i_3	**Hasil**
Merah	0	0	1	001
Hijau	0	1	0	010
Biru	1	0	0	001

Contoh pengaplikasian *one-hot encoding* pada 3 variabel yaitu merah, hijau, dan biru. Dari *one-hot encoding* didapatkan nilai merah diubah menjadi [0,0,1], nilai hijau menjadi [0,1,0], dan nilai biru menjadi [0,0,1]. Tipe gaya ini biasa diaplikasikan pada data yang *NaN* tapi tidak memiliki urutan nilainya, seperti subjek, lokasi dan sebagainya [24].

2. *Label encoding :* Berbeda dengan *one-hot encoding*, *label encoding* memasukkan informasi besar dan kecilnya suatu nilai, jadi transformasi data ini biasanya dilakukan pada data yang memiliki urutan nilai [25], seperti ukuran baju : S, M, L, XL, dan XXL yang berarti S paling kecil dan XXL menjadi besar. Jika menggunakan format *label encoding* adalah sebagai berikut:

Tabel 2.3 Contoh pengaplikasian *label encoding* pada 5 ukuran baju

Nilai	**Nilai setelah dikonversi**
S	1
M	2
L	3
XL	4
XXL	5

Normalisasi data merupakan suatu teknik transformasi data yang mengubah skala data atribut sehingga data tidak memiliki rentang nilai yang besar, seperti -1 hingga 1 atau 0 hingga 1. Untuk menghindari mesin salah mengambil perbandingan terhadap suatu fitur, maka data harus dinormalisasikan. Menormalkan data berarti

memberikan bobot yang sama pada semua atribut. Dengan normalisasi data masukan pada data pelatihan dapat membantu mempercepat fase pembelajaran model. Salah satu metode pendekatan yang umum adalah *min-max normalization*. Metode ini melakukan transformasi linear pada data asli sehingga data memiliki nilai rentang minimum dan maksimum yang kecil [22][26]. Normalisasi dengan metode ini digambarkan pada rumus berikut :

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{2.4}$$

Atau jika nilai ingin digeneralisasi menjadi:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} (new\_max_x - new\_min_x) + new\_min_x \tag{2.5}$$

Keterangan variabel:

x'	: nilai setelah di normalisasi
x	: nilai asli
$\max(x)$	: nilai maksimum
$\min(x)$	: nilai nimimum
$new\_max_x - new\_min_x$	: nilai maksimum dan minimum yang diinginkan (misalnya 0 dan 1 atau -1 dan 0)

2.7.5. *Feature Selection*

Dalam proses pembangunan model *machine learning*, pemilihan fitur merupakan langkah penting untuk meningkatkan akurasi serta efisiensi model. Fitur yang tidak relevan atau tidak berpengaruh secara signifikan terhadap target prediksi dapat menyebabkan model menjadi tidak optimal, mengalami *overfitting*, atau bahkan gagal mengenali pola dengan benar (*overfitting*) (dimensi tinggi). Oleh karena itu, perlu dilakukan proses seleksi fitur yang efektif guna memastikan bahwa hanya fitur penting saja yang digunakan sebagai *input* ke dalam model [24].

Terdapat beberapa cara untuk menilai seberapa penting sebuah atribut atau input dalam menentukan target. Salah satu metode yang banyak digunakan adalah *mutual information* (MI). MI berasal dari teori informasi dan digunakan untuk mengukur seberapa besar informasi yang dimiliki satu variabel terhadap variabel lainnya dalam hal ini, antara fitur *input* dan label target [27].

$$MI(X;Y) = \sum_{x \in X} \sum_{y \in Y} P(x,y) \cdot \log_2 \left(\frac{P(x,y)}{P(x) \cdot P(y)} \right) \quad (2.6)$$

Dimana:

X : nilai yang ingin dibandingkan
Y : target pembanding
$x \in X$:semua nilai X
$y \in Y$: semua nilai Y
P : probabilitas

Meskipun MI merupakan ukuran yang kuat untuk menentukan hubungan antar variabel, metode ini memiliki kelemahan ketika digunakan untuk membandingkan fitur – fitur dengan jumlah kategori yang sangat berbeda. Dalam kondisi seperti ini, MI cenderung memberikan nilai yang lebih tinggi pada fitur dengan jumlah kategori lebih banyak, meskipun kontribusi informasinya terhadap target tidak selalu lebih besar.

Adjusted Mutual Information (AMI). AMI merupakan versi penyempurnaan dari MI yang mengoreksi efek bias akibat jumlah kategori melalui pengurangan nilai *Expected Mutual Information* (EMI). EMI adalah nilai MI yang diharapkan muncul secara acak berdasarkan jumlah kategori dari masing – masing variabel [28].

$$AMI(X,Y) = \frac{MI(X,Y) - E[MI(X,Y)]}{\max\big(H(X), H(Y)\big) - E[MI(X,Y)]} \quad (2.7)$$

Dengan:

$MI(X,Y)$:*mutual information* antara X dan Y
H :Entropi, keberagaman nilai
max :membandingkan nilai terbsear antar variabel
$E[MI(X,Y)]$:EMI, ekspetasi MI karena kebetulan, tanpa relasi nyata antar variabel

Dimana:

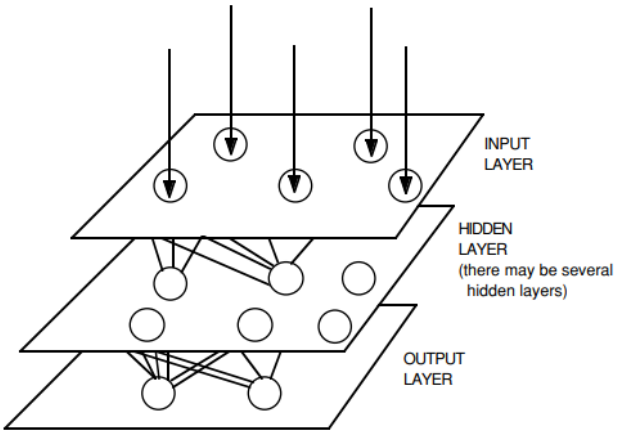
$$EMI(X,Y) = \sum_{x \in X} \sum_{y \in Y} P_{x,y}^{acak} \cdot \log_2 \left(\frac{P_{x,y}^{acak}}{P_x \cdot P_y} \right) \quad (2.8)$$

Dengan adanya koreksi menggunakan EMI, nilai AMI akan lebih adil dan objektif dalam membandingkan seberapa besar informasi yang sebenarnya diberikan oleh masing – masing fitur terhadap target, tanpa dipengaruhi oleh jumlah kategorinya. Oleh karena itu, AMI sangat berguna dalam proses seleksi fitur, terutama saat dataset memiliki variabel dengan skala kategori yang tidak sebanding.

2.8. ANN

Artificial Neural Network (ANN) bisa diartikan menjadi jaringan syaraf tiruan jadi jika dijelaskan secara mudah, ANN adalah arsitektur pemrosesan informasi yang meniru (terinspirasi) jaringan syaraf manusia yaitu otak manusia [29]. Cara kerja otak manusia yang diktahui saat ini memanglah belum sepenuhnya dapat dipahami, namun beberapa aspek dari dari pemrosesan sudah diketahui. Sel paling dasar dari otak manusia adalah jenis sel yang tidak seperti bagian tubuh lainnya, sel ini adalah satu – satunya bagian tubuh yang tidak tergantikan secara perlahan, sehingga sel – sel ini memberi kita kemampuan untuk mengingat, berpikir dan menerapkan pengalaman sebelumnya dalam setiap tindakan kita. Sel – sel ini memiliki jumlah berkisar 100 miliar dengan lebih dari 100 jenis, ini dikenal sebagai *neuron*. Setiap *neuron* dapat terhubung dengan hingga 200 ribu *neuron* lainnya, dan kekuatan otak manusia berasal dari banyaknya hubungan ini, dan hubungan – hubungan yang terjadi berasal dari genetik dan pembelajaran.

Pertama kali dikenalkan oleh Warren McCulloch (nerofisiolog) dan Walter Pitts (matematikawan) pada tahun 1943. Mengembangkan model komputasi dari *neuron* biologis dengan dasar konsep matematika dan logika. Mereka menciptakan model sederhana yang menunjukkan bagaimana *neuron* dapat memproses *input* biner (0 dan 1) melalu fungsi logika. Setiap *neuron* menerima sejumlah *input* dan melakukan operasi logika sederhana seperti AND, OR, dan NOT dan hasilnya adalah nilai *output* biner yang ditentukan oleh jumlah sinyal *input*. Ini adalah fondasi dari *neuron* buatan, yang kemudian menjadi dasar dari ANN [30].



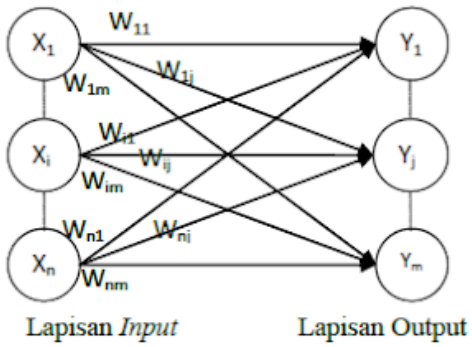
Gambar 2.3 Arsitektur Jaringan Syaraf Tiruan [30]

Berdasarkan Gambar 2.3 menunjukan bagaimana jaringan syaraf tiruan dibangun secara umum, jadi secara umum jaringan syaraf tiruan terdiri menjadi 3 bagian, yaitu:

a. *Input layer* adalah lapisan yang terdiri dari beberapa *neuron* yang akan menerima sinyal dari luar kemudian akan diteruskan ke *neuron* yang berada di *hidden layer*.
b. *Hidden layer* adalah lapisan tersembunyi yang akan meneruskan data berdasarkan bobot dan aktivasi yang ada pada suatu *neuron*. Bobot ditentukan oleh mesin untuk menentukan seberapa besar pengaruh suatu fitur yang diterima. Bobot ditentukan oleh mesin berdasarkan *batchsize* dan model *optimizer* yand dipakai.
c. *Output layer* adalah lapisan yang berfungsi hasil dari proses yang terjadi dari *input* hingga *hidden layer*. *Output layer* juga menggunakan fungsi aktivasi untuk mentransformasi data. Namun dalam beberapa masalah seperti regresi, *output layer* tidak menggunakan fungsi aktivasi karena *output* yang diinginkan merupakan nilai kontinu.

