

00 DETEKSI *CREDIT CARD FRAUD* PERBANKAN
MENGUNAKAN METODE *ENSEMBLE STACKING*



SKRIPSI

Diajukan sebagai salah satu syarat untuk menyelesaikan Pendidikan Diploma Empat
(D-4) Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro
Politeknik Negeri Ujung Pandang

ANDI MUH.MUHAYYAN

42518003

PROGRAM STUDI TEKNIK KOMPUTER JARINGAN
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI UJUNG PANDANG MAKASSAR

2023

HALAMAN PENGESAHAN

Skripsi dengan judul “DETEKSI *CREDIT CARD FRAUD* PERBANKAN MENGGUNAKAN METODE *ENSEMBLE STACKING*” oleh Andi Muh.Muhayyan Nomor Induk Mahasiswa 425 18 003 telah diterima dan disahkan sebagai salah satu syarat untuk memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Makassar, 21 Agustus 2023

Mengesahkan,

Pembimbing I,



Mevlanie Olivya, S.T., M.T.
NIP. 198205032014042002

Pembimbing II,



Irfan Syamsuddin S.T., M.Com.ISM., Ph.D.
NIP. 197312202000031008

Mengetahui,

Koordinator Program Studi
Teknik Komputer dan Jaringan
Politeknik Negeri Ujung Pandang



Eddy Tungadi, S.T., M.T.
NIP. 197908232010121001

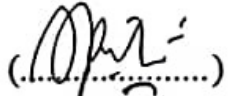
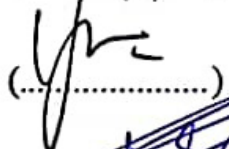


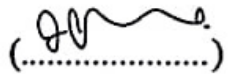

HALAMAN PENERIMAAN

Pada hari ini _____, Tim Penguji Ujian Skripsi telah menerima dengan baik skripsi oleh mahasiswa: Andi Muh.Muhayyan NIM 425 18 003 dengan judul ” **DETEKSI CREDIT CARD FRAUD PERBANKAN MENGGUNAKAN METODE ENSEMBLE STACKING**”

Makassar,

2023

Tim Penguji Ujian Sidang Skripsi:

1. Irmawati, S.T.,M.T.	Ketua	()
2. Muhammad Nur Yasir Utomo, S.ST, M.Eng.	Sekretaris	()
3. Muh. Fajri Raharjo, S.T, M.T.	Anggota	()
4. Eddy Tungadi, S.T.,M.T.	Anggota	()
5. Meylanie Olivya,S.T.,M.T.	Anggota	()
6. Prof.Irfan Syamsuddin,S.T. M.Com.ISM,Ph.D.	Anggota	()

KATA PENGANTAR

Puji syukur senantiasa penulis panjatkan ke hadirat Allah SWT atas Berkah dan Rahmat-Nya yang telah memberikan kesehatan dan keselamatan kepada penulis sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat dan salam kepada baginda Rasul Muhammad SAW sebagai sebaik-baik panutan bagi seluruh manusia.

Skripsi ini disusun guna memenuhi salah satu syarat untuk menyelesaikan studi serta dalam rangka memperoleh gelar Diploma IV (D-4/S1 Terapan) pada Program Studi Teknik Komputer dan Jaringan Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

Penulis menyadari bahwa keberhasilan penyusunan skripsi ini tidak lepas dari bantuan berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, dengan rendah hati penulis mengucapkan terima kasih kepada:

1. Orang tua penulis yakni A.M.Syukur dan Andi Kartini, serta keluarga besar Penulis yang sampai saat ini selalu memberikan semangat, motivasi, dukungan, bimbingan dan doa kepada penulis.
2. Bapak Ir. Ilyas Mansur, M.T. selaku Direktur Politeknik Negeri Ujung Pandang.
3. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.
4. Bapak Eddy Tungadi, S.T., M.T. selaku Koordinator Program Studi Teknik Komputer dan Jaringan.
5. Ibu Meylanie Olivya, S.T., M.T. selaku pembimbing I, dan Bapak Irfan Syamsuddin, S.T. M.Com.ISM., Ph.D. selaku pembimbing II. Terima kasih atas segala ilmu, motivasi, nasehat, arahan, bantuan dan kesediaan waktu dan kesabarannya dalam membimbing penulis hingga terselesaikannya penelitian ini.
6. Seluruh dosen dan Staf Jurusan Teknik Elektro, Khususnya Program Studi D4 Teknik Komputer dan Jaringan.
7. Teman-teman seperjuangan di Program Studi TKJ angkatan 2018 dan khususnya

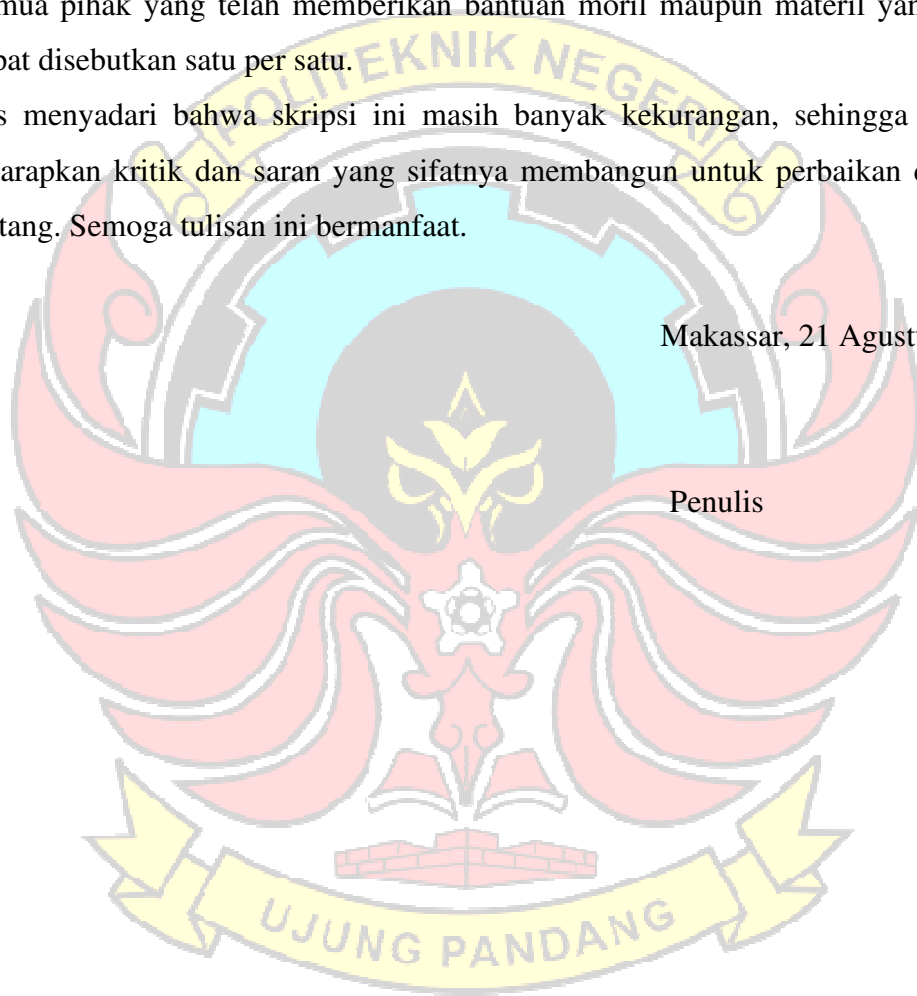
teman-teman TKJ A angkatan 2018 yang telah berjuang bersama selama 4 tahun dan mengajarkan banyak hal kepada penulis baik dari segi akademik maupun non akademik.

8. Kepada Wulandari Putri yang telah kebersamai dan memberikan semangat kepada penulis sehingga skripsi ini dapat terselesaikan.
9. Semua pihak yang telah memberikan bantuan moril maupun materil yang tidak dapat disebutkan satu per satu.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan, sehingga penulis mengharapkan kritik dan saran yang sifatnya membangun untuk perbaikan di masa mendatang. Semoga tulisan ini bermanfaat.

Makassar, 21 Agustus 2023

Penulis



DAFTAR ISI

HALAMAN PENGESAHAN.....	ii
HALAMAN PENERIMAAN.....	iii
KATA PENGANTAR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	xi
SURAT PERNYATAAN.....	xii
RINGKASAN.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Ruang Lingkup Penelitian.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	3
BAB II TINJAUAN PUSTAKA.....	5
2.1. Kejahatan Perbankan.....	5
2.2. <i>Fraud</i>	6
2.3. Sejarah <i>Credit Card Fraud</i> di Indonesia.....	7
2.4. Asal Mula <i>Credit Card</i> Dipalsukan.....	9
2.5. Machine Learning.....	9
2.5.1. Definisi <i>Machine Learning</i>	9
2.5.2. Kategori <i>Machine Learning</i>	10
2.6. <i>Data Mining</i>	12
2.7. <i>Ensemble Learning</i>	15
2.8. Klasifikasi.....	18
2.9. <i>Confusion Matrix</i>	27
2.10. <i>Imbalanced Data</i>	29

2.11. <i>Random Oversampling</i>	30
2.12. <i>K-Fold Cross Validation</i>	30
2.13. <i>Information Gain</i>	31
2.14. <i>Orange Data Mining</i>	31
2.15. Penelitian Terkait.....	32
BAB III METODELOGI PENELITIAN	36
3.1. Tempat dan Waktu Penelitian.....	36
3.2. Alat dan Bahan Penelitian.....	36
3.2.1 Perangkat Keras (<i>Hardware</i>).....	36
3.2.2 Perangkat Lunak (<i>Software</i>)	37
3.3 Metode Penelitian	37
3.3.1 Studi Literatur.....	37
3.3.2 Pengumpulan Data.....	38
3.3.3 <i>Preprocessing</i>	38
3.3.4 Klasifikasi.....	39
3.3.5 Metode <i>Ensemble Stacking</i>	41
3.3.6 Analisis Hasil.....	41
BAB IV HASIL DAN PEMBAHASAN	43
4.1. Deskripsi Dataset	43
4.2. <i>Preprocessing</i>	45
4.2.1 <i>Data Cleaning</i>	45
4.2.2 <i>Data Normalization</i>	46
4.2.3 <i>Feature Selection</i>	47
4.3. Klasifikasi.....	49
4.3.1 Tanpa Metode <i>Ensemble Stacking</i>	50
4.3.2 Menggunakan Metode <i>Ensemble Stacking Imbalanced Data</i>	54
4.3.3 Menggunakan Metode <i>Ensemble Stacking Dengan Class Balancing</i>	57
4.3.4 Menggunakan Metode <i>Ensemble Stacking Class Balancing Dengan</i>	

Masing-Masing Algoritma Sebagai <i>Aggregate</i>	62
4.4. Analisis Hasil.....	67
4.4.1 <i>Area Under Curve</i> (AUC).....	67
4.4.2 <i>F1-Score</i>	69
4.4.3 <i>Precision</i>	70
4.4.4 <i>Recall</i>	71
4.4.5 <i>Specificity</i>	72
4.4.6 Akurasi (CA)	74
4.4.7 Durasi Pengujian.....	75
4.4.8 Perbandingan Metode <i>Ensemble Stacking Imbalanced Data, Ensemble Stacking Class Balancing</i> dan Metode <i>Smote</i>	76
4.4.9 Perbandingan performa model tanpa <i>Ensemble Stacking</i> dan Metode <i>Ensemble Stacking</i> dengan data normal	79
BAB V PENUTUP.....	83
5.1 Kesimpulan	83
5.2 Saran	84
DAFTAR PUSTAKA	86
LAMPIRAN.....	90

DAFTAR GAMBAR

Gambar 2. 1 <i>Bagging</i> dalam <i>machine learning</i>	16
Gambar 2. 2 <i>Boosting</i> dalam <i>machine learning</i>	17
Gambar 2. 3 <i>Stacking</i> dalam <i>machine learning</i>	18
Gambar 2. 4 Model <i>Random Forest</i>	21
Gambar 2. 5 Model SVM.....	22
Gambar 2. 6 Model <i>Decision Tree</i>	23
Gambar 2. 7 Model <i>Neural Network</i>	24
Gambar 2. 8 Model KNN.....	26
Gambar 2. 9 Tabel <i>Confusion Matrix</i>	27
Gambar 2. 10 <i>Imbalanced Data</i>	29
Gambar 2. 11 Logo Orange.....	32
Gambar 3. 1 Alur Penelitian.....	37
Gambar 3. 2 <i>Preprocessing Data</i>	38
Gambar 3. 3 <i>Flowchart Ensemble Stacking</i>	41
Gambar 3. 4 Alur Proses Klasifikasi.....	40
Gambar 4. 1 Struktur Dataset.....	45
Gambar 4. 2 <i>Import Dataset</i>	45
Gambar 4. 3 Data Target.....	46
Gambar 4. 4 <i>Data Cleaning</i>	46
Gambar 4. 5 <i>Data Transformation</i>	47
Gambar 4. 6 Data Sebelum <i>Reduction</i>	48
Gambar 4. 7 Proses <i>Feature Selection</i>	48
Gambar 4. 8 Data setelah <i>Feature Selection</i>	49
Gambar 4. 9 Pembagian Data.....	50
Gambar 4. 10 Proporsi Pembagian Data.....	51
Gambar 4. 11 Penerapan Model Klasifikasi	51
Gambar 4. 12 <i>Cross Validation</i>	52

Gambar 4. 13 Pengujian Data <i>Testing</i> Tanpa Metode <i>Ensemble Stacking</i>	52
Gambar 4. 14 Alur Pengujian Tanpa Metode <i>Ensemble Stacking</i>	53
Gambar 4. 15 Jumlah <i>Features</i>	54
Gambar 4. 16 <i>Information Gain</i>	55
Gambar 4. 17 <i>Features Relevant</i>	55
Gambar 4. 18 Pembagian <i>Data Training</i> dan <i>Data Testing</i>	55
Gambar 4. 19 Pengujian Data <i>Testing</i> Dengan Metode <i>Ensemble Stacking</i> <i>Imbalanced Data</i>	56
Gambar 4. 20 Visualisasi Target Y Sebelum <i>Class Balancing</i>	57
Gambar 4. 21 Memilih <i>Class</i> Untuk <i>Balancing</i>	58
Gambar 4. 22 <i>Random Oversampling</i>	59
Gambar 4. 23 Penggabungan Data.....	59
Gambar 4. 24 Visualisasi <i>Class</i> Setelah <i>Balancing</i>	60
Gambar 4. 25 Alur Pengujian <i>Ensemble Stacking</i> Dengan <i>Class Balancing</i>	61
Gambar 4. 26 Alur Pengujian Algoritma <i>Random Forest</i> Sebagai <i>Aggregate</i>	62
Gambar 4. 27 Alur Pengujian Algoritma <i>Neural Network</i> Sebagai <i>Aggregate</i>	63
Gambar 4. 28 Alur Pengujian Algoritma <i>Decision Tree</i> Sebagai <i>Aggregate</i>	64
Gambar 4. 29 Alur Pengujian Algoritma <i>kNN</i> Sebagai <i>Aggregate</i>	65
Gambar 4. 30 Alur Pengujian Algoritma <i>SVM</i> Sebagai <i>Aggregate</i>	66
Gambar 4. 31 Diagram Batang Nilai <i>Area Under Curve (AUC)</i>	68
Gambar 4. 32 Diagram Batang Nilai <i>F1- Score</i>	69
Gambar 4. 33 Diagram Batang Nilai <i>Precision</i>	70
Gambar 4. 34 Diagram Batang Nilai <i>Recall</i>	71
Gambar 4. 35 Diagram batang Nilai <i>Specificity</i>	72
Gambar 4. 36 Diagram batang Nilai Akurasi.....	74
Gambar 4. 37 Perbandingan nilai <i>Train Time</i>	77
Gambar 4. 38 Perbandingan nilai <i>Test time</i>	78
Gambar 4. 39 Perbandingan nilai <i>Train Time</i>	81
Gambar 4. 40 Perbandingan nilai <i>Test time</i>	81

DAFTAR TABEL

Tabel 4. 1 Informasi Dataset	43
Tabel 4. 2 Performa Model Tanpa Metode <i>Ensemble Stacking</i>	52
Tabel 4. 3 Performa Model Dengan Metode <i>Ensemble Stacking Imbalanced Data</i> ..	56
Tabel 4. 4 Performa Stacking dengan Class Balancing	61
Tabel 4. 5 Performa Model <i>Stacking Class Balancing</i> Dengan <i>Random Forest</i> Sebagai <i>Aggregate</i>	63
Tabel 4. 6 Performa Model <i>Stacking Class Balancing</i> Dengan <i>Neural Network</i> Sebagai <i>Aggregate</i>	64
Tabel 4. 7 Performa Model <i>Stacking Class Balancing</i> Dengan <i>Decision Tree</i> Sebagai <i>Aggregate</i>	65
Tabel 4. 8 Performa Model <i>Stacking Class Balancing</i> Dengan <i>kNN</i> Sebagai <i>Aggregate</i>	66
Tabel 4. 9 Performa Model <i>Stacking Class Balancing</i> Dengan <i>SVM</i> Sebagai <i>Aggregate</i>	66
Tabel 4. 10 Durasi Pengujian Tanpa <i>Ensemble Stacking</i>	75
Tabel 4. 11 Durasi Pengujian Menggunakan <i>Ensemble Stacking Imbalanced Data</i> ..	75
Tabel 4. 12 Durasi Pengujian Menggunakan <i>Ensemble Stacking dengan Class</i> <i>Balancing</i>	75
Tabel 4. 13 Durasi Pengujian Menggunakan <i>Ensemble Stacking Class Balancing</i> dengan <i>Aggregate</i>	76
Tabel 4. 14 Perbandingan <i>Ensemble Stacking Imbalance Data</i> , <i>Ensemble Stacking</i> <i>Class Balancing</i> dan Metode <i>Smote</i>	76
Tabel 4. 15 Performa Model Tanpa Metode <i>Ensemble Stacking</i>	79
Tabel 4. 16 Performa Metode <i>Ensemble Stacking</i> data normal	79

SURAT PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Andi Muh.Muhayyan

NIM : 425 18 003

Menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul “**Deteksi Credit Card Fraud Perbankan Menggunakan Metode Ensemble Stacking**” merupakan gagasan dan hasil karya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi mana pun.

Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut di atas tidak benar, saya siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, 21 Agustus 2023

Andi Muh.Muhayyan

NIM. 42518003

DETEKSI *CREDIT CARD FRAUD* PERBANKAN MENGGUNAKAN METODE *ENSEMBLE STACKING*

RINGKASAN

Kejahatan yang melibatkan transaksi di bank hingga saat ini masih sulit diatasi secara keseluruhan, ada beberapa kejahatan di dunia perbankan yang mengancam simpanan nasabah, salah satunya adalah *fraud*. Identifikasi kecurangan (*fraud*) dapat dilihat dari hasil audit internal dan eksternal, audit eksternal dilakukan untuk meminta bantuan pihak luar dalam melakukan deteksi kecurangan dalam perusahaan, serta melakukan analisa jika auditor internal mengalami kesulitan. Klasifikasi data yang tidak seimbang merupakan masalah yang krusial pada bidang *machine learning* dan *data mining* sering dijumpai dalam identifikasi kasus *fraud*, ketidakseimbangan data memberikan dampak yang buruk pada hasil klasifikasi dimana kelas minoritas sering disalah klasifikasikan sebagai kelas mayoritas. Oleh karena itu, penulis ingin mengidentifikasi kasus *fraud* pada transaksi perbankan menggunakan metode *Ensemble Stacking*, hasil identifikasi kasus diharapkan mampu meningkatkan performa model dalam penggunaan metode *Ensemble Stacking* untuk mengidentifikasi kasus *fraud*. Pada penelitian ini metode *Ensemble Stacking* terbukti mampu mengatasi masalah ketidakseimbangan data dengan melakukan proses sintesis data terdekat untuk memunculkan data baru hingga data menjadi seimbang. Model *Stacking* secara keseluruhan mampu meningkatkan akurasi dari kelima model yang diuji, dengan presentase peningkatan nilai rata-rata sebesar 14,2% yang mampu meningkatkan kemampuan untuk mengidentifikasi kasus *fraud* pada transaksi kartu kredit perbankan. Walaupun nilai akurasi yang menjadi fokus utama untuk penilaian kinerja model namun parameter *sensitivity* dan *specifity* juga sangat berpengaruh terhadap kinerja model klasifikasi pada kasus *imbalanced data*.

Kata Kunci: *Fraud*, Bank, *Ensemble Stacking*, *Imbalanced Data*, *Machine Learning*, *Orange Data Mining*.

BAB I PENDAHULUAN

1.1 Latar Belakang

Kartu kredit merupakan alternatif uang tunai yang dapat digunakan konsumen untuk ditukar dengan barang dan jasa yang mereka inginkan di tempat-tempat yang dapat menerima pembayaran dengan kartu kredit (*merchant*) (Subagyo, 2005), sehingga memudahkan pelanggan untuk membayar. Di satu sisi, kondisi tersebut memudahkan kebebasan konsumen untuk memperoleh barang dan/atau jasa yang diinginkan dan memilih barang dan/atau jasa yang dibutuhkannya. Selain itu, pengaruh budaya hidup modern dan gaya hidup konsumtif yang mempengaruhi masyarakat juga mempengaruhi perubahan cara masyarakat dalam berbelanja, dari awal penggunaan uang tunai menjadi penggunaan kartu kredit, dalam hal ini kartu kredit. Banyak kartu kredit dipilih untuk keamanan dan kepraktisan transaksi pemegangnya.

Sistem pembayaran elektronik mendominasi era teknologi ini dan telah membangkitkan minat yang besar dari investor, pelaku bisnis, perusahaan jasa pembayaran elektronik, perusahaan kartu kredit. Namun, kemudahan ini juga disertai dengan risiko yang harus diambil ketika menggunakan sistem perdagangan perdagangan seperti itu. Masalah utama yang dihadapi adalah maraknya penyalahgunaan teknologi untuk melakukan kejahatan, dan mengingat transaksi elektronik sering mengandalkan teknologi Internet, kasus kejahatan Internet secara langsung terkait dengan kerentanan transaksi elektronik dan pembayaran melalui Internet. Mengingat transaksi elektronik pada umumnya dilakukan melalui pembayaran kartu kredit sebagai aktivasi atau otentikasi transaksi, tentunya kejahatan teknologi internet juga terkait dengan sistem pembayaran menggunakan kartu kredit, sehingga muncullah apa yang dinamakan dengan tindak penipuan atau penyalahgunaan kartu kredit (*credit card fraud*).

Fraud dapat ditemukan pada berbagai organisasi perusahaan maupun pemerintahan. Secara umum, *fraud* dapat diartikan sebagai sebuah tindakan penipuan

atau manipulasi yang dilakukan seseorang dalam sebuah organisasi dengan tujuan untuk memperoleh keuntungan tertentu yang merugikan pihak lain. Menurut (Karyono ,2013), *fraud* dapat diartikan sebagai kecurangan yang mengandung makna suatu penyimpangan dan perbuatan melanggar hukum (*illegal act*), yang dilakukan dengan sengaja untuk tujuan tertentu misalnya menipu atau memberikan gambaran keliru (*mislead*) kepada pihak-pihak lain, yang dilakukan oleh orang-orang baik dari dalam maupun dari luar organisasi.

Dalam bidang perbankan, *fraud* dapat ditemukan pada berbagai transaksi dan praktik akuntansi. Tindakan *fraud* berakibat pada kerugian keuangan serta penurunan tingkat kepercayaan nasabah terhadap pihak bank. Penanganan tindakan *fraud* dapat dilakukan dengan audit kecurangan (*fraud audit*) yang bertujuan untuk menemukan kecurangan (Soejono, 2000). Namun pada pelaksanaannya, *fraud* sangat sulit dideteksi karena auditor harus memiliki kemampuan khusus mengenai karakteristik *fraud* yang tersaji dalam laporan keuangan. Selain itu, auditor juga membutuhkan waktu yang cukup lama untuk mendeteksi *fraud* terutama dalam mengatasi persoalan dengan frekuensi data yang besar (Watts, 1986).

Terdapat beberapa metode yang bisa dilakukan untuk melakukan pendeteksian terhadap kasus data transaksi pada *credit card fraud*. Salah satu diantaranya yaitu dengan menggunakan metode *Ensemble Stacking*. *Ensemble Stacking* itu sendiri secara umum dapat diartikan sebagai model yang menggunakan algoritma pembelajaran yang berbeda dan kemudian algoritma penggabung dilatih untuk membuat prediksi akhir menggunakan prediksi yang dihasilkan oleh algoritma dasar. Penggabung ini dapat berupa teknik ansambel apa pun.

Berdasarkan uraian diatas, penelitian ini mencoba menentukan apakah pendekatan dengan metode *Ensemble Stacking* mampu mendeteksi transaksi yang *fraud* atau tidak *fraud* dengan baik. Penelitian ini dilakukan dengan membahas bagaimana metode *Ensemble Stacking* dapat digunakan untuk membuat prediksi dengan performa yang lebih baik atau mendeteksi *fraud* dalam transaksi perbankan

secara cepat dan akurat. Penelitian ini akan direpresentasikan menjadi sebuah sistem yang menerapkan metode tersebut. Hasil implementasi sistem akan membuktikan seberapa efektif dan akurat penggunaan metode *Ensemble Stacking* dalam mendeteksi *fraud* dalam sebuah transaksi.

1.2 Rumusan Masalah

Adapun rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut :

1. Bagaimana menerapkan *Ensemble Stacking* dalam meningkatkan akurasi pendeteksian *fraud* dalam transaksi perbankan.
2. Bagaimana dampak penerapan *Ensemble Stacking* terhadap data transaksi yang *imbalance* dan *balance*?

1.3 Ruang Lingkup Penelitian

Ruang lingkup yang dibahas dalam penelitian ini ada dua, yaitu:

1. Implementasi metode *Ensemble Stacking* dilakukan dengan menggunakan aplikasi *Orange Data Mining*.
2. Penelitian ini akan menggunakan data transaksi dari *UCI Machine Learning Repository: Bank Marketing Data Set* tahun 2013.

1.4 Tujuan Penelitian

Adapun tujuan dilaksanakannya penelitian ini adalah sebagai berikut :

1. Untuk menerapkan *Ensemble Stacking* dalam meningkatkan akurasi pendeteksian *fraud* dalam transaksi perbankan.
2. Mengetahui dampak penerapan *Ensemble Stacking* terhadap data transaksi yang *imbalance* dan *balance*.

1.5 Manfaat Penelitian

Secara umum, penelitian ini diharapkan mampu membantu pihak nasabah dalam mengklasifikasi atau mendeteksi *fraud* pada sebuah transaksi. Secara khusus, penelitian ini diharapkan mampu memberikan gambaran bagaimana mendeteksi *fraud* dalam sebuah transaksi dengan menggunakan metode *Ensemble Stacking* untuk

menghasilkan akurasi dalam mendeteksi *fraud* serta kelayakan *credit card*.



BAB II TINJAUAN PUSTAKA

2.1. Kejahatan Perbankan

Perkembangan dunia perbankan saat ini sangatlah pesat. Perkembangan tersebut tentu tidak luput dari kejahatan-kejahatan yang mengintai dunia perbankan. Ada hal serius yang harus dihadapi oleh setiap bank, baik bank konvensional maupun bank syariah. Beberapa jenis perbuatan pelanggaran hukum yang berhubungan dengan kegiatan-kegiatan dalam dunia perbankan yaitu, penghimpunan dana tanpa izin (*Unlicensed Fundraising*), jaminan dokumen palsu (*Fake Document Guarantee*), dan kecurangan kartu kredit (*Credit Card Fraud*). Diantara ketiga macam kejahatan perbankan diatas, kecurangan kartu kredit (*Credit Card Fraud*) ini merupakan salah satu kejahatan perbankan yang sangat di khawatirkan oleh nasabah serta pihak bank itu sendiri.

Kasus *fraud* pada dunia perbankan terus mengintai nasabah, hal itu terjadi di salah satu kasus dana nasabah lenyap di Maybank yang menurut Ketua Dewan Komisioner LPS merupakan sebuah kasus *fraud*. Otoritas Jasa Keuangan meminta PT Bank Maybank Indonesia Tbk (BNII) melakukan investigasi terkait dengan hilangnya dana sebesar Rp 22 miliar di rekening tabungan milik atlet *e-sport* Winda Lunardi dan ibunya sebesar Rp 20 miliar di kantor cabang Maybank ;Cipular (*Dana Nasabah Hilang di Maybank, Bos LPS: Itu Kasus Fraud!*, n.d). Kejadian tersebut sangat berbahaya bagi industri perbankan, khususnya kepercayaan nasabah dalam menyimpan uangnya di perbankan.

Bank Indonesia pernah mengeluarkan regulasi melalui Surat Edaran Bank Indonesia (SEBI) No.13/28/DPNP, lewat surat tersebut *fraud* didefinisikan sebagai tindakan penyimpangan yang sengaja dilakukan untuk mengelabui, menipu, atau memanipulasi bank atau nasabah yang dilakukan di lingkungan bank dan/atau menggunakan sarana bank yang mengakibatkan nasabah atau pihak lain menderita kerugian dan pelaku *fraud* mendapatkan keuntungan keuangan baik secara langsung maupun tidak langsung (Narendra, 2021).

