



TESIS - IF185401

**PENGEMBANGAN PREDIKSI KERUSAKAN  
PERANGKAT LUNAK MENGGUNAKAN  
KOMBINASI OVERSAMPLING DAN  
UNDERSAMPLING**

**AIZUL FAIZ ISWAFAZA**

**NRP. 6025201007**

**DOSEN PEMBIMBING:**

**Ir. Siti Rochimah, MT., Ph.D**

**PROGRAM MAGISTER**

**RUMPUN MATA KULIAH MANAJEMEN CERDAS INFORMASI**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA**

**2022**

*[Halaman ini sengaja dikosongkan]*



**TESIS - IF185401**

**PENGEMBANGAN PREDIKSI KERUSAKAN  
PERANGKAT LUNAK MENGGUNAKAN KOMBINASI  
OVERSAMPLING DAN UNDERSAMPLING**

**AIZUL FAIZ ISWAFAZA**

**NRP. 6025201007**

**DOSEN PEMBIMBING :**

**Ir. Siti Rochimah, MT., Ph.D**

**PROGRAM MAGISTER**

**RUMPUN MATA KULIAH REKAYASA PERANGKAT LUNAK**

**DEPARTEMEN TEKNIK INFORMATIKA**

**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**

**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**

**SURABAYA**

**2022**

*[Halaman ini sengaja dikosongkan]*

# LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
**Magister Komputer (M.Kom.)**  
di  
**Institut Teknologi Sepuluh Nopember**

Oleh:

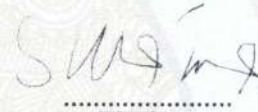
**Aizul Faiz Iswafaza**  
**6025201007**

Tanggal Ujian: 29 April 2022  
Periode Wisuda: September 2022

Disetujui oleh:

## Pembimbing

1. Ir. Siti Rochimah, M.T., Ph.D.  
NIP. 19681002 199403 2 001



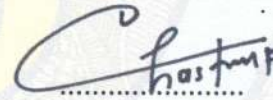
.....

## Penguji:

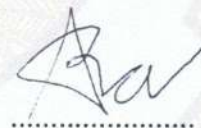
1. Daniel Oranova S, S.Kom., M.Sc., P.D.Eng.  
NIP. 19741123 200604 1 001
2. Dr.Eng. Chastine Fatichah S.Kom., M.Kom  
NIP. 19751220 200112 2 002
3. Dr.Eng. Radityo Anggoro, S.Kom., M.Sc.  
NIP. 19841016 200812 1 002



.....



.....



.....

Kepala Departemen Teknik Informatika  
Fakultas Teknologi Elektro dan Informatika Cerdas



**Dr. Eng. Chastine Fatichah, S.Kom., M.Kom.**

NIP. 19751220 200112 2 002

*[Halaman ini sengaja dikosongkan]*

## **Pengembangan Prediksi Kerusakan Perangkat Lunak Menggunakan Kombinasi Oversampling dan Undersampling**

Nama mahasiswa : Aizul Faiz Iswafaza  
NRP : 6025201007  
Pembimbing : Ir. Siti Rochimah MT., P.Hd

### **ABSTRAK**

Kualitas perangkat lunak dapat ditingkatkan dengan melakukan pengujian perangkat lunak, akan tetapi semakin banyak fitur yang dikembangkan semakin banyak juga sumber daya yang dibutuhkan. Karena itu prediksi kecacatan perangkat lunak (SDP) diperkenalkan. Akan tetapi dataset yang memunculkan berbagai macam permasalahan seperti redudansi data dan kelas tidak seimbang.

Penelitian ini mengusulkan tiga model kombinasi oversampling dan undersampling dalam pengembangan SDP. Model pertama (FCM) merupakan kombinasi dengan alur melakukan undersampling terlebih dahulu kemudian baru oversampling, model kedua (SCM) merupakan kombinasi dengan alur melakukan oversampling kemudian undersampling dan model (TCM) ketiga penggabungan dari hasil metode oversampling dan undersampling. RSMOTE dan ENN digunakan sebagai metode oversampling dan undersampling pada penelitian ini. Hasil dari FCM, SCM dan TCM memberikan serangkaian dataset baru yang lebih seimbang dan juga bersih dari data ambigu, bising dan duplikasi.

