Klasifikasi *Turnover* Karyawan Menggunakan Algoritma XGBoost (Studi kasus: Divisi Engineering, Perusahaan Jasa Pertambangan)

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata-1
Program Studi Teknik Industri - Fakultas Teknologi Industri
Universitas Islam Indonesia

Nama : Luthfiyyah Wasiilah Maahiroh
No. Mahasiswa : 19522336

PROGRAM STUDI TEKNIK INDUSTRI PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
YOGYAKARTA
2024

PERNYATAAN KEASLIAN

Saya mengakui bahwa tugas akhir ini adalah hasil karya saya sendiri kecuali kutipan dan ringkasan yang seluruhnya sudah saya jelaskan sumbernya. Jika dikemudian hari ternyata terbukti pengakuan saya ini tidak benar dan melanggar peraturan yang sah maka saya bersedia ijazah yang telah saya terima ditarik kembali oleh Universitas Islam Indonesia.

Yogyakarta, 07 – 02 – 2024

(Luthfiyyah Wasiilah Maahiroh)
19522336

SURAT BUKTI PENELITIAN

UNIVERSITAS ISLAM INDONESIA

FAKULTAS TEKNOLOGI INDUSTRI
Gedung KH. Mas Mansur
Kampus Terpadu Universitas Islam Indonesia
Jl. Kaliurang km 14,5 Yogyakarta 55584
T. (0274) 898444 ext. 4100, 4101
F. (0274) 895007
E. fti@uii.ac.id
W. fti.uii.ac.id

SURAT KETERANGAN PENELITIAN

Nomor: 002/Ka.Lab.Datmin/70/Lab.Datmin/II/2024

Assalamu'alaikum Warahmatullahi Wabarakatuh

Kami yang bertanda tangan dibawah ini, menerangkan bahwa mahasiswa dengan keterangan sebagai berikut :

Nama	:	Luthfiyyah Wasiilah Maahiroh
No. Mhs	:	19522336
Dosen Pembimbing	:	Annisa Uswatun Khasanah, ST., M.B.A., M.Sc..

Telah selesai melaksanakan penelitian yang berjudul "Klasifikasi Turnover Karyawan Menggunakan Algoritma XGBoost (Studi Kasus: Divisi Engineering, Perusahaan Jasa Pertambangan)" di Laboratorium Data Mining, Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia tercatat mulai tanggal 04 September 2023 sampai dengan tanggal 04 Desember 2023

Demikian surat keterangan kami keluarkan, agar dapat dipergunakan sebagaimana mestinya.

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 07 Februari 2024

Kepala Laboratorium
Data Mining

Annisa Uswatun Khasanah, ST., M.B.A., M.Sc.

LEMBAR PENGESAHAN PEMBIMBING

Klasifikasi *Turnover* Karyawan Menggunakan Algoritma XGBoost (Studi kasus: Divisi Engineering, Perusahaan Jasa Pertambangan)

TUGAS AKHIR

Disusun Oleh :

Nama : Luthfiyyah Wasiilah Maahiroh
No. Mahasiswa : 19522336

Yogyakarta, 07 02 2024

Dosen Pembimbing

(Annisa Uswatun Khasanah, S.T., M.Sc.)

LEMBAR PENGESAHAN DOSEN PENGUJI

Klasifikasi *Turnover* Karyawan Menggunakan Algoritma XGBoost
(Studi kasus: Divisi Engineering, Perusahaan Jasa Pertambangan)

TUGAS AKHIR

Disusun Oleh :
Nama : Luthfiyyah Wasiilah Maahiroh
No. Mahasiswa : 19 522 336

Telah dipertahankan di depan sidang penguji sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Teknik Industri Fakultas Tekonologi Industri Universitas Islam Indonesia

Yogyakarta, 07 - Maret – 2024

Tim Penguji

Annisa Uswatun Khasanah, S.T., M.Sc.
Ketua

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM
Anggota I

Ir. Vembri Noor Helia, S.T., M.T., IPM
Anggota II

Mengetahui,
Ketua Program Studi Teknik Industri Program Sarjana
Fakultas Teknologi Industri
Universitas Islam Indonesia

Ir. Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM.
015220101

HALAMAN PERSEMBAHAN

Tugas Akhir ini Saya persembahkan untuk kedua orang tua, kedua saudara, keluarga besar, dan seluruh pihak yang senantiasa mendoakan dan mendukung di setiap perjalanan dan perjuangan yang Saya lalui selama ini.

MOTTO

Karena sesungguhnya sesudah kesulitan itu ada kemudahan, sesungguhnya sesudah kesulitan itu ada kemudahan. Maka apabila kamu telah selesai (dari suatu urusan), kerjakanlah dengan sungguh-sungguh (urusan) yang lain, dan hanya kepada Tuhanmulah hendaknya kamu berharap. — Q.S. Al Insyirah: 5-8 —

KATA PENGANTAR

Segala puji dan syukur bagi Allah SWT, yang dengan rahmat, petunjuk dan kasih sayang-Nya, penulis dapat menyelesaikan tugas akhir yang berjudul **"Klasifikasi Turnover Karyawan Menggunakan Algoritma XGBoost (Studi kasus: Divisi Engineering, Perusahaan Jasa Pertambangan)"** sebagai pemenuhan salah satu syarat untuk memperoleh gelar Sarjana Strata-1 Program Studi Teknik Industri, Fakultas Teknologi Industri, Universitas Islam Indonesia.

Penulis menyadari bahwa dalam proses penulisan tugas akhir ini, penulis menghadapi berbagai tantangan. Namun, dengan bantuan, petunjuk, dan kerjasama dari berbagai pihak, serta berkat rahmat Allah SWT, semua tantangan tersebut dapat diatasi. Untuk itu, penulis ingin menyampaikan rasa terima kasih dan penghormatan kepada:

1. Bapak Prof., Dr., Ir., Hari Purnomo, M.T., IPU, ASEAN.Eng selaku Dekan Fakultas Teknologi Industri, Universitas Islam Indonesia.
2. Bapak Ir., Muhammad Ridwan Andi Purnomo, S.T., M.Sc., Ph.D., IPM selaku Ketua Program Studi Teknik Industri Program Sarjana, Universitas Islam Indonesia.
3. Ibu Annisa Uswatun Khasanah, ST., M.B.A., M.Sc. selaku Dosen Pembimbing Tugas Akhir yang telah memberikan bimbingan, ilmu, waktu dan dorongan kepada penulis selama penyusunan Tugas Akhir ini.
4. Seluruh pihak PT. PNR yang telah mengizinkan dan memberikan dukungan selama peneliti melaksanakan penelitian.
5. Seluruh keluarga besar terutama kedua orang tua tersayang, abi Sugiarto dan ummi Rin Lestari yang kasih sayang dan doanya tak pernah terhenti. Serta kedua saudara kandung tercinta, Nabiilah Suhaimah Sobriyyati dan Aqilah Al Mardhiyyah yang selalu memberikan dukungan dan semangat.
6. Rekan-rekan seperjuangan Teknik Industri 2019, terutama Azzati Sahirah Elfahmi, Nur Widyasanti dan Putri Meilani yang telah menjadi sahabat seperjuangan sejak masa awal kuliah.
7. Rekan-rekan Laboratorium Data Mining Teknik Industri, yang telah memberikan lingkungan belajar yang sangat baik untuk mengembangkan diri dan pengetahuan.
8. Rekan-rekan seperjuangan SMP dan SMA IT Darul Quran Mulia, yang masih selalu saling mendukung, mengingatkan dan menjaga lingkungan pertemanan yang baik.
9. Semua pihak yang tidak dapat dituliskan satu persatu, yang telah terlibat dan memberikan dukungan dalam penyelesaian penulisan Tugas Akhir.

Penulis sepenuhnya menyadari bahwa Tugas Akhir ini mungkin memiliki kekurangan. Oleh karena itu, penulis sangat menghargai kritik serta saran yang membangun untuk perbaikan di masa mendatang. Penulis berharap bahwa Tugas Akhir ini dapat memberikan inspirasi kepada pembaca, untuk melakukan pengembangan dan mencapai kebermanfaatan yang lebih baik lagi.

ABSTRAK

Efisiensi sumber daya manusia dan manajemennya, sangat berkaitan dengan perusahaan yang bergantung pada karyawan untuk mendapatkan keuntungan di pasar yang kompetitif. Industri pertambangan merupakan salah satu bidang industri yang keberhasilan dalam organisasinya ditentukan oleh karyawan. PT. PNR merupakan kontraktor spesialis, yang menyediakan jasa pertambangan komprehensif kepada pemilik tambang. Pada salah satu jalur tenaga ahli yang dimiliki PT. PNR, terdapat peningkatan *turnover* yang signifikan. Hal ini tentu akan memberikan kerugian, mengingat besarnya dampak negatif yang terjadi apabila fenomena *turnover* tidak dikelola secara tepat seperti yang disebutkan penelitian terdahulu. Serta pengamatan terdahulu telah menunjukan bahwa karyawan yang pergi adalah yang paling berbakat, karena bagi karyawan berbakat tersebut lebih mudah mendapatkan pekerjaan alternatif. Penelitian ini membandingkan model klasifikasi XGBoost tanpa dan dengan SMOTE untuk pembangunan model klasifikasi *turnover* karyawan. Pada penelitian ini, label kelas memiliki proporsi yang tidak seimbang yaitu 91,08% kelas negatif dan 8,92% kelas positif. Hasil analisis menunjukan bahwa model XGBoost dengan SMOTE memiliki performa klasifikasi yang lebih unggul, terutama dalam menyeimbangkan prediksi kelas data minoritas dan mayoritas. Hasil tersebut menunjukkan pentingnya penanganan ketidakseimbangan kelas dalam pembuatan model klasifikasi. Model klasifikasi terbaik yang dihasilkan dari penelitian ini dapat digunakan oleh perusahaan sebagai bahan untuk manajemen retensi karyawan. Sehingga perusahaan dapat melakukan intervensi lebih awal untuk mencegah karyawan berhenti secara sukarela.

Kata Kunci: Industri Pertambangan, Klasifikasi, SMOTE, *Turnover*, XGBoost.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I

PENDAHULUAN

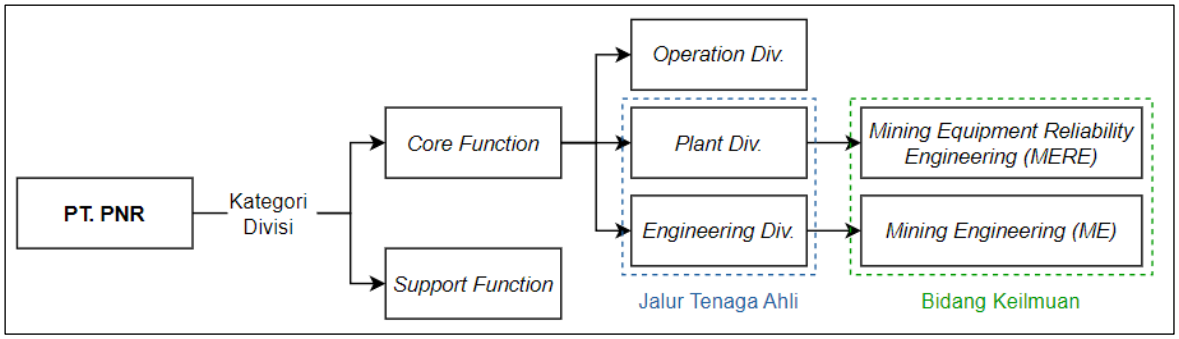
1.1 Latar Belakang

Persaingan pasar mendorong organisasi dalam mempekerjakan dan mempertahankan karyawan yang sangat berbakat (Anwar & Abdullah, 2021). Perputaran karyawan sebagai kepergian modal intelektual dari organisasi pemberi kerja pada periode tertentu, disebut dengan *turnover* karyawan (Palupi, 2021; Prawitasari, 2016; Punnoose & Ajit, 2016). Terdapat dua tipe *turnover*, yaitu sukarela dan tidak sukarela (White, 2022).

Dikatakan *turnover* sukarela, apabila seorang karyawan memutuskan untuk meninggalkan organisasi. Sedangkan *turnover* tidak sukarela adalah ketika suatu organisasi memutuskan untuk mengeluarkan karyawan dari posisi saat ini (Chhinzer, 2021). *Turnover* sukarela menjadi salah satu tantangan dan pertimbangan tersendiri bagi manajerial, karena mengindikasikan ketidakpuasan pekerjaan. Namun juga memungkinkan alasan sebenarnya tidak diungkapkan oleh karyawan yang pergi, karena masih membutuhkan organisasi untuk memberikan referensi di masa depan (Dwesini, 2019).

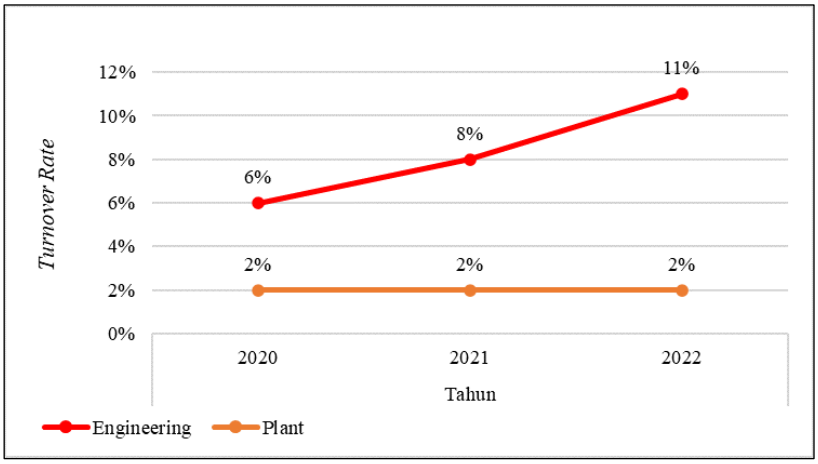
Turnover karyawan merupakan faktor yang mempengaruhi perilaku organisasi, dimana dengan mengelolanya (mengendalikan dan mengurangi) dapat secara efektif meningkatkan kinerja karyawan, komitmen karyawan terhadap perusahaan, penghematan waktu dan biaya perekrutan, lama bekerja (*years of service*) karyawan, pencapaian target yang tepat waktu hingga pencapaian tujuan yang sesuai (Lee & Liu, 2021; Palupi, 2021; Prawitasari, 2016; Punnoose & Ajit, 2016). Efisiensi sumber daya manusia (SDM) dan manajemennya, sangat berkaitan dengan perusahaan yang bergantung pada karyawan untuk mendapatkan keuntungan di pasar yang kompetitif (Anwar & Abdullah, 2021; Collins, 2020). Industri pertambangan merupakan salah satu bidang industri yang keberhasilan dalam organisasinya ditentukan oleh karyawan (Prabowo, 2019).

PT. PNR merupakan kontraktor spesialis, yang menyediakan jasa pertambangan komprehensif kepada pemilik tambang. Berdasarkan penelitian Dwesini (2019), pengamatan telah menunjukan bahwa karyawan yang pergi adalah yang paling berbakat, karena bagi karyawan berbakat tersebut lebih mudah mendapatkan pekerjaan alternatif. Jika dilihat berdasarkan struktur organisasi PT. PNR pada Gambar 1.1, terdapat jalur tenaga ahli yang merupakan bagian dari bisnis inti, yang menjadi keunggulan kompetitif perusahaan. Dalam hal ini, karyawan tenaga ahli di PT. PNR adalah seseorang yang memiliki pengetahuan dan keterampilan yang mendalam pada bidang keilmuan atau keahlian yang dimaksud, yaitu *Mining Engineering* (ME) dan *Mining Equipment Reliability Engineering* (MERE) dengan golongan minimal 4C. Divisi yang masuk ke dalam jalur tenaga ahli adalah Divisi Plant dan Divisi Engineering.



Gambar 1.1 Struktur Tenaga Ahli PT. PNR

Secara spesifik pada karyawan jalur tenaga ahli, perkembangan kejadian *turnover* sukarela yang terjadi dari tahun 2020 hingga 2022 dapat dilihat pada Gambar 1.2.



Gambar 1.2 Grafik Kejadian *Turnover* Sukarela PT. PNR (2020-2022)

Perkembangan kejadian *turnover* pada Gambar 1.2 ditunjukan oleh *turnover rate* yang merupakan persentase karyawan meninggalkan perusahaan dalam jangka waktu tertentu. Pada data tersebut digunakan jangka waktu yang umum untuk menghitung *turnover rate* karyawan yaitu jangka waktu tahunan. Nilai *turnover rate* ini dapat membantu perusahaan dalam memahami tingkat pergantian karyawan. Berdasarkan Putra & Surya (2020), *turnover rate* tahunan dapat dihitung melalui pembagian antara jumlah karyawan yang keluar, dengan rata-rata karyawan dalam setahun dan dikalikan 100%. Rata-rata karyawan yang dimaksud adalah jumlah karyawan awal tahun ditambah dengan jumlah karyawan akhir tahun, kemudian dibagi dua. Tabel 1.1 berikut merupakan contoh perhitungan *turnover rate* tahunan pada Divisi Engineering.

Tabel 1.1 Contoh Perhitungan *Turnover Rate* Tahunan (Divisi Engineering)

Tahun	**Jumlah karyawan awal tahun (orang) (1)**	**Jumlah karyawan akhir tahun (orang) (2)**	**Rata-rata (1+2)/2 (3)**	**Karyawan keluar (orang) (4)**	***Turnover Rate* (4/3)*100% (5)**
2020	208	195	201,5	12	5,96%
2021	188	248	218	17	7,80%
2022	177	324	250,5	28	11,18%

Turnover karyawan dikatakan normal apabila berkisar antara 5-10% per tahun. Jika lebih dari 10% maka dikategorikan tinggi (Putra & Surya, 2020). Mengacu pada Gallup dalam penelitian yang dilakukan oleh Iskandar dan Rahadi (2021), *turnover* yang ideal adalah 10% dalam setahun. Namun, persentase ideal ini bisa berbeda antara satu industri dengan industri lain dan satu perusahaan dengan perusahaan lainnya. Berdasarkan studi yang digunakan oleh PT. PNR, angka *turnover* sukarela yang dialami pada jalur tenaga ahli masih lebih tinggi dari industri sejenis di bidang pertambangan yakni sebesar 1,3% ataupun jasa pertambangan sebesar 1,5%. Divisi Engineering merupakan divisi pada jalur tenaga ahli, yang memiliki persentase *turnover* tinggi dan mengalami kenaikan yang signifikan setiap tahunnya. Sehingga penelitian ini akan berfokus pada divisi tersebut.

Turnover dapat memberikan dampak negatif seperti proyek dan target tertunda, pembubaran tim, kekurangan SDM (terutama pada divisi dengan *turnover* tinggi), kesulitan melakukan perekrutan untuk mencari kandidat dengan berbagai kriteria dalam waktu singkat, memakan waktu dan biaya yang lebih, hingga gangguan dalam produktivitas tempat kerja, moral dan strategi pertumbuhan jangka panjang (Chiat & Panatik, 2019; Noviyanti, 2018; Palupi, 2021; Prawitasari, 2016; Punnoose & Ajit, 2016).

Dampak negatif berupa kekurangan SDM dikonfirmasi oleh Communication & Advocacy Section Head PT. PNR, yang menyatakan bahwa ketika terjadi *turnover* pada Divisi Engineering, maka akan meningkatkan potensi terjadinya kekurangan *man power* dan adanya *lag competency*. Disampaikan juga bahwa dari kejadian *turnover* ini, muncul gangguan dalam produktivitas tempat kerja, dimana tim menjadi kurang kuat karena terlalu banyak pergantian anggota di dalamnya. Terlebih pada posisi tertentu yang berhubungan langsung dengan *customer*, karena adanya *relation* yang sudah terbangun antara karyawan yang sudah keluar dengan *customer* terkait. Pada Divisi Engineering sendiri, posisi yang berhubungan langsung dengan *customer* adalah Mine Plan Expert dan para Expert Coordinator, dimana karyawan pada posisi tersebut diharuskan untuk presentasi 2-3 kali sehari kepada *customer*.

Sedangkan untuk dampak negatif *turnover* dalam hal memakan waktu dan biaya yang lebih, berkaitan dengan sistem *development* yang menjadi fokus perusahaan. Sejak awal karyawan berkarir di PT. PNR, perusahaan cenderung lebih fokus terhadap pengembangan karyawan-karyawannya. Sehingga, kerugian yang dirasakan dari keluarnya karyawan adalah dari sisi usaha, waktu dan biaya yang telah diinvestasikan perusahaan terhadap proses pengembangan karyawan tersebut. Hal ini telah dikonfirmasi oleh Engineering Department Head di salah satu *site* PT. PNR. Didukung pula oleh informasi mengenai *loss cost* yang disampaikan oleh seorang Expert Track Management Officer, bahwa kerugian materi telah terjadi karena sudah banyaknya *development* yang diberikan, namun pada akhirnya karyawan tersebut memilih untuk meninggalkan perusahaan.

Tabel 1.2 *Loss Cost Resign Expert* (Periode 2021 - Mei 2022)

Golongan	**Jumlah Orang**	***Loss Cost* (Rupiah)**			
		Recruitment Cost	***Training Cost***	***Certification Cost***	***Total Cost***
4C	16	168.109.696	1.294.396.626	244.000.000	1.706.506.322
4D	20	210.137.120	1.904.135.655	336.000.000	2.450.272.775
4E	2	21.013.712	229.231.636	38.000.000	288.245.348
4F	2	21.013.712	405.001.194	42.000.000	468.014.906
Total	**40**	**420.274.240**	**3.832.765.111**	**660.000.000**	**4.913.039.351**

Tabel 1.2 menunjukan data *loss cost* dari *expert* PT. PNR yang telah keluar, per periode 2021 sampai dengan Mei 2022. *Recruitment Cost* merupakan biaya yang dikeluarkan dalam proses *recruitment & selection* untuk mendapatkan satu orang karyawan FGDP (*Fresh Graduate Development Program*) yang diproyeksikan menjadi *expert*. *Training Cost* adalah jumlah total biaya pelatihan yang diikuti oleh *expert* selama menjadi karyawan di PT. PNR. Sedangkan

Certification Cost adalah jumlah total biaya sertifikasi kompetensi *expert* yang diikuti selama menjadi karyawan di PT. PNR. Sejauh ini memang belum pernah dilakukan analisis perbandingan *loss cost resign expert* dengan pengembalian/keuntungan yang didapatkan perusahaan dari kinerja *expert*. Namun perusahaan telah memandang *loss cost resign expert* ini sebagai masalah.

Sejauh ini, program retensi yang telah diterapkan oleh PT. PNR terhadap karyawan *expert* adalah dengan memberikan rencana *scholarship*, membedakan perhitungan *salary increase* di awal tahun dan membedakan proses kenaikan golongan. Serta terdapat penalti atas keputusan keluar oleh karyawan dalam bentuk uang. Namun berdasarkan dampak-dampak negatif yang telah dirasakan oleh PT. PNR karena kejadian *turnover* yang terus meningkat ini, perusahaan berharap dapat memahami faktor-faktor pendorong utama *turnover*, dan melakukan antisipasi dengan memprediksi karyawan yang akan keluar. Sehingga, karyawan yang terprediksi akan keluar ini dapat diberikan perlakuan tertentu, yang pada akhirnya diharapkan dapat mengendalikan kejadian *turnover* dan mengurangi dampak negatif yang akan terjadi. Salah satu cara yang dapat digunakan untuk mengelola (mengendalikan dan mengurangi) *turnover* karyawan, adalah dengan memanfaatkan fungsi *data mining* yaitu klasifikasi dan prediksi (Gao et al., 2019; Manurung et al., 2021; Punnoose & Ajit, 2016; Zhao et al., 2019). Menggabungkan *data mining* dan teknik analisis prediktif yang canggih, dipastikan akan sangat meningkatkan kinerja dari manajemen SDM itu sendiri (Manurung et al., 2021).

Dalam penelitian Chanodkar et al. (2019) juga disebutkan bahwa dengan beragamnya solusi untuk *turnover* karyawan, organisasi lebih memilih teknik *machine learning* untuk memprediksi kejadiannya. Algoritma *Extreme Gradient Boosting* (XGBoost) merupakan salah satu algoritma *machine learning* dengan konsep *tree based* yang telah terbukti dapat digunakan dalam kasus klasifikasi dan prediksi *turnover* karyawan (Duan, 2022; Juvitayapun, 2021; Kovvuri & Dommeti, 2022; Tao et al., 2021; Tharani & Raj, 2020; Zhao et al., 2019). Algoritma ini memiliki kelebihan berupa pemanfaatan memori, *runtime* yang relatif rendah, penanganan kebisingan data, ketahanan terhadap *outlier* serta hasil yang akurat dan evaluasi kinerja yang baik (Juvitayapun, 2021; Ke et al., 2022; Kovvuri & Dommeti, 2021; Punnoose & Ajit, 2016; Sholikhati, 2022).

Meskipun XGBoost merupakan algoritma yang kompleks, namun penelitian yang dilakukan oleh Zhao et al. (2019), telah membuktikan bahwa pada kasus klasifikasi *turnover* karyawan dengan ukuran data kecil yaitu 100 baris data, XGBoost memberikan kinerja yang baik

berdasarkan nilai metrik presisi, F1 dan ROC. Tabel 1.3 berikut memberikan alasan tidak dipilihnya beberapa algoritma pengklasifikasi lain, berdasarkan kondisi data dalam penelitian ini.

Tabel 1.3 Perbandingan Metode Pengklasifikasi Lainnya

Algortima Lain	Kekurangan	Kondisi Data Penelitian
Support Vector Machine (SVM)	Sensitif terhadap *outlier* dan memiliki keterbatasan jika jumlah atribut relatif banyak (Foley, 2022; Iddrus & Junaedi, 2022).	Terdapat cukup banyak *outlier* dan jumlah fitur setelah dilakukan *one hot encoding* mencapai 80 kolom.
Naïve Bayes	Tidak ideal untuk kumpulan data dengan banyak variabel numerik (Husin, 2023).	Dalam kumpulan data penelitian, terdapat 7 atribut numerik dari keseluruhan atribut yang berjumlah 16 (sebelum dilakukan *one hot encoding*).
Random Forest	Cenderung bias saat berhadapan dengan variabel kategorik (Trivusi, 2022).	Terdapat 9 variabel (atribut) yang bersifat kategorik dari keseluruhan atribut yang berjumlah 16 (sebelum dilakukan *one hot encoding*).
ID3	Tidak bisa digunakan untuk himpunan data yang bernilai numerik (Suyanto, 2022).	Terdapat 7 atribut numerik.
Gradient Boosted Tree (GBT)	Sensitif terhadap *outlier* (Zuhairah, 2022).	Terdapat cukup banyak *outlier* dalam data penelitian
AdaBoost	Rentan terhadap *outlier* (Mortara et al., 2023).	

Menurut penelitian Sholikhati (2022), konsep dari klasifikasi adalah membangun sebuah model prediktif yang akan memprediksi label kelas data. Namun, masalah yang umumnya terjadi adalah label kelas dari contoh yang ada memiliki distribusi atau proporsi yang tidak seimbang (*imbalance data*). Masalah ini dapat mempengaruhi kinerja dari model dalam memprediksi kelas yang minoritas. Dalam beberapa penelitian, metode *Synthetic Minority Oversampling Technique* (SMOTE) digunakan untuk menangani masalah tersebut, karena lebih efektif mengurangi ketidakseimbangan data sampel melalui pembuatan data sintetik yang relatif dekat dengan contoh yang ada di fitur kelas minoritas (Sholikhati, 2022; Tao et al., 2021). Metode SMOTE juga terbukti mampu memperbaiki kinerja model, dimana model dapat memprediksi secara akurat pada semua kelas respon (Syukron et al., 2020). Pada penelitian ini, label kelas memiliki proporsi yang tidak seimbang yaitu 91,08% kelas negatif (156 karyawan masih bertahan) dan 8,92% kelas positif (14 karyawan telah meninggalkan perusahaan).

Kondisi ketidakseimbangan proporsi label kelas ini dapat menyebabkan model *machine learning* menjadi bias terhadap kelas mayoritas, dan menghasilkan kinerja yang buruk pada kelas minoritas. Metode SMOTE digunakan dalam penyeimbangan data tersebut karena cara kerja *oversampling*-nya yang membuat sampel sintetis dari kelas minoritas, bukan dengan membuat duplikat. Sehingga dapat membantu mencegah terjadinya *overfitting*. *Overfitting* terjadi ketika model terlalu kompleks dan mempelajari terlalu banyak detail dan *noise* dari data

pelatihan, yang mengakibatkan performanya buruk pada data yang belum pernah dilihat sebelumnya.

Klasifikasi akan diukur kinerja atau performansinya, dengan suatu matriks yang disebut *confusion matrix* dan metrik *Area Under Curve* (AUC) (Rachmi, 2020). *Confusion matrix* merupakan salah satu metode pengukuran keputusan paling klasik dalam *supervised machine learning* (Xu et al., 2020). AUC adalah metrik populer untuk mengukur kinerja pengklasifikasi. AUC diperoleh dengan menghitung luas di bawah kurva ROC (*Receiver Operating Characteristic*) (Wardhani et al., 2019). Penelitian oleh Ke et al. (2022) dan Mardiansyah et al. (2021) menunjukan bahwa integrasi SMOTE dengan XGBoost dapat mencapai hasil optimasi yang handal khususnya pada pemilihan *hyperparameter* dan atribut terbaik secara otomatis, serta memiliki kinerja model tertinggi diantara model lainnya yang dijadikan pembanding dalam penelitian.

Berdasarkan paparan permasalahan di atas, dilakukanlah penelitian "Klasifikasi *Turnover* Karyawan Menggunakan Algoritma XGBoost (Studi kasus: Divisi Engineering, Perusahaan Jasa Pertambangan). Penelitian ini menjadi penting, melihat besarnya dampak negatif yang terjadi apabila fenomena *turnover* tidak dikelola secara tepat seperti yang disebutkan penelitian terdahulu. Pada PT. PNR, belum terdapat pemanfaatan lanjutan dari data historis yang dimiliki, dengan menggunakan teknik yang ada dalam *data mining* dan algoritma *machine learning* untuk pengelolaan *turnover*. Sementara luasnya tantangan revolusi data yang dialami oleh sektor industri pertambangan, dapat menurunkan margin keuntungan secara drastis (Qi, 2020). Dalam klasifikasi, terdapat *record data* yang berisi atribut/prediktor dan label kelas prediksi. Pada penelitian ini, atribut didasarkan pada penelitian sebelumnya dan dibagi menjadi atribut demografis serta *job related and organizational*. Cara ini sudah dijelaskan dalam penelitian Khera & Divya (2019) yang menyatakan bahwa atribut-atribut yang sama (dalam klasifikasi dan prediksi *turnover* karyawan) dapat digunakan untuk memprediksi pergantian karyawan dalam suatu organisasi.

Data karyawan berdasarkan atribut demografis serta *job related and organizational* yang diperoleh, bersumber dari *bank data* yang dimiliki oleh Divisi Human Capital & Talent Development PT. PNR, dengan total 170 data karyawan (156 orang bertahan dan 14 lainnya telah meninggalkan perusahaan), 19 atribut/prediktor dan satu label kelas. Dari 19 atribut tersebut, terdapat tiga atribut yang spesifik dalam industri pertambangan. Hal ini dilakukan untuk mendukung keterbaruan penelitian, dan mempertimbangkan sorotan awal dari tantangan

turnover sukarela yaitu mengindikasikan ketidakpuasan pekerjaan (Dwesini, 2019). Ketiga atribut tersebut adalah *Site*, *Assignment Letter* dan *Production Plan*. Untuk meningkatkan kinerja model, atribut yang ada akan dilakukan seleksi (*feature selection*) dahulu dan melakukan *hyperparameter tuning* (Juvitayapun, 2021; Sholikhati, 2022; Tao et al., 2021). Penerapan *feature selection* dan *hyperparameter tuning* ini terbukti berguna dalam meningkatkan kemampuan prediktif model pada kasus klasifikasi XGBoost dengan *dataset* berukuran kecil (Shafila, 2020; Zhao et al., 2019). Hasil penelitian ini akan digunakan oleh perusahaan untuk melakukan antisipasi terhadap kejadian *turnover* karyawan.

1.2 Rumusan Masalah

Rumusan masalah dalam menjalankan proses penelitian ini adalah:

1. Bagaimana hasil performansi model klasifikasi XGBoost tanpa SMOTE dan XGBoost dengan SMOTE dalam memprediksi kejadian *turnover* karyawan Divisi Engineering, PT. PNR?
2. Berdasarkan model klasifikasi terbaik, atribut apa yang paling mempengaruhi kejadian *turnover* karyawan Divisi Engineering, PT. PNR?
3. Apa usulan perbaikan yang dapat diberikan kepada PT. PNR dalam menyelesaikan permasalahan yang terjadi?

1.3 Tujuan Penelitian

Berikut ini merupakan tujuan dari penelitian yang akan dilakukan:

1. Mendapatkan hasil performansi model klasifikasi XGBoost tanpa SMOTE dan XGBoost dengan SMOTE dalam memprediksi kejadian *turnover* karyawan Divisi Engineering, PT. PNR.
2. Mengetahui atribut yang paling mempengaruhi kejadian *turnover* karyawan Divisi Engineering, PT. PNR berdasarkan model klasifikasi terbaik.
3. Memberikan usulan perbaikan kepada PT. PNR dalam menyelesaikan permasalahan yang terjadi.

1.4 Manfaat Penelitian

Manfaat dari dilakukannya penelitian ini adalah:

1. Bagi Akademik.
    a. Mengaplikasikan keilmuan industri terhadap klasifikasi *turnover* karyawan.
    b. Referensi pembelajaran terkait pemanfaatan *data mining* dan *machine learning* dalam pengelolaan *turnover* karyawan.
2. Bagi Perusahaan.
    a. Memperkenalkan fungsi serta manfaat *data mining* dan *machine learning* dalam manajemen SDM.
    b. Memberikan model klasifikasi dan pengetahuan mengenai atribut yang paling mempengaruhi kejadian *turnover* pada karyawan terkait.
    c. Menjadi bahan evaluasi internal bagi manajemen untuk menyusun strategi dalam mempertahankan karyawan yang berbakat.