Faktor penyebab terjadinya tindakan *fraud* (kecurangan) seringkali disebabkan oleh beberapa alasan baik yang berasal dari dalam diri maupun dari luar dirinya. Secara umum, ada tiga hal yang mendorong terjadinya *fraud* yakni dorongan (*pressure*), peluang (*opportunity*) serta pembenaran (*rationalization*) atas tindakan yang dilakukan (Soekardi, 2012). Dorongan (*pressure*) merupakan dorongan yang timbul dari keinginan pelaku untuk memperoleh hidup lebih baik. Keinginan ini seringkali dipengaruhi oleh keadaan ekonomi yang buruk serta dorongan lingkungan untuk memiliki gaya hidup yang mewah. Peluang (*opportunity*) merupakan faktor terjadinya *fraud* yang dipengaruhi oleh lemahnya pengelolaan manajemen sehingga memicu seseorang berani untuk melakukan tindakan *fraud*. Rasionalisasi (*rationalization*) merupakan alasan rasional yang dilakukan oleh pelaku tindakan *fraud* untuk menutupi tindakan yang dilakukan. Jenis-jenis *fraud* dapat dikelompokkan menjadi 3 yakni :

1. Kecurangan pegawai (*Employee Fraud*) yakni kecurangan yang dilakukan oleh pegawai dalam suatu organisasi kerja.
2. Kecurangan manajemen (*Management Fraud*) yakni kecurangan yang dilakukan oleh pihak manajemen dengan menggunakan laporan keuangan atau transaksi keuangan sebagai *fraud*.
3. Kecurangan komputer (*Computer Fraud*) yakni kecurangan yang dilakukan dalam pencatatan komputer yang meliputi pencatatan operasional atau pembukuan dalam suatu perusahaan.

2.2. Fraud

Bank Indonesia pernah mengeluarkan regulasi melalui Surat Edaran Bank Indonesia (SEBI) No.13/28/DPNP, lewat surat tersebut *fraud* didefinisikan sebagai tindakan penyimpangan yang sengaja dilakukan untuk mengelabui, menipu, atau memanipulasi bank atau nasabah yang dilakukan di lingkungan bank dan/atau menggunakan sarana bank yang mengakibatkan nasabah atau pihak lain menderita kerugian dan pelaku *fraud* mendapatkan keuntungan keuangan baik secara langsung

maupun tidak langsung (Narendra, 2021).

2.3. Sejarah *Credit Card Fraud* di Indonesia

Pertama kali kartu kredit yang masuk ke Indonesia terjadi sekitar tahun 1980-an. Diperkenalkan oleh Bank Duta yang pada saat itu menjalin kerjasama dengan VISA dan *MasterCard International*. Bank Duta merupakan bank yang pertama kali menerbitkan dan memasarkan kartu kredit di Indonesia. Dulu kartu kredit terbitan bank Duta khusus bagi nasabahnya sendiri dan tidak bebas bagi kalangan umum. Target market dari bisnis kartu kredit ini hanya ditujukan kepada kalangan orang kaya, pengusaha, pejabat, dan orang-orang kelas atas lain.

Pemakaian kartu kredit sudah cukup meluas. Bahkan seringkali seseorang memegang beberapa kartu kredit sekaligus. Yang namanya *Visa*, *Master Card*, *American Express* adalah diantara nama-nama kartu kredit yang seringkali terdapat dalam dompet seseorang. Seiring dengan pesatnya penggunaan kartu kredit kartu kredit tersebut, penyalahgunaannya seperti yang disebutkan diatas juga banyak terjadi. Disamping itu, ternyata juga serangkali terjadi bahwa para pihak yang terlibat dalam penggunaan /penerbitan/pemakaian kartu kredit tidak selamanya melaksanakan prestasinya seperti yang diperjanjikan, baik karena kesengajaan, kesilapan maupun karena seribu satu alasan lainnya. Karena itu, kehadiran sektor hukum yang adil, tegas dan terprediksi untuk menata penggunaan kartu kredit tentu merupakan kebutuhan dunia bisnis yang nyata dalam praktek. Pasalnya, karena tentunya bank dan para pihak yang terlibat dalam hubungan dengan kartu kredit ini ingin agar kedudukannya terlindungi secara hukum, dengan hak dan kewajibannya yang beralasan dan transparan.

Seiring dengan berkembangnya perdagangan secara online, juga terbuka peluang munculnya tindakan-tindakan anti-sosial dan perilaku kejahatan yang sebelumnya dianggap tidak mungkin terjadi. Kejahatan yang lahir sebagai dampak negatif dari perkembangan internet ini sering disebut *cyber crime* (www.interpol.go.id). Menurut Indradi (2006), *credit card fraud* atau *carding* adalah

penipuan kartu kredit bila pelaku mengetahui nomor kartu kredit seseorang yang masih berlaku, maka pelaku dapat membeli barang secara *on-line* yang tagihannya dialamatkan pada pemilik asli kartu kredit tersebut, sedangkan pelakunya dinamakan *carder*.

Kejahatan menggunakan perangkat komputer dan jaringan internet (*cyber crime*) di Indonesia, termasuk tertinggi di dunia. Untuk kasus pembobolan kartu kredit (*credit card fraud/carding*) saja, berdasarkan hasil riset dari *Clear Commerce Inc*, sebuah perusahaan teknologi informasi (TI) yang berbasis di Texas, AS, pada tahun 2005, Indonesia berada pada posisi ke-2 teratas sebagai negara asal *carder* terbanyak di dunia setelah Ukraina (Donny, 2002). Hasil riset tersebut mengentakkan kesan bahwa Indonesia tidak berbuat banyak untuk melakukan perubahan sepanjang 2002 hingga 2003 lalu, Ketika posisinya “baru” pada urutan kedua setelah Ukraina. Padahal, saat itu citra Internet Indonesia sudah dijatuhkan beramai-ramai oleh media massa luar negeri semisal majalah *Time Business Week*, yang turut mengutip hasil riset *ClearCommerce* pada saat itu. Tak cukup hanya itu, hingga saat ini nyaris semua pengguna situs lelang kenamaan *eBay.com* sangat “takut” apabila bertransaksi dengan seseorang yang meminta pengiriman barangnya ditujukan ke suatu alamat di Indonesia. Bagi mereka, alamat di Indonesia sudah masuk dalam catatan *black-list* mereka. Minimnya pihak internasional yang mau melayani transaksi kartu kredit *online (payment gateway)* bagi pemilik *merchant* ataupun *consumer* dari Indonesia, tentu akan semakin mengucilkan Indonesia (Donny, 2004)

Salah satu kasus pencurian identitas kartu kredit pada Oktober 2009, Harry Agus Tumewu alias Hendro Danutirto dan Henky Otto Tumewu alias Alexander Wenas Oktober 2009, mengajak Henky Otto Tumewu alias Alexander Wenas bekerja sama dengan Harry Agus Tumewu dalam melakukan pencurian identitas melalui mesin EDC (*Electronic Data Capture*) yaitu mesin yang digunakan untuk transaksi dengan kartu kredit dan kartu *debit* jika customer berbelanja atau bertransaksi tanpa menggunakan uang *cash*. Lalu identitas dalam kartu kredit tersebut kemudian dimasukkan dalam sistem komputer, seolah-olah transaksi terjadi pada *merchant*

yang memiliki mesin EDC. Jika ada pembayaran dari pihak Bank Permata untuk transaksi yang sebenarnya tidak ada, hasilnya dibagi dua dengan pembagian masing-masing 50% untuk *merchant* dan terdakwa bersama teman-temannya sebagai coordinator wilayah serta 50% untuk Henky Otto Tumewu alias Alexander Wenas bersama dengan Henry Martinus Tumewu alias Denny Korompis, Harry Agus Tumewu alias Hendro Danutirto dan Miko (keduanya sebelumnya pernah bekerja di bagian Kartu Kredit Bank Permata dan mengetahui sistem pelaksanaan transaksi melalui kartu kredit Bank Permata).

2.4. Asal Mula Credit Card Dipalsukan

Penggunaan Kartu Kredit di Indonesia lebih rentan mengalami pemalsuan. Hal ini disebabkan mekanisme penggunaan kartu kredit terlebih dahulu menggunakan pendanaan dari bank. Risiko pemalsuan lebih kecil terjadi di kartu debit, sebab transaksi kartu debit lebih menggunakan dana nasabah itu sendiri. Sementara kartu kredit transaksi awalnya menggunakan dana bank. Baru ditagihkan kepada konsumen pemegang kartu. Ini membuat kartu kredit lebih banyak jadi sasaran pemalsuan (Zaid Azla, 2014).

Modus pemalsuan adalah menggunakan alat yang disebut *skimmer*. Biasanya alat ini dipasang pada mesin *Electronic Data Captured* (EDC) yang digunakan di berbagai pusat perbelanjaan. Begitu digesek oleh kasir pada saat transaksi pembayaran oleh konsumen, secara otomatis data nasabah pemegang kartu ter-*copy*. Data yang digunakan ini kemudian digandakan untuk membuat kartu kredit palsu. Saat ini migrasi kartu debit magnetic ke *chip* diyakini akan meminimalkan pemalsuan kartu kredit. Sebab teknologi chip menggunakan pengkodean berlapis sehingga kerahasiaan data lebih sulit untuk dibobol. Proses migrasi kartu magnetik ke *chip* hanya diketahui oleh pihak manajemen bank.

2.5. Machine Learning

2.5.1. Definisi Machine Learning

Machine Learning merupakan salah satu cabang dari ilmu Kecerdasan Buatan,

khususnya yang mempelajari tentang bagaimana komputer mampu belajar dari data untuk meningkatkan kecerdasannya. Berikut ini adalah pengertian *Machine Learning* menurut beberapa ahli :

- a) Menurut Arthur (1959): Kemampuan komputer untuk melakukan pembelajaran tanpa harus menjelaskan atau terprogram secara eksplisit kepada komputer.
- b) Menurut Tom Mitchel (1997): Komputer yang memiliki kemampuan melakukan belajar dari pengalaman terhadap tugas-tugasnya dan mengalami peningkatan kinerja.
- c) Menurut Budiharto (2016): Tipe dari kecerdasan buatan yang menyediakan komputer dengan kemampuan untuk belajar dari data, tanpa secara eksplisit harus mengikuti instruksi terprogram.

2.5.2. Kategori *Machine Learning*

Pada dasarnya, *Machine Learning* diklasifikasikan menjadi tiga kategori utama, yaitu:

1) *Supervised Learning*

Supervised Learning, juga dikenal sebagai pembelajaran mesin yang diawasi, adalah subkategori *Machine Learning* dan kecerdasan buatan . Ini ditentukan oleh penggunaan kumpulan data berlabel untuk melatih algoritma yang mengklasifikasikan data atau memprediksi hasil secara akurat. Saat data input dimasukkan ke dalam model, ia menyesuaikan bobotnya hingga model dipasang dengan tepat, yang terjadi sebagai bagian dari proses validasi silang. Pembelajaran yang diawasi membantu organisasi memecahkan 7 berbagai masalah dunia nyata dalam skala besar, seperti mengklasifikasikan spam dalam folder terpisah dari kotak masuk.

Supervised Learning menggunakan satu set pelatihan untuk mengajarkan model untuk menghasilkan *output* yang diinginkan. *Dataset* pelatihan ini mencakup *input* dan *output* yang benar, yang memungkinkan model untuk belajar dari waktu ke waktu. Algoritma mengukur akurasi melalui fungsi kerugian, menyesuaikan

hingga kesalahan cukup diminimalkan (Cloud, 2020a).

Supervised learning banyak digunakan dalam memprediksi pola dimana pola tersebut sudah ada contoh data yang lengkap, jadi pola yang terbentuk adalah hasil pembelajaran data lengkap tersebut. Tentunya jika kita memasukan data baru, setelah kita melakukan ETL (Extract Transform Load) maka kita mendapat info feature feature dari *sample* baru tersebut. Kemudian dari *feature-feature* tersebut di *compare* dengan *pattern clasification* dari model yang didapat dari *labeled data*. Setiap label akan di *compare* sampai selesai, dan yang memiliki *percentage* lebih banyak akan diambil sebagai prediksi akhir (Ciputra Universitas, 2019).

Pembelajaran yang diawasi dapat dipisahkan menjadi dua jenis masalah saat penggalian data, klasifikasi dan regresi:

- a) Klasifikasi menggunakan algoritma untuk secara akurat menetapkan data uji ke dalam kategori tertentu. Ini mengenali entitas tertentu dalam kumpulan data dan mencoba menarik beberapa kesimpulan tentang bagaimana entitas tersebut harus diberi label atau didefinisikan. Algoritma klasifikasi yang umum adalah pengklasifikasi *linier*, *Support Vector Machine (SVM)*, *decision tree*, *k-nearest neighbor*, dan *random forest*.
- b) Regresi digunakan untuk memahami hubungan antara variabel terikat dan variabel bebas. Ini biasanya digunakan untuk membuat proyeksi, seperti untuk pendapatan penjualan untuk bisnis tertentu. *Regresi linier*, *regresi logistik*, dan *regresi polinomial* adalah algoritma *regresi* yang populer.

2) *Unsupervised Learning*

Unsupervised learning, juga dikenal sebagai pembelajaran mesin tanpa pengawasan, menggunakan algoritma pembelajaran mesin untuk menganalisis dan mengelompokkan kumpulan data yang tidak berlabel. Algoritma ini menemukan pola tersembunyi atau pengelompokan data tanpa perlu campur tangan manusia. Kemampuannya untuk menemukan persamaan dan perbedaan informasi menjadikannya solusi *ideal* untuk analisis data eksplorasi, strategi penjualan silang,

segmentasi pelanggan, dan pengenalan gambar (Cloud, 2020b).

Unsupervised learning adalah salah satu tipe algoritma machine learning yang digunakan untuk menarik kesimpulan dari datasets yang terdiri dari *input data* dan *labeled response*. Metode *unsupervised learning* yang paling umum adalah analisa *cluster*, yang digunakan pada analisa data untuk mencari pola-pola tersembunyi atau pengelompokan dalam data (Nurhayati, 2019).

3) *Semi-supervised Learning*

Metode *Semi-supervised learning* merupakan sebuah solusi untuk memilih metode *Machine Learning* dibandingkan dua metode yang telah disebutkan sebelumnya. *Semi-supervised learning* akan menggunakan data-data yang telah ditraining dengan data yang mempunyai label tertentu dan tidak menggunakan label. Hal tersebut sangat berguna ketika terjadi masalah pada aktivitas mengekstrak fitur yang relevan dan masalah pada *volume* data yang besar. *Semi-supervised learning* sangat ideal untuk citra medis, di mana sejumlah kecil data pelatihan dapat menghasilkan peningkatan akurasi yang signifikan. Misalnya, ahli radiologi dapat memberi label pada sebagian kecil *CT scan* untuk tumor atau penyakit sehingga mesin dapat lebih akurat memprediksi pasien mana yang mungkin memerlukan perhatian medis lebih (Julianna, 2021).

2.6. *Data Mining*

Data mining merupakan proses yang mempekerjakan satu atau lebih teknik pembelajaran komputer (*machine learning*) untuk menganalisis dan mengekstraksi pengetahuan (*knowledge*) secara otomatis. *Data mining* dapat disebut juga sebagai suatu pembelajaran berbasis induksi (*induction-based learning*) yang merupakan sebuah proses pembentukan berbagai definisi dari berbagai konsep yang akan dipelajari. *Data mining* berisi pencarian trend atau pola yang diinginkan dalam database yang besar untuk membantu pengambilan keputusan di waktu yang akan datang. Pola-pola ini akan dikenali oleh perangkat tertentu yang dapat memberikan suatu analisis yang berguna dan berwawasan yang kemudian dapat dipelajari.

Konsep dasar *data mining* mengambil ide dari konsep *machine learning*, pengenalan pola, statistik, dan *database systems* (Hermawati, 2013).

Data mining terdapat istilah *Knowledge Discovery in Database* (KDD) yang digunakan untuk menggali informasi tersembunyi dalam suatu basis data. Dalam proses KDD terdapat beberapa proses yakni :

1. *Data Cleaning*,

Data cleaning merupakan tahap awal dalam penambangan data yang bertujuan untuk menghapus data yang *noise* yang terdapat dalam data

2. *Data Integration*,

Data integration merupakan tahap penggabungan data dari sumber yang berbeda.

3. *Data Selection*

Data selection merupakan proses penyeleksian data yang relevan dari sebuah database. Pada tahap ini, setiap fitur di cek apakah relevan atau tidak untuk dilakukan penambangan data.

4. *Data Transformation*

Data transformation merupakan tahap dimana data mengalami transformasi ke dalam bentuk yang sesuai. Tujuan dari transformasi data ini adalah agar bentuk yang dihasilkan lebih mudah diolah dan dipahami.

5. *Data Mining*

Proses *data mining* merupakan tahap penambangan/pengolahan data diterapkan dan diaplikasikan dengan menggunakan algoritma khusus yang telah dipilih untuk mengekstrak pola.

6. *Pattern Evaluation*

Pada tahap ini dilakukan proses identifikasi pola yang menarik. Pola tersebut akan direpresentasikan dalam bentuk pengetahuan berdasarkan beberapa pengukuran penting.

7. *Knowledge Presentation*

Tahap ini merupakan tahap akhir dalam *data mining* dimana pengetahuan yang diperoleh direpresentasikan kepada pengguna.

Dalam *Data mining*, terdapat himpunan data (*dataset*) yang merupakan kumpulan data dari objek dan fiturnya. Fitur merupakan sifat atau karakteristik dari suatu objek. Misalnya warna mata seseorang, suhu, dan sebagainya. Fitur juga dikenal sebagai variabel, *field*, karakteristik atau fitur. Kumpulan dari fitur menggambarkan sebuah objek. Objek juga disebut dengan *record*, titik, kasus, *sample*, entitas atau *instance*.

Berdasarkan tugas yang dapat dilakukan, *data mining* dibagi menjadi beberapa kelompok, yaitu (Larose, 2005) :

1. Deskripsi

Deskripsi dari pola dan kecenderungan sering memberikan kemungkinan penjelasan untuk suatu pola atau kecenderungan.

2. Estimasi

Estimasi hampir sama dengan klasifikasi, kecuali variabel target estimasi lebih ke arah *numerik* daripada ke arah kategori. Pembangunan model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya, estimasi nilai dari variabel target dibuat berdasarkan nilai dari variabel prediksi.

3. Prediksi

Prediksi hampir sama dengan klasifikasi dan estimasi, kecuali bahwa dalam prediksi nilai dari hasil akan ada di masa mendatang.

4. Klasifikasi

Dalam klasifikasi, terdapat target variabel kategori. Sebagai contoh, penggolongan pendapatan dapat dipisahkan dalam tiga kategori, yaitu pendapatan tinggi, pendapatan sedang, dan pendapatan rendah.

5. Pengklusteran

Pengklusteran merupakan pengelompokan *record*, pengamatan, atau memperhatikan dan membentuk kelas dari berbagai objek yang memiliki kemiripan. Klaster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster

lain.

6. Asosiasi

Tugas asosiasi dalam data mining adalah menemukan fitur yang muncul dalam satu waktu. Dalam dunia bisnis, asosiasi umumnya disebut dengan analisis keranjang belanja.

Sebuah sistem yang baik yang dibangun dengan menggunakan prinsip *data mining* menunjukkan kualitas data yang baik. Untuk melihat baik buruknya kualitas data perlu ditinjau kembali permasalahan kualitas data yang ditinjau dari aspek pengukuran data dan pengumpulan data. Kesalahan pengukuran (*Measurement error*) mengacu pada permasalahan hasil dari proses pengukuran. Masalah yang umumnya terjadi adalah nilai yang dicatat berbeda dari nilai sebenarnya. Pada fitur kontinyu, beda *numerik* dari hasil pengukuran dengan nilai sebenarnya disebut dengan *error*. Dalam hal ini, jenis kesalahan pengukuran dibagi menjadi beberapa jenis yakni:

- 1) *Noise*, merupakan komponen *random* dari suatu *error* pengukuran,
- 2) *Bias*, merupakan suatu variasi pengukuran dari kuantitas yang sedang diukur dengan pengurangan antara *mean* dan nilai kuantitas yang diketahui,
- 3) *Precision*, yakni kedekatan dari pengukuran berulang (dari kuantitas yang sama) satu dengan yang lainnya. Biasanya diukur dengan standar *deviasi*,
- 4) *Accuracy*, merupakan kedekatan pengukuran terhadap nilai sebenarnya dari kuantitas yang diukur.

2.7. Ensemble Learning

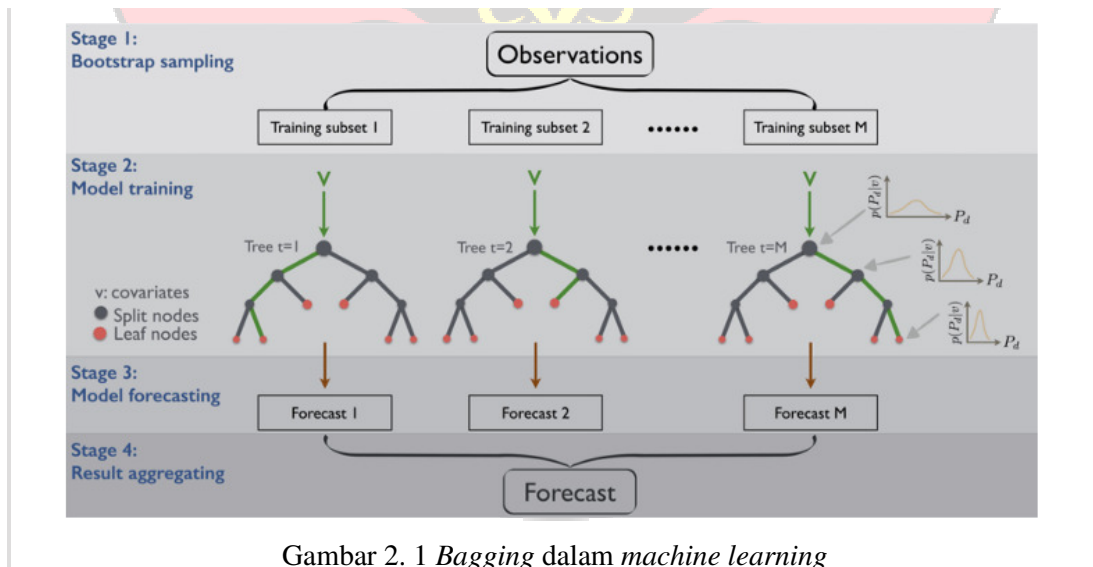
Ensemble Learning atau *ensemble method* adalah metode di mana beberapa algoritma pembelajaran digunakan secara bersamaan, lalu dikombinasikan untuk mendapatkan hasil pemodelan yang lebih akurat. Dalam proses metode *ensemble*, model-model individu ini biasa disebut anggota *ensemble* (*ensemble member*). Tujuan melakukan *Ensemble Learning* adalah untuk mendapatkan model dengan akurasi yang lebih tinggi daripada jika hanya menggunakan satu algoritma/ model

saja. *Ensemble Learning* adalah cara yang ampuh untuk meningkatkan kinerja model (Saddam, 2021).

Cara sebuah algoritma mempelajari data dengan menggunakan kombinasi dari beberapa algoritma atau model untuk mendapatkan output dengan akurasi yang lebih tinggi jika dibandingkan dengan menggunakan satu algoritma atau model saja. Cara melakukan *ensemble learning* adalah dengan proses *bagging*, *boosting*, dan *stacking*.

a) *Bagging*

Bagging, juga dikenal sebagai *bootstrap aggregating*, adalah proses dengan menggunakan beberapa model dari algoritma yang sama, dan melatih setiap model pada sampel berbeda dari *dataset* yang sama. Prediksi yang dibuat oleh setiap model kemudian digabungkan menggunakan statistik sederhana, seperti *voting* atau rata-rata. Dalam proses ini, teknik *bootstrap* digunakan untuk membuat beberapa set sampel dari satu *dataset* yang sama. Sedangkan dalam prosesnya, *bagging* hampir selalu menggunakan *unpruned decision tree*.



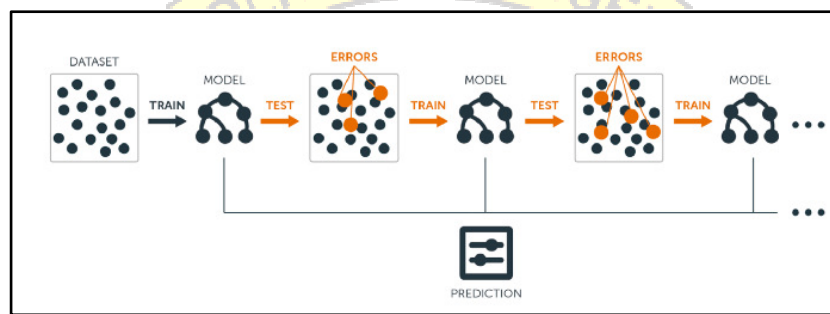
Gambar 2. 1 *Bagging* dalam *machine learning*

Sumber: <https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc68705a2>

b) *Boosting*

Boosting adalah variasi *bagging*, di mana setiap model individu dibangun secara berurutan, mengulangi yang sebelumnya. Dalam prosesnya, setiap titik data yang salah diklasifikasikan oleh model sebelumnya ditekankan dalam model berikutnya. Hal ini dilakukan untuk meningkatkan akurasi model secara keseluruhan.

Berikut adalah diagram untuk lebih memahami prosesnya:



Gambar 2. 2 *Boosting* dalam *machine learning*

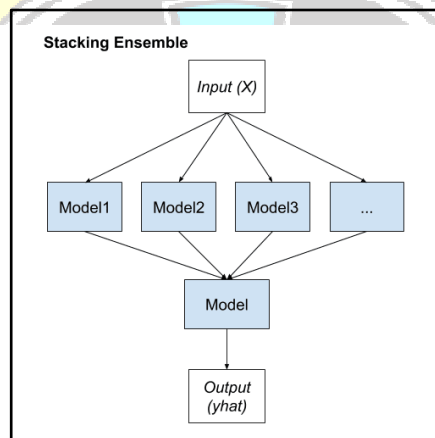
Sumber: <https://towardsdatascience.com/simple-guide-for-ensemble-learning-methods-d87cc68705a2>

Setelah model pertama dibangun, titik yang diklasifikasikan/diprediksi secara salah, diambil sebagai tambahan pada sampel *bootstrap* kedua untuk melatih model kedua. Kemudian, model *ensemble* (model 1 dan 2) digunakan terhadap dataset uji. Proses ini kemudian dilanjutkan sampai mendapatkan model terbaik.

c) *Stacking*

Stacking adalah algoritma pembelajaran mesin yang menggabungkan berbagai model pembelajaran mesin untuk membuat prediksi yang terbaik. *Ensemble stacking* menggunakan algoritma *meta learning* untuk mempelajari cara terbaik menggabungkan prediksi dari beberapa algoritma pembelajaran mesin (Brownlee, 2020). Penggunaan teknik *stacking* dalam *ensemble learning* memanfaatkan kemampuan berbagai model yang menghasilkan performa baik pada klasifikasi atau regresi dan dapat membuat prediksi yang memiliki kinerja lebih baik dibandingkan

dengan model tunggal apapun dalam *ensemble* tersebut (Brownlee, 2020). Penumpukan mengeksploitasi fakta bahwa algoritma pembelajaran mesin dapat memiliki keterampilan yang berbeda untuk masalah yang sama. Oleh karena itu, alih-alih mempercayai satu model untuk membuat prediksi, kelebihan *Stacking* itu sendiri memungkinkan kita menggunakan model yang berbeda untuk membangun satu model yang kuat berdasarkan semua model dasar individu melibatkan pemasangan banyak jenis model yang berbeda pada data yang sama dan menggunakan model lain untuk mempelajari cara terbaik menggabungkan prediksi.



Gambar 2. 3 *Stacking* dalam *machine learning*

Sumber: <https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/>

Beberapa elemen kunci dalam proses *stacking* adalah:

- Set data pelatihan tidak berubah.
- Algoritma ML yang berbeda untuk setiap anggota *ensemble* (*ensemble member*).
- Model ML digunakan untuk mempelajari cara terbaik menggabungkan model prediksi.

2.8. Klasifikasi

Klasifikasi adalah teknik *supervised learning* dari *data mining*, yang terdiri dari sekumpulan kelas yang telah ditentukan sebelumnya dan atas dasar kelas yang telah ditentukan ini, objek baru diklasifikasikan. Teknik klasifikasi akan

mengklasifikasikan data berdasarkan *data training* dan menghasilkan pengelompokan atau model dan menggunakannya dalam mengklasifikasikan data baru (Gupta et al., 2015). Hasil penelitian (Johra, 2018) menyatakan dalam studi yang dilakukannya tentang efek dari *training data* pada model klasifikasi menunjukkan bahwa *error* terkecil didapatkan pada jumlah training data 79,09 sampai 86,19%.

Salah satu langkah yang perlu diperhatikan saat melakukan klasifikasi terlebih dahulu dilakukan tahap seleksi fitur. *Information gain* merupakan salah satu metode seleksi fitur yang banyak dipakai oleh peneliti untuk menentukan batas dari kepentingan sebuah atribut (Kashanian & Dabaghi, 2017). Nilai *information gain* diperoleh dari nilai *entropy* sebelum pemisahan dikurangi dengan nilai *entropy* setelah pemisahan. Pengukuran nilai ini hanya digunakan sebagai tahap awal untuk penentuan atribut yang nantinya akan digunakan atau dibuang. Atribut yang memenuhi kriteria pembobotan yang nantinya akan digunakan dalam proses klasifikasi sebuah algoritma. Pemilihan fitur dengan *information gain* dilakukan dalam 3 tahapan, yaitu:

1. Menghitung nilai *information gain* untuk setiap atribut dalam *dataset original*.
2. Tentukan batas (*threshold*) yang diinginkan. Hal ini akan memungkinkan atribut yang berbobot sama dengan batas atau lebih besar akan dipertahankan serta membuang atribut yang berada dibawah batas.
3. Dataset diperbaiki dengan pengurangan atribut.