Data baru yang dihasilkan model FCM, SCM dan TCM diterapkan pada model prediksinya seperti *Artificial Neural Network*, *Recurrent Neural Network* dan *Convolutional Neural Network*. Kemudian dievaluasi menggunakan pengukuran *F-Measure*, *ROC/AUC* dan Akurasi. Dari hasil evaluasi memberikan hasil yang lebih baik dibandingkan dengan penggunaan data original, RSMOTE dan ENN ditunjukkan dengan rata-rata nilai ketika menggunakan model FCM dengan ketiga model prediksi sebesar 0.866, 0.868 dan 0.869. model SCM sebesar 0.855, 0.879 dan 0.882. Dan model TCM sebesar 0.879, 0.864 dan 0.860.

**Kata kunci:** Kerusakan Perangkat Lunak, Prediksi, Kombinasi *Oversampling dan Undersampling*, *Deep Learning*, *RSMOTE*, *ENN*

*[Halaman ini sengaja dikosongkan]*



## **An Approach of Software Defect Prediction Using a Combination of Oversampling and Undersampling**

Nama mahasiswa : Aizul Faiz Iswafaza  
Student Identity Number : 6025201007  
Supervisor : Ir. Siti Rochimah MT., P.Hd

### **ABSTRACT**

Software quality can be improved by testing the software, but the more features that are developed, the more resources are needed. Therefore software defect prediction (SDP) was introduced. However, datasets that raise various kinds of problems such as data redundancy and unbalanced classes.

This study proposes three combination models of oversampling and undersampling in the development of SDP. The first model (FCM) is a combination with the flow of undersampling first and then oversampling, the second model (SCM) is a combination with the flow of oversampling then undersampling and the third model (TCM) combining the results of the oversampling and undersampling methods. RSMOTE and ENN were used as oversampling and undersampling methods in this study. The results from FCM, SCM and TCM provide a new set of datasets that are more balanced and also clear of ambiguous, noisy and duplication of data.

The new data generated by the FCM, SCM and TCM models are applied to their prediction models such as Artificial Neural Network, Recurrent Neural Network and Convolutional Neural Network. Then evaluated using F-Measure, ROC/AUC and Accuracy measurements. From the evaluation results, it gives better results compared to the use of the original data, RSMOTE and ENN indicated by the average value when using the FCM model with the three prediction models of 0.866, 0.868 and 0.869. SCM models are 0.855, 0.879 and 0.882. And the TCM models are 0.879, 0.864 and 0.860.

**Kata kunci:** *Software Defect, Prediction, Combine of Oversampling and Undersampling, Deep Learning, RSMOTE, ENN*

*[Halaman ini sengaja dikosongkan]*

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT, atas limpahan rezeki, berkah, nikmat dan karunia-Nya kepada penulis sehingga dapat menyelesaikan penelitian ini yang berjudul “Pengembangan Kerusakan Perangkat Lunak Menggunakan Kombinasi Oversampling dan Undersampling”. Penulis menyadari bahwa tanpa bantuan dari berbagai pihak, penelitian ini tidak akan selesai seperti saat ini. Oleh karena itu, penulis ingin menyampaikan terima kasih kepada:

1. Allah SWT yang memberikan nikmat iman, islam, kesehatan, waktu, petunjuk serta berbagai kemudahan sehingga penulis dapat menyelesaikan penelitian ini.
2. Keluarga tercinta yang selalu memberikan dukungan berupa doa, materi dan semangat kepada penulis.
3. Ibu Prof. Siti Rochimah, MT. Ph.D. selaku pembimbing yang telah meluangkan waktu untuk membimbing serta memberikan arahan dalam penelitian ini.
4. Bapak Daniel Oranova S, S. Kom., M.Sc., P.D.Eng., Ibu Dr. Eng. Chastine Fatichah, S. Kom., M. Kom., dan Dr. Eng. Radityo Anggoro, S. Kom., M.Sc. selaku dosen penguji yang telah memberikan saran dan arahan penelitian kepada penulis.
5. Rekan-rekan mahasiswa S2 Teknik Informatika, ITS Surabaya.