1.5 Batasan Penelitian

Berikut ini merupakan batasan penelitian yang ditetapkan:

1. Tipe *turnover* yang diteliti adalah *voluntary turnover* atau *turnover* sukarela.
2. Data yang digunakan adalah data karyawan pada Divisi Engineering PT. PNR golongan 4C ke atas, mulai Januari 2023 hingga Agustus 2023. Pertimbangannya adalah untuk tahun 2020-2022 masih dalam status masa *Covid-19* yang dikhawatirkan mempengaruhi kejadian *turnover* pada masa itu. Sedangkan pada tahun sebelum 2020, jumlah *expert* masih sangat sedikit karena baru "*established*" atau dikembangkan dan membutuhkan waktu lebih untuk penarikan datanya.

BAB II

TINJAUAN PUSTAKA

2.1 Kajian Literatur

Berikut merupakan kajian dari beberapa literatur terdahulu. Kajian terhadap beberapa literatur terdahulu dilakukan, untuk mendukung penelitian yang akan dilaksanakan. Penelitian pertama yang dilakukan oleh Chanodkar et al. (2019) menggunakan *machine learning* untuk memprediksi tingkat *turnover* karyawan di organisasi. Dibandingkan dengan algoritma *Logistic Regression*, *Naïve Bayes*, *AdaBoost*, dan *Random Forest* (RF), *Support Vector Machine* (SVM) menjadi algoritma pengklasifikasi terbaik (akurasi 88,29%) dengan atribut yang paling mempengaruhi *turnover* (*key factors*) adalah seringnya perjalanan bisnis, jarak dari rumah, usia, lembur dan kesenjangan dalam promosi. Prediksi yang akurat tentang *turnover* karyawan dapat membantu organisasi dalam merencanakan retensi karyawan dan pertumbuhan organisasi.

Penelitian kedua yang dilakukan oleh Duan (2022) melakukan analisis faktor-faktor penyebab *turnover* karyawan, untuk memprediksi dan mengendalikan kecenderungan *turnover*, serta memberikan rekomendasi saran kepada perusahaan. Model XGBoost (akurasi 98,8%, presisi 97,7%, *recall* 95,4%, *F1-score* 96,5%, AUC 0,99) memiliki akurasi yang lebih baik daripada model *Logistic* dalam memprediksi kecenderungan karyawan untuk keluar. Hasil analisis menunjukan bahwa karyawan yang cenderung keluar adalah karyawan yang memiliki tingkat kepuasan rendah, hasil evaluasi rendah atau tinggi, jumlah proyek banyak atau sedikit, rata-rata jam kerja tinggi, lama bekerja 3-5 tahun, tidak mengalami kecelakaan kerja, tidak mendapatkan promosi dan memiliki gaji rendah atau medium.

Penelitian mengenai pengaruh kepuasan kerja terhadap niat *turnover* pada karyawan di industri pertambangan telah dilakukan oleh Farizi & Tanuwijaya (2022). Hipotesis awal yang digunakan oleh penelitian ini adalah bahwa kepuasan kerja memiliki pengaruh negatif terhadap niat *turnover*. Penelitian ini menggunakan analisis *Structural Equation Model* (SEM) dengan *software* SmartPLS3 dalam pengolahan datanya. Hasil penelitian menunjukan bahwa kepuasan kerja berpengaruh negatif dan signifikan terhadap niat *turnover*. Artinya, jika perusahaan mampu meningkatkan kepuasan kerja karyawan, maka keinginan karyawan untuk keluar dari perusahaan akan menurun.

Gao et al. (2019) dalam penelitiannya melakukan prediksi *turnover* karyawan dengan algoritma *Weighted Quadratic Random Forest* (WQRF). Penelitian ini melakukan seleksi atribut (*feature selection*) terlebih dahulu untuk meningkatkan kinerja. Penelitian ini juga menilai bahwa tahapan pemeringkatan atribut sangat penting untuk melihat atribut yang paling mempengaruhi kejadian *turnover*. Dengan performa AUC model WQRF sebesar 88,1% ditemukan bahwa tingginya *turnover* dipengaruhi oleh lembur yang meningkat, lama bekerja dan usia yang lebih muda, pendapatan bulanan yang rendah, persentase kenaikan gaji yang rendah dan jarak yang jauh dari rumah.

Penelitian selanjutnya yang dilakukan oleh Juvitayapun (2021) menyebutkan bahwa metode tradisional dalam mengidentifikasi faktor pendorong *turnover* sukarela seperti *exit interview*, tidak dapat mendeteksi alasan mendasar yang paling penting dan kemungkinan pengunduran diri karyawan. Dengan metode *modern* berupa pengembangan model untuk mendeteksi kemungkinan *turnover*, kekurangan pada metode tradisional tersebut dapat diperbaiki melalui. Hasil menunjukan bahwa metode SMOTE dan XGBoost memiliki kinerja terbaik dengan akurasi 98,03%, skor F1 90,79%, presisi 97,18% dan *recall* 85,19%. Atribut yang paling mempengaruhi *turnover* adalah kenaikan gaji di bawah rata-rata pasar dan pola penggunaan jam cuti tahunan pada dua bulan terakhir yang lebih banyak.

Penelitian keenam yang dilakukan oleh Ke et al. (2022) menggunakan model gabungan XGBoost-SMOTE untuk mengoptimalkan model numerik ramalan kualitas udara. Hasil menunjukan bahwa model XGBoost-SMOTE dapat mencapai hasil optimasi yang handal khususnya pada pemilihan *hyperparameter* dan atribut terbaik secara otomatis. Model XGBoost-SMOTE ini dapat menyeimbangkan proporsi kelas data yang tidak merata. Teknik SMOTE dapat memperbaiki kekurangan metode *random oversampling* yaitu banyaknya kemunculan sampel duplikat dalam *training set* yang berujung pada masalah *overfitting model*.

Khera & Divya (2019) dalam penelitiannya melakukan pengembangan model prediksi *turnover* karyawan, yang dapat memberikan kesempatan bagi organisasi untuk menangani masalah *turnover* dan meningkatkan retensi. Dengan akurasi model SVM yang digunakan sebesar 85%, atribut terpenting yang ditemukan adalah usia, jenis kelamin, status perkawinan, *job level, job profile, job role*, perjalanan bisnis dan atrisi. Penelitian mengatakan bahwa secara umum atribut yang mempengaruhi *turnover* adalah atribut dari segi demografis serta *job related and organizational*.

Penelitian kedelapan yang dilakukan oleh Kovvuri & Dommeti (2022) membandingkan algoritma *Logistic Regression*, *Naïve Bayes*, *Random Forest* dan XGBoost untuk memprediksi *turnover* karyawan. Dataset untuk melatih algoritma berasal dari *Kaggle* yang terdiri dari 4.653 baris data dan 8 atribut. Atribut tersebut adalah pendidikan, tahun bergabung, kota, tingkat gaji, usia, jenis kelamin, keterlibatan dalam proyek dan pengalaman di bidang saat ini. Algoritma XGBoost memberikan performa terbaik dibandingkan dengan algoritma lainnya, dengan nilai F1 dan kurva ROC secara berurutan mencapai 77% dan 88%.

Penelitian selanjutnya oleh Mardiansyah et al. (2021), menggabungkan pendekatan SMOTE untuk *imbalance data settlement*, dan dilanjutkan perhitungan model prediksinya dengan menggunakan algoritma XGBoost. Penelitian ini menggunakan empat data berbeda yang semuanya mengalami ketimpangan proporsi kelas data. Untuk melihat kekuatan metode SMOTEXGBoost, model yang dihasilkan oleh SMOTEXGBoost dibandingkan performanya berdasarkan kurva ROC-AUC dengan beberapa algoritma lainnya yaitu *Logistic Regression*, *Random Forest* dan XGBoost tunggal. Hasil menunjukan SMOTEXGBoost memiliki nilai AUC tertinggi di antara model lainnya untuk keempat data yang digunakan, yaitu sebesar 98,88%; *Winconsin* 99,93%; *Glass* 99,80% dan *E-coli* 99,40%.

Noviyanti (2018) dalam penelitiannya menyatakan bahwa prediksi *turnover* karyawan merupakan salah satu cara untuk mengetahui perkembangan perusahaan. Dampak negatif *turnover* yang diangkat dalam penelitian ini adalah penurunan kinerja, produktivitas hingga strategi organisasi. Percobaan dilakukan dengan menggunakan data yang berasal dari *Human Capital* sebuah perusahaan IT. Atribut yang menjadi prediktor *turnover* yaitu jenis kelamin, tanggal masuk karyawan, tanggal keluar karyawan dan status karyawan. Dalam penelitian ini disebutkan bahwa prediksi *turnover* karyawan dapat membantu perusahaan dalam mengembangkan perencanaan karyawan sehingga mendukung pencapaian target perusahaan.

Palupi (2021) dalam penelitiannya melakukan klasifikasi faktor tingginya *turnover* karyawan pada perusahaan IT. Pada data awal dilakukan eliminasi atribut sehingga atribut yang digunakan terdiri dari nama, usia, lama bekerja, pendidikan, kepuasan terhadap perusahaan, loyalitas dan KPI. Berdasarkan metrik evaluasi kinerja model klasifikasi, algoritma *Naïve Bayes* berbasis *Particle Swarm Optimization* (PSO) memiliki kinerja yang lebih baik dengan nilai akurasi 94,17%, presisi 89,80%, *recall* 95,65% dan AUC 0.96.

Selanjutnya, penelitian yang dilakukan oleh Sholikhati (2022) membandingkan kinerja algoritma XGBoost dan SMOTE-XGBoost menggunakan Python, untuk mengklasifikasikan kerentanan pasien terhadap penyakit *stroke*. SMOTE dipilih untuk penanganan *imbalance data*, karena lebih efektif dimana data sintetik dibuat relatif dekat dengan contoh yang ada di fitur kelas minoritas. Hasil menunjukan bahwa untuk kasus klasifikasi status *stroke* pasien, model SMOTE yang diintegrasikan meningkatkan kinerja XGBoost sebanyak 1%. Dimana nilai AUC untuk model XGBoost adalah 80,5% dan SMOTE-XGBoost adalah 81,5%.

Penelitian berikutnya oleh Syukron et al. (2020) membandingkan integrasi metode SMOTE terhadap dua algoritma yaitu RF dan XGBoost, pada kasus klasifikasi tingkat penyakit hepatitis C dengan data yang tidak seimbang. Parameter yang dilakukan *tuning* pada XGBoost adalah *learning rate, max depth, min child weight, gamma, colsample by tree* dan *n_estimators*. Hasil menunjukan bahwa baik SMOTE untuk RF maupun XGBoost, tidak ada perbedaan yang jauh dari segi nilai akurasi. Namun nilai *recall* meningkat dari yang sebelumnya tidak menggunakan SMOTE adalah 0% (RF) dan 0,65% (XGBoost), menjadi 75,55% (RF) dan 76,82% (XGBoost) setelah menggunakan SMOTE. Tingginya *recall* mengindikasikan bahwa model mampu memprediksi kelas minoritas dengan benar. Sehingga SMOTE mampu memperbaiki kinerja model, dimana model dapat memprediksi secara akurat pada semua kelas respon.

Tao et al. (2021) dalam penelitiannya membangun *benchmark model* yang membandingkan algoritma XGBoost dengan ID3 *decision tree*, CART *decision tree, AdaBoost, Random Forest* dan *Gradient Boosting Decision Tree* (GBDT). Hasil menunjukan bahwa XGBoost unggul dibandingkan model *benchmark* lainnya. Perlakuan *feature selection* dalam penelitian ini memberikan peningkatan pada performa model, sehingga dinilai sangat membantu dalam memprediksi perilaku *turnover*. Secara berurutan nilai AUC dan *recall* XGBoost yang diperoleh sebelum *feature selection* adalah 0,9082 dan 0,8426. Sedangkan setelah *feature selection* adalah 0,9224 dan 0,8487. *Key feature* berdasarkan model XGBoost adalah usia, status perkawinan, total waktu bekerja, total perusahaan yang pernah ditempati, kepuasan terhadap lingkungan, lama bekerja, kepuasan kerja dan lama bersama manajer saat ini.

Penelitian oleh Tharani & Raj (2020) melakukan prediksi *turnover* karyawan pada sebuah perusahaan IT dan ITES dengan total data 416 karyawan. Penelitian menyebutkan bahwa dengan adanya perumusan strategi untuk mengurangi *turnover* pekerja, maka perusahaan akan memiliki keunggulan kompetitif dibandingkan dengan organisasi lain. Metode semacam

penelitian ini dapat membantu manajemen dalam memahami faktor-faktor pengaruh *turnover* karyawan. Diantara beberapa algoritma lainnya (LR, NB, *Kernel* SVM, KNN, RF, *Artificial Neural Network*), algoritma XGBoost menunjukan kinerja yang relatif lebih baik dengan nilai akurasi, *recall*, presisi dan skor F yang tinggi. Atribut yang terbukti mempengaruhi *turnover* dalam kasus penelitian ini adalah pendidikan, peluang pekerjaan alternatif, jenis kelamin, kesediaan pindah tempat kerja, stres kerja, sikap terhadap COVID.

Penelitian Zhao et al. (2019) membandingkan kinerja 10 algoritma *machine learning* dalam 3 ukuran data berbeda pada kasus *turnover* karyawan. Secara umum, kinerja terbaik diperoleh dari pengklasifikasi berbasis pohon (XGBoost, GBT, RF, DT), dengan XGBoost sebagai pengklasifikasi terbaik. Untuk ukuran data kecil yaitu 100 data yang berasal dari *bank data*, XGBoost berkinerja dengan baik berdasarkan nilai metrik presisi, F1 dan ROC. XGBoost yang menggunakan pendekatan *ensemble* mampu mengurangi ketidakstabilan pohon dan kemungkinan bias, serta meningkatkan kemampuan prediksi dengan waktu *running* yang lebih cepat daripada GBT. Berdasarkan *ensemble method* (GBT, XGB, RF), tiga atribut terpenting yang mempengaruhi *turnover* adalah kenaikan gaji, lama bekerja dan usia.

Dari literatur-literatur di atas, terlihat bahwa kasus *turnover* karyawan penting untuk diteliti mengingat dampak dan manfaat yang dirasakan dari pengelolaannya. Penelitian ini melakukan klasifikasi menggunakan algoritma *machine learning* XGBoost, yang telah terbukti memiliki kinerja baik untuk beberapa kasus termasuk *turnover* karyawan. Atribut yang digunakan pada penelitian ini merujuk pada atribut-atribut yang sudah pernah digunakan dan terbukti berpengaruh terhadap kejadian *turnover* karyawan pada penelitian sebelumnya. Serta menggunakan atribut tambahan yang disesuaikan dengan industri pertambangan dan permintaan perusahaan tempat dilakukannya penelitian. Karena data yang dimiliki mengandung ketidakseimbangan kelas data, maka metode SMOTE dipilih untuk menangani masalah tersebut. Pemaksimalan kinerja model diusahakan melalui proses seleksi atribut dan *hyperparameter tuning*. *Hyperparameter tuning* dilakukan secara otomatis menggunakan metode *RandomizedSearchCV* untuk mengoptimalkan proses menemukan *hyperparameter* yang terbaik. Proses melihat *features importance* dilakukan untuk melihat atribut apa saja yang paling mempengaruhi kejadian *turnover* di divisi terkait.

2.2 Research Gap

Berikut merupakan perbedaan penelitian yang saat ini dilakukan dengan penelitian terdahulu.

Tabel 2.1 *Research Gap*

Penelitian	***Employee Turnover***	**Industri Pertambangan**	**Klasifikasi**	**XGBoost**	***Imbalance Data***	**SMOTE**
Chanodkar et al. (2019)	✓		✓			
Duan (2022)	✓		✓	✓		
Farizi & Tanuwijaya (2022)	✓	✓				
Gao et al. (2019)	✓		✓		✓	
Juvitayapun (2021)	✓		✓	✓	✓	✓
Ke et al. (2022)			✓	✓	✓	✓
Khera & Divya (2019)	✓		✓			
Kovvuri & Dommeti (2022)	✓		✓	✓	✓	
Mardiansyah et al. (2021)			✓	✓	✓	✓
Noviyanti (2018)	✓		✓			
Palupi (2021)	✓		✓			
Sholikhati (2022)			✓	✓	✓	✓
Syukron et al. (2020)			✓	✓	✓	✓
Tao et al. (2021)	✓		✓	✓	✓	✓
Tharani & Raj (2020)	✓		✓	✓		
Zhao et al. (2019)	✓		✓	✓		
Usulan	**✓**	**✓**	**✓**	**✓**	**✓**	**✓**

Keterbaruan penelitian yang akan dilakukan dapat dilihat dari tujuan, bidang industri dari subjek yang diteliti, serta atribut dan sumber data yang digunakan. Penelitian ini bertujuan melakukan klasifikasi kejadian *turnover* pada karyawan industri pertambangan, tepatnya perusahaan jasa pertambangan. Berdasarkan Tabel 2.1, diketahui bahwa penelitian yang spesifik membahas klasifikasi *turnover* karyawan di industri pertambangan masih kurang. Hal ini bisa jadi disebabkan oleh kerahasiaan data, hingga variasi kondisi kerja dan bidang/jenis pekerjaan di industri pertambangan. Meskipun terdapat beberapa penelitian yang mengkaji terkait *turnover* karyawan pertambangan, namun penelitian-penelitian tersebut lebih berfokus pada analisis faktor-faktor yang mempengaruhi *turnover* karyawan daripada melakukan prediksi *turnover* karyawan dengan menggunakan metode *machine learning*. Beberapa atribut terdahulu digunakan, dengan juga mempertimbangkan atribut yang spesifik dalam industri pertambangan yaitu *Site*, *Assignment Letter* dan *Production Plan*. Data penelitian ini bersumber langsung dari *bank data* Divisi Human Capital & Talent Development PT. PNR.

2.3 Landasan Teori

2.3.1 *Turnover* Karyawan

Turnover karyawan adalah perputaran karyawan sebagai kepergian modal intelektual dari organisasi pemberi kerja pada periode tertentu (Palupi, 2021; Prawitasari, 2016; Punnoose & Ajit, 2016). Berdasarkan inisiatif atau keputusan, *turnover* karyawan dibagi menjadi dua tipe yaitu sukarela dan tidak sukarela (sudah direncanakan) (White, 2022). Dikatakan *turnover* sukarela apabila seorang karyawan memutuskan untuk meninggalkan organisasi. Sedangkan *turnover* tidak sukarela adalah ketika suatu organisasi memutuskan untuk mengeluarkan karyawan dari posisi saat ini (Chhinzer, 2021).

Turnover karyawan adalah perilaku aktual dari niat atau keinginan pindah dari pekerjaan secara sukarela, yang biasa disebut dengan *turnover intention* (Samson & Suliystiorini, 2020). Berikut ini merupakan beberapa dampak negatif dari fenomena *turnover* karyawan (Chiat & Panatik, 2019; Noviyanti, 2018; Palupi, 2021; Prawitasari, 2016; Punnoose & Ajit, 2016).

1. Tertundanya proyek dan target sehingga tidak selesai sesuai waktunya.
2. Pembubaran tim.
3. Kekurangan SDM (terutama pada posisi/divisi dengan *turnover* tinggi).
4. Kesulitan melakukan perekrutan untuk mencari kandidat dengan berbagai kriteria dalam waktu singkat.
5. Memakan waktu dan biaya yang lebih.
6. Gangguan dalam produktivitas tempat kerja, moral dan strategi pertumbuhan jangka panjang.

Kemudian dengan mengelola (mengendalikan dan mengurangi) *turnover*, dapat secara efektif memberikan manfaat-manfaat sebagai berikut (Lee & Liu, 2021; Palupi, 2021; Prawitasari, 2016; Punnoose & Ajit, 2016).

1. Meningkatkan kinerja karyawan dan komitmen karyawan terhadap perusahaan.
2. Penghematan waktu dan biaya perekrutan.
3. Memperpanjang lama bekerja (*years of service*) karyawan.
4. Pencapaian target yang tepat waktu hingga pencapaian tujuan yang sesuai.

2.3.2 Industri dan Jasa Pertambangan

Merujuk pada UU Nomor 4 Tahun 2009 tentang Pertambangan Mineral dan Batubara, pertambangan adalah sebagian atau seluruh tahapan kegiatan dalam rangka penelitian, pengelolaan dan pengusahaan mineral atau batubara yang meliputi penyelidikan umum, eksplorasi, studi kelayakan, konstruksi, penambangan, pengolahan dan pemurnian, pengangkutan dan penjualan, serta kegiatan pascatambang. Sedangkan jasa pertambangan menurut Peraturan Pemerintah (PP) Nomor 96 Tahun 2021 tentang Pelaksanaan Kegiatan Usaha Pertambangan Mineral dan Batubara didefinisikan sebagai jasa penunjang yang berkaitan dengan kegiatan usaha pertambangan.

Dapat diketahui bahwa jasa pertambangan merupakan bagian integral dari industri pertambangan. Dimana jasa pertambangan adalah sektor yang menyediakan layanan penunjang, yang berkaitan dengan kegiatan usaha pertambangan seperti penelitian, pengelolaan, dan pengusahaan mineral atau batubara. Industri pertambangan merupakan salah satu bidang industri yang keberhasilan dalam organisasinya ditentukan oleh karyawan (Prabowo, 2019). Sehingga efisiensi SDM dan manajemennya menjadi penting, untuk mendapatkan keuntungan di pasar yang kompetitif (Anwar & Abdullah, 2021; Collins, 2020).

2.3.3 *Data Mining* dan Klasifikasi

Data mining merupakan proses penggalian informasi yang berharga dan berguna di dalam *database* (Hajar et al., 2020). *Data mining* merupakan salah satu tahapan dalam keseluruhan proses *Knowledge Discovery in Database* (KDD), dimana KDD adalah proses menentukan informasi yang berguna dalam data (Setio et al., 2020). Fungsi *data mining* dibagi menjadi dua kategori yaitu fungsi mayor atau fungsi utama yang meliputi fungsi klasifikasi, pengelompokan dan asosiasi, serta fungsi minor atau fungsi tambahan yaitu meliputi fungsi deskripsi, estimasi, dan prediksi (Effendi & Rahmawati, 2018).

Penelitian ini memanfaatkan fungsi klasifikasi, yang merupakan proses menemukan model yang menjelaskan atau membedakan kelas data, dan dapat memperkirakan (prediksi) kelas dari suatu objek yang labelnya tidak diketahui (Febriani & Sulistiani, 2021). Terdapat empat komponen dasar dari proses klasifikasi (Novianti, 2019) , yaitu:

1. Kelas atau label kelas, adalah variabel dependen yang berupa kategori dan menjelaskan sebuah label dari objek setelah proses klasifikasi.

2. Prediktor atau atribut, merupakan variabel independen yang mewakili karakteristik data.
3. *Training set*, adalah data pelatihan yang merupakan kumpulan data berisi nilai-nilai untuk dua komponen sebelumnya (kelas dan prediktor/atribut) dan digunakan untuk pembangunan model.
4. *Testing set*, merupakan data pengujian yang akan diklasifikasikan oleh model dan berguna untuk mengukur atau mengevaluasi tingkat kinerja model.

Bidang *data mining* cukup banyak didukung cabang ilmu lain dalam teknologi informasi seperti statistik, *machine learning*, visualisasi data dan lainnya (Febriani & Sulistiani, 2021).

2.3.4 *Machine Learning*

Machine learning digunakan dalam teknik *data mining* untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis (Munti et al., 2018). Menurut Wahyono (2018), *machine learning* merupakan salah satu cabang dari ilmu *Artificial Intelligence*, khususnya yang mempelajari tentang bagaimana komputer mampu belajar dari data untuk meningkatkan kecerdasannya. Diantara beragamnya solusi pengendalian *turnover*, penggunaan *machine learning* lebih dipilih oleh organisasi untuk memprediksi kejadiannya (Chanodkar et al., 2019).

2.3.5 XGBoost

Menurut Syukron et al. (2020) dan Tim Datasans (2023), *Extreme Gradient Boosting* atau XGBoost adalah salah satu metode dalam *machine learning* yang tergolong ke dalam *ensemble method*. *Ensemble method* adalah teknik pembelajaran mesin yang menggabungkan beberapa model dasar untuk menghasilkan prediksi yang lebih akurat dan stabil. XGBoost menggunakan salah satu konsep dari *ensemble method* yang disebut dengan *boosting*. Konsep *boosting* melakukan pembangunan beberapa pohon melalui perbaikan kesalahan yang dilakukan oleh model sebelumnya dengan memberikan bobot lebih pada data yang salah diklasifikasikan.

Proses optimasi (menjadikan model seakurat mungkin dalam memprediksi data pelatihan) pada algoritma XGBoost selama pelatihan model dilakukan dengan meminimumkan fungsi objektif. Fungsi objektif adalah ukuran seberapa baik model dalam memprediksi hasil yang benar. Perlu diperhatikan bahwa “meminimumkan fungsi objektif” memiliki arti meminimumkan kesalahan, atau memaksimalkan kebaikan model.

Fungsi objektif XGBoost bisa disebut sebagai *loss function*, dimana nilai *loss function* yang tinggi menandakan bahwa model yang dihasilkan sangat buruk dan berlaku sebaliknya (Syahrani, 2019). Fungsi objektif memiliki dua bagian yaitu fungsi kerugian (L) yang mengukur seberapa jauh prediksi model dari nilai sebenarnya, dan fungsi regularisasi (Ω) yang mengukur kompleksitas model. Sehingga fungsi objektif dapat ditulis dalam persamaan (2.1).

$$Obj = L + \Omega \tag{2.1}$$

$$L = -y \cdot \log(\hat{y}) - (1 - y) \cdot log(1 - \hat{y}) \tag{2.2}$$

$$\Omega = \gamma \mathrm{T} + \lambda \sum W_j{}^2 \tag{2.3}$$

Untuk klasifikasi biner, biasanya menggunakan fungsi kerugian *log-loss* yang ditunjukan persamaan (2.2). Dimana y adalah label sebenarnya (0 atau 1) dan $\hat{y}$ adalah probabilitas prediksi model. Fungsi regularisasi XGBoost berdasarkan persamaan (2.3) memiliki dua bagian. Bagian pertama $\gamma \mathrm{T}$ (γ adalah parameter gamma dan T adalah jumlah daun dalam pohon) bertujuan mengontrol ukuran pohon dan mencegah model terlalu kompleks. Bagian kedua $\lambda \sum W_j{}^2$ (λ adalah parameter *reg_lambda* dan W_j adalah bobot daun) bertujuan mengontrol bobot daun dan mencegah bobot terlalu besar. Kedua bagian fungsi regularisasi tersebut memberikan kemampuan kepada XGBoost dalam mencegah terjadinya *overfitting* pada model, dan dengan demikian meningkatkan kemampuan generalisasi model. Berikut contoh sederhana perhitungan fungsi objektif dalam XGBoost. Diberikan empat baris data yang dapat dilihat pada Tabel 2.2, dan parameternya adalah pohon dengan 4 daun, *gamma* = 0,1 dan *reg_lambda* = 0.

Tabel 2.2 Data Percobaan Perhitungan Fungsi Objektif XGBoost

No	$\boldsymbol{y}$	$\boldsymbol{\hat{y}}$
1	0	0,1
2	1	0,9
3	0	0,2
4	1	0,8

1. Fungsi kerugian (L) untuk tiap baris data.
   a. $L1 = -0 \cdot \log(0{,}1) - (1 - 0) \cdot log(1 - 0{,}1) = 0{,}105$
   b. $L2 = -1 \cdot \log(0{,}9) - (1 - 1) \cdot log(1 - 0{,}9) = 0{,}105$

c. $L3 = -0 \cdot \log(0{,}2) - (1-0) \cdot log(1-0{,}2) = 0{,}223$

d. $L4 = -1 \cdot \log(0{,}8) - (1-1) \cdot log(1-0{,}8) = 0{,}223$

Fungsi kerugian keseluruhan adalah rata-rata dari fungsi kerugian seluruh baris data, maka $L = \frac{L1+L2+L3+L4}{4} = \frac{0{,}105+0{,}105+0{,}223+0{,}223}{4} = \frac{0{,}656}{4} = 0{,}164$.

2. Fungsi regularisasi (Ω).
   a. Bagian pertama $\gamma T = 0{,}1 \cdot 4 = 0{,}4$
   b. Bagian kedua $\lambda \sum {W_j}^2$, walaupun parameter bobot daun tidak diketahui nilainya, namun diketahui bahwa *reg_lambda* (λ) = 0. Karena apapun yang dikalikan dengan nol adalah nol, maka bagian kedua dari fungsi regularisasi adalah 0. Maka diperoleh fungsi regularisasi $\Omega = \gamma T + \lambda \sum {W_j}^2 = 0{,}4 + 0 = 0{,}4$.

Setelah mengetahui nilai dari fungsi kerugian dan fungsi regularisasi, dapat diketahui fungsi objektif yaitu $Obj = L + \Omega = 0{,}164 + 0{,}4 = 0{,}564$. Berdasarkan Gajendra (2022) dan Dayananda dalam Rachmi (2020), secara sederhana penyusunan algoritma XGBoost terdiri atas beberapa tahapan berikut.

1. Input data dan penentuan parameter-parameter yang digunakan.

   Contoh terdapat *dataset* pada Tabel 2.3 yang terdiri dari variabel independen (X) *salary* dan *credit*, variabel dependen (Y) *approval*, 7 baris data, dan beberapa parameter pada Tabel 2.4 yang digunakan. Untuk variabel *credit* yang memiliki tiga kategori *instance*, B memiliki arti *"Bad"*, G adalah *"Good"* dan N adalah *"Normal"*.

Tabel 2.3 Data Percobaan Penyusunan XGBoost

Salary	***Credit***	***Approval***
≤ 50	B	0
≤ 50	G	1
≤ 50	G	1
> 50	B	0
> 50	G	1
> 50	N	1
≤ 50	N	0

Tabel 2.4 Parameter Percobaan Penyusunan XGBoost

Parameter	*max_depth*	*learning_rate*	*gamma*	*base_score*	*reg_lambda*
Nilai	2	0,1	1	0,5	0

2. Prediksi awal.

   Karena data contoh di atas hanya memiliki dua label kelas yaitu 0 dan 1, maka parameter *base_score* sebesar 0,5 digunakan sebagai nilai prediksi atau probabilitas awal (*P*) untuk seluruh titik data dalam *dataset*.

3. Perhitungan *residuals* atau *error*

   Residuals dengan lambang $\hat{Y}$ dihitung pada semua titik data dari prediksi sebelumnya. Hasil perhitungan *residuals* pada Tabel 2.5 didapatkan melalui persamaan (2.4).

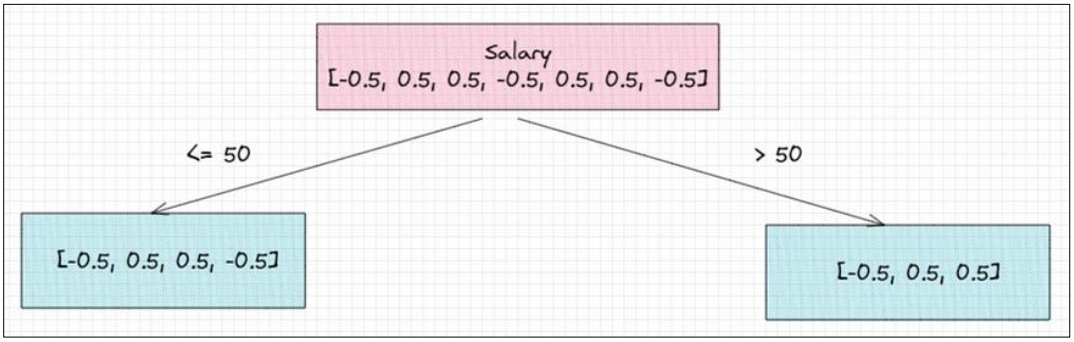
$$\hat{Y} = Prediksi\ Aktual - Prediksi\ Awal \tag{2.4}$$

Tabel 2.5 Perhitungan *Residuals* Awal

Salary	***Credit***	***Approval*** **(Prediksi Aktual)**	**P1** **(Prediksi Awal)**	$\hat{Y}$ **(Residual)**
≤ 50	B	0	0,5	-0,5
≤ 50	G	1	0,5	0,5
≤ 50	G	1	0,5	0,5
> 50	B	0	0,5	-0,5
> 50	G	1	0,5	0,5
> 50	N	1	0,5	0,5
≤ 50	N	0	0,5	-0,5

4. Pembangunan model

   a. Pemisahan/*splitting* data menjadi dua bagian. Pohon pada XGBoost harus menjadi pohon dengan keputusan biner (dua) seperti pada Gambar 2.1. Karena variabel *salary* memiliki dua kategori *instance*, maka variabel ini akan dipisah terlebih dahulu. Nilai yang ada pada kotak *leaf node* merupakan nilai residual masing-masing *node*.



Gambar 2.1 *Splitting* berdasarkan *Salary*

Sumber: Gajendra (2022)

b. Perhitungan nilai *similarity* dan *gain*. Perhitungan nilai *gain* dengan persamaan (2.5) hanya diterapkan pada *root* pohon, sedangkan perhitungan nilai *similarity* (2.6) diterapkan di semua node. Hasil yang didapatkan terlihat pada Gambar 2.2.

$$Gain\ Score = \left(Left_{similarity} + Right_{similarity}\right) - Root_{similarity} \quad (2.5)$$

$$Similarity\ Score = \frac{(\sum \hat{Y}_i)^2}{\sum[Previous\ P \cdot (1 - Previous\ P)] + \lambda} \quad (2.6)$$

Keterangan persamaan:

- $\hat{Y}_i$ = Residual ke-i
- λ = *reg_lambda* (berkaitan dengan pemangkasan *node*)
- $Previous\ P$ = Probabilitas sebelumnya

Berikut merupakan perhitungannya.