Klasifikasi merupakan pengelompokan fitur ke dalam kelas yang sesuai. Vektor fitur pelatihan tersedia dan telah diketahui kelas-kelasnya, kemudian vektor fitur pelatihan tersebut dimanfaatkan untuk merancang pemilah. Pengenalan pola ini disebut terbimbing, *supervised* (Dillak et al., 2012).

Penentuan atau pengelompokan data dengan klasifikasi dibagi menjadi beberapa jenis model yang dinamakan sebagai keputusan berbasis pengklasifikasi (Gupta et al., 2015). Beberapa model yang dimaksud adalah:

2.7.1 Random Forest (RF)

Random Forest adalah sebuah metode *ensemble learning* yang pertama kali diperkenalkan oleh Breiman pada tahun 2001 yang merupakan kombinasi dari pohon klasifikasi sedemikian rupa sehingga setiap pohon bergantung pada nilai acak vektor sampel secara mandiri dan dengan distribusi yang sama untuk semua pohon di hutan (Xiao et al., 2012).

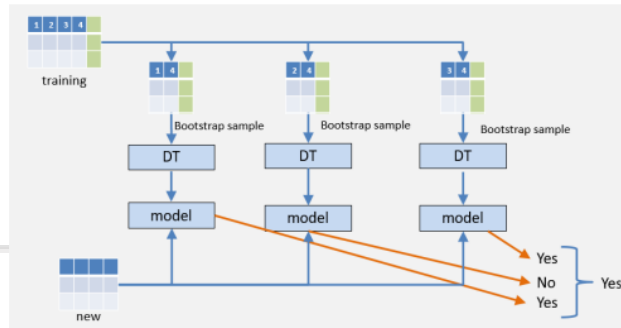
Random Forest telah banyak digunakan baik untuk klasifikasi dan regresi karena kinerjanya yang unggul dan strukturnya yang sederhana (Lin et al., 2017). *Random Forest* biasa digunakan untuk menyelesaikan masalah yang berhubungan dengan klasifikasi, regresi, dan sebagainya. Ada dua hal yang membuat algoritma ini disebut *random*, yaitu:

1. Setiap pohon tumbuh pada sampel bootstrap yang berbeda diambil dari data latih secara acak.
2. Dalam setiap node split selama pembentukan decision tree, sebagian sampel dari variabel dipilih dari kumpulan data yang asli dan kemudian yang terbaik akan digunakan dalam node tersebut.

Algoritma ini berupa kombinasi dari beberapa *tree predictors* atau bisa disebut *decision trees* dimana setiap *tree* bergantung pada nilai *random vector* yang dijadikan sampel secara bebas dan merata pada semua *tree* dalam *forest* tersebut. Hasil prediksi dari *Random Forest* didapatkan melalui hasil terbanyak dari setiap *individual decision tree* (voting untuk klasifikasi dan rata-rata untuk regresi). Untuk RF yang terdiri dari N *trees* dirumuskan sebagai:

$$l(y) = \operatorname{argmax}_c \left(\sum_{n=1}^N I_{h_n}(y) = c \right)$$

dimana I adalah fungsi indikator dan h_n adalah tree ke- n dari *Random Forest* (Liparas et al., 2014).



Gambar 2. 4 Model *Random Forest*

Sumber: https://medium.com/?/@harshdeepsingh_35448/

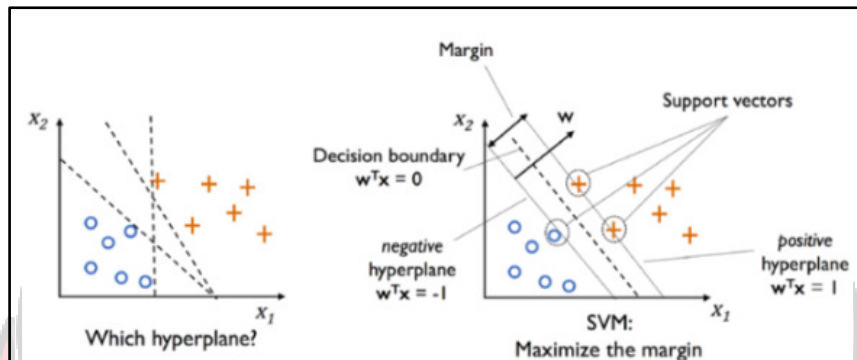
Random Forest memiliki mekanisme internal yang menyediakan estimasi dari *generalization error*-nya sendiri yang disebut *out-of-bag (OOB) error estimate*. Dalam pembentukan *tree* hanya $\frac{2}{3}$ dari data asli yang digunakan dalam pengambilan sampel *bootstrap*. Sedangkan $\frac{1}{3}$ sisanya diklasifikasikan oleh *tree* yang terbentuk dan digunakan untuk menguji performanya. *OOB error estimation* adalah rata-rata dari kesalahan prediksi untuk setiap kasus *training* yang menggunakan *tree* yang tidak mengikutsertakan dalam sampel *bootstrap*-nya. Kemudian, saat *Random Forest* dibuat, semua *training cases* menyusuri setiap pohon dan matriks kedekatan setiap kasus dihitung berdasarkan pasangan kasus yang sampai di terminal *node* yang sama (Liparas et al., 2014).

2.7.2 *Support Vector Machine*

SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization (SRM)* dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*. *Support Vector Machine (SVM)* pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition*. Sebagai salah satu metode *pattern recognition*, usia SVM terbilang masih relatif muda. Walaupun demikian, evaluasi kemampuannya dalam berbagai aplikasinya menempatkannya sebagai *state of the art* dalam *pattern recognition* (Cortes et al., 1995).

SVM digunakan untuk mencari *hyperplane* terbaik dengan memaksimalkan

jarak antar kelas. *Hyperplane* adalah sebuah fungsi yang dapat digunakan untuk pemisah antar kelas. Dalam 2-D fungsi yang digunakan untuk klasifikasi antar kelas disebut sebagai *line whereas*, fungsi yang digunakan untuk klasifikasi antar kelas dalam 3-D disebut *plane similarly*, sedangkan fungsi yang digunakan untuk klasifikasi di dalam ruang kelas dimensi yang lebih tinggi disebut *hyperplane*.



Gambar 2. 5 Model SVM

Sumber: <https://Penjelasan Sederhana tentang Apa Itu SVM? | by Samsudiney | Medium>

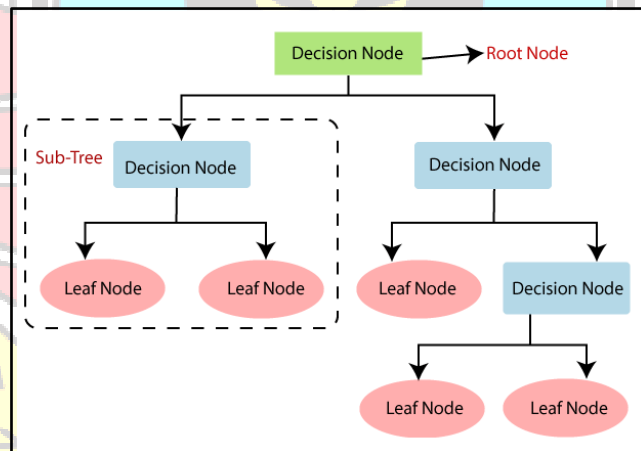
Support Vector Machine (SVM) saat ini mendapat perhatian besar dari komunitas *machine learning*. Ini menawarkan strategi pembelajaran baru untuk mendapatkan *hyperplane* yang optimal dan menghasilkan generalisasi yang tinggi terhadap pola yang tidak terlihat. Ini juga membuka kemungkinan untuk memasukkan pengetahuan sebelumnya dari domain dengan merancang fungsi kernel tertentu untuk mengukur tingkat kesamaan data. Fitur-fitur ini menjelaskan mengapa dalam beberapa penelitian, SVM digeneralisasi lebih baik daripada metode klasifikasi konvensional termasuk jaringan saraf yang dilatih oleh algoritma *backpropagation* (Nugroho, 2008).

2.7.3 Decision Tree (Tree)

Decision Tree adalah algoritma yang paling banyak digunakan untuk masalah klasifikasi. Sebuah *Decision Tree* terdiri dari beberapa simpul yaitu *tree's root*, *internal nod* dan *leafs*. Konsep entropi digunakan untuk penentuan pada atribut mana sebuah pohon akan terbagi (*split*). Dalam pohon keputusan, setiap simpul

internal membagi ruang menjadi dua atau lebih sesuai dengan fungsi diskrit dari input atribut nilai. Dalam kasus yang paling sederhana dan paling sering, setiap tes menganggap sebagai atribut tunggal, sehingga ruang dipartisi kosong disesuaikan dengan nilai atribut. Klasifikasi menggunakan *Decision Tree* dilakukan oleh routing dari simpul akar sampai tiba di simpul daun. Algoritma *decision tree* antara lain ID3, C4.5, C5.0, and CART.

Setelah *Decision Tree* dibangun, setiap kasus ditugaskan untuk setiap j daun mana $j = 1 \dots N$ dengan berat w_j adalah 0 atau 1. jika setiap atribut uji dikenal untuk i . Pada awalnya, seluruh populasi membentuk akar pohon yang berbeda. Untuk menghasilkan berbagai cabang pohon, fitur yang dipilih dibedakan dari karakteristik populasi. Fitur ini disebut tes yang menghasilkan *node* anak baru. Kekuatan diskriminasi dapat diukur dengan *Shannon entropi gain G*. Semakin tinggi *entropy* sebuah sampel, semakin tidak murni sampel tersebut (Oktafia & Pardede, 2008).



Gambar 2. 6 Model *Decision Tree*

Sumber: <https://Klasifikasi Decision Tree dengan R> | by Halima Tusyakdiah | Medium

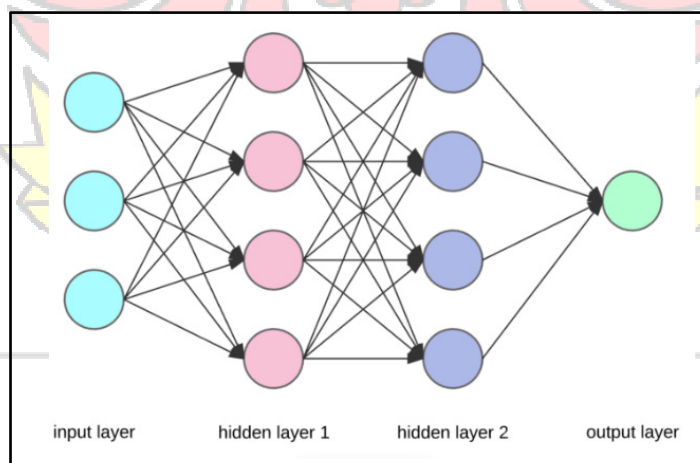
Decision tree adalah sebuah struktur pohon, dimana setiap *node* pohon merepresentasikan atribut yang telah diuji setiap cabang merupakan suatu pembagian hasil uji dan *node* daun (*leaf*) merepresentasikan kelompok kelas tertentu. Level *node* teratas dari sebuah *decision tree* adalah *node* akar (*root*) biasanya berupa atribut yang

paling memiliki pengaruh terbesar pada suatu kelas tertentu. Pada proses mengklasifikasi data yang tidak diketahui. Struktur *Decision Tree* nilai atribut akan diuji dengan cara melacak jalur dari node akar (*root*) sampai *node* akhir (daun) dan kemudian akan diprediksi kelas yang dimiliki oleh suatu data baru tertentu.

Metode *Decision Tree* telah banyak digunakan dalam masalah klasifikasi, diantaranya adalah untuk memprediksi resiko sebelum kehamilan (Nai-Arun & Moungrmai, 2015). Penelitian tersebut menjelaskan bahwa presisinya mencapai 90%. Namun ada gambar yang sulit diklasifikasikan sehingga presisinya berkurang menjadi 30% - 40%. Data bertipe *numerik* juga telah dilakukan objek loyalitas pelanggan dan mencapai akurasi 97.5% (Wibawa et al., 2018).

2.7.4 Neural Network (NN)

Neural Network (Jaringan Saraf Tiruan) adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel saraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Jaringan Saraf Tiruan, seperti manusia, belajar dari suatu contoh. Jaringan Saraf Tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran (Nurmahaludin, 2014).



Gambar 2. 7 Model Neural Network

Sumber: <https://Neural Network dengan R dan Python> | by Jimmy | Medium

Jaringan Saraf tiruan dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi dengan asumsi:

- a. Pemrosesan informasi terjadi pada sejumlah elemen sederhana yang disebut dengan *neuron*.
- b. Sinyal dikirimkan di antara *neuron* tersebut melalui suatu interkoneksi.
- c. Penghubung antar *neuron* memiliki bobot yang akan memperkuat atau memperl lemah sinyal.
- d. Untuk menentukan *output*, tiap *neuron* menggunakan fungsi aktivasi yang dikenakan pada jumlah *input* terbobot yang diterima.

Output Y menerima input dari *neuron* x_1 dan x_2 dengan bobot hubungan masing-masing adalah w_1 dan w_2 Kedua *neuron* yang ada dijumlahkan dengan penghitungan.

$$net = x_1w_1 + x_2w_2$$

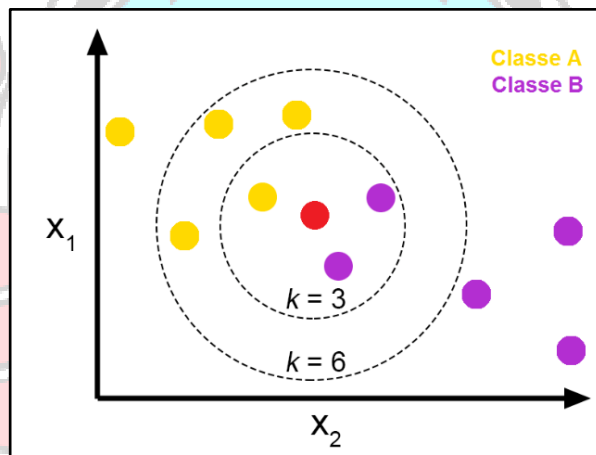
Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $Y = f(net)$. Jika nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan (Yogi & Projo, n.d.).

Telah banyak penelitian yang menggunakan metode *Neural Network* untuk teknik klasifikasi diantaranya untuk klasifikasi data *microarray* dengan akurasi rata-rata diatas 90% (Kumar et al., 2015). Metode *neural network* memiliki kelebihan yaitu toleransi yang tinggi terhadap data yang mengganggu, kemampuan untuk mengklasifikasikan pola, cocok untuk input dan output yang bersifat *continue*, berhasil untuk data nyata yang ada didunia. Metode *neural network* juga masih memiliki kelemahan diantaranya membutuhkan sejumlah parameter yang terbaik ditentukan secara empiris, waktu percobaan yang lama, dan *interpretability* yang buruk (Gupta et al., 2015).

2.7.5 K-Nearest Neighbor (K-NN)

Algoritma K-NN (*K-Nearest Neighbor*) adalah sebuah algoritma untuk

mengklasifikasikan objek berdasarkan data latih yang mempunyai jarak paling dekat dengan objek tersebut (Leidiyana, 2013). Algoritma K-NN bersifat sederhana, bekerja berdasarkan jarak terpendek dari data uji ke data latih untuk menentukan kelas dari data tersebut. Setelah mengumpulkan data-data pada kelompok k tertentu, kemudian diambil kelas data mayoritas untuk dijadikan sebagai kelas prediksi dari data uji. K-NN memiliki beberapa kelebihan yaitu tahan terhadap data yang memiliki *noise* dan efektif terhadap data latih yang berjumlah besar dan memiliki performa cukup baik. Namun waktu yang digunakan untuk komputasi sangatlah lama jika data latihnya besar dan sangat *sensitive* dengan ciri yang redundan atau relevan (Gata, 2017).



Gambar 2. 8 Model KNN

Sumber: <https://Klasifikasi-K-NN-Menggunakan-Data-Tooth-Growth-di-R> | by QOLBIYATUL LINA | Medium

Ada banyak cara untuk mengukur jarak kedekatan antar data pada algoritma K-NN diantaranya menggunakan *Euclidean distance*. *Euclidean distance* merupakan cara yang sering digunakan untuk menghitung jarak antar data. Jarak ini digunakan untuk menguji interpretasi kedekatan jarak antara dua objek. Berikut adalah persamaan dari *Euclidean distance* (Leidiyana, 2013) :

$$dist = \sqrt{\sum_{i=1}^n (X_{i2} - X_{i1})^2}$$

dengan $dist$ adalah Jarak, X_{i2} adalah Data uji, X_{i1} adalah Data Sampel, i adalah Atribut dan n adalah jumlah atribut.

2.9. Confusion Matrix

Confusion matrix adalah alat yang berguna untuk menganalisis seberapa baik atau seberapa akurat metode klasifikasi dapat mengenali objek pengamatan dari kelas yang berbeda (Maalouf, M., & Siddiqi, 2014).

		Actual Values	
		1 (Positive)	0 (Negative)
Predicted Values	1 (Positive)	TP (True Positive)	FP (False Positive) <i>Type I Error</i>
	0 (Negative)	FN (False Negative) <i>Type II Error</i>	TN (True Negative)

Gambar 2. 9 Tabel *Confusion Matrix*

Sumber: <https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62>

Beberapa ukuran kinerja klasifikasi yang dapat diperoleh dari confusion matrix

$$\text{Akurasi} = \frac{TP+TN}{TP+FP+FN+TN} \quad (10)$$

$$\text{Recall (Sensitivity)} = \frac{TP}{TP+FN} \quad (11)$$

$$\text{Specificity} = \frac{TN}{FP+TN} \quad (12)$$

$$\text{Precision} = \frac{TP}{TP+FP} \quad (12)$$

$$\text{F1 score} = \frac{2(\text{Recall} \times \text{Precision})}{\text{Recall} + \text{Precision}} \quad (12)$$

Akurasi adalah metode yang paling umum digunakan untuk menilai kinerja klasifikasi. Namun, untuk kasus imbalanced data, akurasi menempatkan lebih banyak bobot pada kelas mayoritas atau kelas yang tidak menjadi perhatian. Oleh karena itu,

akurasi tidak boleh digunakan sebagai ukuran kinerja klasifikasi kasus *imbalanced* data (Maalouf, M., & Trafalis, 2011), sehingga perlu juga memperhatikan *sensitivity* dan *specificity*.

Sensitivity (recall) merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif, sedangkan Parameter *specificity* menurut (Arthana.R, 2019) adalah parameter kebenaran dalam hal memprediksi negatif (*True Negative*) dibandingkan dengan keseluruhan data negatif. Dalam istilah yang lebih mudah dipahami, *specificity* adalah kemampuan model untuk mengidentifikasi seberapa banyak kasus negatif yang benar-benar negatif.

Nilai *True Negative (TN)* memberikan informasi Ketika *classifier* melakukan klasifikasi data bernilai benar, sedangkan *False Positive (FP)* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data.. Pada kasus *imbalanced* data nilai *sensitivity* diperlukan untuk mendeteksi kelas minoritas (*Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?* | by Salma Ghoneim | Towards Data Science, n.d.).

Parameter *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif (Fajar Nugroho, 2019), pada penelitian ini *precision* dibutuhkan untuk menentukan transaksi nasabah yang normal (mayoritas). Perbedaan antara *sensitivity* dan *precision* di antara keduanya, *precision* punya variabel *False Positive (FP)* sedangkan *sensitivity* punya variabel *False Negative (FN)*. Pada penelitian kali ini dibutuhkan nilai *recall* yang tinggi untuk menghindari kasus model memprediksi transaksi yang tidak bernilai *fraud* (normal) padahal sebenarnya merupakan kasus *fraud* (Islam et al., 2020).

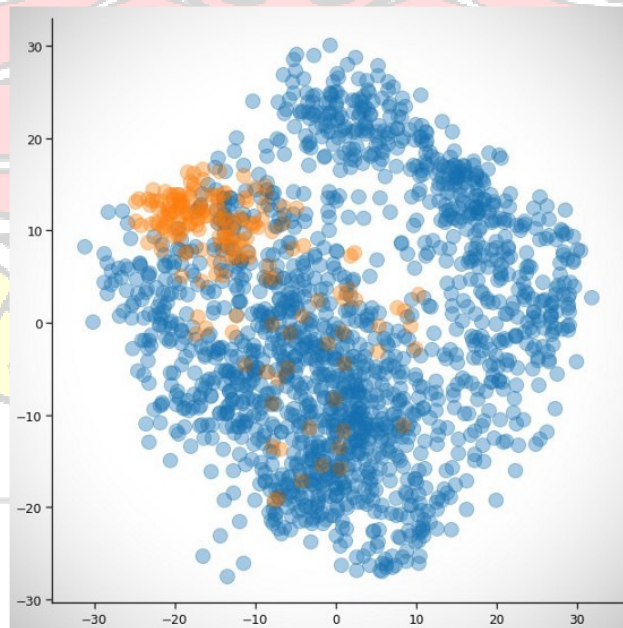
Lalu ada parameter *F1 score* yang merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan. Evaluasi *F1 score* dapat digunakan sebagai salah satu performa model jika ada *tradeoff* yaitu nilai *precision* yang tinggi mengakibatkan nilai *recall* yang rendah, begitupun sebaliknya (Juba & Le, 2019).

Selain itu, untuk kasus *imbalanced* data dapat diatasi dengan menggunakan

Geometric Mean (G-mean) dan kurva *Receiver Operating Characteristic* (ROC) karena tidak tergantung pada distribusi objek pengamatan antar kelas (Kubat, M., & Matwin, 1997). Nilai tunggal yang dapat digunakan untuk mengukur kinerja klasifikasi pada kurva ROC adalah *Area Under Curve the ROC* (AUC).

2.10. *Imbalanced Data*

Pada proses klasifikasi biasanya terdapat beragam permasalahan pada data, baik dari *preprocessing data*, *modeling*, *evaluasi*, dan lainnya. Kadang salah satu hal yang kurang disadari dari proses klasifikasi adalah pada jumlah ataupun proporsi dari label/*class* yang kita hadapi. Bisa jadi yang kita data yang kita hadapi adalah *imbalanced data*. *Imbalanced data* merupakan data yang biasanya diolah secara klasifikasi dengan salah satu kelas/label pada datanya mempunyai nilai yang sangat jauh berbeda jumlahnya dari kelas lainnya. Pada *imbalanced data*, biasanya dataset memiliki data dengan kelas yang sedikit (*rare class*) dan data dengan kelas yang banyak (*abundant class*) (Fernández et al., 2018). Beberapa contoh kasus yang sering terjadi imbalance dataset yaitu *credit scoring*, *fraud*, data penyakit, dsb.



Gambar 2. 10 *Imbalanced Data*

Sumber: <https://medium.com/analytics-vidhya/balance-your-data-using-smote-98e4d79fcddb>

Imbalanced class adalah kondisi distribusi antar kelas yang tidak seimbang pada suatu dataset, dimana salah satu kelasnya memiliki jumlah data yang sangat besar (kelas mayoritas) dibanding kelas lainnya (kelas minoritas) (He & Garcia, 2009). Perbedaan jumlah data yang besar antar kelas dapat mengakibatkan model klasifikasi sering tidak dapat memprediksikan kelas minoritas dengan tepat sehingga banyak data tes yang seharusnya berada pada kelas minoritas diprediksikan salah oleh model klasifikasi (Sun et al., 2009). Untuk mengatasi permasalahan *imbalance class*, salah satu metode yang digunakan adalah sampling. Metode sampling melakukan modifikasi terhadap distribusi data antar kelas mayoritas dan kelas minoritas pada dataset training untuk menyeimbangkan jumlah data tiap kelas (Hoang et al., 2009).

2.11. Random Oversampling

Random Oversampling merupakan teknik untuk meningkatkan jumlah *minority class* sebanyak jumlah *majority class*. Beberapa metode yang termasuk dalam *oversampling* yaitu *Naive random over-sampling*, ADASYN, SMOTE, dll. Tujuan dari *oversampling* ini sendiri adalah mencoba menyeimbangkan dataset dengan meningkatkan ukuran sampel langka. Tidak ada keunggulan absolut dari satu metode resampling atas yang lain. Penerapan kedua metode ini bergantung pada use case yang digunakan dan dataset itu sendiri. Kombinasi *over*-dan *under-sampling* sering berhasil juga (Binus, 2019). *Random Under Sampling* merupakan salah satu metode *resampling* yang sering digunakan untuk menangani ketidakseimbangan kelas. Prinsip utama metode ini adalah membuat jumlah kelas mayoritas sama dengan jumlah kelas minoritas. Implementasi metode *random under sampling* adalah memilih dataset terlebih dahulu kemudian menghitung jumlah kelas mayoritas dan minoritas. Setelah itu, ambil kelas mayoritas secara acak yang jumlah datanya sama dengan kelas minoritas. Metode memiliki keunggulan dalam menghemat waktu prediksi jika diterapkan dalam model klasifikasi.

2.12. K-Fold Cross Validation

K-fold cross validation merupakan sebuah pendekatan yang bertujuan untuk

mengevaluasi kinerja algoritma dengan terlebih dahulu membagi data menjadi *data training* (data latih) dan *data testing* (data uji). Dalam proses *cross validation*, data akan dibagi dalam k buah partisi dengan ukuran yang sama dengan nilai k lebih dari sama dengan 2. Selanjutnya, setiap jumlah folder yang ditentukan akan dibagi menjadi satu pecahan yang akan menjadi data testing dan sisanya akan menjadi data training. Prosedur tersebut dilakukan sebanyak k kali sehingga setiap pecahan akan menjadi *data testing* tepat satu kali dan akan menjadi data training sebanyak $k-1$ kali.

2.13. Information Gain

Information Gain adalah salah satu metode dari seleksi fitur, dalam proses *Information Gain* fitur akan diranking, ranking fitur yang terbesar merupakan fitur yang paling relevan dan memiliki koneksi yang kuat dengan kumpulan data yang terkait (Varghese & Sushmita, 2014). Teknik ini meranking fitur dengan menghitung *Entropy* dari salah satu kelas sebelum dan setelah melakukan proses pengamatan terhadap fitur yang ada pada satu data yang sama (Purbasari & Nugroho, 2013)

2.14. Orange Data Mining

Orange merupakan *software open source* untuk pengolahan *Data Analytics / Data Mining* yang dikembangkan di Laboratorium Bioinformatika Fakultas Ilmu Komputer dan Informasi *University of Ljubljana*, dengan target menghadirkan sebuah *software* untuk mengolah data secara visual tanpa perlu pengalaman membuat program sebelumnya.. Ini dapat digunakan baik melalui skrip *Python* sebagai *plug-in Python*, atau melalui pemrograman visual antarmuka (Andrybrew, 2015).

Orange menyediakan banyak *widget* yang dapat diletakkan pada *canvas / drawing board* kemudian dapat dihubungkan dengan *widget-widget* lainnya. Dengan adanya media *canvas* ini, dapat memudahkan *user* bermain dengan data dan melakukan proses *data analytics* secara intuitif. *Software* ini juga dapat digunakan untuk pemilihan eksperimen, pemodelan prediktif, dan sistem rekomendasi dan dapat

digunakan untuk penelitian genomik, biomedis, bioinformatika, dan pengajaran.



