PREDIKSI TURNOVER KARYAWAN DENGAN MODEL EXTREME GRADIENT BOOSTING (STUDI KASUS: UNIVERSITAS DINAMIKA)

Program Studi

S1 Sistem Informasi

Oleh:

ABIGAIL EXCELSIS DEO

21410100045

FAKULTAS TEKNOLOGI DAN INFORMATIKA

UNIVERSITAS DINAMIKA

2025

PREDIKSI TURNOVER KARYAWAN DENGAN MODEL EXTREME GRADIENT BOOSTING (STUDI KASUS: UNIVERSITAS DINAMIKA)

TUGAS AKHIR

Diajukan sebagai salah satu syarat untuk menyelesaikan
Program Sarjana

Oleh:

Nama	**: Abigail Excelsis Deo**
NIM	**: 21410100045**
Program Studi	**: S1 Sistem Informasi**

FAKULTAS TEKNOLOGI DAN INFORMATIKA
UNIVERSITAS DINAMIKA
2025

Tugas Akhir

PREDIKSI TURNOVER KARYAWAN DENGAN MODEL EXTREME GRADIENT BOOSTING (STUDI KASUS: UNIVERSITAS DINAMIKA)

Dipersiapkan dan disusun Oleh

Abigail Excelsis Deo

NIM: 21410100045

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: Selasa, 19 Agustus 2025

Susunan Dewan Pembahas

Pembimbing

I. **Agus Dwi Churniawan, S.Si., M.Kom.**

NIDN. 0723088002

II. **Pradita Maulidya Effendi, M.Kom.**

NIDN. 0720089401

Pembahas

I. **Julianto Lemantara, S.Kom., M.Eng.**

NIDN. 0722108601

Digitally signed by Julianto
Date: 2025.08.19 17:39:40 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

Untuk memperoleh gelar sarjana

Digitally signed by Julianto
Date: 2025.08.21 15:44:32 +07'00'

Julianto Lemantara, S.Kom., M.Eng.

NIDN. 0722108601

Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

Happiness can exist only in acceptance.

-George Orwell-

Kupersembahkan Tugas Akhir kepada keluarga, dosen pembimbing, dosen wali serta teman-teman yang memberi semangat dan motivasi dan semua orang yang terlibat.

PERNYATAAN

PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH

Sebagai mahasiswa **Universitas Dinamika**, Saya :

Nama	: **Abigail Excelsis Deo**
NIM	: **21410100045**
Program Studi	: **S1 Sistem Informasi**
Fakultas	: **Fakultas Teknologi dan Informatika**
Jenis Karya	: **Laporan Tugas Akhir**
Judul Karya	: **PREDIKSI TURNOVER KARYAWAN DENGAN MODEL EXTREME GRADIENT BOOSTING (STUDI KASUS: UNIVERSITAS DINAMIKA)**

Menyatakan dengan sesungguhnya bahwa :

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, Saya menyetujui memberikan kepada **Universitas Dinamika** Hak Bebas Royalti Non-Eksklusif (***Non-Exclusive Royalty Free Right***) atas seluruh isi/sebagian karya ilmiah Saya tersebut diatas untuk disimpan, dialihmediakan, dan dikelola dalam bentuk pangkalan data (***database***) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama Saya sebagai penulis atau pencipta dan sebagai pemilik Hak Cipta.
2. Karya tersebut diatas adalah hasil karya asli Saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya, atau pendapat orang lain yang ada dalam karya ilmiah ini semata-mata hanya sebagai rujukan yang dicantumkan dalam Daftar Pustaka Saya.
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiasi pada karya ilmiah ini, maka Saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada Saya.

Demikan surat pernyataan ini Saya buat dengan sebenar-benarnya.

Surabaya, 31 Juli 2025

Abigail Excelsis Deo
NIM : 21410100045

ABSTRAK

Tingginya tingkat turnover karyawan merupakan tantangan signifikan bagi organisasi, termasuk institusi pendidikan tinggi, karena dapat menyebabkan kerugian finansial dan operasional. Penelitian ini bertujuan untuk membangun dan mengevaluasi model machine learning untuk memprediksi turnover karyawan di Universitas Dinamika. Metode yang digunakan adalah Extreme Gradient Boosting (XGBoost), sebuah algoritma ensemble learning yang dikenal memiliki performa tinggi. Data yang digunakan adalah data karyawan Universitas Dinamika selama tiga tahun terakhir, yang mencakup fitur – fitur seperti umur, jenis kelamin, status nikah, tipe karyawan, lama kerja, jarak tinggal, dan presensi, seperti tepat waktu, terlambat dengan ijin, terlambat tanpa ijin, ijin, dst. Proses penelitian meliputi beberapa tahap, mulai dari data preprocessing hingga pelatihan model menggunakan 5-Fold Cross Validation untuk memastikan evaluasi yang robust. Hasil evaluasi menunjukkan bahwa model XGBoost mampu memprediksi turnover cukup baik, dengan rata-rata akurasi sebesar 93,66% ± 3,31%, presisi sebesar 85,24% ± 15,68%, recall sebesar 73.33% ± 17,00% dan F1-Score sebesar 76,77% ± 11,22% pada data validasi. Analisis feature importance mengidentifikasi bahwa alpha, lama kerja, dan tepat waktu merupakan tiga faktor paling berpengaruh dalam prediksi. Model yang telah dilatih kemudian diimplementasikan ke dalam sebuah dashboard website interaktif menggunakan Streamlit. Dashboard ini menyajikan hasil prediksi, tingkat risiko, serta analisis faktor pendorong turnover secara visual, sehingga dapat berfungsi sebagai sistem pendukung keputusan yang praktis bagi manajemen untuk merancang strategi retensi karyawan yang lebih efektif dan berbasis data.

Kata Kunci: *turnover karyawan, prediksi, machine learning, XGBoost, sistem pendukung keputusan*

KATA PENGANTAR

Puji dan syukur disampaikan kepada Tuhan Yang Maha Esa, karena rahmat dan karunia-Nya, penulis dapat menyelesaikan laporan tugas akhir yang berjudul "Prediksi Turnover Karyawan dengan Model Extreme Gradient Boosting (Studi Kasus: Universitas Dinamika)". Laporan Tugas Akhir disusun sebagai salah satu syarat untuk menyelesaikan program studi Sistem Informasi di Universitas Dinamika. Penulis sadar bahwa laporan tidak akan terselesaikan tanpa adanya dukungan, bantuan, serta bimbingan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih kepada:

1. Orang tua serta keluarga yang telah berdoa, mendukung, serta memberi semangat selama program dan penyusunan laporan berlangsung
2. Bapak Agus Dwi Churniawan selaku dosen pembimbing satu dalam penelitian Tugas Akhir
3. Ibu Pradita Maulidya Effendi selaku dosen pembimbing dua dalam penelitian Tugas Akhir
4. Bapak Julianto Lemantara selaku dosen penguji pada Tugas Akhir ini
5. Bu Oktaviani dan Bu Erna Joeniawati yang telah membantu dalam penelitian Tugas Akhir
6. Serta pihak – pihak lain yang membantu dalam pengerjaan dan penulisan laporan yang tidak dapat disebutkan secara detail

Selama masa penyusunan laporan, penulis mengetahui bahwa masih memiliki banyak kekurangan dan jauh dari kata sempurna. Oleh karena itu, kritik dan saran sangat diharapkan untuk membangun dan memperbaiki di masa mendatang. Penulis berharap bahwa laporan ini dapat bermanfaat bagi semua orang yang membaca. Demikian kata pengantar penulis sampaikan, semoga Tuhan Yang Maha Esa memberikan rahmat dan karunia-Nya kepada kita semua.

Surabaya, 1 Juli 2025

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

DAFTAR LAMPIRAN

Halaman

BAB I
PENDAHULUAN

1.1. Latar Belakang

Perusahaan yang ingin bertahan dan berkembang di era digital saat ini harus mampu mengelola sumber daya secara efektif, termasuk tenaga kerja. Praktik bisnis konvensional yang sebelumnya diandalkan kini tidak lagi cukup efisien, terutama dengan meningkatnya kompleksitas teknologi, lonjakan data, serta berubahnya preferensi konsumen (Maharana dkk., 2022). Dalam konteks ini, digitalisasi menjadi solusi strategis yang memungkinkan perusahaan untuk mempertahankan produktivitas dan menghindari kerugian finansial (Mhatre dkk., 2020; Savitri dkk., 2024). Salah satu aspek penting dari sumber daya perusahaan yang perlu dikelola secara digital dan strategis adalah karyawan.

Tingginya angka *turnover* karyawan menjadi tantangan serius bagi perusahaan, baik dari sisi finansial maupun operasional. *Turnover* dapat menurunkan produktivitas, mengganggu stabilitas tim, serta meningkatkan biaya rekrutmen dan pelatihan (Anusha & Rajesh, 2024; Atef dkk., 2022; Yin dkk., 2024). Di Indonesia, permasalahan ini cukup signifikan. Berdasarkan data survei Hay Group yang dikutip oleh Afina Nur'aini Tsaqila & Lisa Widawati (2025), Indonesia sempat menempati peringkat ketiga negara dengan tingkat *turnover* tertinggi di dunia, yaitu sebesar 25,8%. Fenomena serupa juga terjadi di lingkungan pendidikan tinggi. Tingkat *turnover* tenaga pendidik di Jawa Timur mengalami peningkatan sekitar 7,63%, seiring dengan berkurangnya jumlah tenaga pendidik dari 34.752 orang pada tahun 2023 menjadi 32.099 orang pada tahun 2024. Hal ini mencerminkan adanya peningkatan perpindahan atau pengurangan tenaga pendidik dalam kurun waktu tersebut. Oleh karena itu, organisasi termasuk perguruan tinggi perlu memiliki strategi berbasis data untuk mengantisipasi kehilangan karyawan yang bernilai tinggi.

Universitas Dinamika, sebelumnya dikenal sebagai STIKOM Surabaya, merupakan institusi pendidikan tinggi swasta yang berfokus pada bidang teknologi dan manajemen. Saat ini, universitas ini memiliki sekitar 250 karyawan yang terdiri dari tenaga pendidik dan tenaga kependidikan. Meskipun tingkat *turnover*

karyawan belum terlihat sebagai isu yang dominan, pengembangan model prediksi turnover tetap menjadi langkah strategis yang penting. Dengan membangun sistem prediksi yang andal, universitas dapat mengantisipasi potensi kehilangan sumber daya manusia yang bernilai di masa depan dan mengambil langkah proaktif, seperti program pengembangan karier atau pemberian insentif yang sesuai. Oleh karena itu, penelitian ini bertujuan untuk mengembangkan model prediksi turnover yang disesuaikan dengan karakteristik data karyawan Universitas Dinamika, sehingga dapat diimplementasikan secara efektif apabila dibutuhkan.

Berbagai faktor diketahui memengaruhi keputusan karyawan untuk meninggalkan perusahaan, seperti usia, masa kerja, tingkat stres, serta gaji (Arromrit dkk., 2023; Kudirat Bukola Adeusi dkk., 2024; Nikunlaakso dkk., 2024). Memahami keterkaitan faktor-faktor ini melalui pendekatan analitik dapat membantu perusahaan dalam mempertahankan tenaga kerja (Chowdhury dkk., 2023). Salah satu metode yang semakin banyak digunakan untuk tujuan ini adalah machine learning, yang mampu mengolah data karyawan dan memprediksi kemungkinan terjadinya *turnover* secara lebih akurat. Model *machine learning* dapat mengidentifikasi variabel-variabel signifikan seperti gaji, masa kerja, ketidakhadiran, penilaian kinerja, jabatan, usia, jenis kelamin, dan lokasi tempat tinggal yang menjadi indikator keputusan keluar dari perusahaan (Isha dkk., 2024; Stachova dkk., 2021). Melalui hasil prediksi tersebut, perusahaan dapat mengambil langkah preventif untuk mencegah *turnover* terjadi dengan program pengembangan karyawan ataupun pemberian insentif yang relevan (Alhamad dkk., 2024).

Berbagai studi menunjukkan bahwa model *machine learning* mampu memprediksi *turnover* karyawan secara akurat (Isha dkk., 2024; Kumar dkk., 2023; Maharana dkk., 2022). Di antara banyak algoritma yang digunakan, *Extreme Gradient Boosting* (XGBoost) secara konsisten menunjukkan akurasi tinggi dan keandalan prediksi. Berdasarkan keunggulan tersebut, penelitian ini menggunakan model XGBoost untuk menguji kemampuannya dalam memprediksi *turnover* karyawan berdasarkan beragam variabel.

Berdasarkan temuan tersebut, penelitian ini bertujuan untuk menguji kembali akurasi model XGBoost dalam memprediksi *turnover* karyawan, dengan menggunakan data karyawan dari Universitas Dinamika yang mencakup beragam

fitur seperti gaji, masa kerja, ketidakhadiran, penilaian kinerja, jabatan, serta demografis. Adapun data yang digunakan dalam penelitian ini merupakan data karyawan Universitas Dinamika selama tiga tahun terakhir. Hasil dari penelitian ini diharapkan dapat menjadi referensi bagi organisasi dalam memilih model prediksi turnover yang efektif. Selain itu, untuk memudahkan implementasi, model dikembangkan dalam bentuk website agar hasil prediksi dapat diakses secara *real-time* dan digunakan langsung oleh pihak manajemen dalam pengambilan keputusan.

1.2. Rumusan Masalah

Berdasarkan latar belakang, maka dapat dirumuskan sebuah permasalahan, yaitu bagaimana membangun dan mengimplementasikan model prediksi *turnover* karyawan menggunakan algoritma XGBoost berdasarkan data karyawan Universitas Dinamika selama tiga tahun terakhir, serta menyajikannya dalam bentuk *website* interaktif yang dapat digunakan oleh pihak manajemen.

1.3. Batasan Masalah

Dalam pembuatan Tugas Akhir ini, diterapkan beberapa batasan masalah sebagai berikut:

1. Bahasa pemrograman yang digunakan adalah Python.
2. Penelitian ini hanya menggunakan algoritma *Extreme Gradient Boosting* (XGBoost) sebagai metode pemodelan.
3. Data yang digunakan merupakan data karyawan Universitas Dinamika selama tiga tahun terakhir.
4. Atribut yang dianalisis meliputi: umur, jenis kelamin, status nikah, tipe karyawan, lama kerja, presensi, dan jarak tinggal.
5. Dashboard yang digunakan pada penelitian merupakan tipe *dashboard* analitis.
6. Evaluasi dilakukan melalui skema *K-Fold Cross Validation* dengan metrik akurasi, presisi, recall, dan F1 score.
7. Model prediksi *turnover* karyawan yang dikembangkan hanya mempertimbangkan faktor-faktor internal yang terdapat dalam data, tanpa

memperhitungkan faktor eksternal seperti kompetitor perusahaan atau kondisi pasar tenaga kerja.

1.4. Tujuan

Penelitian ini bertujuan untuk membangun dan mengevaluasi model prediksi *turnover* karyawan menggunakan algoritma XGBoost, serta mengidentifikasi fitur-fitur yang paling berpengaruh terhadap keputusan karyawan untuk keluar. Model yang dihasilkan kemudian diimplementasikan dalam bentuk *dashboard* web analitis untuk digunakan sebagai alat bantu pengambilan keputusan oleh divisi SDM.

1.5. Manfaat

Manfaat yang diperoleh dari penelitian ini antara lain, yaitu:

1. Menjadi acuan bagi penelitian selanjutnya dalam mengimplementasikan algoritma XGBoost pada masalah *turnover* karyawan.
2. Memberikan dukungan keputusan, khususnya Universitas Dinamika, dalam melakukan prediksi risiko *turnover* berdasarkan data historis.
3. Menunjukkan penerapan praktis *machine learning* melalui implementasi *website* sederhana untuk memfasilitasi pengambilan keputusan berbasis data.