$Left_{similarity} =$

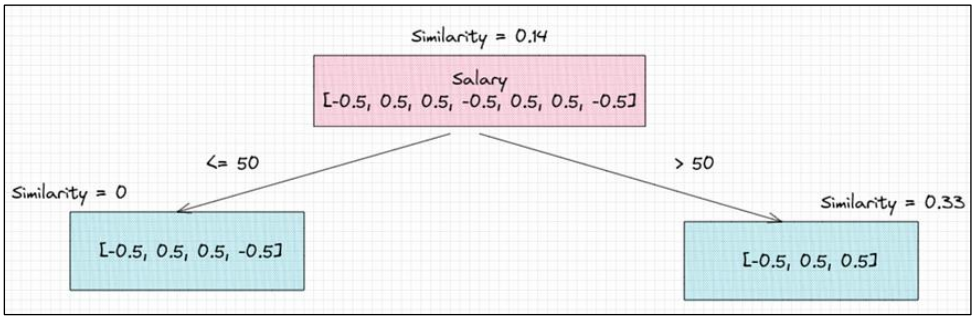
$$\frac{(-0{,}5 + 0{,}5 + 0{,}5 - 0{,}5)^2}{\left[\left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right)\right] + 0} = 0$$

$Right_{similarity} =$

$$\frac{(-0{,}5 + 0{,}5 + 0{,}5)^2}{\left[\left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right)\right] + 0} = 0{,}33$$

$Root_{similarity} = 0{,}14$

$Gain\ Score = (0 + 0{,}33) - 0{,}14 = 0{,}21$



Gambar 2.2 *Similarity* dan *Gain* pada *Splitting* Pertama

Sumber: Gajendra (2022)

c. Pemisahan kembali pada pohon untuk memperluas pohon keputusan. Disini akan digunakan variabel yang belum dilakukan *splitting* yaitu *credit*. Namun karena variabel *credit* memiliki tiga kategori *instance* sementara pohon XGBoost harus menjadi pohon dengan keputusan biner, maka pemisahan pohon akan dicoba berdasarkan beberapa kombinasi variabel *credit*. Pemisahan yang dipilih adalah yang memiliki nilai *gain* tertinggi. Berikut contoh perluasan pada *left node*.

• Pemisahan berdasarkan kombinasi variabel *credit "Bad – Good* dan *Normal"*, dengan hasil yang dapat dilihat pada Gambar 2.3.

$$Left_{similarity} =$$

$$\frac{(-0{,}5)^2}{\left[\left(0{,}5 \cdot (1-0{,}5)\right)\right] + 0} = 1$$

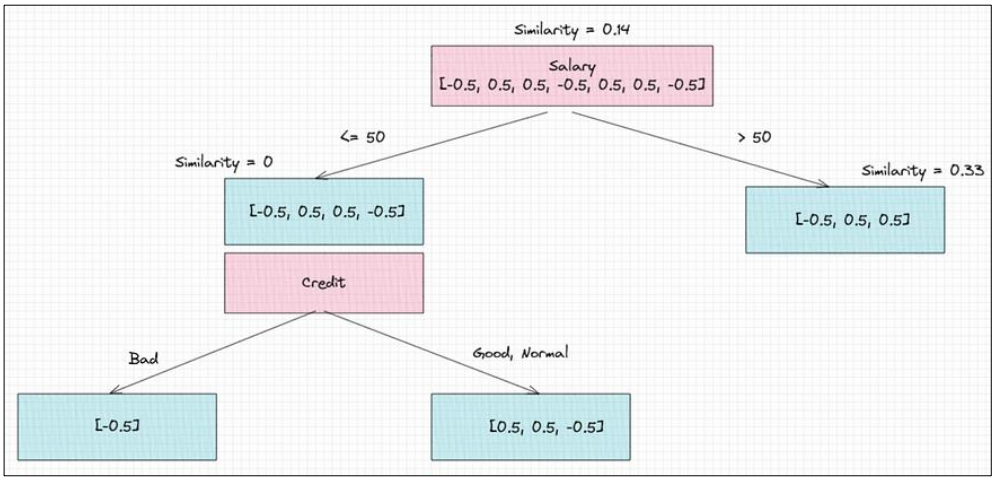
$$Right_{similarity} =$$

$$\frac{(0{,}5 + 0{,}5 - 0{,}5)^2}{\left[\left(0{,}5 \cdot (1-0{,}5)\right) + \left(0{,}5 \cdot (1-0{,}5)\right) + \left(0{,}5 \cdot (1-0{,}5)\right)\right] + 0} = 0{,}33$$

$$Root_{similarity} =$$

$$\frac{(-0{,}5 + 0{,}5 + 0{,}5 - 0{,}5)^2}{\left[\left(0{,}5 \cdot (1-0{,}5)\right) + \left(0{,}5 \cdot (1-0{,}5)\right) + \left(0{,}5 \cdot (1-0{,}5)\right) + \left(0{,}5 \cdot (1-0{,}5)\right)\right] + 0} = 0$$

$$Gain\ Score = (1 + 0{,}33) - 0 = 1{,}33$$



Gambar 2.3 *Splitting* berdasarkan *Credit* (*Bad – Good* dan *Normal*)

Sumber: Gajendra (2022)

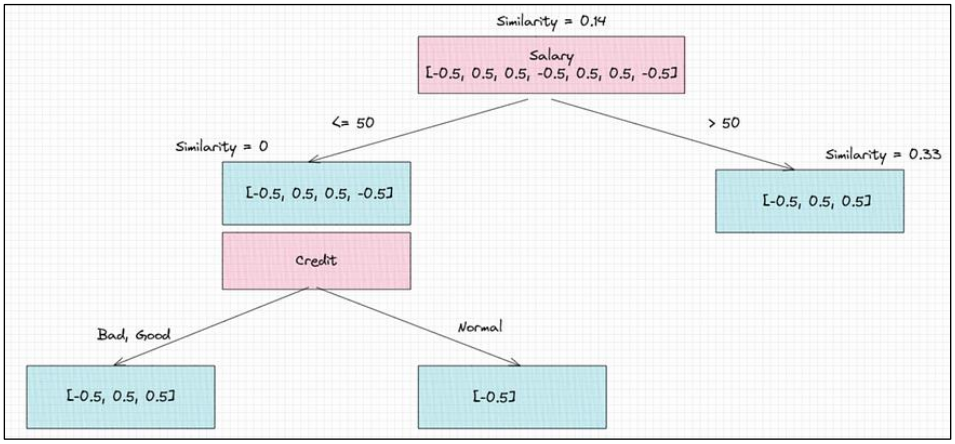
• Pemisahan berdasarkan kombinasi variabel *credit "Bad* dan *Good – Normal"*, dengan hasil yang dapat dilihat pada Gambar 2.4.

$$Left_{similarity} = \frac{(-0{,}5 + 0{,}5 + 0{,}5)^2}{\left[\left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right)\right] + 0} = 0{,}33$$

$$Right_{similarity} = \frac{(-0{,}5)^2}{\left[\left(0{,}5 \cdot (1 - 0{,}5)\right)\right] + 0} = 1$$

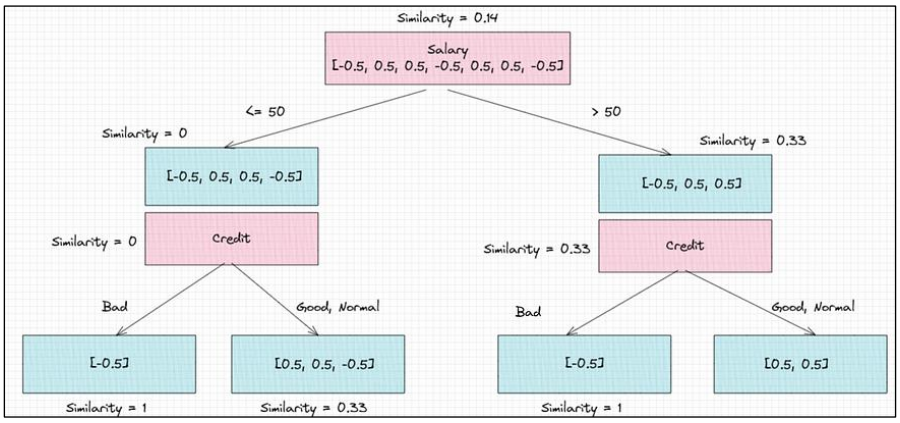
$$Root_{similarity} = \frac{(-0{,}5 + 0{,}5 + 0{,}5 - 0{,}5)^2}{\left[\left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right) + \left(0{,}5 \cdot (1 - 0{,}5)\right)\right] + 0} = 0$$

$$Gain\ Score = (1 + 0{,}33) - 0 = 1{,}33$$



Gambar 2.4 *Splitting* berdasarkan *Credit* (*Bad* dan *Good – Normal*)
Sumber: Gajendra (2022)

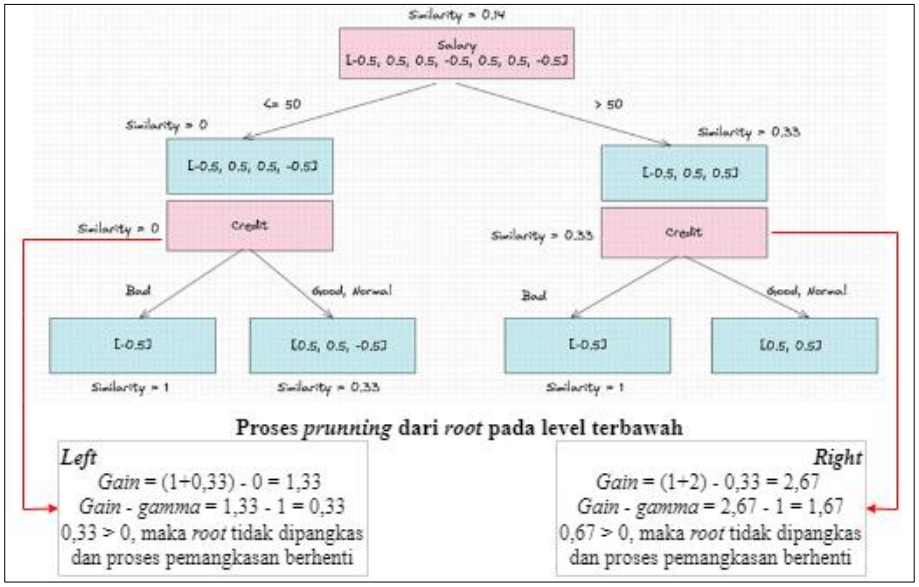
Pemisahan berdasarkan kombinasi variabel *credit* "*Bad* dan *Normal – Good*" juga tentunya perlu dilakukan. Namun untuk mempersingkat, pada contoh ini akan dilewati dan diasumsikan bahwa pemisahan yang dipilih adalah pemisahan yang berdasarkan kombinasi variabel *credit "Bad – Good* dan *Normal"*. Hal serupa juga dilakukan pada *right node*, sehingga didapatkan perluasan pohon keputusan seperti yang terlihat pada Gambar 2.5.



Gambar 2.5 Pohon Keputusan setelah Diperluas

Sumber: Gajendra (2022)

d. Pemangkasan/*pruning* melalui eliminasi bagian pohon yang memiliki kekuatan klasifikasi yang kurang, dengan melihat selisih antara nilai *gain* dengan *gamma*. Pemangkasan ini dilakukan mulai dari *root* yang berada di level terbawah, dengan aturan pemangkasan yaitu jika selisih *gain* dan *gamma* kurang dari nol, maka *root* tersebut dipangkas. Namun jika lebih dari nol, maka *root* tersebut tidak dipangkas dan pemangkasan akan berhenti. Hasil pemangkasan terlihat pada Gambar 2.6.



Gambar 2.6 Pemangkasan pada Pohon Keputusan

Sumber: Gajendra (2022)

e. Perhitungan nilai *output* untuk *base model*. *Base model output* dihitung, karena akan menjadi salah satu nilai yang digunakan dalam menghitung probabilitas baru dari setiap titik data. Nilai *base model output* diketahui dengan mengkonversi probabilitas awal (P) ke *odds* dengan persamaan (2.7).

$$log(odds) = log\left(\frac{P}{1-P}\right) \tag{2.7}$$

$log(odds) = log\left(\frac{0,5}{1-0,5}\right) = log\left(\frac{0,5}{0,5}\right) = 0$ (nilai *output* P awal 0,5 adalah 0)

f. Perhitungan probabilitas dan residual baru dari setiap titik data. Untuk menghitung nilai probabilitas baru tiap data, digunakanlah persamaan (2.8) dimana σ adalah *learning rate*. Persamaan (2.9) menunjukan fungsi sigmoid yang digunakan.

$$P_n = \sigma(Base\ model + \alpha \times Node\ Similarity) \tag{2.8}$$

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

Contoh perhitungan untuk data pada baris pertama (Tabel 2.6), yang pada pohon keputusan memiliki nilai *node similarity* sebesar 1. Dengan nilai *learning_rate* = 0,1 maka perhitungan probabilitas dan residual barunya adalah sebagai berikut.

$P_2 = \sigma(0 + 0,1 \times 1) = \sigma(0,1)$

$\sigma(0,1) = \frac{1}{1+\mathrm{e}^{(-0,1)}} \approx 0,525$ (pada sumber dibulatkan menjadi 0,6)

$\hat{Y}_2 = Prediksi\ Aktual - Prediksi\ Awal = 0 - 0,6 = -0,6$

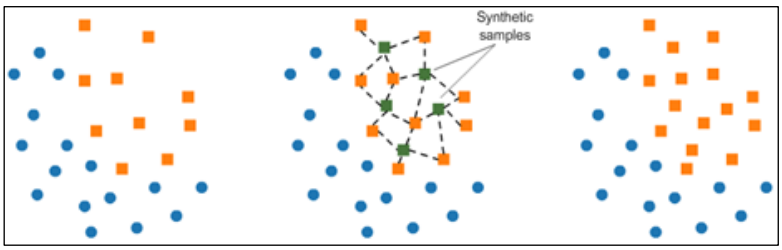
Tabel 2.6 Probabilitas dan Residual Baru dari Data Baris Pertama

Salary	**Credit**	***Approval*** **(Prediksi Aktual)**	**P1 (Prediksi Awal)**	$\hat{Y}_1$ **(Residual Awal)**	**P2 (Prediksi Baru)**	$\hat{Y}_2$ **(Residual Baru)**
≤ 50	B	0	0,5	-0,5	0,6	-0,6

5. Pengulangan langkah keempat untuk membangun pohon lain.

2.3.6 *Imbalance Data* dan SMOTE

Imbalance data atau tidak seimbangnya proporsi label kelas data adalah salah satu masalah yang umum terjadi dalam analisis klasifikasi. Masalah ini dapat mempengaruhi kinerja dari model dalam memprediksi kelas yang minoritas. Untuk menyeimbangkan proporsi label kelas, terdapat beberapa metode yang salah satunya adalah *Synthetic Minority Oversampling Technique* atau SMOTE. Teknik SMOTE diterapkan pada *training set*, dan digunakan untuk merekonstruksi sampel minoritas (*oversampling*) berdasarkan *nearest neighbors* seperti yang terlihat pada Gambar 2.7, agar kinerja peramalan meningkat dan menghindari terjadinya *overfitting* pada model yang dihasilkan (Ke et al., 2022; Syukron et al., 2020). Berdasarkan cara kerjanya, SMOTE dinilai lebih efektif dalam mengurangi ketimpangan data sampel (Sholikhati, 2022; Tao et al., 2021). SMOTE terbukti mampu memperbaiki kinerja model, dimana model dapat memprediksi secara akurat pada semua kelas respon (Syukron et al., 2020).



Gambar 2.7 Ilustrasi Cara Kerja SMOTE

Sumber: Sampurna (2021)

2.3.7 *Hyperparameter Tuning*

Hyperparameter tuning merupakan tahapan penyesuaian parameter eksternal (*hyperparameter*) dari model pembelajaran mesin yang berguna untuk meningkatkan kinerja model (Juvitayapun, 2021; Sholikhati, 2022; Tao et al., 2021). Biasanya nilai optimal *hyperparameter* untuk menghasilkan model terbaik ini tidak diketahui. Sehingga model diperintahkan untuk menjelajahi dan memilih arsitektur model yang optimal secara otomatis. Salah satu algoritma yang dapat digunakan untuk mencari parameter terbaik secara otomatis adalah *Randomized Search Cross Validation* (*RandomizedSearchCV*). Algoritma ini melakukan validasi beberapa model dan *hyperparameter*, dengan melakukan pencarian acak berdasarkan ruang parameter yang telah ditentukan (tidak mencoba semua nilai). Proses pencarian acak akan berhenti ketika jumlah *loop* sudah sesuai dengan iterasi yang diinginkan (Maghfiroh et al., 2023).

RandomizedSearchCV terbukti dapat menemukan parameter terbaik untuk algoritma klasifikasi yang digunakan, serta memiliki kelebihan berupa efisiensi komputasi, kecepatan dan penanganan ruang *hyperparameter* yang besar. Pada kasus klasifikasi XGBoost dengan *dataset* kecil, penelitian oleh Shafila (2020) yang menggunakan 59 sampel dan 29 variabel, telah membuktikan bahwa proses *hyperparameter tuning* ini dapat meningkatkan performa model. Parameter yang digunakan untuk klasifikasi dengan XGBoost pada penelitian ini dapat dilihat pada Tabel 2.7 Parameter Klasifikasi.

Tabel 2.7 Parameter Klasifikasi

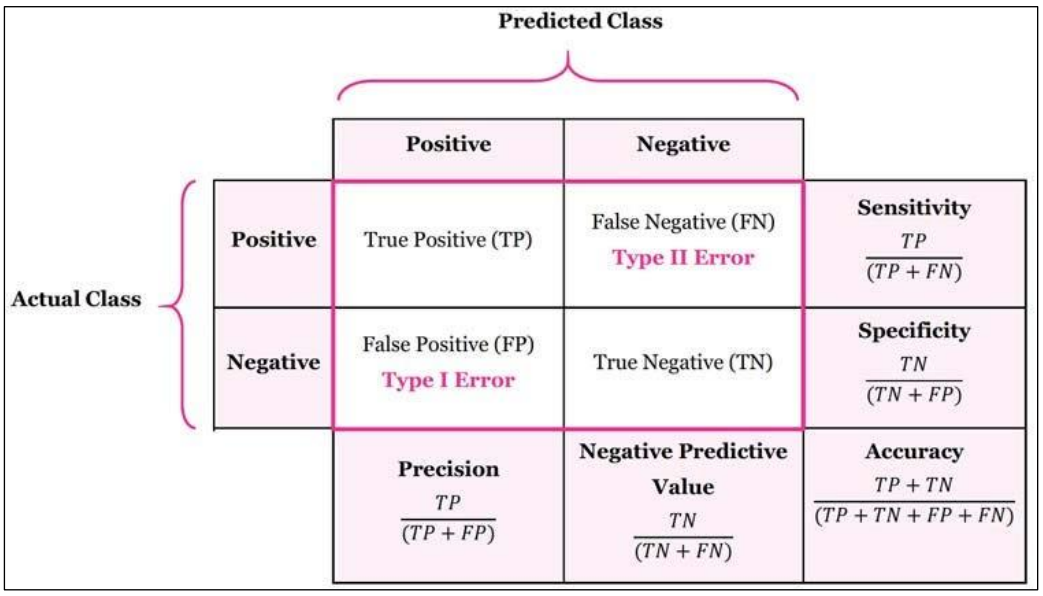
Parameter	Keterangan	Nilai Parameter
reg_alpha	Fungsi regularisasi L1 untuk mengatur kompleksitas model. Semakin besar nilainya, maka model semakin konservatif dan dapat mencegah *overfitting*.	0 sampai 1
reg_lambda	Fungsi regularisasi L2 untuk mengatur kompleksitas model. Semakin besar nilainya, maka model semakin konservatif.	0 sampai 1
max_leaves	Jumlah daun maksimum dalam pohon. Semakin besar nilainya, maka memungkinkan model untuk belajar struktur yang lebih kompleks.	2 sampai 10
scale_pos_ weight	Digunakan untuk menangani ketidakseimbangan kelas. Pengaturan *(len(y_train) - y_train.sum())/y_train.sum()* digunakan untuk menghitung rasio antara jumlah sampel negatif dan positif dalam data latih, untuk penyeimbangan data.	*(len(y_train) - y_train.sum()) / y_train.sum()*
n_estimators	Jumlah pohon yang akan dibuat. Nilai yang lebih besar dapat menangkap lebih banyak detail.	100, 200, 300
max_depth	Kedalaman maksimum dari pohon, yang pada gilirannya mempengaruhi jumlah daun dalam pohon. Nilai yang lebih besar memungkinkan model untuk belajar struktur yang lebih kompleks. Namun, pohon yang lebih dalam juga lebih mungkin *overfitting*.	4, 5, 6, 7, 8
min_child_ weight	Nilai minimum bobot yang diperlukan untuk membuat node baru. Nilai yang lebih besar membuat model lebih konservatif.	0, 1, 2, 3, 4, 5, 6, 7
learning_rate	Kecepatan belajar model. Nilai yang lebih kecil membuat model belajar lebih lambat dan menurunkan resiko *overfitting*.	0.01, 0.025, 0.05, 0.1, 0.2, 0.3
gamma	Nilai minimum penurunan *loss* yang diperlukan untuk membuat *split* baru. Nilai yang lebih besar membuat model lebih konservatif.	0, 0.1, 0.2, 0.3, 0.4, 1.0, 1.5, 2.0
colsample_ bylevel	Persentase fitur yang digunakan di setiap level pohon. Nilai yang lebih besar akan menghasilkan model yang lebih besar dan lebih kompleks, namun dapat berpotensi menyebabkan *overfitting*.	log2, sqrt, 0.25, 1.0
subsample	Persentase sampel yang digunakan untuk setiap pohon. Nilai yang lebih besar akan menghasilkan model yang lebih besar dan lebih kompleks, namun dapat berpotensi menyebabkan *overfitting*.	0.15, 0.5, 0.75, 1.0

2.3.8 Evaluasi Model

Klasifikasi akan diukur kinerja atau performansinya, dengan suatu matriks yang disebut *confusion matrix* dan metrik *Area Under Curve* (AUC) (Rachmi, 2020).

1. *Confusion Matrix*

   Confusion matrix merupakan salah satu metode pengukuran keputusan paling klasik dalam *supervised machine learning* (Xu et al., 2020). *Confusion matriks* mengandung informasi yang membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan hasil klasifikasi yang seharusnya (Caesaria et al., 2020). Gambar 2.8 berikut merupakan ilustrasi *confusion matrix* untuk kasus *binary classification* (klasifikasi dengan dua label kelas). Pada kasus *turnover* karyawan, umumnya kelas positif mewakili karyawan yang pergi (meninggalkan perusahaan). Sedangkan kelas negatif mewakili karyawan yang tidak pergi (Tharani & Raj, 2020).



Gambar 2.8 *Confusion Matrix*

Sumber: Sampurna (2021)

Confusion matrix menjadi dasar dalam perhitungan beberapa metrik pengukur kinerja model klasifikasi. Berikut merupakan metrik yang dimaksud (Sholikhati, 2022).

a. Akurasi, merupakan ukuran kemampuan model memprediksi secara benar.
b. *Misclassification Rate*, adalah kebalikan dari akurasi yaitu ukuran berapa kali pengklasifikasi melakukan kesalahan prediksi.

c. *Precision (positive)/Positive Predictive Value* (PPV), adalah proporsi prediksi positif yang benar dari total prediksi positif. Pada kasus *turnover* karyawan, PPV adalah proporsi karyawan yang diprediksi akan meninggalkan perusahaan (positif) dan benar-benar meninggalkan, dibandingkan dengan semua karyawan yang diprediksi akan meninggalkan perusahaan. Jika PPV tinggi, maka model memiliki performa yang baik dalam memprediksi karyawan yang akan pergi. Sebaliknya, jika PPV rendah, berarti model membuat banyak prediksi positif yang salah (*Type I Error*).
d. *Precision (negative)/Negative Predictive Value* (NPV), adalah proporsi prediksi negatif yang benar dari total prediksi negatif. Pada kasus *turnover* karyawan, NPV adalah proporsi karyawan yang diprediksi akan bertahan (negatif) dan benar-benar bertahan, dibandingkan dengan semua karyawan yang diprediksi akan bertahan. Jika NPV tinggi, berarti model memiliki performa yang baik dalam memprediksi karyawan yang akan bertahan.
e. *Recall (positive)/Sensitivity*, mengukur proporsi prediksi positif yang benar dari total aktual positif. Pada kasus *turnover* karyawan, *sensitivity* mengukur sejauh mana model dapat mengidentifikasi karyawan yang sebenarnya akan meninggalkan perusahaan. Jika model memiliki *sensitivity* yang tinggi, maka model memiliki performa yang baik dalam menangkap karyawan yang sebenarnya akan pergi. Sebaliknya, jika *sensitivity* rendah, berarti model melewatkan banyak karyawan yang sebenarnya akan pergi (*Type II Error*).
f. *Recall (negative)/Specificity*, mengukur proporsi prediksi negatif yang benar dari total aktual negatif. Pada kasus *turnover* karyawan, *specificity* mengukur sejauh mana model dapat mengidentifikasi karyawan yang sebenarnya akan bertahan di perusahaan. Jika model memiliki *specificity* yang tinggi, maka model memiliki performa yang baik dalam menangkap karyawan yang sebenarnya akan bertahan. Sebaliknya, jika *specificity* rendah, berarti model melakukan banyak kesalahan dalam mengidentifikasi karyawan yang sebenarnya akan bertahan sebagai karyawan yang akan pergi (*Type I Error*).

g. *F1-Score*, adalah metrik evaluasi klasifikasi biner yang mengukur seberapa baik model dalam menyeimbangkan presisi dan *recall*. Jika nilai *F1-Score* semakin mendekati 1, maka model memiliki presisi dan *recall* yang semakin sempurna.

$$F1\ Score = \frac{2(Recall \times Presisi)}{(Recall + Presisi)} \tag{2.10}$$

2. *Area Under Curve* (AUC)

AUC adalah metrik populer untuk mengukur kinerja pengklasifikasi. AUC diperoleh dengan menghitung luas di bawah kurva ROC (*Receiver Operating Characteristic*), dan menjadi metrik pengukur kinerja model klasifikasi yang direkomendasikan untuk kasus *imbalance data* (Wardhani et al., 2019). Menurut Gorunescu dalam Rachmi (2020), terdapat beberapa kategori keakuratan hasil klasifikasi berdasarkan nilai AUC seperti dalam Tabel 2.8 berikut.

Tabel 2.8 Kategori Nilai AUC

Nilai AUC	**Kategori**
0,90 – 1,00	Sangat Baik
0,80 – 0,90	Baik
0,70 – 0,80	Cukup Baik
0,60 – 0,70	Kurang Baik
0,50 – 0,60	Buruk

BAB III

METODE PENELITIAN

3.1 Objek Penelitian

Penelitian ini berfokus pada usaha memahami kejadian *turnover* karyawan melalui pengembangan model klasifikasi. Pemilihan kinerja model klasifikasi terbaik dilakukan dengan membandingkan algoritma XGBoost tanpa SMOTE dan XGBoost dengan SMOTE. Harapannya, model yang terbaik dapat membantu PT. PNR untuk menyusun strategi dalam mempertahankan karyawan yang berbakat.

3.2 Lokasi Penelitian

Penelitian ini dilaksanakan di PT. PNR, yang merupakan kontraktor spesialis penyedia jasa pertambangan komprehensif kepada pemilik tambang. Divisi yang terkait dalam lokasi penelitian adalah Divisi Human Capital & Talent Development di *site* Jakarta.

3.3 Data Penelitian

Data penelitian ini adalah data sekunder berupa data karyawan yang dikumpulkan dari *bank data* Divisi Human Capital & Talent Development PT. PNR. Karyawan yang dimaksud adalah karyawan tenaga ahli Divisi Engineering PT. PNR golongan 4C ke atas. Rentang waktu rekapitulasi data yang diambil adalah mulai Januari hingga Agustus 2023. *Dataset* terdiri dari karyawan yang masih bertahan hingga Agustus 2023, dan karyawan yang keluar dari perusahaan antara Januari 2023 hingga Agustus 2023. Pertimbangannya adalah untuk tahun 2020-2022 masih masuk kedalam status masa *Covid-19* yang dikhawatirkan mempengaruhi kejadian *turnover* pada masa itu. Sedangkan pada tahun sebelum 2020, jumlah *expert* masih sangat sedikit karena baru "*established*" atau dikembangkan, dan membutuhkan waktu lebih untuk penarikan datanya. Untuk *dataset* karyawan yang masih bertahan, data diambil berdasarkan rekapitulasi bulan Agustus 2023. Sedangkan untuk *dataset* karyawan yang sudah keluar, data diambil berdasarkan rekapitulasi di bulan karyawan tersebut keluar dari perusahaan. Total data berjumlah 170 baris data (156 *Leave = No*, 14 *Leave = Yes*) dengan 19 atribut dan satu label kelas. Dari data yang dimiliki terlihat bahwa terdapat ketidakseimbangan, dengan persentase label kelas *Leave = Yes* (meninggalkan perusahaan) hanya sebesar 8,24% dari total data.

3.4 Variabel Penelitian

Variabel dalam penelitian ini terdiri dari variabel independen yang berisi 19 atribut dan satu variabel dependen yang disebut dengan target atau label kelas. Definisi operasional dari setiap variabel penelitian dapat dilihat pada Tabel 3.1 berikut.

Tabel 3.1 Atribut dan Label Kelas untuk Penelitian

No	Varibel	Definisi	Tipe Data	Referensi
Variabel Independen (Demografis)				
1	ID	Nomor identitas karyawan	*Integer*	(Gao et al., 2019)
2	*Gender*	Jenis kelamin karyawan	*Nominal Categorical*	(Gao et al., 2019; Khera & Divya, 2019; Kovvuri & Dommeti, 2022; Noviyanti, 2018; Tao et al., 2021; Tharani & Raj, 2020; Zhao et al., 2019)
3	*Age*	Umur karyawan dalam satuan tahun	*Integer*	(Chanodkar et al., 2019; Gao et al., 2019; Juvitayapun, 2021; Khera & Divya, 2019; Kovvuri & Dommeti, 2022; Palupi, 2021; Tao et al., 2021; Zhao et al., 2019)
4	*Generation*	Generasi karyawan yang didasarkan pada tahun kelahirannya	*Nominal Ordinal*	(Redafanza et al., 2023)
5	*Marital Status*	Status karyawan apakah sudah menikah atau belum, dan jumlah anak yang dimiliki	*Nominal Categorical*	(Gao et al., 2019; Juvitayapun, 2021; Khera & Divya, 2019; Tao et al., 2021; Tharani & Raj, 2020)
6	*Last Education*	Tingkat pendidikan terakhir yang telah dicapai oleh karyawan	*Ordinal Categorical*	(Gao et al., 2019; Juvitayapun, 2021; Khera & Divya, 2019; Kovvuri & Dommeti, 2022; Palupi, 2021; Tao et al., 2021; Tharani & Raj, 2020; Zhao et al., 2019)
7	*Education Field*	Bidang studi yang diambil oleh karyawan berdasarkan pendidikan terakhirnya	*Nominal Categorical*	(Gao et al., 2019; Tao et al., 2021; Zhao et al., 2019)
Variabel Independen (*Job Related and Organizational*)				
8	*Site*	Lokasi kerja karyawan	*Nominal Categorical*	Perusahaan
9	*Department*	Departemen tempat bekerja karyawan	*Nominal Categorical*	(Duan, 2022; Gao et al., 2019; Khera & Divya, 2019; Tao et al., 2021; Zhao et al., 2019)
10	*Job Level*	Level atau tingkatan pekerjaan karyawan	*Ordinal Categorical*	(Gao et al., 2019; Khera & Divya, 2019; Tao et al., 2021; Zhao et al., 2019)
11	*Job Role*	Jabatan atau posisi pekerjaan karyawan	*Nominal Categorical*	(Gao et al., 2019; Khera & Divya, 2019; Tao et al., 2021)
12	*Specialization Area*	Area spesialisasi pekerjaan karyawan	*Nominal Categorical*	(Zhao et al., 2019)
13	*Training*	Jumlah pelatihan yang telah selesai diikuti karyawan	*Integer*	(Gao et al., 2019; Khera & Divya, 2019; Tao et al., 2021)

No	Varibel	Definisi	Tipe Data	Referensi
14	*Competency*	Nilai kompetensi atau keterampilan karyawan	*Ordinal Categorical*	(Maulida & Rusilowati, 2020)
15	*Years of Service*	Masa kerja karyawan di perusahaan saat ini dalam satuan tahun	*Integer*	(Duan, 2022; Gao et al., 2019; Juvitayapun, 2021; Khera & Divya, 2019; Palupi, 2021; Tao et al., 2021; Zhao et al., 2019)
16	*Current Role Tenure*	Masa kerja karyawan di posisi saat ini dalam satuan bulan	*Integer*	(Gao et al., 2019; Juvitayapun, 2021; Khera & Divya, 2019)
17	*Last Promotion*	Jarak waktu dari promosi terakhir yang didapatkan karyawan dalam satuan bulan	*Integer*	(Chanodkar et al., 2019; Duan, 2022; Gao et al., 2019; Khera & Divya, 2019; Tao et al., 2021)
18	*Assignment Letter*	Jumlah surat tugas yang pernah dibebankan kepada karyawan	*Integer*	Perusahaan
19	*Production Plan*	Jumlah rencana/permintaan produksi yang perlu dicapai dalam satuan ton	*Integer*	Perusahaan
Variabel Dependen				
20	*Leave*	Apakah karyawan telah meninggalkan perusahaan	*Binary*	(Alaskar et al., 2019; Gao et al., 2019; Palupi, 2021)

Berdasarkan penelitian Khera & Divya (2019), secara umum atribut yang mempengaruhi *turnover* terbagi menjadi demografis serta *job related and organizational*. Atribut nomor 1 hingga 7 tergolong kedalam atribut demografis. Sedangkan atribut nomor 8 hingga 19 merupakan atribut *job related and organizational*.

Penelitian ini menggunakan tiga atribut yang spesifik dalam industri pertambangan, untuk mendukung keterbaruan penelitian. Ketiga atribut spesifik ini, seluruhnya dipertimbangkan berdasarkan sorotan awal dari tantangan *turnover* sukarela, yaitu mengindikasikan ketidakpuasan pekerjaan (Dwesini, 2019). Penelitian oleh Farizi & Tanuwijaya (2022) juga telah membuktikan bahwa kepuasan kerja berpengaruh secara negatif dan signifikan terhadap niat *turnover* pada karyawan di industri pertambangan.