Gambar 2. 11 Logo Orange

2.15. Penelitian Terkait

Beberapa penelitian terkait penggunaan metode *Ensemble Stacking* dalam menyelesaikan ketidakseimbangan data serta akurasi optimal, seperti pada jurnal dengan judul “Mendeteksi serangan *web* menggunakan *random undersampling* dan *ensemble learners*“, menyebutkan bahwa Ketidakseimbangan kelas merupakan pertimbangan penting untuk keamanan siber dan pembelajaran mesin (*machine learning*) (Richard et al., 2021). Berikut adalah beberapa studi literatur yang menjadi referensi pada penelitian ini:

Tabel 2. 1 Literatur *Review*

No.	Judul	Penulis	Tahun	Metode	Hasil
1.	Serangan Siber dan Deteksi Penipuan Menggunakan <i>Ensemble Stacking</i>	Raha Soleymanzadeh, Mustafa Aljasim, Muhammad Waseem Qadeer, Rasha Kashef	2022	Metode <i>Ensemble Stacking</i> untuk secara efektif mengungkap serangan siber di IoT dengan kinerja tinggi	Menunjukkan peningkatan yang signifikan dibandingkan dengan sebagian besar makalah yang diulas dan dibahas di Bagian 2
2.	Teknik <i>Ensemble</i> untuk Deteksi Penipuan Kartu Kredit	Satya Dileep Penmetsa, Sabah Mohammed	2021	Metode <i>ensemble stacking machine learning</i>	Penumpukan berdasarkan algoritma <i>machine learning</i> yang

				seperti regresi <i>logistic</i> , <i>K-nearest neighbor</i> , <i>random forest</i> , <i>naïve bayes</i> , <i>multilayer perceptron</i> , <i>adaboost</i>	kuat menghasilkan kinerja yang lebih tinggi daripada model pembelajaran mesin individu mana pun
3.	Penerapan <i>Ensemble Stacking</i> untuk Klasifikasi <i>Multi Kelas</i>	Rio Ananda Fitriansyah	2016	<i>Ensemble method stacking</i> diterapkan, <i>decision tree</i> dan <i>naïve bayes</i> digunakan sebagai <i>Classifier</i> dasar	Hasil pengujian menunjukkan metode <i>ensemble stacking</i> hanya mampu melampaui pada beberapa <i>dataset</i> jika dibandingkan dengan <i>hybrid classifier</i>
4.	<i>Optimized Stacking Ensemble (OSE)</i> untuk Penipuan Kartu Kredit Deteksi Menggunakan Model <i>Sintetis Minoritas Oversampling</i>	Karen Charly Veigas, Durga Srilekha Reguladda, Sujutha Arun Kokatnoor	2021	Metode <i>Synthetic Minority Oversampling Technique (SMOTE)</i> dan <i>Generative Adversarial Networks (GAN)</i> untuk memperkirakan distribusi data di antara dua kelas dalam dataset asli	Hasil empiris yang diperoleh dalam makalah ini menunjukkan bahwa kualitas dataset akhir sangat meningkat ketika SMOTE dan GAN digunakan sebagai algoritma <i>oversampling</i> , model yang dioptimalkan dibangun menggunakan <i>Stacking Classifier</i> yang

					menggabungkan model <i>Multi-Perceptron</i> yang ditingkatkan GAN dengan model klasifikasi standar lainnya seperti KNN dan SVM.
5.	Mendeteksi Serangan Web Menggunakan <i>Random Undersampling</i> dan <i>Ensemble Learners</i>	Richard Zuech, John Hancock, Taghi M.Khosroftaar	2021	Metode untuk klasifikasi yaitu : <i>Decision Tree (DT)</i> , <i>Random Forest (RF)</i> , <i>CatBoost (CB)</i> , <i>LightGBM (LGB)</i> , <i>XGBoost (XGB)</i> , <i>Naïve Bayes (NB)</i> , dan <i>Logistic Regression (LR)</i>	Pengklasifikasi <i>ensemble</i> umumnya berkinerja lebih baik daripada tiga rekan <i>non-ensemble</i> , tetapi hanya setelah <i>undersampling</i> acak diterapkan. Sebelum RUS diterapkan, semua pengklasifikasi rusak dengan kinerja klasifikasi yang buruk. Interaksi antara <i>classifer</i> dan rasio <i>sampling</i> juga signifikan secara statistik.
6.	Teknik Evolusi Diferensial pada Pemilihan Berbotoh Metode <i>Stacking Ensemble</i> untuk Penipuan	Kgaugelo Moses Dolo	2019	Metode <i>Synthetic Minority Oversampling Technique (SMOTE)</i> dan <i>Safe Level-Synthetic</i>	Menunjukkan bahwa Evolusi Diferensial Metode optimasi adalah fungsi pembobotan yang baik, yang

	Kartu Kredit Deteksi		<i>Minority Oversampling Technique (SL-SMOTE)</i>	dapat diadopsi sebagai fungsi bobot sistematis untuk metode ensemble susun pemungutan suara berat dari berbagai metode klasifikasi
--	----------------------	--	---	--



BAB III METODELOGI PENELITIAN

3.1. Tempat dan Waktu Penelitian

Tempat penelitian dilaksanakan di Laboratorium CNAP Kampus 1 Politeknik Negeri Ujung Pandang Jl.Perintis Kemerdekaan KM 10 Makassar, Sulawesi Selatan pada bulan Maret 2023 sampai dengan bulan Agustus 2023.

3.2. Alat dan Bahan Penelitian

Penelitian ini dimulai dengan menentukan kebutuhan yang diperlukan. Kebutuhan tersebut meliputi informasi mengenai perangkat keras, perangkat lunak, dan aplikasi pendukung lainnya yang digunakan dalam penelitian ini. Berikut spesifikasi yang dibutuhkan.

3.2.1 Perangkat Keras (*Hardware*)

Tabel 3. 1 Perangkat Keras (*Hardware*)

No	Perangkat Keras	Deskripsi	Keterangan
1	1 Unit Laptop	Processor : Intel(R) Core(TM) i5- 11400H CPU @ 2.70GHz (12CPUs), 2.7 GHz Memory : 16,00 GB RAM, Storage : 512 GB SSD	Sebagai media untuk menjalankan perangkat lunak yang dibutuhkan.

3.2.2 Perangkat Lunak (*Software*)

Tabel 3. 2 Perangkat Lunak (Software)

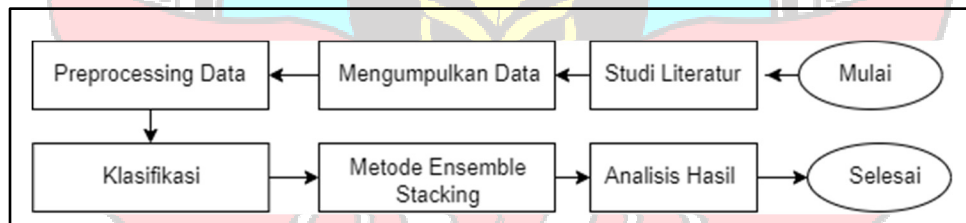
No	Perangkat Lunak	Keterangan
1	Sistem Operasi Windows 11	Sistem operasi yang digunakan
2	Orange <i>Data Mining</i> Version 3.32.0	Untuk melakukan proses <i>data mining</i> atau <i>data analytic</i>

3.2.3 Bahan

Tabel 3. 3 Bahan

No	Bahan	Keterangan
1	Dataset Credit Card Fraud	UCI <i>Machine Learning Repository: Bank Marketing Data Set</i> , n.d

3.3 Metode Penelitian



Gambar 3. 1 Alur Penelitian

Berdasarkan gambar 3.1 di atas dapat dilihat bahwa alur pada penelitian ini dimulai dari studi literatur sampai dengan analisis hasil yang penjelasannya dapat dilihat pada poin berikut :

3.3.1 Studi Literatur

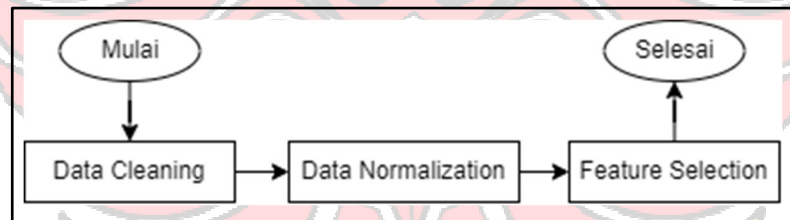
Studi literatur merupakan tahapan mempelajari metode yang akan digunakan pada penelitian, yaitu mempelajari proses algoritma yang digunakan dengan membaca berbagai referensi dan literatur berupa jurnal ilmiah atau artikel ilmiah yang berkaitan dengan metode *Stacking* diakses melalui halaman web di internet.

3.3.2 Pengumpulan Data

Penelitian ini menggunakan dataset dari website *opensource* atau publik (UCI *Machine Learning Repository: Bank Marketing Data Set*, n.d.) dengan ukuran file 3,664 *kilobytes*, dataset berkaitan dengan promosi pihak bank menggunakan data panggilan telepon dari sebuah lembaga perbankan Portugal. Dataset ini mengambil data asli yang diperoleh dari bulan Mei 2008 hingga bulan Juni 2013 dengan jumlah total 45211 baris data. Dataset yang diperoleh dari penelitian (Moro et al., 2014) telah dilakukan tahap *preprocessing* untuk memilih data yang benar-benar dibutuhkan agar data tersebut memiliki ukuran yang tidak terlalu besar untuk dilakukan penelitian.

3.3.3 Preprocessing

Data preprocessing merupakan sekumpulan teknik yang diterapkan pada *database* untuk menghapus *noise*, *missing value*, dan data yang tidak konsisten. *Data preprocessing* dibagi menjadi beberapa langkah, yaitu *cleaning data*, *data transformation*, dan *feature selection*.



Gambar 3. 2 *Preprocessing Data*

Data preprocessing ini digunakan karena dalam data *realtime database* seringkali tidak lengkap dan tidak konsisten sehingga mengakibatkan hasil *data mining* tidak tepat dan kurang akurat. Oleh karena itu, untuk meningkatkan kualitas data yang akan dianalisis, perlu dilakukan langkah-langkah *preprocessing* data. Data yang baru saja dikumpulkan kemungkinan besar memiliki banyak bagian yang tidak relevan bahkan ada bagian yang hilang. Oleh karena itu perlu adanya proses pembersihan data atau biasa dikenal dengan *data cleaning*.

Hal yang bisa diatasi menggunakan *data cleaning* adalah penanganan *missing*

value dan *noise*. *Missing value* merupakan kondisi dimana adanya data yang hilang atau tidak lengkap di dalam *database*. Cara untuk mengatasi *missing value* adalah dengan mengabaikan *row* atau baris data tersebut lalu mengisi *missing value* tersebut.

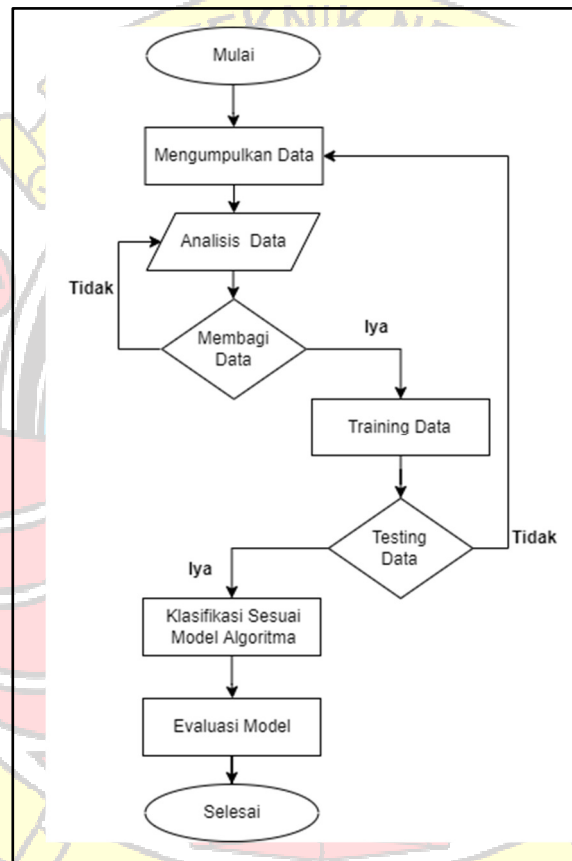
Proses selanjutnya akan dilakukan data *transformation* untuk mengubah data dalam bentuk yang sesuai dalam proses *data mining*. Teknik yang dilakukan untuk data *transformation* adalah *normalization*, pemilihan atribut, dan *discretization*. *Normalization* dilakukan untuk menskalakan nilai data dalam rentang nilai tertentu, misalnya -1 sampai 1 atau 0 sampai 1. Teknik kedua adalah pemilihan atribut. Pemilihan atribut merupakan proses pemilihan atribut yang diberikan untuk proses *data mining*. Terakhir adalah *feature selection*, adalah salah satu teknik penting dan sering digunakan dalam tahap pre-processing. Teknik ini mengurangi jumlah fitur yang terlibat dalam menentukan suatu nilai kelas target. Fitur yang diabaikan biasanya berupa fitur yang tidak relevan dan data berlebih. Tujuan utama dari seleksi fitur ialah memilih fitur terbaik dari suatu kumpulan data fitur.

3.3.4 Klasifikasi

Secara umum pada tahap ini data akan melewati dua tahapan proses, yakni *data training* dan *data testing*. *Data training* bertujuan untuk melatih data yang akan dianalisa oleh mesin dan akan memberikan validasi data. *Dataset* yang dilatih terlebih dahulu akan menghasilkan keseimbangan pada masalah *overfitting* dan *underfitting*. Sedangkan pada proses *data testing dataset* akan diuji untuk dilihat tingkat akurasi atau bisa juga dilihat performanya sesuai model yang kita pilih.

Pada proses klasifikasi ini, hasil akurasi bergantung pada algoritma klasifikasi yang digunakan. Tiap model atau algoritma klasifikasi memiliki beragam variabel yang berbeda, sehingga akan mempengaruhi performansi tiap-tiap algoritma yang digunakan. Penelitian ini akan menggunakan lima model algoritma klasifikasi, yaitu *random forest*, *decision tree*, *support vector machine (SVM)*, *neural network*, dan *k-nearest neighbor (KNN)*.

Penelitian kali ini akan menggunakan pembagian dataset menggunakan metode deterministik/*holdout*, yaitu dengan menentukan rasio pembagian dari kedua *dataset* tersebut. Oleh karena itu, pada penelitian ini digunakan 80% dari keseluruhan *dataset* untuk *training data* dan sisanya 20% data digunakan untuk *testing data* yang dapat menghasilkan tingkat kinerja klasifikasi. Gambaran alur proses klasifikasi dapat dilihat pada Gambar 3.4



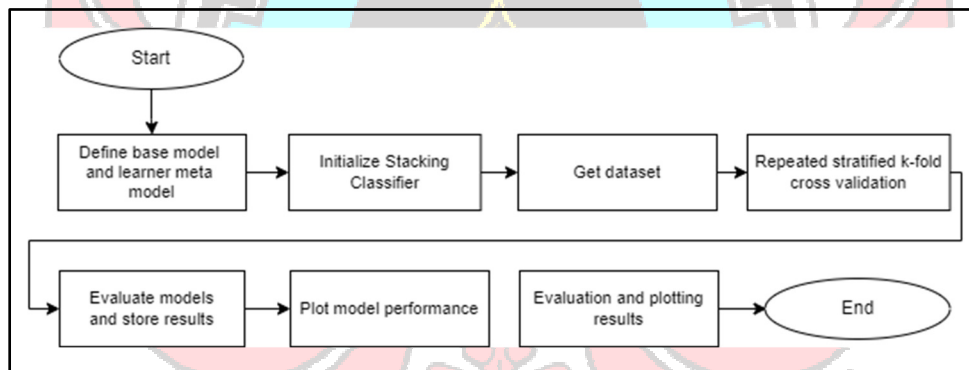
Gambar 3. 3 Alur Proses Klasifikasi

Tahap pertama yaitu melakukan pengumpulan dataset yang akan diolah dalam *orange* kemudian data di analisis, tahap analisis pertama yang dilakukan disini adalah menganalisis dataset apakah di dalam dataset tersebut terdapat *missing value*. Proses selanjutnya ialah melakukan pembagian data, proporsi pembagian data yang dipakai disini 80:20, 80% sebagai data *training* dan 20% sebagai data *testing*. Kemudian dilakukan pengujian terhadap dataset menggunakan model algoritma yang

sebelumnya sudah disepakati. Setelah pengujian dilakukan maka tahap selanjutnya adalah melihat output/hasil dari pengujian yang telah dilakukan. Hasil dari pengujian ini bisa dilihat pada *widget test & score*. Hasil pengujian akan dibandingkan dengan beberapa metode pengujian yang dilakukan dan akan dipilih model/metode mana yang paling cocok dalam mendeteksi *credit card fraud* dalam transaksi perbankan.

3.3.5 Metode *Ensemble Stacking*

Gambar 3.3 dibawah menjelaskan alur proses *ensemble stacking*. Dalam *ensemble stacking*, hasil prediksi dari masing-masing *base classifiers* digunakan sebagai input data latih untuk *meta classifier*. *Meta classifier* yang digunakan pada penelitian ini yaitu *Logistic Regression*. Hasil prediksi dari *ensemble stacking* disimpan dan divisualisasikan berdampingan dengan hasil prediksi dari *base classifiers* untuk dievaluasi.



Gambar 3. 4 *Flowchart Ensemble Stacking*

3.3.6 Analisis Hasil

Model algoritma klasifikasi yang telah diuji melalui rangkaian data testing akan dilakukan evaluasi kinerja klasifikasi. Ukuran kinerja klasifikasi yang digunakan dalam penelitian ini akan memperhatikan hasil dari berbagai parameter. Setelah hasil dari pengujian telah diketahui, maka akan dilakukan analisis perbandingan dari hasil semua algoritma yang telah diuji untuk melihat tingkat akurasi yang paling optimal untuk memecahkan masalah dari kasus yang telah disebutkan dan akan dilakukan perbandingan dengan penelitian sebelumnya.

Proses analisa dari hasil pengujian ini bisa dilakukan dengan menggunakan widget *Confusion matrix* yang terdapat pada *orange*. *Confusion matrix* digunakan untuk mengukur performa dalam permasalahan klasifikasi biner maupun permasalahan klasifikasi *multiclass*. Klasifikasi biner hanya menghasilkan dua *ouput* kelas (label), seperti “Ya” atau “Tidak”, “0” atau “1” untuk setiap data *input* yang diberikan. Kelas utama biasanya dinotasikan sebagai data positif dan yang lainnya sebagai data negatif. Hal tersebut diimplementasikan pada studi kasus di dalam penelitian ini.



BAB IV HASIL DAN PEMBAHASAN

4.1. Deskripsi Dataset

Dataset yang digunakan dalam penelitian ini bersumber dari *website opensource* atau publik (UCI Machine Learning Repository: Bank Marketing Data Set, n.d.). Dataset ini mengambil data asli yang diperoleh dari bulan Mei 2008 hingga bulan Juni 2013 dengan jumlah total 45211 baris data. Kumpulan dataset yang diperoleh menunjukkan ketidakseimbangan kelas pada data yang terindikasi *fraud* dengan jumlah data hanya 5289 atau 11,7% dari total jumlah data.

Tabel 4. 1 Informasi Dataset

No.	Fitur	Tipe	Keterangan
1.	<i>Age</i>	Numerik	Umur
2.	<i>Job</i>	Kategorikal	Pekerjaan
3.	<i>Marital</i>	Numerik	Status Pernikahan
4.	<i>Education</i>	Kategorikal	Pendidikan
5.	<i>Default</i>	Kategorikal	Status kartu kredit
6.	<i>Balance</i>	Numerik	Jumlah Transaksi
7.	<i>Housing</i>	Kategorikal	Status Rumah Pinjaman
8.	<i>Loan</i>	Kategorikal	Status Pinjaman Bank
9.	<i>Contact</i>	Kategorikal	Kontak
10.	<i>Day</i>	Kategorikal	Log Panggilan (pekan) Terakhir

11.	<i>Month</i>	Kategorikal	Log Panggilan (bulan) Terakhir
12.	<i>Duration</i>	Numerik	Durasi Panggilan Terakhir
13.	<i>Campaign</i>	Numerik	Jumlah <i>Campaign</i> Bank
14.	<i>Pdays</i>	Numerik	Jumlah Hari Setelah <i>Campaign</i>
15.	<i>Previous</i>	Numerik	Jumlah Kontak sebelum <i>Campaign</i>
16.	<i>Poutcome</i>	Kategorikal	Hasil <i>Campaign</i>
17.	<i>Y (target)</i>	Kategorikal	Indikasi <i>fraud</i>

Pada Tabel 4.1 terdapat informasi mengenai fitur-fitur yang terdapat pada dataset untuk digunakan pada penelitian ini. Fitur yang terdapat pada dataset secara keseluruhan memiliki dua tipe data yaitu tipe kategorikal dan tipe numerik. Untuk target fitur yang akan digunakan dilakukan *labeling* pada transaksi yang terindikasi *fraud*.

Dataset yang diunduh sudah berbentuk format *csv* yang terdiri dari 45211 baris dan 17 jumlah kolom, seperti yang terlihat pada Gambar 4.1.

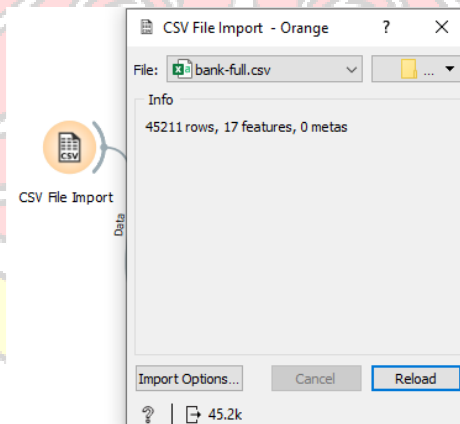
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O		
1	age;	job;	marital;	education;	default;	balance;	housing;	loan;	contact;	day;	month;	duration;	campaign;	pdays;	previous;	poutcome;	Y
2	58;	management;	married;	tertiary;	no;	2143;	yes;	no;	unknown;	5;	may;	261;	1;	-1;	0;	unknown;	no
3	44;	technician;	single;	secondary;	no;	29;	yes;	no;	unknown;	5;	may;	151;	1;	-1;	0;	unknown;	no
4	33;	entrepreneur;	married;	secondary;	no;	2;	yes;	no;	unknown;	5;	may;	76;	1;	-1;	0;	unknown;	no
5	47;	blue-collar;	married;	unknown;	no;	1506;	yes;	no;	unknown;	5;	may;	92;	1;	-1;	0;	unknown;	no
6	33;	unknown;	single;	unknown;	no;	1;	no;	no;	unknown;	5;	may;	198;	1;	-1;	0;	unknown;	no
7	35;	management;	married;	tertiary;	no;	231;	yes;	no;	unknown;	5;	may;	139;	1;	-1;	0;	unknown;	no
8	28;	management;	single;	tertiary;	no;	447;	yes;	yes;	unknown;	5;	may;	217;	1;	-1;	0;	unknown;	no
9	42;	entrepreneur;	divorced;	tertiary;	yes;	2;	yes;	no;	unknown;	5;	may;	380;	1;	-1;	0;	unknown;	no
10	58;	retired;	married;	primary;	no;	121;	yes;	no;	unknown;	5;	may;	50;	1;	-1;	0;	unknown;	no
11	43;	technician;	single;	secondary;	no;	593;	yes;	no;	unknown;	5;	may;	55;	1;	-1;	0;	unknown;	no
12	41;	admin.	divorced;	secondary;	no;	270;	yes;	no;	unknown;	5;	may;	222;	1;	-1;	0;	unknown;	no
13	29;	admin.	single;	secondary;	no;	390;	yes;	no;	unknown;	5;	may;	137;	1;	-1;	0;	unknown;	no
14	53;	technician;	married;	secondary;	no;	6;	yes;	no;	unknown;	5;	may;	517;	1;	-1;	0;	unknown;	no
15	58;	technician;	married;	unknown;	no;	71;	yes;	no;	unknown;	5;	may;	71;	1;	-1;	0;	unknown;	no
16	57;	services;	married;	secondary;	no;	162;	yes;	no;	unknown;	5;	may;	174;	1;	-1;	0;	unknown;	no
17	51;	retired;	married;	primary;	no;	229;	yes;	no;	unknown;	5;	may;	353;	1;	-1;	0;	unknown;	no
18	45;	admin.	single;	unknown;	no;	13;	yes;	no;	unknown;	5;	may;	98;	1;	-1;	0;	unknown;	no
19	57;	blue-collar;	married;	primary;	no;	52;	yes;	no;	unknown;	5;	may;	38;	1;	-1;	0;	unknown;	no
20	60;	retired;	married;	primary;	no;	60;	yes;	no;	unknown;	5;	may;	219;	1;	-1;	0;	unknown;	no
21	33;	services;	married;	secondary;	no;	0;	yes;	no;	unknown;	5;	may;	54;	1;	-1;	0;	unknown;	no
22	28;	blue-collar;	married;	secondary;	no;	723;	yes;	yes;	unknown;	5;	may;	262;	1;	-1;	0;	unknown;	no

Gambar 4. 1 Struktur Dataset

4.2. Preprocessing

4.2.1 Data Cleaning

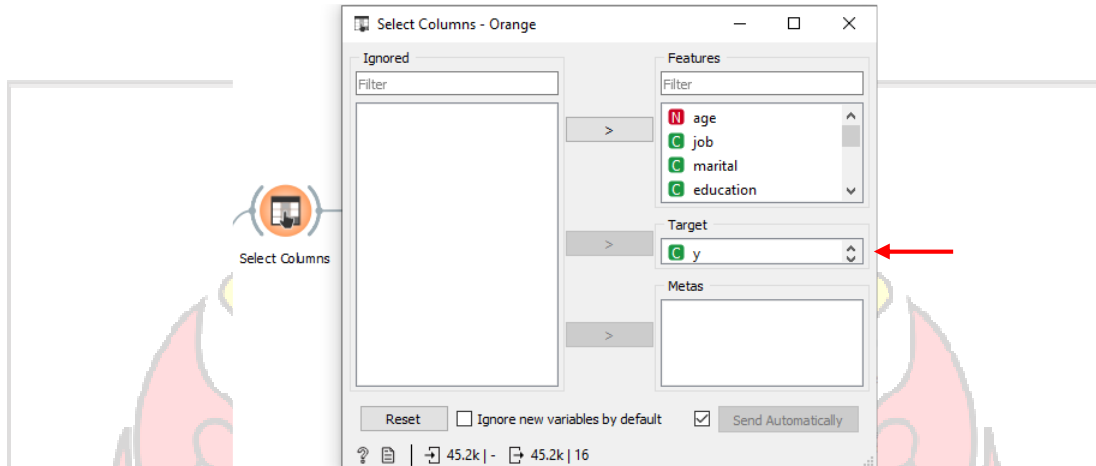
Proses pertama yang dilakukan disini adalah *import* dataset masuk kedalam aplikasi *orange*, proses ini dimulai dalam *widget CSV File Import* yang bertujuan untuk membersihkan data dari *missing value* atau data yang bernilai *null*,



Gambar 4. 2 Import Dataset

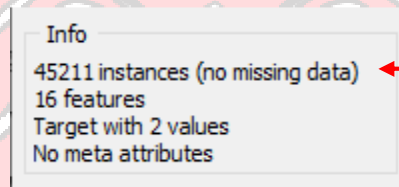
Setelah proses *import*, tahap selanjutnya adalah menentukan target di dalam dataset tersebut. Penentuan target ini berdasarkan dengan value yang terdapat pada fitur *Y* yang dimana fitur *Y* ini merupakan fitur yang dari awal memang sudah

disetting sebagai data yang terindikasi *fraud* (Moro, S at all, 2012). Penentuan target ini diatur dalam *widget Select Columns*, yang akan dijadikan target dalam dataset ini ialah fitur Y. Bisa dilihat pada Gambar 4.3.



Gambar 4. 3 Data Target

Bisa dilihat pada Gambar 4.4 dataset yang diambil dari penelitian sebelumnya sudah tidak terdapat baris yang mengandung *missing value*.



Gambar 4. 4 *Data Cleaning*

Dataset yang diperoleh tidak memiliki *row* atau baris data yang mempunyai *missing value* sehingga pada tahap *data cleaning* tidak diperlukan lagi metode untuk mengisi nilai *missing value* yang dimaksud seperti pada Gambar 4.4.

4.2.2 *Data Normalization*

Data yang diperoleh setelah diunduh berbentuk file dengan format *.csv* yang masih tidak beraturan seperti pada Gambar 4.1, data yang seharusnya terdiri dari beberapa kolom masih terlihat menyatu dan harus dipisahkan menjadi beberapa kolom. *Data Normalization* ini sendiri diatur di dalam *excel* agar fitur-fitur yang akan

digunakan dapat terlihat seperti pada Gambar 4.5.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
2	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
3	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
4	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no
5	47	blue-collar	married	unknown	no	1506	yes	no	unknown	5	may	92	1	-1	0	unknown	no
6	33	unknown	single	unknown	no	1	no	no	unknown	5	may	198	1	-1	0	unknown	no
7	35	management	married	tertiary	no	231	yes	no	unknown	5	may	139	1	-1	0	unknown	no
8	28	management	single	tertiary	no	447	yes	yes	unknown	5	may	217	1	-1	0	unknown	no
9	42	entrepreneur	divorced	tertiary	yes	2	yes	no	unknown	5	may	380	1	-1	0	unknown	no
10	58	retired	married	primary	no	121	yes	no	unknown	5	may	50	1	-1	0	unknown	no
11	43	technician	single	secondary	no	593	yes	no	unknown	5	may	55	1	-1	0	unknown	no
12	41	admin.	divorced	secondary	no	270	yes	no	unknown	5	may	222	1	-1	0	unknown	no
13	29	admin.	single	secondary	no	390	yes	no	unknown	5	may	137	1	-1	0	unknown	no
14	53	technician	married	secondary	no	6	yes	no	unknown	5	may	517	1	-1	0	unknown	no
15	58	technician	married	unknown	no	71	yes	no	unknown	5	may	71	1	-1	0	unknown	no

Gambar 4. 5 Data Transformation

Dataset perbankan sebelumnya berisi informasi-informasi penting nasabah yang berkaitan dengan nominal transaksi akan dilakukan tahap *transformation*, seperti pada atribut *balance* pada Gambar 4.5 akan dilakukan transformasi data menjadi sekumpulan nilai yang memiliki *range* atau interval. Rentang nilai tersebut diartikan sebagai nominal yang dikeluarkan nasabah saat transaksi, kemudian dilakukan *transformation* untuk menjadi sebuah nilai numerik yang lebih kecil agar mudah untuk dianalisa.

4.2.3 Feature Selection

Dataset yang berisi data penting dari nasabah merupakan sekumpulan data yang diubah menjadi sekumpulan data yang betipe numerik dan kategorikal serta memiliki atribut yang lebih sedikit dari sebelumnya. Hal tersebut mampu meningkatkan efisiensi kerja saat menganalisa data karena mengurangi penyimpanan pada memori. Atribut yang digunakan pada penelitian sebelumnya telah dilakukan *feature selection* merupakan sekumpulan data nasabah yang sebenarnya terdiri dari banyak atribut. Namun setelah direduksi pada tahap *preprocessing*, data tersebut akan lebih mudah diolah untuk penerapan model *machine learning*.