Tidak dapat dipungkiri laporan penelitian ini masih memiliki kekurangan. Oleh karena itu, penulis mengharapkan saran dan kritik sebagai bahan perbaikan di masa yang akan datang dan semoga penelitian ini mampu memberikan manfaat bagi semua pembaca

Surabaya, Juni 2022

Aizul Faiz Iswafaza

## DAFTAR ISI

LEMBAR PENGESAHAN TESIS .....	i
ABSTRAK .....	iii
ABSTRACT .....	v
KATA PENGANTAR.....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL .....	xi
DAFTAR PERSAMAAN .....	1
BAB 1 PENDAHULUAN.....	3
1. 1. Latar Belakang.....	3
1. 2. Perumusan Masalah .....	6
1. 3. Tujuan dan Manfaat Penelitian .....	6
1. 4. Kontribusi Penelitian .....	6
1. 5. Batasan Masalah .....	7
BAB 2 KAJIAN PUSTAKA .....	3
2. 1. Referensi Penelitian ( <i>Previous Work</i> ).....	3
2. 2. Kesalahan Perangkat Lunak.....	5
2. 3. NASA MDP dan AEEEM .....	5
2. 4. <i>Oversampling dan undersampling</i> .....	6
2. 5. <i>Robust Synthetic Minority Oversampling Technique (RSMOTE)</i> .....	9
2. 6. <i>Edited Nearest Neighbors (ENN)</i> .....	12
2. 7. Fitur Seleksi menggunakan <i>Random Forest (RF)</i> .....	13
2. 8. <i>Artificial Neural Network (ANN)</i> .....	14
2. 9. <i>Convolutional Neural Network (CNN)</i> .....	16

2. 10. <i>Recurrent Neural Network (RNN)</i> .....	17
BAB 3 METODOLOGI PENELITIAN.....	20
3. 1. Pengumpulan Dataset .....	20
3. 2. Praproses dan Pembagian Dataset .....	23
3. 3. Model Kombinasi <i>Oversampling dan undersampling</i> .....	24
3. 4. Fitur Seleksi.....	27
3. 5. Klasifikasi atau Pembuatan Model Prediksi.....	27
3. 6. Analisis Hasil .....	30
BAB 4 HASIL DAN PEMBAHASAN.....	33
4. 1. Implementasi Model Kombinasi (FCM, SCM dan TCM) .....	33
4. 2. Implementasi Fitur Seleksi .....	40
4. 3. Implementasi Model Prediksi ANN .....	50
4. 4. Implementasi Model Prediksi CNN .....	56
4. 5. Implementasi Model Prediksi RNN .....	63
<b>4. 6. Perbandingan Hasil Prediksi</b> .....	<b>70</b>
<b>4. 7. Analisis Hasil Penelitian</b> .....	<b>78</b>
BAB 5 KESIMPULAN DAN SARAN .....	84
<b>5. 1. Kesimpulan</b> .....	<b>84</b>
<b>5. 2. Saran</b> .....	<b>85</b>
DAFTAR PUSTAKA .....	87
LAMPIRAN A .....	94
BIODATA PENULIS .....	101

## DAFTAR GAMBAR

Gambar 2.1 Distribusi <i>kelas</i> pada NASA MDP .....	7
Gambar 2.2 Ilustrasi Teknik <i>Oversampling</i> .....	8
Gambar 2.3 Ilustrasi Teknik <i>Undersampling</i> .....	8
Gambar 2.4 Tiga bagian utama pada RSMOTE (Chen et al., 2021).....	10
Gambar 2.5 Reprsentasi RSMOTE.....	11
Gambar 2.6 Contoh Alur Metode RF (Sumber: <a href="http://www.deepai.org">www.deepai.org</a> ) .....	14
Gambar 2.7 Komponen artificial neural network (Sumber: <a href="http://www.medium.com">www.medium.com</a> )	15
Gambar 2.8 Contoh artificial neural network.....	15
Gambar 2.9 Skema ANN MLP .....	16
Gambar 2.10 Ilustrasi Metode CNN.....	17
Gambar 3.1 Alur Metodologi Penelitian .....	20
Gambar 3.2 Combination Undersampling-Oversampling (FCM).....	25
Gambar 3.3 Combination Oversampling-Undersampling (SCM Model) .....	25
Gambar 3.4 Merge of Oversampling and Undersampling (TCM Model).....	26
Gambar 4.1 Grafik Pengukuran F-Measure pada hasil prediksi menggunakan ANN, CNN dan RNN.....	79
Gambar 4.2 Perbandingan data distribusi pada AR1.....	81
Gambar 4.3 Perbandingan data distribusi pada KC1.....	82
Gambar 4.4 Perbandingan data distribusi pada AR3.....	82