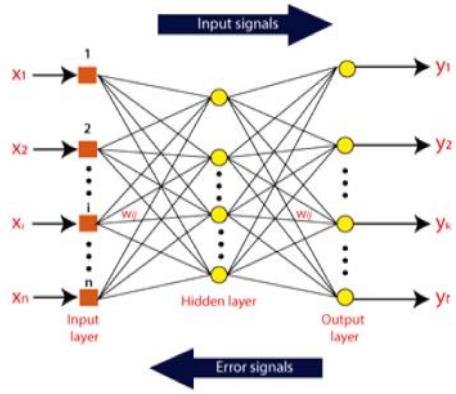
Awalnya arsitektur ANN dibedakan menjadi 2 [31], yaitu:

1. Jaringan lapisan Tunggal (*single layer network*), jaringan ini terdiri dari 1 *layer input* dan 1 *layer output*. Setiap *neuron* yang ada dilapisan *input* selalu terhubung dengan setiap *neuron* yang ada di *layer output*. Jaringan ini hanya menerima *input* kemudian secara langsung akan mengolahnya menjadi outout tanpa harus melalui proses lainnya. Contoh penerapan jaringan lapisan tunggal yaitu Adaline, Hopefield, Perceptron.



Gambar 2.4 Arsitektur Jaringan Syaraf Tunggal [32]

2. Jaringan multi tingkat (*multilayer network*), jaringan ini menambahkan 1 lapisan pemrosesan diantara lapisan *input* dan *output*, yaitu hidden *layer*. Di hidden *layer*, informasi diproses berdasarkan beberapa syarat untuk dapat diteruskan. Jaringan multi tingkat ini dapat menyelesaikan permasalahan yang lebih kompleks dibandingkan jaringan lapisan tunggal. Namun dalam segi waktu, karena memerlukan proses tambahan yang ada di hidden *layer* membuat proses latihan dan hasil cenderung lebih lama. Contoh algoritma ANN yang menggunakan metode ini yaitu Madaline, *Backproagation, Neocognitron.*

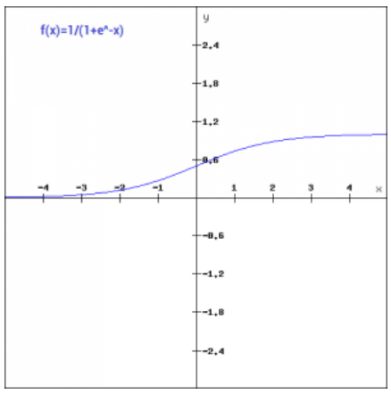


Gambar 2.5 Arsitektur Jaringan Syaraf Multi Tingkat [33]

Learning rate adalah salah satu parameter yang penting dalam *training* data, ini akan men-koreksi bobot pada waktu proses pelatihan. Nilai *learning rate* ini berada pada range 0 sampai 1. Dalam proses penentuan nilai, jika nilai *learning rate* terlalu besar maka proses pemberian bobot akan cenderung lebih cepat, karena mesin akan memberikan nilai bobot dengan proses yang lebih sederhana sehingga ketelitiannya akan semakin berkurang. Sebaliknya jika *learning rate* semakin kecil maka ketelitiannya lebih bagus, namun waktu yang dibutuhkan dalam proses pemberian nilai bobot semakin lama. Maka *learning rate* harus disesuaikan agar memberikan solusi terbaik (*overshooting*) [34].

Fungsi aktivasi berperan apakah sinyal dari *input neuron* harus diteruskan atau tidak yang diatur dari bobot dan fungsi apa yang diterapkan pada suatu *neuron*. Fungsi aktivasi merupakan parameter penting yang menentukan fungsi matematika yang akan diterapkan oleh model machine *learning* untuk memproses data *input*, sehingga dapat menghasilkan *output* yang diharapkan [35]. Ada beberapa fungsi aktivasi yang sering digunakan dalam ANN, yaitu:

1. Fungsi sigmoid



Gambar 2.6 Kurva Fungsi Sigmoid[36]

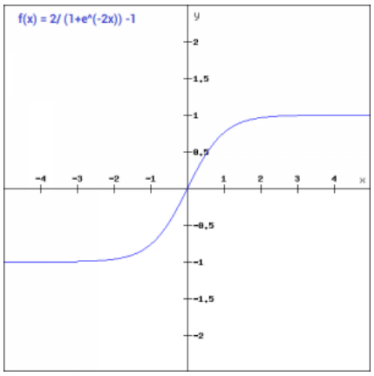
Fungsi sigmoid merupakan fungsi non-linear yang memiliki nilai dengan range 0 sampai dengan 1, kurva sigmoid memiliki bentuk **S** atau dikenal sebagai kurva logistik. Fungsi ini mengubah nilai *input* kontinu menjadi *output* yang berada pada rentang antara 0 dan 1. Pada sumbu **x**, kita melihat *input*nya, sedangkan pada sumbu **y**, kita melihat *output* dari fungsi tersebut. Jika digambarkan dengan rumus maka fungsi sigmoid dapat dirumuskan menjadi:

$$F(x) = \frac{1}{1 + e^{-x}} \tag{2.9}$$

Diturunkan menjadi :

$$F'(x) = (F(x)(1 - F(x))) \tag{2.10}$$

2. Fungsi TanH



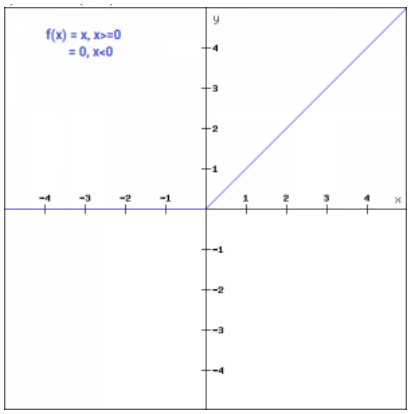
Gambar 2.7 Kurva Fungsi TanH[36]

Singkatan dari Tangen Hiperbolik. Fungsi TanH mirip dengan fungsi sigmoid teteapi simetris terhadap titik akhirnya. Fungsi Tanh menghasilkan *output* yang

berada di antara -1 dan 1 seperti yang ada pada Gambar 2.7, tidak seperti sigmoid yang hanya berkisar dari 0 hingga 1. Jika dirumuskan menjadi:

$$F(x) = \tanh(x) = \frac{2}{1 + e^{-x}} - 1 \tag{2.11}$$

3. Fungsi *Rectified Linear Unit* (ReLU)

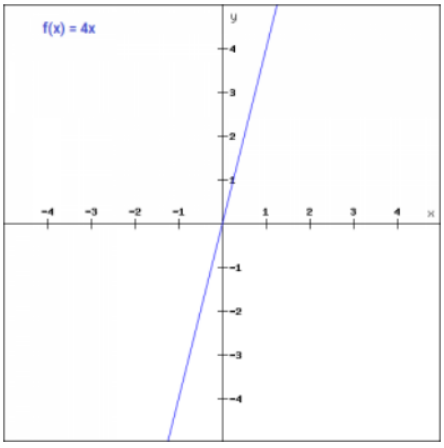


Gambar 2.8 Kurva Fungsi ReLU[36]

Ini merupakan fungsi aktivasi non-linier yang banyak digunakan di jaringan syaraf tiruan karena fungsi ini akan memberikan batas pada bilangan nol, artinya apabila x ≤ 0 maka x = 0 dan apabila x > 0 maka x = x. Jadi fungsi ini hanya mengaktifkan *neuron* jika *output* transformasi linier bukan nol. Secara sistematis dapat dirumuskan menjadi:

$$F(x) = \max(0, x) \tag{2.12}$$

4. Fungsi linear



Gambar 2.9 Kurva Fungsi Linear[36]

Di fungsi ini memiliki nilai keluaran yang konstanta dengan nilai *input* yang diberikan, menyiratkan bahwa bobot bias akan diperbarui, dinilai tidak cocok untuk jaringan syaraf karena tidak memperbaiki kesalahan yang ada dari setiap iterasi

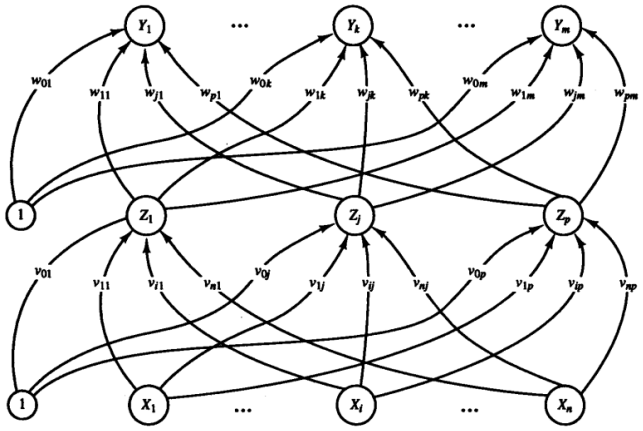
yang dilakukan. Fungsi ini sering digunakan pada lapisan *output* ketika prediksi nilai kontinu seperti dalam regresi. Jika dirumuskan menjadi: $F(x) = x$.

Selain dari 4 fungsi diatas, terdapat fungsi lain yang kiranya dapat membantu pembangunan arsitektur *machine learning*, namun dalam konteks ini yaitu jaringan syaraf tiruan prediksi beberapa fitur, maka hanya dipilih 4 saja. Selain itu masih ada fungsi lain sepeti *Binary Step Function, Leaky ReLU, Parametrized ReLU, Exponential Linear Unit, Swish*, dan *SoftMax* [36]. *Batch size*, atau ukuran batch, adalah istilah yang digunakan dalam konteks pembelajaran mesin, yang mengacu pada jumlah sampel pelatihan yang diproses dalam satu iterasi. Ukuran *batch* ini merupakan *hyperparameter* yang menentukan jumlah data yang diproses sebelum parameter model diperbarui. Setelah seluruh batch selesai, prediksi hasil dibandingkan dengan target *output*, dan kesalahan (*error*) dihitung. Berdasarkan *error* tersebut, algoritma pembaruan diterapkan untuk meningkatkan kinerja model. Dataset pelatihan dapat dipecah menjadi beberapa *batch*. Biasanya, ukuran *batch* yang sering digunakan adalah 32, 64, atau 128.

Optimizer adalah fungsi yang membantu dalam memilih bobot model yang optimal, sehingga dapat mengurangi kesalahan secara signifikan selama proses pelatihan. Salah satu optimizer yang populer adalah Adam, sebuah metode dengan *learning rate adaptif*. Adam menghitung *learning rate* khusus untuk setiap parameter dalam model. Nama "Adam" merupakan singkatan dari "*adaptive moment estimation*," karena ia menggunakan estimasi dari momen pertama dan kedua dari gradien untuk menyesuaikan *learning rate* setiap bobot di jaringan *neuron* [37].

2.8.1. *Backpropagation*

Backpropagation adalah salah satu algoritma inti yang memungkinkan jaringan saraf tiruan (*Artificial Neural Networks*) untuk belajar. Algoritma ini bekerja dengan menyesuaikan bobot pada jaringan berdasarkan perbedaan antara hasil yang diprediksi oleh jaringan dan hasil yang sebenarnya (*target output*). Terdapat beberapa tahapan dalam pelatihan data.