Dataset pada penelitian kali ini menerapkan teknik *preprocessing* yaitu *select relevant features* untuk mereduksi data sebelumnya, sehingga sudah siap digunakan. Untuk melihat gambaran bagaimana teknik tersebut mereduksi data seperti pada

Gambar 4.6 terdapat 16 jumlah *features* serta satu kelas target dengan dua jumlah *values*, kemudian akan direduksi menjadi hanya sepuluh fitur saja pada Gambar 4.7 dengan memilih sepuluh *components*.

Data Table - Orange

Info
45211 instances (no missing data)
16 features
Target with 2 values
No meta attributes

Variables
 Show variable labels (if present)
 Visualize numeric values
 Color by instance classes

Selection
 Select full rows

Restore Original Order
Send Automatically

	y	age	job	marital	education	default	balance	housing	no
1	no	58	management	married	tertiary	no	2143	yes	no
2	no	44	technician	single	secondary	no	29	yes	no
3	no	33	entrepreneur	married	secondary	no	2	yes	yes
4	no	47	blue-collar	married	unknown	no	1506	yes	no
5	no	33	unknown	single	unknown	no	1	no	no
6	no	35	management	married	tertiary	no	231	yes	no
7	no	28	management	single	tertiary	no	447	yes	yes
8	no	42	entrepreneur	divorced	tertiary	yes	2	yes	no
9	no	58	retired	married	primary	no	121	yes	no
10	no	43	technician	single	secondary	no	593	yes	no
11	no	41	admin.	divorced	secondary	no	270	yes	no
12	no	29	admin.	single	secondary	no	390	yes	no
13	no	53	technician	married	secondary	no	6	yes	no
14	no	58	technician	married	unknown	no	71	yes	no
15	no	57	services	married	secondary	no	162	yes	no
16	no	51	retired	married	primary	no	229	yes	no
17	no	45	admin.	single	unknown	no	13	yes	no

45.2k | 45.2k | 45.2k

Gambar 4. 6 Data Sebelum *Reduction*

Select Relevant Features

Score
Information Gain

Number of features
 Fixed: 10
 Proportion: 75.00%

Gambar 4. 7 Proses *Feature Selection*

Info
45211 instances (no missing data)
10 features
Target with 2 values
No meta attributes

Variables
 Show variable labels (if present)
 Visualize numeric values
 Color by instance classes

Selection
 Select full rows

Restore Original Order
 Send Automatically

	y	duration	poutcome	month	pdays	contact	previous	housing	job
1	no	261	unknown	may	-1	unknown	0	yes	managemer
2	no	151	unknown	may	-1	unknown	0	yes	technician
3	no	76	unknown	may	-1	unknown	0	yes	entrepreneu
4	no	92	unknown	may	-1	unknown	0	yes	blue-collar
5	no	198	unknown	may	-1	unknown	0	no	unknown
6	no	139	unknown	may	-1	unknown	0	yes	managemer
7	no	217	unknown	may	-1	unknown	0	yes	managemer
8	no	380	unknown	may	-1	unknown	0	yes	entrepreneu
9	no	50	unknown	may	-1	unknown	0	yes	retired
10	no	55	unknown	may	-1	unknown	0	yes	technician
11	no	222	unknown	may	-1	unknown	0	yes	admin.
12	no	137	unknown	may	-1	unknown	0	yes	admin.
13	no	517	unknown	may	-1	unknown	0	yes	technician
14	no	71	unknown	may	-1	unknown	0	yes	technician
15	no	174	unknown	may	-1	unknown	0	yes	services
16	no	353	unknown	may	-1	unknown	0	yes	retired
17	no	98	unknown	may	-1	unknown	0	yes	admin.
18	no	38	unknown	may	-1	unknown	0	yes	blue-collar
19	no	219	unknown	may	-1	unknown	0	yes	retired
20	no	54	unknown	may	-1	unknown	0	yes	services

Gambar 4. 8 Data setelah *Feature Selection*

Perbedaan yang terlihat tertera pada info data yang ada pada Gambar 4.6 menunjukkan adanya 16 jumlah *features*, lalu kemudian dipilih fitur yang relevan dengan hanya mengambil 10 fitur saja menggunakan teknik *information gain* sesuai pada Gambar 4.7. *Information gain* dalam *machine learning* digunakan untuk mengukur seberapa relevan / berpengaruh sebuah *feature* terhadap hasil pengukuran. Penggunaan teknik ini dapat mereduksi dimensi *feature* dengan cara mengukur reduksi *entropy* sebelum dan sesudah pemisahan. Dapat dilihat pada Gambar 4.8 data setelah dilakukan *feature selection* berubah menjadi 10 fitur saja sesuai pada info tabel.

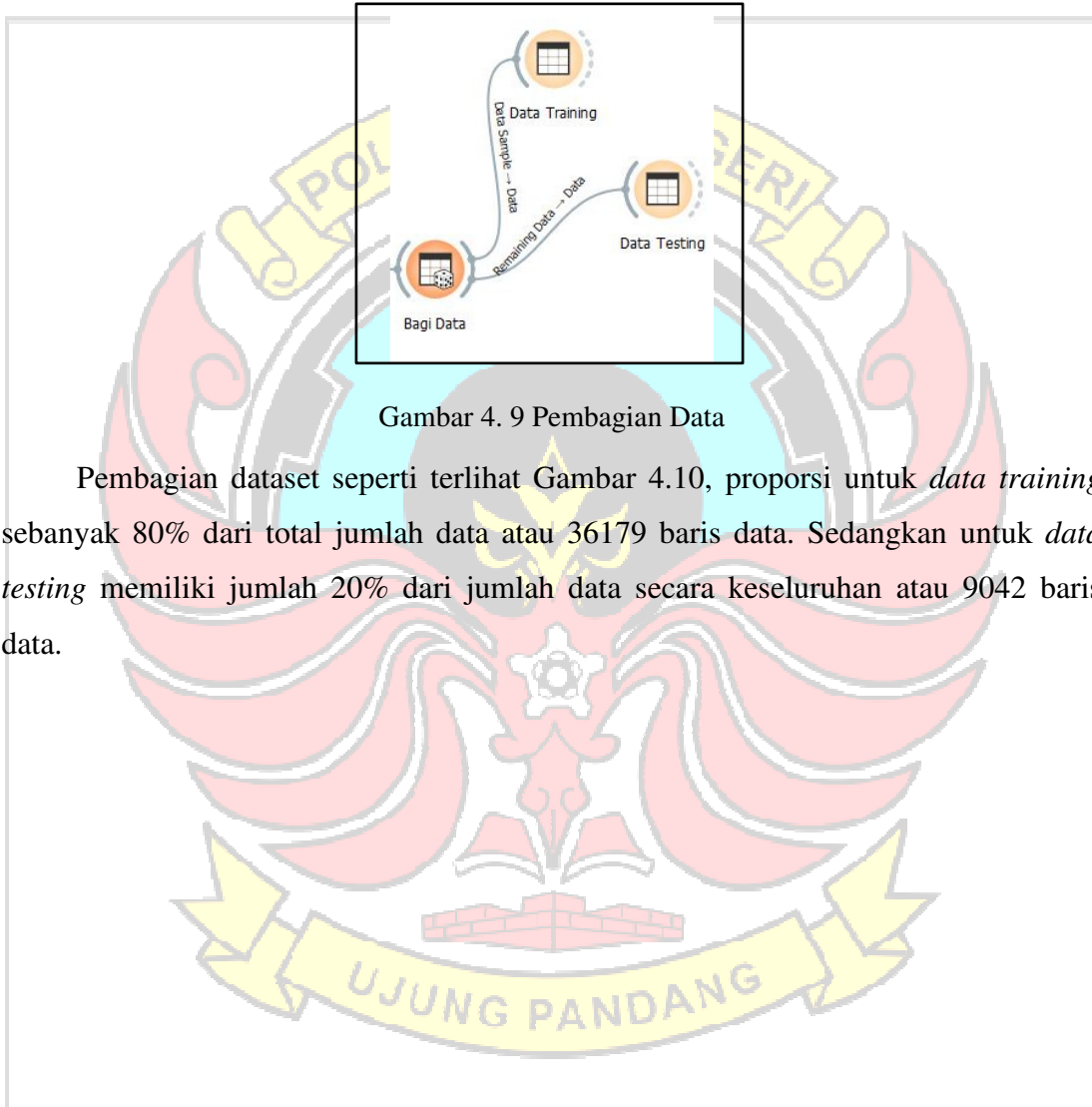
4.3. Klasifikasi

Untuk melakukan proses klasifikasi data, penelitian ini secara umum membagi kedua tahap, yaitu tanpa menggunakan metode *Ensemble Stacking* dan menggunakan metode *Ensemble Stacking*. Masing-masing proses akan menerapkan lima model

klasifikasi yang sama berbasis *Machine Learning*.

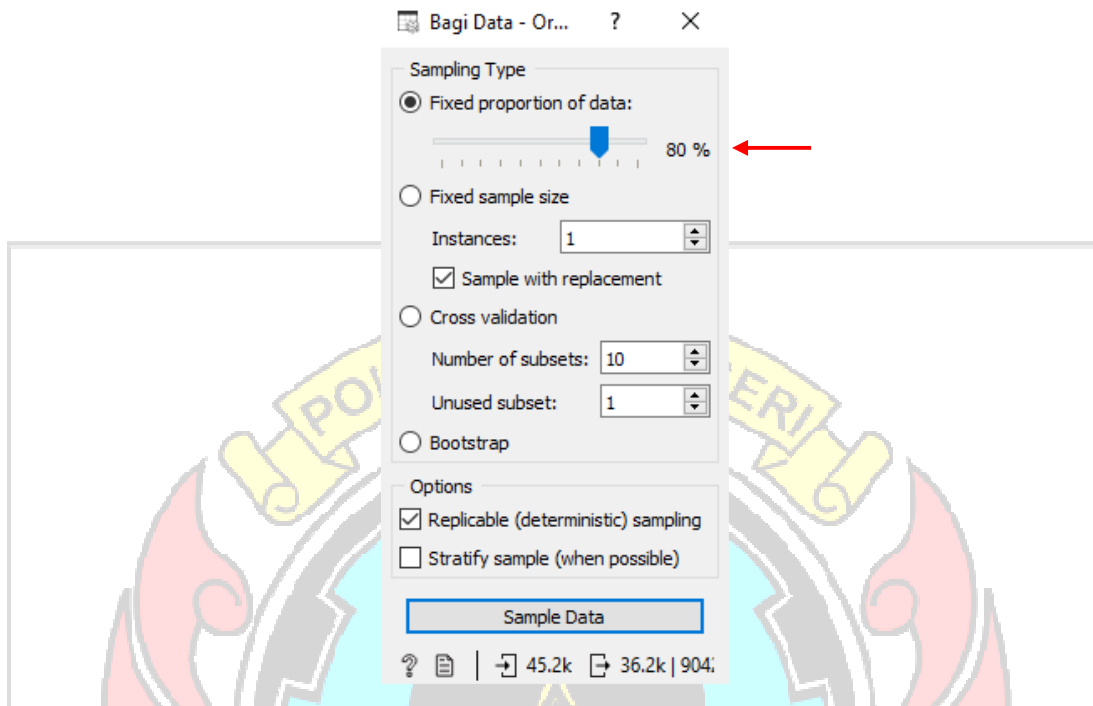
4.3.1 Tanpa Metode *Ensemble Stacking*

Sebelum menerapkan model *machine learning* terlebih dahulu membagi data menjadi dua bagian, yaitu *data training* dan *data testing*.



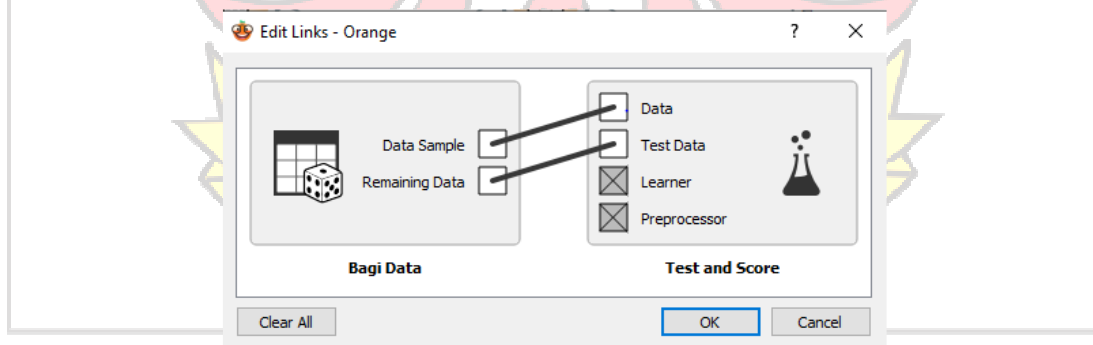
Gambar 4. 9 Pembagian Data

Pembagian dataset seperti terlihat Gambar 4.10, proporsi untuk *data training* sebanyak 80% dari total jumlah data atau 36179 baris data. Sedangkan untuk *data testing* memiliki jumlah 20% dari jumlah data secara keseluruhan atau 9042 baris data.



Gambar 4. 10 Proporsi Pembagian Data

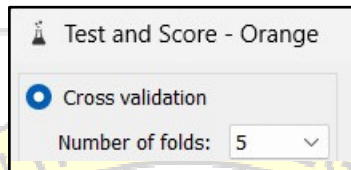
Proses selanjutnya setelah dilakukan *split data* akan langsung dilakukan pemodelan menggunakan lima model klasifikasi yang telah disebutkan sebelumnya. Penerapan model klasifikasi terlebih diimplementasikan melalui data *sample* (data *training*) dan *remaining data* (data *testing*) seperti pada Gambar 4.11.



Gambar 4. 11 Penerapan Model Klasifikasi

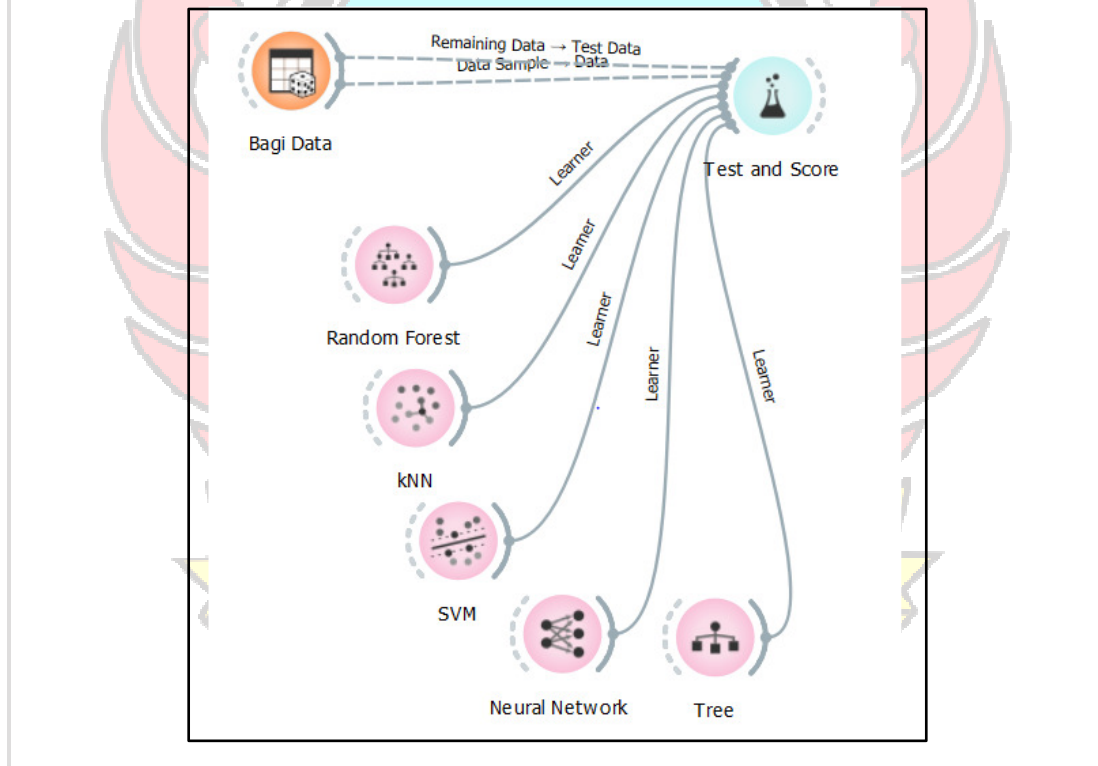
Tahap *Cross Validation* dapat dilakukan pada *widget test and score*, *Cross Validation dalam orange* membagi data menjadi beberapa bagian. Algoritma ini diuji

dengan memberikan contoh dari satu bagian pada satu waktu; model tersebut diinduksi dari kelipatan lain dan contoh dari kelipatan yang diulurkan diklasifikasikan. Hal ini diulangi untuk semua kelipatan, berikut adalah gambaran *cross validation* yang terdapat pada *orange data mining*



Gambar 4. 12 *Cross Validation*

Langkah berikutnya mengimplementasikan kelima model klasifikasi ke *data testing* untuk menguji *data testing* sekaligus melihat kinerja dari *classifier*.



Gambar 4. 13 Pengujian *Data Testing* Tanpa Metode *Ensemble Stacking*

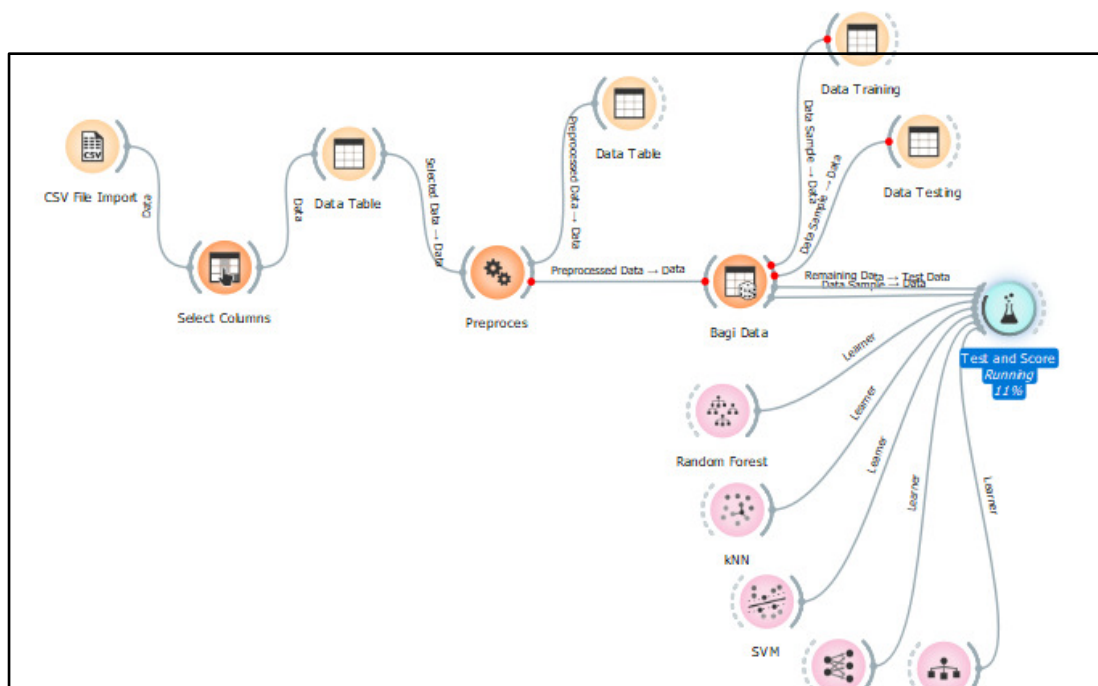
Tabel 4. 2 Performa Model Tanpa Metode *Ensemble Stacking*

Model	AUC	F1	<i>Precision</i>	<i>Recall</i>	<i>Specificity</i>	Akurasi (CA)
-------	-----	----	------------------	---------------	--------------------	--------------

<i>Random Forest</i>	0.891	0.889	0.885	0.897	0.475	0.897
<i>Neural Network</i>	0.906	0.890	0.887	0.896	0.508	0.896
<i>Tree</i>	0.591	0.877	0.872	0.886	0.433	0.886
k-NN	0.752	0.866	0.859	0.881	0.353	0.881
SVM	0.492	0.782	0.799	0.766	0.264	0.766

Nilai yang diperoleh dari pengujian tiap model memiliki rentang nilai dari 0 sebagai nilai terendah hingga nilai 1 yang mencapai nilai tertinggi. Untuk mengidentifikasi fitur yang terjadi *fraud*, maka pengujian yang dilakukan ditargetkan pada fitur *Y* dengan tipe data kategorikal yang mempunyai value 0 (transaksi normal) dan 1 (indikasi *fraud*).

Pada Tabel 4.2 bisa dilihat nilai akurasi yang mendeskripsikan tingkat keakuratan algoritma, model *Random Forest* menunjukkan nilai tertinggi dibandingkan yang lainnya, bahkan model klasifikasi *Random Forest* memiliki nilai *F1*, *precision* dan *recall* tertinggi dibanding model lainnya. Nilai akurasi tersebut harusnya bisa menjadi parameter keberhasilan dalam pemodelan klasifikasi, namun



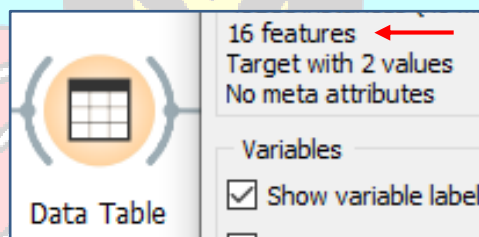
Gambar 4. 14 Alur Pengujian Tanpa Metode *Ensemble Stacking*

nilai akurasi tersebut hanya terjadi pada kelas mayoritas saja. Berikut alur *full* pengujian tanpa menggunakan metode *Ensemble Stacking*.

4.3.2 Menggunakan Metode *Ensemble Stacking Imbalanced Data*

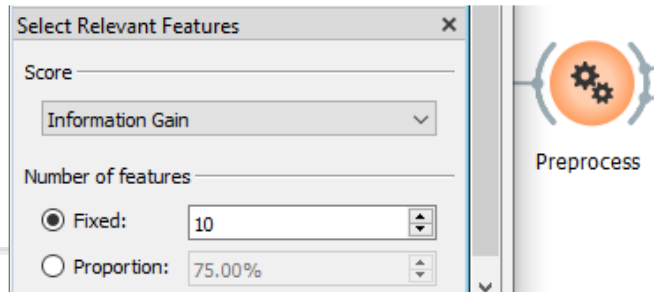
Data yang sudah dibagi akan dilakukan proses klasifikasi data menggunakan kelima model algoritma seperti yang dilakukan pada penelitian tanpa menggunakan metode *Ensemble Stacking* sebelumnya. Langkah selanjutnya mengimplementasikan kelima model klasifikasi ke *data testing* untuk menguji *data testing* sekaligus melihat kinerja dari *classifier* saat digunakan metode *Ensemble Stacking*.

Langkah pertama adalah mengimport *dataset* masuk kedalam aplikasi *orange*, *import dataset* ini dimasukkan kedalam *widget CSV File Import* kemudian tambahkan *widget Select Columns* untuk menentukan target. *Widget Select Columns* akan disambungkan ke *Widget Data Table*. Pada *Widget Data Table* akan terlihat 16 fitur karena 1 fitur lagi di jadikan sebagai .



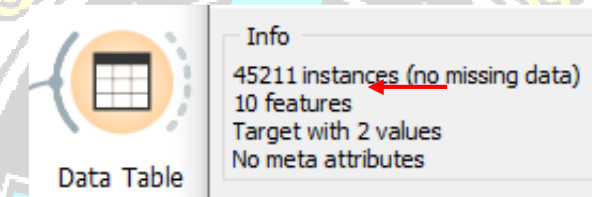
Gambar 4. 15 Jumlah *Features*

Setelah proses data table, selanjutnya adalah tahap *Preprocessing* yang dimana akan digunakan *Information Gain* untuk menyeleksi fitur yang akan dimasukkan kedalam proses berikutnya, proses *Information Gain* ini terdapat pada *widget preprocess*.



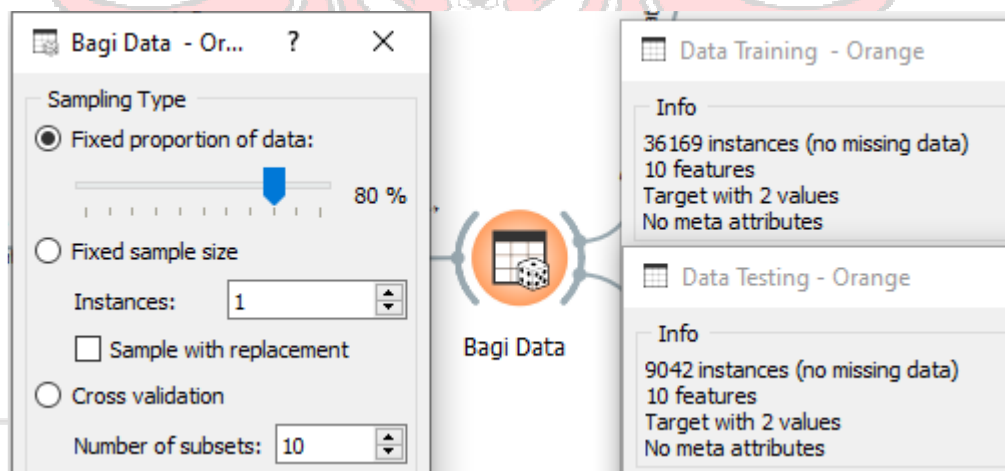
Gambar 4. 16 *Information Gain*

Setelah diterapkan *Information Gain* pada dataset maka fitur yang tersisa berjumlah 10 fitur, pengecekan fitur dapat dilihat pada *Widget Data Table*.



Gambar 4. 17 *Features Relevant*

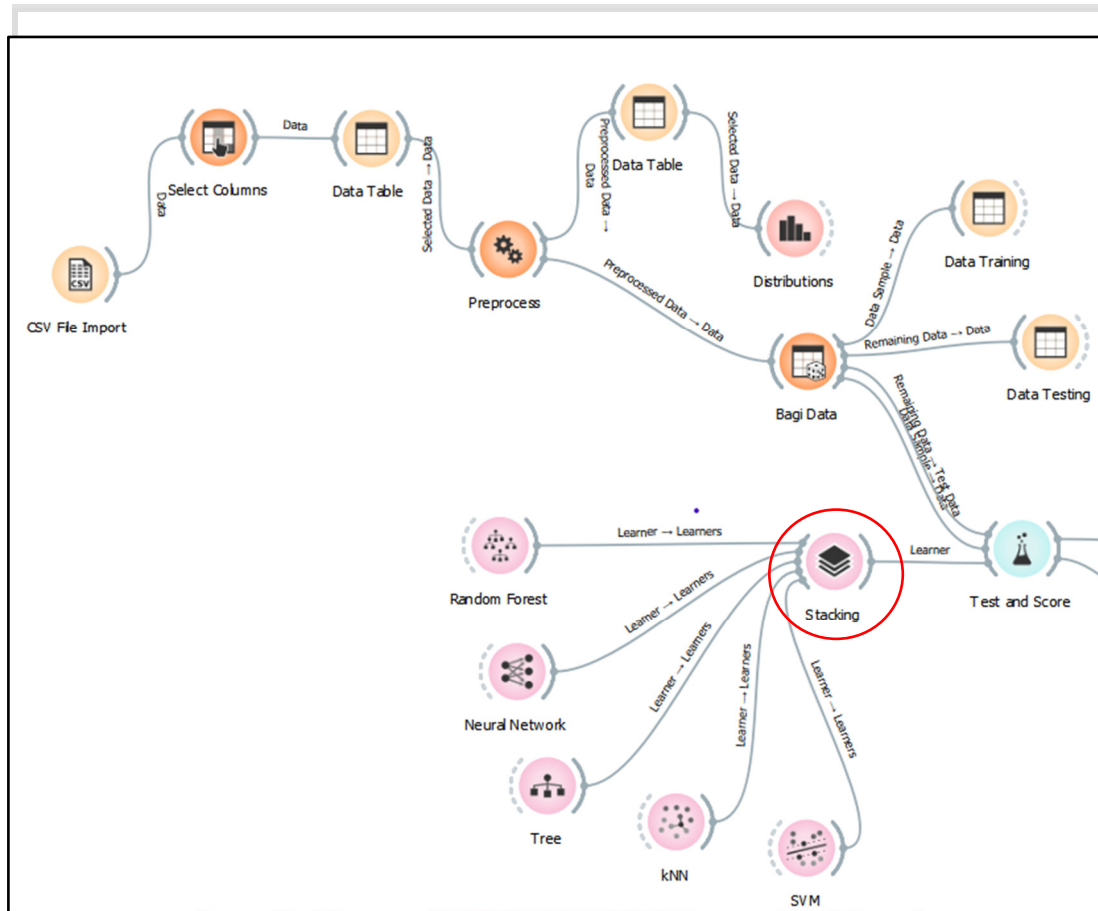
Tahap berikutnya adalah pembagian data, pembagian data ini 80:20 yang mana 80% data masuk ke dalam *Data Training* dan 20% masuk ke *Data Testing*.



Gambar 4. 18 Pembagian *Data Training* dan *Data Testing*

Alur pengujian dapat dilihat pada Gambar 4.18 yang menunjukkan bahwa *widget* tiap model disambungkan menuju *widget Test and Score* untuk mengambil

data yang sudah diuji. Garis yang menyambungkan antara *widget Split Data* / bagi data dan *Test and Score* terdapat dua keterangan data, *Data Sample* sebagai data latih sedangkan *Remaining Data* sebagai data uji. Kemudian *widget Stacking* di tempatkan di antara *widget Test and Score* dan kelima algoritma yang digunakan.