Pada PT. PNR, data survei kepuasan kerja karyawan bersifat sangat rahasia dan dimiliki langsung oleh perusahaan induk dari PT. PNR. Sehingga, untuk mempertahankan integritas dan kerahasiaan data tersebut, peneliti mencari atribut alternatif yang dapat mempengaruhi kepuasan kerja. Atribut alternatif tersebut dipilih berdasarkan diskusi mendalam mengenai kebutuhan dan ketersediaan data PT. PNR, dan dijadikan sebagai atribut spesifik dalam penelitian ini.

Berikut merupakan penjelasan mengenai ketiga atribut spesifik dalam penelitian.

1. *Site* (lokasi kerja).

   Lokasi kerja dapat mempengaruhi berbagai aspek pengalaman kerja karyawan, termasuk kondisi kerja, akses ke fasilitas, dan jarak dari rumah. Hal ini dapat mempengaruhi kepuasan kerja dan keputusan untuk tetap atau meninggalkan perusahaan. Menurut seorang Engineering Department Head Site, kasus di lapangan yaitu pernah terjadi keluhan komunikasi dengan keluarga, karena beberapa *site* memiliki fasilitas sinyal yang kurang bagus. Sehingga kualitas komunikasi dengan keluarga tidak sebaik di *site* yang sinyalnya bagus.

2. *Assignment Letter* (jumlah surat tugas).

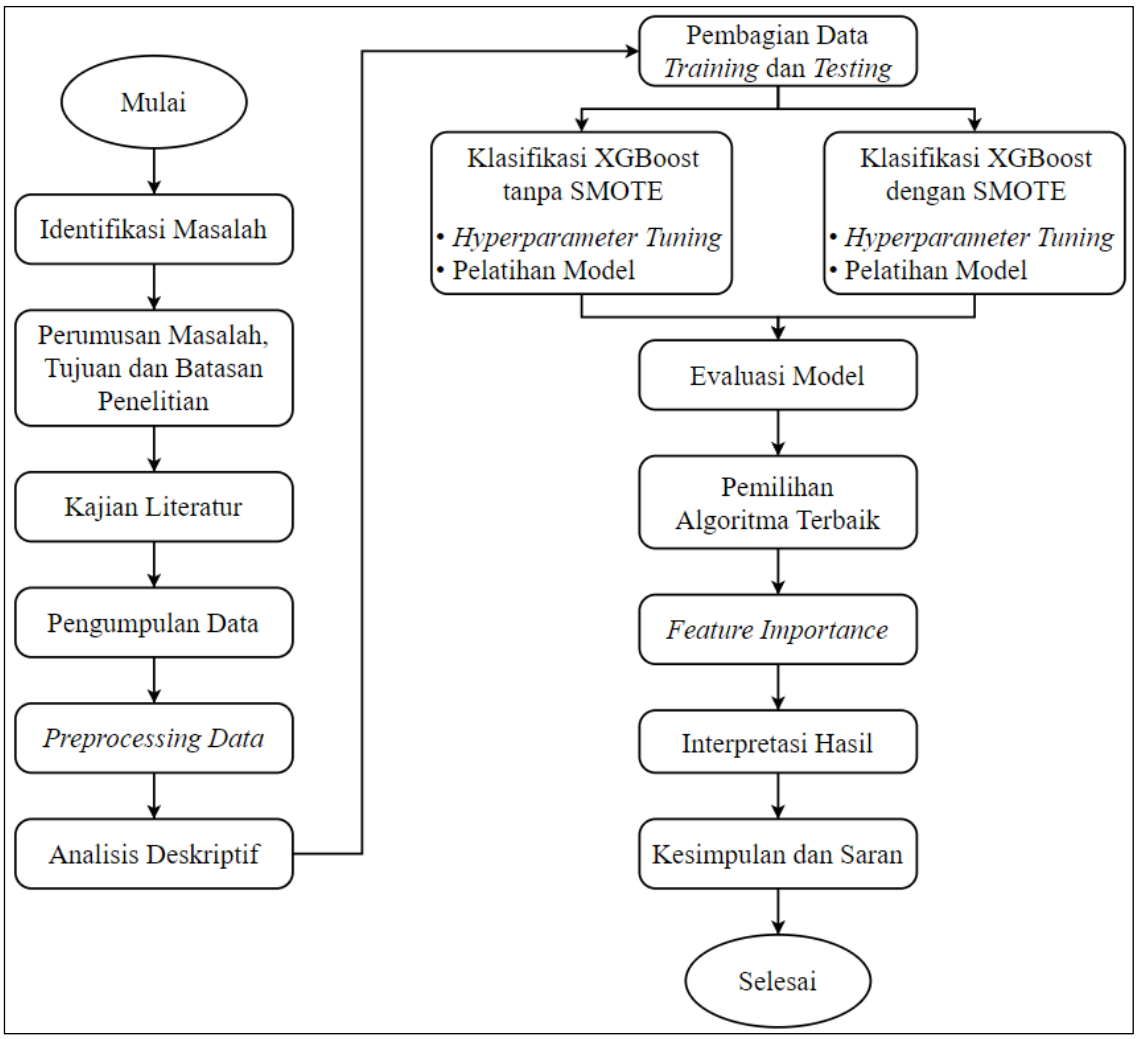
   Jumlah surat tugas dapat mencerminkan beban kerja karyawan, dimana jika seorang karyawan menerima banyak surat tugas, maka menunjukkan bahwa karyawan tersebut memiliki beban kerja yang tinggi, yang bisa mempengaruhi kepuasan kerja dan akhirnya keputusan untuk tetap atau meninggalkan perusahaan. Menurut seorang Expert Track Management Officer, hal ini terbukti terjadi di lapangan. Sebagai contoh, karyawan *Site* A di Jakarta diberikan surat tugas untuk ke *Site* B yang berlokasi di Sumatera. Maka ia akan mengerjakan tugas yang diberikan selama berada di *Site* B, dan kemungkinan besar juga masih harus mengerjakan tugas yang biasanya ia lakukan di *site* asalnya (*Site* A).

3. *Production Plan* (rencana produksi).

   Rencana produksi bisa mempengaruhi tekanan kerja dan harapan yang diberikan kepada karyawan, yang pada akhirnya akan berdampak pada kepuasan kerja yang dirasakan. Pada PT. PNR, rencana produksi ini muncul dari *demand*/permintaan produksi yang diberikan oleh *owner*/pemilik tambang sebagai *customer*. *Production plan* ini secara langsung ditargetkan kepada karyawan yang bekerja di *site* selain Jakarta, karena memang aktivitas penambangan hanya bisa dilakukan diluar *site* berlokasi Jakarta (hanya di *site* berlokasi Kalimantan dan Sumatera). Namun menurut seorang Expert Track Management Section Head, asumsinya *production plan* ini menjadi target bersama untuk semua *site* termasuk *site* yang berlokasi di Jakarta. Sehingga walaupun untuk *site* berlokasi Jakarta tidak ada *production plan* secara langsung, namun dapat diasumsikan menggunakan rata-rata dari *production plan* seluruh *site* berlokasi Kalimantan dan Sumatera.

3.5 Alur Penelitian

Berikut ini merupakan alur dari penelitian yang dilakukan, yang disajikan dalam bentuk diagram alur atau *flowchart*.



Gambar 3.1 Alur Penelitian

Penjelasan mengenai alur penelitian pada Gambar 3.1 Alur Penelitian adalah sebagai berikut.

1. Identifikasi Masalah.

   Peneliti melakukan pemahaman terhadap masalah yang sedang dihadapi melalui observasi dan diskusi dengan pihak terkait di PT. PNR. Hasilnya menunjukan bahwa memang terjadi peningkatan kejadian *turnover* pada karyawan jalur tenaga ahli Divisi Engineering. PT. PNR juga belum melakukan pemanfaatan lanjutan dari data

historis yang dimiliki, dengan menggunakan teknik yang ada dalam *data mining* dan algoritma *machine learning* untuk pengelolaan masalah *turnover* tersebut.

2. Perumusan Masalah, Tujuan dan Batasan Penelitian.

   Perumusan masalah dilakukan berdasarkan latar belakang yang telah disusun, dan kemudian merancang tujuan penelitian yang berkaitan langsung dengan rumusan masalah tersebut agar penelitian menjadi relevan. Untuk memfokuskan penelitian, ditetapkan batasan-batasan masalah yang diterapkan dalam penelitian, untuk membantu peneliti merancang metodologi penelitian dan analisis data yang tepat.

3. Kajian Literatur.

   Sub-bab pertama dari tahapan ini merupakan kajian beberapa literatur terdahulu untuk mendukung penelitian yang dilakukan. Untuk melihat perbedaan dan keunikan penelitian yang dilakukan, disusunlah sub-bab kedua yaitu *research gap*. Dan sub-bab terakhir yaitu landasan teori memberikan penjelasan terkait acuan teori yang digunakan dalam melakukan penelitian.

4. Pengumpulan Data.

   Untuk mencapai tujuan penelitian dan menyelesaikan permasalahan, peneliti mengumpulkan informasi yang relevan. Data penelitian ini adalah data sekunder yang berasal dari *bank data* Divisi Human Capital & Talent Development PT. PNR.

5. *Preprocessing Data*.

   Merupakan proses persiapan data, dimana pada penelitian ini terdiri dari mengecek data yang hilang dan terduplikasi, transformasi data (memberikan label untuk data kategorik) dan *feature selection*. *Feature selection* pada penelitian ini menggunakan *correlation matrix*. Seleksi atribut atau *feature selection* dilakukan untuk membantu meningkatkan akurasi model dan mengurangi kompleksitas komputasi.

6. Analisis Deskriptif.

   Analisis deskriptif terhadap data yang dimiliki dilakukan untuk melihat gambaran data yang digunakan.

7. Pembagian Data *Training* dan *Testing*.

   Pembagian data *training* dan *testing* atau *splitting data* dengan *StratifiedKFold*, yang merupakan teknik validasi silang yang membagi data menjadi ‘k’ lipatan sama besar. Data *training* digunakan untuk melatih model, sedangkan data *testing* untuk mengevaluasi model. Dengan cara kerja tersebut, *StratifiedKFold* memastikan model

dilatih dan diuji pada bagian yang berbeda dari data, dan setiap kelas direpresentasikan dengan baik di setiap lipatan.

8. Klasifikasi dengan XGBoost tanpa SMOTE dan XGBoost dengan SMOTE.

   Pada klasifikasi XGBoost tanpa SMOTE, data *training* yang telah dipisah dari *data testing* akan langsung digunakan untuk melatih model dengan terlebih dahulu mencari *hyperparameter* terbaik. Sedangkan pada klasifikasi XGBoost dengan SMOTE, data *training* akan diseimbangkan terlebih dahulu proporsi kelas datanya menggunakan teknik oversampling SMOTE, dan kemudian digunakan untuk tahap selanjutnya yaitu mencari *hyperparameter* terbaik dan melatih model.

9. Evaluasi Model.

   Tahapan ini memanfaatkan *confusion matrix* dan metrik AUC untuk mengukur kinerja atau performa model yang terbentuk. Dari *confusion matrix* didapatkan metrik-metrik performa mulai dari akurasi hingga *F1-Score*.

10. Pemilihan Algoritma Terbaik.

    Hasil evaluasi dari setiap model akan dibandingkan untuk melihat algoritma apa yang memberikan performa terbaik dalam memprediksi data *testing*. Serta dilakukan pengecekan tanda-tanda *overfitting* pada model, dengan membandingkan performa *train* dan *test set* yang didapatkan.

11. *Features Importance*.

    Tahapan ini didasarkan pada algoritma yang terbaik, dan dilakukan untuk melihat seberapa penting setiap fitur dalam membuat prediksi berdasarkan algoritma tersebut.

12. Interpretasi Hasil.

    Peneliti membahas temuan yang diperoleh dan menjelaskan maksudnya. Temuan ini akan dibandingkan juga dengan penelitian terdahulu untuk melihat apakah terdapat persamaan atau perbedaan.

13. Kesimpulan dan Saran.

    Peneliti menyimpulkan hasil penelitian untuk menjawab rumusan masalah dan tujuan penelitian yang telah dirumuskan di awal penelitian. Saran diberikan sebagai bahan perbaikan dan pengembangan di masa datang.

BAB IV

PENGUMPULAN DAN PENGOLAHAN DATA

4.1 Pengumpulan Data

Data yang dibutuhkan dalam penelitian ini merupakan data historis karyawan Divisi Engineering PT. PNR golongan 4C ke atas (*expert*), yang bersumber dari *bank data* Divisi Human Capital & Talent Development PT. PNR. Rentang waktu data historis yang diambil adalah mulai dari Januari hingga Agustus 2023. Pertimbangannya adalah untuk tahun 2020-2022 masih dalam status masa *Covid-19* yang dikhawatirkan mempengaruhi kejadian *turnover* pada masa itu. Sedangkan pada tahun sebelum 2020, jumlah *expert* masih sangat sedikit karena baru "*established*" atau dikembangkan, dan membutuhkan waktu lebih untuk penarikan datanya. Data dikumpulkan kedalam satu file *excel* dan diubah ke dalam format CSV (*Comma Separated Values*) agar lebih kompatibel dengan *platform* Google Collab yang digunakan untuk pengolahan data. Total data yang digunakan berjumlah 170 baris data karyawan, dengan 19 atribut dan satu label kelas.

4.2 *Pre-Processing*

Berikut merupakan tahapan *pre-processing* sebagai usaha mempersiapkan data, serta meningkatkan efisiensi dan efektivitas proses analisis data.

4.2.1 Pengecekan *Missing Value,* Duplikasi dan *Outlier*

Pada tahap ini dilakukan proses identifikasi data yang hilang atau tidak lengkap, data berulang (duplikat) dan data pencilan (*outlier*) dalam *dataset*. *Missing value* dan duplikasi dalam data dapat mengganggu analisis dan biasanya perlu dihapus atau diperbaiki. Untuk mengecek keberadaan *missing value* dan duplikasi data, dapat dilihat melalui *output* ringkasan dari *DataFrame* dan pengecekan duplikasi pada Tabel 4.1.

Tabel 4.1 *Output* Pengecekan *Missing Value* dan Duplikasi

***RangeIndex*: 170 *entries*, 0 to 169**
***Data columns* (*total* 20 *columns*):**

Index	*Column*	*Non-Null Count*	*Dtype*
0	*ID*	*170 non-null*	*int64*
1	*Gender*	*170 non-null*	*object*
2	*Age*	*170 non-null*	*int64*
3	*Generation*	*170 non-null*	*object*
4	*Marital Status*	*170 non-null*	*object*
5	*Last Education*	*170 non-null*	*object*
6	*Education Field*	*170 non-null*	*object*
7	*Site*	*170 non-null*	*object*
8	*Department*	*170 non-null*	*object*
9	*Job Level*	*170 non-null*	*object*
10	*Job Role*	*170 non-null*	*object*
11	*Specialization Area*	*170 non-null*	*object*
12	*Training*	*170 non-null*	*int64*
13	*Competency*	*170 non-null*	*object*
14	*Years of Service*	*170 non-null*	*int64*
15	*Current Role Tenure*	*170 non-null*	*int64*
16	*Last Promotion*	*170 non-null*	*int64*
17	*Assignment Letter*	*170 non-null*	*int64*
18	*Production Plan*	*170 non-null*	*int64*
19	*Leave*	*170 non-null*	*object*

***dtypes*: *int64*(8), *object*(12)**
***memory usage*: 26.7+ KB**
***Number of duplicate rows* = 0**

Tabel 4.1 tersebut memberikan informasi bahwa *DataFrame* memiliki 170 baris, yang dinomori dari 0 sampai 169, 20 kolom total yang terdiri dari 8 kolom dengan tipe data int64 (*integer*/mewakili angka bulat) dan 12 kolom dengan tipe data *object* (mewakili tipe data teks/*string* atau kategorik), 170 nilai *non-null* yang artinya tidak ada *missing value* (data yang hilang), dan perkiraan penggunaan memori oleh *DataFrame* sebesar 26.7+ KB. Jumlah nilai *non-null* dalam setiap kolom dapat dijadikan acuan untuk pengecekan *missing value*. Karena jumlah entri *non-null* dalam setiap kolom tidak ada yang kurang dari jumlah total entri (170), maka dapat disimpulkan bahwa tidak terdapat *missing value* pada *DataFrame*. Untuk pengecekan duplikasi data, diketahui dari *output* tersebut bahwa `Number of duplicates = 0` yang artinya juga tidak terdapat data yang terduplikasi atau berulang. Untuk pengecekan *outlier* dilakukan dengan memanfaatkan `print` jumlah *outlier* pada masing-masing atribut yang dapat dilihat pada Tabel 4. 2. Terlihat bahwa pada data penelitian mengandung *outlier*, yang mana kondisi ini menjadi salah satu pertimbangan penggunaan XGBoost dibandingkan beberapa algoritma pengklasifikasi lainnya seperti yang disebutkan pada latar belakang sebelumnya. XGBoost telah terbukti memiliki keunggulan dalam menangani keberadaan *outlier* ini secara otomatis.

Tabel 4. 2 *Output* Pengecekan *Outlier*

Atribut	Jumlah *Outlier*	Persentase *Outlier* dari Keseluruhan Data
Gender	7	4.12%
Age	8	4.71%
Marital Status	15	8.82%
Last Education	10	5.88%
Education Field	1	0.59%
Site	0	0.00%
Department	64	37.65%
Job Level	25	14.71%
Job Role	4	2.35%
Training	10	5.88%
Competency	78	45.88%
Years of Service	2	1.18%
Current Role Tenure	29	17.06%
Last Promotion	0	0.00%
Assignment Letter	24	14.12%
Production Plan	0	0.00%

4.2.2 Transformasi Data

Penelitian ini menerapkan dua tahap transformasi data untuk mempersiapkan *dataset* ke analisis yang lebih lanjut. Tahap pertama melibatkan penggunaan *Label Encoder* pada semua variabel kategorik, baik yang bersifat ordinal maupun nominal. Tujuan dari tahap ini adalah untuk mengubah variabel kategorik menjadi representasi numerik yang dapat digunakan dalam proses seleksi atribut. Meskipun *Label Encoder* memberikan urutan pada variabel, namun peneliti memahami bahwa ini adalah kompromi yang diperlukan untuk dapat melakukan seleksi atribut menggunakan *correlation matrix*.

Maksud kompromi disini adalah, sebelum melakukan seleksi atribut, peneliti tidak menggunakan *One-Hot Encoder* (dimana setiap kategori *instance* variabel menjadi kolom variabel baru) dalam pelabelan variabel kategorik bersifat nominal. Hal ini dikarenakan peneliti tidak ingin menimbulkan kerancuan, karena jika kolom variabel baru hasil *One-Hot Encoder* memiliki korelasi tinggi dengan variabel lainnya, maka secara teknis dapat menghapus satu level dari variabel kategorik bersifat nominal tersebut. Contoh misal terdapat variabel kategorik nominal dengan nama *Department* yang memiliki kategori *instance* A, B dan C. Dengan *One-Hot Encoder*, maka kategori *instance* variabel tersebut menjadi kolom variabel baru bernama *Department*_A, *Department*_B dan *Department*_C. Jika ternyata hasil *correlation matrix* menunjukan bahwa *Department*_A berkorelasi dengan variabel lain, maka akan menimbulkan kerancuan dalam pertimbangan proses seleksi atribut. Dimana jika *Department*_A ini yang dihapus, maka secara teknis satu level dari variabel kategorik (dalam hal ini, '*Department*') menjadi berkurang.

Namun setelah seleksi atribut selesai dilakukan, peneliti kemudian melakukan tahap transformasi data yang kedua. Pada tahap transformasi yang kedua ini, peneliti menggunakan *One-Hot Encoder* pada variabel kategorik nominal dan menggunakan *Ordinal Encoder* untuk variabel kategorik ordinal. Tujuan tahapan ini adalah mempersiapkan data untuk pembangunan model klasifikasi, dengan memastikan bahwa model tidak membuat asumsi urutan yang tidak ada pada variabel kategorik nominal. Dengan demikian, hasil dari proses transformasi data ini adalah dua jenis data. Yaitu yang pertama adalah "Data untuk Seleksi Atribut", yang menggunakan hasil *Label Encoder* untuk semua variabel kategorik. Serta data yang kedua adalah "Data untuk Pembangunan Model", yang menggunakan hasil *Ordinal Encoder* untuk variabel kategorik ordinal dan hasil *One-Hot Encoding* untuk variabel kategorik nominal. Peneliti melakukan pendekatan ini untuk memastikan bahwa setiap tahap analisis menggunakan data yang paling sesuai dengan kebutuhannya.

4.2.3 Data untuk Seleksi Atribut

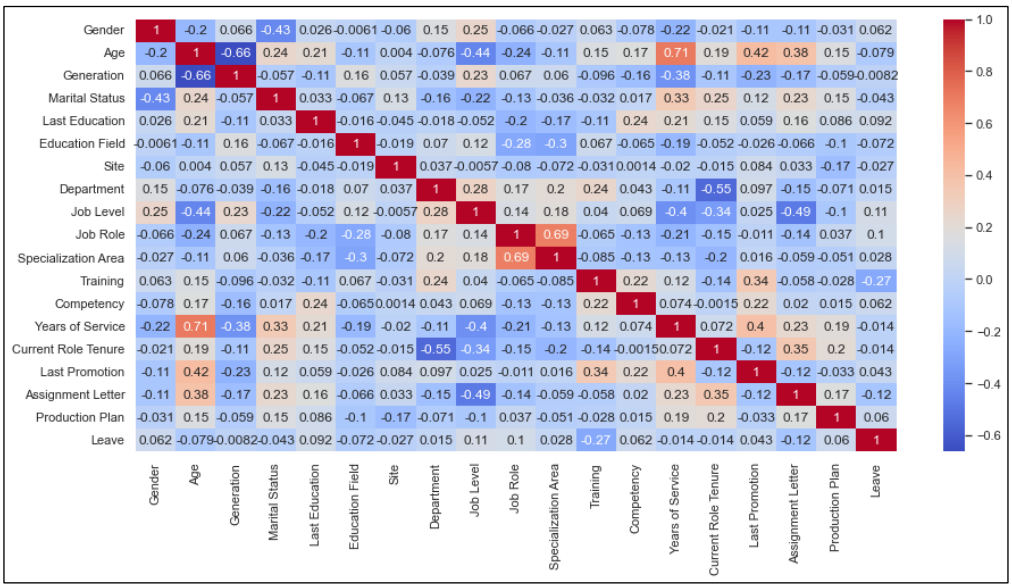
Proses seleksi atribut akan menggunakan data hasil *Label Encoder* di semua variabel kategorik. Tabel 4.3 menunjukan lima baris data teratas hasil *Label Encoder*, yang dapat diketahui bahwa untuk data variabel kategorik, semuanya telah berubah menjadi bentuk numerik. Data inilah yang kemudian digunakan dalam proses seleksi atribut selanjutnya. Peneliti menggunakan *Label Encoder* sebagai solusi yang lebih sederhana dan efisien, dengan syarat bahwa perlu dipahami mengenai angka yang dihasilkan adalah representasi arbitrer dari kategori (tidak mencerminkan urutan/jarak).

Tabel 4.3 Data untuk Seleksi Atribut

Index	Gender	Generation	Marital Status	Last Education	Education Field	Site
0	1	1	1	2	15	0
1	1	1	0	2	14	0
2	1	1	1	2	6	0
3	1	1	1	2	16	0
4	1	1	1	2	16	0
Index	**Department**	**Job Level**	**Job Role**	**Specialization Area**	**Competency**	**Leave**
0	5	1	12	3	0	0
1	5	1	17	7	3	0
2	5	1	18	7	0	0
3	5	1	11	3	0	0
4	5	1	5	3	2	0

4.2.4 Seleksi Atribut

Tahap seleksi atribut atau *feature selection* pada penelitian ini dilakukan dengan menghapus salah satu dari dua atau lebih variabel independen yang saling berkorelasi tinggi. Untuk mempermudah pembacaan nilai-nilai korelasi *pearson* yang dihasilkan, maka digunakanlah visualisasi dengan *correlation matrix*. Cara seleksi atribut seperti ini dapat memungkinkan algoritma *machine learning* untuk bekerja lebih cepat dalam proses pelatihan model, mengurangi kompleksitas model sehingga *output* dapat lebih mudah diinterpretasikan, dan dapat meningkatkan nilai akurasi model (Rachmi, 2020).



Gambar 4.1 *Correlation Matrix*

Gambar 4.1 menunjukan nilai-nilai korelasi dari seluruh atribut yang digunakan. Atribut yang berkorelasi cukup tinggi diantaranya adalah *Age* dengan *Years of Service* (0,71), *Job Role* dengan *Specialization Area* (0,69) dan *Age* dengan *Generation* (-0,66). Dari pasangan atribut yang saling berkorelasi tinggi tersebut, atribut yang dipilih adalah yang memiliki *instance* lebih heterogen dan korelasi lebih tinggi terhadap target. Sebaliknya, atribut yang dihapus adalah yang memiliki *instance* lebih homogen dan korelasi lebih rendah terhadap target. Atribut dengan heterogenitas nilai yang lebih tinggi akan lebih dipilih karena memberikan informasi yang lebih beragam kepada model, sehingga memungkinkan model untuk belajar pola yang lebih kompleks dalam data dan menghasilkan prediksi yang lebih akurat. Selain itu, atribut dengan nilai yang homogen akan cenderung memiliki variasi yang rendah, sehingga bisa menyebabkan model

mempelajari kebisingan dalam data yang pada akhirnya menyebabkan *overfitting*. Sedangkan atribut dengan nilai korelasi terhadap target lebih tinggi akan lebih dipilih karena dapat membantu model klasifikasi mempelajari hubungan yang lebih kuat antara atribut dan target. Sehingga model dapat memprediksi target dengan lebih baik berdasarkan nilai atribut (prediksi menjadi lebih akurat). Selain itu, memilih atribut yang berkorelasi dengan target akan membantu menghasilkan model yang lebih mudah dipahami, karena atribut yang dipilih memiliki pengaruh yang jelas terhadap target.

Korelasi antara *Age* dengan *Years of Service* tidak ada atribut yang dihapus, karena pertimbangan pihak perusahaan yang merasa bahwa kedua atribut tersebut dibutuhkan untuk digunakan dalam klasifikasi kasus *turnover* karyawan. Korelasi *Job Role* dengan *Specialization Area*, digunakan *Job Role* saja dan menghapus *Specialization Area*. Dengan pertimbangan *Job Role* memiliki *instance* yang lebih spesifik (heterogen) dan lebih berkorelasi tinggi terhadap target dibandingkan dengan *Specialization Area*. Sedangkan korelasi *Age* dengan *Generation*, digunakan *Age* saja dan menghapus *Generation*. Dengan pertimbangan *Age* yang dirasa penting tadi, serta memiliki *instance* yang lebih heterogen dan lebih berkorelasi tinggi terhadap target dibandingkan dengan *Generation*. Dengan dihapusnya kedua atribut tersebut (*Specialization Area* dan *Generation*), maka total atribut menjadi sejumlah 16 atribut.

4.2.5 Data untuk Pembangunan Model

Setelah dilakukan seleksi atribut, dilakukanlah transformasi dengan *One-Hot Encoder* untuk variabel kategorik nominal, dan *Ordinal Encoder* untuk variabel kategorik ordinal. Cara ini dilakukan untuk membantu algoritma klasifikasi dalam memastikan bahwa model tidak membuat asumsi urutan yang tidak ada pada variabel kategorik nominal, dan mengenali urutan pada variabel kategorik ordinal. Pada variabel kategorik nominal yang diterapkan *One-Hot Encoder*, akan membentuk kolom variabel baru untuk setiap kategori *instance* yang dimilikinya. Sebagai contoh, pada variabel *Marital Status* dengan lima kategori yaitu *S0(Single0)*, *M0(Married0)*, *M1(Married1)*, *M2(Married2)*, dan *M3(Married3)*, akan membentuk lima kolom variabel baru bernama *Marital Status_S0(Single0)*, *Marital Status_M0(Married0)*, *Marital Status_M1(Married1)*, *Marital Status_M2(Married2)* dan *Marital Status_M3(Married3)*. Terlihat pada Tabel 4.4, kolom variabel baru tersebut akan berisikan nilai 0.0 jika karyawan tidak memiliki

status pernikahan tersebut, dan akan berisikan nilai 1.0 jika karyawan memiliki status pernikahan tersebut. Hal ini berlaku untuk semua variabel kategorik nominal lainnya.

Tabel 4.4 Transformasi Atribut Kategorik Nominal dengan *One Hot Encoder*

Index	***Marital Status_ S0(Single0)***	***Marital Status_ M0(Married0)***	***Marital Status_ M1(Married1)***	***Marital Status_ M2(Married2)***	***Marital Status_ M3(Married3)***
0	0	0	1	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	1	0	0
4	0	0	1	0	0

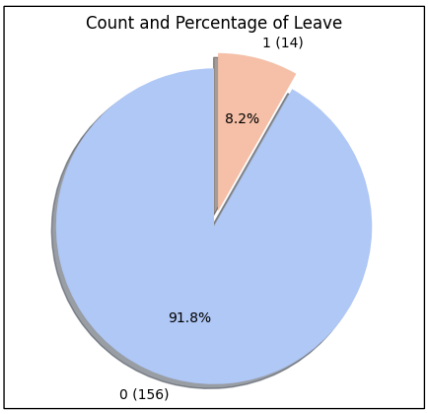
Kemudian untuk variabel kategorik ordinal, transformasi numeriknya mempertimbangkan urutan yang dapat dilihat pada Tabel 4.5. Data untuk pembangunan model ini memiliki total kolom sejumlah 80 kolom.

Tabel 4.5 Transformasi Atribut Kategorik Ordinal dengan *Ordinal Encoder*

Variabel Kategorik Ordinal					
Last Education		***Job Level***		***Competency***	
SMK	0.0	Pratama	0.0	A	0.0
D1	1.0	Madya	1.0	B	1.0
D3	2.0	Utama	2.0	C	2.0
S1	3.0			D	3.0
S2	4.0				

4.3 Analisis Deskriptif

Untuk melihat gambaran data yang digunakan, dilakukanlah tahapan analisis deskriptif. Pada tahap ini, data yang digunakan adalah data hasil seleksi atribut. Berikut merupakan beberapa analisis deskriptif yang dilakukan, beserta penjelasannya.



Gambar 4.2 Proporsi Label Kelas Data

Pada Gambar 4.2 diketahui bahwa dari 170 total data yang dimiliki, terdapat 91,8% atau setara 156 orang karyawan yang masih bertahan di perusahaan (0 = *No*). Sedangkan 8,2% lainnya atau setara 14 orang karyawan telah meninggalkan perusahaan (1 = *Yes*).

Tabel 4.6 *Statistics Summary* Atribut Numerik

Atribut	***mean***	***std***	***min***	**25%**	**50%**	**75%**	***max***
Age	35,35	4,34	28	32,25	35	37	53
Training	6,05	5,84	0	2	4,5	8	37
Years of Service	11,16	3,49	5	9	11	13	27
Current Role Tenure	22,72	22,30	0	10,25	17	20	80
Last Promotion	33,61	23,51	1	13	31	55	103
Assignment Letter	2,73	5,18	0	0	0	3,75	19
Production Plan	5231049,26	1832578,00	2000000	4000000	5762500	6525000	8333000

Tabel 4.6 menunjukan ringkasan statistik dari seluruh data numerik yang digunakan. Interpretasi dari tabel tersebut adalah sebagai berikut.

1. *Age*: Rata-rata usia adalah 35,35 tahun dengan standar deviasi 4,34 tahun. Hal ini berarti usia karyawan bervariasi sekitar 4,34 tahun di sekitar rata-rata. Usia minimum adalah 28 tahun dan maksimum 53 tahun. Nilai median (50%) adalah 35 tahun, yang berarti setengah dari karyawan berusia di bawah 35 tahun dan setengahnya lagi berusia di atas 35 tahun.
2. *Training*: Rata-rata jumlah pelatihan yang diikuti oleh karyawan adalah 6,05 pelatihan, dengan standar deviasi 5,84. Nilai standar deviasi tersebut menunjukkan variasi yang cukup besar dalam jumlah pelatihan yang diikuti oleh karyawan. Beberapa karyawan tidak mengikuti pelatihan sama sekali (minimum), sementara ada karyawan yang mengikuti hingga 37 pelatihan (maksimum).
3. *Years of Service*: Rata-rata lama bekerja adalah 11,16 tahun dengan standar deviasi 3,49 tahun. Nilai minimum adalah selama 5 tahun dan maksimum selama 27 tahun.
4. *Current Role Tenure*: Rata-rata masa jabatan saat ini adalah 22,72 bulan dengan standar deviasi 22,30 bulan. Nilai standar deviasi tersebut menunjukkan variasi yang sangat besar dalam masa jabatan. Beberapa karyawan terlihat baru saja memulai perannya (minimum 0 bulan), sementara yang lain telah berada dalam peran saat ini selama 80 bulan (maksimum).
5. *Last Promotion*: Rata-rata waktu sejak promosi terakhir adalah 33,61 bulan dengan standar deviasi 23,51 bulan. Beberapa karyawan baru saja dipromosikan (minimum

1 bulan), sementara yang lain sudah menunggu hingga 103 bulan sejak promosi terakhir didapatkan (maksimum).

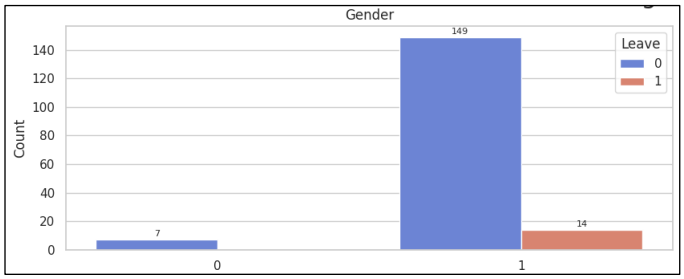
6. *Assignment Letter*: Rata-rata jumlah surat tugas adalah 2,73 dengan standar deviasi 5,18. Beberapa karyawan tidak memiliki surat tugas (minimum), sementara yang lain memiliki hingga 19 surat tugas (maksimum).
7. *Production Plan*: Rata-rata rencana produksi adalah 5231049.26 dengan standar deviasi 1832578.00. Rencana produksi minimum adalah 2000000 dan maksimum adalah 8333000.