BAB II
LANDASAN TEORI

2.1. Penelitian Terdahulu

Penelitian terdahulu merupakan landasan penting dalam penyusunan penelitian ini. Beberapa studi yang relevan digunakan sebagai acuan dan referensi bagi penulis untuk mengembangkan penelitian yang lebih komprehensif dan mendalam. Penelitian-penelitian tersebut tidak hanya membantu dalam memahami metode dan model yang telah digunakan sebelumnya, tetapi juga memberikan wawasan tentang hasil yang telah dicapai. Tabel 2.1 berikut merangkum penelitian-penelitian terdahulu yang dijadikan sebagai bahan acuan dalam penelitian ini:

Tabel 2.1 Penelitian Terdahulu

Judul	Penulis	Hasil	Persamaan dan Perbedaan
Statistical Analysis and Prediction of Employee Turnover	Duan (2022)	Studi menggunakan beberapa algoritma, seperti Decision Tree, Random Forest, dan XGBoost untuk memprediksi *turnover* karyawan. Hasil evaluasi studi menunjukkan bahwa XGBoost memberikan performa terbaik dengan akurasi yang tinggi dan F1-score yang konsisten. Dalam studi ini turut membahas *feature* penting seperti jam kerja, usia, dan kepuasan kerja.	Studi ini menggunakan dataset publik "IBM HR Analytics Employee Attrition & Performance" serta membandingkan berbagai algoritma, termasuk XGBoost. Sementara penelitian ini akan menggunakan data internal dengan fokus utama pada algoritma XGBoost, berdasarkan temuan sebelumnya yang menunjukkan keunggulan algoritma ini dalam konteks prediksi *turnover*. Selain itu, pada penelitian ini membangun sebuah website supaya model yang dihasilkan dapat digunakan secara langsung sehingga memungkinkan untuk dilakukan analisis prediktif secara real-time.
An Intelligent Analysis and Prediction of Employee Attrition Rate in Healthcare Using Machine	Egwom .O. Jessica dkk. (2024)	Studi berfokus pada sektor kesehatan dengan menggunakan data dari Kaggle dengan menggunakan teknik resampling SMOTETOMEK untuk mengatasi data yang tidak seimbang. Algoritma yang diuji antara lain SVM, KNN, Random	Pada studi terdahulu melakukan komparasi beberapa algoritma pada data "IBM HR Analytics Employee Attrition & Performance". Sekitar 35 *feature* yang digunakan pada data set tersebut serta penerapan

Judul	Penulis	Hasil	Persamaan dan Perbedaan
Learning Techniques		Forest, dan XGBoost. Hasil studi menunjukkan bahwa Random Forest memiliki akurasi tertinggi dengan 98%, namun XGBoost juga tampil kuat dalam hal presisi dan *recall*.	SMOTETOMEK untuk mengurangi *imbalance* data, sedangkan pada studi ini akan menggunakan data Universitas Dinamika dengan *feature* yang berfokus pada faktor internal dan *feature* demografis. Selain itu, model XGBoost yang telah dibuat akan diintegrasikan ke dalam sebuah *website* sehingga analisis dapat diakses secara lebih praktis dan fleksibel.
Employee Attrition Analysis Using XGBoost	Isha dkk. (2024)	Studi berfokus untuk menemukan model algoritma yang cocok dengan "IBM HR Analytics Employee Attrition & Performance" yang terdapat pada Kaggle. Pada studi ini, Isha dkk. membagi data menjadi 80:20 dengan 24 *features* yang memiliki hubungan dengan attrition. Hasil akhir dari studi tersebut merupakan algoritma XGBoost dengan akurasi mencapai 88%. Selain itu, studi yang dilakukan Isha dkk. menunjukkan bahwa penghasilan bulanan merupakan faktor terbesar karyawan melakukan *turnover*, diikuti dengan jarang rumah dengan kantor dan umur.	Studi yang dilakukan oleh Isha dkk. menggunakan data "IBM HR Analytics Employee Attrition & Performance" dengan melakukan komparasi antar algoritma, sementara penelitian ini akan menggunakan data internal Universitas Dinamika dengan fokus pada algoritma XGBoost. Model XGBoost yang berhasil dibangun nantinya akan di-*deploy* ke dalam *website*. Hal tersebut memungkinkan akses yang lebih mudah bagi pengguna.

2.2. Karyawan

Karyawan merupakan sumber daya yang penting bagi sebuah perusahaan. Hal ini dikarenakan karyawan dapat menentukan keberhasilan dan keberlanjutan dari perusahaan atau organisasi. Pernyataan tersebut didukung oleh sebuah studi yang menyatakan bahwa karyawan yang memiliki kinerja yang baik dalam perusahaan atau organisasi (Kitwange & Habi, 2024). Sejalan dengan hasil studi sebelumnya, studi yang dilakukan oleh Rasyid dkk. menunjukkan bahwa kinerja karyawan berbanding lurus dengan keberhasilan sebuah perusahaan atau organisasi (Rasyid dkk., 2024). Oleh karena peranannya yang signifikan bagi perusahaan atau organisasi maka perusahaan perlu memperlakukan karyawannya dengan lebih berhati – hati, baik dari sisi kebijakan maupun tingkat kesejahteraan karyawan.

Dalam mempertahankan kinerja karyawan ataupun meningkatkannya, perusahaan perlu mengawasi tingkat kepuasan dari para karyawan. Tingkat kepuasan karyawan yang tinggi dapat mengurangi kemungkinan karyawan untuk keluar dari perusahaan atau *turnover*. Oleh karena itu, perusahaan perlu mengelola beban kerja pada karyawan dan menerapkan budaya kerja yang baik untuk menjaga kepuasan karyawan mereka (Yin dkk., 2024).

2.3. Turnover

Turnover merupakan fenomena organisasi yang merujuk pada tingkat pergantian karyawan dalam suatu perusahaan. Menurut Sudrajat dkk. (2024), *turnover* dapat dipahami dari tiga dimensi filosofis: ontologi, yang mendefinisikan keberadaannya dalam organisasi; epistemologi, yang mengeksplorasi pemahaman dan pengetahuan mengenai *turnover*; serta aksiologi, yang menelaah hubungan antara pengetahuan tentang *turnover* dengan nilai-nilai organisasi.

Turnover dapat berdampak signifikan terhadap berbagai aspek kinerja organisasi, termasuk produktivitas, moral karyawan, kepuasan pelanggan, dan hasil keuangan (Anusha & Rajesh, 2024). Tingkat *turnover* yang tinggi dapat meningkatkan biaya rekrutmen dan pelatihan, serta menurunkan kualitas layanan dan keterlibatan karyawan.

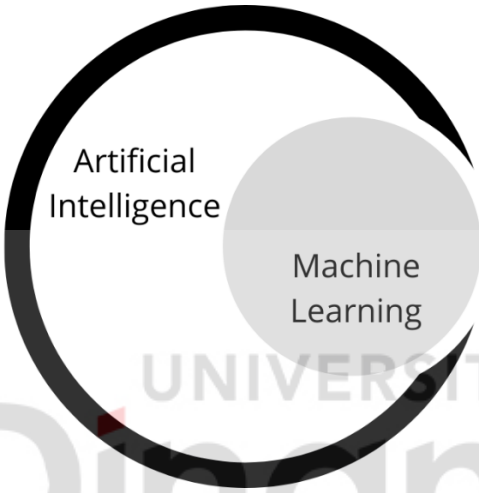
Faktor – faktor yang memengaruhi *turnover* telah banyak diteliti dalam berbagai konteks organisasi. Menurut Recilla dkk. (2024), lingkungan kerja, beban kerja, kompensasi, tekanan kerja, serta keseimbangan antara pekerjaan dan kehidupan pribadi merupakan penyebab terjadinya *turnover*. Selain itu, penelitian oleh Hom & Seo (2024), mengidentifikasi bahwa faktor *embeddedness* (keterikatan) pekerjaan, seperti keterikatan emosional dan keterlibatan sosial di tempat kerja, memainkan peran penting dalam mempertahankan karyawan.

Sejalan dengan perkembangan teknologi informasi, pendekatan berbasis data menjadi semakin relevan dalam menganalisis dan memprediksi *turnover*. Menurut Isha dkk. (2024) dan Stachova dkk. (2021), penggunaan metode machine learning memungkinkan perusahaan untuk mengidentifikasi variabel-variabel signifikan yang memengaruhi keputusan karyawan untuk keluar, seperti gaji, usia, masa kerja, ketidakhadiran, penilaian kinerja, jabatan, jenis kelamin, dan lokasi tempat tinggal.

Dengan mengenali pola-pola ini, organisasi dapat merancang strategi retensi karyawan yang lebih tepat sasaran dan berbasis data.

2.4. Machine Learning

Dalam satu dekade terakhir, Machine Learning (ML) telah menguasai perindustrian (Geron, 2019). ML, saat ini, telah digunakan dalam berbagai bidang, seperti memproses pencarian *website*, mengidentifikasi ucapan seseorang pada sebuah *handphone*, hingga mengalahkan juara dunia dalam Go.



Gambar 2.1 Diagram dari Machine Learning (ML)
(Jenis dkk., 2023)

Pada Gambar 2.1 terlihat bahwa ML merupakan salah satu bagian dari *Artificial Intelligence* (AI) (Yousef & Allmer, 2023). ML diciptakan dengan tujuan untuk menirukan kemampuan manusia dalam mempelajari pola dan beradaptasi pada kondisi – kondisi tertentu dan tidak terduga (Mueller & Massaron, 2019). Dengan mempelajari pola – pola pada data set, ML dapat lebih baik dan cepat dalam memprediksi atau menganalisis sebuah permasalahan dibandingkan manusia. Akhirnya, manusia dapat bekerja lebih efisien dengan bantuan ML. Meskipun begitu, manusia masih perlu mempertimbangkan serta menganalisis hasil dari ML dengan mempertimbangkan moral dan etika (Mueller & Massaron, 2019).

2.5. Extreme Gradient Boosting (XGBoost)

2.5.1 Pengertian XGBoost

Extreme Gradient Boosting atau XGBoost adalah algoritma yang berbasis ensemble learning yang dirancang meningkatkan efisiensi dan akurasi model prediktif (Zhang dkk., 2024). XGBoost sendiri merupakan metode yang menggabungkan beberapa pohon keputusan dengan tujuan menghasilkan prediksi yang lebih baik dibandingkan metode tunggal (Chen & Guestrin, 2016; Maulid, 2023). Untuk mencegah *overfitting*, XGBoost memiliki fitur *Regularization* yang mampu mengontrol kompleksitas model (Shaik dkk., 2024). Dengan kemampuan dan kelebihan tersebut, XGBoost digunakan dalam beberapa aplikasi seperti mendeteksi kesalahan pada sistem Photovoltaic Array, menangani data yang hilang pada model prediksi di turbin gas, melakukan prediksi *turnover* pada karyawan, dst.

2.5.2 Jenis Klasifikasi

Dalam konteks *machine learning*, klasifikasi data dapat dibedakan menjadi tiga tipe utama, yaitu:

1. Binary Classification: Target hanya memiliki dua kelas, misalnya 0 dan 1 atau True dan False. Contoh: Prediksi turnover (ya atau tidak).
2. Multiclass Classification: Target terdiri dari lebih dari dua kelas (misal: Dataset Iris memiliki kelas Setosa, Versicolor, dan Virginica).
3. Regression: Target bersifat kontinu (misal: prediksi harga rumah, suhu, dsb.)

Penentuan bentuk data dapat dilihat melelui jenis target variabel dari keseluruhan data:

1. Jika label hanya dua nilai diskrit (nilai yang berisi bilangan bulat atau dapat dihitung) maka akan termasuk ke dalam binary classification.
2. Jika tiga atau lebih nilai diskrit maka akan termasuk ke dalam tipe multiclass.
3. Jika nilai target bersifat kontinu (nilai yang diperoleh dari hasil pengukuran) maka data tersebut masuk ke dalam regresi.

Kesalahan dalam mengkategorikan dapat menyebabkan penggunaan fungsi loss yang tidak sesuai sehingga berdampak buruk pada performa model. Untuk binary classification, loss yang digunakan adalah logistic loss (binary cross entropy). Penggunaan loss yang salah – misalnya Mean Squared Error (MSE) untuk

data biner – dapat menyebabkan hasil yang buruk dan prediksi di luar dari rentang valid (0 – 1).

2.5.3 Algoritma XGBoost

Algoritma XGBoost bekerja melalui alur yang sistematis dan berulang untuk membangun sebuah model prediksi yang kuat (Chen & Guestrin, 2016; Dwinanda dkk., 2023; Mushava & Murray, 2024). Prosesnya diawali dengan sebuah prediksi awal yang kemudian disempurnakan secara bertahap dengan memperbaiki error dari tahap-tahap sebelumnya. Berikut merupakan tahapan detail dari alur kerja XGBoost beserta formula yang digunakan.

A. Inisialisasi Prediksi Awal (Log-Odds)

Proses pelatihan dimulai dengan membuat prediksi awal (base score) untuk semua data. Untuk masalah klasifikasi biner, prediksi awal ini umumnya dihitung sebagai nilai log-odds (juga disebut logit). Rumus yang digunakan dapat dilihat pada persamaan 1.

$$\hat{y}^{(0)} = \ln\left(\frac{\bar{y}}{1-\bar{y}}\right) \quad (1)$$

Keterangan:

$\hat{y}^{(0)}$: Prediksi awal dalam bentuk log-odds

$\bar{y}$: $Jumlah\ kasus\ positif\ /\ total\ data$

Nilai log-odds awal kemudian diubah menjadi probabilitas menggunakan fungsi sigmoid, seperti yang terlihat pada persamaan (2), yang memastikan hasilnya berada di antara 0 dan 1.

$$\hat{p} = \sigma(\hat{y}^{(t-1)}) = \frac{1}{1+e^{-\hat{y}^{(t-1)}}} \quad (2)$$

Keterangan:

$\hat{p}$: Probabilitas prediksi

e : Bilangan Euler (konstanta matematika, ~2.718)

$\hat{y}$: Nilai log-odds sebelumnya

B. Menghitung Residual (Gradien dan Hessian)

Langkah berikutnya adalah menghitung seberapa jauh prediksi menyimpang dari nilai actual atau nilai target sebenarnya. Untuk kasus klasifikasi biner digunakan perhitungan loss dengan loss logistic (Mushava & Murray, 2024), perhitungan tersebut memiliki formula sebagai berikut:

B.1. Gradien: Mengukur error dari prediksi. Formula matematika yang digunakan dapat dilihat pada persamaan (3).

$$g_i = \hat{p}_i - y_i \tag{3}$$

Keterangan:

g_i : Nilai gradien untuk data ke-i

p_i : Probabilitas prediksi untuk data ke-i (hasil persamaan 2)

y_i : Nilai aktual (label) untuk data ke-i

B.2. Hessian: mengukur seberapa cepat gradien berubah, digunakan untku optimasi. Rumus yang digunakan dapat dilihat pada persamaan (4).

$$h_i = \hat{p}_i(1 - \hat{p}_i) \tag{4}$$

Keterangan:

h_i : Nilai hessian untuk data ke-i

$\hat{p}_i$: Probabilitas prediksi untuk data ke-i

C. Membangun Pohon Keputusan

Sebuah pohon keputusan dibangun untuk memprediksi residual. Langkah kuncinya adalah menemukan split terbaik untuk data. Hal ini ditentukan dengan menghitung Gain untuk setiap kemungkinan split. Split dengan Gain tertinggi akan dipilih. Formula untuk Gain dapat dilihat pada persamaan (5).

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_R + H_L + \lambda}\right] - \gamma \tag{5}$$

Keterangan:

G_R, G_R: Jumlah gradien dan hessian untuk data di cabang kanan setelah split

G_L, G_L : Jumlah gradien dan hessian untuk data di cabang kiri setelah split

λ (lambda) : Parameter regularisasi (L2) untuk mengontrol kompleksitas model

γ (gamma) : Parameter regularisasi yang juga berfungsi untuk mengontrol kompleksitas dengan memberikan penalti pada jumlah daun.

D. Menghitung Nilai Daun

Setelah struktur pohon ditentukan, nilai prediksi (*leaf value* atau *weight*) dihitung untuk setiap daun (*leaf*) dari pohon tersebut. Nilai ini digunakan untuk memperbarui prediksi model secara keseluruhan. Formula untuk *leaf value* seperti pada persamaan (6).

$$\omega_j = -\frac{G_j}{H_j + \lambda} \tag{6}$$

Keterangan:

ω_j : Nilai prediksi pada sebuah daun (*leaf*) di pohon keputusan

G_j : Jumlah total gradien dari semua data yang ada di daun tersebut

H_j : Jumlah total hessian dari semua data yang ada di daun tersebut

λ : Parameter regularisasi (L2)

E. Memperbarui Prediksi

Prediksi log-odds dari iterasi sebelumnya diperbarui dengan menambahkan prediksi dari pohon yang baru (ω_j), yang diskalakan dengan learning rate (η) Persamaannya dapat dilihat pada persamaan (7).

$$\hat{y}^{(t)} = \hat{y}^{(t-1)} + \eta \times \omega_j \tag{7}$$

Keterangan:

$\hat{y}^{(t)}$: Nilai prediksi log-odds yang telah diperbarui pada iterasi ke-t

$\hat{y}^{(t-1)}$: Nilai log-odds dari iterasi sebelumnya

η : *Learning rate*, parameter yang mengontrol laju pembelajaran

ω_j : Nilai prediksi yang dihasilkan oleh pohon baru

F. Iterasi

Langkah B hingga E diulangi sebanyak jumlah pohon yang ditentukan (n_estimators). Dalam setiap iterasi, sebuah pohon baru dibangun untuk memperbaiki residual dari gabungan prediksi semua pohon sebelumnya. Proses ini berlanjut hingga kriteria berhenti terpenuhi, yang pada akhirnya menghasilkan model prediksi yang sangat akurat dan kuat. Setelah semua iterasi selesai, skor akhir log-odds $\hat{y}^{(t)}$ dikonversi menjadi probabilitas (nilai antara 0 dan 1) menggunakan fungsi sigmoid (persamaan 2).