Gambar 4. 19 Pengujian Data *Testing* Dengan Metode *Ensemble Stacking Imbalanced Data*

Langkah berikutnya adalah melihat performa tiap model *classifier* menggunakan *widget Test and Score*, untuk nilai tiap model akan ditunjukkan pada Tabel 4.3 :

Tabel 4. 3 Performa Model Dengan Metode *Ensemble Stacking Imbalanced Data*

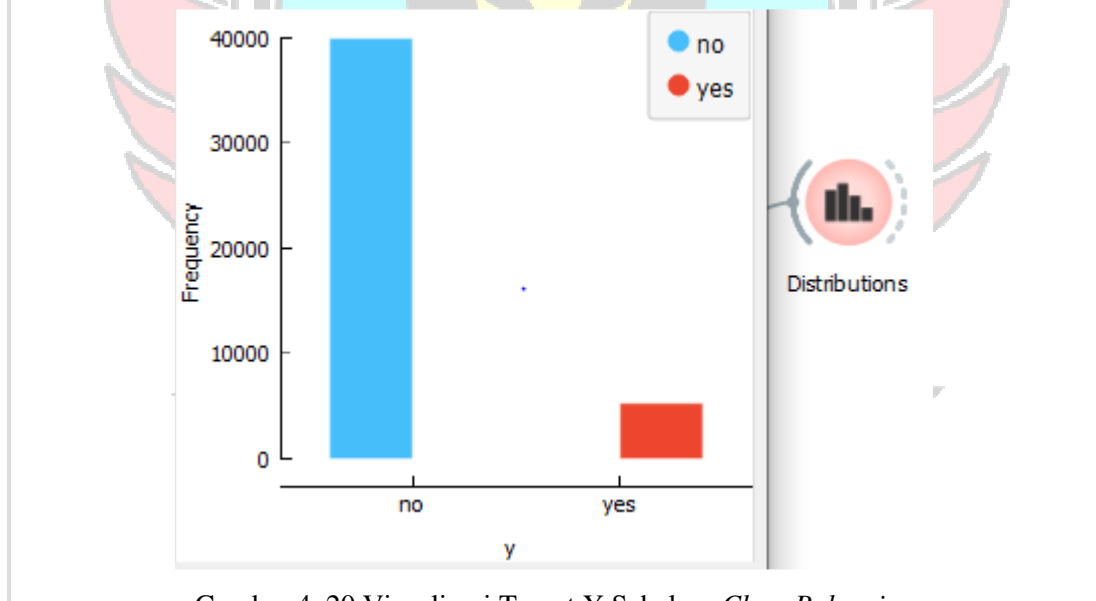
Model	AUC	F1	Precision	Recall	Specificity	Akurasi
-------	-----	----	-----------	--------	-------------	---------

						(CA)
<i>Stacking</i>	0.909	0.890	0.887	0.900	0.461	0.900

Pada Tabel 4.3 menunjukkan bahwa nilai rata-rata akurasi dari penggunaan metode *Ensemble Stacking* menyentuh nilai akurasi yang cukup tinggi yakni di atas 80%, bahkan nilai AUC, *F1 Score*, *Precision* dan *Recall* tidak ada yang menyentuh nilai dibawah 70% dari gabungan kelima model klasifikasi yang digunakan, hanya saja terdapat nilai terendah sebesar 47% di nilai *Specificity*.

4.3.3 Menggunakan Metode *Ensemble Stacking* Dengan *Class Balancing*

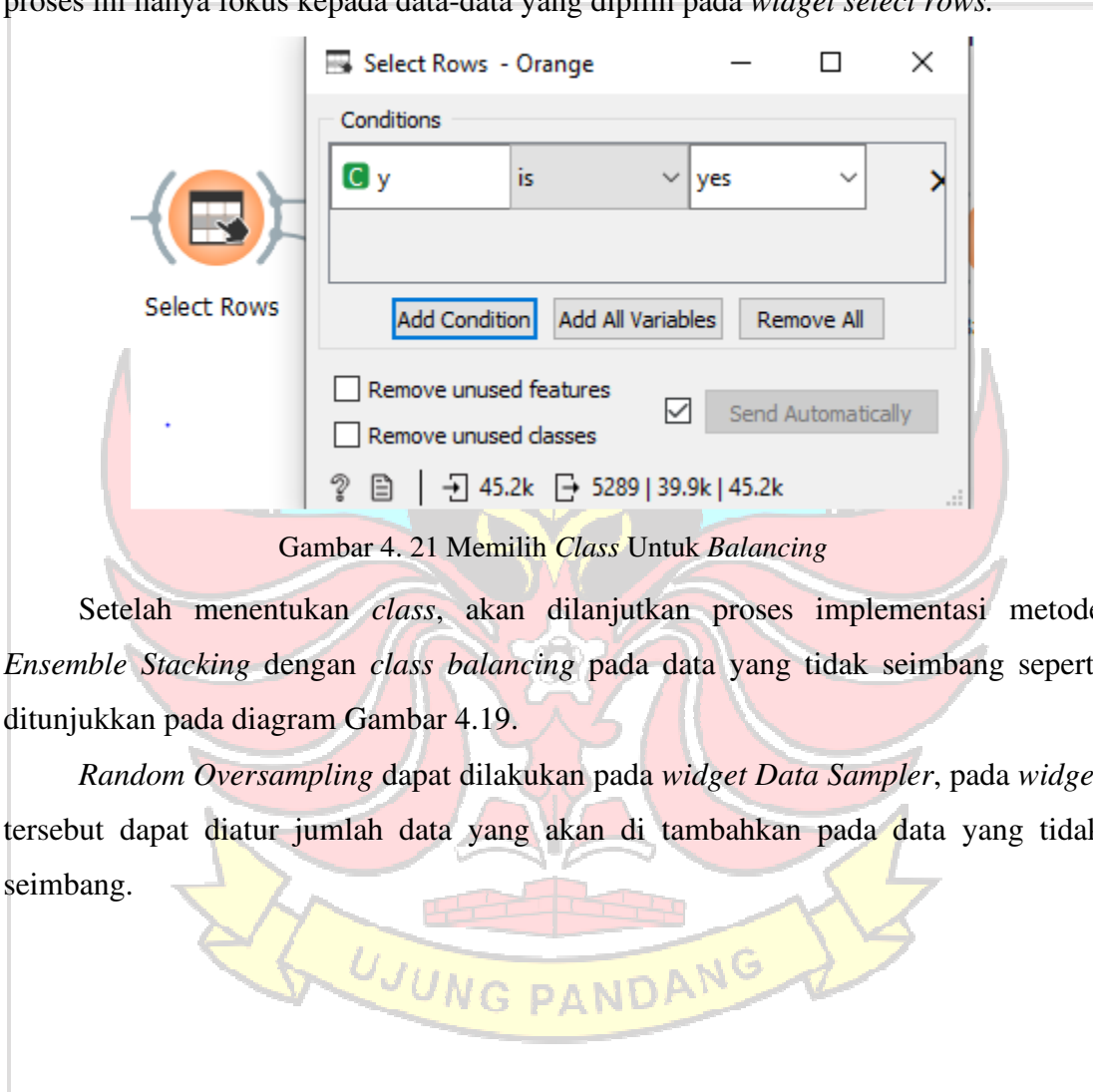
Hal pertama yang dilakukan setelah *import dataset* menggunakan aplikasi *orange* adalah melakukan *class balancing* terhadap dataset yang akan diolah. Pada penelitian ini fitur *Y* memiliki kelas *imbalance* dimana *Y* memiliki *value Yes* merupakan data minoritas seperti yang terlihat pada Gambar 4.19



Gambar 4. 20 Visualisasi Target Y Sebelum *Class Balancing*

Visualisasi diatas dapat dilihat pada *Widget Distributions*. Diagram batang *Y Yes* yang ditunjukkan dengan warna diagram merah mendiskripsikan data kasus *fraud* yang akan dilakukan *random oversampling* karena memiliki jumlah yang sangat

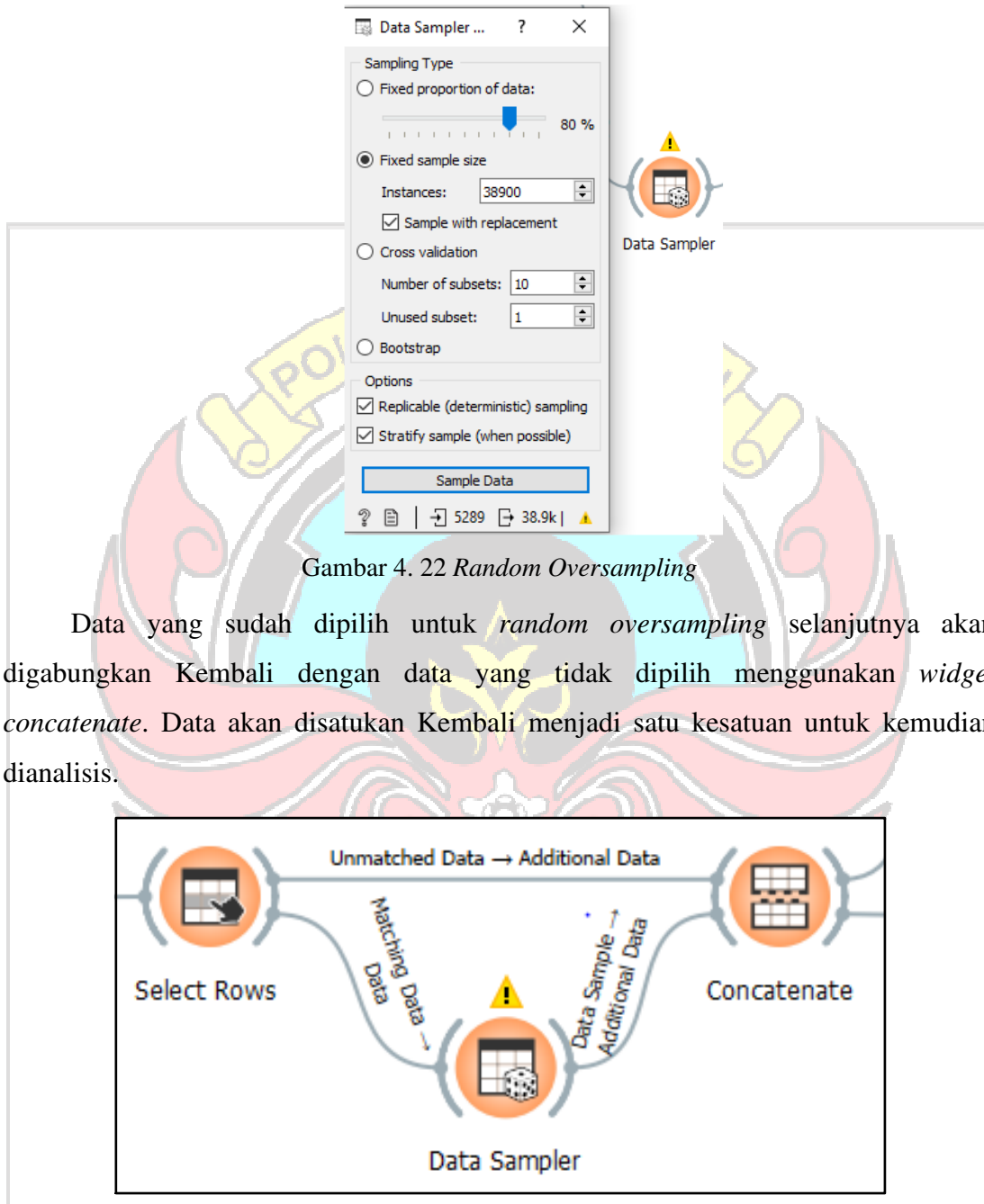
timpang dengan *Y No* yang ditunjukkan oleh diagram batang berwarna biru. Oleh karena itu, maka pada aplikasi *orange* akan memilih baris atau *select rows* yang bernilai *Yes* seperti pada Gambar 4.20. Baris tersebut dipilih untuk dipisahkan dengan *class* lain yang tidak diterapkan Teknik *random oversampling*, sehingga proses ini hanya fokus kepada data-data yang dipilih pada *widget select rows*.



Gambar 4. 21 Memilih *Class* Untuk *Balancing*

Setelah menentukan *class*, akan dilanjutkan proses implementasi metode *Ensemble Stacking* dengan *class balancing* pada data yang tidak seimbang seperti ditunjukkan pada diagram Gambar 4.19.

Random Oversampling dapat dilakukan pada *widget Data Sampler*, pada *widget* tersebut dapat diatur jumlah data yang akan di tambahkan pada data yang tidak seimbang.



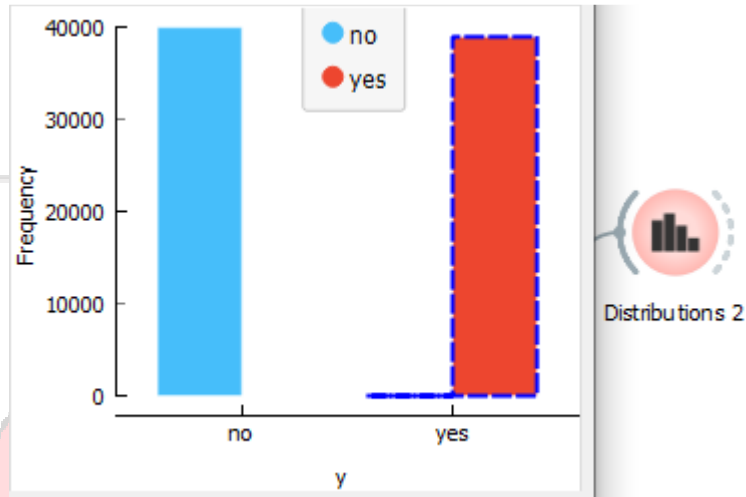
Gambar 4. 22 *Random Oversampling*

Data yang sudah dipilih untuk *random oversampling* selanjutnya akan digabungkan Kembali dengan data yang tidak dipilih menggunakan *widget concatenate*. Data akan disatukan Kembali menjadi satu kesatuan untuk kemudian dianalisis.

Gambar 4. 23 Penggabungan Data

Kemudian akan dilihat perbandingan distribusi *class* setelah menggunakan metode *Ensemble Stacking* dengan *Class Balancing* akan tampak seperti pada Gambar 4.22, jumlah *class* yang sebelumnya jauh berbeda sudah memiliki jumlah

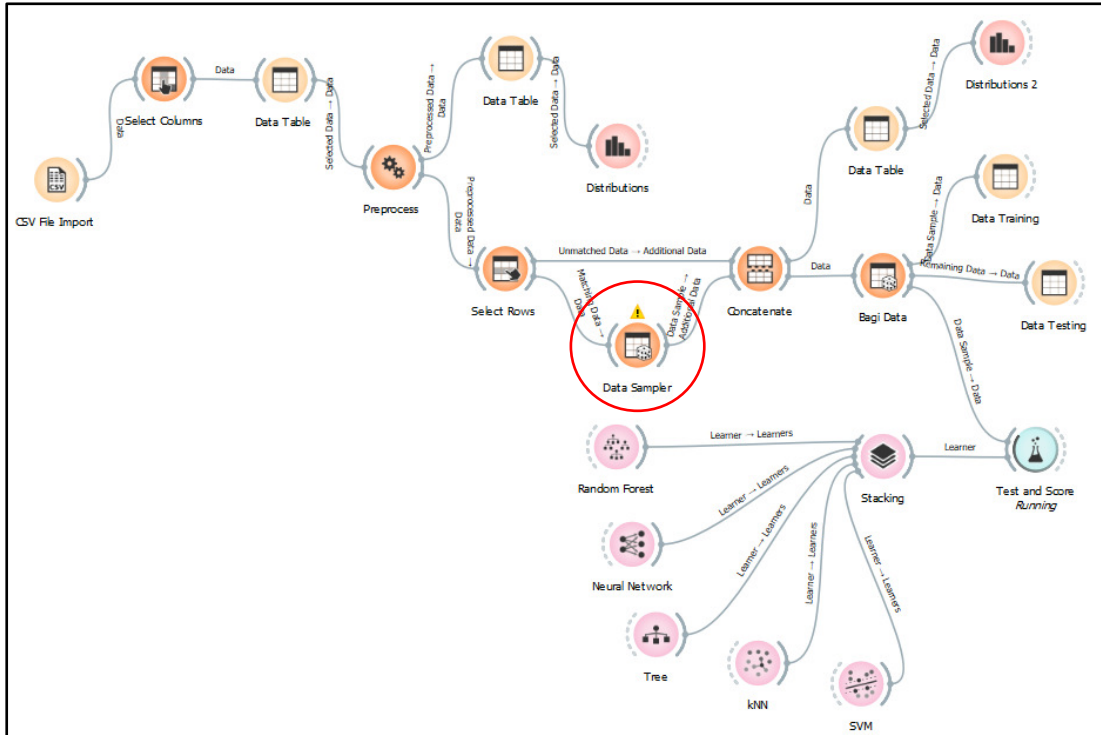
yang hampir mendekati/seimbang.



Gambar 4. 24 Visualisasi *Class* Setelah *Balancing*

Data kelas minoritas setelah *random oversampling* menggunakan metode *Ensemble Stacking* dengan *class balancing* memiliki presentasi jumlah data yang lebih baik.

Selanjutnya dapat dilihat alur pengujian/penelitian pada aplikasi *orange* yang akan tampak pada Gambar 4.23 di bawah ini.



Gambar 4. 25 Alur Pengujian *Ensemble Stacking* Dengan *Class Balancing*

Langkah berikutnya adalah melihat performa tiap model *classifier* menggunakan *widget Test and Score*, untuk nilai tiap model akan ditunjukkan pada Tabel 4.4 :

Tabel 4. 4 Performa Stacking dengan Class Balancing

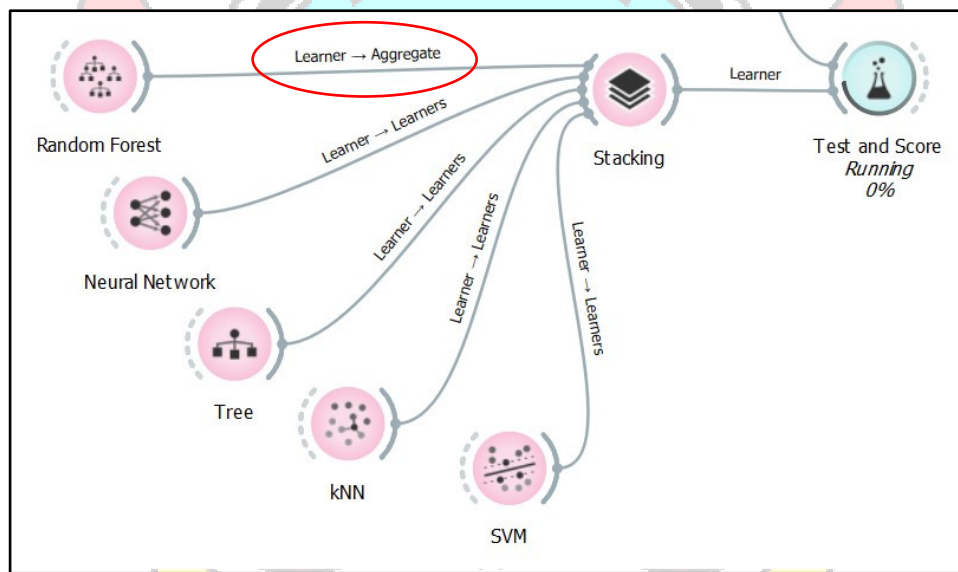
Model	AUC	F1	<i>Precision</i>	<i>Recall</i>	<i>Specifity</i>	Akurasi (CA)
<i>Stacking</i> dengan <i>Class Balancing</i>	0.992	0.961	0.962	0.961	0.962	0.961

Pada Tabel 4.4 menunjukkan bahwa nilai rata-rata akurasi dari penggunaan metode *Ensemble Stacking* menyentuh nilai akurasi yang sangat bagus yakni 90%.

4.3.4 Menggunakan Metode *Ensemble Stacking Class Balancing* Dengan Masing-Masing Algoritma Sebagai *Aggregate*

Pengujian yang ke empat ini hampir sama persis dengan alur pengujian dari *Ensemble Stacking* dengan *Class Balancing*, hanya saja yang membedakan ada pada bagian penggunaan algoritma nya. Tahap pengujian *Aggregate* ini dilakukan pada masing-masing algoritma yang digunakan. Fungsi *Aggregate* pada pengujian ini adalah menggabungkan model *input* yang ada pada algoritma yang dijadikan sebagai *Aggregate* itu sendiri.

Pada gambar di bawah bisa dilihat alur pengujian dari kelima algoritma yang digunakan.

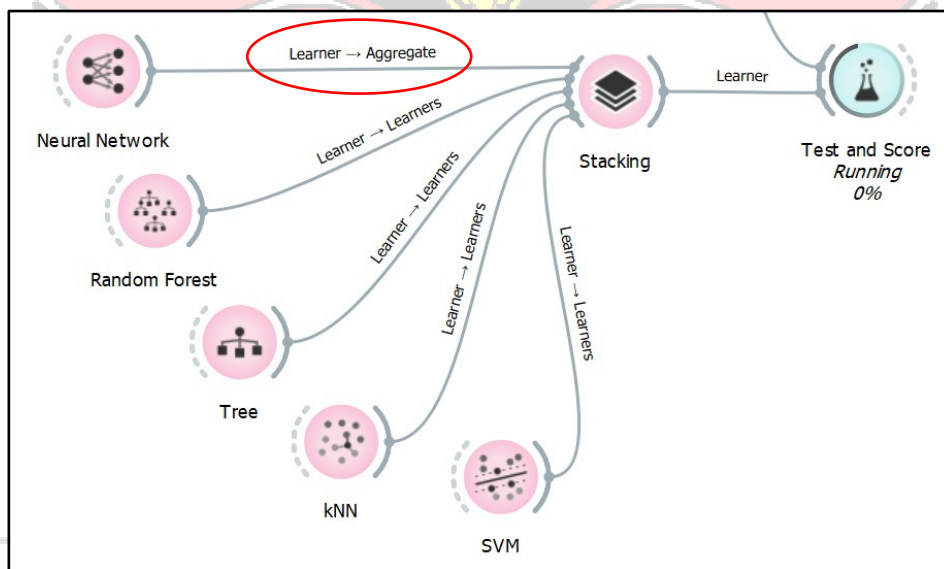


Gambar 4. 26 Alur Pengujian Algoritma *Random Forest* Sebagai *Aggregate*

Tabel 4. 5 Performa Model *Stacking Class Balancing* Dengan *Random Forest* Sebagai *Aggregate*

Model	AUC	F1	Precision	Recall	Specifity	Akurasi (CA)
<i>Stacking Class Balancing Dengan Random Forest Sebagai Aggregate</i>	0.970	0.944	0.945	0.944	0.945	0.944

Pada Tabel 4.5 di atas menunjukkan nilai rata-rata mencapai nilai 90%, namun jika dibandingkan dengan nilai akurasi dari pengujian *Ensemble Stacking Class Balancing* yang terdapat pada tabel 4.4 maka dapat dilihat bahwa nilai yang dihasilkan bisa terhitung sangat sempurna yang dimana nilainya mencapai 96% dan bahkan nilai AUC nya mencapai 99%.

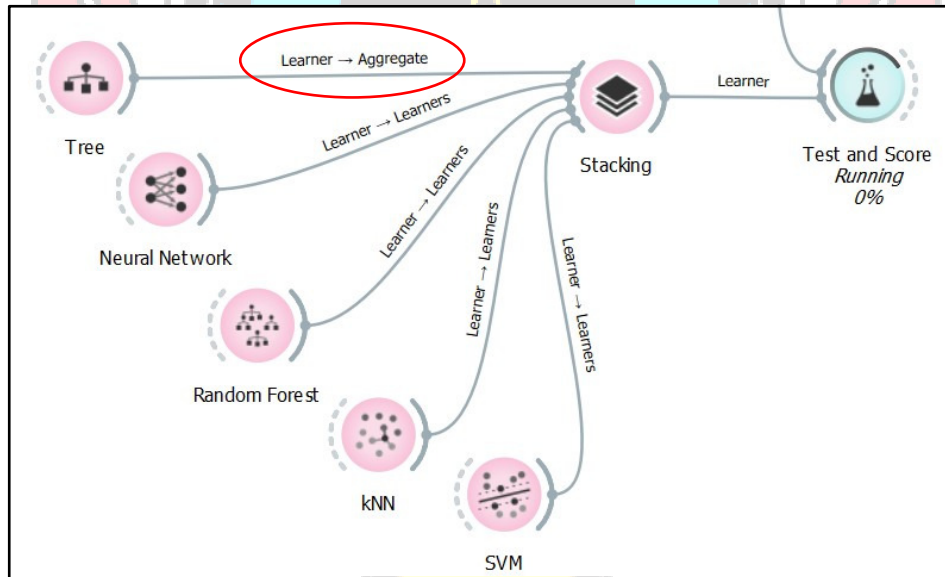


Gambar 4. 27 Alur Pengujian Algoritma *Neural Network* Sebagai *Aggregate*

Tabel 4. 6 Performa Model *Stacking Class Balancing* Dengan *Neural Network* Sebagai *Aggregate*

Model	AUC	F1	Precision	Recall	Specifity	Akurasi (CA)
<i>Stacking Class Balancing Dengan Neural Network Sebagai Aggregate</i>	0.944	0.894	0.909	0.895	0.898	0.895

Pada Tabel 4.6 di atas menunjukkan nilai rata-rata mencapai nilai 80%. Terdapat dua nilai tertinggi yang didapat pada parameter AUC yang nilainya sebesar 94.4% dan parameter *Precision* dengan nilai sebesar 90.9%.

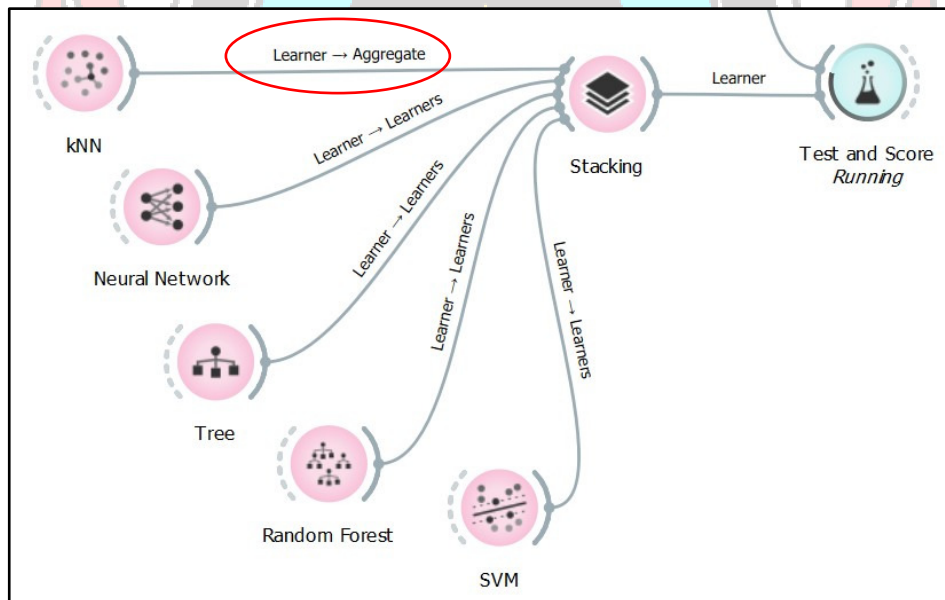


Gambar 4. 28 Alur Pengujian Algoritma *Decision Tree* Sebagai *Aggregate*

Tabel 4. 7 Performa Model *Stacking Class Balancing* Dengan *Decision Tree* Sebagai *Aggregate*

Model	AUC	F1	Precision	Recall	Specifity	Akurasi (CA)
<i>Stacking Class Balancing</i> Dengan <i>Decision Tree</i> Sebagai <i>Aggregate</i>	0.950	0.956	0.956	0.956	0.956	0.956

Pada Tabel 4.7 di atas menunjukkan nilai rata-ratanya mencapai 90%, bahkan rata-rata nilai *Decision Tree* yang dijadikan *Aggregate* mampu mengalahkan nilai rata-rata *Neural Network* yang dijadikan *Aggregate* hanya mampu mencapai 80%.