Tabel 4.7 *Statistics Summary* Atribut Kategorik

Atribut	***unique***	***top***	***freq***
Gender	2	1	163
Marital Status	5	2	67
Last Education	5	2	160
Education Field	18	16	43
Site	17	9	44
Department	8	5	106
Job Level	3	1	145
Job Role	19	11	40
Competency	4	1	92
Leave	2	0	156

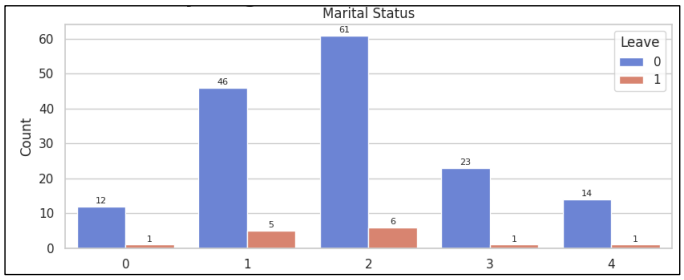
Tabel 4.7 menunjukan ringkasan statistik dari seluruh data kategorik yang digunakan. Kolom *unique* menunjukan berapa banyak nilai unik dalam atribut, atau bisa disebut berapa banyak kategori yang dimiliki oleh atribut. Kolom *top* menunjukan dari sekian banyak kategori yang dimiliki, maka kategori dalam kolom *top* adalah yang mendominasi dari semua kategori lain di dalam atribut. Sedangkan kolom *freq* adalah berapa banyak frekuensi atau kemunculan kategori yang menjadi *top*. Contoh pada atribut *Gender*, terdapat dua kategori unik (*unique* = 2) didalamnya yaitu Perempuan (0) dan Laki-Laki (1). Dimana, kategori Laki-Laki mendominasi (*top* = 1) dengan jumlah kemunculan 163 kali (*freq* = 163).

Selanjutnya, akan dianalisis terkait proporsi dari setiap kategori yang dimiliki oleh setiap atribut kategorik, terhadap variabel target. Seperti yang diketahui, bahwa kejadian *turnover* yang terdata dalam penelitian ini adalah 14 orang karyawan. Sehingga akan dianalisis karyawan dengan karakteristik apa, yang paling banyak menyumbangkan kejadian *turnover* tersebut. *Instance* dari variabel target diwakili dengan angka 0 yang berarti karyawan masih bertahan di perusahaan (*Leave = No*), dan angka 1 yang berarti karyawan telah pergi/meninggalkan perusahaan (*Leave = Yes*).



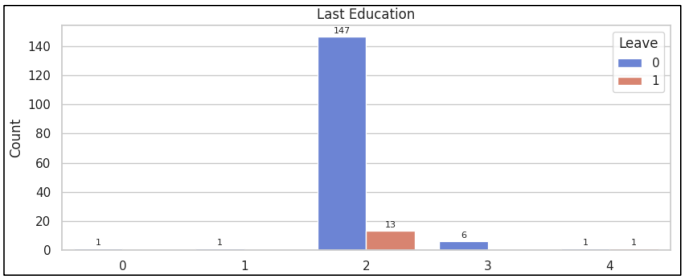
Gambar 4.3 Distribusi *Gender* terhadap Target

Dari Gambar 4.3, kejadian *turnover* berdasarkan atribut *Gender* sepenuhnya didominasi oleh karyawan laki-laki (1), dimana 14 diantara 163 karyawan laki-laki telah meninggalkan perusahaan. Sedangkan 7 orang karyawan perempuan (0) lainnya, semua masih bertahan.



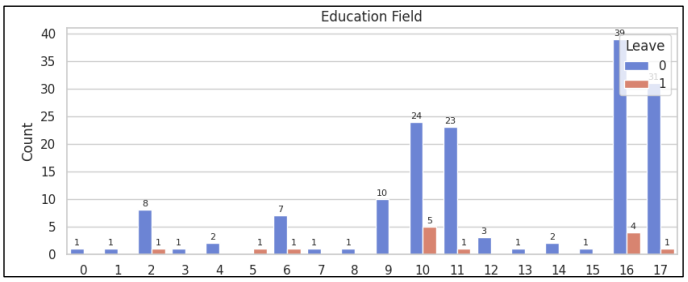
Gambar 4.4 Distribusi *Marital Status* terhadap Target

Gambar 4.4 menunjukan bahwa berdasarkan atribut *Marital Status*, karyawan didominasi oleh status pernikahan M2/Married2 (2) sebanyak 67 orang, dimana 6 diantaranya (8,95%) meninggalkan perusahaan. Disusul oleh M1/Married1 (1) sebanyak 51 orang, dimana 5 diantaranya (9,80%) meninggalkan perusahaan. Selanjutnya M3/Married3 (3) sebanyak 24 orang, dimana 1 diantaranya (4,17%) meninggalkan perusahaan. Status pernikahan S0/Single0 (4) berjumlah 15 orang, dimana 1 diantaranya (6,67%) meninggalkan perusahaan. Dan pada urutan terakhir yaitu status pernikahan M0/Married0 (0) sebanyak 13 orang, dimana 1 diantaranya (7,69%) meninggalkan perusahaan.



Gambar 4.5 Distribusi *Last Education* terhadap Target

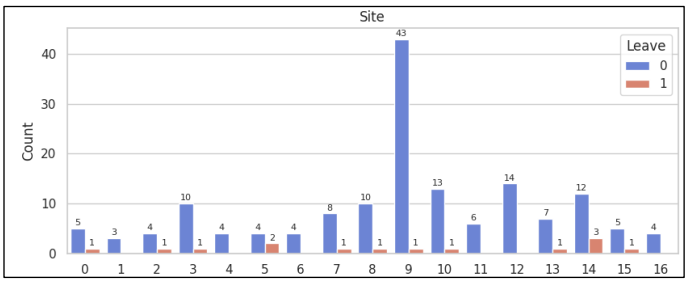
Berdasarkan Gambar 4.5, jika dilihat dari atribut *Last Education*, karyawan didominasi oleh tingkat pendidikan akhir S1 (2) sebanyak 160 orang, dimana 13 diantaranya (8,84%) meninggalkan perusahaan. Disusul oleh tingkat pendidikan akhir S2 (3), yang semuanya masih bertahan. Selanjutnya tingkat pendidikan akhir SMK (4) sebanyak 2 orang, dimana 1 diantaranya (50%) meninggalkan perusahaan. Pada urutan terakhir, tingkat pendidikan akhir D1 (0) dan D3 (0) masing-masing berjumlah 1, yang seluruhnya masih bertahan.



Gambar 4.6 Distribusi *Education Field* terhadap Target

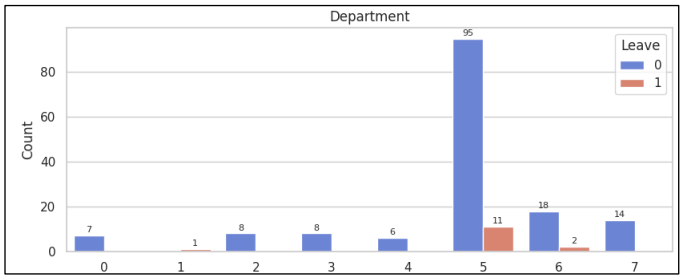
Gambar 4.6 menunjukan bahwa berdasarkan atribut *Education Field*, karyawan didominasi oleh bidang pendidikan Teknik Pertambangan (16) sebanyak 43 orang, dimana 4 diantaranya (9,30%) meninggalkan perusahaan. Disusul oleh bidang pendidikan Teknik Sipil (17) sebanyak 32 orang, dimana 1 diantaranya (3,12%) meninggalkan perusahaan. Kemudian bidang pendidikan Teknik Geologi (10) sebanyak 29 orang, dimana 5 diantaranya (17,24%) meninggalkan perusahaan. Diikuti bidang pendidikan Teknik Industri (11) sebanyak 24 orang, dimana 1 diantaranya (4,17%) meninggalkan perusahaan. Selanjutnya bidang pendidikan Teknik Geodesi (9) sebanyak 10 orang, yang semuanya masih bertahan. Bidang pendidikan Geologi (2) sebanyak 9 orang, dimana 1 diantaranya (11,11%) meninggalkan perusahaan. Bidang pendidikan Statistika (6) sebanyak 8 orang, dimana 1 diantaranya (12,5%) meninggalkan perusahaan. Bidang

pendidikan Teknik Informatika (12) sebanyak 3 orang, yang semuanya masih bertahan. Bidang pendidikan Manajemen Informatika (4) dan Teknik Otomotif (14) masing-masing berjumlah 2 orang, yang seluruhnya masih bertahan. Urutan terakhir adalah bidang pendidikan Akuntansi (0), Fisika (1), Geostatistik (3), Matematika (5), Tambang (7), Teknik Elektro (8), Teknik Lingkungan (13) dan Teknik Perminyakan (15) masing-masing berjumlah 1 orang, dan tidak ada karyawan yang meninggalkan perusahaan kecuali pada bidang pendidikan Matematika yaitu 1 dari 1 orang (100%) telah meninggalkan perusahaan.



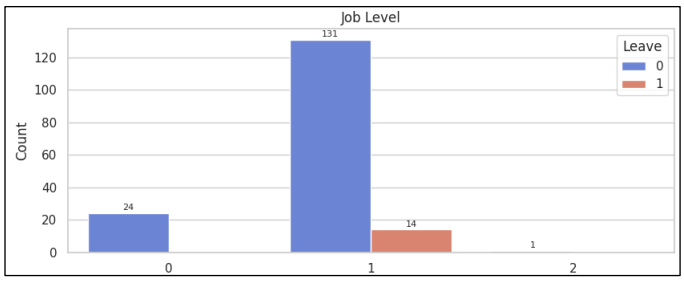
Gambar 4.7 Distribusi *Site* terhadap Target

Berdasarkan Gambar 4.7, jika dilihat dari atribut *Site*, karyawan didominasi oleh *site* 9 sebanyak 44 orang, dimana 1 diantaranya (2,27%) meninggalkan perusahaan. Disusul oleh *site* 14 sebanyak 15 orang, dimana 3 diantaranya (20%) meninggalkan perusahaan. Selanjutnya *site* 10 dan KPCS 12 masing-masing berjumlah 14 orang. Pada *site* 12 tidak ada karyawan yang meninggalkan perusahaan, sedangkan pada *site* 10 terdapat 1 dari 14 orang (7,14%) yang meninggalkan perusahaan. Disusul oleh *site* 3 dan 8 masing-masing berjumlah 11 orang, dan masing-masing 1 diantaranya (9,09%) meninggalkan perusahaan. Kemudian *site* 7 sebanyak 9 orang, dimana 1 diantaranya (11,11%) meninggalkan perusahaan. *Site* 13 sebanyak 8 orang, dimana 1 diantaranya (12,5%) meninggalkan perusahaan. *Site* 0, 5, 11, dan 15 masing-masing berjumlah 6 orang. Pada *site* 11, tidak ada karyawan yang meninggalkan perusahaan. Pada *site* 0 dan 15 masing-masing terdapat 1 dari 6 orang (16,67%) meninggalkan perusahaan. Sedangkan pada *site* 5 terdapat 2 dari 6 orang (33,33%) meninggalkan perusahaan. Kemudian *site* 2 sebanyak 5 orang, dimana 1 diantaranya (20%) meninggalkan perusahaan. Untuk *site* 4, 6 dan 16, masing-masing berjumlah 4 orang dan semuanya masih bertahan. Pada posisi terakhir terdapat *site* 1 sebanyak 3 orang dan semuanya masih bertahan.



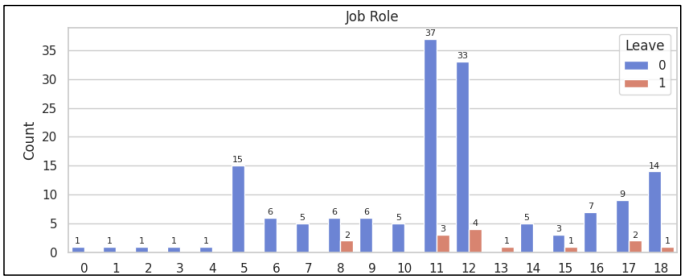
Gambar 4.8 Distribusi *Department* terhadap Target

Gambar 4.8 menunjukan bahwa karyawan didominasi oleh departemen 5 sebanyak 106 orang, dimana 11 diantaranya (10,38%) meninggalkan perusahaan. Lalu departemen 6 sebanyak 20 orang, dimana 2 diantaranya (11,11%) meninggalkan perusahaan. Kemudian departemen 7 sebanyak 14 orang, yang semuanya masih bertahan. Diikuti oleh departemen 2 dan 3 masing-masing berjumlah 8 orang dan seluruhnya masih bertahan. Kemudian departemen 0 sebanyak 7 orang, yang semuanya masih bertahan. Untuk departemen 4 sebanyak 6 orang dan semuanya masih bertahan. Posisi terakhir yaitu departemen 1 berjumlah 1 orang dan telah meninggalkan perusahaan.



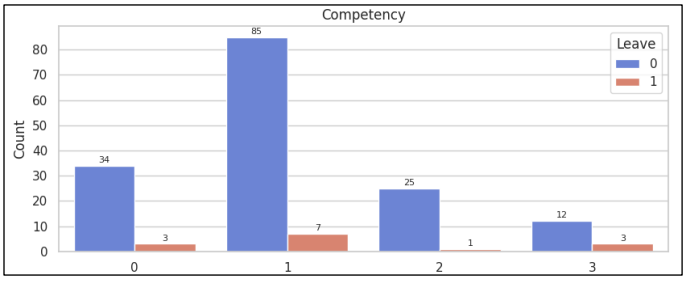
Gambar 4.9 Distribusi *Job Level* terhadap Target

Berdasarkan Gambar 4.9, karyawan didominasi oleh *Job Level* Pratama (1) sebanyak 145 orang, 14 diantaranya (10,69%) meninggalkan perusahaan. Disusul level Madya (0) sebanyak 24 orang dan Utama (2) sebanyak 1 orang, yang pada kedua level ini tidak ada karyawan yang meninggalkan perusahaan.



Gambar 4.10 Distribusi *Job Role* terhadap Target

Gambar 4.10 menunjukan bahwa berdasarkan atribut *Job Role*, karyawan didominasi oleh posisi 11 sebanyak 40 orang, dimana 3 diantaranya (7,5%) meninggalkan perusahaan. Disusul oleh posisi 12 sebanyak 37 orang, dimana 4 (10,81%) diantaranya meninggalkan perusahaan. Kemudian posisi 5 dan 18 masing-masing berjumlah 15 orang, dan hanya pada posisi 18 saja yang terdapat karyawan yang meninggalkan perusahaan yaitu 1 dari 15 orang (6,67%). Disusul oleh 17 sebanyak 11 orang, dimana 2 diantaranya (18,18%) meninggalkan perusahaan. Selanjutnya posisi 8 sebanyak 10 orang, dimana 2 diantaranya (20%) meninggalkan perusahaan. Posisi 16 sebanyak 7 orang yang semuanya masih bertahan. Untuk posisi 6 dan 9 masing-masing sebanyak 6 orang dan semuanya masih bertahan. Untuk posisi 7, 10 dan 14 masing-masing berjumlah 5 orang dan semuanya masih bertahan. Untuk posisi 15 sebanyak 4 orang, 1 diantaranya (25%) meninggalkan perusahaan. Terakhir untuk posisi 0, 1, 2, 3, 4 dan 13 masing-masing berjumlah satu, dan hanya pada posisi 13 saja yang terdapat karyawan yang meninggalkan perusahaan yaitu 1 dari 1 orang (100%).



Gambar 4.11 Distribusi *Competency* terhadap Target

Berdasarkan Gambar 4.11, jika dilihat dari atribut *Competency*, karyawan didominasi oleh kompetensi B (1) sebanyak 92 orang, dimana 7 diantaranya (7,61%) meninggalkan perusahaan. Disusul oleh kompetensi A (0) sebanyak 37 orang, dimana 3 diantaranya

(8,82%) meninggalkan perusahaan. Kemudian kompetensi C (2) sebanyak 26 orang, dimana 1 diantaranya (4%) meninggalkan perusahaan. Dan terakhir adalah kompetensi D (3) sebanyak 15 orang, dimana 3 diantaranya (20%) meninggalkan perusahaan.

4.4 Pembangunan Model Klasifikasi

Pembangunan model klasifikasi diawali dengan tahap membagi data (*splitting data*) menjadi data uji (*training*) dan data latih (*testing*). Data *training* akan digunakan untuk melatih model, sedangkan data *testing* digunakan untuk mengevaluasi model. Pada penelitian ini, pembagian data dilakukan dengan metode *StratifiedKFold*. Metode ini melakukan teknik validasi silang yang membagi data menjadi ‘k’ lipatan atau bagian.

Tabel 4.8 Lipatan *Splitting Data*

Fold 1	***Fold 2***	***Fold 3***	***Fold 4***	***Fold 5***
X_train shape: (136, 79) *y_train shape: (136,)* *X_test shape: (34, 79)* *y_test shape: (34,)*	*X_train shape: (136, 79)* *y_train shape: (136,)* *X_test shape: (34, 79)* *y_test shape: (34,)*	*X_train shape: (136, 79)* *y_train shape: (136,)* *X_test shape: (34, 79)* *y_test shape: (34,)*	*X_train shape: (136, 79)* *y_train shape: (136,)* *X_test shape: (34, 79)* *y_test shape: (34,)*	*X_train shape: (136, 79)* *y_train shape: (136,)* *X_test shape: (34, 79)* *y_test shape: (34,)*

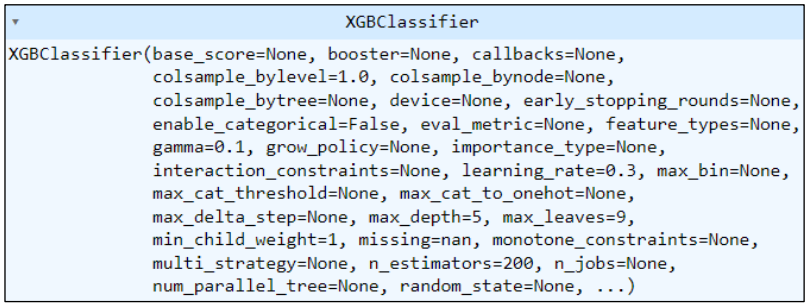
Tabel 4.9 Proporsi Label Kelas Setelah *Splitting Data*

Sampel	**Jumlah Sampel**
Kelas positif dalam *training set*	11
Kelas negatif dalam *training set*	125
Kelas positif dalam *test set*	3
Kelas negatif dalam *test set*	31

Tabel 4.8 dan Tabel 4.9 menunjukkan hasil *splitting* dengan metode *StratifiedKFold* 5 lipatan (*fold*). Pada tahap ini, data awal terdiri dari 170 sampel dengan 79 kolom. Setelah *splitting*, data *training* (*X_train*) terdiri dari 136 sampel (11 sampel positif dan 125 sampel negatif), sedangkan data uji (*X_test*) memiliki 34 sampel (3 sampel positif dan 31 sampel negatif). Dari Tabel 4.9, terbukti bahwa *StratifiedKFold* memastikan setiap lipatan memiliki distribusi kelas yang sama seperti dataset asli. Cara ini bertujuan agar setiap lipatan memiliki representasi yang baik dari semua kelas. Kemudian setiap sampel terdiri dari 79 kolom (terdiri dari atribut dan label/target) sesuai dengan jumlah sampel di setiap himpunan data. Data yang sudah dibagi menjadi *training set* dan *testing set* inilah yang kemudian digunakan untuk tahapan selanjutnya dalam pembangunan model klasifikasi.

4.4.1 Klasifikasi XGBoost tanpa SMOTE

Pada klasifikasi XGBoost tanpa menggunakan SMOTE, data uji (*training set*) akan langsung digunakan untuk pelatihan model. Langkah awal yang dilakukan dalam penelitian ini adalah mencari nilai optimal *hyperparameter* (*hyperparameter tuning*) secara otomatis menggunakan *RandomizedSearchCV*. Dari beberapa nilai parameter yang didefinisikan pada Tabel 2.7 Parameter Klasifikasi, nilai optimal *hyperparameter* yang didapatkan untuk XGBoost tanpa SMOTE dengan *best K (cross validation)* sebesar 2 adalah *subsample* = 1,0; *scale_pos_weight* = 11,36; *reg_lambda* = 1; *reg_alpha* = 0,1; *n_estimators* = 200; *min_child_weight* = 1; *max_leaves* = 9; *max_depth* = 5; *learning_rate* = 0,3; *gamma* = 0,1; *colsample_bylevel* = 1,0 dengan *best score* = 0,92. Nilai 11,36 yang didapatkan pada parameter *scale_pos_weight* sesuai dengan pengaturan *(len(y_train) - y_train.sum()) / y_train.sum()* yaitu (136-11)/11=11,36. Nilai-nilai terbaik dari proses *tuning* tersebut kemudian digunakan dalam pelatihan model klasifikasi XGBoost tanpa SMOTE.



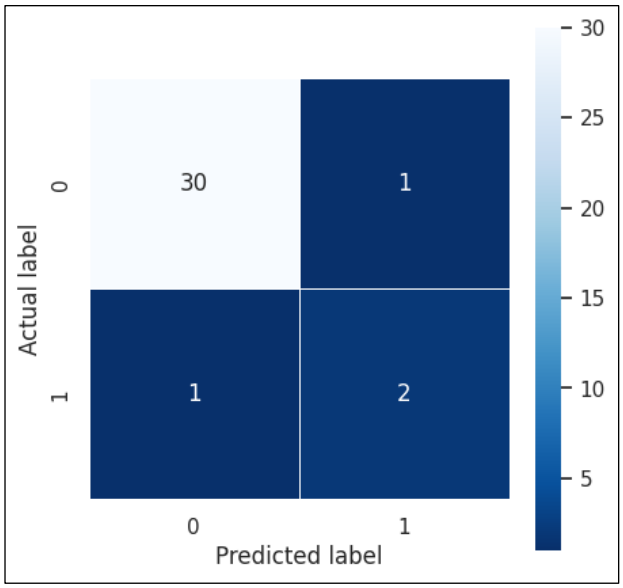
Gambar 4.12 Konfigurasi Pelatihan Model XGBoost tanpa SMOTE

Gambar 4.12 memberikan informasi mengenai hasil konfigurasi dari model XGBoost yang telah dilatih. Berikut ini merupakan penjelasannya.

1. *n_estimators* = 200. Model menggunakan 200 pohon keputusan. Setiap pohon mempelajari sebagian kecil dari data dan memberikan prediksinya. Prediksi dari semua pohon kemudian digabungkan untuk menghasilkan prediksi akhir.
2. *max_depth* = 5. Setiap pohon dalam model dapat memiliki kedalaman maksimum 5, yang berarti setiap pohon dapat memiliki hingga 5 tingkat pertanyaan sebelum mencapai keputusan akhir.
3. *learning_rate* = 0,3. Nilai 0,3 berarti model Anda melakukan langkah-langkah cukup besar saat memperbarui bobotnya.

4. *min_child_weight* = 1. Nilai 1 berarti bahwa setiap simpul harus memiliki setidaknya satu sampel sebelum dapat dibagi lebih lanjut. Hal ini dapat membantu mencegah model membagi simpul berdasarkan fitur yang hanya relevan untuk sejumlah kecil sampel, yang dapat menyebabkan *overfitting*.
5. *gamma* = 0,1. Nilai 0,1 berarti bahwa penurunan dalam fungsi kerugian harus setidaknya 0,1 sebelum simpul dapat dibagi lebih lanjut. Hal ini dapat membantu mencegah model membagi simpul berdasarkan fitur yang tidak memberikan peningkatan signifikan dalam kinerja model, yang dapat membantu mencegah *overfitting*.
6. *colsample_bylevel* = 1,0. Nilai 1,0 berarti bahwa semua fitur dapat dipilih pada setiap tingkat pohon. Sehingga model memiliki kebebasan penuh untuk memilih fitur mana pun yang paling informatif pada setiap tingkat pohon.
7. *max_leaves* = 9. Nilai 9 berarti bahwa setiap pohon dalam model dapat memiliki hingga 10 simpul akhir. Ini membatasi kompleksitas model dan dapat membantu mencegah *overfitting*.

Setelah melakukan pelatihan model dengan memanfaatkan data uji (*training*), selanjutnya dilakukan evaluasi model yang memanfaatkan data latih (*testing*). Evaluasi model ini akan diukur menggunakan *confusion matrix* dan AUC-ROC.



Gambar 4.13 *Confusion Matrix* Model XGBoost tanpa SMOTE

Confusion Matrix pada Gambar 4.13 memberikan gambaran mengenai bagaimana ketepatan model klasifikasi XGBoost tanpa SMOTE dalam memprediksi data latih. Interpretasi dari *confusion matrix* yang didapatkan tersebut adalah sebagai berikut.

1. *True Positives* (TP). Model dengan benar memprediksi bahwa karyawan akan meninggalkan perusahaan. Terlihat bahwa TP = 2.
2. *True Negatives* (TN): Model dengan benar memprediksi bahwa karyawan akan bertahan di perusahaan. Terlihat bahwa TN = 30.
3. *False Positives* (FP). Model salah memprediksi bahwa karyawan akan meninggalkan perusahaan. Padahal sebenarnya, karyawan tersebut bertahan. Terlihat bahwa FP = 1.
4. *False Negatives* (FN). Model salah memprediksi bahwa karyawan akan bertahan di perusahaan. Padahal sebenarnya, karyawan tersebut meninggalkan perusahaan. Terlihat bahwa FN = 1.

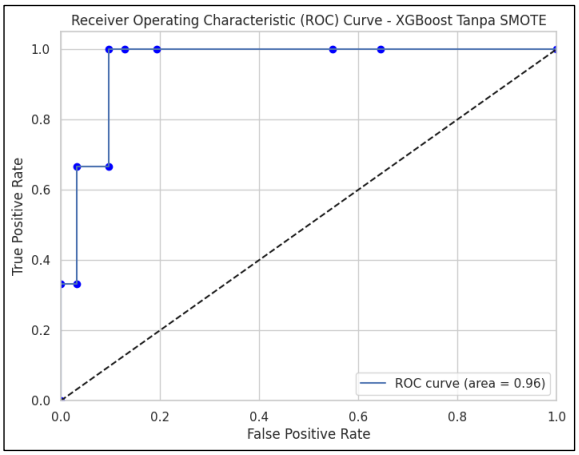
Berdasarkan nilai-nilai *confusion matrix*, diketahui bahwa total data latih dengan label kelas positif (*Leave = Yes*) adalah 3 ($TP + FN = 2 + 1 = 3$) dan label kelas negatif (*Leave = No*) adalah 31 ($TN + FP = 30 + 1 = 31$). Kemudian dapat dihitung beberapa metrik evaluasi model klasifikasi seperti yang terlihat pada Tabel 4.10 dengan perhitungan berikut.

1. *Accuracy* $= \frac{TP+TN}{Total} = \frac{2+30}{34} = \frac{32}{34} = 0{,}94$
2. *Misclassification Rate* $= \frac{FP+FN}{Total} = \frac{1+1}{34} = \frac{2}{34} = 0{,}06$
3. *Precision (negative)* $= \frac{TN}{TN+FN} = \frac{30}{30+1} = \frac{30}{31} = 0{,}97$
4. *Precision (positive)* $= \frac{TP}{TP+FP} = \frac{2}{2+1} = \frac{2}{3} = 0{,}67$
5. *Recall (negative)* $= \frac{TN}{TN+FP} = \frac{30}{30+1} = \frac{30}{31} = 0{,}97$
6. *Recall (positive)* $= \frac{TP}{TP+FN} = \frac{2}{2+1} = \frac{2}{3} = 0{,}67$
7. *F1-Score (negative)* $= 2 \times \frac{Presisi(negative) \times Recall(negative)}{Presisi(negative)+Recall(negative)} = 2 \times \frac{0{,}97 \times 0{,}97}{0{,}97+0{,}97} = 0{,}97$
8. *F1-Score (positive)* $= 2 \times \frac{Presisi(positive) \times Recall(positive)}{Presisi(positive)+Recall(positive)} = 2 \times \frac{0{,}67 \times 0{,}67}{0{,}67+0{,}67} = 0{,}67$

Tabel 4.10 Performa Model XGBoost tanpa SMOTE

Class	*Precision*	*Recall*	*F1-Score*	*Support*
0	0,97	0,97	0,97	31
1	0,67	0,67	0,67	3
Accuracy			**0,94**	**34**

Untuk metrik AUC pada penelitian ini, didapat dengan memanfaatkan fungsi *roc_auc_score* dalam pustaka *scikit-learn* di *Python*. Nilai AUC yang didapatkan adalah sebesar 0,96. Kurva ROC dapat dilihat pada Gambar 4.14.



Gambar 4.14 Kurva ROC XGBoost tanpa SMOTE

Berdasarkan nilai-nilai metrik yang telah didapatkan, dapat diketahui bahwa model XGBoost tanpa SMOTE ini efektif dalam memprediksi karyawan yang akan tetap di perusahaan (kelas negatif atau 0), dengan tingkat akurasi yang cukup tinggi. Namun, kemampuan model dalam memprediksi karyawan yang akan keluar dari perusahaan (kelas positif atau 1) masih perlu ditingkatkan. Sehingga dilakukanlah usaha penyeimbangan proporsi label kelas dengan melanjutkan analisis ke tahap klasifikasi XGBoost dengan SMOTE.

4.4.2 Klasifikasi XGBoost dengan SMOTE

Berbeda dengan klasifikasi XGBoost tanpa SMOTE sebelumnya, pada klasifikasi XGBoost dengan menggunakan SMOTE, data uji (*training set*) akan dilakukan penyeimbangan proporsi kelas data terlebih dahulu. Terlihat pada Tabel 4.11 Penerapan SMOTE pada *Training Set*, bahwa SMOTE hanya diterapkan pada data uji, sehingga total

data dari yang sebelumnya sebanyak 136 sampel menjadi 250 sampel, dengan jumlah kolom yang tetap sama yaitu 79 kolom. Jumlah data pada data latih (*testing set*) tidak diterapkan SMOTE sehingga tidak mengalami perubahan (tetap 34 baris). Metode SMOTE menyeimbangkan proporsi label kelas data, sehingga data hasil SMOTE pada penelitian ini akan memiliki 125 sampel negatif dan 125 sampel positif seperti yang terlihat pada Tabel 4.12.

Tabel 4.11 Penerapan SMOTE pada *Training Set*

Data	***Data Shape Before SMOTE***	***Data Shape After SMOTE***
X_train	*(136, 79)*	*(250, 79)*
y_train	*(136,)*	*(250,)*
X_test	*(34, 79)*	*(34, 79)*
y_test	*(34,)*	*(34,)*

Tabel 4.12 Proporsi Label Kelas pada *Training Set* Setelah SMOTE

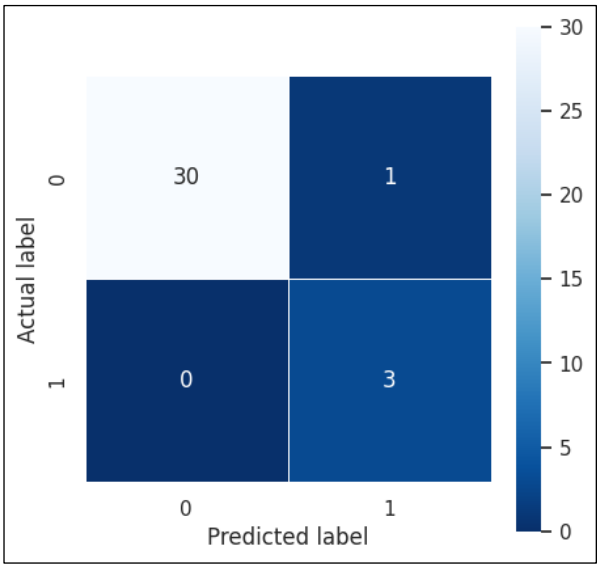
Label Kelas	**Jumlah**
0 (*negative*)	125
1 (*positive*)	125
Name: Leave, dtype: int64	

Untuk melihat apakah SMOTE signifikan dalam memvariasikan data, dilakukanlah pengujian *Kolmogorov-Smirnov*. Uji ini merupakan uji non-parametrik yang digunakan untuk membandingkan dua distribusi probabilitas. Dalam konteks SMOTE pada penelitian ini, uji *Kolmogorov-Smirnov* digunakan untuk membandingkan distribusi data asli dan data sintetis yang dihasilkan oleh SMOTE. Dengan uji *Kolmogorov-Smirnov,* didapatkan hasil statistik 0,02664 dan *p-value* 0,000100511. Nilai tersebut menunjukkan bahwa terdapat perbedaan signifikan antara distribusi data asli dan data sintetis (*p-value* > 0,05). Selain dengan melakukan uji *kolmogorov-smirnov,* metrik evaluasi *F1-Score* juga dapat dijadikan acuan untuk melihat apakah SMOTE membantu dalam meningkatkan variabilitas data. Ketika *F1-Score* model klasifikasi yang diintegrasikan SMOTE memiliki nilai lebih tinggi dibandingkan *F1-Score* model klasifikasi tanpa SMOTE, maka SMOTE dapat membantu dalam meningkatkan variabilitas data.

Setelah penerapan SMOTE selesai, langkah analisis berikutnya adalah *hyperparameter tuning*. Dari beberapa nilai parameter yang didefinisikan pada Tabel 2.7 Parameter Klasifikasi, nilai optimal *hyperparameter* yang didapatkan untuk XGBoost dengan SMOTE dimana *best K (cross validation)* sebesar 4 adalah *subsample* = 1,0;

scale_pos_weight = 1,0; *reg_lambda* = 1; *reg_alpha* = 0,1; *n_estimators* = 200; *min_child_weight* = 1; *max_leaves* = 9; *max_depth* = 5; *learning_rate* = 0,3; *gamma* = 0,1; *colsample_bylevel* = 1,0 dengan *best score* = 0,99.