Gambar 2.10 Alur Backpropagation dengan 1 *Hidden Layer* [31]

Keterangan variabel yang akan digunakan dalam algoritma *backpropagation* :

X = vektor *input*
t = vektor target *output*
δ_k = nilai yang menunjukan seberapa besar kesalahan pada *output* yang dihasilkan pada unit Y_k, jika $\delta_k$semakin besar maka semakin besar juga perubahan yang terjadi pada bobot w_{jk}.
δ_j = nilai yang menunjukan seberapa besar kesalahan yang harus diperbaiki pada bobot v_{ij}, ini dihitung berdasarkan *error* dengan alur mundur dari *output layer*.
α = *learning rate.*
x_i= *input* di unit i, dimana data tidak berubah : data yang masuk = data yang keluar.
v_{oj}= bias sebelum masuk ke hidden unit Z_j.
Z_j= hidden unit j, nilai *input* di unit ini akan dilambangkan dengan z_in_j, dan didapatkan dari :

$$z\_in_j = v_{oj} + \sum_i x_i v_{ij} \qquad (2.13)$$

Nilai bias ditambahkan dengan jumlah hasil kali antara nilai asli *input* dengan bobot v_{ij} , jika pada Gambar 2.10 bias diberi nilai dengan 1 dan nilai *hidden unit* j merupakan hasil tambah dengan nilai bias "1" ini membuat nilai selalu ada (*real*). Untuk nilai *output* dari unit Z_j menggunakan rumus fungsi aktivasi yang dipakai.

Dalam algoritma *backpropagation* terdapat 3 tahapan utama dalam pelatihan model. Berikut tahapan – tahapan data dihitung dalam memperbaiki model :

Step 0 Inisialisasi bobot dan bias, nilai diubah menjadi nilai kecil acak
Step 1 **while** *stopping condition = FALSE*, **do** Step 2 - Step 9
Step 2 **For each** *pelatihan*, **do** Step 3 - Step 8

Tahap Maju (feedforward)

Step 3 : Setiap nilai *input* yang dimasukan ke unit *input* diteruskan ke *hidden unit.*

Step 4 : Setiap *input* nilai yang diteruskan, dijumlahkan dengan bobot dan bias yang ada dengan rumus pada *Equation (2.13)* .

Mengaplikasikan fungsi aktivasi pada *hidden unit* yang dipakai dengan hasil jumlah sebelumnya.

Mengirimkan hasilnya ke lapisan selanjutnya (*output layer*)

Step 5 : Setiap *output* unit menjumlahkan bobot dan bias yang ada dengan signal yang dikirimkan dengan rumus yang sama *Equation (2.13).*

Mengaplikasikan fungsi aktivasi pada *output unit* yang dipakai dengan hasil jumlah sebelumnya.

Tahap Backpropagation

Step 6 : Setiap unit Y_k menyesuaikan pola dari t dengan nilai y_k. Pola ini menunjukan seberapa besar kesalahan yang terjadi (δ_k) dengan rumus:

$$\delta_k = (t_k - y_k) f'(y\_in_k) \qquad (2.14)$$

t_k= target *output* k, dan

$f'(y\_in_k)$ = adalah turunan dari fungsi aktivasi yang digunakan.

Menghitung koreksi bobot yang akan dilakukan pada w_{jk} nanti, dengan:

$$\Delta w_{jk} = \alpha \delta_k Z_j \qquad (2.15)$$

Hasil ini juga akan digunakan untuk mengkoreksi bias yang ada dengan:

$$\Delta w_{0k} = \alpha \delta_k \qquad (2.16)$$

Setelah itu nilai δ_k dikirimkan ke lapisan belakangnya (*hidden layer*)

Step 7 : Setiap unit pada *hidden layer,* menghitung nilai *error* yang terjadi pada tiap unitnya, menggunakan rumus:

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \qquad (2.17)$$

Lalu dikalikan dengan turunan dari fungsi aktivasi yang dipakai :

$$\delta_j = \delta\_in_j f'(z\_in_j) \qquad (2.18)$$

Sehingga dapat menghitung nilai koreksi yang akan digunakan untuk memperbaharui bobot dengan:

$$\Delta v_{ij} = \alpha \delta_j x_i \qquad (2.19)$$

Dan menghitung koreksi nilai bias untuk diperbaharui nanti:

$$\Delta v_{ij} = \alpha \delta_j \qquad (2.20)$$

Catatan: δ_j tidak dikirimkan kembali karena tidak melibatkan pembaruan pada unit *input*, tetapi jika arsitektur yang digunakan adalah *backpropagation* dengan lebih dari 1 hidden *layer* maka δ_j akan diteruskan sampai ke unit *input* + 1 .

Tahap Pembaruan bobot

Step 8 : Setiap *output* unit memperbarui bobot :

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \qquad (2.21)$$

Dan biasnya:

$$w_{0k}(baru) = w_{0k}(lama) + \Delta w_{0k} \qquad (2.22)$$

Begitu pula dengan unit di *hidden layer*

Step 9 : Test stopping condition (stopping condition = True jika *epoch* sudah tercapai) [31]

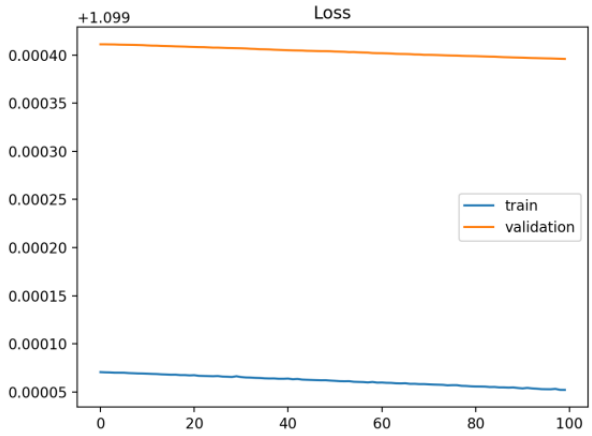
Dalam menilai kinerja model dalam memprediksi sesuatu diperlukan pengamatan tersendiri antara hasil yang diberikan dalam pembelajaran dan hasil yang diberikan dalam validasi, perbandingan antara kurva pembelajaran dan kurva validasi sering disebut sebagai *learning curve*. Kurva pembelajaran adalah alat diagnostik yang banyak digunakan dalam machine *learning* untuk algoritma yang belajar dari data *training* secara bertahap. Model dievaluasi dengan data *training* dan *data validation* setelah setiap pembaruan selama proses *training*, dan plot kinerja model yang diukur dapat menunjukkan kurva pembelajaran. Kurva pembelajaran dapat digunakan untuk mendiagnosis model yang *underfit*, *overfit*, atau *good fit* serta apakah data *training* dan data *validation* cukup *representative*. Kurva diukur dari

epochs dan *loss* yang terjadi. Berikut *learning curve* yang sering dialami dalam pelatihan model.

Overfitting adalah ketika model terlalu kompleks sehingga "mempelajari" data pelatihan dengan sangat baik, termasuk *noise* dan detail yang tidak relevan. Hal ini menyebabkan performa model sangat baik pada data pelatihan tetapi buruk saat digunakan pada data baru (uji), karena model tidak dapat digeneralisasi. Kurva pelatihan dalam kasus ini biasanya menunjukkan kesalahan yang sangat rendah pada data pelatihan tetapi tinggi pada data evaluasi.

Underfitting adalah kebalikan dari *overfitting*, di mana model terlalu sederhana untuk menangkap pola dalam data. Model tidak dapat mempelajari hubungan antara fitur dan label dengan baik, sehingga performanya buruk baik pada data pelatihan maupun data evaluasi. Kurva pelatihan dan evaluasi dalam situasi *underfitting* cenderung sama-sama menunjukkan kesalahan yang tinggi.

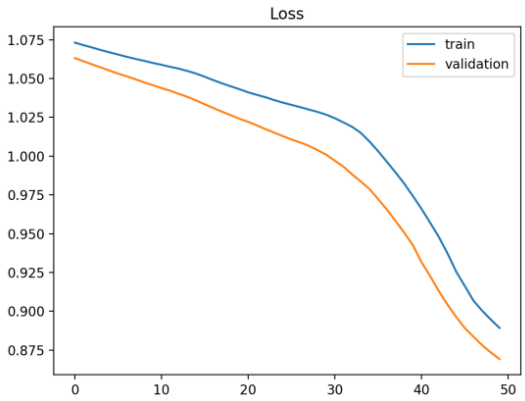
Good fit atau *fit* yang baik terjadi ketika model memiliki keseimbangan antara terlalu kompleks (*overfitting*) dan terlalu sederhana (*underfitting*). Dalam kasus ini, model mampu mempelajari pola yang relevan dari data pelatihan tanpa "mempelajari" *noise*, serta mampu digeneralisasi dengan baik pada data baru. Kurva pelatihan dan evaluasi akan memiliki kesalahan yang relatif rendah dan tidak terlalu berbeda jauh [23].



Gambar 2.11 *Learning Curve* pada *Underfit Model* (1)[23]

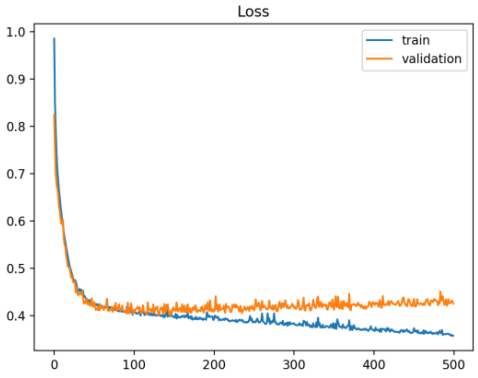
Kurva pembelajaran yang menunjukan model mengalami *underfit*. Model mengalami *underfit* karena data yangdigunakan dalam pembelajaran kurang atau

tidak memiliki kapasitas yang cukup selama pebelajaran sehingga model tidak menemukan pola yang relevan.



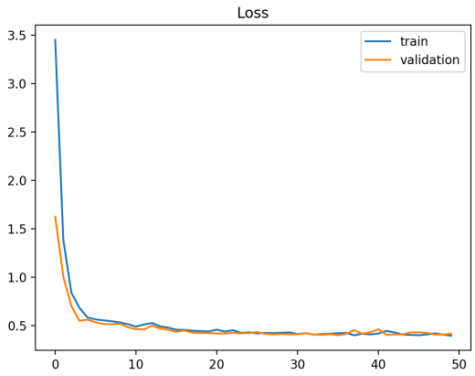
Gambar 2.12 *Learning Curve* pada *Underfit Model* (2)[23]

Kurva pembelajaran yang menunjukan model mengalami underfit. Namun berbeda dengan Gambar 2.11, terlihat bahwa data mulai mengenali pola yang ada, tapi karena hasil pelatihan dan hasil validasi tidak bersentuhan dapat disimpulkan bahwa model mengalami *underfit* karena proses pembelajaran yang kurang.