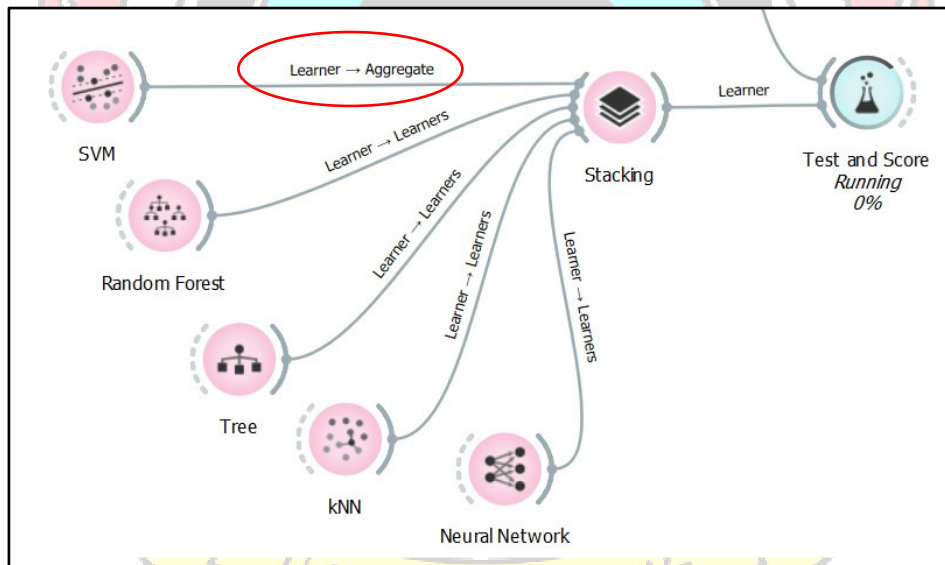


Gambar 4. 29 Alur Pengujian Algoritma *kNN* Sebagai *Aggregate*

Tabel 4. 8 Performa Model *Stacking Class Balancing* Dengan *kNN* Sebagai *Aggregate*

Model	AUC	F1	Precision	Recall	Specifity	Akurasi (CA)
<i>Stacking Class Balancing</i> Dengan <i>kNN</i> Sebagai <i>Aggregate</i>	0.982	0.960	0.961	0.960	0.961	0.960

Pada Tabel 4.8 di atas menunjukkan nilai rata-ratanya mencapai 90% dan bahkan nilai dari *kNN Aggregate* ini berada di atas dari nilai yang dihasilkan oleh *Decision Tree* sebagai *Aggregate*.



Gambar 4. 30 Alur Pengujian Algoritma SVM Sebagai *Aggregate*

Tabel 4. 9 Performa Model *Stacking Class Balancing* Dengan SVM Sebagai *Aggregate*

Model	AUC	F1	Precision	Recall	Specifity	Akurasi (CA)
<i>Stacking Class Balancing</i>	0.397	0.935	0.937	0.935	0.936	0.935

Dengan SVM						
Sebagai						
Aggregate						

Pada Tabel 4.9 di atas menunjukkan nilai rata-ratanya berkisar di 90%, Algoritma SVM sebagai *Aggregate* ini mendapatkan nilai yang lumayan bagus dibanding dengan algoritma *Neural Network* yang digunakan sebagai *Aggregate*.

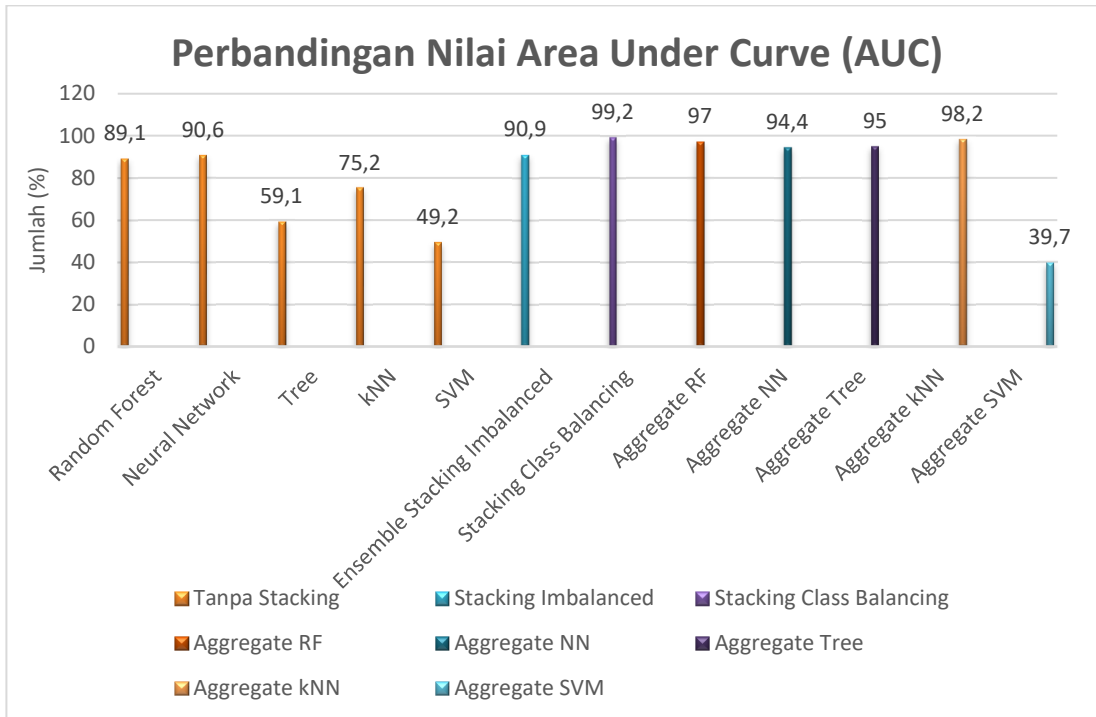
4.4. Analisis Hasil

Hasil proses pembuatan *instance* sintesis dengan metode *Ensemble Stacking* digabungkan dengan dataset asli menghasilkan dataset baru yang telah diklasifikasikan. Untuk mengetahui kinerja dari model *classifier* metode hasil penelitian ini digunakan lima *classifier*, yaitu *Random Forest*, *Neural Network*, *Decision Tree*, kNN, SVM, dan. Hasil dataset dibandingkan dengan berbagai parameter performa model, yaitu *area under curve* (AUC), *F1-score*, *precision*, *specifity*, akurasi.

Selanjutnya adalah melihat performa dalam diagram batang yang dihasilkan oleh *Aggregate* pada *Stacking Class Balancing* ini. Untuk nilai performanya dalam diagram batang dapat dilihat pada gambar 4.30 sampai gambar 4.35

4.4.1 Area Under Curve (AUC)

Setelah dilakukan uji performa masing-masing model algoritma klasifikasi, nilai AUC menunjukkan beberapa perbedaan presentase tanpa metode *Ensemble Stacking*, dengan metode *Ensemble Stacking Imbalanced Data*, menggunakan metode *Ensemble Stacking* dengan *Class Balancing* dan menggunakan *Stacking Balancing* dengan masing-masing algoritma sebagai *Aggregate*. Berikut hasil penilaian model berdasarkan AUC yang ditampilkan dalam bentuk diagram batang :

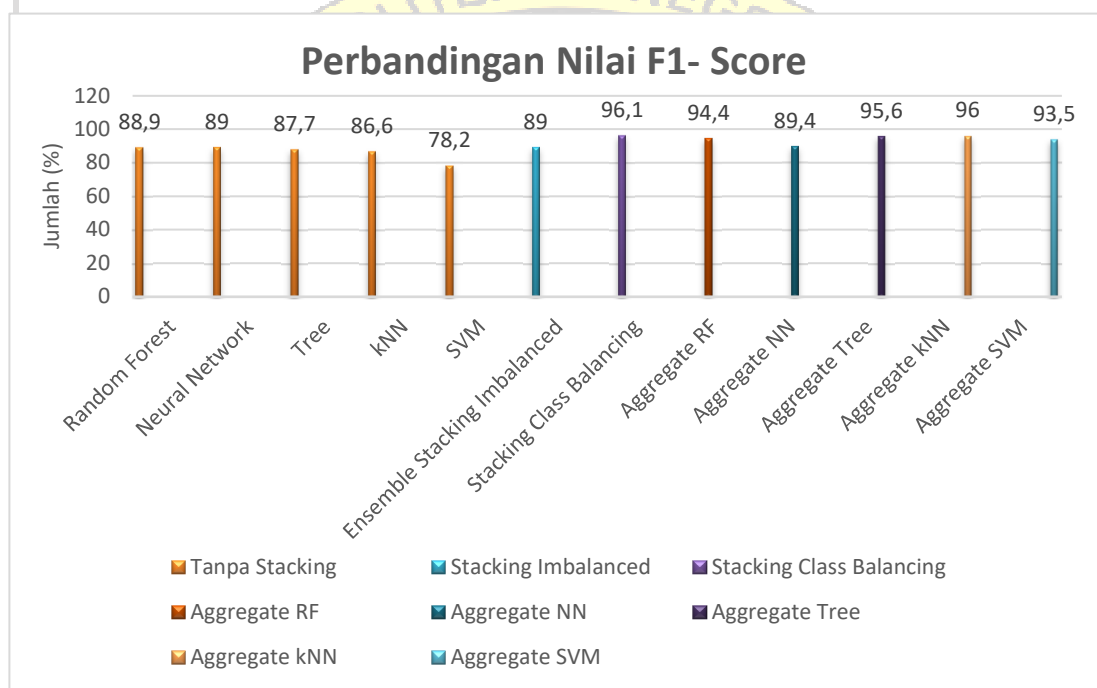


Gambar 4. 31 Diagram Batang Nilai *Area Under Curve* (AUC)

Pada gambar di atas bisa dilihat bahwa nilai AUC pada model klasifikasi menggunakan metode *Ensemble Stacking* dengan *Class Balancing* menunjukkan nilai presentase yang sangat bagus yakni 99,2%. Untuk AUC yang memiliki nilai terbesar kedua setelah *Ensemble Stacking* dengan *Class Balancing* terjadi pada model *Stacking Balancing* dengan *Aggregate kNN* dengan nilai sebesar 98,2%. Sementara itu, untuk nilai yang terendah atau tidak terlalu signifikan terjadi pada model *Stacking Balancing* dengan *Aggregate SVM* yaitu 39,7%.

4.4.2 F1-Score

Setelah dilakukan uji performa masing-masing model algoritma klasifikasi, nilai *F1-Score* menunjukkan beberapa perbedaan presentase tanpa metode *Ensemble Stacking*, dengan metode *Ensemble Stacking Imbalanced Data*, menggunakan metode *Ensemble Stacking* dengan *Class Balancing* dan menggunakan *Stacking Balancing* dengan masing-masing algoritma sebagai *Aggregate*. Berikut hasil penilaian model berdasarkan *F1-Score* yang ditampilkan dalam bentuk diagram batang :

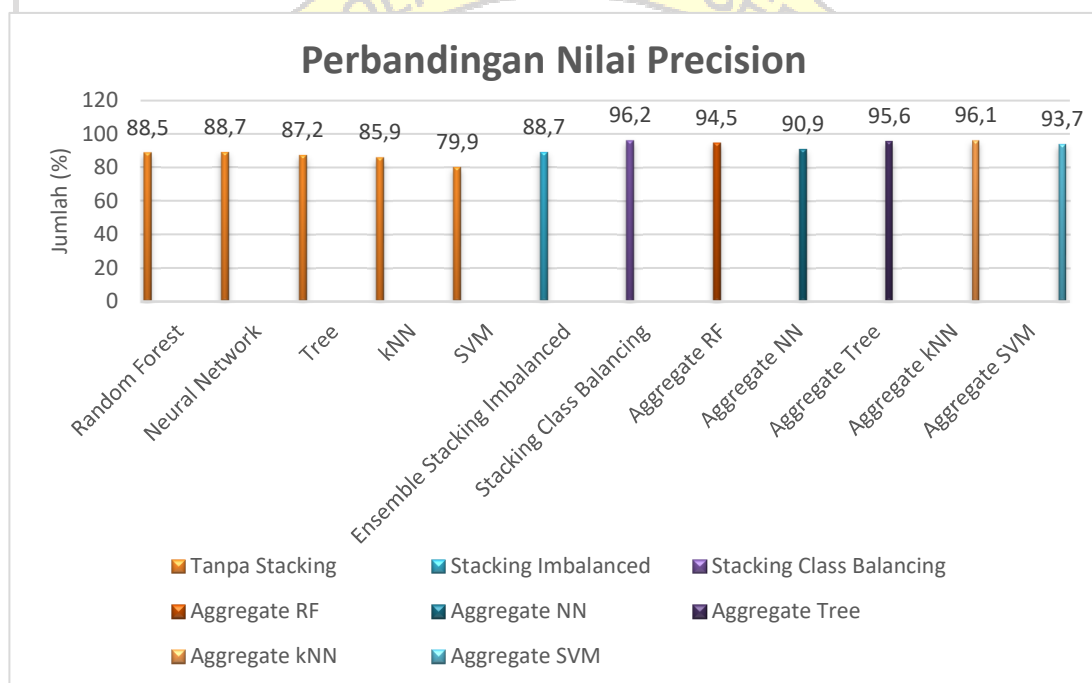


Gambar 4. 32 Diagram Batang Nilai F1- Score

Parameter *F1-Score* merupakan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. Pada gambar 4,31 di atas bisa dilihat bahwa model *Ensemble Stacking* dengan *Class Balancing* memiliki nilai yang hampir menyentuh nilai sempurna, yakni 96,1%. Nilai presentase F1 terbesar kedua terjadi pada *Stacking Balancing* dengan *Aggregate* kNN yakni 96%. Sedangkan untuk nilai presentase terendah terjadi pada model klasifikasi tanpa *Stacking* SVM, dengan nilai ketiganya sebesar 78.2%.

4.4.3 Precision

Setelah dilakukan uji performa masing-masing model algoritma klasifikasi, nilai *Precision* menunjukkan beberapa perbedaan presentase tanpa metode *Ensemble Stacking*, dengan metode *Ensemble Stacking Imbalanced Data*, menggunakan metode *Ensemble Stacking* dengan *Class Balancing* dan menggunakan *Stacking Balancing* dengan masing-masing algoritma sebagai *Aggregate* . Berikut hasil penilaian model *Precision* yang ditampilkan dalam bentuk diagram batang :



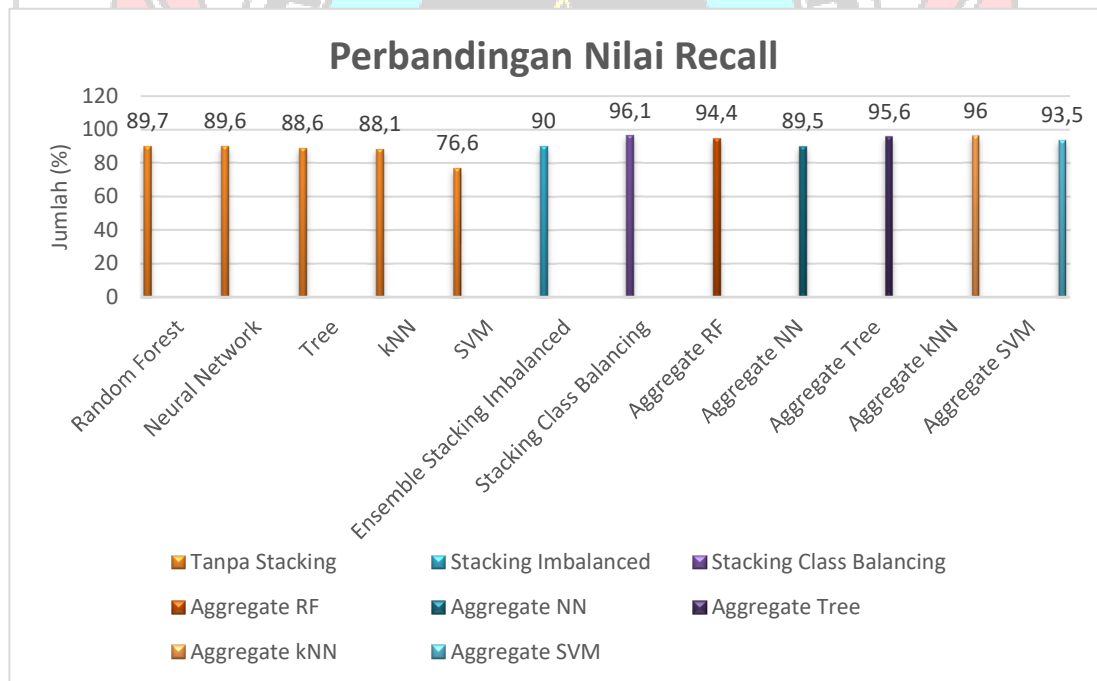
Gambar 4. 33 Diagram Batang Nilai *Precision*

Parameter *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Pada gambar 4.32 di atas bisa dilihat bahwa satu dari lima nilai *precision* model klasifikasi tanpa menggunakan metode *Ensemble Stacking* menunjukkan nilai persentase terendah yang terjadi pada klasifikasi model tanpa *Stacking SVM*, yakni 79.9%. Dan nilai tertinggi tetap di raih metode *Ensemble Stacking* dengan *Class Balancing* dengan nilai sebesar 96,2%. Kemudian disusul oleh model *Stacking Balancing* dengan *Aggregate kNN* sebesar 96.1%

Jumlah nilai *precision* yang tinggi diperlukan untuk data yang ingin terjadinya jumlah *true positif* (TP) pada data tinggi. Sementara untuk penelitian ini lebih diperlukan nilai *false positif* (FP) tinggi yaitu label data yang terindikasi *fraud*. Data yang tidak berlabel *fraud* yakni nilai *true positif* (TP) diperlukan juga pada penelitian ini, namun tidak memiliki urgensi yang lebih mengacu pada tujuan penelitian.

4.4.4 Recall

Setelah dilakukan uji performa masing-masing model algoritma klasifikasi, nilai *Recall* menunjukkan beberapa perbedaan presentase tanpa metode *Ensemble Stacking*, dengan metode *Ensemble Stacking Imbalanced Data*, menggunakan metode *Ensemble Stacking* dengan *Class Balancing* dan menggunakan *Stacking Balancing* dengan masing-masing algoritma sebagai *Aggregate*. Berikut hasil penilaian model *Recall* yang ditampilkan dalam bentuk diagram batang :



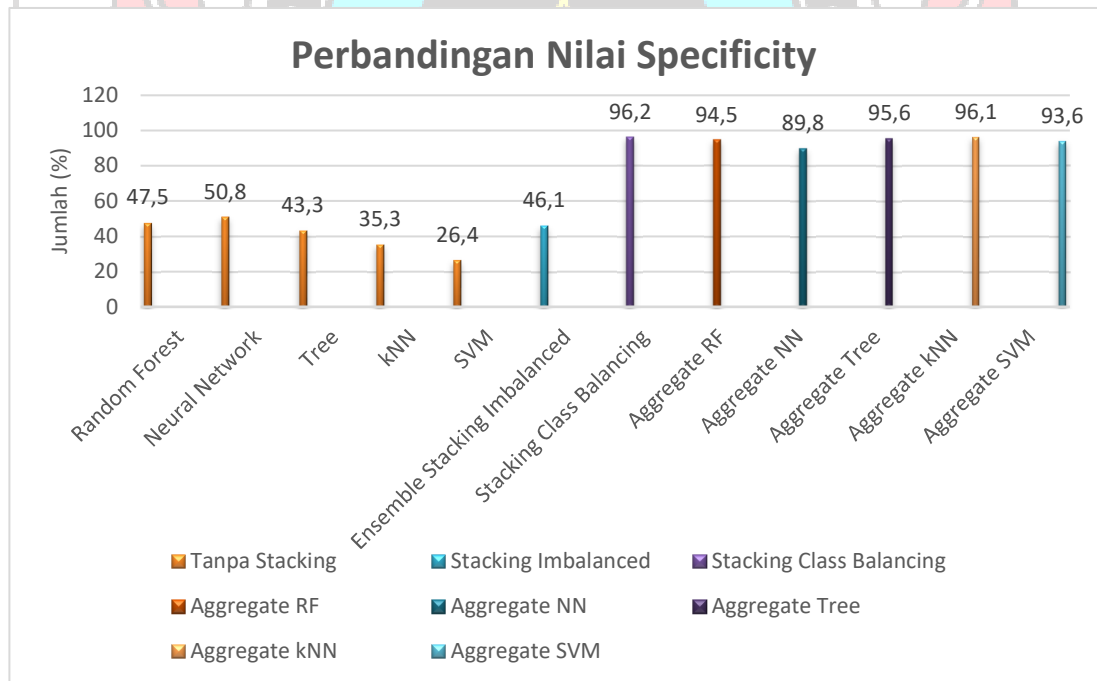
Gambar 4. 34 Diagram Batang Nilai *Recall*

Parameter *recall* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif. Pada gambar 4.33 di atas bisa dilihat bahwa

empat dari lima nilai *Recall* model klasifikasi tanpa metode *Ensemble Stacking* menunjukkan nilai presentase di atas 80%, hanya satu model saja yang memiliki nilai *recall* yang turun yaitu SVM, dengan presentase nilai *recall* sebesar 76,6%. Peningkatan nilai *recall* terbesar terjadi pada model *Ensemble Stacking* dengan *Class Balancing* dengan peningkatan sebesar 96,1%.

4.4.5 Specificity

Setelah dilakukan uji performa masing-masing model algoritma klasifikasi, nilai *Specificity* menunjukkan beberapa perbedaan presentase tanpa metode *Ensemble Stacking*, dengan metode *Ensemble Stacking Imbalanced Data*, menggunakan metode *Ensemble Stacking* dengan *Class Balancing* dan menggunakan *Stacking Balancing* dengan masing-masing algoritma sebagai *Aggregate*. Berikut hasil penilaian model *Specificity* yang ditampilkan dalam bentuk diagram batang :



Gambar 4. 35 Diagram batang Nilai *Specificity*

Parameter *specificity* ini *urgent* disaat kebutuhan model yang diinginkan adalah model yang nilai *False Positive* nya cukup tinggi dibanding nilai *True Negative* (TN), dan pada kasus penelitian ini parameter *specificity* dibutuhkan untuk nilai *False*

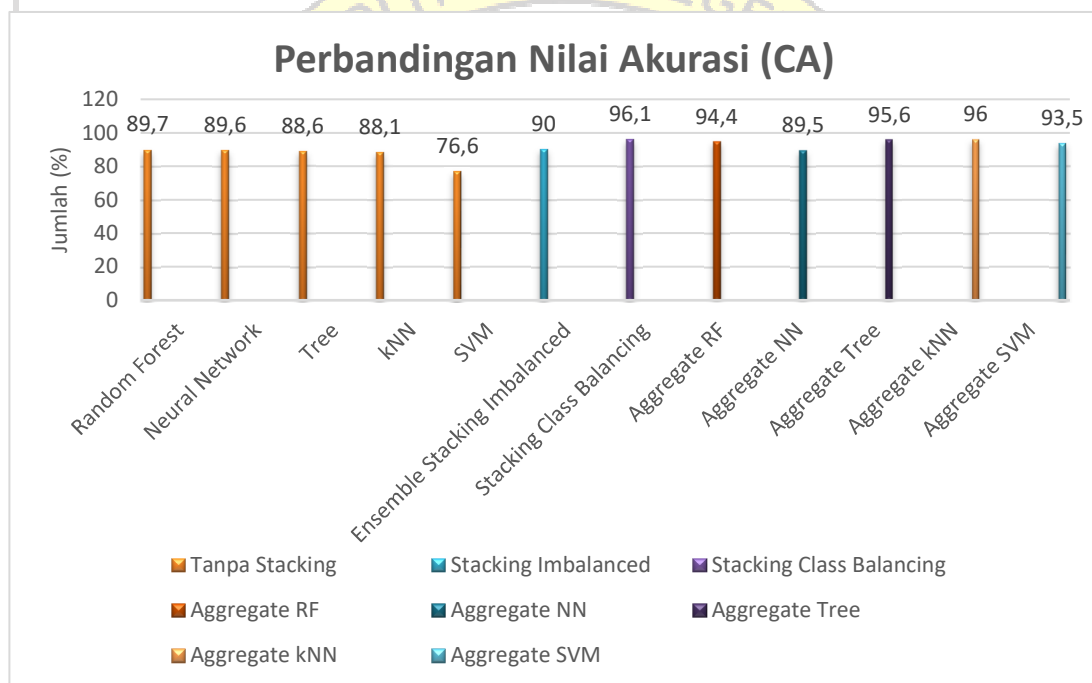
Positive (FP) yang rendah/kecil.

Fungsi dari parameter *specificity* ini sendiri yakni untuk mengidentifikasi sebanyak mungkin kasus negatif yang sebenarnya negatif (misalnya, dalam pemeriksaan keamanan transaksi, prediksi orang yang terjangkit penyakit, dll). Pada gambar 4.34 di atas bisa dilihat bahwa peningkatan nilai *Specificity* terjadi pada model yang menggunakan *Ensemble Stacking* dengan *Class Balancing* yakni sebesar 96,2%. Sedangkan nilai terendah terdapat pada model klasifikasi SVM tanpa *Stacking* yakni sebesar 26,4%.

Setelah melihat hasil dari parameter *Specificity* ini, saya menemukan hasil bahwa *Ensemble Stacking* dengan *Class Balancing* mampu memberikan perubahan hasil yang cukup meningkat dalam parameter *Specificity* ini. Penyebab terjadinya perubahan nilai yang sangat signifikan dalam penggunaan *ensemble stacking* ini terdapat pada nilai *True Negative* (TN) yang lebih rendah dibanding dengan nilai *True Negative* (TN) dari kelima model algoritma *machine learning* tanpa *stacking* dan dengan *stacking* saja. Peningkatan nilai *specifity* yang tinggi berpengaruh pada prediksi kasus *fraud* yang diharapkan tidak terjadi *False Positive* (FP) yang besar, sehingga algoritma bisa mengklasifikasikan atau memprediksi kasus yang benar-benar terjadi *fraud*.

4.4.6 Akurasi (CA)

Setelah dilakukan uji performa masing-masing model algoritma klasifikasi, nilai Akurasi menunjukkan beberapa perbedaan presentase tanpa metode *Ensemble Stacking*, dengan metode *Ensemble Stacking Imbalanced Data*, menggunakan metode *Ensemble Stacking* dengan *Class Balancing* dan menggunakan *Stacking Balancing* dengan masing-masing algoritma sebagai *Aggregate* . Berikut hasil penilaian model Akurasi yang ditampilkan dalam bentuk diagram batang :



Gambar 4. 36 Diagram batang Nilai Akurasi

Parameter akurasi merupakan rasio prediksi Benar (positif dan negatif) dengan keseluruhan data. Pada gambar 4.36 di atas bisa dilihat bahwa nilai akurasi dari keseluruhan model klasifikasi yang digunakan pada penelitian ini memiliki nilai akurasi yang lebih baik setelah menggunakan metode *Ensemble Stacking* dengan *Class Balancing*. Bahkan penggunaan metode *Ensemble Stacking* dengan *Class Balancing* mampu mendapatkan nilai yang sangat baik, yakni sebesar 96,1%. Angka tersebut membuat rata-rata nilai akurasi semakin baik sesuai yang diharapkan pada tujuan penelitian di awal. Kemudian nilai tertinggi kedua di susul oleh metode

Stacking Balancing dengan *Aggregate kNN* dengan nilai sebesar 96%.