Terlihat bahwa nilai optimal hasil *tuning* hampir sama dengan XGBoost tanpa SMOTE sebelumnya, dengan perbedaan hanya terletak pada nilai parameter *scale_pos_weight* dan *best score*-nya saja. Hal ini mungkin dikarenakan terdapat beberapa *hyperparameter* yang memiliki pengaruh lebih besar (terhadap kinerja model) daripada *hyperparameter* yang lain. Sehingga penambahan sampel sintetis oleh SMOTE hanya mempengaruhi parameter *scale_pos_weight* dan peningkatan *best score* hasil *tuning*. Parameter *scale_pos_weight* yang bernilai 1 sesuai dengan pengaturan *(len(y_train) - y_train.sum()) / y_train.sum()* yaitu (250-125)/125=1. Namun meskipun nilai optimal *hyperparameter* untuk kedua model adalah sama, tidak menjamin bahwa kinerja model yang dihasilkan juga akan sama. Hal ini terbukti pada evaluasi model yang akan dijelaskan selanjutnya.



Gambar 4.15 *Confusion Matrix* Model XGBoost dengan SMOTE

Berikut merupakan interpretasi dari *confusion matrix* pada Gambar 4.15 *Confusion Matrix* Model XGBoost dengan SMOTE, yang didapatkan dari pelatihan model XGBoost dengan SMOTE.

1. *True Positives* (TP). Model dengan benar memprediksi bahwa karyawan akan meninggalkan perusahaan. Terlihat bahwa TP = 3.

2. *True Negatives* (TN): Model dengan benar memprediksi bahwa karyawan akan bertahan di perusahaan. Terlihat bahwa TN = 30.
3. *False Positives* (FP). Model salah memprediksi bahwa karyawan akan meninggalkan perusahaan. Padahal sebenarnya, karyawan tersebut bertahan. Terlihat bahwa FP = 1.
4. *False Negatives* (FN). Model salah memprediksi bahwa karyawan akan bertahan di perusahaan. Padahal sebenarnya, karyawan tersebut meninggalkan perusahaan. Terlihat bahwa FN = 0.

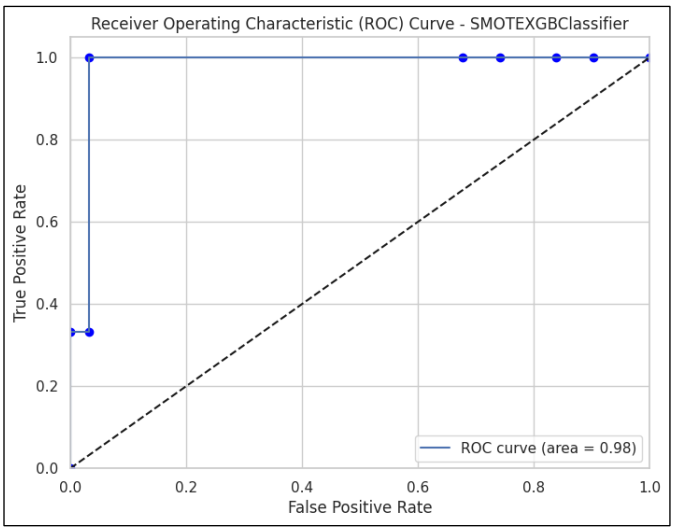
Berdasarkan nilai-nilai *confusion matrix* di atas, diketahui bahwa total data latih dengan label kelas positif (*Leave = Yes*) adalah 3 ($TP + FN = 3 + 0 = 3$) dan label kelas negatif (*Leave = No*) adalah 31 ($TN + FP = 30 + 1 = 31$). Kemudian dihitung beberapa metrik evaluasi model klasifikasi seperti yang terlihat pada Tabel 4.13 dengan perhitungan berikut.

1. *Accuracy* $= \frac{TP+TN}{Total} = \frac{3+30}{3+30+1+0} = \frac{33}{34} = 0{,}97$
2. *Misclassification Rate* $= \frac{FP+FN}{Total} = \frac{1+0}{34} = \frac{1}{34} = 0{,}03$
3. *Precision (negative)* $= \frac{TN}{TN+FN} = \frac{30}{30+0} = \frac{30}{30} = 1{,}00$
4. *Precision (positive)* $= \frac{TP}{TP+FP} = \frac{3}{3+1} = \frac{3}{4} = 0{,}75$
5. *Recall (negative)* $= \frac{TN}{TN+FP} = \frac{30}{30+1} = \frac{30}{31} = 0{,}97$
6. *Recall (positive)* $= \frac{TP}{TP+FN} = \frac{3}{3+0} = \frac{3}{3} = 1{,}00$
7. *F1-Score (negative)* $= 2 \times \frac{Presisi(negative) \times Recall(negative)}{Presisi(negative)+Recall(negative)} = 2 \times \frac{1{,}00 \times 0{,}97}{1{,}00+0{,}97} = 0{,}98$
8. *F1-Score (positive)* $= 2 \times \frac{Presisi(positive) \times Recall(positive)}{Presisi(positive)+Recall(positive)} = 2 \times \frac{0{,}75 \times 1{,}00}{0{,}75+1{,}00} = 0{,}86$

Tabel 4.13 Performa Model XGBoost dengan SMOTE

Class	***Precision***	***Recall***	***F1-Score***	***Support***
0	1,00	0,97	0,98	31
1	0,75	1,00	0,86	3
Accuracy			**0,97**	**34**

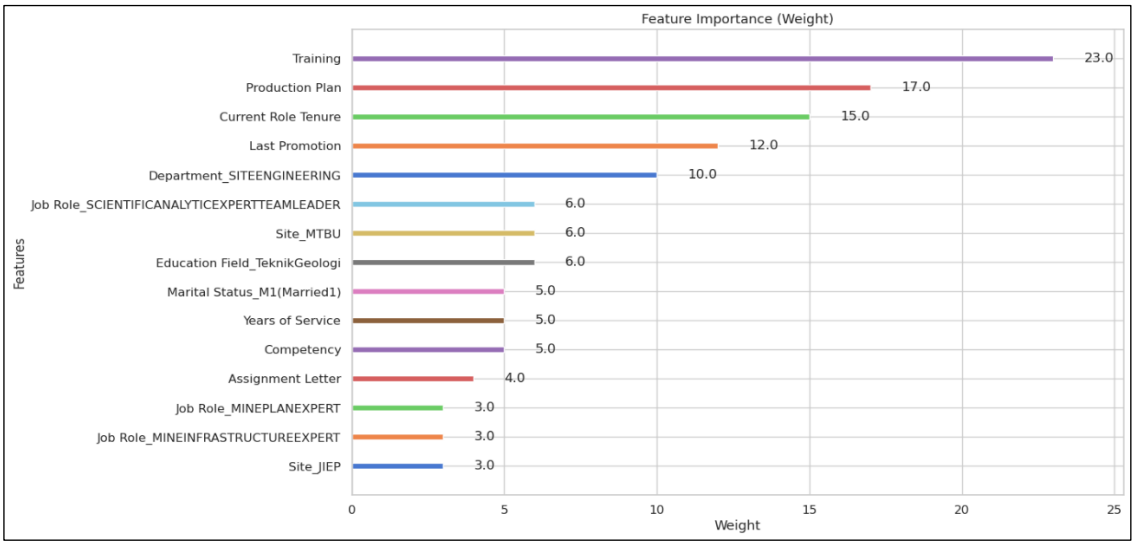
Nilai AUC yang didapatkan untuk klasifikasi XGBoost dengan SMOTE adalah sebesar 0,98. Dengan kurva ROC yang dapat dilihat pada Gambar 4.16 Kurva ROC XGBoost dengan SMOTE.



Gambar 4.16 Kurva ROC XGBoost dengan SMOTE

4.5 *Features Importance* berdasarkan Model Terbaik

Proses melihat *features importance* dilakukan untuk mengidentifikasi atribut apa saja yang paling berkontribusi terhadap kejadian *turnover*. Tahapan ini didasarkan pada algoritma yang terbaik yaitu XGBoost dengan SMOTE, untuk mengevaluasi kepentingan setiap fitur dalam membuat prediksi berdasarkan algoritma tersebut.



Gambar 4.17 *Features Importance*

Gambar 4.17 *Features Importance* menunjukan 15 atribut terpenting dalam kasus klasifikasi *turnover* pada penelitian ini. Hal tersebut menunjukkan bahwa 15 atribut terpenting ini memiliki pengaruh lebih besar terhadap keputusan karyawan untuk bertahan atau meninggalkan perusahaan, dibandingkan dengan atribut-atribut lainnya. Atribut *Training* dengan bobot sebesar 23,0 menjadi atribut paling penting dalam memprediksi *turnover*. Hal tersebut menunjukkan bahwa jumlah pelatihan yang telah selesai diikuti karyawan memiliki pengaruh besar terhadap keputusan karyawan untuk bertahan atau meninggalkan perusahaan.

Disusul pada empat urutan selanjutnya yaitu *Production Plan* (bobot 17,0); *Current Role Tenure* (bobot 15,0); *Last Promotion* (bobot 12,0) dan *Department_SITEENGINEERING* (bobot 10,0). Secara keseluruhan, plot ini memberikan wawasan mengenai atribut mana yang paling berpengaruh dalam memprediksi *turnover* karyawan, yang dapat sangat berguna untuk perusahaan dalam merancang strategi retensi karyawan.

BAB V

PEMBAHASAN

5.1 Perbandingan Performa Model

Berikut merupakan perbandingan performa model XGBoost tanpa dan dengan SMOTE, untuk melihat model mana yang memberikan hasil terbaik dalam kasus penelitian ini.

Tabel 5.1 Perbandingan Performa Model

Metrik Evaluasi	XGBoost Tanpa SMOTE	XGBoost Dengan SMOTE
Accuracy	0,94	0,97
Misclassification Rate	0,06	0,03
Precision (negative)	0,97	1,00
Precision (positive)	0,67	0,75
Recall (negative)	0,97	0,97
Recall (positive)	0,67	1,00
F1-Score (negative)	0,97	0,98
F1-Score (positive)	0,67	0,86
AUC	0,96	0,98

Berdasarkan Tabel 5.1 Perbandingan Performa Model, dapat diketahui bahwa model klasifikasi XGBoost dengan SMOTE memiliki peningkatan kinerja yang signifikan dibandingkan dengan model XGBoost tanpa SMOTE. Interpretasi tiap metrik evaluasi dijelaskan pada bagian berikut.

1. Akurasi meningkat dari 0,94 menjadi 0,97. Peningkatan yang terjadi menunjukkan bahwa model XGBoost dengan SMOTE lebih efektif dalam memprediksi, baik karyawan yang akan tetap maupun yang akan meninggalkan perusahaan.
2. Tingkat kesalahan klasifikasi menurun dari 0,06 menjadi 0,03. Penurunan ini menunjukan bahwa model XGBoost dengan SMOTE melakukan lebih sedikit kesalahan dalam prediksinya.
3. Presisi meningkat untuk kelas negatif dari 0,97 menjadi 1,00 dan untuk kelas positif dari 0,67 menjadi 0,75. Peningkatan ini menunjukkan bahwa model XGBoost dengan SMOTE lebih akurat dalam memprediksi kedua kelas.
4. *Recall* untuk kelas negatif tidak mengalami perubahan, sementara *recall* untuk kelas positif meningkat secara signifikan dari 0,67 menjadi 1,00. Hal ini menunjukkan bahwa model XGBoost dengan SMOTE dapat mengidentifikasi semua kelas positif (karyawan meninggalkan perusahaan) dengan benar.

5. *F1-Score* meningkat untuk kelas negatif dari 0,97 menjadi 0,98 dan kelas positif dari 0,67 menjadi 0,86. Peningkatan ini menunjukkan bahwa model XGBoost dengan SMOTE memiliki keseimbangan lebih baik antara presisi dan *recall*.
6. AUC meningkat dari 0,96 menjadi 0,98 menunjukkan bahwa model XGBoost dengan SMOTE memiliki performa yang sangat baik dalam membedakan antara karyawan yang akan bertahan dan yang akan meninggalkan perusahaan.

Model XGBoost tanpa SMOTE memang tampaknya memiliki performa yang sudah bagus jika dilihat hanya berdasarkan pada metrik akurasi saja (0,94 atau 94%). Namun jika dilihat berdasarkan metrik-metrik evaluasi lainnya, diketahui bahwa model melakukan pekerjaan yang baik dalam memprediksi kasus negatif tetapi kurang efektif dalam memprediksi kasus positif. Hal ini terbukti dari nilai *recall* positif yang masih rendah yaitu hanya sebesar 0,67 atau 67% saja. Dan jika dilihat pada nilai *F1-Score* positif yang cenderung masih rendah (0,67 atau 67%), menandakan bahwa model XGBoost tanpa SMOTE masih mengalami kesulitan dalam menyeimbangkan presisi dan *recall* untuk kelas positif yang nilainya cukup berbeda jauh. Hasil ini sesuai dengan penelitian Sholikhati (2022), yang menjelaskan bahwa masalah *imbalance data* dapat mempengaruhi kinerja dari model dalam memprediksi kelas yang minoritas. Hasil penelitian ini juga sejalan dengan penelitian Kovvuri & Dommeti (2022), yang menyatakan bahwa metrik akurasi saja dinilai tidak valid dalam kasus data yang tidak seimbang. Jika hanya menggunakan metrik akurasi saja, maka tidak dapat membedakan antara jumlah contoh yang diklasifikasikan dengan benar dari kelas yang berbeda, dan dapat menyebabkan kesimpulan yang salah.

Pada model XGBoost dengan SMOTE, dilakukan penyeimbangan proporsi label kelas, sebagai usaha menyeimbangkan performa model dalam memprediksi kasus positif dan negatif. Hasilnya, model XGBoost dengan SMOTE mengalami peningkatan performa yang sangat baik dengan akurasi sebesar 0,97 atau 97% dan nilai AUC sebesar 0,98 atau 98%. Tingginya nilai AUC ini menandakan bahwa model memiliki kemampuan yang baik dalam membedakan kelas positif dan negatif. Namun, perlu diperhatikan bahwa akan selalu ada *trade-off* antara presisi dan *recall*. Dari hasil yang didapatkan, terdapat *trade-off* antara presisi dan *recall,* terutama pada kelas positif. Nilai 0,75 atau 75% pada presisi positif menandakan bahwa dari semua prediksi positif yang dibuat oleh model, yang benar-benar positif adalah 75%. Sedangkan *recall* positif memiliki nilai yang

sempurna yaitu 1,00 atau 100% yang artinya model berhasil mengidentifikasi semua kasus positif. Nilai presisi yang lebih rendah dari *recall* tadi menunjukan bahwa model mungkin menghasilkan lebih banyak *False Positive* (*Type I Error*), dimana model memprediksi kasus sebagai positif ketika sebenarnya adalah negatif.

Jika dipertimbangkan pada konteks kasus penelitian, dimana *turnover* pada PT. PNR menimbulkan *loss cost* yang tinggi dari sisi investasi pengembangan karyawan, maka akan menjadi lebih penting untuk berusaha mengurangi *Type II Error* (*false negative*/FN). Maksud dari *Type II Error* dalam kasus ini adalah model memprediksi karyawan akan bertahan di perusahaan, namun sebenarnya meninggalkan perusahaan (model tidak mengidentifikasi karyawan yang sebenarnya akan pergi). Dengan mengurangi *Type II Error*, model akan lebih baik dalam mengidentifikasi karyawan yang mungkin akan pergi, sehingga perusahaan dapat mengambil tindakan retensi yang tepat dan mencegah *turnover* tersebut. *Recall* positif yang tinggi dapat menjadi acuan bahwa model berhasil mengurangi *Type II Error*, dan ini terjadi pada model XGBoost dengan SMOTE.

Karena akan selalu adanya *trade-off* antara presisi dan *recall*, maka penting untuk melihat rata-rata harmonik keduanya atau *F1-score*. Model XGBoost dengan SMOTE memiliki *F1-Score* yang tinggi untuk kelas negatif maupun positif (0,98 dan 0,86), yang menunjukkan bahwa model XGBoost dengan SMOTE memiliki keseimbangan antara presisi dan *recall* kelas negatif maupun positif yang lebih baik dibandingkan dengan model XGBoost tanpa SMOTE. Hal ini menunjukkan bagaimana pentingnya penanganan ketidakseimbangan kelas dalam pembuatan model prediktif. Hasil yang didapat dalam penelitian ini sejalan dengan penelitian oleh Syukron et al. (2020), yang menyatakan bahwa metode SMOTE mampu memperbaiki kinerja model, dimana model dapat memprediksi secara akurat pada semua kelas respon.

Berdasarkan analisis tersebut, diketahui bahwa pada kasus dalam penelitian ini, model XGBoost yang diintegrasikan dengan SMOTE dipilih sebagai model yang lebih baik, dibandingkan dengan XGBoost tanpa SMOTE. Hasil akhir pemilihan model dalam penelitian ini memiliki kesamaan dengan penelitian terdahulu yang menyatakan bahwa integrasi SMOTE dengan XGBoost dapat mencapai hasil optimasi yang handal, serta memiliki kinerja model tertinggi diantara model lainnya yang dijadikan pembanding dalam penelitian (Ke et al., 2022; Mardiansyah et al., 2021; Syukron et al., 2020).

5.2 Pengecekan *Overfitting*

Pada penelitian ini, usaha pencegahan *overfitting* model dilakukan dengan penerapan SMOTE dan penyetelan *hyperparameter* dengan metode *cross validation*. Kemudian pengecekan *overfitting* dilakukan dengan dua cara, yaitu membandingkan performa yang didapatkan antara data *training* dan data *testing* serta mengevaluasi kinerja model pada data ekstrim. Untuk cara yang pertama yaitu membandingkan performa yang didapatkan antara data *training* dan data *testing*, beberapa percobaan kombinasi *hyperparameter* dilakukan, untuk melihat apakah kombinasi akhir yang digunakan dalam penelitian ini merupakan kombinasi yang terbaik dalam memperbaiki kinerja model dan mengurangi *overfitting*. Dari banyaknya percobaan kombinasi yang dilakukan, ditampilkan lima percobaan kombinasi *hyperparameter* pada model XGBoost tanpa SMOTE dan XGBoost dengan SMOTE, yang secara berurutan dapat dilihat pada Tabel 5.2 dan Tabel 5.3 di bawah ini.

Tabel 5.2 Percobaan Kombinasi *Hyperparameter* XGBoost tanpa SMOTE

Percobaan	Kombinasi dan Performa
Percobaan 1	***Kombinasi Hyperparameter*** *subsample* = 1,0; *scale_pos_weight* = 11,36; *reg_lambda* = 0,2; *reg_alpha* = 0,4; *n_estimators* = 300; *min_child_weight* = 7; *max_leaves* = 4; *max_depth* = 5; *learning_rate* = 0,2; *gamma* = 1,5; *colsample_bylevel* = 1,0 **Performa *Training Set*** *Accuracy* = 0,9044; *Precision* = 0,4583; *Recall* = 1,0000; *F1-Score* = 0,6286; ROC AUC = 0,9942 **Performa *Test Set*** *Accuracy* = 0,8529; *Precision* = 0,3750; *Recall* = 1,0000; *F1-Score* = 0,5455; ROC AUC = 0,9570 **Selisih Performa** *Accuracy* = 0,0515; *Precision* = 0,0833; *Recall* = 0,0000; *F1-Score* = 0,0831; ROC AUC = 0,0372
Percobaan 2	***Kombinasi Hyperparameter*** *subsample* = 1,0; *scale_pos_weight* = 11,364; *reg_lambda* = 0,5; *reg_alpha* = 0,1; *n_estimators* = 200; *min_child_weight* = 6; *max_leaves* = 5; *max_depth* = 6; *learning_rate* = 0,3; *gamma* = 0,2; *colsample_bylevel* = 1,0 **Performa *Training Set*** *Accuracy* = 0,9779; *Precision* = 0,7857; *Recall* = 1,0000; *F1-Score* = 0,8800; ROC AUC = 0,9985 **Performa *Test Set*** *Accuracy* = 0,8824; *Precision* = 0,4000; *Recall* = 0,6667; *F1-Score* = 0,5000; ROC AUC = 0,9462 **Selisih Performa** *Accuracy* = 0,0955; *Precision* = 0,3857; *Recall* = 0,3333; *F1-Score* = 0,3800; ROC AUC = 0,0523
Percobaan 3	***Kombinasi Hyperparameter*** *subsample* = 1,0; *scale_pos_weight* = 11,364; *reg_lambda* = 0,8; *reg_alpha* = 0,2; *n_estimators* = 300; *min_child_weight* = 1; *max_leaves* = 3; *max_depth* = 6; *learning_rate* = 0,3; *gamma* = 1,0; *colsample_bylevel* = 1,0 **Performa *Training Set*** *Accuracy* = 0,9706; *Precision* = 0,7333; *Recall* = 1,0000; *F1-Score* = 0,8462; ROC AUC = 1,0000 **Performa *Test Set*** *Accuracy* = 0,8529; *Precision* = 0,2500; *Recall* = 0,3333; *F1-Score* = 0,2857; ROC AUC = 0,9355 **Selisih Performa** *Accuracy* = 0,1177; *Precision* = 0,4833; *Recall* = 0,6667; *F1-Score* = 0,5605; ROC AUC = 0,0645
Percobaan 4	***Kombinasi Hyperparameter*** *subsample* = 0,75; *scale_pos_weight* = 11,364; *reg_lambda* = 0,30000000000000004; *reg_alpha* = 0,6; *n_estimators* = 100; *min_child_weight* = 6; *max_leaves* = 5; *max_depth* = 8; *learning_rate* = 0,05; *gamma* = 1,0; *colsample_bylevel* = 1,0

Percobaan	Kombinasi dan Performa
	Performa *Training Set*
	Accuracy = 0,9191; *Precision* = 0,5000; *Recall* = 1,0000; *F1-Score* = 0,6667; ROC AUC = 0,9935
	Performa *Test Set*
	Accuracy = 0,9118; *Precision* = 0,5000; *Recall* = 1,0000; *F1-Score* = 0,6667; ROC AUC = 0,9462
	Selisih Performa
	Accuracy = 0,0073; *Precision* = 0,0000; *Recall* = 0,0000; *F1-Score* = 0,0000; ROC AUC = 0,0473
Percobaan 5	***Kombinasi Hyperparameter***
	subsample = 1,0; *scale_pos_weight* = 11,36; *reg_lambda* = 1; *reg_alpha* = 0,1; *n_estimators* = 200; *min_child_weight* = 1; *max_leaves* = 9; *max_depth* = 5; *learning_rate* = 0,3; *gamma* = 0,1; *colsample_bylevel* = 1,0
	Performa *Training Set*
	Accuracy = 1,0000; *Precision* = 1,0000; *Recall* = 1,0000; *F1-Score* = 1,0000; ROC AUC = 1,0000
	Performa Test Set
	Accuracy = 0,9412; *Precision* = 0,6667; *Recall* = 0,6667; *F1-Score* = 0,6667; ROC AUC = 0,9570
	Selisih Performa
	Accuracy = 0,0588; *Precision* = 0,3333; *Recall* = 0,3333; *F1-Score* = 0,3333; ROC AUC = 0,0430

Berdasarkan Tabel 5.2, semua percobaan XGBoost tanpa SMOTE menunjukkan tanda *overfitting* dengan tingkat yang bervariasi. Berdasarkan besarnya selisih performa, *overfitting* paling jelas terlihat pada Percobaan 3, disusul oleh Percobaan 2, 5, 1 dan 4. Meskipun Percobaan 4 memiliki selisih sangat kecil, namun model terlihat memiliki presisi sangat kecil, yang artinya model terlalu banyak memprediksi positif palsu. Percobaan 4 juga memiliki *F1-Score* positif rendah, yang berarti model kesulitan menyeimbangkan presisi dan *recall* positif. Dengan kejelasan *overfitting* dan performa yang cenderung belum baik, menunjukan bahwa model XGBoost tanpa SMOTE ini kurang baik untuk digunakan, atau perlu penyesuaian proporsi data lebih lanjut.

Tabel 5.3 Percobaan Kombinasi *Hyperparameter* XGBoost dengan SMOTE

Percobaan	Kombinasi dan Performa
Percobaan 1	**Kombinasi *Hyperparameter***
	subsample = 0,75; *scale_pos_weight* = 1,0; *reg_lambda* = 0,9; *reg_alpha* = 0,4; *n_estimators* = 200; *min_child_weight* = 4; *max_leaves* = 6; *max_depth* = 7; *learning_rate* = 0,2; *gamma* = 1,0; *colsample_bylevel* = 1,0
	Performa *Training Set*
	Accuracy = 0,9760; *Precision* = 0,9685; *Recall* = 0,9840; *F1-Score* = 0,9762; ROC AUC = 0,9986
	Performa *Test Set*
	Accuracy = 0,9118; *Precision* = 0,5000; *Recall* = 0,6667; *F1-Score* = 0,5714; ROC AUC = 0,9570
	Selisih Performa
	Accuracy = 0,0642; *Precision* = 0,4685; *Recall* = 0,3173; *F1-Score* = 0,4048; ROC AUC = 0,0416
Percobaan 2	***Kombinasi Hyperparameter***
	subsample = 1,0; *scale_pos_weight* = 1,0; *reg_lambda* = 0,8; *reg_alpha* = 0,2; *n_estimators* = 300; *min_child_weight* = 1; *max_leaves* = 3; *max_depth* = 6; *learning_rate* = 0,3; *gamma* = 1,0; *colsample_bylevel* = 1,0
	Performa *Training Set*
	Accuracy = 0,9920; *Precision* = 0,9920; *Recall* = 0,9920; *F1-Score* = 0,9920; ROC AUC = 0,9995
	Performa *Test Set*
	Accuracy = 0,9118; *Precision* = 0,5000; *Recall* = 0,6667; *F1-Score* = 0,5714; ROC AUC = 0,9462
	Selisih Performa
	Accuracy = 0,0802; *Precision* = 0,4920; *Recall* = 0,3253; *F1-Score* = 0,4206; ROC AUC = 0,0533

Percobaan	Kombinasi dan Performa
Percobaan 3	***Kombinasi Hyperparameter*** *subsample* = 0,75; *scale_pos_weight* = 1,0; *reg_lambda* = 0,5; *reg_alpha* = 0,7; *n_estimators* = 200; *min_child_weight* = 1; *max_leaves* = 7; *max_depth* = 6; *learning_rate* = 0,05; *gamma* = 0,2; *colsample_bylevel* = 0,25 **Performa *Training Set*** *Accuracy* = 0,9920; *Precision* = 0,9920; *Recall* = 0,9920; *F1-Score* = 0,9920; ROC AUC = 0,9999 **Performa *Test Set*** *Accuracy* = 0,9412; *Precision* = 0,6667; *Recall* = 0,6667; *F1-Score* = 0,6667; ROC AUC = 0,9892 **Selisih Performa** *Accuracy* = 0,0508; *Precision* = 0,3253; *Recall* = 0,3253; *F1-Score* = 0,3253; ROC AUC = 0,0107
Percobaan 4	***Kombinasi Hyperparameter*** *subsample* = 1,0; *scale_pos_weight* = 1,0; *reg_lambda* = 0,30000000000000004; *reg_alpha* = 0,6; *n_estimators* = 100; *min_child_weight* = 2; *max_leaves* = 7; *max_depth* = 4; *learning_rate* = 0,1; *gamma* = 0,3; *colsample_bylevel* = 0,25 **Performa *Training Set*** *Accuracy* = 0,9920; *Precision* = 0,9920; *Recall* = 0,9920; *F1-Score* = 0,9920; ROC AUC = 0,9999 **Performa *Test Set*** *Accuracy* = 0,9412; *Precision* = 0,6667; *Recall* = 0,6667; *F1-Score* = 0,6667; ROC AUC = 0,9677 **Selisih Performa** *Accuracy* = 0,0508; *Precision* = 0,3253; *Recall* = 0,3253; *F1-Score* = 0,3253; ROC AUC = 0,0322
Percobaan 5 (yang digunakan)	***Kombinasi Hyperparameter*** *subsample* = 1,0; *scale_pos_weight* = 1,0; *reg_lambda* = 1; *reg_alpha* = 0,1; *n_estimators* = 200; *min_child_weight* = 1; *max_leaves* = 9; *max_depth* = 5; *learning_rate* = 0,3; *gamma* = 0,1; *colsample_bylevel* = 1,0 **Performa *Training Set*** *Accuracy* = 1,0000; *Precision* = 1,0000; *Recall* = 1,0000; *F1-Score* = 1,0000; ROC AUC = 1,0000 **Performa *Test Set*** *Accuracy* = 0,9706; *Precision* = 0,7500; *Recall* = 1,0000; *F1-Score* = 0,8571; ROC AUC = 0,9785 **Selisih Performa** *Accuracy* = 0,0294; *Precision* = 0,2500; *Recall* = 0,0000; *F1-Score* = 0,1429; ROC AUC = 0,0215

Model XGBoost dengan SMOTE dalam penelitian ini juga diuji coba, dan ternyata menunjukan indikasi *overfitting* dengan tingkat yang berbeda-beda di setiap percobaan yang dapat dilihat pada Tabel 5.3. Jika dilihat berdasarkan besarnya selisih performa, *overfitting* paling jelas terlihat pada Percobaan 2, disusul oleh Percobaan 1, Percobaan 4, Percobaan 3 dan yang terakhir adalah Percobaan 5. Percobaan 5 memiliki selisih performa paling rendah di antara semua percobaan, yang artinya mengalami *overfitting* yang paling rendah. Dimana performa *test set* sedikit lebih kecil daripada performa *training set*. Performa *train set* yang sempurna dengan sedikit penurunan performa pada *test set* dalam Percobaan 5, menunjukan kemungkinan model memiliki tantangan dalam hal generalisasi, walaupun peneliti sudah mencoba menerapkan berbagai tindakan pencegahan *overfitting* untuk meningkatkan kemampuan generalisasi model seperti yang sudah dijelaskan sebelumnya. Meskipun model yang dihasilkan memiliki sedikit indikasi *overfitting* dan performa presisi pada data *testing* bernilai 75%, nilai *Accuracy* (97,06%) dan ROC AUC (97,85%) pada data *testing* masih cukup tinggi. Selain itu, *Recall* pada data *testing* adalah 100%, yang berarti model mampu mengidentifikasi semua *instance*

positif (karyawan meninggalkan perusahaan) dengan benar. Namun, peluang untuk peningkatan performa model akan selalu dapat dilakukan.

Pengecekan *overfitting* dengan cara kedua yaitu mengevaluasi kinerja model pada data ekstrim, dilakukan dengan terlebih dahulu mengidentifikasi data ekstrim yang akan digunakan. Proses identifikasi data ekstrim dicoba dengan dua metode yaitu *Z-Score* dan *Inter Quartile Range* atau IQR. Didapatkan hasil identifikasi data ekstrim yang sama berdasarkan kedua metode tersebut, dengan total data ekstrim sebanyak 15 baris. Model XGBoost dengan SMOTE yang telah terbentuk kemudian digunakan untuk memprediksi data-data ekstrim tersebut, dan didapatkan hasil evaluasi kinerja yang sempurna (mencapai 100% baik pada *accuracy*, *precision*, *recall*, *F1-Score* maupun ROC AUC). Peningkatan performa model pada data ekstrim ini terjadi karena model telah "*memorize*" data pelatihan dengan sangat baik, termasuk data yang ekstrim. Oleh karena itu, ketika diterapkan pada data ekstrim, model memberikan hasil yang sangat baik bahkan sempurna, namun ini mengindikasikan *overfitting* pada model.

5.3 Peluang Peningkatan Performa Model

Karena ditemukannya sedikit indikasi *overfitting* yang terjadi, diketahui bahwa model XGBoost dengan SMOTE yang dihasilkan dalam penelitian ini mengalami tantangan dalam proses generalisasi model. Generalisasi model merupakan kemampuan model untuk membuat prediksi yang akurat pada data baru yang belum pernah dilihat sebelumnya, berdasarkan apa yang telah dipelajari dari data latih. Jika model dapat menggeneralisasi dengan baik, maka model tersebut akan memiliki performa yang baik tidak hanya pada data latih, tetapi juga pada data baru. Metrik evaluasi dapat digunakan untuk mengukur performa model tersebut, dan peningkatan pada metrik evaluasi menandakan bahwa model menjadi lebih baik dalam memprediksi data baru. Sehingga, jika metrik evaluasi menunjukkan peningkatan, model tersebut dianggap mampu menggeneralisasi dengan baik, karena dapat memprediksi data baru dengan lebih akurat.

Tabel 5.4 Perbandingan Performa Model Penelitian Terdahulu

Penelitian	Metode	Jumlah Data	Akurasi	Presisi	*Recall*	*F1-Score*	AUC
Mardiansyah et al. (2021)	SMOTE & XGBoost	1000	-	-	-	-	98,88%
		214	-	-	-	-	99,93%
		699	-	-	-	-	99,80%
		337	-	-	-	-	99,40%

Penelitian	Metode	Jumlah Data	Akurasi	Presisi	*Recall*	*F1-Score*	AUC
Yang & Guan (2022)	SMOTE-ENN & XGBoost	3527	93,44%	92,66%	97,16%	94,86%	92,24%
Lin et al. (2021)	SMOTE & XGBoost	100000	92,80%	93,34%	92,80%	92,81%	-
	SMOTE-ENN & XGBoost	100000	94,17%	94,22%	93,84%	93,89%	-
Penelitian saat ini	SMOTE & XGBoost	170	97,06%	75%	100%	85,71%	97,85%

Ketiga penelitian dalam Tabel 5.4 di atas memiliki jumlah data yang lebih banyak dibandingkan dengan jumlah data dalam penelitian ini. Hal ini bisa menjadi salah satu faktor yang mempengaruhi peningkatan performa model, yang pada akhirnya dapat meningkatkan generalisasi model. Penelitian Mardiansyah et al. (2021), dengan metode yang sama namun jumlah data yang lebih banyak, memiliki nilai AUC yang lebih tinggi dibandingkan dengan penelitian ini. Kemudian penelitian Lin et al. (2021) dengan data yang lebih banyak, menunjukan bahwa integrasi *Edited Nearest Neighbours* (ENN) pada SMOTE & XGBoost memberikan peningkatan dan keseimbangan nilai presisi dan *recall* model, sehingga nilai *F1-Score* yang didapatkan juga tinggi. Hasil tersebut didukung oleh penelitian Yang & Guan (2022), yang juga memperoleh performa model yang sangat baik ketika mengintegrasikan ENN pada SMOTE dan XGBoost serta menggunakan jumlah data yang lebih banyak. Sehingga dapat disimpulkan bahwa dengan jumlah data yang lebih banyak, model dapat belajar dari variasi yang lebih besar dalam data, sehingga dapat membantu meningkatkan kemampuan generalisasi dan mengatasi *overfitting* model. Teknik penyeimbangan data dengan ENN yang diintegrasikan dengan SMOTE juga dapat memberikan peluang peningkatan performa model. Hal ini merupakan sebuah saran langkah pengembangan dari penelitian saat ini, untuk mencapai model yang lebih *robust* (mampuan menangani variasi dalam data input dengan tetap memberikan hasil yang akurat) di kemudian hari.