Gambar 2.13 *Learning Curve* pada *Overfit Model[23]*

Kurva model yang mengalami *overfit*. Pada setiap *epoch*, model belajar dengan baik dan kurva *loss* menurun, menandakan model menemukan pola pada dataset. Namun, setelah *epoch* ke-100, terjadi perbedaan yang semakin besar antara kurva *train* dan *validation*. Kurva train terus menurun, sedangkan kurva *validation* stagnan dan fluktuatif di sekitar nilai tertentu. Hal ini merupakan indikasi awal *overfitting*, di mana model sangat baik dalam memprediksi data pelatihan tetapi kurang mampu melakukan generalisasi pada data validasi. Kurva pembelajaran yang menunjukkan model good fit ditandai dengan penurunan loss dan pertemuan kurva train serta validasi, seperti pada Gambar 2.14.



Gambar 2.14 *Learning* Curve pada *Good Fit Model[23]*

Selain dari itu, ada masih banyak lagi pola *learning curve* yang sering ditemukan dalam membandingkan *loss* dan *epochs* pada suatu model, seperti terlihat pada Gambar 2.12 yang menunjukan model *underfit* karena dataset pelatihan yang kecil (a), model yang menunjukan *overfit* karena model terlalu terlatih pada data pelatihan dan tidak menggeneralisasi dengan baik ke data validasi, sehingga hasilnya tidak konsisten. Fluktuasi ini juga bisa disebabkan oleh noise atau variabilitas dalam data validasi, yang menyebabkan performa model tidak stabil. (b), dan kurva *underfit* pada data pelatihan dan *overfit* pada data validasi. Ini berarti model tidak cukup mempelajari data pelatihan dengan baik sehingga hasilnya kurang akurat di pelatihan (*underfitting*), tetapi entah bagaimana kinerjanya pada data validasi sangat baik (*overfitting*).

2.8.2. Optimasi Arsitektur

Untuk mendapatkan performa terbaik dari *Artificial Neural Network* (ANN), diperlukan penyesuaian arsitektur model dan parameter pelatihan. Proses ini dikenal sebagai *hyper-Parameter Optimization* (HPO), secara kesimpulan, HPO adalah proses menemukan sekumpulan parameter hiper untuk mencapai kerugian minimum atau akurasi maksimum dari suatu jaringan objektif [38].

Salah satu pendekatan yang umum digunakan dalam optimasi hyperparameter adalah *grid search*. Grid search melakukan pencarian secara menyeluruh (*exhaustive search*) terhadap kombinasi hiperparameter yang telah ditentukan oleh pengguna. Metode ini sangat bergantung pada pengetahuan awal pengguna terhadap hyperparameter yang digunakan, karena seluruh kandidat kombinasi harus

dihasilkan secara manual. Keunggulan utama grid search terletak pada kesederhanaan konsep dan implementasinya, serta kemampuannya untuk menemukan kombinasi optimal apabila sumber daya komputasi yang tersedia mencukupi. Setiap percobaan pada grid search berjalan secara independen, sehingga memungkinkan proses paralelisasi yang efisien dan fleksibel dalam pemanfaatan sumber daya [38].

2.8.3. Pengukuran Kinerja Algoritma

Tahap evaluasi adalah tahap krusial dalam pengembangan model *machine learning*, karena tahapan ini akan mengukur seberapa baik model dapat melakukan prediksi terhadap data yang belum pernah dilihat sebelumnya. *Confusion matrix* adalah salah satu pendekatan umum yang digunakan dalam mengukur kinerja model. Pendekatan ini membandingkan hasil prediksi yang dihasilkan dengan nilai asli dari data, sehingga memungkinkan analisa mendalam mengenai jenis kesalahan yang terjadi. *Confusion matrix* terdiri dari 4 komponen utama [22].

1. *True Positive* (TP), tupel positif dengan nilai label positif.
2. *True Negative* (TN), tupel negatif dengan nilai label negatif.
3. *False Positive* (FP), tupel positif dengan nilai label negatif.
4. *False Negative* (FN), tupel negatif dengan nilai positif.

Dengan menggunakan confusion matrix, performa model dapat dianalisis secara menyeluruh dan lebih mudah disajikan dalam beberapa ukuran evaluasi seperti *F1-score*, recall, dan *Matthews Correlation Coefficient* (MCC).

1. Akurasi , persentase kebenaran hasil prediksi dengan data asli.

$$akurasi = \frac{(TP + TN)}{(TP + FN + FP + TN)} \tag{2.23}$$

2. *Precision*, proporsi data yang diprediksi positif dan benar-benar positif.

$$presisi = \frac{TP}{(TP + FP)} \tag{2.24}$$

3. *Recall*, proporsi data positif yang berhasil dikenali dengan benar oleh model.

$$recall = \frac{TP}{(TP + FN)} \tag{2.25}$$

4. *F1-Score*, rata – rata harmonik dari presisi dan *recall*, digunakan saat keseimbangan keduanya penting.

$$F1Score = 2 \times \frac{presisi \times recall}{presisi + recall} \quad (2.26)$$

5. MCC, nilai korelasi antara prediksi dan label sebenarnya, megukur TP, TN, FP, dan FN secara seimbang, cocok untuk data tidak seimbang [39].

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \quad (2.27)$$

Permutation feature importance adalah salah satu metode yang biasa digunakan dalam analisa fitur dalam konteks *machine learning* untuk menentukan seberapa berpengaruh sebuah *input* terhadap peforma dari model yang sudah dilatih.

Permutation feature importance (PFI) mengukur peningkatan *error* prediksi model setelah nilai – nilai suatu fitur diacak, sehingga hubungan antara fitur tersebut dengan target sesungguhnya menjadi terputus.

Konsep ini sangat sederhana: Suatu fitur dianggap "penting" apabila pengacakan nilainya menyebabkan *error* model meningkat, karena pada kasus tersebut model sangat bergantung pada fitur tersebut dalam melakukan prediksi. Sebaliknya, fitur dikategorikan "tidak penting" apabila pengacakan nilainya tidak berpengaruh terhadap *error* model, yang menunjukkan bahwa model tidak memanfaatkan fitur tersebut dalam proses prediksi [40].

2.9. Penelitian Terkait

Dalam penyusunan, terdapat beberapa penelitian terkait yang dijadikan sebagai referensi dan rujukan mengenai metode yang digunakan, agar metode – metode yang dipilih dapat membuat tujuan dari skripsi ini mendapatkan hasil yang diinginkan. Berikut adalah penelitian – penelitian yang dipakai dalam skripsi ini sebagai rujukan:

Penelitian yang dilakukan oleh Olta Llaha dan Azir Aliu [41] dalam memvisualisasikan data mahasiswa yang terbagi menjadi 3 kategori yaitu : mahasiswa baru, mahasiswa saat ini, dan mahasiswa yang lulus. Tujuan dari

penelitian ini untuk mempermudah pihak akademik dalam meningkatkan kepekaan pihak akademik terhadap masalah yang ada, khususnya terkait dengan kinerja akademik, pendaftaran, dan kelulusan mahasiswa. Dengan menyajikan data dalam bentuk visual yang mudah dipahami, penelitian ini ingin membantu pihak akademik untuk lebih cepat dan akurat dalam mengidentifikasi masalah.

Penelitian yang dilakukan oleh Dhea Fahira dan Vivi Silvia [42], dalam mencari faktor yang mempengaruhi lama mencari kerja, data uji menggunakan data kuisioner dan data Badan Pusat Statistik (BPS), didapatkan bahwa umur adalah faktor yang paling berpengaruh terhadap lama mencari kerja, setiap bertambah umur pencari kerja, maka akan membutuhkan waktu yang lebih lama untuk mendapatkan pekerjaan. Oleh karena itu, kelulusan yang cepat dapat menjadi bekal utama dalam mengurangi risiko keterlambatan memperoleh pekerjaan dan meningkatkan peluang diterima di dunia kerja.

Penelitian untuk mencari faktor – faktor yang mempengaruhi keterlambatan penyelesaian studi mahasiswa di salah satu prodi pada universitas di Sumatera Utara [43]. Penelitian menggunakan 2 cara pendekatan untuk mendapatkan data uji, yaitu wawancara dan analisis dokumen. Didapatkan faktor yang mempengaruhi keterlambatan penyelesaian studi terbagi menjadi 2 faktor yaitu internal dan eksternal, internal yang berarti faktor yang timbul dari dirinya sendiri, seperti motivasi, bakat, dan minat. Sedangkan faktor eksternal adalah faktor yang ditimbulkan oleh aspek yang terjadi diluar sehingga mempengaruhi subjek, seperti terlambat suatu mata kuliah, ekonomi, organisasi, pergaulan dan sebagainya.

Penelitian yang dilakukan oleh Imam Riadi, dkk [44], dalam memprediksi KTW dengan KNN menggunakan 93 data. Didapatkan hasil bahwa prediksi tetap bisa dilakukan dengan hasil akurasi 78% dan presisi 100% Meski demikian, keakuratan model dapat meningkat seiring bertambahnya jumlah data pelatihan. Hal ini menunjukkan bahwa teknik *machine learning* memiliki kemampuan dan potensi yang baik dalam melakukan prediksi terhadap KTW.

Penelitian yang dilakukan oleh Isna Oktadiani dalam memprediksi Masa Studi Mahasiswa pada suatu universitas [45]. Data yang digunakan diambil dari instansi bersangkutan untuk dikembangkan model yang dapat memprediksi mahasiswa

yang belum lulus. Model menggunakan algoritma *Naïve Bayes Classifier* pada 3553 data dengan atribut program studi, IPK, pekerjaan ibu, penghasilan ibu, periode masuk, pekerjaan ayah, penghasilan ayah, jalur masuk, jenis kelamin, dan asal sekolah. Metode Naïve Bayes Classifier menunjukkan performa yang cukup baik dalam penelitian ini, namun terdapat beberapa kelemahan. Nilai presisi yang hanya mencapai 54,545% mengindikasikan bahwa separuh prediksi positif tidak sesuai dengan data sebenarnya. Sementara itu, nilai recall sebesar 67,220% menunjukkan kemampuan model yang cukup dalam menemukan data relevan, dan akurasi sebesar 79,925% tergolong baik, meskipun akurasi tinggi tidak selalu mencerminkan performa optimal pada dataset tertentu.Penulisan ini menyampaikan poin-poin utama dengan jelas, menjaga keseimbangan antara menyebutkan kekuatan dan kelemahan, serta menggunakan bahasa yang lugas.