4.4.7 Durasi Pengujian

Jumlah durasi yang dibutuhkan masing-masing model tanpa menggunakan metode *Ensemble Stacking* untuk melakukan proses *training* dan *testing data* terdapat pada Tabel 4.10

Tabel 4. 10 Durasi Pengujian Tanpa *Ensemble Stacking*

Model	Train Time (s)	Test Time (s)
<i>Random Forest</i>	19.560	4.507
<i>Neural Network</i>	3923.642	4.968
<i>Tree</i>	184.814	0.043
kNN	14.001	160.431
SVM	44.278	6.330

Jumlah durasi yang dibutuhkan masing-masing model dengan menggunakan metode *Ensemble Stacking Imbalanced Data* untuk melakukan proses *training* dan *testing data* terdapat pada Tabel 4.11

Tabel 4. 11 Durasi Pengujian Menggunakan *Ensemble Stacking Imbalanced Data*

Model	Train Time (s)	Test Time (s)
<i>Ensemble Stacking Imbalanced Data</i>	2423.666	41.156

Sementara itu, jumlah durasi yang dibutuhkan masing-masing model dengan menggunakan metode *Ensemble Stacking* dengan *Class Balancing* untuk melakukan proses *training* dan *testing data* terdapat pada Tabel 4.12

Tabel 4. 12 Durasi Pengujian Menggunakan *Ensemble Stacking dengan Class Balancing*

Model	Train Time (s)	Test Time (s)
<i>Ensemble Stacking dengan Class Balancing</i>	2284.952	100.664

Sementara itu, jumlah durasi yang dibutuhkan masing-masing model dengan menggunakan metode *Ensemble Stacking* dengan *Class Balancing* untuk melakukan proses *training* dan *testing data* terdapat pada Tabel 4.13

Tabel 4. 13 Durasi Pengujian Menggunakan *Ensemble Stacking Class Balancing* dengan *Aggregate*

Model	Train Time (s)	Test Time (s)
<i>Aggregate Random Forest</i>	14447.690	140.475
<i>Aggregate Neural Network</i>	5711.074	271.620
<i>Aggregate Decision Tree</i>	14096.394	146.786
<i>Aggregate kNN</i>	14286.903	27.285
<i>Aggregate SVM</i>	14460.408	138.486

4.4.8 Perbandingan Metode Ensemble Stacking Imbalanced Data, Ensemble Stacking Class Balancing dan Metode Smote

Pada tahap kali ini akan dilakukan perbandingan hasil manakah yang lebih akurat dalam mendeteksi kasus *fraud* dalam *dataset* yang semula *imbalanced* kemudian dijadikan *balance* antara metode *Ensemble Stacking Class Balancing* dengan Metode *Smote*. Perbandingan tersebut dapat dilihat pada tabel di bawah ini :

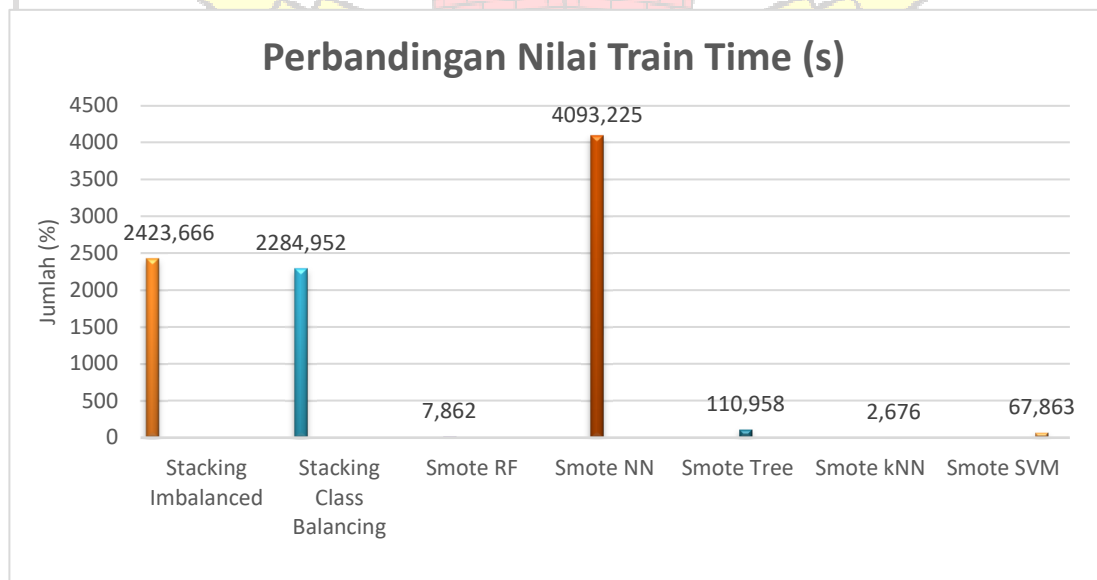
Tabel 4. 14 Perbandingan *Ensemble Stacking Imbalance Data, Ensemble Stacking Class Balancing* dan Metode *Smote*

	Model	AUC	F1	Precision	Recall	Specificity	Akurasi (CA)
<i>Imbalanced Data</i>	<i>Stacking Imbalanced Data</i>	0.909	0.890	0.887	0.900	0.461	0.900
<i>Stacking Balancing</i>	<i>Ensemble Stacking dengan Class</i>	0.992	0.961	0.962	0.961	0.962	0.961

	<i>Balancing</i>						
<i>Smote</i>	<i>Random Forest</i>	0.984	0.942	0.944	0.942	0.942	0.942
	<i>Neural Network</i>	0.942	0.885	0.887	0.885	0.885	0.885
	<i>Decision Tree</i>	0.926	0.919	0.920	0.920	0.920	0.920
	kNN	0.942	0.880	0.885	0.880	0.880	0.880
	SVM	0.537	0.510	0.511	0.511	0.511	0.511

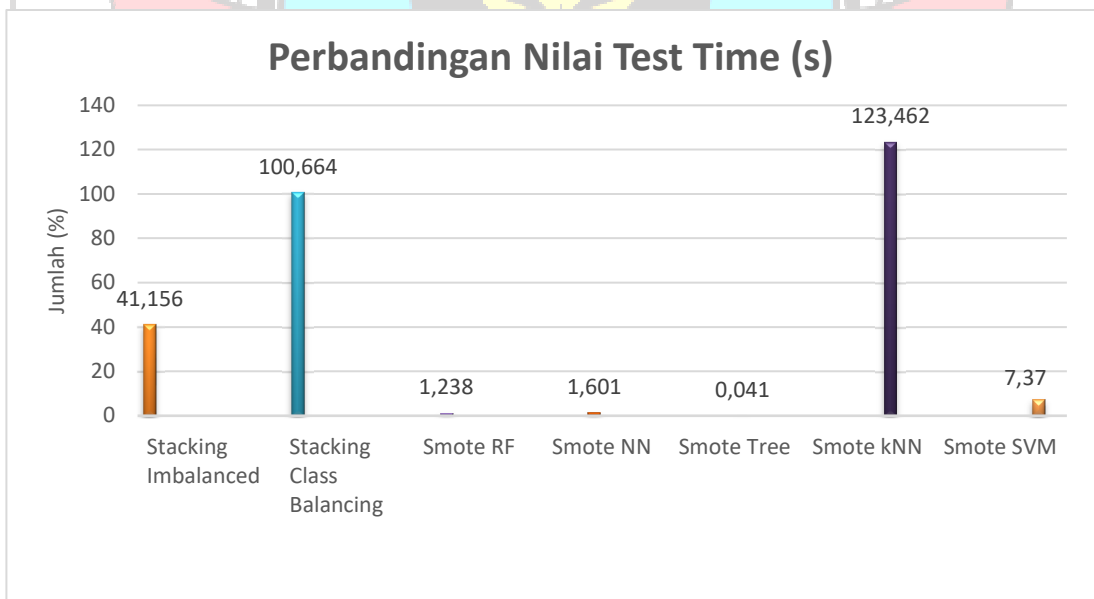
Pada tabel di atas sudah dapat terlihat hasil yang sangat jelas, yang dimana metode *Ensemble Stacking Class Balancing* mampu mendapatkan hasil yang lebih akurat. Hal itu dapat dilihat dari semua parameter mulai dari nilai AUC sampai nilai akurasi *Ensemble Stacking Class Balancing* lebih bagus dibandingkan dengan nilai yang didapat pada metode *Smote*. Nilai rata-rata yang didapatkan pada metode *Ensemble Stacking* dengan *Class Balancing* di atas 90%, bahkan nilai AUC nya hampir mendekati nilai yang sempurna.

Sementara itu, jumlah durasi yang dibutuhkan masing-masing model dengan menggunakan metode *Ensemble Stacking imbalanced Data*, *Ensemble Stacking Class Balancing* dan *Smote* untuk melakukan proses *training* dan *testing data* terdapat pada gambar 4. 37



Gambar 4. 37 Perbandingan nilai *Train Time*

Bisa dilihat pada gambar diagram diatas nilai *train time* tertinggi di dapat pada *Smote NN* yang berarti metode *Smote NN* memerlukan waktu yang cukup lama untuk mengolah data *training*. Sedangkan nilai *train time* tercepat didapatkan pada metode *smote kNN* sebesar 2.676 detik. Namun jika di bandingkan antara kecepatan mengolah data *training* antara metode *Stacking imbalanced* dan *Stacking balancing*, maka diperoleh hasil metode *stacking balancing* lebih cepat di banding dengan *stacking imbalanced*. Dilihat dari jumlah data training yang diolah pada metode *Stacking Balancing* terdapat perbedaan jumlah data yang sangat besar dibanding dengan *Stacking Imbalanced* yang dimana jumlah data pada *Stacking Balancing* sebesar 63.058 dan jumlah data pada *Stacking Imbalanced* sebesar 36.169. Perbedaan jumlah ini memberikan suatu dampak/penilaian yang berarti metode *Stacking Balancing* ini lebih unggul dibanding *Stacking Imbalanced* dalam hal kecepatan mengolah suatu data.



Gambar 4. 38 Perbandingan nilai *Test time*

Bisa dilihat pada gambar diatas, metode *Smote kNN* memerlukan waktu yang cukup lama untuk mengola data testing yang dimana pembagian data *testing* ini sebesar 20%. Nilai tercepat diperoleh oleh metode *Smote Tree* dikarenakan metode

ini hanya melakukan 1 pengujian algoritma. Dibandingkan dengan metode antara *Stacking Balancing* dan *Stacking Imbalanced* maka didapatkan hasil nilai *test time Stacking Imbalanced* lebih cepat memproses data testing dibanding *Stacking Balancing*. Salah satu faktor yang mempengaruhi proses pengolahan data *testing* pada *Stacking Balancing* adalah banyaknya data yang diolah sebesar 15.764 data, sedangkan jumlah data yang diolah dalam *Stacking Imbalanced* hanya sebesar 9.042 data. Perbedaan jumlah data tersebut tentunya berpengaruh terhadap kecepatan suatu metode dalam mengolah data.

4.4.9 Perbandingan performa model tanpa *Ensemble Stacking* dan Metode *Ensemble Stacking* dengan data normal

Pada tahap kali ini akan dilakukan perbandingan hasil manakah yang lebih akurat dalam mendeteksi kasus *fraud* dalam *dataset* yang normal antara model tanpa *Ensemble Stacking* dengan metode *Ensemble Stacking* data normal. Perbandingan tersebut dapat dilihat pada tabel di bawah ini :

Tabel 4. 15 Performa Model Tanpa Metode *Ensemble Stacking*

Model	AUC	F1	Precision	Recall	Specificity	Akurasi (CA)
<i>Random Forest</i>	0.891	0.889	0.885	0.897	0.475	0.897
<i>Neural Network</i>	0.906	0.890	0.887	0.896	0.508	0.896
<i>Tree</i>	0.591	0.877	0.872	0.886	0.433	0.886
k-NN	0.752	0.866	0.859	0.881	0.353	0.881
SVM	0.492	0.782	0.799	0.766	0.264	0.766

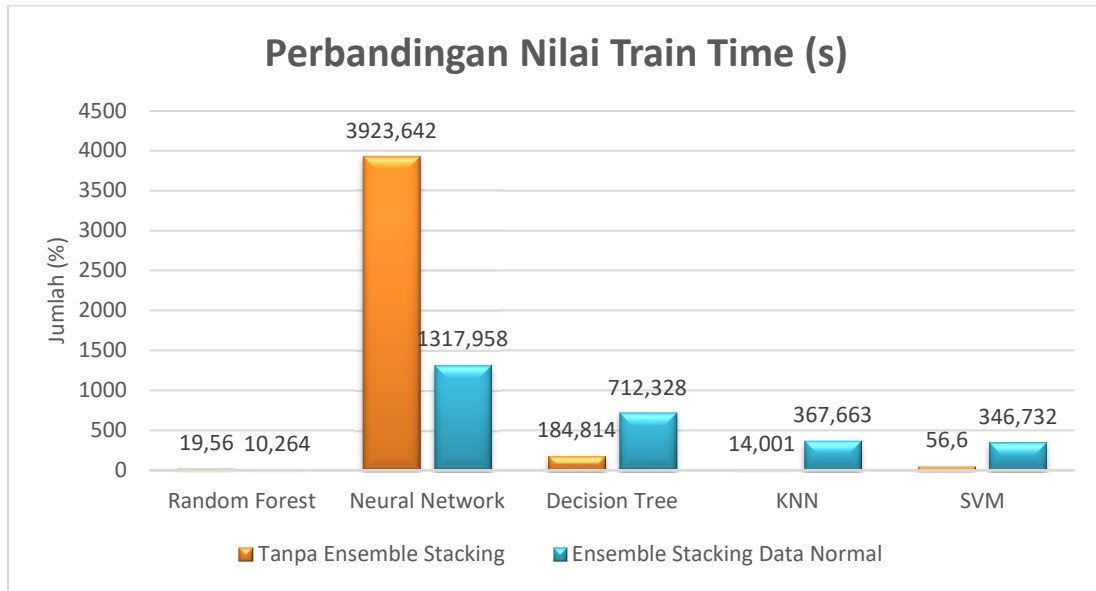
Tabel 4. 16 Performa Metode *Ensemble Stacking* data normal

Model	AUC	F1	Precision	Recall	Specificity	Akurasi
-------	-----	----	-----------	--------	-------------	---------

						(CA)
<i>Random Forest</i>	0.894	0.888	0.885	0.899	0.443	0.899
<i>Neural Network</i>	0.905	0.885	0.882	0.897	0.422	0.897
<i>Tree</i>	0.589	0.875	0.870	0.887	0.397	0.887
k-NN	0.748	0.847	0.847	0.883	0.203	0.883
SVM	0.506	0.828	0.780	0.883	0.117	0.883

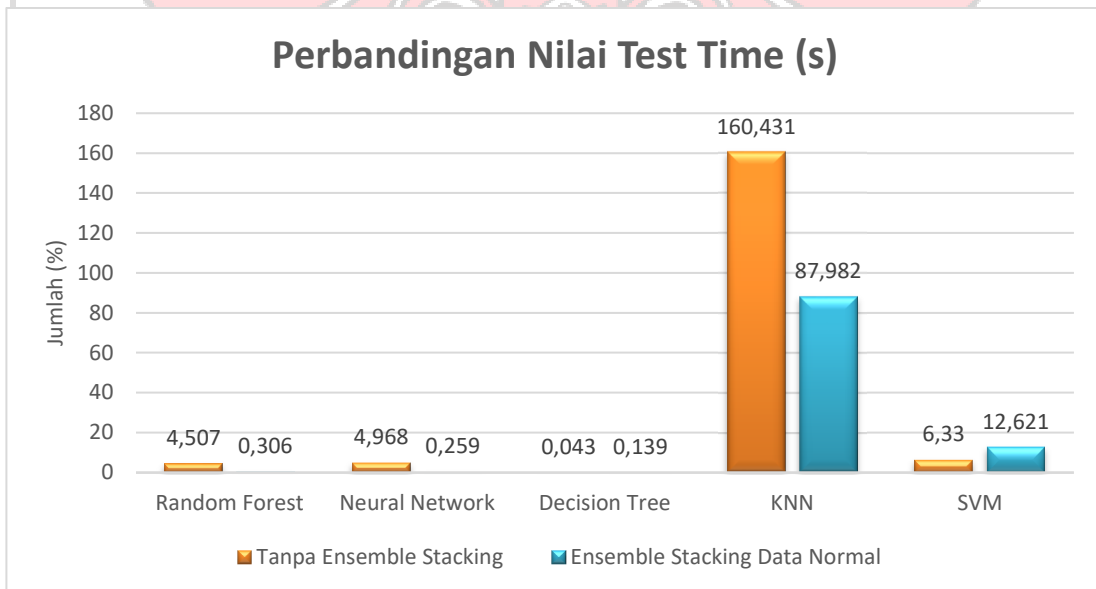
Pada kedua tabel di atas sudah dapat terlihat hasil yang sangat jelas, yang dimana metode *Ensemble Stacking* data normal mampu mendapatkan hasil yang lebih akurat. Hal itu dapat dilihat dari parameter akurasi (CA) lebih bagus dibandingkan dengan nilai yang didapat pada model tanpa *Ensemble Stacking*. Begitu pula dengan parameter *recall* yang juga meningkat dibanding model tanpa *Ensemble Stacking*. Nilai rata-rata yang didapatkan pada metode *Ensemble Stacking* dengan data normal di atas 88%.

Sementara itu, jumlah durasi yang dibutuhkan masing-masing model dengan menggunakan model tanpa *Ensemble Stacking* dan metode *Ensemble Stacking* data normal untuk melakukan proses *training* dan *testing data* terdapat pada gambar 4. 37



Gambar 4. 39 Perbandingan nilai *Train Time*

Bisa dilihat pada gambar diagram diatas nilai *train time* tertinggi di dapat pada *Neural Network* tanpa *Stacking* yang berarti model *Neural Network* tanpa *Stacking* memerlukan waktu yang cukup lama untuk mengolah data *training*. Sedangkan nilai *train time* tercepat didapatkan pada model SVM tanpa *Stacking* sebesar 19,56 detik.



Gambar 4. 40 Perbandingan nilai *Test time*

Bisa dilihat pada gambar diatas, model kNN tanpa *Stacking* memerlukan waktu yang cukup lama untuk mengola data *testing* yang dimana pembagian data *testing* ini sebesar 20%. Nilai tercepat diperoleh oleh model *Tree* tanpa *Stacking* sebesar 0.043 detik. Dari kelima model algoritma di atas metode *Ensemble Stacking* lebih cepat dalam mengolah data *testing*.



BAB V PENUTUP

5.1 Kesimpulan

Berdasarkan hasil penelitian mengenai deteksi *credit card fraud* pada transaksi perbankan yang menggunakan data transaksi dari kartu kredit sebuah bank di Portugal tahun 2013, maka penulis menarik kesimpulan sebagai berikut :

1. Pada penelitian ini telah berhasil diterapkan metode *Ensemble Stacking* dalam meningkatkan akurasi pendeteksian *fraud*. Di penelitian ini terdapat delapan macam pengujian yang dilakukan terhadap data transaksi yang terindikasi adanya *fraud* sebagai bahan perbandingan yaitu :

- 1) Pengujian tanpa *stacking*,
- 2) *Stacking Imbalanced Data*,
- 3) *Stacking* dengan *Class Balancing*,
- 4) *Stacking Class Balancing Aggregate Random Forest*,
- 5) *Stacking Class Balancing Aggregate Neural Network*,
- 6) *Stacking Class Balancing Aggregate Decision Tree*,
- 7) *Stacking Class Balancing Aggregate kNN*,
- 8) *Stacking Class Balancing Aggregate SVM*

Hasil pengujian dengan menggunakan metode *Ensemble Stacking Class Balancing* terbukti mampu meningkatkan performa model yang ditinjau dari parameter penilaian yang diantaranya yaitu AUC, F1-Score, *precision*, *recall*, *specificity*, dan akurasi (CA). Penggunaan metode *Ensemble Stacking* dengan *Class Balancing* secara keseluruhan mampu meningkatkan nilai hasil dari keenam parameter dengan presentase peningkatan nilai rata-rata sebesar 14,2% dibanding penggunaan *Stacking Imbalanced Data*. Nilai presentase dalam metode *Ensemble Stacking Class Balancing* terbukti mampu meningkat sebesar 32,1% dari penggunaan metode tanpa *Stacking*. Peningkatan yang signifikan juga terjadi pada parameter *specificity* dengan nilai rata-rata presentase peningkatan sebesar 60,7%.

Jadi bisa ditarik kesimpulan bahwa penggunaan metode Ensemble Stacking Class Balancing mampu memberikan dampak yang sangat baik terhadap deteksi transaksi *credit card fraud* perbankan.

2. Pada kasus fraud dalam data transaksi perbankan telah diketahui/didapatkan dampak yang cukup signifikan yang dihasilkan setelah menerapkan *Ensemble Stacking* pada data yang *imbalance* dan yang *balance*. Parameter yang menjadi acuan dalam kinerja model setelah dilakukan *oversampling* pada data ialah parameter *area under curve* (AUC) dan *specificity*. Walaupun nilai akurasi yang menjadi faktor utama dalam penilaian kinerja model namun parameter *area under curve* (AUC) dan *specificity* juga sangat berpengaruh terhadap kinerja klasifikasi pada kasus *imbalanced data*. Dalam penelitian ini penerapan *Ensemble Stacking* pada data *balance* berdampak meningkatkan nilai *area under curve* (AUC) dari keseluruhan model dengan nilai rata-rata sebesar 26,6% serta meningkatkan nilai *specificity* dengan rata-rata peningkatan sebesar 60,7%. Dan dari catatan durasi pengujian *Ensemble Stacking Class Balancing* nilai *Train Time* nya lebih cepat 138.718 detik dibanding dengan *Stacking Imbalanced Data*, namun dari catatan *Test Time Stacking Imbalanced Data* lebih cepat 59.508 detik dibanding *Stacking Class Balancing*. Kemampuan dari *Ensemble Stacking* itu sendiri yang mampu melakukan percepatan olah data dengan jumlah data training sebesar 80%, faktor itulah yang mempengaruhi proses pengolahan data yang cepat dalam metode *Stacking Class Balancing* khususnya pada data training ini.

5.2 Saran

Untuk mengembangkan penelitian ini, dapat dilakukan dengan mengeksplor model algoritma *machine learning* lainnya. Pada pemilihan model algoritma *machine learning* ini harus memiliki beberapa acuan kinerja diantaranya yaitu algoritma harus memiliki akurasi performa yang baik, performa *recall* yang baik, performa *precision* yang tinggi, dan performa *specificity* yang tinggi. Penulis juga menyarankan pada

tahap melakukan *random oversampling* ataupun *undersampling* dapat memanfaatkan fitur *widget python script* yang terdapat pada *orange data mining version 3.32.0*.



DAFTAR PUSTAKA

- Andrybrew, 2015, <https://andrybrew.blog/2015/06/14/data-analytics-menggunakan-orange/>
- Arthana, R. (2019). <https://medium.com/@rey1024/mengenal-accuracy-precision-recall-dan-specificity-serta-yang-diprioritaskan-b79ff4d77de8>
- Brownlee, J. (2020). *Data preparation for machine learning: data cleaning, feature selection, and data transforms in Python. Machine Learning Mastery.*
- Binus, 2019. *Imbalanced dataset.* <https://socs.binus.ac.id/2019/12/26/imbalanced-dataset/#:~:text=Kelas%20yang%20tidak%20seimbang%20adalah,deteksi%20penipuan%2C%20klasifikasi%20emosi%20dll>
- Ciputra Universitas. (2019). *Perbedaan Supervised Learning and Unsupervised Learning* Information Communication Technology. <https://www.uc.ac.id/ict/perbedaan-supervised-learning-and-unsupervised-learning/>
- Cloud, E. I. (2020a). (2020a). *What is Supervised Learning?* | IBM. <https://www.ibm.com/cloud/learn/supervised-learning>
- Cloud, E. I. (2020b). *What is Unsupervised Learning?* | IBM. <https://www.ibm.com/cloud/learn/unsupervised-learning>
- Cortes, C., & Vapnik, V. (1995). *Support-vector networks. Machine learning*, 20(3), 273-297.
- Dana Nasabah Hilang di Maybank, Bos LPS: Itu Kasus Fraud! (n.d.). Retrieved June 10, 2021, from <https://www.cnbcindonesia.com/market/20201110145127-17-200752/dana-nasabah-hilang-di-maybank-bos-lps-itu-kasus-fraud>
- Dolo, K. M. (2019). *Differential evolution technique on weighted voting stacking ensemble method for credit card fraud detection* (Doctoral dissertation).
- Donny B.U. 2002. *Cyberfraud Indonesia Menguartirkan.* The Jakarta Post, 22 Juli

2002. (<http://bebas.vlsm.org/v17/com/ictwatch/paper/paper007.htm>, diakses 21 Oktober 2013)

Donny B.U. 2004. *Cyberfraud: Pertaruhan Citra Indonesia di Industri e-Commerce Global*. Majalah Warta Ekonomi, edisi No.15, tahun XVI, 28 Juli 2004

Jakarta Post, 22 Juli 2002. (<http://bebas.vlsm.org/v17/com/ictwatch/paper/paper007.htm>, diakses 21 Oktober 2013)

Fitriansyah, R. A. (2017, February). Penerapan *Ensemble Stacking* Untuk Klasifikasi Multi Kelas. In *Annual Research Seminar (ARS)* (Vol. 2, No. 1, pp. 240-243).

Ghoneim Salma, 2019, Accuracy, Recall, Precision, F-Score & Specificity, which to optimize on?

Hermawati, F. A. 2013. *Data Mining*. Yogyakarta: Penerbit Andi.

Indradi, Ade Ary Syam. 2006. *Carding: Modus Operasi, Penyelidikan dan Penindakan*. Jakarta: Pensil-324

Iriadi, N., & Leidiyana, H. (2013). Prediksi Pinjaman Kredit dengan Support Vector Machine dan K-nearest Neighbors pada Koperasi Serba USAha. *PIKSEL: Penelitian Ilmu Komputer Sistem Embedded and Logic*, 1(2), 115-124.

Julianna, D. (2021). *Supervised vs. Unsupervised Learning: What's the Difference?* | IBM. 74 <https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning>

Karyono 2013. *Forensic fraud*. Yogyakarta: CV Andi Offset

Kumar, M., Singh, S., & Rath, S. K. (2015). Classification of microarray data using functional link neural network. *Procedia Computer Science*, 57, 727-737.

Larose, D. T. (2005). *Discovering Knowledge in Data. United States of America: John Wiley & Sons, Inc.*

Moro, S., Rita, P., and Cortez, P.. (2012). Bank Marketing. UCI Machine Learning Repository. <https://doi.org/10.24432/C5K306>

Nai-Arun, N., & Moungrmai, R. (2015). Comparison of classifiers for the risk of diabetes prediction. *Procedia Computer Science*, 69, 132-142.

Narendra, P. N. (2021). Perbankan “Digerogoti” *Fraud* dari Dalam. <https://www.hukumonline.com/berita/baca/lt5829aca28a189/perbankan-digerogoti-fraud-dari-dalam/>

Nurhayati, Buslim, I. R. P. (2019). Pengembangan Algoritma *Unsupervised Learning Technique* Pada *Big Data Analysis* di Media Sosial sebagai media promosi *Online* Bagi Masyarakat. *Jurnal Teknik Informatika*, 12(1), 79–96. <https://doi.org/10.15408/jti.v12i1.11342>

Nurmahaludin. (2014). Analisis Perbandingan Metode Jaringan Syaraf Tiruan dan Regresi Linear Berganda pada Prakiraan Cuaca. *Jurnal INTEKNA*, 2, 102–209. <https://ejurnal.poliban.ac.id/index.php/intekna/article/view/389>

Penmetsa, S. D., & Mohammed, S. (2021). *Ensemble Techniques for Credit Card Fraud Detection*.

Purbasari, I. Y., & Nugroho, B. (2013). *Benchmarking* Alogoritma Pemilihan Atribut Pada Klasifikasi *Data Mining*. *Jurnal SNASTIA*.

Romdhoni, H., 2020, <https://www.inanews.co.id/2020/05/kejahatan-perbankan-yang-wajib-diketahui/>

Saddam, H., 2021, <https://geospasialis.com/ensemble-learning/>

Soekardi Husodo, “Faktor-faktor Pemicu Terjadinya *Fraud* Perbankan”, *Makalah* disampaikan pada Seminar Nasional Infobank dalam Membangun Komitmen Pengurus dan Manajemen Bank dalam Penerapan Strategi *anti fraud*, Le Meridien Hotel, 7 Maret 2012.

Soleymanzadeh, R., Aljasim, M., Qadeer, M. W., & Kashef, R. (2022). Cyberattack

and Fraud Detection Using Ensemble Stacking. *AI*, 3(1), 22-36.

Subagyo, *Bank dan Lembaga Keuangan Lainnya*, (Yogyakarta, Bagian Penerbitan Sekolah Tinggi Ilmu Ekonomi YKPN, 2005), hlm.39.

Veigas, K. C., Regulagadda, D. S., & Kokatnoor, S. A. (2021). Optimized Stacking Ensemble (OSE) for Credit Card Fraud Detection using Synthetic Minority Oversampling Model. *Indian Journal of Science and Technology*, 14(32), 2607-2615.

Watts, R. L., & Zimmerman, J. L. (1986). *Positive accounting theory*.

Wibawa, A. P., Purnama, M. G. A., Akbar, M. F., & Dwiyanto, F. A. (2018). Metode-metode Klasifikasi. *Prosiding Seminar Ilmu Komputer Dan Teknologi Informasi*, 3(1), 134–138.

Zaid Azla, 2014, <https://keuangan.kontan.co.id/news/kartu-kredit-lebih-rentan-dipalsukan>

Zuech, R., Hancock, J., & Khoshgoftaar, T. M. (2021). Detecting web attacks using random undersampling and ensemble learners. *Journal of Big Data*, 8(1), 1-20.

LAMPIRAN

Lampiran 1 tabel *Confusion Matrix* model tanpa menggunakan *Ensemble Stacking*

1. Random Forest

		Predicted		Σ
		no	yes	
Actual	no	92.5 %	41.4 %	31943
	yes	7.5 %	58.6 %	4226
Σ		33211	2958	36169

2. Neural Network

		Predicted		Σ
		no	yes	
Actual	no	92.9 %	43.1 %	31943
	yes	7.1 %	56.9 %	4226
Σ		32835	3334	36169

3. Decision Tree

		Predicted		Σ
		no	yes	
Actual	no	91.9 %	48.3 %	31943
	yes	8.1 %	51.7 %	4226
Σ		33193	2976	36169

4. KNN

		Predicted		Σ
		no	yes	
Actual	no	90.9 %	51.7 %	31943
	yes	9.1 %	48.3 %	4226
Σ		33789	2380	36169

5. SVM

		Predicted		Σ
		no	yes	
Actual	no	88.7 %	86.4 %	31943
	yes	11.3 %	13.6 %	4226
Σ		30360	5809	36169

Lampiran 2 tabel *Confusion Matrix* model menggunakan metode *Ensemble Stacking Imbalanced Data*

		Predicted		Σ
		no	yes	
Actual	no	92.5 %	34.7 %	7979
	yes	7.5 %	65.3 %	1063
Σ		8379	663	9042

Lampiran 3 tabel *Confusion Matrix* model menggunakan metode *Ensemble Stacking* dengan *Class Balancing*

		Predicted		Σ
		no	yes	
Actual	no	98.2 %	5.9 %	32073
	yes	1.8 %	94.1 %	30985
Σ		30690	32368	63058

Lampiran 4 tabel *Confusion Matrix* model menggunakan metode *Ensemble Stacking* *Class Balancing* dengan masing-masing algoritma sebagai *aggregate*

1. Random Forest

		Predicted		Σ
		no	yes	
Actual	no	96.1 %	7.1 %	32073
	yes	3.9 %	92.9 %	30985
Σ		31007	32051	63058

2. Neural Network

		Predicted		Σ
		no	yes	
Actual	no	98.4 %	16.9 %	32073
	yes	1.6 %	83.1 %	30985
Σ		26305	36753	63058

3. Decision Tree

		Predicted		Σ
		no	yes	
Actual	no	95.2 %	3.9 %	32073
	yes	4.8 %	96.1 %	30985
Σ		32442	30616	63058

4. KNN

		Predicted		Σ
		no	yes	
Actual	no	97.3 %	5.2 %	32073
	yes	2.7 %	94.8 %	30985
Σ		31272	31786	63058

5. SVM

		Predicted		Σ
		no	yes	
Actual	no	96.1 %	8.9 %	32073
	yes	3.9 %	91.1 %	30985
Σ		30370	32688	63058