2.5.4 Penjelasan Parameter Penting dalam XGBoost

Dalam implementasinya, algoritma XGBoost digunakan dengan memanggil kelas XGBClassifier dari library xgboost di Python. Model ini dapat dikustomisasi melalui berbagai parameter, seperti pada Tabel 2.2.

Tabel 2.2 Penjelasan Parameter XGBoost

Parameter	Penjelasan
learning_rate	Parameter ini mengatur seberapa besar kontribusi setiap pohon baru dalam mempelajari kesalahan sebelumnya. Nilai yang kecil akan membuat model belajar secara perlahan namun lebih stabil, sedangkan nilai besar dapat mempercepat pembelajaran namun berisiko overfitting.
max_depth	Menentukan kedalaman maksimum setiap pohon. Semakin besar nilainya, semakin kompleks pohon yang dibentuk. Namun, pohon yang terlalu dalam dapat menyebabkan model menangkap noise dari data pelatihan.
n_estimators	Mengatur jumlah pohon yang akan dibangun secara bertahap. Semakin banyak pohon dapat meningkatkan akurasi, tetapi juga meningkatkan waktu komputasi dan risiko overfitting jika tidak disertai regularisasi.
subsample	Menentukan proporsi sampel data pelatihan yang digunakan untuk membangun setiap pohon. Nilai di bawah 1.0 menyebabkan model melakukan subsampling, yang dapat membantu mengurangi overfitting dan meningkatkan generalisasi.
gamma	Menetapkan ambang minimum untuk pengurangan loss yang diperlukan agar pemisahan (split) pada simpul (node) dilakukan. Semakin besar nilai gamma, semakin konservatif model dalam membuat split, sehingga dapat berfungsi sebagai bentuk regularisasi tambahan.
reg_lambda	Parameter regularisasi L2 (ridge). Menambahkan penalti terhadap besarnya nilai parameter untuk mencegah overfitting. Nilai yang lebih besar akan mendorong model agar lebih sederhana.
reg_alpha	Parameter regularisasi L1 (lasso). Mendorong sparsity atau pengurangan kompleksitas model dengan mengurangi beberapa bobot fitur menjadi nol. Cocok digunakan ketika diduga banyak fitur tidak relevan.
objective	Mendefinisikan tujuan dari model. Untuk kasus klasifikasi biner seperti prediksi turnover, umumnya digunakan binary:logistic, yang menghasilkan output berupa probabilitas kelas.

Parameter	Penjelasan
eval_metric	Menentukan metrik yang digunakan untuk mengevaluasi performa model selama pelatihan. Salah satu metrik umum adalah logloss, yang mengukur seberapa baik prediksi probabilistik sesuai dengan label sebenarnya.
seed / random_state	Memberikan titik awal acak yang tetap agar hasil pelatihan model dapat direproduksi secara konsisten.

Penggunaan parameter-parameter ini sangat memengaruhi perilaku dan hasil model. Oleh karena itu, pemilihan nilai-nilainya perlu disesuaikan dengan karakteristik data dan tujuan pemodelan.

2.6. K-Fold Cross Validation

K-Fold Cross Validation merupakan sebuah metode statistik untuk mengevaluasi performa model prediktif dengan membagi dataset menjadi k subset (fold). Setiap subset digunakan secara bergantian sebagai data validasi, sementara sisanya digunakan sebagai data latihan. Menurut Seki dkk. (2024), metode ini memiliki keunggulan signifikan karena menghasilkan evaluasi model yang lebih stabil dan representatif, terutama model *machine learning* seperti XGBoost dan ANN. Terdapat beberapa tujuan dari penggunaan metode ini, seperti mengukur sejauh mana performa model terhadap data yang belum pernah dilihat, mengurangi variansi evaluasi, menstabilkan hasil evaluasi melalui rata-rata dari k eksperimen, serta menjaga efisiensi penggunaan data (terutama ketika dataset terbatas). Secara umum, K-Fold Cross Validation menerapkan 5 fold. Dalam studinya, Fuglkjær dkk. (2024) menggunakan 5 fold pada Stratified K-Fold Cross Validation dan berhasil memperoleh performa model yang sangat baik. K-Fold Cross Validation terbagi menjadi beberapa jenis, seperti.

1. Standard K-Fold

Metode ini membagi dataset secara acak menjadi k fold yang sama besar. Setiap fold akan digunakan sebagai data uji satu kali dan sisanya sebagai data latih. Cocok untuk dataset besar dan seimbang.

2. Stratified K-Fold

Metode yang membagi data dengan mempertahankan distribusi proporsional dari setiap kelas pada setiap fold. Idealnya untuk klasifikasi data yang imbalance atau tidak seimbang.

3. Repeated K-Fold

Melakukan k-fold secara berulang (dengan pembagian data berbeda setiap pengulangan), lalu hasilnya dirata-rata untuk stabilitas performa.

4. Leave-One-Out (LOOCV)

Kasus khusus k-fold dimana k = n (jumlah data). Setiap fold hanya berisi satu data untuk pengujian. Umumnya digunakan untuk data yang sangat kecil.

5. Group K-Fold

Digunakan ketika data memiliki struktur grup (misalnya data dari satu pasien atau user), dan penting agar grup tidak terbagi ke dalam training dan testing secara bersamaan.

6. Time Series K-Fold

Fold dibagi dengan mempertimbangkan urutan waktu. Fold yang lebih baru tidak boleh digunakan untuk pelatihan model yang mengevaluasi data masa lalu. Digunakan untuk forecasting.

Dalam penelitian ini, pemilihan metode validasi model menjadi krusial karena karakteristik dataset yang digunakan bersifat imbalance (label dataset tidak seimbang). Artinya, jumlah data pada satu kelas jauh lebih banyak dibandingkan kelas lainnya. Penggunaan metode Standard K-Fold pada dataset sangat berisiko, karena pembagian acak dapat menghasilkan *fold* (subset data) yang tidak memiliki sampel dari kelas minoritas sama sekali. Hal ini akan menyebabkan proses training menjadi tidak representatif dan hasil evaluasi performa model menjadi sangat bias.

Oleh karena itu, Stratified K-Fold Cross Validation dipilih sebagai metode evaluasi. Metode ini dapat memastikan bahwa setiap *fold* memiliki representasi proporsional dari setiap kelas, sesuai dengan distribusi pada dataset keseluruhan. Dengan demikian, setiap model yang dilatih dan diuji dalam *k* iterasi akan terekspos pada distribusi kelas yang sama, sehingga evaluasi performa yang dihasilkan akan lebih akurat, stabil, dan valid untuk mengukur kemampuan generalisasi model pada data yang tidak seimbang.

2.7. Confusion Matrix

Guna menampilkan performa atau kinerja dari sebuah model klasifikasi dibutuhkan suatu metode, yaitu Confusion matrix (Sanchhaya Education Private

Limited, 2025). Confusion Matrix menyajikan hasil prediksi model dalam bentuk empat kategori, yaitu:

1. *true positive* (TP): kondisi ketika prediksi dan data aktual positif
2. *true negative* (TN): kondisi dimana prediksi dan data aktual negatif
3. *false positive* (FP): kondisi ketika prediksi positif dan data aktual negatif
4. *false negative* (FN): kondisi ketika prediksi negatif dan data aktual positif

Dengan keempat kategori tersebut, dapat digunakan untuk menghitung metrik lanjutan, seperti akurasi, presisi, recall, serta F1 score. Berikut merupakan penjelasan mengenai masing – masing metrik.

2.7.1 Akurasi

Akurasi dari biner klasifikasi dapat dirumuskan seperti pada persamaan 8:

$$Akurasi = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

Akurasi merupakan sebuah cara untuk mengukur prediksi benar dibandingkan dengan seluruh data yang diuji. Akurasi sendiri tidak cukup untuk merepresentasikan bahwa model bekerja dengan baik terutama ketika dataset tidak seimbang. Mehan (2025) menyatakan bahwa akurasi tinggi bisa saja menyesatkan karena model dapat mencapai nilai tinggi hanya dengan memprediksi mayoritas kelas.

2.7.2 Presisi

$$Presisi = \frac{TP}{TP + FP} \tag{9}$$

Presisi merupakan metode yang berfokus pada prediksi positif. Presisi penting dalam konteks manajemen risiko, seperti mendeteksi karyawan resign. FP yang tinggi dapat menyebabkan pengambilan keputusan yang keliru. Menurut Bibers & Abdallah (2025), presisi sangat penting saat prediksi positif memiliki konsekuensi atau dampak besar.

2.7.3 Recall

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

Recall menunjukkan seberapa baik model untuk mendeteksi semua kasus positif. Recall menjadi penting ketika kesalahan kelalaian (FN: gagal mendeteksi karyawan yang akan turnover) harus diminimalkan, seperti pada kasus turnover karyawan produktif. Liu dkk. (2025) dan Yaragunda dkk. (2025) mencatat bahwa recall tinggi penting dalam konteks perencanaan retensi SDM.

2.7.4 F1 score

$$F1\ Score = 2\ \times \frac{Presisi \times Recall}{Presisi + Recall} \quad (11)$$

F1 score adalah rata – rata harmonis dari presisi dan recall. F1 score digunakan ketika keseimbangan antara FP (keliru memprediksi karyawan bertahan) dan FN (keliru memprediksi karyawan resign) sangat krusial. Dalam data yang tidak seimbang, F1 score memberikan gambaran yang lebih representatif daripada akurasi. Polat dkk. (2025) dan Singh dkk. (2025) menunjukkan bahwa F1 score adalah metrik utama dalam skenario klasifikasi yang imbalance atau tidak seimbang.

Oleh karena dalam penelitian ini data bersifat imbalance atau tidak seimbang antara karyawan yang turnover dan bertahan maka fokus utama evaluasi adalah recall dan F1 score.

2.8. Dashboard

Dashboard merupakan sebuah media untuk menampilkan data dengan basis teknologi informasi. Dashboard menampilkan data – data penting secara *real-time* dalam bentuk grafis atau numerik/angka untuk mendukung pemantauan dan analisis (Saragih dkk., 2021). Dashboard selain digunakan untuk alat pemantauan, juga digunakan sebagai alat untuk mendukung keputusan. Terdapat beberapa fungsi utama dashboard menurut Gallagher dkk. (2025), seperti melacak status operasional

secara *real-time,* menilai pencapaian KPI dan SLA, mendukung strategi organisasi berbasis analitik, dan menyediakan akses terhadap data performa organisasi. Dashboard dapat diklasifikasikan menjadi beberapa tipe, seperti (Patria, 2024; Piras dkk., 2025):

1. Dashboard strategis

Dashboard yang digunakan untuk menyajikan indikator kinerja utama (KPI) bagi manajemen tingkat atas. Dashboard dirancang untuk C-level eksekutif dan pengambilan keputusan tingkat tinggi. Bertugas untuk menyajikan data yang dapat membantu dalam mengevaluasi strategi, kinerja organisasi, dan arah bisnis secara keseluruhan.

2. Dashboard operasional

Dashboard ini digunakan untuk memantau aktivitas operasional harian secara *real-time*. Umumnya, dashboard ini digunakan oleh manajer lini dan supervisor untuk menangani proses bisnis yang sedang berlangsung.

3. Dashboard analitis

Dashboard yang mendukung analisis mendalam terhadap data historis dan tren. Dashboard tersebut sering kali digunakan oleh analis dan manajer menengah. Dashboard analitis merupakan jenis dashboard yang dinilai paling relevan untuk digunakan pada penelitian ini. Hal ini dikarenakan dashboard jenis ini mendukung pengguna untuk dapat melakukan analisis terhadap data historis dan tren, terutama data karyawan di penelitian ini.

4. Dashboard taktis

Dashboard yang menghubungkan antara tujuan dari strategi dan implementasi operasional. Dashboard tersebut sering kali digunakan untuk manajer departemen untuk mengawasi proyek jangka menengah, campaign pemasaran, ataupun efisiensi anggaran.

2.9. Feature Importance

Dalam pengembangan model machine learning yang kompleks seperti Extreme Gradient Boosting (XGBoost), salah satu tantangan terbesar adalah masalah "kotak hitam" (black box). Model mampu memberikan prediksi dengan akurasi tinggi, namun proses pengambilan keputusannya sering kali tidak

transparan. Padahal, dalam kasus bisnis yang krusial seperti prediksi turnover karyawan, pemahaman tentang "mengapa" seorang karyawan diprediksi akan resign sama pentingnya dengan prediksi itu sendiri. Di sinilah interpretabilitas model melalui feature importance memegang peranan vital.

Feature importance adalah serangkaian teknik yang digunakan untuk mengukur dan mengurutkan seberapa besar kontribusi setiap fitur (variabel) terhadap hasil prediksi sebuah model. Dengan kata lain, teknik ini membantu kita mengidentifikasi faktor-faktor mana yang paling berpengaruh dalam pengambilan keputusan model.

2.9.1 Metode Pengukuran Feature Importance pada XGBoost

Pada algoritma XGBoost, yang membangun modelnya melalui serangkaian decision tree secara bertahap, terdapat beberapa metrik untuk mengukur feature importance. Tiga yang paling umum digunakan adalah:

1. Weight

Metrik ini adalah yang paling sederhana, yaitu hanya dengan menghitung frekuensi kemunculan sebuah fitur di seluruh decision tree yang dibangun. Meskipun mudah dihitung, metrik ini memiliki kelemahan signifikan, terutama pada data yang tidak seimbang.

2. Cover

Metrik ini bekerja dengan mengevaluasi jangkauan atau cakupan sebuah fitur dengan menghitung rata-rata jumlah observasi (data) yang dipengaruhi setiap kali fitur tersebut digunakan untuk membagi pohon (Wang dkk., 2024).

3. Gain

Dianggap sebagai metrik yang paling informatif, Gain mengukur kontribusi nyata sebuah fitur terhadap performa model. Secara spesifik, metrik ini menghitung rata-rata peningkatan akurasi atau penurunan error yang terjadi setiap kali sebuah fitur digunakan untuk memecah simpul (node) pada decision tree (Li dkk., 2025). Secara umum, rumus Feature Importance dengan metrik Gain dituliskan pada persamaan 12 .

$$Importance_{Gain}(f) = \sum_{s \in S_f} Gain_s \qquad (12)$$

Atau dalam bentuk penulisan ringkas seperti pada persamaan 13 .

$$Importance(Fitur\ x) = \sum Gain\ (split\ fitur\ x) \quad (13)$$

Keterangan:

1. S_f = himpunan semua split dalam semua pohon di mana fitur f digunakan
2. $Gain_s$ = nilai peningkatan fungsi objektif akibat split s

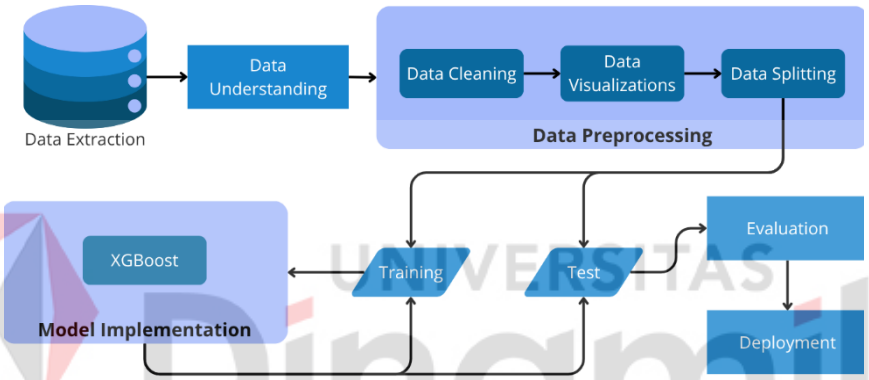
2.9.2 Pemilihan Metrik

Pemilihan metrik feature importance menjadi sangat krusial ketika dihadapkan pada masalah ketidakseimbangan kelas (class imbalance). Hal tersebut dikarenakan model dapat cenderung bias terhadap mayoritas kelas (Yenurkar dkk., 2023). Dalam skenario ini, metrik Weight berpotensi menyesatkan. Hal ini dikarenakan fitur bisa saja memiliki skor 'Weight' yang tinggi hanya karena sering muncul pada data mayoritas sehingga fitur tersebut mungkin saja tidak relevan untuk data minoritas (Meddage dkk., 2024). Oleh karena itu, metrik Gain menjadi pilihan yang jauh lebih unggul. Karena Gain berfokus pada peningkatan performa prediksi, ia mampu mengidentifikasi fitur yang secara efektif membantu model membedakan antara kelas mayoritas dan minoritas (Duckworth dkk., 2021). Beberapa studi menegaskan bahwa Gain memberikan insight yang lebih kaya dan akurat karena berhubungan langsung dengan kekuatan prediksi model (Alsahaf dkk., 2022; Azeem & Dev, 2024).