## DAFTAR TABEL

Tabel 3.1 Metric used on MDP Dataset .....	20
Tabel 3.2 Daftar MDP dataset yang digunakan .....	21
Tabel 3.3 Dataset AEEEM yang digunakan .....	22
Tabel 3.4 <i>Attributes</i> dari dataset AEEEM.....	22
Tabel 3.5 Contoh Hasil Normalisasi Pada Dataset MDP.....	24
Tabel 3.6 Konfigurasi Model Prediksi <i>Deep Learning</i> .....	27
Tabel 4.1 Hasil Oversampling dan undersampling dengan FCM .....	33
Tabel 4.2 Hasil Oversampling dan undersampling dengan Model SCM.....	35
Tabel 4.3 Hasil Oversampling dan undersampling dengan Model TCM .....	36
Tabel 4.4 Perbandingan hasil resampling menggunakan ENN, RSMOTE, FCM, SCM dan TCM.....	39
Tabel 4.5 Hasil Fitur Seleksi Dataset PC1 .....	40
Tabel 4.6 Hasil Fitur Seleksi Dataset PC4 .....	41
Tabel 4.7 Hasil Fitur Seleksi Dataset PC5 .....	41
Tabel 4.8 Hasil Fitur Seleksi Dataset AR1 .....	42
Tabel 4.9 Hasil Fitur Seleksi Dataset AR3 .....	42
Tabel 4.10 Hasil Fitur Seleksi Dataset AR4 .....	43
Tabel 4.11 Hasil Fitur Seleksi Dataset KC1 .....	43
Tabel 4.12 Hasil Fitur Seleksi Dataset KC2 .....	44
Tabel 4.13 Hasil Fitur Seleksi Dataset KC3 .....	45
Tabel 4.14 Hasil Fitur Seleksi Dataset MC1.....	45
Tabel 4.15 Hasil Fitur Seleksi Dataset MC2.....	46
Tabel 4.16 Hasil Fitur Seleksi Dataset CM1.....	46
Tabel 4.17 Hasil Fitur Seleksi Dataset JM1 .....	47
Tabel 4.18 Hasil Fitur Seleksi Dataset MW1.....	47
Tabel 4.19 Hasil Fitur Seleksi Dataset EQ.....	48
Tabel 4.20 Hasil Fitur Seleksi Dataset JDT .....	48
Tabel 4.21 Hasil Fitur Seleksi Dataset Lucene .....	49
Tabel 4.22 Hasil Fitur Seleksi Dataset Mylyn .....	49
Tabel 4.23 Hasil Fitur Seleksi Dataset PDE .....	49

Tabel 4.24 Hasil evaluasi prediksi ANN menggunakan data FCM .....	50
Tabel 4.25 Hasil evaluasi prediksi ANN menggunakan data SCM .....	51
Tabel 4.26 Hasil evaluasi prediksi ANN menggunakan data TCM .....	51
Tabel 4.27 Hasil evaluasi prediksi CNN menggunakan data FCM.....	56
Tabel 4.28 Hasil evaluasi prediksi CNN menggunakan data kombinasi oversampling dan undersampling model SCM .....	57
Tabel 4.29 Hasil evaluasi prediksi CNN menggunakan data kombinasi oversampling dan undersampling model TCM .....	58
Tabel 4.30 Hasil Prediksi RNN menggunakan Data kombinasi oversampling dan undersampling Model FCM .....	63
Tabel 4.31 Hasil Prediksi RNN menggunakan Data kombinasi oversampling dan undersampling Model SCM .....	64
Tabel 4.32 Hasil Prediksi RNN menggunakan Data kombinasi oversampling dan undersampling Model TCM .....	65
Tabel 4.33 Perbandingan Hasil F-Measure pada Model Prediksi ANN .....	71
Tabel 4.34 Perbandingan Hasil F-Measure pada Model Prediksi CNN.....	72
Tabel 4.35 Perbandingan Hasil F-Measure pada Model Prediksi RNN.....	73
Tabel 4.36 Perbandingan Hasil ROC/AUC pada Model Prediksi ANN .....	73
Tabel 4.37 Perbandingan Hasil ROC/AUC pada Model Prediksi CNN .....	74
Tabel 4.38 Perbandingan Hasil ROC/AUC pada Model Prediksi RNN .....	75
Tabel 4.39 Perbandingan Hasil Akurasi pada Model Prediksi ANN .....	76
Tabel 4.40 Perbandingan Hasil Akurasi pada Model Prediksi CNN .....	77
Tabel 4.41 Perbandingan Hasil Akurasi pada Model Prediksi RNN .....	78



## DAFTAR PERSAMAAN

Persamaan 1 .....	24
Persamaan 2 .....	31
Persamaan 3 .....	31
Persamaan 4 .....	31

*[Halaman ini sengaja dikosongkan]*

## BAB 1

### PENDAHULUAN

Karya ilmiah ini diawali dengan pendahuluan yang berisikan gambaran umum isi penelitian yang akan dilakukan. Bab ini menjabarkan beberapa hal, yaitu latar belakang, rumusan masalah, tujuan dan manfaat penelitian, kontribusi penelitian dan batasan masalah.