Dalam penelitian yang dilakukan Universitas Dehasen Bengkulu, diterapkan dua algoritma *machine learning* yaitu *Decision Tree* dan *Artificial Neural Network* (ANN) untuk memprediksi kelulusan mahasiswa tepat waktu[46]. Penelitian ini menggunakan data mahasiswa angkatan 2006 – 2010 sebagai data *training* dan mahasiswa angkatan 2011 – 2012 sebagai *data testing* dengan total sampel sebanyak 600 data. Fitur yang dianalisis meliputi identitas mahasiswa, latar belakang akademik, ekonomi orang tua, serta nilai IP dan IPK semester 1 hingga 4. Berdasarkan hasil pengujian, ANN menunjukkan tingkat akurasi lebih tinggi sebesar 79.74% dibanding *Decision Tree* yang memperoleh akurasi sebesar 74.51%. Penelitian ini menyimpulkan bahwa metode ANN memiliki keunggulan dalam menangani data label dan mampu memberikan prediksi yang lebih akurat terhadap kemungkinan kelulusan mahasiswa tepat waktu. Temuan ini menunjukkan relevansi penggunaan *machine learning* dalam mendukung pengambilan keputusan strategis di lingkungan pendidikan tinggi.

Penelitian yang dilakukan dalam mencari model terbaik dalam memprediksi kelulusan tepat waktu menggunakan KNN dan *Random Forest*, menggunakan data campuran dari 4 perguruan tinggi berkisar 1739 dengan 10 atribut dan didapatkan bahwa pendekatan menggunakan algoritma *Random Forest* dengan menggunakan *Particle Swarn Optimization* (PSO) memiliki akurasi 97,89% [47].

Penelitian menggunakan *Neural Network, Decision Tree* dan *Support Vector Machine* dalam memprediksi KTW pada suatu universitas di Indonesia. Data yang digunakan termuat jenis kelamin, jurusan slta, asal slta, ipk semester 1, ipk semester 2, ipk semester 3, ipk semester 3, ipk semester 4, ipk semester 5, dan ipk semester 6. Didapatkan bahwa model terbaik dalam memprediksi yaitu SVM dengan nilai akurasi 85,18% [48].

Penelitian yang dilakukan Ayman Alfahid dalam membandingkan model CART, CIT, SVM, GLM, ANN, *Naïve Bayes*, *Bagged* CART, *Random Forest*, ADABOOST, dan XGBOOST dalam memprediksi *drop out* mahasiswa *Saudi University* .Didapatkan bahwa SVM dan *Random Forest* adalah model yang terbaik karena menghasilkan peforma yang seimbang (akurasi, presisi, *recall*, MCC). Sebaliknya, Naïve Bayes adalah model yang tertinggi [49].

Beradasarkan dari penelitian – penelitian diatas, dalam memprediksi kelulusan tepat waktu, dapat disimpulkan dalam tabel berikut:

Tabel 2. 4 Rangkuman Penelitian Terkait

No.	Peneliti	Objek	Algoritma	Jumlah Fitur	Evaluasi
1.	Isna Oktadiani, dkk (2023) [45]	Perguruan tinggi X.	Naïve Bayes Classifier,	12	ACC :79.93% Precision :54,54% Recall :67,22% AUC :0,844
2.	Eko Prasetiyo Rohmawan (2019) [46]	KTW Universitas Dehasen Bengkulu Teknik Informatika (2009 - 2011)	*Decision Tree,* ANN	17	ACC: ANN :79.74% DT :74.51%
3.	Indra Irawan, dkk (2023) [47]	Gabungan 4 data Universitas	KNN, *Random Forest*	10	ACC: RF : 97,89% KNN : 96,74%
4.	Verry Riyanto, dkk (2019) [48]	Suatu Universitas di Indonesia	*Neural Network, Decision Tree,* SVM	9	ACC: SVM : 85,18% DT : 84,96% NN :84,68%

No.	Peneliti	Objek	Algoritma	Jumlah Fitur	Evaluasi
5.	Ayman Alfahid (2024) [49]	Universitas Saudi	CART, CIT, SVM, GLM, ANN, *Naïve Bayes*, *Bagged* CART, *Random Forest*, *ADABOOST,* *XGBOOST*	10	ACC: CART :79% CIT :81% SVM :83% GLM :82% ANN :81% *Naïve Bayes* :70% *Bagged* CART :81% *Random Forest* :83% *ADABOOST* :80% *XGBOOST* :82%

Terdapat beberapa hasil yang menunjukan bahwa model *Decision Tree* (DT) yang terbaik dalam memprediksi kelulusan tepat waktu. Dalam beberapa kasus, model ANN menunjukan peforma yang terbaik dalam memprediksi kelulusan tepat waktu. Seperti pada penelitian Verry Riyanto dkk., akurasi ANN mencapai 84,68% dan SVM dengan (85,18%). Sedangkan pada penelitian Eko Prasetiyo Rohmawan, ANN memberikan hasil akurasi yang lebih tinggi dari DT, dan pada penelitian Ayman Alfahid, ANN dan DT sama – sama memperoleh akurasi 81%.

Kesimpulannya tidak ada satu algoritma yang selalu unggul di semua kasus. ANN, DT, dan SBM saling bergantian memberikan hasil terbaik, tergantung pada karakteristik data dan objek penelitian. Namun, ANN memiliki keunggulan dalam kestabilan peforma. Akurasi ANN cenderung konsisten di atas 80% pada berbagai studi, dan tidak pernah menjadi yang terburuk di antara algoritma yang dibandingkan.

III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat

Penelitian dimulai dari akhir bulan Dessember 2024 dan diperkirakan akan rangkum pada Semptember 2025. Bertempatan di Laboratorium Komputer Jurusan Teknik Elektro Universitas Lampung dengan waktu penelitian pada Tabel 3.1.

Tabel 3.1 Jadwal Penelitian

No.	Kegiatan	2024	2025								
		Des	Jan	Feb	Mar	Apr	Mei	Jun	Jul	Agu	Sep
1.	Studi literatur										
2.	Business Understanding										
3.	Data Understanding										
4.	Data Preparation										
5.	Modelling										
6.	Evaluasi										
7.	Analisis hasil										

3.2. Alat dan Bahan Penelitian

Adapun alat dan bahan dari penelitian ini adalah sebagai berikut:

3.2.1. Alat

Penelitian ini menggunakan perangkat keras dan (*hardware*) dan perangkat lunak (*software*) dengan spesifikasi yang ditunjukan pada Tabel 3.2.

Tabel 3.2 Alat Penelitian

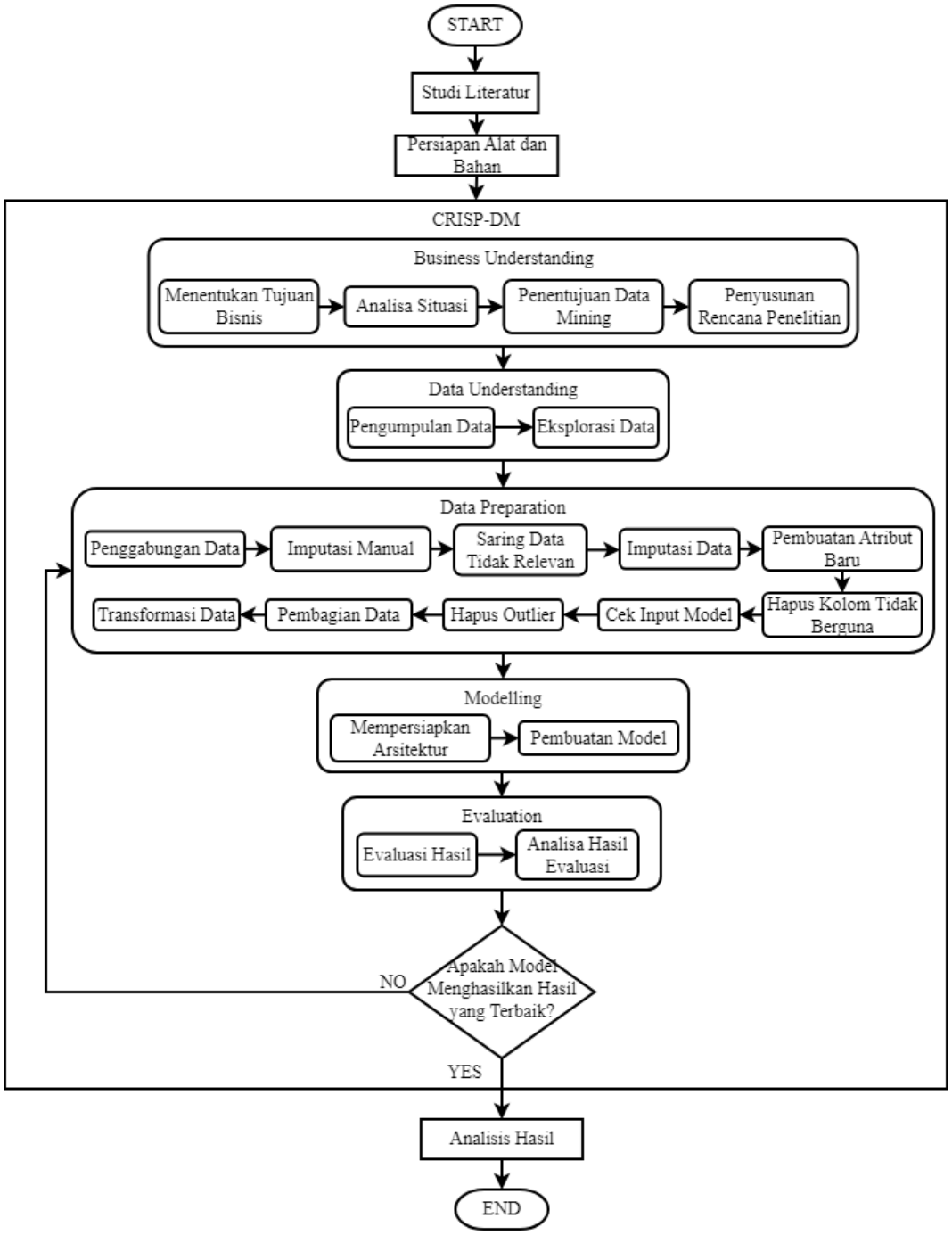
No.	Nama Alat	Spesifikasi	Deskripsi
1.	Komputer	• *Intel@Core^{tm} I5-6200U 2,3GHz Quad Core* • Ram 8 GB • Windows 10	Perangkat keras yang digunakan untuk pengembangan dan pengujian model machine *learning*.
2.	Code Editor	• Visual Studio Code v1.93.1	Kode editor dalam menulis kode dalam pengembangan penelitian.
3.	Library	• Tensorflow v4.19	Library untuk penerapan pembelajaran mesin penelitian.
4.	Google Colab	• Python 3.11.13	Perangkat lunak yang digunakan untuk menghitung *mutual information*

3.2.2. Bahan

Dalam penelitian ini, data yang digunakan adalah data mahasiswa dari suatu universitas di Lampung, baik mahasiswa yang sudah lulus maupun yang belum lulus dari angkatan 2018 hingga 2023 dengan total 44981 *records*.