Dengan menggunakan Gain, penelitian ini tidak hanya bertujuan untuk membangun model prediksi turnover yang akurat, tetapi juga untuk melakukan seleksi fitur yang dapat diinterpretasikan (interpretable feature selection). Hasilnya dapat memberikan petunjuk lebih lanjut bagi tim HR mengenai variabel-variabel apa saja yang menjadi pendorong utama turnover di organisasi (Z. Liu dkk., 2021; Se dkk., 2025).

BAB III
METODOLOGI PENELITIAN

Arsitektur penelitian yang diilustrasikan pada Gambar 3.1 diadopsi dari penelitian Isha dkk. (2024) dan disesuaikan dengan kebutuhan studi ini. Setiap tahap, mulai dari ekstraksi data hingga deployment, dirancang untuk memastikan proses pengembangan model berjalan secara sistematis dan terstruktur. Pendekatan ini memungkinkan setiap langkah, mulai dari pembersihan data hingga evaluasi, dapat divalidasi dan dipertanggungjawabkan secara ilmiah, sehingga model prediksi yang dihasilkan memiliki landasan yang kuat dan dapat diandalkan.



Gambar 3.1 Metodologi Penelitian
(Sumber: Isha dkk., 2024)

3.1. Data Extraction

Tahap pertama dalam penelitian ini adalah pengumpulan data karyawan yang akan digunakan sebagai data set. Data bersumber dari Badan Kepegawaian di Universitas Dinamika. Dalam prosesnya, terdapat beberapa *feature* atau atribut yang diajukan kepada Badan Kepegawaian, seperti pada Tabel 3.1. Pengajuan atribut-atribut ini didasarkan pada studi literatur dan penelitian terdahulu yang mengidentifikasi faktor-faktor seperti demografi, masa kerja, dan riwayat kehadiran sebagai variabel yang berpotensi memengaruhi keputusan turnover karyawan. Data yang diminta mencakup periode tiga tahun terakhir untuk memastikan dataset memiliki volume dan variasi historis yang cukup untuk melatih model secara efektif. Data mentah ini menjadi input fundamental yang akan diolah pada tahap-tahap berikutnya.

Tabel 3.1 Feature – feature yang diajukan

Feature	Tipe Data	Penjelasan
Gaji	Kategorial	Kisaran upah bulanan karyawan, dibagi ke dalam beberapa rentang kategori.
Masa kerja	Integer	Lama waktu karyawan telah bekerja (Dalam tahun)
Ketidakhadiran	Integer	Jumlah hari atau frekuensi ketidakhadiran tanpa keterangan atau dengan keterangan
Penilaian kinerja	Integer	Hasil evaluasi kinerja karyawan
Jabatan	Kategorial (Integer)	Posisi atau peran karyawan di dalam organisasi
Usia	Integer	Umur karyawan saat data diambil
Jenis kelamin	Kategorial (integer)	Jenis kelamin karyawan
Status pernikahan	Kategorial (integer)	Status pernikahan karyawan
Alamat tempat tinggal	Teks	Alamat tempat tinggal karyawan

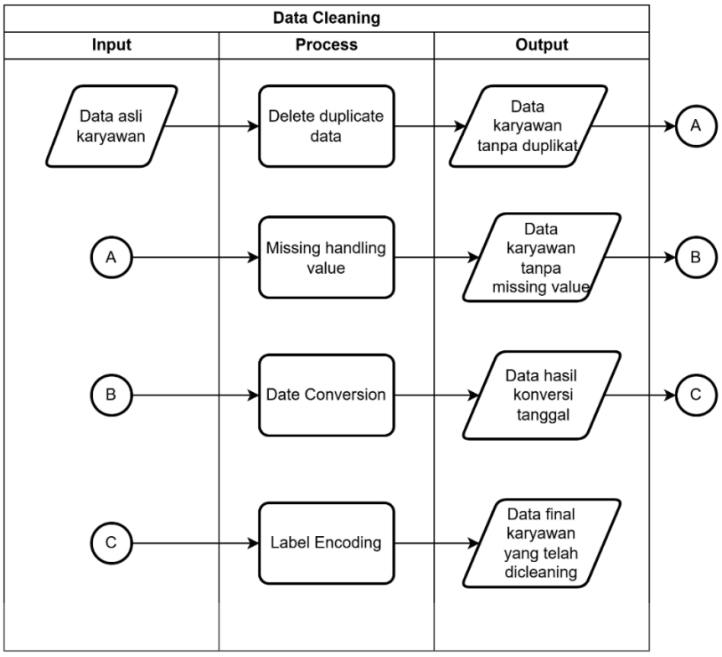
3.2. Data Understanding

Setelah data didapatkan, proses dilanjutkan dengan melakukan eksplorasi awal untuk memahami data yang akan digunakan untuk penelitian. Langkah – langkah dalam tahap ini dimulai dari mengecek jumlah baris dan kolom dari data set dan mengidentifikasi kolom dan tipe data yang terdapat pada data dan akan digunakan nantinya. Tahapan ini krusial untuk membentuk fondasi analisis. Dengan mengidentifikasi tipe data setiap kolom, maka dapat direncanakan teknik preprocessing yang tepat. Misalnya, kolom – kolom yang dengan tipe data tanggal akan memerlukan konversi khusus, sementara kolom kategorikal perlu diubah menjadi format numerik. Analisis awal ini juga membantu dalam mendeteksi anomali atau potensi masalah sejak dini, seperti adanya nilai yang hilang (missing values) atau data yang tidak konsisten, yang akan ditangani pada tahap data preprocessing berikutnya.

3.3. Data Preprocessing

Data yang telah dikumpulkan akan melalui proses preprocessing agar siap digunakan dalam pemodelan. Tahapan ini meliputi:

3.3.1 Data Cleaning



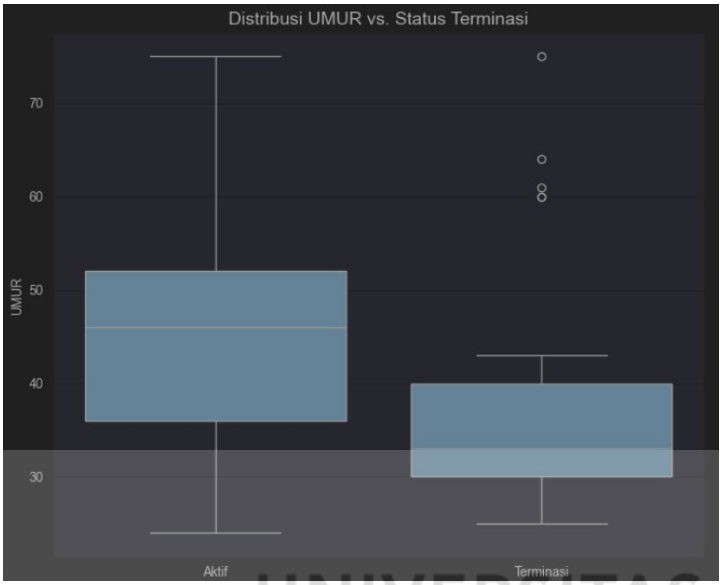
Gambar 3.2 Diagram IPO Data Cleaning

Data cleaning adalah salah satu tahapan yang bertujuan untuk membersihkan data yang akan digunakan untuk pemodelan XGBoost. Gambar 3.2 merupakan diagram yang menggambarkan bagaimana proses di tahapan data cleaning berjalan. Berikut merupakan penjelasan tentang proses – proses yang berjalan dalam tahapan ini.

1. Delete duplicate data: bertujuan menghapus data duplikat untuk mencegah bias dalam model. Fungsi yang umumnya digunakan yaitu drop_duplicates() pada library pandas.
2. Missing Value Handling: proses ini bertujuan untuk mengatasi missing values agar dataset lengkap dan siap diproses oleh algoritma machine learning. Terdapat beberapa metode yang dapat digunakan pada tahap ini, antara lain drop, imputation, dan interpolation.
3. Date Conversion: proses ini berjalan dengan mengubah kolom tanggal menjadi format numerik yang nantinya dapat diproses oleh algoritma XGBoost.
4. Label Encoding: mengubah kolom kategorial menjadi numerik agar dapat diproses oleh algoritma machine learning. Proses ini berjalan dengan mengubah

setiap kategori yang unik menjadi angka integer (misalnya, "Laki-laki" menjadi 0, "Perempuan" menjadi 1).

3.3.2 Data Visualizations



Gambar 3.3 Contoh Grafik Boxplot

Visualisasi data memegang peran penting untuk memahami pola, hubungan, dan distribusi antar fitur dalam data. Pada penelitian ini, salah satu teknik visualisasi yang digunakan adalah boxplot. Boxplot dipilih karena kemampuannya yang efektif dalam menyajikan ringkasan statistik dari distribusi data numerik, seperti persebaran, nilai tengah (median), dan pencilan (outlier) secara visual.

Representasi visual dari boxplot memungkinkan analisis perbandingan yang jelas antara kelompok data yang berbeda. Misalnya, dengan membandingkan boxplot fitur-fitur seperti gaji, usia, atau lama kerja antara kelompok karyawan yang *turnover* dan yang tidak, dapat dengan cepat diidentifikasi apakah terdapat perbedaan distribusi yang signifikan. Penggunaan boxplot sangat intuitif untuk menyoroti perbedaan median, rentang interkuartil (IQR), dan variabilitas data secara keseluruhan, yang menjadikannya alat yang sangat berguna dalam tahap eksplorasi data untuk hipotesis awal mengenai faktor-faktor yang mungkin mempengaruhi *turnover* karyawan.

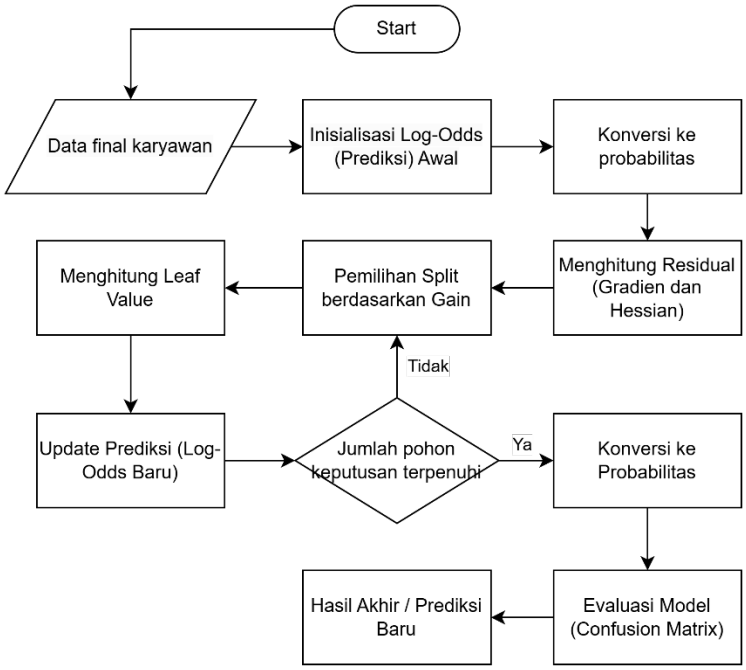
3.3.3 Data Splitting

Pada penelitian ini, pembagian data dilakukan menggunakan metode K-Fold Cross Validation dengan jumlah 5 fold, di mana data dibagi menjadi data training sebesar 80% dan data testing sebesar 20% dari total keseluruhan data. Penerapan K-Fold Cross Validation bertujuan untuk meningkatkan stabilitas dan generalisasi model dengan memastikan bahwa setiap subset data digunakan baik untuk pelatihan maupun validasi secara bergantian.

Pemilihan 5 fold dengan perbandingan 80:20 dipertimbangkan karena merupakan praktik umum yang memberikan keseimbangan antara keseimbangan antara bias dan varians, memastikan estimasi performa model yang lebih stabil dan dapat diandalkan. Selain itu, menurut Geron (2019) dalam bukunya Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow , jika jumlah data sangat besar, perbandingan data training dan testing dapat diturunkan (misalnya, 90:10 atau lebih), karena subset data yang lebih kecil sudah cukup untuk merepresentasikan distribusi keseluruhan data. Namun, pada penelitian ini, jumlah data tidak terlalu besar sehingga perbandingan 80:20 tetap relevan untuk memastikan model memiliki cukup data untuk pelatihan tanpa mengorbankan kemampuan evaluasi yang akurat.

3.4. Model Implementation

Pada tahap ini, dilakukan pembangunan model menggunakan algoritma XGBoost (Extreme Gradient Boosting), dengan mengimplementasikan K-Fold Cross Validation sebanyak 5 fold guna memastikan bahwa model memiliki performa yang stabil dan generalisasi yang baik. Alur proses implementasi model ini dapat dilihat pada flowchart Gambar 3.4.



Gambar 3.4 Flowchart Model Implementation XGBoost

Proses dimulai dengan data final karyawan yang telah melalui tahap data cleaning dan preprocessing, digunakan sebagai input awal. Tahap selanjutnya adalah inisialisasi prediksi awal dalam bentuk log-odds (logit), yang menghasilkan prediksi awal berbasis distribusi label target.

Prediksi awal ini menjadi dasar dalam perhitungan residual, yang mencakup nilai gradien dan hessian. Gradien menggambarkan selisih antara label aktual dan prediksi, sedangkan hessian menunjukkan perubahan gradien terhadap prediksi. Nilai residual ini menjadi dasar untuk proses boosting.

Kemudian, sistem membangun sebuah pohon keputusan (decision tree) dengan melakukan pemilihan split terbaik berdasarkan nilai gain. Setiap split bertujuan meminimalkan loss function secara optimal. Setelah struktur pohon terbentuk, sistem menghitung nilai setiap daun (leaf value) berdasarkan agregasi gradien dan hessian di masing-masing leaf.

Selanjutnya, prediksi diperbarui dengan menambahkan output dari pohon baru ke prediksi sebelumnya. Proses dari perhitungan residual hingga pembaruan prediksi ini diulangi secara iteratif, di mana setiap iterasi akan menghasilkan pohon baru yang bertujuan untuk memperbaiki kesalahan (residual) dari keseluruhan prediksi sebelumnya.

Iterasi ini akan berhenti ketika salah satu dari kriteria berikut terpenuhi: jumlah maksimum pohon tercapai. Setelah iterasi selesai, prediksi akhir dalam bentuk log-odds dikonversi ke bentuk probabilitas. Probabilitas inilah yang digunakan dalam tahap evaluasi model, yang dilakukan menggunakan metrik seperti confusion matrix untuk menilai kualitas klasifikasi model terhadap data latih.

3.5. Evaluation

Setelah model dibangun, proses evaluasi dilakukan untuk mengukur kinerja model. Mengingat penelitian ini mengimplementasikan 5-Fold Cross Validation, proses evaluasi dilakukan secara berulang untuk mendapatkan hasil yang lebih stabil dan andal.

Pada setiap lipatan (*fold*) validasi, kinerja model akan diukur menggunakan Confusion Matrix sebagai alat utama. Dari matriks ini, akan dihitung serangkaian metrik performa. Untuk memastikan hasil evaluasi tidak bias dan menunjukkan konsistensi model, keseluruhan hasil evaluasi akan ditampilkan dalam bentuk tabel, yang mencakup nilai rata-rata serta standar deviasi dari kelima *fold* tersebut. Nilai rata-rata akan menjadi skor kinerja akhir, sementara standar deviasi yang rendah akan mengindikasikan bahwa performa model stabil di berbagai segmen data.

Dari Confusion Matrix ini, beberapa metrik evaluasi dapat dihitung, di antaranya:

1. Akurasi: Proporsi total prediksi yang benar dari seluruh data.
2. Precision: Mengukur ketepatan prediksi terhadap kelas *turnover*.
3. Recall: Menunjukkan seberapa baik model mendeteksi seluruh kasus *turnover* yang sebenarnya.
4. F1-Score: Nilai harmonisasi antara *precision* dan *recall*, yang sangat berguna ketika data tidak seimbang.

3.6. Deployment

Model yang telah dilatih dengan algoritma XGBoost diimplementasikan ke dalam sebuah dashboard analitis, atau lebih tepatnya *predictive dashboard*. Dashboard tersebut akan berbasis website menggunakan framework Streamlit.

Pada dashboard akan disematkan fungsi untuk menggunggah file yang berisi data karyawan. File tersebut diunggah dalam format csv ataupun excel. Dashboard nantinya akan menghasilkan prediksi untuk setiap data karyawan yang diunggah dan menyediakan opsi download hasil prediksi dalam format csv untuk mendukung keputusan. Selain itu, untuk lebih mudah dalam memahami dan analisa, dashboard akan menyajikan beberapa grafik, seperti Pie Chart dan Bar Chart.