5.4 Analisis Perbaikan Masalah

Permasalahan yang dialami oleh PT. PNR adalah tingginya tingkat *turnover* karyawan Divisi Engineering yang menjadi salah satu tenaga ahli dan *core function* perusahaan. Hal ini memberikan dampak negatif kepada perusahaan, berupa peningkatan potensi kekurangan SDM berbakat, munculnya gangguan dalam produktivitas tempat kerja, serta memakan waktu dan biaya yang lebih. Sejauh ini PT. PNR belum memiliki pemanfaatan lanjutan dari data historis yang dimiliki, dengan menggunakan teknik yang ada dalam

data mining dan algoritma *machine learning* untuk pengelolaan *turnover*. Sementara luasnya tantangan revolusi data (seperti pengumpulan, analisis dan penggunaan data) yang dialami oleh sektor industri pertambangan, dapat menurunkan margin keuntungan secara drastis (Qi, 2020).

Melalui penelitian ini, beberapa rekomendasi praktis diberikan untuk memenuhi kebutuhan perusahaan, dalam memahami faktor-faktor pendorong utama *turnover*, dan melakukan antisipasi dengan memprediksi karyawan yang akan keluar. Penelitian ini menghasilkan model klasifikasi yang dapat melakukan prediksi, apakah seorang karyawan akan terprediksi bertahan atau meninggalkan perusahaan. Model klasifikasi yang memanfaatkan integrasi SMOTE dengan algoritma pengklasifikasi XGBoost, terbukti memiliki performa yang baik dalam melakukan prediksi pada penelitian ini. Serta dapat diketahui atribut apa saja yang paling mempengaruhi kejadian *turnover* berdasarkan model tersebut.

Berikut merupakan rekomendasi praktis yang diberikan berdasarkan hasil penelitian, dan diskusi dengan pihak perusahaan agar sesuai dengan kondisi terkini di perusahaan.

1. Perusahaan dapat menggunakan model hasil penelitian ini dalam manajemen retensi karyawan. Perusahaan dapat bekerjasama dengan tim *Information Technology* (IT) yang dimiliki, untuk mengembangkan sistem atau aplikasi yang secara khusus dapat melakukan prediksi *turnover* dengan menggunakan model tersebut. Penerapan aplikasi ini dapat dilakukan secara berkala oleh Management dan Divisi Human Capital & Talent Development, untuk dapat memperkirakan karyawan mana sajakah yang terprediksi akan meninggalkan perusahaan.
2. Bagi Management, hasil klasifikasi dapat dimanfaatkan untuk membuat *action plan* sebagai *retention program* terhadap karyawan yang diperkirakan akan keluar, berdasarkan *features importance order* yang didapatkan.
3. Bagi Divisi Human Capital & Talent Development, hasil klasifikasi akan membantu monitor dan identifikasi karyawan yang terprediksi *turnover*. Sehingga personel yang memberikan program *coaching & counseling*, dapat memberikan saran, bimbingan dan pengembangan yang lebih sesuai untuk karyawan terkait.

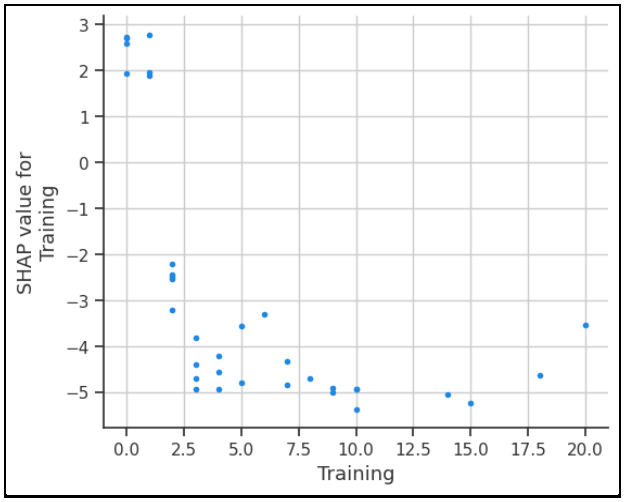
Untuk memenuhi rekomendasi praktis yang dijelaskan pada nomor dua dan tiga, maka penting untuk memahami lebih baik mengenai faktor-faktor yang paling berpengaruh dalam model. Peneliti mencoba untuk membahas lima fitur teratas, dengan pertimbangan

bobot yang lebih bervariasi, dibandingkan fitur-fitur setelahnya. Pemberian rekomendasi berdasarkan fitur teratas ini memanfaatkan plot *Shapley Additive Explanations* (SHAP) dependensi, yang dapat membantu dalam menentukan kapan rekomendasi perlu diberikan. SHAP dapat memberikan informasi mengenai dampak nilai suatu fitur pada prediksi model (Tao et al., 2021).

1. *Training*.

   Berdasarkan Gambar 5.1 di bawah, didapatkan interpretasi sebagai berikut.

   a. Ketika jumlah *training* yang diterima karyawan kurang dari 2, cenderung terjadi peningkatan probabilitas *turnover* (SHAP *value* positif).
   b. Ketika jumlah *training* yang diterima karyawan lebih dari 2, cenderung terjadi penurunan probabilitas *turnover* (SHAP *value* negatif).
   c. SHAP *value* terendah terjadi saat pelatihan yang diberikan berjumlah 10 sampai 15, yang artinya ketika jumlah pelatihan telah mencapai 10 hingga 15 pelatihan, *turnover* terhambat dengan jelas.
   d. Hambatan tersebut kembali berkurang (*plot* data mulai naik menuju arah positif) setelah jumlah pelatihan mencapai 17 pelatihan.



Gambar 5.1 SHAP Dependensi Atribut *Training*

Interpretasi tersebut menunjukkan bahwa dalam retensi karyawan, penting untuk memberikan jumlah pelatihan yang sesuai. Karyawan yang merasa mendapatkan jumlah pengembangan dan pelatihan yang memadai, akan lebih cenderung merasa puas sehingga bertahan di perusahaan. Namun, jika dilihat berdasarkan *loss cost*

yang dialami perusahaan, biaya pelatihan menjadi bagian dari *loss cost* tersebut ketika karyawan memutuskan untuk keluar. Sehingga, jika hanya terfokus pada jumlah pelatihan saja, akan tetap ada risiko ketidakberhasilan investasi pelatihan jika karyawan tersebut keluar. Menurut Sepang et al. (2023), pelatihan yang baik dapat mengurangi keinginan karyawan untuk meninggalkan perusahaan. Hal ini berlaku sebaliknya, dimana pelatihan yang buruk akan meningkatkan keinginan karyawan untuk meninggalkan perusahaan.

Selama ini, perusahaan telah menerapkan program retensi berdasarkan *training*, berupa penalti atas keikutsertaan karyawan dalam *training* tertentu. Penalti tersebut adalah tidak diperbolehkannya karyawan *resign* selama 2 tahun mendatang, setelah mengikuti *training* tertentu yaitu salah satunya *technical training* yang memiliki biaya tinggi dan membutuhkan *agreement* dalam pelaksanaannya. Selain itu, perusahaan telah menyediakan mekanisme permohonan *training* oleh karyawan, yang dilakukan melalui atasannya dalam proses *coaching & counseling*. Berdasarkan pengetahuan yang didapatkan dari plot SHAP atribut *training*, berikut merupakan saran rekomendasi program retensi yang dapat diberikan.

a. Untuk karyawan dengan jumlah *training* kurang dari 2, perusahaan perlu menambahkan jumlah *training* yang mereka ikuti, dengan mempertimbangkan manfaat pelatihan berdasarkan jenisnya. Jenis pelatihan teknis dapat membantu meningkatkan keterampilan kerja. Kemudian jenis pelatihan *soft skills* berguna untuk meningkatkan kemampuan komunikasi, kepemimpinan, kerja sama tim dan kemampuan sosial interpersonal lainnya. Serta jenis pelatihan pengembangan diri berguna untuk meningkatkan motivasi dan kepuasan kerja.
b. Untuk karyawan dengan jumlah *training* 2 hingga 15, perusahaan perlu meningkatkan efektivitas program pelatihan dengan memastikan pelatihan relevan terhadap kebutuhan karyawan dan organisasi. Hal ini dapat dicapai dengan melakukan analisis kebutuhan pelatihan untuk memastikan pemilihan jenis *training* yang tepat. Pemberian *feedback* dan penilaian yang konstruktif juga perlu dilakukan, sehingga *engagement* karyawan dalam pelatihan dapat terus meningkat.
c. Untuk karyawan dengan jumlah *training* lebih dari 15 dimana karyawan ini mengalami peningkatan kembali dalam kemungkinan *turnover*, bisa saja

disebabkan karena merasa jenuh dan kurangnya peluang untuk menggunakan keterampilan dan pengetahuan dari *training*. Sehingga perusahaan perlu mempertimbangkan pengembangan karir karyawan, dimana pelatihan perlu disertai peluang kemajuan karyawan dalam berkarir dan mengimplementasikan keterampilan serta pengetahuan yang didapatkan dari *training*. Hal ini akan meningkatkan kecenderungan karyawan untuk bertahan, karena karyawan dapat melihat jalan karir kedepan yang jelas.

d. Terkait rentang waktu pemberian *training*, tidak ada tetapan baku yang berlaku. Namun tetap dapat dipertimbangkan berdasarkan jenis dan tujuan *training*, kebutuhan karyawan dan organisasi, serta anggaran dan sumber daya yang dimiliki oleh perusahaan.

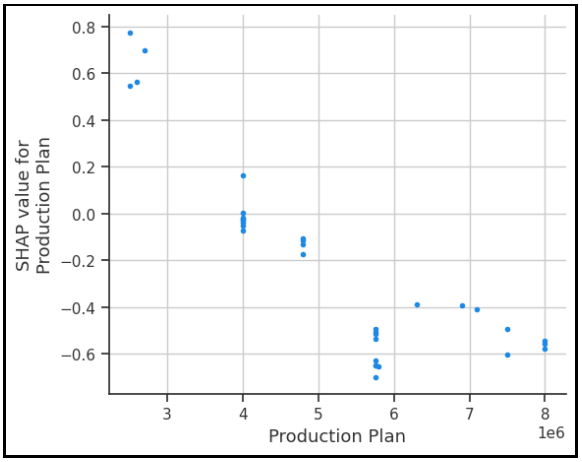
2. *Production Plan*

Berdasarkan Gambar 5.2 di bawah, didapatkan interpretasi sebagai berikut.

a. Ketika *production plan* yang ditargetkan kurang dari 4 juta ton, cenderung terjadi peningkatan probabilitas *turnover* (SHAP *value* positif).

b. Ketika *production plan* yang ditargetkan lebih dari 4 juta ton, cenderung terjadi penurunan probabilitas *turnover* (SHAP *value* negatif).

c. SHAP *value* terendah terjadi saat *production plan* yang ditargetkan berjumlah antara 5-6 ton dan 7-8 ton, yang artinya ketika *production plan* yang ditargetkan mencapai jumlah tersebut, *turnover* terhambat dengan jelas.



Gambar 5.2 SHAP Dependensi Atribut *Production Plan*

Ketika rencana produksi sebagai target yang harus dicapai dirasa terlalu tinggi oleh karyawan, maka akan mempengaruhi tekanan kerja yang dirasakan. Namun, tidak menutup kemungkinan bahwa target produksi juga bisa dirasa terlalu rendah oleh karyawan, dan memunculkan rasa kurang terlibat atau kurang tertantang dalam diri karyawan. Kedua kondisi tersebut menunjukan bahwa besar kecilnya rencana produksi bisa mempengaruhi tekanan kerja dan harapan yang diberikan kepada karyawan, yang pada akhirnya akan berdampak pada kepuasan kerja. Menurut Farizi & Tanuwijaya (2022), kepuasan kerja memiliki pengaruh negatif dan signifikan terhadap terhadap niat *turnover*. Artinya, jika perusahaan mampu meningkatkan kepuasan kerja karyawan, maka keinginan karyawan untuk keluar dari perusahaan akan menurun. Sehingga tindakan retensi perlu dilakukan, baik untuk kondisi *production plan* tinggi maupun rendah.

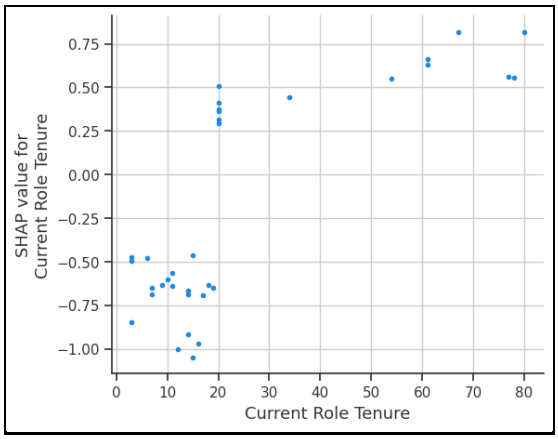
a. Pada saat *production plan* rendah, perusahaan perlu meningkatkan keterlibatan karyawan, melalui proyek khusus atau tugas tambahan yang menantang. Ketika karyawan menyelesaikan proyek atau tugas tersebut, akan muncul rasa keberhasilan atas pencapaian yang diraih, yang pada akhirnya dapat meningkatkan kepuasan kerja dan retensi karyawan. Dengan cara ini, perusahaan dapat memastikan karyawan tetap produktif selama periode *production plan* rendah, sekaligus menjadi kesempatan pengembangan keterampilan dan pengetahuan karyawan.
b. Pada saat *production plan* tinggi, perusahaan perlu memberikan dukungan yang cukup untuk karyawannya, sehingga karyawan merasa mampu mengatasi tantangan dari *production plan* yang tinggi ini. Dukungan yang diberikan dapat berupa dukungan manajerial, sumber daya dan fasilitas yang memadai, serta pengakuan atas kerja keras yang telah karyawan berikan.

3. *Current Role Tenure*

Berdasarkan Gambar 5.3 di bawah, didapatkan interpretasi sebagai berikut.

a. Ketika *current role tenure* atau durasi posisi saat ini kurang dari 20 bulan, cenderung terjadi penurunan probabilitas *turnover* (SHAP *value* negatif).
b. Ketika durasi posisi saat ini lebih dari sama dengan 20 bulan, cenderung terjadi peningkatan probabilitas *turnover* (SHAP value positif).

c. SHAP *value* tertinggi terjadi saat durasi posisi saat ini telah mencapai 60 sampai 80 bulan, yang artinya ketika durasi posisi saat ini telah mencapai angka tersebut, maka besar sekali kemungkinan *turnover* terjadi.



Gambar 5.3 SHAP Dependensi Atribut *Current Role Tenure*

Lama waktu karyawan berada pada posisi saat ini atau bisa disebut dengan durasi posisi saat ini, ternyata berkontribusi tinggi terhadap kejadian *turnover*. Hasil ini sejalan dengan penelitian oleh Gao et al. (2019), yang juga mendapatkan hasil dari model klasifikasi yang dibuat, bahwa durasi posisi saat ini merupakan salah satu atribut yang paling berkontribusi terhadap *turnover*. Hal ini menunjukan bahwa program retensi yang akan dibuat, perlu mempertimbangkan durasi posisi karyawan. Karyawan yang telah cukup lama berada pada posisi saat ini, memiliki kemungkinan untuk memulai mencari peluang baru, yang bisa saja meningkatkan keinginan untuk keluar. Sedangkan karyawan yang baru memulai posisi saat ini, memiliki kecenderungan untuk tetap di perusahaan. Meskipun begitu, tindakan retensi untuk karyawan yang baru memulai posisi saat ini tetap perlu dipersiapkan. Sehingga, rekomendasi retensi yang bisa diberikan berdasarkan *current role tenure* adalah sebagai berikut.

a. Bagi karyawan yang telah cukup lama berada pada posisi saat ini, perusahaan dapat menawarkan peluang pengembangan karir, seperti promosi atau rotasi posisi untuk menjaga keterlibatan dan motivasi dalam diri karyawan.

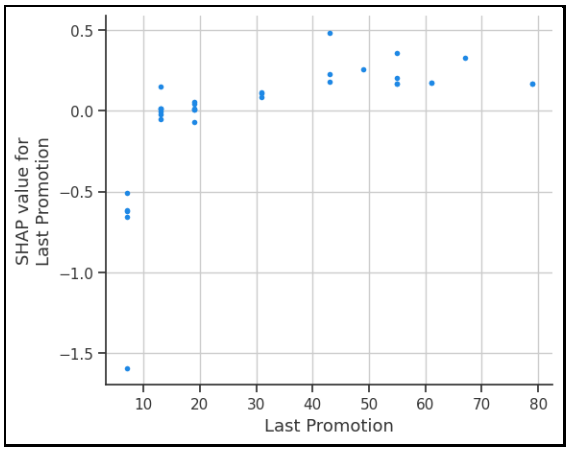
b. Bagi karyawan yang baru memulai posisi saat ini, perusahaan perlu membantu karyawan untuk bertumbuh dalam posisi tersebut. Sehingga rasa percaya diri dan kompeten karyawan dapat terbangun, dan menurunkan kemungkinan *turnover*.

4. *Last Promotion*

Berdasarkan Gambar 5.4 di bawah, didapatkan interpretasi sebagai berikut.

a. Ketika *last promotion* sebagai jarak waktu promosi terakhir kurang dari 10 bulan, cenderung terjadi penurunan probabilitas *turnover* (SHAP *value* negatif).

b. Ketika jarak waktu promosi terakhir lebih dari 10 bulan, cenderung terjadi peningkatan probabilitas *turnover* (SHAP *value* positif).



Gambar 5.4 SHAP Dependensi Atribut *Last Promotion*

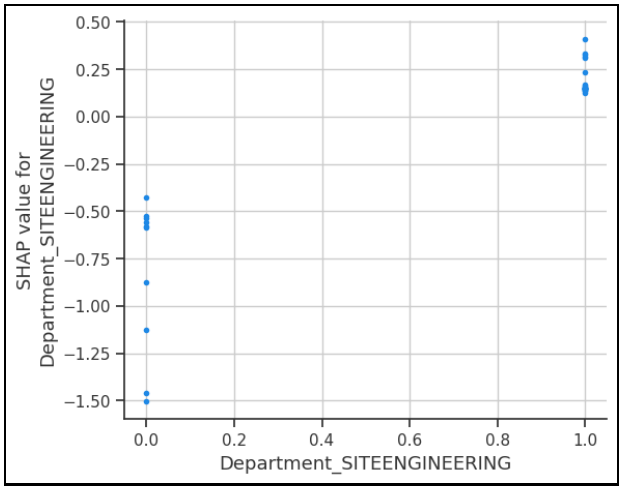
Hasil ini sejalan dengan penelitian oleh Chanodkar et al. (2019), dimana kesenjangan promosi yang terjadi, menjadi salah satu atribut yang sangat berkontribusi dalam kejadian *turnover*. Karyawan yang belum mendapatkan promosi dalam jangka waktu cukup lama, dapat merasa kurang dihargai serta motivasi untuk melanjutkan pekerjaan menjadi menurun. Yang pada akhirnya hal tersebut akan meningkatkan kemungkinan karyawan untuk keluar. Namun tidak menutup kemungkinan bahwa karyawan yang baru saja mendapatkan promosi, memiliki ekspektasi terhadap pertumbuhan mereka di perusahaan yang bisa jadi tidak sesuai dengan realitanya. Sehingga tindakan retensi perlu dilakukan, baik untuk karyawan yang telah lama tidak mendapatkan promosi, maupun karyawan yang baru saja mendapatkan promosi.

a. Untuk karyawan yang telah lama tidak mendapatkan promosi, perusahaan perlu menawarkan promosi atau penghargaan lainnya, sebagai bentuk apresiasi atas kerja keras yang dilakukan dan menjaga motivasi karyawan.
b. Untuk karyawan yang baru saja mendapatkan promosi, perusahaan perlu membangun komunikasi berkelanjutan mengenai ekspektasi karyawan, serta peluang yang dapat dicapai oleh karyawan untuk bertumbuh dalam perusahaan.

5. *Department_SITEENGINEERING*

Berdasarkan Gambar 5.3 di bawah, didapatkan interpretasi sebagai berikut.

a. Ketika karyawan adalah bagian dari dari *Department Site Engineering* (1), cenderung terjadi peningkatan probabilitas *turnover* (SHAP *value* positif).
b. Ketika karyawan adalah bukan bagian dari dari *Department Site Engineering* (0), cenderung terjadi penurunan probabilitas *turnover* (SHAP *value* negatif).



Gambar 5.5 SHAP Dependensi Atribut *Department_SITEENGINEERING*

Karyawan yang berada pada departemen *Site Engineering* memiliki sifat pekerjaan, beban kerja dan tantangan tersendiri yang perlu dihadapi. Pekerjaan pada departemen ini berfokus pada perencanaan penambangan, penghitungan jenis batuan, desain tambang, dan pertimbangan biaya dan keselamatan. Kondisi kerja tersebut menuntut tingkat keahlian dan pengetahuan yang tinggi, sehingga bisa saja melibatkan tingkat stres yang lebih tinggi pula dibandingkan dengan departemen lainnya. Menurut Lestari (2021), tingkat stres yang lebih tinggi akan meningkatkan

keinginan karyawan untuk keluar. Sehingga, rekomendasi tindakan retensi yang dapat diberikan berdasarkan Departemen Site Engineering adalah sebagai berikut.

a. Penyediaan pelatihan dan pengembangan profesional, untuk meningkatkan kompeten dan rasa percaya diri dalam pekerjaan, sehingga membantu mengurangi stres dan meningkatkan kepuasan kerja.
b. *Work-life balance*, dimana perusahaan perlu memastikan bahwa karyawan memiliki *work-life balance* yang baik, mengingat tingkat stres yang mungkin tinggi dalam pekerjaan di departemen ini.
c. Dukungan baik dari manajemen dan atasan, yang bisa sangat berpengaruh terhadap kepuasan kerja karyawan. Perlu dipastikan bahwa pihak manajemen dan atasan memahami tantangan yang dihadapi oleh karyawan di departemen ini, dan berusaha untuk memberikan solusi.
d. Penghargaan dan insentif, untuk meningkatkan kekuatan motivasi karyawan. Hal ini dapat berupa bonus, penghargaan atas prestasi, atau bentuk penghargaan lainnya yang menunjukan bahwa perusahaan menghargai kerja keras karyawan.
e. Lingkungan kerja yang aman, untuk memastikan bahwa karyawan merasa aman di tempat kerja. Tindakan ini dapat dilakukan dengan komunikasi keselamatan kerja yang efektif, keterlibatan karyawan dalam proses peningkatan keselamatan kerja, melakukan audit keselamatan secara berkala, hingga dukungan kesejahteraan mental dan fisik karyawan (seperti fasilitas olahraga, dan program kesehatan karyawan lainnya).

Melalui rekomendasi yang diusulkan, PT. PNR dapat melakukan intervensi lebih awal untuk mencegah karyawannya (khususnya Divisi Engineering) berhenti secara sukarela. Perbaikan masalah yang ditawarkan oleh penelitian ini sejalan dengan penelitian terdahulu, yang menyatakan bahwa memanfaatkan fungsi *data mining* yaitu klasifikasi dan prediksi merupakan salah satu bentuk pengendalian *turnover* karyawan, dan dipastikan akan sangat meningkatkan kinerja dari manajemen SDM itu sendiri (Gao et al., 2019; Manurung et al., 2021; Punnoose & Ajit, 2016; Zhao et al., 2019).

BAB VI

PENUTUP

6.1 Kesimpulan

Berdasarkan analisis dan pembahasan yang telah dilakukan, berikut merupakan kesimpulan yang didapatkan.

1. Performa model XGBoost tanpa SMOTE yaitu akurasi 0,94; *misclassification rate* 0,06; presisi, *recall* dan *F1-Score* negatif 0,97; presisi, *recall* dan *F1-Score* positif 0,67 dan AUC 0,96. Sedangkan performa model XGBoost dengan SMOTE yaitu akurasi 0,97; *misclassification rate* 0,03; presisi negatif 1,00; presisi positif 0,75; *recall* negatif 0,97; *recall* positif 1,00; *F1-Score* negatif 0,98; *F1-Score* positif 0,86 dan AUC 0,98. Sehingga, dalam memprediksi kejadian *turnover* karyawan Divisi Engineering PT. PNR pada penelitian ini, diketahui bahwa metode XGBoost dengan SMOTE memiliki performa lebih baik dibandingkan dengan metode XGBoost tanpa SMOTE. Khususnya dalam menyeimbangkan prediksi (*F1-Score*) pada kelas data minoritas (positif) dan kelas data mayoritas (negatif).
2. Berdasarkan model klasifikasi terbaik, XGBoost dengan SMOTE, didapatkan lima atribut yang paling mempengaruhi kejadian *turnover* karyawan Divisi Engineering PT. PNR secara berurutan yaitu *Training*, *Production Plan*, *Current Role Tenure*, *Last Promotion* dan *Department_SITEENGINEERING*.
3. Penelitian ini memberikan usulan perbaikan kepada PT. PNR, dimana perusahaan dapat menggunakan model klasifikasi terbaik yang dihasilkan, untuk mengembangkan sistem atau aplikasi prediksi *turnover*. Serta merancang program retensi berdasarkan lima atribut terpenting yang didapatkan. Sehingga PT. PNR dapat melakukan intervensi lebih awal untuk mencegah karyawannya (khususnya Divisi Engineering) berhenti secara sukarela.

6.2 Saran

Saran yang dapat peneliti berikan untuk perbaikan dan pengembangan lanjutan di masa mendatang adalah sebagai berikut.

1. PT. PNR sebaiknya menjadikan penelitian ini sebagai acuan untuk melakukan intervensi lebih awal terhadap kejadian *turnover* karyawan Divisi Engineering.
2. Mengembangkan kemampuan generalisasi model dengan memperbanyak data input dan percobaan teknik penyeimbangan data ENN, sehingga dapat dicapai model yang lebih *robust*.
3. Menggunakan algoritma pengklasifikasi lainnya sebagai bahan perbandingan.

DAFTAR PUSTAKA

Alaskar, L., Crane, M., & Alduailij, M. (2019). Employee Turnover Prediction Using Machine Learning. In *First International Conference on Computing, ICC 2019 Proceedings, Part II* (Vol. 1098). Springer International Publishing. https://doi.org/10.1007/978-3-030-36365-9

Anwar, G., & Abdullah, N. N. (2021). The impact of Human resource management practice on Organizational performance. *International Journal of Engineering, Business and Management (IJEBM)*, *5*(1), 35–47. https://doi.org/10.22161/ijebm.5.1.4

Caesaria, A. K., Astiningrum, M., & Syulistyo, A. R. (2020). Identifikasi Komponen Gui Pada Prototipe Aplikasi Mobile. *Jurnal Informatika Polinema*, *6*(2), 51–56. https://doi.org/10.33795/jip.v6i2.321

Chanodkar, A., Changle, R., & Deepesh. (2019). Prediction of Employee Turnover in Organizations using Machine Learning Algorithms. *Prestige International Journal of Management and Research*, *12R (1-2)*, 222–229.

Chhinzer, N. (2021). Contrasting voluntary versus involuntary layoffs: Antecedents and outcomes. *Canadian Journal of Administrative Sciences*, *38*(2), 177–192. https://doi.org/10.1002/cjas.1594

Chiat, L. C., & Panatik, S. A. (2019). *Perceptions of Employee Turnover Intention by Herzberg ' s Motivation-Hygiene Theory : A Systematic Literature Review*. *1*(2), 3–8.

Collins, C. J. (2020). Expanding the resource based view model of strategic human resource management strategic human resource management. *The International Journal of Human Resource Management*, *0*(0), 1–28. https://doi.org/10.1080/09585192.2019.1711442

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

Dwesini, N. F. (2019). Causes and prevention of high employee turnover within the hospitality industry: A literature review. *African Journal of Hospitality, Tourism and Leisure*, *8*(3), 1–15.

Effendi, M. M., & Rahmawati, D. (2018). PREDIKSI PENJUALAN PRODUK ROTI MENGGUNAKAN ALGORITMA C4.5 PADA PT. PRIMA TOP BOGA. *Jurnal SIGMA*, *9*(2). https://doi.org/10.1016/j.gecco.2019.e00539%0Ahttps://doi.org/10.1016/j.foreco.2018.06.029%0Ahttp://www.cpsg.org/sites/cbsg.org/files/documents/Sunda Pangolin National Conservation Strategy and Action Plan %28LoRes%29.pdf%0Ahttps://doi.org/10.1016/j.forec

Farizi, M., & Tanuwijaya, J. (2022). Career Satisfaction Mempunyai Pengaruh Yang Signifikan Terhadap Turnover Intention Pada Karyawan di Industri Pertambangan.

SEIKO : Journal of Management & Business, *5*(2), 137–146. https://doi.org/10.37531/sejaman.vxix.2353

Febriani, S., & Sulistiani, H. (2021). ANALISIS DATA HASIL DIAGNOSA UNTUK KLASIFIKASI GANGGUAN KEPRIBADIAN MENGGUNAKAN ALGORITMA C4.5. *Jurnal Teknologi Dan Sistem Informasi (JTSI)*, *2*(4), 89–95.

Foley, M. (2022). *Supervised Machine Learning*. rstudio/bookdown. https://bookdown.org/mpfoley1973/supervised-ml/

Gajendra. (2022). *Gradient Boosting & Extreme Gradient Boosting (XGBoost) Understanding Gradient Boosting & Extreme Gradient Boosting (XGBoost)*. Medium. https://medium.com/@gajendra.k.s/gradient-boosting-extreme-gradient-boosting-xgboost-de865b871203

Gao, X., Wen, J., & Zhang, C. (2019). An Improved Random Forest Algorithm for Predicting Employee Turnover. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2019/4140707

Hajar, S., Novany, A. A., Windarto, A. P., Wanto, A., & Irawan, E. (2020). Penerapan K-Means Clustering pada ekspor minyak kelapa sawit menurut negara tujuan. *Seminar Nasional Teknologi Komputer & Sains (SAINTEKS) 2020*, 314–318.

Husin, N. (2023). Komparasi Algoritma Random Forest, Naïve Bayes, dan Bert Untuk Multi-Class Classification Pada Artikel Cable News Network (CNN). *Jurnal Esensi Infokom : Jurnal Esensi Sistem Informasi Dan Sistem Komputer*, *7*(1), 75–84. https://doi.org/10.55886/infokom.v7i1.608

Iddrus, & Junaedi, H. (2022). Prediksi Kelulusan Mahasiswa Menggunakan Support Vector Machine Berbasi Particle Swarm Optimization. *SMARTICS Journal*, *8*(1), 1. https://doi.org/10.21067/smartics.v8i1.6879

Iskandar, Y. C., & Rahadi, D. R. (2021). *Strategi Organisasi Penanganan Turnover melalui Pemberdayaan Karyawan*. *19*(1), 102–116.

Juvitayapun, T. (2021). Employee Turnover Prediction: The impact of employee event features on interpretable machine learning methods. *2021 13th International Conference Knowledge and Smart Technology*, 181–185. https://doi.org/10.1109/KST51265.2021.9415794

Ke, H., Gong, S., He, J., Zhang, L., & Mo, J. (2022). A hybrid XGBoost-SMOTE model for optimization of operational air quality numerical model forecasts. *Frontiers in Environmental Science*. https://doi.org/10.3389/fenvs.2022.1007530

Khera, S. N., & Divya. (2019). Predictive Modelling of Employee Turnover in Indian IT Industry Using Machine Learning Techniques. *Vision*, *23(1)*, 12–21. https://doi.org/10.1177/0972262918821221

Kovvuri, S. R., & Dommeti, L. S. D. (2022). *Employee Turnover Prediction A Comparative Study of Supervised Machine Learning Models*. Blekinge Institute of Technology.

Lee, Y.-S., & Liu, W.-K. (2021). The Moderating Effects of Employee Benefits and Job Burnout among the Employee Loyalty, Corporate Culture and Employee Turnover. *Universal Journal of Management*, *9*(2), 62–69.

https://doi.org/10.13189/ujm.2021.090205

Lestari, S. M. (2021). Pengaruh Stres Kerja Terhadap Turnover Intention Dimediasi Oleh Kepuasan Kerja (Studi Pada Karyawan Tetap Bri Kc Tanjung Redeb-Berau). *Jurnal Ilmiah Mahasiswa FEB Universitas Brawijaya*, *9*(1), 1–18.

Lin, M., Zhu, X., Hua, T., Tang, X., Tu, G., & Chen, X. (2021). *Detection of Ionospheric Scintillation Based on XGBoost Model Improved by SMOTE-ENN Technique*. 1–22. https://doi.org/ 10.3390/rs13132577

Maghfiroh, A., Findawati, Y., & Indahyanti, U. (2023). *Klasifikasi Penipuan pada Rekening Bank menggunakan Pendekatan Ensemble Learning*. *4*(4), 1883–1891. https://doi.org/10.47065/bits.v4i4.3212

Manurung, D. D. E., Sandi, F., Akbardipura, F., Ashfahan, H., & Prasvita, D. S. (2021). Prediksi Pengunduran Diri Karyawan Perusahaan " Y " Menggunakan Random Forest. *Seminar Nasional Mahasiswa Ilmu Komputer Dan Aplikasinya (SENAMIKA)*, *September*, 202–213.