3.3. Tahapan Penelitian

Dalam penelitian terdapat beberapa tahapan agar penelitian ini berjalan dengan lancar, tahapan awal adalah tinjauan pustaka, yaitu mencari dan mempelajari teori serta hasil – hasil dari penelitian sebelumnya yang relevan. Sumber yang digunakan berasal dari jurnal ilmiah, buku, maupun artikel. Pengetahuan yang diperoleh dari tahapan ini diharapkan dapat digunakan sebagai dasar teoritis untuk mendukung proses penelitian. Tahapan persiapan alat dan bahan dilakukan untuk memenuhi persiapan akan penelitian, proses utama dalam penelitian ini mengikuti kerangka kerja CRISP-DM, yang mencakup tahapan pemahaman bisnis, pemahaman data, persiapan data, pembangunan model, evaluasi dan implementasi. Tahapan berikutnya adalah penyusunan laporan penelitian yang memuat seluruh hasil dan proses yang telah dilalui, sehingga laporan dapat dipresentasikan dalam seminar hasil serta seminar komprehensif.



Gambar 3.1 Tahapan Penelitian

3.3.1. Tahap Pemahaman Bisnis

Tahapan ini bertujuan untuk memahami permasalahan yang akan diselesaikan melalui serangkaian proses, yaitu menentukan tujuan bisnis, menliai situasi yang terjadi, menetapkan tujuan *data mining*, dan penyusunan rencana penelitian. Langkah ini penting dilakukan agar proses pengelohan data dan pembangunan

model prediktif tetap sejalan dengan kebutuhan dan masalah yang ingin diatasi. Tujuan bisnis dari penelitian ini membantu pihak akademik universitas terkait dalam meningkatkan mutu pendidikan melalui Indikator Kinerja Utama (IKU) melalui meningkatkan persentase Kelulusan Tepat Waktu (KTW). Situasi yang terjadi saat ini menunjukan bahwa identifikasi mahasiswa yang beresiko tidak lulus tepat waktu masih dilakukan secara manual dan belum memanfaatkan data historis secara tepat, sehingga intervensi sering terlambat. Oleh karena itu, tujuan *data mining* ditetapkan yaitu untuk membangun model untuk prediksi KTW berbasis data akademik dan berdasarkan keseluruhan tahapan tersebut, dirumuskan rencana penelitian yang sesuai.

3.3.2. Tahap Pemahaman Data

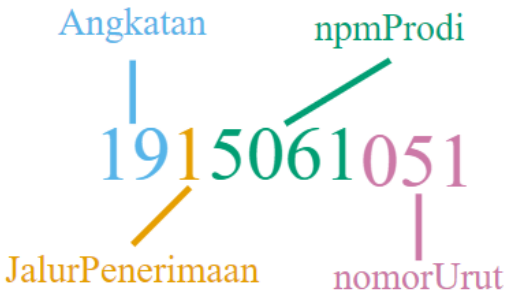
Tahapan ini bertujuan untuk memperoleh wawasan awal terhadap karakteristik data yang akan digunakan. Proses dimulai dari pengumpulan data dari sumber yang relevan, diikuti eksplorasi untuk memahami struktur, isi, dan kualitas data. Eksplorasi mencakup pemeriksaan jenis data, distribusi nilai, dan identifikasi atribut – atribut yang tersedia, mencari informasi tersembunyi yang dapat memperkaya fitur dalam proses *data mining*. Selanjutnya dilakukan analisis deskriptif dan visual terhadap data, seperti distribusi mahasiswa yang lulus tepat waktu dan tidak, untuk memahami kondisi secara menyeluruh. Proses ini juga mencakup identifikasi atribut yang relevan terhadap kelulusan menggunakan perhitungan *Mutual Information* (MI), yang membantu dalam pemilihan fitur agar pembangunan model lebih optimal.

3.3.3. *Data Preparation*

Tahapan ini mencakup proses penggabungan data, imputasi manual, penyaringan data tidak relevan, imputasi lanjutan, pembuatan atribut baru, penghapusan kolom tidak perlu, pemeriksaan input model, penghapusan *outlier*, pembagian data, dan transformasi data. Imputasi manual dilakukan karena data yang diterima tidak selalu lengkap atau sempurna. Oleh karena itu, beberapa atribut dilengkapi secara

manual menggunakan sumber data yang relevan dan terpercaya untuk memastikan keakuratan.

Penyaringan data tidak relevan dilakukan untuk memastikan data hanya mencakup mahasiswa jenjang Strata 1 (S1). Meskipun data awal diambil berdasarkan pola digit tertentu pada NPM yang mengindikasikan jenjang S1, nyatanya masih ditemukan data dari jenjang lain seperti D3, D4, Magister, Spesialis, Profesi, dan S3. Oleh karena itu, dilakukan penyaringan tambahan dengan menghapus entri yang mengandung kata – kata tersebut pada kolom namaProdi dan fakultas. Tahap ini juga mencakup deteksi dan penanganan duplikasi NPM guna menjaga keunikan data. Imputasi Data dilakukan untuk melengkapi atribut yang tidak tersedia secara langsung dalam dataset awal, seperti fakultas, namaProdi, angkatan, dan jalurPenerimaan. Proses pelengkapan ini dilakukan secara manual dengan merujuk pada struktur standar NPM mahasiswa yang terdiri dari 10 digit. Masing – masing digit atau rangkaian digit pada NPM mengindikasikan informasi tertentu seperti tahun masuk, kode fakultas, dan program studi. Ilustrasi struktur NPM dapat dilihat pada Gambar 3.2. Sementara itu, atribut kodeProdi dilengkapi melalui referensi yang relevan dan aman untuk memastikan akurasi data.



Gambar 3.2 Struktur digit NPM

Pada tahap pembuatan atribut baru, dua atribut baru yang ditambahkan adalah “durasi“ dan “label”. Atribut durasi merepresentasikan lama studi mahasiswa dalam satuan tahun, yang dihitung berdasarkan selisih semester antara awal masuk dan semester terakhir yang tercatat. Selisih ini kemudian dikonversi ke tahun dan dibulatkan ke setengah tahun terdekat agar hasil lebih representatif. Sementara itu, atribut label digunakan untuk menunjukkan apakah mahasiswa menyelesaikan studi tepat waktu. Mahasiswa dikategorikan tepat waktu (label = 1) apabila statusnya

sudah tidak aktif dan lama studinya kurang dari atau sama dengan 4 tahun. Sebaliknya, mahasiswa yang masih aktif atau memiliki durasi studi lebih dari 4 tahun diberi label 0.

Selanjutnya dilakukan pembagian data, tahap ini bertujuan untuk memisahkan data menjadi data latih dan data uji dengan mempertahankan keragaman kombinasi atribut penting untuk fitur yang digunakan sebagai *input*. Pendekatan ini dipilih untuk menghindari ketidakseimbangan distribusi yang dapat terjadi apabila pembagian dilakukan secara acak murni, misalnya ketika suatu kombinasi atribut tertentu hanya terdapat pada salah satu subset. Metode pembagian dilakukan dengan terlebih dahulu mengelompokkan data berdasarkan kombinasi atribut, kemudian secara acak memilih sebagian kecil dari setiap kelompok untuk dimasukkan ke data uji sesuai rasio yang ditentukan. Jika jumlah data uji belum mencukupi, pemilihan tambahan dilakukan secara acak dari keseluruhan data yang tersisa. Seluruh data asli tetap digunakan dalam data latih, sementara data uji menjadi subset yang representatif terhadap keragaman populasi. Dengan cara ini, diharapkan hasil pengujian dapat mencerminkan performa model secara lebih realistis tanpa mengorbankan keberagaman data latih.

3.3.4. *Modelling*

Model dibangun menggunakan algoritma *Artificial Neural Network* (ANN) yang diimplementasikan dengan TensorFlow.js. Fungsi aktivasi pada layer terakhir menggunakan sigmoid, yang menghasilkan *output* probabilitas antara 0 dan 1 untuk menentukan prediksi kelulusan tepat waktu. Untuk penentuan *hyperparameter* digunakan pendekatan campuran. *Grid search* dilakukan pada parameter yang dianggap paling berpengaruh terhadap kapasitas model dan perilaku pelatihan — yaitu jumlah neuron per layer, jumlah hidden layer, bentuk arsitektur jaringan, dan learning rate, sedangkan *hyperparameter* lainnya seperti fungsi aktivasi *hidden layer*, *dropout*, dan lainnya disesuaikan melalui penalaan manual bertahap (*manual iterative tuning*). Beberapa konfigurasi model ditetapkan dan tidak diuji, adapun konfigurasi tetap yang digunakan adalah sebagai berikut:

Tabel 3.3 Konfigurasi tetap ANN

Parameter		Nilai
Input layer neuron		85
Fungsi aktivasi pada *layer output*		Sigmoid
Jumlah *epoch*		1000
shuffle		True
Fungsi *loss*		binaryCrossentropy
optimizer		Adam
batchsize		Total data/50 ATAU minimum:32
callback	*EarlyStopping*	• monitor : val_loss, Memperhatikan validasi *loss*. • Patience :150, *epoch* tanpa peningkatan sebelum berhenti. • restoreBestWeight : false, memuat bobot terbaik setelah berhenti awal.
	ReduceLROnPlateau	• monitor :val_loss, Memperhatikan validasi *loss.* • factor : 0.5, *learning rate* akan dikalikan 0.5 jika fungsi aktif. • Patience : patience_earlyStopping/15, Jika validasi loss tidak membaik selama patience pada EarlyStopping/5 maka fungsi akan diaktifkan. • Min_lr : 1×10^{-8}, Nilai minimum untuk *learning rate.*
	modelCheckpoint	• Monitor : val_loss, Memperhatikan validasi *loss.* • saveBestOnly : true, Menyimpan konfigurasi bobot terbaik menurut parameter monitor.

Dalam proses pelatihan model *Artificial Neural Network*, digunakan tiga jenis *callback* untuk mengoptimalkan proses *training* dan mencegah *overfitting*. *Callback* pertama adalah EarlyStopping, berfungsi untuk menghentikan proses pelatihan ketika tidak ada perbaikan pada *validation loss* dalam sejumlah *epoch* berturut – turut, yang disebut dengan *patience*. *Callback* ini membantu menghindari *overfitting* dan membuang waktu pada *epoch* yang tidak produktif. *Callback* kedua yaitu ReduceLROnPlateau, berfungsi untuk menurunkan *learning rate* secara otomatis ketika *validation loss* mengalami stagnasi. Ini membantu model keluar

dari *plateau* (dataran datar) dalam fungsi *loss*, dan memperhalus proses konvergensi. *Callback* ketiga adalah modelCheckpoint, yang digunakan untuk menyimpan bobot model terbaik selama pelatihan, berdasarkan nilai *validation loss* terendah. Dengan ini, meskipun *training* sempat *overfitting* di akhir, model dapat dikembalikan ke kondisi terbaiknya.