#### 1. 1. Latar Belakang

Berbagai teknik telah diusulkan dalam upaya meningkatkan kualitas perangkat lunak (PL). Salah cara untuk meningkatkan kualitas PL dengan mengurangi kecacatan pada saat pengembangan sebuah fitur (P. He et al., 2014). Hingga saat ini berbagai cara telah dikembangkan salah satunya dengan memprediksi kecacatan pada suatu PL sehingga mampu memperkirakan komponen perangkat lunak yang memungkinkan untuk mengalami kesalahan. Dalam melakukan prediksi, dibutuhkan data yang berisikan hasil pengukuran langsung metrik pengujian PL (Pralienka et al., 2017). Salah satu metrik digunakan adalah *McCabe Metrics*. Telah banyak model prediksi kecacatan perangkat lunak (SDP) yang telah dikembangkan. Pada penelitian yang dilakukan (Malhotra, 2015), telah mengadopsi beberapa metode klasifikasi *ensemble* yang digunakan dalam *machine learning* antara lain *Support Vector Machine* (SVM), *Naïve Bayes* (NB) dan *Decision Tree* (DT). Dengan melakukan SDP, alokasi biaya dan sumberdaya untuk melakukan peningkatan kualitas perangkat lunak menjadi lebih hemat.

*MDP Software Engineering Repository* merupakan salah satu dataset yang sering digunakan dalam melakukan eksperimen pada SDP. Dataset tersebut terdiri dari berbagai macam metrik antara lain *McCabe Metrics* yang digunakan untuk menghitung kompleksitas, *line of code* (loc), *total number of operand* (TNO) dan *Halstead's Metrics* (Leslie et al., 2017; Massoudi et al., 2021; Putri & Wahono, 2015; Yousef, 2015). Pada dataset tersebut berisikan beberapa project yang dengan hasil perhitungan pengujian perangkat lunak tiap fungsi yang dikembangkan. Project yang ada pada dataset tersebut antara lain KC1, KC2, KC3, MW1, JM1, PC1 dan lain-lain.

Dari pengembangan penelitian SDP, terdapat beberapa object penelitian yang selalu menjadi pembahasan utama, yaitu redundansi data, relevansi fitur, korelasi, sample yang hilang dan kelas yang tidak seimbang. Hal tersebut dikarenakan pada data yang digunakan untuk melakukan prediksi hanya tergolong kedalam dua kelas yaitu *defect* dan *non-defect* atau yang lebih dikenal dengan *binary classes*, permasalahan tersebut membuat kurang efektifnya hasil prediksi yang dilakukan.

Banyaknya jumlah fitur yang ada pada dataset penelitian SDP, menyebabkan tingkat kebutuhan relevansi fitur sangatlah penting (Fan et al., 2020). Untuk menangani kebutuhan fitur tersebut dapat dilakukan dengan melakukan seleksi fitur. Pada beberapa penelitian yang sudah ada, berbagai macam model telah digunakan seperti *Particle Swarm Optimization* (PSO) (Wahono & Suryana, 2013), *Chi-Square*, *Information Gain* (IG) (Arshad et al., 2018) and *Relief* (Khadijah et al., 2020). Empat penelitian tersebut mampu memberikan fitur yang relevan dengan menghilangkan fitur yang memiliki skor terkecil yang mengartikan bahwa fitur tersebut tidak informatif (Kuhn & Johnson, 2013; Liu & Motoda, 1998). Jika SDP tetap dilakukan tanpa melakukan seleksi fitur, maka banyaknya jumlah *attribute* akan memperlambat proses dalam memprediksi (Kuhn & Johnson, 2013).