BAB IV
HASIL DAN PEMBAHASAN

4.1. Data Extraction

Tahapan awal dalam proses penelitian dimulai dengan pengajuan permintaan data kepada Badan Kepegawaian Universitas Dinamika. Tabel 4.1 merupakan atribut data yang diajukan kepada Badan Kepegawaian Universitas Dinamika. Setelah proses pengajuan dilakukan, tidak semua data dapat diberikan oleh pihak Badan Kepegawaian karena pertimbangan privasi dan keamanan data. Beberapa atribut seperti gaji dan penilaian kinerja dikecualikan dari data yang diberikan. Atribut data yang berhasil diperoleh dapat dilihat pada Tabel 4.1.

Tabel 4.1 Feature – feature yang diberikan

No.	Feature	Tipe Data	Keterangan
1.	NAMA	Teks	Inisial karyawan
2.	ALAMAT (Kel., Kec.)	Teks	Alamat tempat tinggal berupa nama kelurahan, kecamatan, dan kota
3.	KARY_TYPE	Kategorial	Jenis kepegawaian
4.	MULAI_KERJA	Tanggal / Date	Tanggal mulai bekerja di institusi
5.	TGL_KELUAR	Tanggal / Date (nullable)	Tanggal keluar dari pekerjaan (jika ada)
6.	UMUR	Integer	Usia karyawan pada saat data diambil
7.	JENIS_KELAMIN	Kategorial	Jenis kelamin (Laki-laki dan Perempuan)
8.	STS_NIKAH	Kategorial	Status pernikahan, seperti menikah, belum menikah, dan janda
9.	T	Integer	Tepat waktu selama periode 1 tahun
10.	LDI	Integer	Terlambat dengan ijin selama periode 1 tahun
11.	LTI	Integer	Terlambat dengan ijin selama periode 1 tahun
12.	I	Integer	Jumlah izin yang diambil karyawan selama periode 1 tahun
13.	S	Integer	Jumlah sakit selama periode 1 tahun
14.	A	Integer	Jumlah ketidakhadiran tanpa keterangan (alpha) selama periode 1 tahun
15.	D	Integer	Jumlah cuti dinas selama periode 1 tahun
16.	CD	Integer	Cuti diluar tanggungan (misalnya cuti hamil atau alasan pribadi tertentu) selama periode 1 tahun
17.	CP	Integer	Cuti pribadi selama periode 1 tahun
18.	CN	Integer	Cuti menikah selama periode 1 tahun
19.	CL	Integer	Cuti melahirkan selama periode 1 tahun
20.	CB	Integer	Cuti besar selama periode 1 tahun
21.	CSJ	Integer	Cuti sakit selama periode 1 tahun

Data yang diperoleh merupakan data internal yang mencakup informasi karyawan dari tahun 2022 hingga 2024. Untuk memberikan gambaran lebih jelas

mengenai struktur data yang digunakan, Tabel 4.2 berikut menampilkan cuplikan sampel dari data karyawan yang telah diperoleh.

Tabel 4.2 Cuplikan Data Karyawan Tahun 2022 – 2024

KARY_TYPE	STATUS	MULAI_KERJA	TGL_KELUAR	UMUR	JENIS_KELAMIN
TD	M	11/19/2007 00:00:00	08/31/2022 00:00:00	43	Pria
TD	M	09/01/2018 00:00:00	04/30/2022 00:00:00	33	Wanita
T	M	05/01/2022 00:00:00	12/23/2022 00:00:00	30	Wanita

4.2. Data Understanding

Dataset ini terdiri dari 205 baris dan 21 kolom awal. Setelah dilakukan analisis awal, diketahui bahwa sekitar 15% dari data merupakan karyawan yang telah mengundurkan diri, yang menunjukkan adanya sedikit ketidakseimbangan kelas (imbalanced dataset). Terdapat juga satu nilai yang hilang (*missing value*) pada kolom UMUR yang kemudian ditangani pada tahap *preprocessing*.

4.3. Data Preprocessing

Tahap data preprocessing merupakan langkah krusial untuk mempersiapkan dataset agar siap digunakan untuk pemodelan machine learning. Pada tahap ini, dilakukan serangkaian proses mulai dari data cleaning untuk menangani nilai yang hilang, rekayasa fitur (*feature engineering*) untuk menciptakan variabel baru yang lebih informatif, hingga encoding untuk mengubah data kategorikal menjadi format numerik, kemudian dilanjutkan *data visualizations*, dan *data splitting*.

4.3.1 Data Cleaning

Langkah pertama dalam *preprocessing* adalah menghapus data duplikat yang berpotensi menyebabkan bias dalam proses pelatihan model. Fungsi duplicated().sum() guna untuk mendeteksi data yang identic kemudian drop_duplicates() dari library pandas digunakan untuk menghapus entri data yang identik. Gambar 4.1 merupakan penerapan kode untuk mendeteksi data duplikat. Namun seperti yang terlihat, karena data duplikat tidak terdeteksi maka proses tidak akan dilanjutkan sampai drop_duplicated().

```
excel.duplicated().sum()
✓ [11] 58ms
np.int64(0)
```

Gambar 4.1 Kode untuk Deteksi Data Duplikat

Selanjutnya, dilakukan penanganan terhadap missing value, jika ditemukan. Untuk proses ini, telah disiapkan skenario imputasi menggunakan median atau nilai tengah. Median dipilih karena metode tersebut lebih tahan *outlier* atau pencilan dibandingkan metode lain seperti rata – rata atau *mean*. Gambar 4.2 merupakan penerapan kode untuk mendeteksi data – data yang kosong.

```
excel.isna().sum()
✓ [27] < 10 ms
```

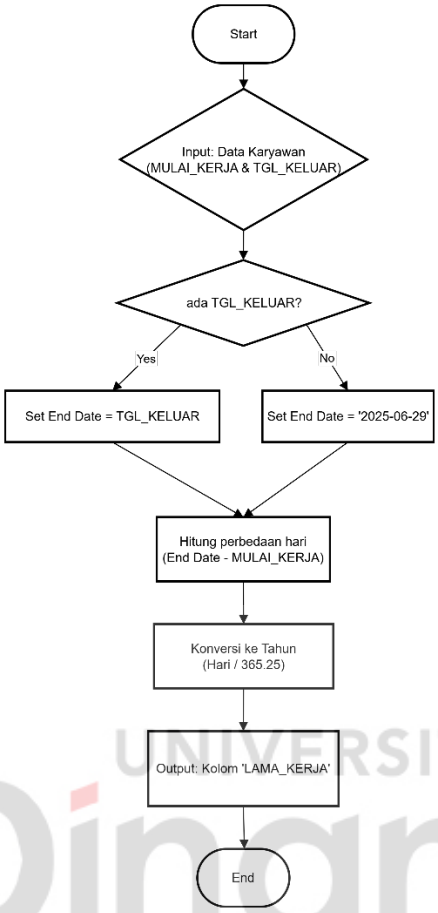
1-10 26 rows × 1 cols

	<unnamed>
NAMA	0
KARY_TYPE	0
STATUS	0
MULAI_KERJA	0
TGL_KELUAR	180
UMUR	1
JENIS_KELAMIN	0
STS_NIKAH	0
T	0
LDI	0

Gambar 4.2 Penerapan Kode untuk Deteksi Data Kosong

Kemudian proses *data cleaning* dilanjutkan ke proses imputasi dengan menggunakan median. Setelah data bersih dari nilai yang hilang, dilakukan rekayasa dan transformasi fitur untuk meningkatkan kualitas prediktor yang akan digunakan oleh model XGBoost.

A. Menghitung Lama Kerja



Gambar 4.3 Flowchart Lama Kerja

Fitur LAMA_KERJA dihitung secara dinamis untuk merepresentasikan masa bakti setiap karyawan dalam satuan tahun. Proses ini dimulai dengan merancang alur logika perhitungannya sebagaimana ditampilkan pada Gambar 4.3.

Flowchart tersebut menjelaskan langkah-langkah berikut: sistem pertama-tama memeriksa apakah atribut TGL_KELUAR tersedia untuk setiap karyawan. Jika ya, maka tanggal tersebut digunakan sebagai tanggal akhir masa kerja. Namun, jika TGL_KELUAR kosong (misalnya karyawan masih aktif), maka tanggal 29 Juli 2025 digunakan sebagai batas waktu (cut-off date) untuk menghitung durasi kerja. Setelah tanggal akhir ditentukan, sistem menghitung selisih hari antara tanggal mulai kerja (MULAI_KERJA) dan tanggal akhir, lalu mengkonversinya ke tahun dengan membagi hasil selisih hari tersebut dengan angka 365.25, yang memperhitungkan tahun kabisat.

```
def hitung_lama_kerja(row):
    end_date = row['TGL_KELUAR'] if pd.notna(row['TGL_KELUAR']) else today
    if pd.isna(row['MULAI_KERJA']):
        return 0
    return (end_date - row['MULAI_KERJA']).days / 365.25

excel['LAMA_KERJA'] = excel.apply(hitung_lama_kerja, axis=1).round(2)
excel[['MULAI_KERJA', 'TGL_KELUAR', 'LAMA_KERJA']].head(5)
✓ [71] 26ms
```

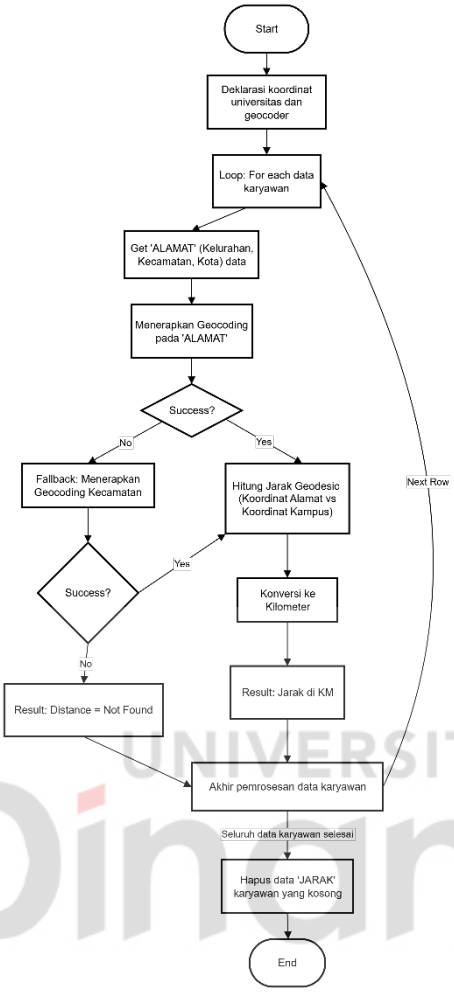
5 rows 5 rows × 3 cols Edit in Data Wrangler AI Quick Ch

	MULAI_KERJA	TGL_KELUAR	LAMA_KERJA
0	1996-02-01	NaT	29.41
1	1999-05-03	NaT	26.16
2	2014-03-03	NaT	11.33
3	1996-10-01	NaT	28.74
4	1989-12-01	NaT	35.58

Gambar 4.4 Kode untuk Hitung Lama Kerja

Implementasi dari logika tersebut dituangkan dalam bentuk kode Python seperti yang ditampilkan pada Gambar 4.4. Fungsi hitung_lama_kerja() menerima satu baris data, melakukan pengecekan kondisi TGL_KELUAR, dan menghitung durasi kerja berdasarkan alur dalam flowchart. Perhitungan ini diterapkan ke seluruh data menggunakan fungsi .apply() pada DataFrame, dan hasil akhirnya disimpan dalam kolom baru bernama LAMA_KERJA. Parameter axis=1 pada fungsi .apply() menginstruksikan pandas untuk menerapkan fungsi hitung_lama_kerja pada setiap baris data, bukan kolom. Ini penting karena perhitungan lama kerja memerlukan akses ke beberapa kolom sekaligus ('MULAI_KERJA' dan 'TGL_KELUAR') untuk setiap entri karyawan. Selanjutnya, fungsi .round(2) digunakan untuk membulatkan hasil perhitungan menjadi dua angka desimal. Langkah ini bertujuan untuk menjaga konsistensi format data dan memastikan fitur 'LAMA_KERJA' memiliki presisi yang seragam di seluruh dataset.

B. Hitung Jarak Tempat Tinggal



Gambar 4.5 Alur Hitung Jarak Tinggal

Fitur JARAK_TINGGAL dibuat dengan menghitung jarak garis lurus (geodesic distance) antara lokasi tempat tinggal karyawan dan Universitas Dinamika. Gambar 4.5 menunjukkan flowchart alur proses perhitungan ini sebagai bagian dari tahapan feature engineering. Proses dimulai dengan deklarasi koordinat kampus sebagai titik referensi, serta inisialisasi layanan geocoder menggunakan pustaka geopy dan layanan Nominatim OpenStreetMap.

Setiap data alamat karyawan diolah secara iteratif. Pertama-tama, sistem mencoba melakukan geocoding secara langsung pada data kolom ALAMAT (kelurahan, kecamatan, dan kota) untuk mendapatkan koordinat lintang dan bujur. Jika proses ini gagal, maka sistem melakukan fallback dengan mencoba menggeocode hanya berdasarkan nama kecamatan dan kota.

Jika proses geocoding berhasil, maka sistem menghitung jarak antara koordinat alamat dan koordinat kampus menggunakan metode geodesik dari geopy.distance.geodesic. Nilai ini kemudian dikonversi ke dalam satuan kilometer dan disimpan dalam kolom baru bernama JARAK_TINGGAL.

Namun, jika geocoding tetap gagal (baik dari alamat lengkap maupun fallback sub-district), maka nilai jarak dianggap tidak ditemukan dan diberi label khusus atau dihapus. Proses ini diakhiri dengan penyaringan untuk menghapus baris yang memiliki nilai JARAK_TINGGAL kosong atau tidak valid.

Selanjutnya, Gambar 4.6 menampilkan cuplikan hasil dari proses implementasi tersebut. Tabel menunjukkan data ALAMAT berserta JARAK_TINGGAL hasil perhitungan. Nilai yang dihasilkan bervariasi, tergantung pada lokasi geografis tempat tinggal karyawan, mulai dari <1 km (sangat dekat dengan kampus), hingga lebih dari 28 km untuk alamat yang berada di luar kota, seperti Gresik atau Sidoarjo.

```
excel[['ALAMAT','JARAK_TINGGAL']].head(10)
```

Executed at 2025.07.27 22:57:17 in 6ms

10 rows | 10 rows × 2 columns

	ALAMAT	JARAK_TINGGAL
0	Morokrembangan, Krembangan, Surabaya	11.68
1	Rangkah, Tambaksari, Surabaya	7.35
2	Gununganyar, Gununganyar, Surabaya	3.76
3	Wedoro Klurak, Candi, Sidoarjo	18.85
4	Ploso, Tambaksari, Surabaya	6.65
5	Penjaringansari, Rungkut, Surabaya	0.93
6	Yosowilangon, Manyar, Gresik	28.36
7	Manukan Tirto, Tandes, Surabaya	13.43
8	Kebraon, Karangpilang, Surabaya	11.15
9	Medokan Semampir, Sukolilo, Surabaya	0.59

Gambar 4.6 Hasil Hitung Jarak Tinggal

C. Encoding Fitur Kategorikal

Metode Label Encoding digunakan untuk mengubah nilai kategorikal pada kolom STS_NIKAH (status pernikahan) dan JENIS_KELAMIN (jenis kelamin) menjadi representasi numerik seperti pada Gambar 4.7. Pendekatan ini sesuai untuk fitur dengan jumlah kategori yang relatif sedikit dan tidak memiliki urutan logis

antar nilainya, sehingga tidak memerlukan pemetaan berbasis hierarki seperti pada ordinal encoding.

Hasil pemetaan dari teks ke angka dapat dilihat pada Gambar 4.8. Misalnya, status 'Belum Menikah' dipetakan ke 0, 'Janda' ke 1, dan 'Menikah' ke 2. Sementara itu, untuk jenis kelamin, 'Pria' dipetakan ke 0 dan 'Wanita' ke 1. Proses ini memungkinkan model machine learning seperti XGBoost menerima input numerik tanpa mengubah struktur kategorikal dasar dari data.

```
status_pernikahan_encoder = LabelEncoder()
sex_encoder = LabelEncoder()

excel['STS_NIKAH'] = status_pernikahan_encoder.fit_transform(excel['STS_NIKAH'])
excel['JENIS_KELAMIN'] = sex_encoder.fit_transform(excel['JENIS_KELAMIN'])
Executed at 2025.07.27 22:57:18 in 3ms

# Buat DataFrame untuk mapping Bahan Pangan
mapping_married = pd.DataFrame({
    "Label": range(len(status_pernikahan_encoder.classes_)),
    "Pernikahan": status_pernikahan_encoder.classes_
})

# Buat DataFrame untuk mapping Daerah
mapping_sex = pd.DataFrame({
    "Label": range(len(sex_encoder.classes_)),
    "Jenis Kelamin": sex_encoder.classes_
})

# Tampilkan dalam bentuk tabel
print("Mapping Pernikahan:")
print(mapping_married.to_string(index=False))  # Hapus index agar lebih rapi

print("\nMapping Jenis Kelamin:")
print(mapping_sex.to_string(index=False))
Executed at 2025.07.27 22:57:18 in 5ms

 Mapping Pernikahan:
  Label    Pernikahan
      0 Belum Menikah
      1         Janda
      2       Menikah

 Mapping Jenis Kelamin:
  Label Jenis Kelamin
      0          Pria
      1        Wanita
```

Gambar 4.7 Mapping Label Encoder

D. Seleksi dan Finalisasi Fitur

Setelah fitur-fitur baru dibuat dan data kategorikal di-encode, kolom-kolom asli yang tidak lagi diperlukan atau sudah direpresentasikan dalam bentuk lain

dihapus. Kolom yang dihapus meliputi: nama, mulai kerja, tanggal keluar, dan alamat. Proses ini menghasilkan dataset akhir yang bersih dan siap untuk tahap pemodelan, dengan struktur akhir seperti terlihat pada Gambar 4.8.