Mardiansyah, H., Sembiring, R. W., & Efendi, S. (2021). Handling Problems of Credit Data for Imbalanced Classes using SMOTEXGBoost. *Journal of Physics: Conference Series*, *1830*. https://doi.org/10.1088/1742-6596/1830/1/012011

Maulida, H., & Rusilowati, U. (2020). Effect of Competence and Career Development on Turnover and It's Impact on Productivity. *Jurnal Manajemen*, *24*(1), 59–73. https://doi.org/10.24912/jm.v24i1.618

Mortara, A. A., Permatasari, M., Desiani, A., Andriani, Y., & Arhami, M. (2023). Perbandingan Algoritma C4 . 5 dan Adaptive Boosting dalam Klasifikasi Penyakit Alzheimer. *Jurnal Teknologi Dan Informasi (JATI)*, *13*. https://doi.org/10.34010/jati.v13i2

Munti, N. Y. S. M., Nurcahyo, G. W., & Santony, J. (2018). Analisis Dan Penerapan Data Mining Untuk Menentukan Gaji Karyawan Tetap Dan Karyawan Kontrak Menggunakan Algoritma K-Means Clustering (Studi Kasus Di Pt Indomex Dwijaya Lestari). *Jiti*, *1*(1), 2–9.

Novianti, D. (2019). Implementasi Algoritma Naïve Bayes Pada Data Set Hepatitis Menggunakan Rapid Miner. *Paradigma - Jurnal Komputer Dan Informatika*, *21*(1), 49–54. https://doi.org/10.31294/p.v21i1.4979

Noviyanti, S. A. (2018). *Prediksi Turnover Karyawan Menggunakan Metode Klasifikasi Naïve Bayes (Survey: PT. XYZ Wilayah Tangerang)* [Universitas Mercu Buana]. https://repository.mercubuana.ac.id/id/eprint/60728

Palupi, E. S. (2021). Employee Turnover Classification Using Pso-Based Naïve Bayes and Naïve Bayes Algorithm in Pt. Mastersystem Infotama. *Jurnal Riset Informatika*, *3*, 233–240. https://doi.org/10.34288/jri.v3i3.232

Prabowo, D. A. (2019). *Pengaruh Kompetensi, Kompensasi, Komunikasi, Disiplin Kerja terhadap Kinerja Karyawan pada Industri Pertambangan (Studi Kasus : PT. Kalimantan Prima Persada)*. http://hdl.handle.net/123456789/15731

Prawitasari, A. (2016). Faktor-Faktor yang Mempengaruhi Turnover Intention Karyawan pada PT. Mandiri Tunas Finance Bengkulu. *Ekombis Review*, 177–186.

Punnoose, R., & Ajit, P. (2016). Prediction of Employee Turnover in Organizations using Machine Learning Algorithms. *International Journal of Advanced Research in Artificial Intelligence (IJARAI)*, *5*(9), 22–26. https://doi.org/10.14569/ijarai.2016.050904

Putra, I. G. A. P., & Surya, I. B. K. (2020). *Pengaruh Stres Kerja terhadap Turnover Intention dengan Dukungan Sosial sebagai Variabel Pemoderasi*. *9*(7), 2790–2809.

Qi, C. chong. (2020). Big data management in the mining industry. *International Journal of Minerals, Metallurgy and Materials*, *27*(2), 131–139. https://doi.org/10.1007/s12613-019-1937-z

Rachmi, A. N. (2020). *Implementasi Metode Random Forest Dan Xgboost Pada Klasifikasi Customer Churn*.

Redafanza, F., Ahluwalia, L., & Putri, A. D. (2023). Pengaruh Job Insecurity dan Role Overload Terhadap Turnover Intention Karyawan Generasi Z di Bandar Lampung. *Strategy of Management and Accounting through Research and Technology (SMART)*, *2*(2), 11–22.

Sampurna, I. O. (2021). *Teknik Resampling untuk Data Tidak Seimbang*. Medium. https://ivanongko.medium.com/teknik-resampling-untuk-data-tidak-seimbang-5d566661101c

Samson, A. M., & Suliystiorini, D. (2020). Person Organization Fit Dan Psyhological Capital Sebagai Prediktor Turnover Intention Pada Karyawan Site Pertambangan. *Seminar Nasional Psikologi Dan Ilmu Humaniora (SENAPIH)*, 97–109.

Sepang, L. K., Tatimu, V., & Rumawas, W. (2023). Pengaruh Pelatihan Kerja Dan Keterlibatan Kerja Terhadap Turnover Intention Karyawan PT. Royal Coconut Airmadidi. In *Productivity* (Vol. 4, Issue 4). https://ejournal.unsrat.ac.id/v3/index.php/productivity/article/view/48268

Setio, P. B. N., Saputro, D. R. S., & Winarno, B. (2020). Klasifikasi dengan Pohon Keputusan Berbasis Algoritme C4.5. *PRISMA, Prosiding Seminar Nasional Matematika*, *3*, 64–71. https://journal.unnes.ac.id/sju/index.php/prisma/article/view/37650

Shafila, G. A. (2020). *Implementasi Metode Extreme Gradient Boosting (XGBoost) untuk Klasifikasi pada Data Bioinformatika (Studi Kasus : Penyakit Ebola, GSE 122692)*. Universitas Islam Indonesia.

Sholikhati, M. E. (2022). *Klasifikasi Penyakit Stroke Menggunakan Metode SMOTE_XGBoost* [Universitas Muhammadiyah Semarang]. http://reader.repository.unimus.ac.id/index.php/display/file/5766/1/

Suyanto. (2022). *Machine Learning Tingkat Dasar Dan Lanjut* (2nd ed.). Informatika Bandung.

Syahrani, I. M. (2019). Analisis Pembandingan Teknik Ensemble Secara Boosting(Xgboost) Dan Bagging (Randomforest) Pada Klasifikasi Kategori Sambatan Sekuens Dna. *Jurnal Penelitian Pos Dan Informatika*, *9*(1), 27. https://doi.org/10.17933/jppi.2019.090103

Syukron, M., Santoso, R., & Widiharih, T. (2020). Perbandingan Metode Smote Random

Forest Dan Smote Xgboost Untuk Klasifikasi Tingkat Penyakit Hepatitis C Pada Imbalance Class Data. *Jurnal Gaussian*, *9*, 227–236. https://doi.org/10.14710/j.gauss.v9i3.28915

Tao, Z., Wu, C., & Zhao, S. (2021). Research on the Prediction of Employee Turnover Behavior and Its Interpretability. *2021 5th International Conference on Electronic Information Technology and Computer Engineering*, 760–767. https://doi.org/10.1145/3501409.3501547

Tharani, S. K. M., & Raj, S. N. V. (2020). Predicting employee turnover intention in IT&ITeS industry using machine learning algorithms. *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*, 508–513.

Tim Datasans. (2023). *Machine Learning Cheatsheet : Model Klasifikasi*. DATASANS. https://datasans.medium.com/

Trivusi. (2022). *Algoritma Random Forest: Pengertian dan Kegunaannya*. https://www.trivusi.web.id/2022/08/algoritma-random-forest.html#kekurangan_algoritma_random_forest

Wahyono, T. (2018). Fundamental of Python for Machine Learning: Dasar-Dasar Pemrograman Python untuk Machine Learning dan Kecerdasan Buatan. In *Gava Media* (Issue September 2018).

Wardhani, N. W. S., Rochayani, M. Y., Iriany, A., Sulistyono, A. D., & Lestantyo, P. (2019). Cross-validation Metrics for Evaluating Classification Performance on Imbalanced Data. *2019 International Conference on Computer, Control, Informatics and Its Applications: Emerging Trends in Big Data and Artificial Intelligence, IC3INA 2019*, 14–18. https://doi.org/10.1109/IC3INA48034.2019.8949568

White, A. (2022). *Organizational Climate Variables and Performance as Predictors of Voluntary and Involuntary Turnover Among Nurses*. Middle Tennessee State University.

Xu, J., Zhang, Y., & Miao, D. (2020). Three-way confusion matrix for classification: A measure driven view. *Information Sciences*, *507*, 772–794. https://doi.org/10.1016/j.ins.2019.06.064

Yang, J., & Guan, J. (2022). *A Heart Disease Prediction Model Based on Feature Optimization and Smote-Xgboost Algorithm*.

Zhao, Y., Hryniewicki, M. K., Cheng, F., Fu, B., & Zhu, X. (2019). Employee turnover prediction with machine learning: A reliable approach. *Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys), 2*, 737–758. https://doi.org/10.1007/978-3-030-01057-7_56

Zuhairah, A. (2022). *Penerapan Algoritma Random Forest, Support Vector Machines (Svm) dan Gradient Boosted Tree (Gbt) Untuk Deteksi Penipuan (Fraud Detection) Pada Transaksi Kartu Kredit*. UIN Syarif Hidayatullah Jakarta.

LAMPIRAN

Script Penelitian

```
STEP 1: IMPORT LIBRARIES
import warnings
warnings.filterwarnings('ignore')

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import OrdinalEncoder
from sklearn.metrics import accuracy_score, precision_score, recall_score
from sklearn.model_selection import StratifiedKFold, RandomizedSearchCV, KFold, cross_val_score
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, roc_curve,
make_scorer
import xgboost as xgb
from xgboost import XGBClassifier
from imblearn.over_sampling import SMOTE

# Set style seaborn
sns.set(style='whitegrid')
STEP 2: READ DATASET
#Load dataset
df = pd.read_csv('Feature Selection.csv')
df.head()
STEP 3: DATASET OVERVIEW
3.1 Dataset Basic Information
#Display a concise summary of the dataframe
df.info()
3.2 Adjust Data for Statistics Summary
#Remove unnecessary features
df = df.drop("ID", axis=1)
df.head()

#Encode Categorical Features
#Import necessary libraries
from sklearn.preprocessing import LabelEncoder
# Create a label encoder object
le = LabelEncoder()
# Define the continuous features
continuous_features = ['Age', 'Training', 'Years of Service', 'Current Role Tenure', 'Last
Promotion', 'Assignment Letter', 'Production Plan']

# Identify the features to be encoded
columns_to_encode = [feature for feature in df.columns if feature not in continuous_features]

# Dictionary to store mapping {original_value: encoded_value}
label_mapping = {}

# Loop over columns to encode
for column in columns_to_encode:
    # Fit and transform the column
    df[column] = le.fit_transform(df[column])

    # Get mapping of original values to encoded values
    label_mapping[column] = dict(zip(le.classes_, le.transform(le.classes_)))

# Print the mappings
for column, mapping in label_mapping.items():
    print(f'Column: {column}')
    for original_value, encoded_value in mapping.items():
        print(f'{original_value}: {encoded_value}')
    print('\n')


#Show data tabel and summary of dataframe
df
```

```
# Get correlation matrix
corr = df.corr()
top_corr_features = corr.index
plt.figure(figsize=(15,7))
g = sns.heatmap(df[top_corr_features].corr(), annot=True, cmap="coolwarm")
# Set a correlation threshold
threshold = 0.6

# Create a boolean mask for correlations greater than the threshold
mask = abs(corr) > threshold

# Get the pairs of variables that have a high correlation
high_corr_var = [(column, row) for column in mask.columns for row in mask.index if
mask[column][row] and column != row]

# Remove duplicate f
high_corr_var = list(set([tuple(sorted(pair)) for pair in high_corr_var]))

# Create a DataFrame to store the pairs and their correlation
corr_df = pd.DataFrame(high_corr_var, columns=['Variabel 1', 'Variabel 2'])

# Add a column for the correlation
corr_df['Nilai Korelasi'] = [corr[pair[0]][pair[1]] for pair in high_corr_var]

# Show the DataFrame
corr_df

#Removing highly correlated features
col_to_drop = ['Generation','Specialization Area']
df = df.drop(columns = col_to_drop, axis=1)
df.head()
3.3 Data Type Transformation

# Identify the features to be converted to object data type
features_to_convert = [feature for feature in df.columns if feature not in continuous_features]

# Convert the identified features to object data type
df[features_to_convert] = df[features_to_convert].astype('object')

df.dtypes
3.4 Statistics Summary for Numerical Variables

# Get the summary statistics for numerical variables
df.describe().T
3.5 Statistics Summary for Categorical Variables
# Get the summary statistics for categorical variables
df.describe(include='object').T
STEP 4: EDA
4.1 Count and Percentage of Target
#Pie diagram of target features
count = [df["Leave"].value_counts()[i] for i in df["Leave"].unique()]
labels = [f"{i} ({count:,})" for i, count in zip(df["Leave"].unique(), count)]
chroma = ["#B0C8F6", "#F5C0A7"]
plt.pie(count, explode = (0, 0.1), autopct='%1.1f%%', colors = chroma, startangle = 90,
shadow=True, labels=labels)
plt.title('Count and Percentage of Leave')
plt.axis('equal')
plt.show()
4.2 Numerical Features vs Target
# Create subplots for kde plots
fig, axes = plt.subplots(4, 2, figsize=(17, 15))

for ax, col in zip(axes.flatten(), continuous_features):
    sns.kdeplot(data=df, x=col, fill=True, linewidth=2, hue='Leave', ax=ax, palette = {0:
'#5B7AE5', 1: '#E97A5F'})
    ax.set_title(f'{col} vs Target')

axes[3,1].axis('off')
plt.suptitle('Distribution of Continuous Features by Target', fontsize=22)
plt.tight_layout()
plt.show()
4.3 Categorical Features vs Target
# List of categorical features
cat_features = [feature for feature in df.columns if feature not in continuous_features]

# Initialize the plot
fig, axes = plt.subplots(5, 2, figsize=(17, 18))




# Plot each feature
for i, ax in enumerate(axes.flatten()):
```

```
    sns.countplot(x=cat_features[i], hue='Leave', data=df, ax=ax, palette={0: '#5B7AE5', 1:
'#E97A5F'})
    ax.set_title(cat_features[i])
    ax.set_ylabel('Count')
    ax.set_xlabel('')
    ax.legend(title='Leave', loc='upper right')

plt.suptitle('Distribution of Categorical Features by Target', fontsize=22)
plt.tight_layout()
plt.show()
STEP 5: DATA PREPROCESSING
df = pd.read_csv('Feature Selection.csv')
col_to_drop = ['ID','Generation','Specialization Area']
df = df.drop(columns = col_to_drop, axis=1)
df.head()
5.1 Encode Ordinal Categorical Features
from sklearn.preprocessing import OrdinalEncoder

# Define the order of categories for each variable
categories = {
    'Last Education': ['SMK', 'D1', 'D3', 'S1', 'S2'],  # Last Education
    'Job Level': ['Pratama', 'Madya', 'Utama'],  # Job Level
    'Competency': ['A', 'B', 'C', 'D']  # Competency
}

# Define the columns to be encoded
columns_to_oe = ['Last Education', 'Job Level', 'Competency']

# Loop over columns to encode
for column in columns_to_oe:

    # Print the number of each category before encoding
    print(f'\nColumn Count: {column}')
    print(df[column].value_counts())

    # Create encoder with categories for current column
    oe = OrdinalEncoder(categories=[categories[column]])

    # Reshape the column into a 2D array
    column_2d = df[column].values.reshape(-1, 1)

    # Create a copy of the column before transformation
    column_before = df[column].copy()

    # Fit and transform the 2D array
    df[column] = oe.fit_transform(column_2d)

    # Print each original category with its encoded value
    print('\nMapping:')
    for category in column_before.unique():
        print(f'{category}: {df[column][column_before == category].unique()[0]}')

# Show data table and summary of dataframe
df
5.2 Encode Non Ordinal Categorical Features
#Encode target feature
df['Leave'] = df['Leave'].map({'No': 0, 'Yes': 1})

from sklearn.preprocessing import OneHotEncoder

# Define the columns to be encoded
columns_to_ohe = ['Gender', 'Marital Status', 'Education Field', 'Site', 'Department', 'Job Role']

# Create the OneHotEncoder
ohe = OneHotEncoder(sparse=False)

# Fit and transform the columns
encoded_columns = ohe.fit_transform(df[columns_to_ohe])

# Get feature names
feature_names = ohe.get_feature_names_out(columns_to_ohe)

# Convert the encoded columns into a DataFrame
df_encoded = pd.DataFrame(encoded_columns, columns=feature_names)

# Drop the original columns from df and add the encoded ones
df = df.drop(columns_to_ohe, axis=1)
df = pd.concat([df, df_encoded], axis=1)

# Show the DataFrame
df

5.3 Split the Dataset
# Define your features and target variable
```

```
X = df.drop('Leave', axis=1)
y = df['Leave']

# Inisialisasi StratifiedKFold
skf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

#Train-Test Split using StratifiedKFold
for train_index, test_index in skf.split(X, y):
    X_train, X_test = X.iloc[train_index], X.iloc[test_index]
    y_train, y_test = y.iloc[train_index], y.iloc[test_index]

# Training data shape before SMOTE
print("X_train before SMOTE: ", X_train.shape)
print("y_train before SMOTE: ", y_train.shape)

# Testing data shape
print("X_test: ", X_test.shape)
print("y_test: ", y_test.shape)

# Sample Class proportion
print("Jumlah sampel positif dalam training set: ", y_train.sum())
print("Jumlah sampel negatif dalam training set: ", len(y_train) - y_train.sum())

print("Jumlah sampel positif dalam test set: ", y_test.sum())
print("Jumlah sampel negatif dalam test set: ", len(y_test) - y_test.sum())
STEP 6: XGBoost without SMOTE
# Define the model
xgb_base = xgb.XGBClassifier()
6.1 Hyperparameter Tuning
# Define the hyperparameters to tune
scale_pos_weight_value_XGB = scale_pos_weight=(len(y_train) - y_train.sum()) / y_train.sum()
param_grid = {
    "reg_alpha": np.arange(0, 1.1, 0.1),  # menghasilkan array [0.0, 0.1, ..., 0.5]
    "reg_lambda": np.arange(0, 1.1, 0.1),  # menghasilkan array [0.0, 0.1, ..., 0.5]
    "scale_pos_weight": [scale_pos_weight_value_XGB],
    "max_leaves": list(range(2, 11)),
    "n_estimators": [100, 200, 300],
    "max_depth": [4, 5, 6, 7, 8],
    "min_child_weight": [0, 1, 2, 3, 4, 5, 6, 7],
    "learning_rate": [0.01, 0.025, 0.05, 0.1, 0.2, 0.3],
    "gamma": [0, 0.1, 0.2, 0.3, 0.4, 1.0, 1.5, 2.0],
    "colsample_bylevel" : ['log2', 'sqrt', 0.25, 1.0],
    "subsample" : [0.15, 0.5, 0.75, 1.0]
}

# Best K for parameter grid
kf_values = [2, 3, 4, 5, 6, 7, 8, 9, 10]
best_k = None
best_score = 0

for k in kf_values:
    kf = KFold(n_splits=k, shuffle=True, random_state=42)

    # Calculate performance score on each K-Fold
    scores = cross_val_score(xgb_base, X_train, y_train, cv=kf, scoring='roc_auc')

    # Calculate average score
    avg_score = scores.mean()

    # Choose the best K-Fold based on average score
    if avg_score > best_score:
        best_score = avg_score
        best_k = k

print(f"Best K-Fold: {best_k}, Best Score: {best_score}")

from sklearn.model_selection import RandomizedSearchCV

# Initialize XGBoost classifier
xgb_base = xgb.XGBClassifier()

# Initialize RandomizedSearchCV
random_search = RandomizedSearchCV(xgb_base, param_distributions=param_grid, n_iter=10, cv=best_k,
scoring='roc_auc', random_state=42)

# Fit the RandomizedSearchCV object
random_search_fitXGB = random_search.fit(X_train, y_train)

# Print the best parameters and the corresponding score
print(f"Best parameters: {random_search.best_params_}")
print(f"Best score: {random_search.best_score_}")

6.2 Classification
# Train the Model
```

```
best_params = random_search.best_params_
# xgb_base = xgb.XGBClassifier(scale_pos_weight=(len(y_train) - y_train.sum()) / y_train.sum(),
**best_params)
xgb_base = xgb.XGBClassifier(**best_params)
xgb_base.fit(X_train, y_train)

# Show scale_pos_weight value
print(xgb_base.scale_pos_weight)

# Predict the test set results
y_pred_base = xgb_base.predict(X_test)
6.3 Model Evaluation
# Model Evaluation for XGBoost Tanpa SMOTE
# Print the classification report
print("Algorithm: XGBoost Tanpa SMOTE\n")
print("Classification report:")
print(classification_report(y_test, y_pred_base))

# Print the accuracy score
print(f"Accuracy Score: {accuracy_score(y_test, y_pred_base)}")

# Accuracy score of training data
train_accuracy_base = accuracy_score(y_train, xgb_base.predict(X_train))
print(f"Train Accuracy: {train_accuracy_base}")

# Accuracy score of testing data
test_accuracy_base = accuracy_score(y_test, y_pred_base)
print(f"Test Accuracy: {test_accuracy_base}")

# Compute and print the AUC score
y_pred_proba_base = xgb_base.predict_proba(X_test)[:, 1]
auc_score_base = roc_auc_score(y_test, y_pred_proba_base)
print(f"Area under ROC curve: {auc_score_base}")

# Plot the confusion matrix
cm_base = confusion_matrix(y_test, y_pred_base)
plt.figure(figsize=(5, 5))
sns.heatmap(cm_base, annot=True, fmt=".0f", linewidths=.5, square=True, cmap='Blues_r')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

# Compute and plot the ROC curve
fpr_base, tpr_base, thresholds_base = roc_curve(y_test, y_pred_proba_base)
plt.figure(figsize=(8, 6))
plt.plot(fpr_base, tpr_base, label='ROC curve (area = %0.2f)' % auc_score_base)
plt.scatter(fpr_base, tpr_base, color='blue', marker='o')  # Menambahkan titik pada setiap titik
ROC
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - XGBoost Tanpa SMOTE')
plt.legend(loc="lower right")
plt.show()

# Save model to file
import joblib
model_base = 'xgb_base_model.pkl'
joblib.dump(xgb_base, model_base)
STEP 7: XGBoost with SMOTE
7.1 SMOTE Implementation
# Inisialisasi SMOTE
smote = SMOTE(random_state=42)

# Fit dan resample training data
X_train_res, y_train_res = smote.fit_resample(X_train, y_train)

# Training data shape after SMOTE
print("X_train after SMOTE: ", X_train_res.shape)
print("y_train after SMOTE: ", y_train_res.shape)

# Testing data shape
print("X_test: ", X_test.shape)
print("y_test: ", y_test.shape)
7.2 Hyperparameter Tuning
# Define the model
xgb_base = xgb.XGBClassifier()
# Define the hyperparameters to tune
scale_pos_weight_value_smote = scale_pos_weight=(len(y_train_res) - y_train_res.sum()) /
y_train_res.sum()

param_grid = {
    "reg_alpha": np.arange(0, 1.1, 0.1),  # menghasilkan array [0.0, 0.1, ..., 0.5]
```

```
    "reg_lambda": np.arange(0, 1.1, 0.1),  # menghasilkan array [0.0, 0.1, ..., 0.5]
    "scale_pos_weight": [scale_pos_weight_value_smote],
    "max_leaves": list(range(2, 11)),
    "n_estimators": [100, 200, 300],
    "max_depth": [4, 5, 6, 7, 8],
    "min_child_weight": [0, 1, 2, 3, 4, 5, 6, 7],
    "learning_rate": [0.01, 0.025, 0.05, 0.1, 0.2, 0.3],
    "gamma": [0, 0.1, 0.2, 0.3, 0.4, 1.0, 1.5, 2.0],
    "colsample_bylevel" : ['log2', 'sqrt', 0.25, 1.0],
    "subsample" : [0.15, 0.5, 0.75, 1.0]
}

# Best K for parameter grid
kf_values = [2, 3, 4, 5, 6, 7, 8, 9, 10]
best_k = None
best_score = 0

for k in kf_values:
    kf = KFold(n_splits=k, shuffle=True, random_state=42)

    # Calculate performance score on each K-Fold
    scores = cross_val_score(xgb_base, X_train_res, y_train_res, cv=kf, scoring='roc_auc')

    # Calculate average score
    avg_score = scores.mean()

    # Choose the best K-Fold based on average score
    if avg_score > best_score:
        best_score = avg_score
        best_k = k

print(f"Best K-Fold: {best_k}, Best Score: {best_score}")

# Initialize XGBoost classifier
xgb_base = xgb.XGBClassifier()

# Initialize RandomizedSearchCV
random_search = RandomizedSearchCV(xgb_base, param_distributions=param_grid, n_iter=10, cv=best_k,
scoring='roc_auc', random_state=42)

# Fit the RandomizedSearchCV object
random_search_fitXGB = random_search.fit(X_train_res, y_train_res)

# Print the best parameters and the corresponding score
print(f"Best parameters: {random_search.best_params_}")
print(f"Best score: {random_search.best_score_}")
7.3 Classification
# Train the Model
best_params = random_search.best_params_
# xgb_smote = xgb.XGBClassifier(scale_pos_weight=(len(y_train_res) - y_train_res.sum()) /
y_train_res.sum(), **best_params)
xgb_smote = xgb.XGBClassifier(**best_params)
xgb_smote.fit(X_train_res, y_train_res)

# Show scale_pos_weight value
print(xgb_smote.scale_pos_weight)

# Predict the test set results
y_pred_smote = xgb_smote.predict(X_test)
7.4 Model Evaluation
# Model Evaluation for XGBoost Dengan SMOTE
# Print the classification report
print("Algorithm: SMOTEXGBClassifier\n")
print("Classification report:")
print(classification_report(y_test, y_pred_smote))

# Print the accuracy score
print(f"Accuracy Score: {accuracy_score(y_test, y_pred_smote)}")

# Accuracy score of training data
train_accuracy_smote = accuracy_score(y_train_res, xgb_smote.predict(X_train_res))
print(f"Train Accuracy: {train_accuracy_smote}")

# Accuracy score of testing data
test_accuracy_smote = accuracy_score(y_test, y_pred_smote)
print(f"Test Accuracy: {test_accuracy_smote}")

# Compute and print the AUC score
y_pred_proba_smote = xgb_smote.predict_proba(X_test)[:, 1]
auc_score_smote = roc_auc_score(y_test, y_pred_proba_smote)
print(f"Area under ROC curve: {auc_score_smote}")

# Plot the confusion matrix
cm_smote = confusion_matrix(y_test, y_pred_smote)
```

```
plt.figure(figsize=(5, 5))
sns.heatmap(cm_smote, annot=True, fmt=".0f", linewidths=.5, square=True, cmap='Blues_r')
plt.ylabel('Actual label')
plt.xlabel('Predicted label')

# Compute and plot the ROC curve
fpr_smote, tpr_smote, thresholds_smote = roc_curve(y_test, y_pred_proba_smote)
plt.figure(figsize=(8, 6))
plt.plot(fpr_smote, tpr_smote, label='ROC curve (area = %0.2f)' % auc_score_smote)
plt.scatter(fpr_smote, tpr_smote, color='blue', marker='o')  # Menambahkan titik pada setiap titik
ROC
plt.plot([0, 1], [0, 1], 'k--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - SMOTEXGBClassifier')
plt.legend(loc="lower right")
plt.show()

# Save model to file
model_smote = 'xgb_smote_model.pkl'
joblib.dump(xgb_smote, model_smote)
Which One is Better?
import joblib
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

# Load a saved model
xgb_base_model = joblib.load('xgb_base_model.pkl')
xgb_smote_model = joblib.load('xgb_smote_model.pkl')

# Predict results for test sets
y_pred_base = xgb_base_model.predict(X_test)
y_pred_smote = xgb_smote_model.predict(X_test)

# Calculates the probability prediction value for ROC AUC
y_pred_proba_base = xgb_base_model.predict_proba(X_test)[:, 1]
y_pred_proba_smote = xgb_smote_model.predict_proba(X_test)[:, 1]

# Calculate ROC AUC
roc_auc_base = roc_auc_score(y_test, y_pred_proba_base)
roc_auc_smote = roc_auc_score(y_test, y_pred_proba_smote)

# Calculate confusion matrix
cm_base = confusion_matrix(y_test, y_pred_base)
cm_smote = confusion_matrix(y_test, y_pred_smote)

# Calculate evaluation metrics
accuracy_base = accuracy_score(y_test, y_pred_base)
precision_base = precision_score(y_test, y_pred_base)
recall_base = recall_score(y_test, y_pred_base)

accuracy_smote = accuracy_score(y_test, y_pred_smote)
precision_smote = precision_score(y_test, y_pred_smote)
recall_smote = recall_score(y_test, y_pred_smote)

# Data for comparation
metrics = ['Accuracy', 'Precision', 'Recall', 'ROC AUC']
values_base = [accuracy_base, precision_base, recall_base, roc_auc_base]
values_smote = [accuracy_smote, precision_smote, recall_smote, roc_auc_smote]

# Bar plot for accuracy, precision, recall, and ROC AUC comparation
bar_width = 0.35
index = np.arange(len(metrics))

fig, ax = plt.subplots(figsize=(12, 6))
bar1 = ax.bar(index, values_base, bar_width, label='XGBoost Tanpa SMOTE')
bar2 = ax.bar(index + bar_width, values_smote, bar_width, label='XGBoost Dengan SMOTE')

ax.set_xlabel('Metrik Evaluasi')
ax.set_ylabel('Nilai Metrik')
ax.set_title('Perbandingan Performa XGBoost Tanpa dan Dengan SMOTE')
ax.set_xticks(index + bar_width / 2)
ax.set_xticklabels(metrics)
ax.legend()

# Add value to bar
for rect in bar1:
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width() / 2, height, f'{height:.2f}', ha='center',
va='bottom')
```

```
for rect in bar2:
    height = rect.get_height()
    ax.text(rect.get_x() + rect.get_width() / 2, height, f'{height:.2f}', ha='center',
va='bottom')

plt.show()
Cek Overfit or Not
import pickle
from sklearn.metrics import accuracy_score, precision_score, recall_score, roc_auc_score, f1_score

# Load the saved models
with open('xgb_base_model.pkl', 'rb') as file:
    xgb_base_model = pickle.load(file)

with open('xgb_smote_model.pkl', 'rb') as file:
    xgb_smote_model = pickle.load(file)

# Predictions on training set
y_train_pred_base = xgb_base_model.predict(X_train)
y_train_pred_smote = xgb_smote_model.predict(X_train_res)

# Predictions on test set
y_test_pred_base = xgb_base_model.predict(X_test)
y_test_pred_smote = xgb_smote_model.predict(X_test)

# Evaluate accuracy, precision, recall, F1 score, and ROC AUC on training set
train_accuracy_base = accuracy_score(y_train, y_train_pred_base)
train_precision_base = precision_score(y_train, y_train_pred_base)
train_recall_base = recall_score(y_train, y_train_pred_base)
train_f1_base = f1_score(y_train, y_train_pred_base)
train_roc_auc_base = roc_auc_score(y_train, xgb_base_model.predict_proba(X_train)[:, 1])

train_accuracy_smote = accuracy_score(y_train_res, y_train_pred_smote)
train_precision_smote = precision_score(y_train_res, y_train_pred_smote)
train_recall_smote = recall_score(y_train_res, y_train_pred_smote)
train_f1_smote = f1_score(y_train_res, y_train_pred_smote)
train_roc_auc_smote = roc_auc_score(y_train_res, xgb_smote_model.predict_proba(X_train_res)[:, 1])

# Evaluate accuracy, precision, recall, F1 score, and ROC AUC on test set
test_accuracy_base = accuracy_score(y_test, y_test_pred_base)
test_precision_base = precision_score(y_test, y_test_pred_base)
test_recall_base = recall_score(y_test, y_test_pred_base)
test_f1_base = f1_score(y_test, y_test_pred_base)
test_roc_auc_base = roc_auc_score(y_test, xgb_base_model.predict_proba(X_test)[:, 1])

test_accuracy_smote = accuracy_score(y_test, y_test_pred_smote)
test_precision_smote = precision_score(y_test, y_test_pred_smote)
test_recall_smote = recall_score(y_test, y_test_pred_smote)
test_f1_smote = f1_score(y_test, y_test_pred_smote)
test_roc_auc_smote = roc_auc_score(y_test, xgb_smote_model.predict_proba(X_test)[:, 1])

# Display the results
print("XGBoost Tanpa SMOTE - Training Set:")
print(f"Accuracy: {train_accuracy_base:.4f}, Precision: {train_precision_base:.4f}, Recall:
{train_recall_base:.4f}, F1-Score: {train_f1_base:.4f}, ROC AUC: {train_roc_auc_base:.4f}")

print("\nXGBoost Dengan SMOTE - Training Set:")
print(f"Accuracy: {train_accuracy_smote:.4f}, Precision: {train_precision_smote:.4f}, Recall:
{train_recall_smote:.4f}, F1-Score: {train_f1_smote:.4f}, ROC AUC: {train_roc_auc_smote:.4f}")

print("\nXGBoost Tanpa SMOTE - Test Set:")
print(f"Accuracy: {test_accuracy_base:.4f}, Precision: {test_precision_base:.4f}, Recall:
{test_recall_base:.4f}, F1-Score: {test_f1_base:.4f}, ROC AUC: {test_roc_auc_base:.4f}")

print("\nXGBoost Dengan SMOTE - Test Set:")
print(f"Accuracy: {test_accuracy_smote:.4f}, Precision: {test_precision_smote:.4f}, Recall:
{test_recall_smote:.4f}, F1-Score: {test_f1_smote:.4f}, ROC AUC: {test_roc_auc_smote:.4f}")

Visualisasi Feature Importance
from xgboost import plot_importance
import matplotlib.pyplot as plt
import seaborn as sns
colors = sns.color_palette("muted", len(xgb_smote.feature_importances_))

# Plot feature importance
fig, ax = plt.subplots(figsize=(16, 8))
plt.subplots_adjust(left=0.2, right=0.8, top=0.9, bottom=0.1, wspace=0.5, hspace=0.5)
max_num_features = 15
plot_importance(xgb_smote, importance_type='weight', xlabel='Weight', ylabel='Features',
title='Feature Importance (Weight)', show_values=True, ax=ax, max_num_features=max_num_features,
color=colors)
plt.show()
```