Redundansi data merupakan masalah utama juga yang diangkat pada penelitian ini. Redundansi data merupakan duplikasi data yang memiliki nilai sama dan berulang dalam sebuah kumpulan data. Undersampling merupakan metode untuk menyelesaikan permasalahan tersebut (Zhu et al., 2020). Beberapa penelitian telah mengembangkan model prediksi dengan menerapkan berbagai macam metode untuk menyelesaikan permasalahan redundansi pada SDP. Pada penelitian (Tahir et al., 2009) menggunakan *Random Undersampling* (RUS). Sesuai dengan nama metode yang digunakan, RUS memilih secara acak data yang akan dihapuskan. Metode tersebut berhasil mengurangi data dan meningkatkan *runtime*, akan tetapi metode tersebut memungkinkan untuk menghilangkan data dengan informasi yang penting (S. Mishra, 2017). Dari penelitian yang dilakukan oleh (Zhu et al., 2020) metode *Edited Nearest Neighbors* (ENN) digunakan untuk menggantikan metode RUS. Hasil penelitian tersebut mampu memberikan hasil klasifikasi yang lebih baik dengan ditunjukkan pada hasil evaluasi memberikan peningkatan nilai akurasi yang

sebelumnya ketika menggunakan RUS hanya mampu memberikan 0,786 dan setelah menggunakan ENN nilainya meningkat menjadi 0,900.

Selain reduksi data, permasalahan yang timbul pada SDP adalah ketidakseimbangan kelas, yaitu kondisi dimana dataset memiliki persebaran data pada setiap kelas memiliki jumlah tidak seimbang. Kondisi tersebut mengakibatkan model apapun yang digunakan untuk melakukan klasifikasi akan menerapkan prinsip *Ocean's Razor*. Prinsip tersebut akan membuat model klasifikasi cenderung menghasilkan jawaban sederhana menggunakan kelas mayoritas (Beyan & Fisher, 2015; Malhotra, Agrawal, et al., 2021). Untuk memperbaiki hal tersebut, metode oversampling digunakan untuk menyeimbangkan distribusi kelas (Fan et al., 2020). Salah satu metode *oversampling* yang sering digunakan adalah *Synthetic Minority Oversampling Technique* (SMOTE) (Chawla et al., 2002; Junnan Li et al., 2021). Tetapi metode tersebut memiliki kelemahan, dimana data *noise* memungkinkan untuk terbentuk. Berbagai macam model pengembangan SMOTE telah dilakukan. *Robust Synthetic Minority Oversampling Technique* (RSMOTE) merupakan salah satunya. Metode tersebut mampu memberikan data baru dengan tingkat noise yang lebih sedikit. Penerapan data baru hasil RSMOTE ketika dilakukan klasifikasi mampu memberikan hasil yang baik, ditunjukkan pada nilai evaluasi *F1-Measure* yang diberikan lebih tinggi dibanding dengan penerapan data baru hasil SMOTE (Chen et al., 2021).

Penelitian ini mengusulkan sebuah pendekatan pada prediksi kesalahan perangkat lunak menggunakan kombinasi *oversampling* dan *undersampling* untuk mengoptimasi model prediksi dengan metode *deep learning*. Untuk *oversampling* sendiri menggunakan RSMOTE dan metode *undersampling* yang digunakan adalah ENN. Kombinasi ini nantinya akan melakukan *oversampling* pada dataset yang digunakan kemudian hasil *oversampling* tersebut akan dibersihkan menggunakan ENN. Pembersihan disini bertujuan untuk menghapus data yang berisik (mengganggu) dan data yang ambigu ketika dihasilkan oleh teknik SMOTE pada kelas mayoritas. Sehingga diharapkan dengan kombinasi ini data yang digunakan untuk prediksi menjadi lebih seimbang dan mengurangi nilai *true negative* atau *false negative*.

Oleh karena itu, penelitian ini bertujuan untuk meningkatkan dan mempertahankan nilai evaluasi dengan menggunakan kombinasi *oversampling* dan *undersampling* RSMOTE dan ENN pada model prediksi *deep learning*. Dari metode tersebut akan dilakukan analisis untuk menemukan kombinasi terbaik pada beberapa metode *deep learning*. Hasil kombinasi tersebut diharapkan mampu meningkatkan nilai evaluasi pada SDP sehingga hasil yang diberikan menjadi semakin relevan dan semakin akurat pada *deep learning*.