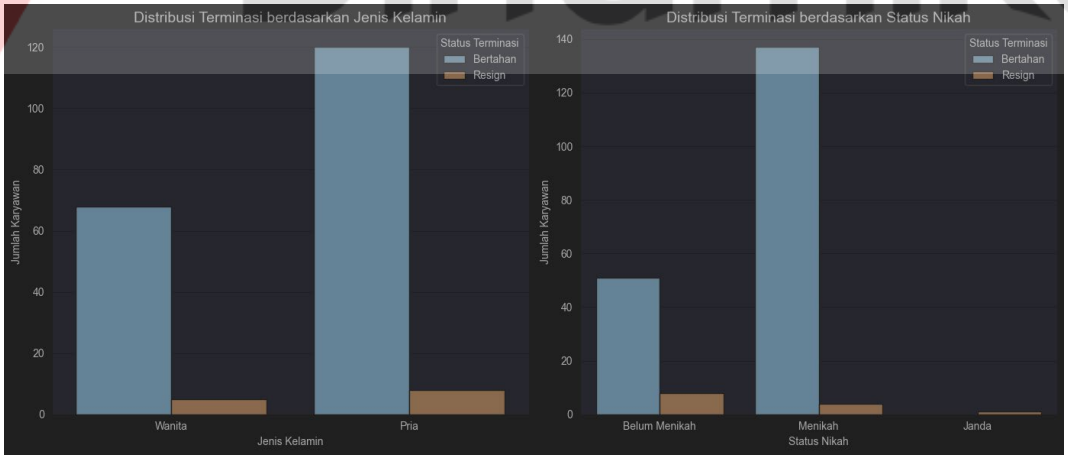
	UMUR	JENIS_KELAMIN	STS_NIKAH	TERMINATED	LAMA_KERJA
0	55	1	0	0	29
1	50	0	2	0	26
2	45	0	2	0	11
3	52	0	2	0	28
4	63	0	2	0	35

Gambar 4.8 Cuplikan Dataset setelah Data Cleaning

4.3.2 Data Visualizations

Visualisasi data dilakukan untuk memahami hubungan antar fitur dengan target prediksi turnover (TERMINATED), serta untuk mendeteksi pola-pola yang mungkin memengaruhi keputusan karyawan untuk keluar dari institusi. Beberapa metode visualisasi yang digunakan antara lain adalah countplot, boxplot, dan heatmap.

A. Visualisasi Fitur Kategorikal terhadap Turnover

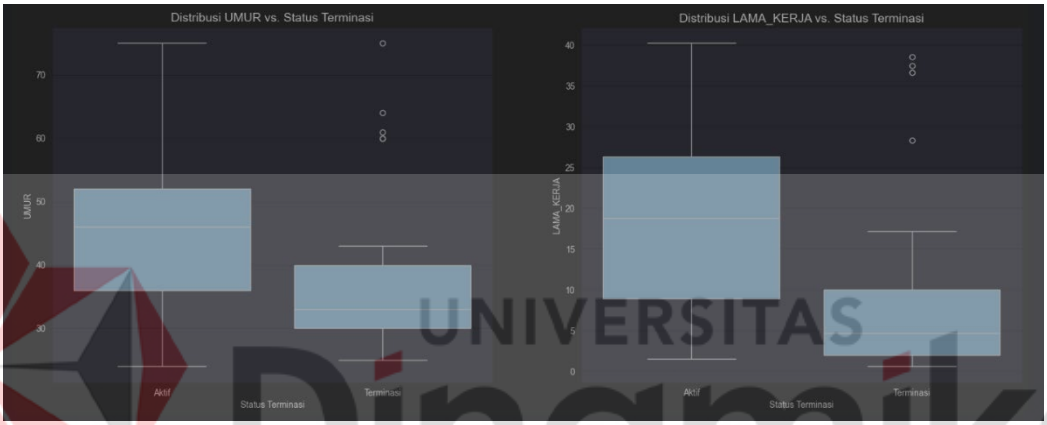


Gambar 4.9 Visualisasi Countplot Jenis Kelamin dan Status Nikah

Gambar 4.9 merupakan visualisasi countplot yang dilakukan untuk dua fitur kategorikal penting yaitu jenis kelamin (grafik sebelah kiri) dan status pernikahan (grafik sebelah kanan). Pada grafik status pernikahan (grafik sebelah kanan), terlihat bahwa karyawan dengan status menikah memiliki jumlah paling banyak,

yaitu kurang lebih 140 karyawan yang mana diikuti oleh karyawan belum menikah dan janda. Jika melihat distribusi karyawan resign pada grafik status pernikahan, dapat disimpulkan bahwa kecenderungan karyawan resign lebih tinggi pada karyawan dengan status belum menikah baru kemudian diikuti status menikah dan janda. Selanjutnya, pada grafik jenis kelamin (grafik sebelah kiri), diketahui pria mendominasi jumlah karyawan dengan 120 karyawan. Namun jika melihat distribusi karyawan resign, disimpulkan bahwa tidak ada perbedaan yang cukup signifikan pada distribusi turnover berdasarkan jenis kelamin.

B. Visualisasi Fitur Numerik terhadap Turnover



Gambar 4.10 Visualisasi boxplot

Gambar 4.10 menampilkan dua boxplot yang membandingkan distribusi nilai dua fitur numerik penting berdasarkan status terminasi karyawan, yaitu antara karyawan aktif dan yang telah berhenti. Grafik sebelah kiri menunjukkan distribusi umur (UMUR), sedangkan grafik sebelah kanan menampilkan distribusi lama masa kerja (LAMA_KERJA).

Sumbu vertikal (Y) masing-masing boxplot merepresentasikan skala nilai dari fitur yang dianalisis: umur dalam satuan tahun dan lama kerja dalam satuan tahun masa pengabdian. Bagian kotak dari boxplot mewakili rentang antar-kuartil (interquartile range), yaitu dari kuartil pertama (Q1) hingga kuartil ketiga (Q3), dengan garis horizontal di tengahnya sebagai penanda nilai median. Sementara itu, garis whiskers di bagian atas dan bawah menunjukkan rentang data yang tidak tergolong pencilan, sedangkan titik-titik di luar whiskers menunjukkan outlier.

Dari visualisasi tersebut, dapat diamati bahwa karyawan yang berhenti memiliki nilai median umur dan lama kerja yang lebih rendah dibandingkan dengan karyawan aktif. Hal ini mengindikasikan bahwa karyawan yang lebih muda dan dengan masa kerja yang relatif lebih singkat cenderung memiliki kemungkinan yang lebih tinggi untuk mengundurkan diri. Perbedaan distribusi ini memberikan indikasi bahwa kedua fitur tersebut berpotensi memiliki kontribusi penting dalam membedakan karakteristik antara karyawan aktif dan karyawan yang telah mengalami terminasi.

4.3.3 Data Splitting

Setelah dataset selesai diproses dan fitur-fitur yang relevan telah direkayasa, langkah selanjutnya adalah memisahkan data menjadi variabel independen (fitur) dan dependen (label). Dalam hal ini, fitur disimpan dalam variabel X, sedangkan label target disimpan dalam variabel y, yang merepresentasikan status terminasi karyawan (TERMINATED). Proses ini ditunjukkan pada Gambar 4.11 bagian atas.

Gambar 4.11 Penerapan Kode untuk K-Fold Validation

Mengingat distribusi kelas dalam data bersifat tidak seimbang, yaitu terdapat 180 data karyawan aktif (kelas 0) dan hanya 30 data karyawan yang berhenti (kelas 1), maka dibutuhkan strategi validasi yang mampu menjaga proporsi kelas tersebut selama pelatihan dan pengujian model.

Metode validasi yang digunakan adalah K-Fold Cross Validation dengan jumlah lipatan (fold) sebanyak lima. Dengan metode ini, dataset dibagi menjadi lima bagian yang kurang lebih berukuran seimbang. Secara bergiliran, empat bagian

digunakan sebagai data pelatihan, sedangkan satu bagian sisanya digunakan sebagai data validasi. Proses ini diulang sebanyak lima kali sehingga setiap data memperoleh kesempatan untuk menjadi bagian dari data uji. Pendekatan ini memberikan evaluasi performa model yang lebih stabil, menyeluruh, dan tidak bergantung pada satu skenario pembagian data tertentu.

Dalam implementasinya digunakan fungsi StratifiedKFold dari pustaka Scikit-learn, yang memastikan bahwa proporsi kelas pada setiap lipatan tetap konsisten. Hal ini sangat penting pada data tidak seimbang seperti ini, agar kelas minoritas tetap terwakili dalam setiap fold, sehingga hasil validasi lebih representatif dan menghindari bias terhadap kelas mayoritas.

Pendekatan validasi ini merujuk pada penjelasan di Subbab 2.6 mengenai K-Fold Cross Validation, di mana evaluasi dilakukan secara berulang dengan pembagian data yang bergantian untuk menguji generalisasi model secara lebih komprehensif.

4.4. Model Implementation

```
param_grid = {
    'n_estimators': [50, 100, 200, 300, 400, 500],
    'learning_rate': [0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0],
    'max_depth': [3, 4, 5, 6, 7, 8, 9, 10],
    'subsample': [0.6, 0.8, 1.0],
    'colsample_bytree': [0.6, 0.8, 1.0],
    'gamma': [0, 1, 2, 3, 4, 5],
    'reg_lambda': [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 1.0, 1.1, 1.2, 1.3, 1.4, 2.0],
    'reg_alpha': [0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0],
}
```

```
xgb_clf = XGBClassifier(objective='binary:logistic', eval_metric='logloss', seed=random_state, enable_categorical=True)
```

```
# Tuning hyperparameter
random_search = RandomizedSearchCV(
    estimator=xgb_clf,
    param_distributions=param_grid,
    n_iter=7000,
    cv=kfold,
    scoring='f1',
    verbose=1,
    n_jobs=-1,
    random_state=random_state
)

# Jalankan pencarian
random_search.fit(X, y)

print("Best Hyperparameters:")
print(random_search.best_params_)

# Gunakan best estimator untuk cross-validation manual lagi
best_model = random_search.best_estimator_
```

```
Fitting 5 folds for each of 7000 candidates, totalling 35000 fits
Best Hyperparameters:
{'subsample': 0.6, 'reg_lambda': 0.7, 'reg_alpha': 0.5, 'n_estimators': 500, 'max_depth': 7, 'learning_rate': 0.9, 'gamma': 5, 'colsample_bytree': 0.6}
```

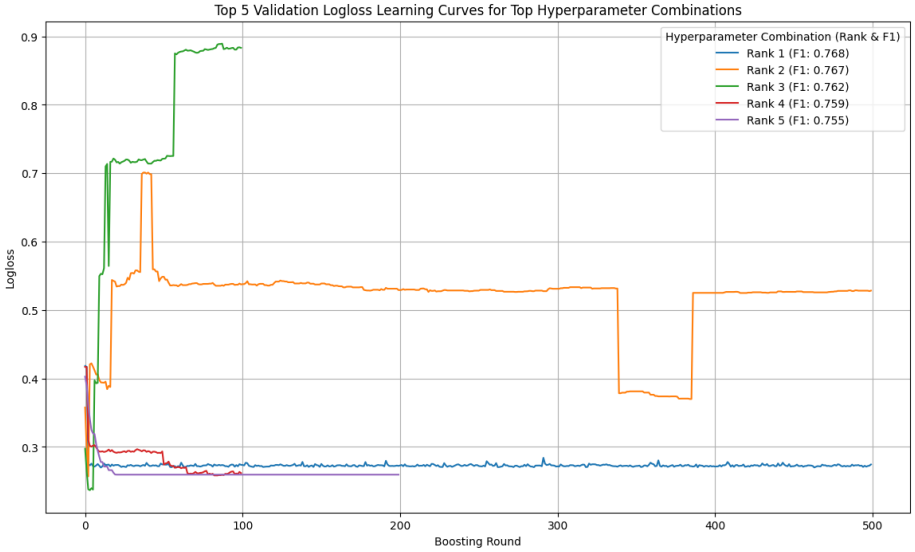
Gambar 4.12 Kode untuk Tuning Parameter Terbaik

Pada tahap ini (Gambar 4.12), dilakukan pembangunan model klasifikasi menggunakan algoritma XGBoost, yang sebelumnya telah dijelaskan pada Subbab

2.5. Implementasi dilakukan dalam kerangka validasi 5-Fold Cross Validation sebagaimana dijabarkan pada Subbab 4.3.3, dengan tujuan untuk mengevaluasi performa model secara menyeluruh dan mengurangi kemungkinan bias akibat pembagian data yang tidak merata.

Model diinisialisasi menggunakan kelas XGBClassifier dari library xgboost, dengan objective='binary:logistic' untuk kasus klasifikasi biner, dan eval_metric='logloss' sebagai metrik evaluasi internal. Untuk memperoleh performa model yang optimal, dilakukan proses penyesuaian *hyperparameter* menggunakan metode RandomizedSearchCV dengan 7000 iterasi pencarian seperti pada Gambar 4.12. Proses ini bertujuan untuk menemukan kombinasi parameter terbaik dari ruang pencarian yang telah ditentukan, menggunakan F1-Score sebagai metrik evaluasi utama. Kombinasi terbaik yang ditemukan adalah {'subsample': 0.6, 'reg_lambda': 0.7, 'reg_alpha': 0.5, 'n_estimators': 500, 'max_depth': 7, 'learning_rate': 0.9, 'gamma': 5, 'colsample_bytree': 0.6}.

Setelah proses pencarian selesai, tidak hanya kombinasi parameter dengan skor F1 tertinggi yang diambil, tetapi juga dilakukan analisis terhadap *learning rate* dari lima kandidat terbaik. Gambar 4.13 menggambarkan perbandingan kurva pembelajaran (*learning curve*) dari lima kombinasi hyperparameter dengan peringkat teratas. Analisis ini memastikan model yang dipilih tidak hanya memiliki skor akhir yang tinggi, tetapi juga stabil dan dapat diandalkan.



Gambar 4.13 Kurva Pembelajaran Logloss untuk 5 Kombinasi *Hyperparameter* Terbaik

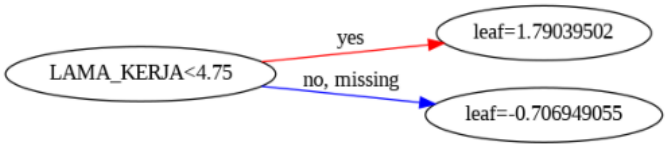
Gambar 4.13 menampilkan perbandingan proses pembelajaran dari lima model teratas yang diukur berdasarkan Logloss pada data validasi. Setiap garis mewakili satu set hyperparameter yang diurutkan berdasarkan F1-Score akhir yang dihasilkannya. Dari Gambar 4.13, dapat ditarik beberapa kesimpulan. Kombinasi Rank 1, 4, dan 5 menunjukkan kurva pembelajaran yang ideal. Nilai Logloss mereka rendah secara konsisten dan sangat stabil sepanjang proses pelatihan. Ini menandakan bahwa model-model ini belajar dengan efisien dan memiliki kemampuan generalisasi yang baik. Sedangkan menunjukkan kurva *Logloss* yang jauh lebih tinggi dan sangat tidak stabil. Fluktuasi yang liar ini mengindikasikan bahwa performa model sangat sensitif terhadap data dan cenderung *overfitting* pada *noise*, sehingga tidak dapat diandalkan.