HPO dilakukan dengan pendekatan gabungan *grid search* dan *manual iterative tuning* untuk menyeimbangkan kebutuhan eksplorasi ruang *hyperparameter* dengan keterbatasan sumber daya komputasi dan waktu. Pendekatan *hybrid* ini dipilih karena *grid search* pada seluruh kombinasi akan sangat mahal secara komputasi, terutama mengingat setiap konfigurasi kandidat dijalankan berulang kali untuk memperoleh estimasi performa yang stabil, oleh karena itu, *grid search* difokuskan pada parameter – parameter yang paling berpengaruh untuk menangkap interaksi antar-parameter tersebut, sedangkan parameter lain yang bersifat *fine-tuning* akan dioptimalkan secara manual dan bertahap setelah kandidat arsitektur terpilih. Hasil ringkasan eksperimen *grid search* disajikan pada tabel berikut:

Tabel 3.4 Tahapan eksplorasi *grid search*

Parameter	**Nilai**
Jumlah neuron setiap *hidden layer*	8, 16, 32, 64
Jumlah *hidden layer*	1, 2, 3, 4, 5
Learning rate	$1 \times 10^{-2}, 1 \times 10^{-3}, 1 \times 10^{-4}, 1 \times 10^{-5}$

Eksplorasi *grid search* dilakukan pada 4 parameter yaitu jumlah neuron pada setiap *hidden layer* yang dilakukan dengan 8 neuron dilanjutkan dengan step $\times 2$ dengan nilai maksimum 64. Untuk eksplorasi jumlah *hidden layer* dimulai dari 1 dilanjutkan dengan step 1 dengan nilai maksimum 7. Dan terakhir eksplorasi *learning rate* dilakukan dengan *step* 10^{-n}, dengan nilai n bertahap dari 2 hingga 5. Secara total, dilakukan eksplorasi sebanyak 80 dengan 10 kali *training* dalam setiap konfigurasi. Parameter – parameter ini dipilih untuk menggunakan pendekatan *grid search* karena beberapa penelitian terdahulu menunjukan bahwa parameter tersebut mempunyau pengaruh yang sangat besar terhadap peforma model dan saling berinteraksi [50][51]. Setelah kandidat parameter *grid search* selesai, dilakukan *manual iterative tuning* secara bertahap terhadap hyperparameter lainnya.

Tabel 3.5 Tahapan eksplorasi *manual tuning*

No.	Parameter	Nilai
1.	Bentuk arsitektur jaringan	Membesar ke *output*, mengecil ke *output*, *flat*, Diamond
2.	Fungsi aktivasi	ReLU, Tanh, ELU, SELU, Softplus, Swish, Mish, GELU, HardSigmoid, leakyReLU, PreLU
3.	Inisialisasi bias	Dengan inisialisasi bias, tanpa inisialisasi bias
4.	*Dropout*	tanpa *dropout*; 0.1; 0.2; 0.3
5.	*ClassWeight*	Dengan *ClassWeright,* tanpa *ClassWeight*
6.	*Validation split*	0.1; 0.2; 0.3
7.	Rasio *data testing*	0%; 10%;; 20%

Eksplorasi bentuk arsitektur jaringan meliputi membesar ke *output* (contoh neuron per layer: 8, 16, 32, 64), mengecil ke *output* (contoh neuron per layer: 64, 32, 16, 8), *flat* (contoh neuron per layer: 8, 8, 8, 8), dan diamond (contoh neuron per layer: 16, 32, 128, 64, 8). Penambahan *dropout* juga diuji untuk mengurangi *overfitting*, dengan rasio dropout diberikan secara merata pada setiap *hidden layer*. Proses pelatihan menggunakan data validasi dengan variasi rasio tertentu untuk membantu proses generalisasi model. Selain itu, dilakukan pengujian terhadap rasio data pengujian, termasuk rasio 0% yang sebenarnya merupakan salinan dari data 10%, sehingga tidak memotong data pelatihan. Eksperimen juga mencakup inisialisasi bias pada layer output menggunakan pendekatan $\log\frac{banyak\ 1}{banyak\ 0}$, yang membantu menyesuaikan bias awal terhadap ketidakseimbangan label pada data.

3.3.5. Evaluasi

Tahap evaluasi bertujuan menilai sejauh mana model memenuhi tujuan bisnis, yaitu memprediksi kelulusan tepat waktu secara andal. Untuk keperluan pemilihan konfigurasi model selama proses pelatihan, penentuan model terbaik dilakukan berdasarkan *validation loss*, karena metrik ini dipantau langsung selama pelatihan dan membantu memilih model yang paling stabil pada data validasi tanpa menggunakan data uji. Selain itu, evaluasi komprehensif juga dilakukan dengan

menghitung metrik klasifikasi seperti *Matthews Correlation Coefficient* (MCC), F1-score, akurasi, dan metrik lainnya untuk menggambarkan kualitas model dari berbagai sudut. Metrik – metrik ini dilaporkan dan dianalisis tetapi tidak digunakan sebagai dasar pemilihan *hyperparameter*. Tahap evaluasi juga mencakup analisis *Permutation Feature Importance* untuk mengukur seberapa besar penurunan performa yang terjadi, sehingga dapat diketahui kontribusi relatif masing – masing fitur terhadap prediksi.

V. KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan disimpulkan beberapa hal berikut:

1. Berdasarkan *Adjusted Mutual Information*, diperoleh fitur yang paling berpengaruh secara berurutan yaitu: nama prodi, fakultas, jenis kelamin, IPK, dan jalur penerimaan.
2. *Artificial Neural Network* (ANN) dengan algoritma *backprogation* berhasil diterapkan untuk memprediksi KTW mahasiswa menggunakan 4 fitur. Konfigurasi terbaik yang diperoleh: penggunaan data validasi sebesar 20%, penggunaan dropout sebesar 0,3, dua *hidden layer* dengan fungsi aktivasi SELU, serta *learning rate* sebesar 1×10^{-2}.
3. Pengimplementasian model *Artificial Neural Network* (ANN) dengan algoritma *backprogation* berhasil diaplikasikan untuk memprediksi KTW mahasiswa menggunakan 5 fitur. Konfigurasi terbaik yang diperoleh: penggunaan data validasi sebesar 30%, penggunaan dropout sebesar 0,1, tiga *hidden layer* dengan fungsi aktivasi ReLU, serta *learning rate* sebesar 1×10^{-2}.
4. Hasil evaluasi menunjukan model menghasilkan *mean validation loss* sebesar 0,6319 pada konfigurasi 4 fitur dan 0,6153 pada konfigurasi 5 fitur. Dengan peforma rata – rata akurasi 3471% hingga 52,16%, F1-score 9,29% hingga 9,24%, *recall* 47,42% hingga 37,64%, dan MCC -0,1 hingga -0,05.

5.2. Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan acuan untuk penelitian selanjutnya:

1. Mengaplikasikan algoritma atau arsitektur yang berbeda, seperti SVM, *decision tree*, atau metode lainnya, untuk melihat performa model terhadap kasus data di perguruan tinggi yang sama.
2. Menambahkan variasi fitur (kolom data) yang lebih informatif dan relevan, seperti pekerjaan orang tua, indeks prestasi (IP) per semester, usia mahasiswa, serta status bekerja atau tidak, agar model memiliki informasi yang lebih lengkap dalam melakukan prediksi.

DAFTAR PUSTAKA

[1] Direktorat Jenderal Pendidikan Tinggi Kementerian Pendidikan dan Kebudayaan, *Buku Panduan Merdeka Belajar-Kampus Merdeka*, 1 ed. Jakarta: Direktorat Jenderal Pendidikan Tinggi Kemdikbud RI, 2020.

[2] A. N. R. Attamimi *dkk.*, "Statistik Pendidikan Tinggi 2021," Sekretaris Direktorat Jenderal Pendidikan Tinggi, Jakarta, Apr 2022.

[3] Kementerian Pendidikan Kebudayaan Riset dan Teknologi, "Buku Panduan Indikator Kinerja Utama PTN," Jakarta, 2023. Diakses: 16 Desember 2024. [Daring]. Tersedia pada: dikti.kemdikbud.go.id

[4] Depdiknas, *Undang-Undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional*. Indonesia, 2003. Diakses: 16 Desember 2024. [Daring]. Tersedia pada: https://jdih.kemdikbud.go.id

[5] United Nations Educational Scientific and Cultural Organization (UNESCO), *Global Education Monitoring Report 2016: Education for people and planet: Creating a sustainable future for all*, 2 ed. France: UNESCO, 2016. doi: 10.54676/AXEQ8566.

[6] Kementrian Pendidikan dan Kebudayaan Republik Indonesia, *Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 4 Tahun 2020 Tentang Perubahan Atas Peraturan Menteri Pendidikan Dan Kebudayaan Nomor 88 Tahun 2014 Tentang Perubahan Perguruan Tinggi Negeri Menjadi Perguruan Tinggi Negeri Badan Hukum*. Indonesia, 2020. [Daring]. Tersedia pada: www.peraturan.go.id

[7] Kementrian Pendidikan dan Kebudayaan Republik Indonesia, *Peraturan Menteri Pendidikan dan Kebudayaan Republik Indonesia Nomor 88 Tahun 2014 Tentang Perubahan Perguruan Tinggi Negeri Menjadi Perguruan Tinggi Negeri Badan Hukum*. Jakarta, Indonesia, 2014.

[8] Independent Evaluation Group, *Higher Education for Development: An Evaluation of the World Bank Group's Support*. Washington, DC: World Bank, 2017. doi: 10.1596/IEG113867.

[9] C. Schröer, F. Kruse, dan J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model," Elsevier B.V., Chicago, Jan 2021. doi: 10.1016/j.procs.2021.01.199.

[10] P. Chapman *dkk.*, "CRISP-DM 1.0 Step-by-step data mining guide," SPSS inc., Chicago, Agu 2000.

[11] J. Resig, B. Bibeault, dan J. Maras, *Secrets of the JavaScript Ninja*, 2 ed. New York: Manning Publications Co., 2016. [Daring]. Tersedia pada: www.allitebooks.com

[12] D. Flanagan, *JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language*, 7 ed. California: O'Reilly Media, 2020.