Pada karya ilmiah ini memiliki susunan penulisan sebagai berikut BAB 2 menjelaskan tentang penelitian sebelumnya dan ide umum dari metode yang digunakan. BAB 3 berisi terkait usulan metode yang digunakan untuk melakukan prediksi kesalahan perangkat lunak dengan beberapa kombinasi yang digunakan, didalam bab tersebut juga dijelaskan detail setiap alur yang dijalankan. BAB 4 menyajikan hasil evaluasi menggunakan berbagai macam metode pengukuran dan skenario uji coba model prediksi. Pada BAB atau bab terakhir berisikan kesimpulan dan saran dari hasil percobaan yang telah dilakukan.

#### 1. 2. Perumusan Masalah

Rumusan masalah yang diangkat dalam penelitian ini adalah sebagai berikut:

1. Bagaimana kombinasi metode *oversampling* dan *undersampling* yang dapat digunakan pada *software defect prediction*?
2. Bagaimana model kombinasi *oversampling* dan *undersampling* yang optimal untuk melakukan *software defect prediction*?
3. Bagaimana hasil evaluasi kinerja model kombinasi *oversampling* dan *undersampling* pada *software defect prediction*?

#### 1. 3. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini untuk mendapatkan model kombinasi *oversampling* dan *undersampling* yang mampu memberikan hasil akurasi dan evaluasi yang optimal dalam melakukan *software defect prediction*.

#### 1. 4. Kontribusi Penelitian

Kontribusi pada penelitian ini berupa pengembangan model kombinasi metode *oversampling* dan *undersampling*. Hasil kombinasi tersebut kemudian digunakan untuk melakukan aktivitas *software defect prediction* dengan model

klasifikasi berupa *deep learning*. Usulan metode ini diharapkan mampu memberikan hasil evaluasi yang lebih baik dalam melakukan prediksi untukantisipasi kesalahan perangkat lunak dalam upaya meningkatkan kualitas perangkat lunak.

#### 1. 5. Batasan Masalah

Batasan masalah pada penelitian ini adalah:

1. Penelitian ini dilakukan untuk mengetahui hasil evaluasi pada model klasifikasi setelah dilakukan penerapan kombinasi metode *oversampling* dan *undersampling* pada *software defect prediction*
2. Penelitian ini dilakukan untuk mengetahui pengaruh penggunaan kombinasi metode *oversampling* dan *undersampling* dalam menyelesaikan permasalahan ketidakseimbangan data pada *software defect prediction*
3. Dataset yang digunakan pada penelitian ini menggunakan NASA MDP dan AEEEM

*[Halaman ini sengaja dikosongkan]*



## BAB 2

### KAJIAN PUSTAKA

Bab ini merupakan pembahasan dari referensi terkait yang telah dilakukan dalam menyelesaikan permasalahan sesuai dengan uraian pada latar belakang, diawali dengan menjabarkan hal-hal yang diterapkan pada metode yang diusulkan, kelemahan yang terdapat pada penelitian sebelumnya, komparasi penelitian sebelumnya. Selanjutnya dilanjutkan dengan kelebihan dari metode yang akan digunakan untuk menyelesaikan permasalahan.

#### 2. 1. Referensi Penelitian (*Previous Work*)

Pada penelitian (Malhotra, Nishant, et al., 2021) mengusulkan penggunaan SMOTE dalam menyelesaikan permasalahan ketidakseimbangan data yang ada pada dataset Apache (Beam, Hive, Geode dan Flink). Dalam upaya untuk mendapatkan hasil yang lebih baik, *author* menambahkan teknik ekstraksi fitur yaitu menggunakan RF Basis PSO. Terdapat 3 matrik yang digunakan dalam melakukan evaluasi yaitu *Area Under ROC Curve*, *Recall* dan *F-Measure*. Hasil dari penelitian dilakukan, menunjukkan SMOTE+PSO memberikan peningkatan nilai evaluasi.

Pada penelitian (Chen et al., 2021) mengusulkan pendekatan dan pengembangan baru pada SMOTE yang diberi nama RSMOTE. Metode tersebut diusulkan karena basic metode SMOTE memberikan performa kinerja yang kurang baik ketika ada gangguan pada label dataset. Selain itu, pendekatan pada penelitian sebelumnya dalam mengatasi gangguan pada label menggunakan teknik *noise filter*. Metode tersebut sulit untuk dilakukan optimasi dan performanya bergantung pada teknik *noise filter* yang digunakan. RSMOTE akan membagi data menjadi dua bagian sample, *borderline* dan *safe samples*. Dari metric evaluasi yang digunakan yaitu berupa *Precision*, *Recall*, *Area Under the Curve (AUC)*, *F1-Measure*, dan *G-Mean*, metode yang diusulkan memberikan performa kinerja yang lebih baik.