Pemilihan model terbaik tidak hanya didasarkan pada skor F1-Score tertinggi. Stabilitas dan konsistensi proses pembelajaran merupakan faktor yang sama pentingnya. Meskipun Peringkat 2 memiliki F1-Score yang hampir identik dengan Peringkat 1 (0.767 vs 0.768), kurva pembelajarannya yang tidak stabil membuatnya menjadi pilihan yang berisiko. Sehingga, *hyperparameter* Rank 1 dipilih sebagai model final karena menawarkan kombinasi terbaik dari kedua aspek.

```
accuracy_scores, precision_scores, recall_scores, f1_scores = [], [], [], []

for train_idx, val_idx in kfold.split(X, y):
    X_train_fold, X_val_fold = X.iloc[train_idx], X.iloc[val_idx]
    y_train_fold, y_val_fold = y.iloc[train_idx], y.iloc[val_idx]

    best_model.fit(X_train_fold, y_train_fold)
    y_pred_fold = best_model.predict(X_val_fold)

    # Simpan metrik
    accuracy_scores.append(accuracy_score(y_val_fold, y_pred_fold))
    precision_scores.append(precision_score(y_val_fold, y_pred_fold))
    recall_scores.append(recall_score(y_val_fold, y_pred_fold))
    f1_scores.append(f1_score(y_val_fold, y_pred_fold))
```

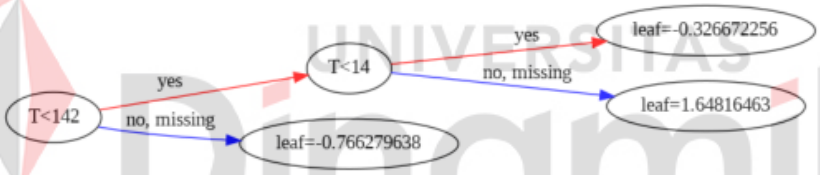
Gambar 4.14 Kode Training Model Berdasarkan Tuning Parameter

Berdasarkan pelatihan pada model (Gambar 4.14) maka menghasilkan beberapa pohon keputusan. Berikut merupakan sampel pohon keputusan yang telah dibangun oleh model XGBoost.



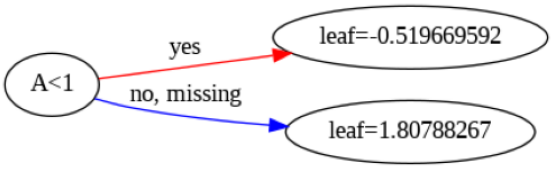
Gambar 4.15 Pohon Keputusan 1

Pohon pertama (Gambar 4.15) memulai proses pemisahan berdasarkan fitur LAMA_KERJA dengan batas pemisahan pada nilai 4.75 tahun. Jika LAMA_KERJA < 4.75, maka data diarahkan ke leaf dengan nilai +1.79, yang berarti kontribusi positif terhadap prediksi kelas 1. Sebaliknya, jika lebih dari atau sama dengan 4.75 tahun, maka diarahkan ke leaf dengan nilai -0.71, yang mengindikasikan kontribusi negatif terhadap probabilitas kelas 1. Artinya, karyawan dengan masa kerja yang lebih singkat cenderung lebih berpotensi termasuk dalam kelas target menurut model ini, mungkin karena pola tertentu dalam data historis.



Gambar 4.16 Pohon Keputusan 2

Pohon kedua (Gambar 4.16) memanfaatkan fitur T, yang merupakan jumlah hari kehadiran tepat waktu. Pemisahan pertama dilakukan pada T < 142, lalu dilanjutkan pemisahan kedua pada T < 14. Jika T ≥ 142 dan T ≥ 14, maka data diarahkan ke leaf bernilai +1.64, yang berarti kontribusi besar terhadap kelas target. Sebaliknya, jika T < 14, maka data diarahkan ke leaf -0.33, menunjukkan pengaruh negatif. Secara keseluruhan, semakin tinggi nilai T (semakin sering hadir tepat waktu), maka kontribusi terhadap prediksi positif semakin besar, yang konsisten secara intuitif — model menghargai disiplin kehadiran.



Gambar 4.17 Pohon Keputusan 3

Pohon ketiga (Gambar 4.17) menggunakan fitur A, yaitu jumlah ketidakhadiran tanpa keterangan (alpha). Jika A < 1, maka masuk ke leaf -0.52, menandakan kontribusi negatif terhadap kelas target. Jika A ≥ 1, maka diarahkan ke leaf +1.81, yang memberi kontribusi logit positif. Hasil ini bisa jadi dipengaruhi oleh interaksi antar fitur dalam boosting lain, atau karena outlier di mana ketidakhadiran justru muncul dari kategori yang sering diprediksi positif, sehingga leaf-nya bersifat kompensatif. Pohon – pohon keputusan berlanjut hingga pohon keputusan ke-500, sesuai dengan n_estimators yang telah diinisialisasi.

4.5. Evaluation

Evaluasi model bertujuan untuk mengukur seberapa baik performa model XGBoost dalam memprediksi kemungkinan terjadinya turnover pada karyawan. Penilaian dilakukan berdasarkan rata-rata metrik evaluasi dengan tambahan standar deviasi yang diperoleh melalui skema 5-Fold Cross Validation. Pendekatan ini dipilih karena mampu memberikan estimasi performa yang lebih stabil dan representatif dibandingkan dengan pembagian data latih dan uji secara tunggal.

Metrik evaluasi yang digunakan meliputi Accuracy, Precision, Recall, dan F1 Score. Rata-rata hasil dari kelima lipatan validasi disajikan pada Tabel 4.3 berikut:

Tabel 4.3 Hasil Metrik Evaluasi

Metrik	Nilai (Rata-rata 5-fold)
Akurasi	93,66% ± 3,31%
Presisi	85,24% ± 15,68%
Recall	73,33% ± 17,00%
F1-Score	76,77% ± 11,22%

Hasil evaluasi menunjukkan bahwa model XGBoost memiliki performa yang cukup baik dalam memprediksi status turnover. Meskipun akurasi tergolong tinggi, dalam konteks klasifikasi yang tidak seimbang seperti kasus ini, metrik Precision dan Recall memiliki peran yang lebih penting untuk dianalisis.

Precision sebesar 85,24% mengindikasikan bahwa dari seluruh prediksi turnover yang dihasilkan model, sekitar 85% benar-benar merupakan karyawan yang mengalami turnover. Nilai ini penting untuk menghindari terlalu banyak "alarm palsu" (false positives) yang dapat mengganggu pengambilan keputusan.

Recall sebesar 73,33% menunjukkan bahwa sekitar 73% dari total karyawan yang benar-benar mengalami turnover berhasil dikenali oleh model. Artinya, masih ada sebagian karyawan yang mengalami turnover tetapi tidak berhasil diprediksi oleh model (false negatives), yang tentu penting untuk diminimalkan jika tujuan utamanya adalah pencegahan turnover sejak dini.

F1 Score yang berada di angka 76,77% menjadi metrik keseimbangan antara Precision dan Recall. Nilai ini menunjukkan bahwa model cukup seimbang dalam mendeteksi turnover sambil tetap menjaga akurasi prediksinya. Dengan demikian, performa model dapat dikatakan efektif, terutama jika diterapkan sebagai sistem pendukung keputusan di lingkungan kerja dengan kondisi data serupa.

4.6. Deployment

Pada tahap *deployment*, model prediksi *turnover* karyawan yang telah dilatih dan dievaluasi diimplementasikan ke dalam sebuah sistem yang praktis dan interaktif. Implementasi ini diwujudkan dalam bentuk dashboard berbasis *website* yang dibangun menggunakan *framework* Streamlit dengan bahasa pemrograman Python. Dashboard ini dirancang untuk menyajikan hasil analisis dan prediksi secara visual agar mudah dipahami oleh pihak manajemen atau divisi HR dalam pengambilan keputusan strategis.

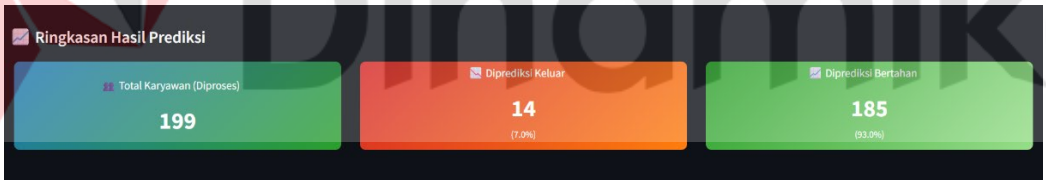
4.6.1 Antarmuka Utama dan Proses Prediksi

Fungsi utama dashboard adalah untuk melakukan prediksi *turnover* secara massal (*batch*) berdasarkan data karyawan yang diunggah. Dashboard ini dirancang dengan fokus pada kemudahan penggunaan sehingga pengguna tidak perlu memiliki pengetahuan teknis tentang machine learning. Alur kerjanya dimulai dari: unggah data, dan sistem akan menangani sisanya. Proses prediksi yang berjalan di latar belakang mencakup semua tahapan yang telah dibahas sebelumnya, mulai dari data cleaning, feature engineering (seperti menghitung lama kerja dan jarak tinggal), hingga penerapan model XGBoost untuk menghasilkan prediksi turnover bagi setiap karyawan dalam data yang diunggah.

Gambar 4.18 Antarmuka Unggah Data pada Dashboard

Seperti yang ditunjukkan pada Gambar 4.18, pengguna dapat mengunggah file data karyawan dalam format .csv atau .xlsx. Untuk memastikan format data yang diunggah sesuai, disediakan juga fasilitas untuk mengunduh file template. Setelah data diunggah, sistem secara otomatis melakukan seluruh proses, mulai dari pra-pemrosesan data hingga penerapan model yang sudah ada untuk menghasilkan prediksi.

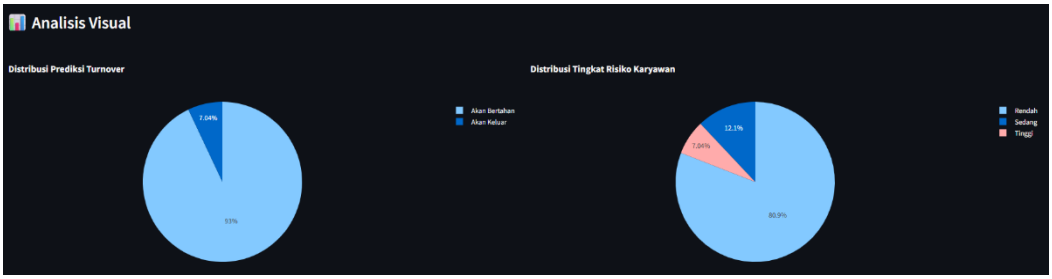
Hasil pertama yang ditampilkan adalah ringkasan metrik utama yang memberikan gambaran umum secara cepat.



Gambar 4.19 Ringkasan Metrik Utama Dashboard

Gambar 4.19 menampilkan tiga kartu metrik utama: total karyawan yang dianalisis, jumlah dan persentase karyawan yang diprediksi akan keluar (turnover), serta jumlah yang akan bertahan.

Selanjutnya, hasil prediksi divisualisasikan dalam bentuk diagram lingkaran untuk memudahkan interpretasi.

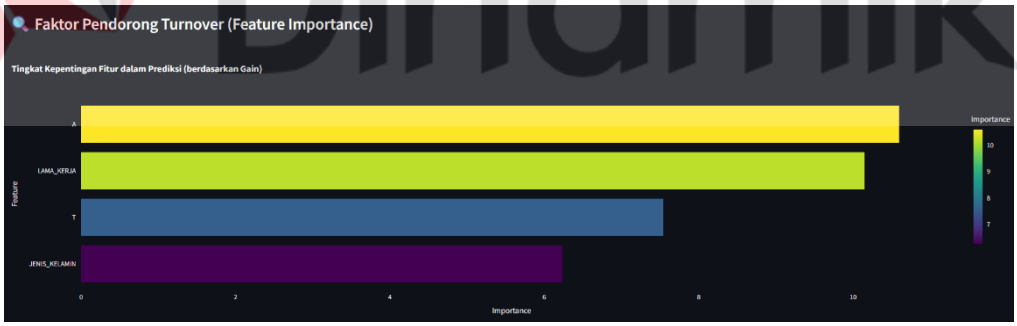


Gambar 4.20 Visualisasi Distribusi Prediksi dan Tingkat Risiko

Pada Gambar 4.20, diagram di sebelah kiri menunjukkan proporsi karyawan yang diprediksi akan keluar versus yang akan bertahan. Diagram di sebelah kanan mengklasifikasikan karyawan yang berisiko *turnover* ke dalam tiga level (Rendah, Sedang, Tinggi) berdasarkan probabilitas prediksi yang dihasilkan oleh model XGBoost.

4.6.2 Analisis Faktor Pendorong Turnover

Dashboard ini tidak hanya berfungsi sebagai alat prediksi, tetapi juga sebagai platform analisis untuk menggali wawasan lebih dalam. Salah satu visualisasi paling strategis yang ditampilkan adalah grafik feature importance.



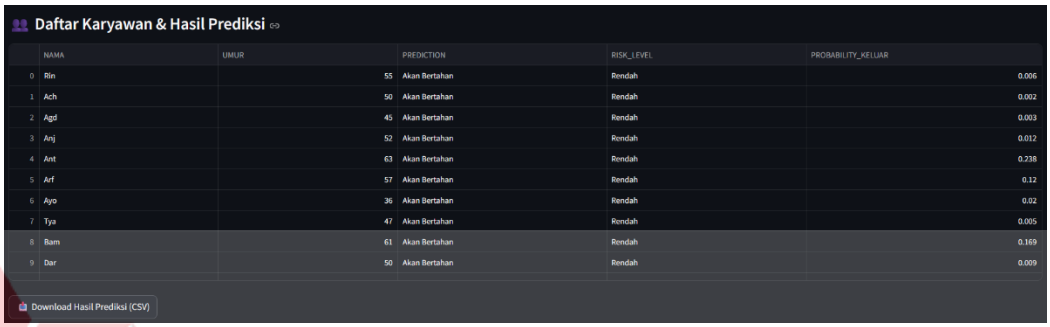
Gambar 4.21 Tingkat Kepentingan Fitur (Feature Importance)

Grafik pada Gambar 4.21 mengurutkan fitur-fitur karyawan dari yang paling berpengaruh hingga yang kurang berpengaruh dalam menentukan prediksi turnover. Dari visualisasi ini, dapat disimpulkan bahwa faktor-faktor seperti A (Alpha), lama kerja, T (Tepat waktu), dan jenis kelamin merupakan pendorong utama yang paling dipertimbangkan oleh model. Adapun angka Gain yang diperoleh antara lain, A: 10,596, lama kerja: 10,149, T: 7,542, dan jenis kelamin: 6,233. Secara teknis, skor yang ditampilkan merupakan nilai rata-rata peningkatan

fungsi objektif (gain) setiap kali fitur digunakan untuk melakukan split. Landasan matematis terkait metrik gain ini telah dijelaskan pada Subbab 2.9.1, khususnya pada Persamaan (12), yang menunjukkan bagaimana kontribusi fitur dihitung berdasarkan perubahan nilai loss setelah split dilakukan.

4.6.3 Penyajian Hasil Prediksi

Hasil akhir dari proses prediksi disajikan dalam bentuk tabel interaktif yang merinci prediksi untuk setiap karyawan.


Daftar Karyawan & Hasil Prediksi

	NAMA	UMUR	PREDICTION	RISK_LEVEL	PROBABILITY_KELUAR
0	Rin	55	Akan Bertahan	Rendah	0.006
1	Ach	50	Akan Bertahan	Rendah	0.002
2	Agd	45	Akan Bertahan	Rendah	0.003
3	Anj	52	Akan Bertahan	Rendah	0.012
4	Ant	63	Akan Bertahan	Rendah	0.238
5	Arf	57	Akan Bertahan	Rendah	0.12
6	Ayo	36	Akan Bertahan	Rendah	0.02
7	Tya	47	Akan Bertahan	Rendah	0.005
8	Ram	61	Akan Bertahan	Rendah	0.169
9	Dar	50	Akan Bertahan	Rendah	0.009

Download Hasil Prediksi (CSV)

Gambar 4.22 Tabel Hasil Prediksi Karyawan

Tabel pada Gambar 4.22 menampilkan informasi penting seperti nama karyawan, data demografis, hasil prediksi (PREDICTION), tingkat risiko (RISK_LEVEL), dan probabilitas untuk keluar (PROBABILITY_KELUAR). Kolom tingkat risiko sendiri dibuat untuk menerjemahkan nilai probabilitas numerik menjadi kategori yang dapat ditindaklanjuti oleh pengguna. Pembagian ini bertujuan untuk memprioritaskan tindakan retensi, di mana karyawan diklasifikasikan sebagai Rendah (probabilitas 0-30%) yang dianggap stabil, Sedang (31-70%) untuk dipantau, dan Tinggi (>70%) yang memerlukan intervensi secepatnya. Pengguna juga dapat mengunduh keseluruhan hasil prediksi ini dalam format file .csv untuk analisis lebih lanjut.

BAB V
PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan, dapat ditarik beberapa kesimpulan sebagai berikut:

1. Model divalidasi menggunakan Confusion Matrix dengan menggunakan metode K-Fold Cross Validation, menghasilkan akurasi 93,66% ± 3,31%, presisi 85,24% ± 15,68%, recall 73.33% ± 17,00%, dan F1 score 76,77% ± 11,22%, menunjukkan performa yang andal namun masih kurang konsisten.
2. Faktor paling berpengaruh dalam prediksi turnover: A (Alpha), lama kerja, T (Tepat waktu), dan jenis kelamin.
3. Model telah diimplementasikan ke dalam dashboard web analitis (Streamlit) yang menyajikan prediksi dan analisis secara visual untuk mendukung pengambilan keputusan HR.