[13] Tensorflow Js Team, "Introduction to TensorFlow." Diakses: 2 September 2024. [Daring]. Tersedia pada: https://www.tensorflow.org/learn?hl=id

[14] S. Abrahams, D. Hafner, E. Erwitt, dan A. Scarpinelli, *TensorFlow for Machine Intelligence A Hands-On Introduction to Learning Algorithms*. Santa Rosa, CA: Bleeding Edge Press, 2016. Diakses: 20 September 2024. [Daring]. Tersedia pada: bleedingedgepress.com

[15] J. Wira dan G. Putra, *Pengenalan Konsep Pembelajaran Mesin dan Deep Learning*, 1.14. Tokyo: self-published, 2020. Diakses: 31 Agustus 2025. [Daring]. Tersedia pada: https://wiragotama.github.io/

[16] R. D. Kesuma dan N. Hasdyna, *Machine Learning*, 1 ed. Lhokseumawe: Unimal Press, 2020.

[17] Google, "What is Machine Learning (ML)?" Diakses: 1 September 2024. [Daring]. Tersedia pada: https://cloud.google.com/learn/what-is-machine-learning?hl=en

[18] R. K. Mishra, G. Y. S. Reddy, dan H. Pathak, "The Understanding of Deep Learning: A Comprehensive Review," *Math. Probl. Eng.*, vol. 2021, hlm. 1–15, Apr 2021, doi: 10.1155/2021/5548884.

[19] I. J. Goodfellow, Y. Bengio, dan A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[20] B. Raharjo, *Pembelajaran Mesin (Machine Learning)*. Semarang: Yayasan Prima Agus Teknik, 2021.

[21] Tensorflow Team, "Data preprocessing for ML: options and recommendations." Diakses: 4 September 2024. [Daring]. Tersedia pada: https://www.tensorflow.org/tfx/guide/tft_bestpractices

[22] J. Han, M. Kamber, dan J. Pei, *Data Mining: Concepts and Techniques*, 3 ed. Waltham, MA: Morgan Kaufmann, 2012. doi: 10.1016/C2009-0-61819-5.

[23] J. Brownlee, *Better Deep Learning | Train Faster, Reduce Overfitting, and Make Better Predictions*, V1.3. Vermont, Victoria: Machine Learning Mastery, 2018. Diakses: 31 Agustus 2025. [Daring]. Tersedia pada: https://machinelearningmastery.com/better-deep-learning/

[24] A. Zheng dan A. Casari, *Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists*, 1st ed. Sebastopol, CA: O'Reilly Media, Inc., 2018.

[25] GeeksforGeeks, “Label Encoding in Python.” Diakses: 14 September 2024. [Daring]. Tersedia pada: https://www.geeksforgeeks.org/ml-label-encoding-of-datasets-in-python/

[26] F. Rehman, “Min Max Normalization in data mining.” Diakses: 14 September 2024. [Daring]. Tersedia pada: https://t4tutorials.com/min-max-normalization-of-data-in-data-mining/

[27] A. Navot, “On the role of feature selection in machine learning,” Hebrew University, Jerusalem, 2006. Diakses: 31 Agustus 2025. [Daring]. Tersedia pada: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=81cf3e83faa347e0ba087a126325660fb92d006d

[28] S. Romano, J. Bailey, V. Nguyen, dan K. Verspoor, “Standardized Mutual Information for Clustering Comparisons: One Step Further in Adjustment for Chance,” dalam *Proceedings of the 31st International Conference on Machine Learning*, E. P. Xing dan T. Jebara, Ed., dalam Proceedings of Machine Learning Research, vol. 32. Bejing, China: PMLR, Agu 2014, hlm. 1143–1151. [Daring]. Tersedia pada: https://proceedings.mlr.press/v32/romano14.html

[29] IBM, “What is a Neural Network?” Diakses: 16 September 2024. [Daring]. Tersedia pada: https://www.ibm.com/topics/neural-networks

[30] K. Suzuki, *Artificial Neural Networks - Architectures and Applications*. Rijeka, Croatia: InTech, 2013. doi: 10.5772/3409.

[31] L. V Fausett, *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*, 1 ed. Englewood Cliff: Prentice Hall, 1994.

[32] M. Riadi, “Jaringan Saraf Tiruan (JST).” Diakses: 16 September 2024. [Daring]. Tersedia pada: https://www.kajianpustaka.com/2016/11/jaringan-saraf-tiruan-jst.html

[33] E. Eyo dan S. Abbey, “Multiclass stand-alone and ensemble machine learning algorithms utilised to classify soils based on their physico-chemical characteristics,” *Journal of Rock Mechanics and Geotechnical Engineering*, vol. 14, no. 2, hlm. 603–615, Apr 2022, doi: 10.1016/j.jrmge.2021.08.011.

[34] A. Silas, I. C. Peace, A. O. Uzoma, dan S. Abasiama Ita, “Effect of Learning Rate on Artificial Neural Network in Machine Learning,” *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 02, 2021, doi: 10.17577/IJERTV4IS020460.

[35] R. H. Dananjaya, S. Sutrisno, dan S. Fitriady, “Penerapan Artificial Neural Network (ANN) dalam Memprediksi Kapasitas Dukung Fondasi Tiang,” *Matriks Teknik Sipil*, vol. 10, no. 4, hlm. 419, Des 2022, doi: 10.20961/mateksi.v10i4.65034.

[36] S. Sharma, S. Sharma, dan A. Athaiya, “Activation Function in Neural Networks,” *International Journal of Engineering Applied Sciences and*

Technology, vol. 04, no. 12, hlm. 310–316, Mei 2020, doi: 10.33564/IJEAST.2020.v04i12.054.

[37] D. P. Kingma dan J. Ba, "Adam: A Method for Stochastic Optimization," dalam *Proceedings of the 3rd International Conference on Learning Representations (ICLR).*, San Diego, CA: arXiv, Des 2015. [Daring]. Tersedia pada: http://arxiv.org/abs/1412.6980

[38] T. Yu dan H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," *arXiv preprint*, Mar 2020, Diakses: 31 Agustus 2025. [Daring]. Tersedia pada: https://arxiv.org/abs/2003.05689

[39] D. Chicco dan G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, hlm. 6, Des 2020, doi: 10.1186/s12864-019-6413-7.

[40] C. Molnar, *Interpretable Machine Learning*, 3 ed. Munich, Germany: Lulu.com, 2025. [Daring]. Tersedia pada: https://christophm.github.io/interpretable-ml-book

[41] O. Llaha dan A. Aliu, "Student Academic Performance Evaluation Using Data Visualization Techniques," *International Conference on Trends in Advanced Research*, vol. 1, no. Vol. 1 (2023): Proceeding Book of 1st International Conference on Trends in Advanced Research ICTAR 2023, hlm. 110–115, Feb 2023, [Daring]. Tersedia pada: http://as-proceeding.com/

[42] D. Fahira dan V. Silvia, "Analisis Lama Mencari Kerja bagi Tenaga Kerja Terdidik di Kota Banda Aceh," *Jurnal Pencerahan*, vol. 17, no. 2, hlm. 36–48, Sep 2023, Diakses: 18 Desember 2024. [Daring]. Tersedia pada: https://jurnalpencerahan.org/index.php/jp/article/view/93

[43] P. Sari, T. Yolanda Virly, T. Rizki Alviani, A. Mahri Kurniawan Rambe, dan C. Fahri, "Analisis Faktor-Faktor Terlambatnya Mahasiswa Mengikuti Perkuliahan (Studi pada mahasiswa Ilmu Al-Qur'an dan Tafsir Fakultas Ushuluddin & Studi Islam UIN Sumatera Utara Tahun 2022)," *Jurnal Ilmiah Multidisiplin*, vol. 1, no. 12, hlm. 700–705, Jan 2024, doi: 10.5281/zenodo.10458922.

[44] I. Riadi, R. Umar, dan R. Anggara, "Prediksi Kelulusan Tepat Waktu Berdasarkan Riwayat Akademik Menggunakan Metode K-Nearest Neighbor," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 11, no. 2, hlm. 249–256, Apr 2024, doi: 10.25126/jtiik.20241127330.

[45] I. Oktadiani dan Helmy Fitriawan, "Penerapan Machine Learning Untuk Prediksi Masa Studi Mahasiswa di Perguruan Tinggi X," *Electrician : Jurnal Rekayasa dan Teknologi Elektro*, vol. 17, no. 3, hlm. 242–252, Sep 2023, doi: 10.23960/elc.v17n3.2529.

[46] E. P. Rohmawan, "Prediksi Kelulusan Mahasiswa Tepat Waktu Menggunakan Metode Desicion Tree dan Artificial Neural Network," *Jurnal*

Ilmiah Matrik, vol. 20. No.1, no. Vol. 20 No. 1 (2018): Jurnal Ilmiah Matrik, hlm. 21–30, Jan 2019.

[47] I. Irawan, M. R. Qisthiano, M. Syahril, dan P. M. Jakak, "Optimasi Prediksi Kelulusan Tepat Waktu: Studi Perbandingan Algoritma Random Forest dan Algoritma K-NN Berbasis PSO," *Jurnal Pengembangan Sistem Informasi dan Informatika*, vol. 4, no. 4, hlm. 26–35, Okt 2023, doi: 10.47747/jpsii.v4i4.1374.

[48] V. Riyanto, A. Hamid, dan R. Ridwansyah, "Prediction of Student Graduation Time Using the Best Algorithm," *Indonesian Journal of Artificial Intelligence and Data Mining*, vol. 2, no. 1, Mar 2019, doi: 10.24014/ijaidm.v2i1.6424.

[49] A. Alfahid, "Algorithmic Prediction of Students On-Time Graduation from the University," *TEM Journal*, vol. 13, no. 1, hlm. 692–698, Feb 2024, doi: 10.18421/TEM131-72.

[50] B. Shi, W. J. Su, dan M. I. Jordan, "On Learning Rates and Schrödinger Operators," *Journal of Machine Learning Research*, vol. 24, hlm. 1–53, Apr 2020, [Daring]. Tersedia pada: http://arxiv.org/abs/2004.06977

[51] A. Alibakshi, "Strategies to develop robust neural network models: Prediction of flash point as a case study," *Anal. Chim. Acta*, vol. 1026, hlm. 69–76, Okt 2018, doi: 10.1016/j.aca.2018.05.015.

[52] Tim Perumus Renstra, "Rencana Strategis Universitas Lampung Tahun 2016-2020," Bandar Lampung, Mei 2015.

[53] E. F. Saputra, "Unila gelar FGD Percepatan Kelulusan Tepat Waktu Melalui MBKM - ANTARA News Lampung." Diakses: 15 Agustus 2025. [Daring]. Tersedia pada: https://lampung.antaranews.com/berita/704769/unila-gelar-fgd-percepatan-kelulusan-tepat-waktu-melalui-mbkm

[54] Kementrian Pendidikan dan Kebudayaan Universitas Lampung, *Peraturan Rektor Universitas Lampung No. 19 Tahun 2020 Tentang Peraturan Akademik*. Indonesia, 2020.