5.2 Saran

Berdasarkan kesimpulan yang telah diperoleh, ada beberapa saran yang dapat dipertimbangkan untuk dilakukan pada penelitian selanjutnya, antara lain:

1. Melakukan studi komparatif dengan algoritma ensemble learning lainnya seperti LightGBM atau CatBoost, dan mengeksplorasi model deep learning untuk melihat apakah ada peningkatan performa yang signifikan.
2. Untuk meningkatkan akurasi model, penelitian selanjutnya dapat mencoba mengintegrasikan fitur-fitur eksternal (seperti standar gaji industri) atau data kualitatif (seperti hasil survei kepuasan kerja, tingkat stres, atau beban kerja) yang dapat memberikan konteks lebih dalam mengenai motivasi karyawan.
3. Meskipun performa validasi sudah baik, terdapat indikasi overfitting pada penelitian ini. Penelitian mendatang dapat menerapkan teknik regularisasi yang lebih lanjut pada XGBoost atau menggunakan metode data augmentation seperti SMOTE untuk menangani ketidakseimbangan kelas secara lebih efektif.

DAFTAR PUSTAKA

Afina Nur'aini Tsaqila, & Lisa Widawati. (2025). Studi Kontribusi Job Insecurity terhadap Turnover Intention Karyawan Generasi Z Kota Bandung. *Bandung Conference Series: Psychology Science*, *5*(1), 931–936. https://doi.org/10.29313/bcsps.v5i1.18102

Alhamad, A. M., Hilan, I. M., Alghowl, I. S. M., Eljaiebi, M. I., & Buraqan, K. K. M. (2024). Predicting Employee Turnover Through Advanced Hr Analytics: Implications For Engagement Strategies. *Educational Administration: Theory and Practice*, 964–972. https://doi.org/10.53555/kuey.v30i5.2995

Alsahaf, A., Petkov, N., Shenoy, V., & Azzopardi, G. (2022). A framework for feature selection through boosting. *Expert Systems with Applications*, *187*, 115895. https://doi.org/10.1016/j.eswa.2021.115895

Anusha, K., & Rajesh, Dr. M. (2024). Impact of Employee Turnover on Organization Performance with Reference to Optum Global Solutions Pvt. Ltd, Hyderabad. *International Journal of Research Publication and Reviews*, *5*(7), 2362–2370. https://doi.org/10.55248/gengpi.5.0724.1810

Arromrit, T., Srisakaew, K., Roswhan, N., & Mahikul, W. (2023). A Supervised Machine Learning Method for Predicting the Employees Turnover Decisions. *2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS)*, 122–127. https://doi.org/10.1109/ICSECS58457.2023.10256357

Atef, M., S. Elzanfaly, D., & Ouf, S. (2022). Early Prediction of Employee Turnover Using Machine Learning Algorithms. *International journal of electrical and computer engineering systems*, *13*(2), 135–144. https://doi.org/10.32985/ijeces.13.2.6

Azeem, M. A., & Dev, S. (2024). A performance and interpretability assessment of machine learning models for rainfall prediction in the Republic of Ireland. *Decision Analytics Journal*, *12*, 100515. https://doi.org/10.1016/j.dajour.2024.100515

Bibers, I., & Abdallah, M. (2025). An ensemble learning framework for enhanced anomaly and failure detection in IoT systems. *Cyber Security and Applications*, *3*, 100105. https://doi.org/10.1016/j.csa.2025.100105

Chen, T., & Guestrin, C. (2016). *XGBoost: A Scalable Tree Boosting System*. https://doi.org/10.1145/2939672.2939785

Chowdhury, S., Joel-Edgar, S., Dey, P. K., Bhattacharya, S., & Kharlamov, A. (2023). Embedding transparency in artificial intelligence machine learning models: managerial implications on predicting and explaining employee turnover. *The International Journal of Human Resource Management*, *34*(14), 2732–2764. https://doi.org/10.1080/09585192.2022.2066981

Duan, Y. (2022). Statistical Analysis and Prediction of Employee Turnover Propensity Based on Data Mining. *2022 International Conference on Big Data, Information and Computer Network (BDICN)*, 235–238. https://doi.org/10.1109/BDICN55575.2022.00052

Duckworth, C., Chmiel, F. P., Burns, D. K., Zlatev, Z. D., White, N. M., Daniels, T. W. V., Kiuber, M., & Boniface, M. J. (2021). Using explainable machine learning to characterise data drift and detect emergent health risks for emergency department admissions during COVID-19. *Scientific Reports*, *11*(1), 23017. https://doi.org/10.1038/s41598-021-02481-y

Dwinanda, M. W., Satyahadewi, N., & Andani, W. (2023). CLASSIFICATION OF STUDENT GRADUATION STATUS USING XGBOOST ALGORITHM. *BAREKENG: Jurnal Ilmu Matematika dan Terapan*, *17*(3), 1785–1794. https://doi.org/10.30598/barekengvol17iss3pp1785-1794

Egwom .O. Jessica, Lawrenc Emmanuel, Moshood A. Hambali, & Kefas Rimamnuskeb Galadima. (2024). An Intelligent Analysis and Prediction of Employee Attrition Rate in Healthcare Using Machine Learning Techniques. *International Journal of Emerging Multidisciplinaries: Computer Science & Artificial Intelligence*, *3*(1). https://doi.org/10.54938/ijemdcsai.2024.03.1.352

Fuglkjær, A., Kilic, D. K., Eskesen, M. H., Poulsen, L. Ø., Niemann, C. U., Jensen, P., Søgaard, K. K., Werling, M., Christensen, F., Simonsen, M. R., Nielsen, I. E., & El-Galaly, T. C. (2024). Machine Learning for Prediction of Serious Infections in Patients with Treatment-Naïve Lymphoma; A Population-Based Study of 701 Patients from the North Denmark Region. *Blood*, *144*(Supplement 1), 3605–3605. https://doi.org/10.1182/blood-2024-193940

Gallagher, M., Novak, C., Pratt, A., Tuohy, J., & Bailey, R. (2025). A Decision-Driven Methodology for Business Intelligence Dashboards in Start-Ups. *2025 Systems and Information Engineering Design Symposium (SIEDS)*, 60–65. https://doi.org/10.1109/SIEDS65500.2025.11021199

Geron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* (2 ed.). O'Reilly Media, Inc.

Hom, P., & Seo, J. (2024). Voluntary Turnover in Organizations. Dalam *Oxford Research Encyclopedia of Business and Management*. Oxford University Press. https://doi.org/10.1093/acrefore/9780190224851.013.448

Isha, Thapliyal, N., Solanki, S., Pandey, N. K., & Papola, S. (2024). Employee Attrition Analysis Using XGBoost. *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, 1–6. https://doi.org/10.1109/IC3SE62002.2024.10593326

Jenis, J., Ondriga, J., Hrcek, S., Brumercik, F., Cuchor, M., & Sadovsky, E. (2023). Engineering Applications of Artificial Intelligence in Mechanical Design and

Optimization. *Machines*, *11*(6), 577. https://doi.org/10.3390/machines11060577

Kitwange, A., & Habi, R. (2024). The Impact of Leave Practices on Employees' Well-being and Organizational Performance: The Case of Morogoro Municipality, Tanzania. *International Journal of Innovative Science and Research Technology (IJISRT)*, 2588–2594. https://doi.org/10.38124/ijisrt/IJISRT24SEP321

Kudirat Bukola Adeusi, Prisca Amajuoyi, & Lucky Bamidele Benjami. (2024). Utilizing machine learning to predict employee turnover in high-stress sectors. *International Journal of Management & Entrepreneurship Research*, *6*(5), 1702–1732. https://doi.org/10.51594/ijmer.v6i5.1143

Kumar, P., Gaikwad, S. B., Ramya, S. T., Tiwari, T., Tiwari, M., & Kumar, B. (2023). Predicting Employee Turnover: A Systematic Machine Learning Approach for Resource Conservation and Workforce Stability. *RAiSE-2023*, 117. https://doi.org/10.3390/engproc2023059117

Li, H., Gao, J., Guo, Y., & Yuan, X. G. (2025). Application of XGBoost model and multi-source data for winter wheat yield prediction in Henan Province of China. *Big Data and Information Analytics*, *9*(0), 29–47. https://doi.org/10.3934/bdia.2025002

Liu, Q., Liu, H., Xu, J., Shao, W., & Bai, Y. (2025). Identifying Primary Brain Tumors and Lung Cancer Brain Metastases by Training XGBoost Models Based on Radiomics Features from Brain MRI Data. *Journal of Medical and Biological Engineering*, *45*(3), 400–406. https://doi.org/10.1007/s40846-025-00953-4

Liu, Z., Thapa, N., Shaver, A., Roy, K., Siddula, M., Yuan, X., & Yu, A. (2021). Using Embedded Feature Selection and CNN for Classification on CCD-INID-V1—A New IoT Dataset. *Sensors*, *21*(14), 4834. https://doi.org/10.3390/s21144834

Maharana, M., Rani, R., Dev, A., & Sharma, A. (2022). Automated Early Prediction of Employee Attrition in Industry Using Machine Learning Algorithms. *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 1–6. https://doi.org/10.1109/ICRITO56286.2022.9965017

Maulid, R. (2023, September 1). *Kenali Ensemble Learning pada Tipe Machine Learning*. https://dqlab.id/kenali-ensemble-learning-pada-tipe-machine-learning.

Meddage, D. P. P., Fonseka, I., Mohotti, D., Wijesooriya, K., & Lee, C. K. (2024). An explainable machine learning approach to predict the compressive strength of graphene oxide-based concrete. *Construction and Building Materials*, *449*, 138346. https://doi.org/10.1016/j.conbuildmat.2024.138346

Mehan, V. (2025). Advanced Artificial Intelligence Driven Framework for Lung Cancer Diagnosis Leveraging SqueezeNet with Machine Learning Algorithms using Transfer Learning. *Medicine in Novel Technology and Devices*, 100383. https://doi.org/10.1016/j.medntd.2025.100383

Mhatre, A., Mahalingam, A., Narayanan, M., Nair, A., & Jaju, S. (2020). Predicting Employee Attrition along with Identifying High Risk Employees using Big Data and Machine Learning. *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 269–276. https://doi.org/10.1109/ICACCCN51052.2020.9362933

Mueller, J. P., & Massaron, L. (2019). *Deep Learning for Dummies* (Vol. 1). John Wiley & Sons, Inc.

Mushava, J., & Murray, M. (2024). Flexible loss functions for binary classification in gradient-boosted decision trees: An application to credit scoring. *Expert Systems with Applications*, *238*, 121876. https://doi.org/10.1016/j.eswa.2023.121876

Nikunlaakso, R., Airaksinen, J. M., Pekkarinen, L., Aalto, V., Toivio, P., Kivimäki, M., Laitinen, J., & Ervasti, J. (2024). *Development and validation of a predictive score for personnel turnover: a data-driven analysis of employee survey responses*. https://doi.org/10.31235/osf.io/254bd

Patria, R. (2024, Januari 8). *Dashboard adalah: Pengertian, Jenis dan Fungsi Dashboard*. https://www.domainesia.com/berita/dashboard-adalah/.

Piras, G., Agostinelli, S., & Muzi, F. (2025). Smart Buildings and Digital Twin to Monitoring the Efficiency and Wellness of Working Environments: A Case Study on IoT Integration and Data-Driven Management. *Applied Sciences*, *15*(9), 4939. https://doi.org/10.3390/app15094939

Polat, O., Ayid Ahmad, A., Oyucu, S., Algül, E., Doğan, F., & Aksöz, A. (2025). Temporal-Spatial Feature Extraction in IoT-Based SCADA System Security: Hybrid CNN-LSTM and Attention-Based Architectures for Malware Classification and Attack Detection. *IEEE Access*, *13*, 102109–102132. https://doi.org/10.1109/ACCESS.2025.3577761

Rasyid, A. R., Wibowo, R., Bait, J. F., & Octavianunisa, W. (2024). HOW WORKLOAD INFLUENCES PERFORMANCE IN RELATION TO WORK EXPERIENCE AND JOB SATISFACTION. *International Conference of Business and Social Sciences*, 988–1001. https://doi.org/10.24034/icobuss.v4i1.582

Recilla, V. J., Enonaria, M. R. A., Florida, R. J., Bustillo, J. C. M., Abalorio, C. C., & Trillo, J. C. (2024). Predicting Employee Turnover Through Genetic Algorithm. *2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 1383–1391. https://doi.org/10.1109/ICESC60852.2024.10689796

Sanchhaya Education Private Limited. (2025, Januari 15). *Understanding the Confusion Matrix in Machine Learning*. https://www.geeksforgeeks.org/confusion-matrix-machine-learning/#what-is-a-confusion-matrix.

Saragih, E. H., Bayupati, I. P. A., & Putri, G. A. A. (2021). PENGEMBANGAN BUSINESS INTELLIGENCE DASHBOARD UNTUK MONITORING AKTIVITAS PARIWISATA (STUDI KASUS: DINAS PARIWISATA PROVINSI BALI) . *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)* , *8*(6), 1159–1168.

Savitri, T. A., Buchori, I., & Supratikta, H. (2024). Exploring The Role of Human Resources Information System in Employee Performance Management: A Systematic Literature Review. *Indonesian Development of Economics and Administration Journal*, *3*(1), 55–64. https://doi.org/10.70001/idea.v3i1.211

Se, C., Champahom, T., Jomnonkwao, S., & Ratanavaraha, V. (2025). *XGBoost-Based Prediction Model for Train Passenger Numbers: Evaluating the Effect of the COVID-19 Pandemic* (hlm. 436–449). https://doi.org/10.1007/978-981-96-6948-6_31

Seki, Y., Shibayama, A., Nishiyama, M., & Kikuchi, M. (2024). Machine learning models for predicting the compressive strengths of ordinary Portland cement concrete and alkali-activated materials. *Sustainable Materials and Technologies*, *42*, e01191. https://doi.org/10.1016/j.susmat.2024.e01191

Shaik, N. B., Jongkittinarukorn, K., & Bingi, K. (2024). XGBoost based enhanced predictive model for handling missing input parameters: A case study on gas turbine. *Case Studies in Chemical and Environmental Engineering*, *10*, 100775. https://doi.org/10.1016/j.cscee.2024.100775

Singh, J. P., Ghosh, D., Singh, J., Bhattacharjee, A., & Gourisaria, M. K. (2025). Optimized DenseNet Architectures for Precise Classification of Edible and Poisonous Mushrooms. *International Journal of Computational Intelligence Systems*, *18*(1), 143. https://doi.org/10.1007/s44196-025-00871-y

Stachova, K., Barokova, A., & Stacho, Z. (2021). Optimization of employee turnover through predictive analysis. *Management Trends in the Context of Industry 4.0*. https://doi.org/10.4108/eai.4-12-2020.2303446

Sudrajat, J., Siow, H. L., & Permana, I. A. (2024). A Review of Employee Turnover from the Perspective of the Philosophy of Science. *Journal of Multi-Disciplines Science (ICECOMB)*, *2*(2), 72–81. https://doi.org/10.59921/icecomb.v2i2.34

Wang, D., Guo, H., Sun, Y., Liang, H., Li, A., & Guo, Y. (2024). Prediction of Oil–Water Two-Phase Flow Patterns Based on Bayesian Optimisation of the XGBoost Algorithm. *Processes*, *12*(8), 1660. https://doi.org/10.3390/pr12081660

Yaragunda, V. R., Vaka, D. S., & Oikonomou, E. (2025). Land Subsidence Susceptibility Modelling in Attica, Greece: A Machine Learning Approach Using InSAR and Geospatial Data. *Earth*, *6*(3), 61. https://doi.org/10.3390/earth6030061

Yenurkar, G. K., Mal, S., Nyangaresi, V. O., Hedau, A., Hatwar, P., Rajurkar, S., & Khobragade, J. (2023). Multifactor data analysis to forecast an individual's severity over novel COVID-19 pandemic using extreme gradient boosting and random forest classifier algorithms. *Engineering Reports*, *5*(12). https://doi.org/10.1002/eng2.12678

Yin, Z., Hu, B., & Chen, S. (2024). *Predicting Employee Turnover in the Financial Company: A Comparative Study of CatBoost and XGBoost Models*. https://doi.org/10.20944/preprints202410.0072.v1

Yousef, M., & Allmer, J. (2023). Deep learning in bioinformatics. *Turkish Journal of Biology*, *47*(6). https://doi.org/10.55730/1300-0152.2671

Zhang, Z., Liu, B., Xie, C., & Yan, E. (2024). Research on Fault Diagnosis Method for Photovoltaic Array Based on XGBoost Algorithm. *EAI Endorsed Transactions on Energy Web*, *12*. https://doi.org/10.4108/ew.7224