Pada penelitian (Kuncheva et al., 2018) *author* melakukan pembuatan ringkasan *keyframe* pada video. Dikarenakan metode saat ini sulit untuk membuat ringkasan yang berbeda setiap frame, *author* mengusulkan sebuah algoritma yang diberinama *Greedy Tabu Selector*. ENN diaplikasikan pada metode yang diusulkan

untuk mendapatkan akurasi tertinggi pada pemilihan subset frame. Metode tersebut kemudian dipasangkan dengan *Convolutional Neural Network* (CNN). Dalam melakukan evaluasi, author membandingkan hasil metode yang diusulkan dengan metode lain yang bernama *closed to centroid*. Hasilnya menunjukkan bahwa, metode yang diusulkan menghasilkan ringkasan yang lebih baik.

Pada Penelitian (Zhu et al., 2020), melakukan eksperimen dalam prediksi identifikasi *Lysine succinylation*. Banyak metode konvensional yang telah diterapkan pada penelitian-penelitian sebelumnya. Akan tetapi, metode tersebut memakan banyak sekali resource dan waktu yang diperlukan untuk melakukan komputasi. Untuk itu penelitian tersebut mengusulkan kombinasi RF dengan berbagai skema *sequences*. Untuk mengatasi permasalahan fundamental pada ruang lingkup prediksi yaitu ketidakseimbangan kelas, penulis menggunakan edited nearest neighbors sebagai teknik undersampling dan adaptive synthetic oversampling. *Two-step feature selection* digunakan untuk menambah optimasi yang ada. Dalam melakukan evaluasi, penelitian tersebut menggunakan metric assessment antara lain *sensitivity, specificity, accuracy* dan *Matthew' Correlation Coefficient* (MCC). Dari hasil penelitian tersebut menunjukkan penggunaan edited nearest neighbors pada obyek tersebut menghasilkan nilai evaluasi yang lebih baik, dan ketika digabungkan dengan teknik oversampling memberikan nilai evaluasi yang jauh lebih baik lagi.

Pada penelitian (Qiao et al., 2020) mengusulkan pendekatan deep learning menggunakan *Deep Neural Network* (DNN) dalam melakukan prediksi software systems defect menggunakan dataset *Medical Imaging System* (MIS) dan NASA MDP (KC2). Untuk mengetahui bahwa metode usulan lebih baik, penelitian tersebut menghitung nilai *Mean Squared Error* (MSE). Metode yang digunakan untuk perbandingan adalah *Support Vector Regression*, Fuzzy SVR dan *Decision Tree Regression*. Hasil dari perbandingan evaluasi menunjukkan bahwa metode yang diusulkan berhasil mengurangi nilai MSE sebesar 14% dan nilai *squared correlation coefficient* meningkat sebesar 8%.

Dari penelitian (Jia et al., 2021) melakukan prediksi kesiapan eskubasi untuk perawatan pasien yang sedang mengalami sakit kritis pada *Intensive Care Unit* (ICU). Data untuk melakukan prediksi berasal dari informasi yang direkam secara

rutin dan bisa didapatkan secara public dengan nama MIMIC-III dataset. Penulis dalam penelitiannya menerapkan *Convolutional Neural Network* (CNN) untuk membantu melakukan prediksi. Hasil penerapan CNN pada prediksi didapatkan hasil akurasi sebesar 86% dan nilai AUC-ROC 0.94. Dengan hasil baik yang diberikan oleh penerapan CNN pada prediksi membantu dokter dalam memahami dokter dalam menentukan keputusan dalam penanganan pasien. Dibandingkan dengan metode lain seperti ANN, *Logistic Regression*, *Support Vector Machine*, *Decision Tree* dan *Random Forest Tree*, dari 6 metric yang digunakan untuk evaluasi CNN memberikan hasil optimal.

Dari beberapa penelitian yang digunakan sebagai *previous work*, penelitian ini akan melakukan mengembangkan *software defect prediction* menggunakan kombinasi RSMOTE dari teknik *oversampling* dan ENN dari teknik *undersampling* yang diterapkan pada *Deep Learning* yaitu CNN. Dari kombinasi tersebut akan dibuat menjadi 3 Model yang ditunjukkan pada Gambar 3.2 hingga 3.4. tujuan pembuatan model tersebut untuk mendapatkan skema alur kombinasi yang memberikan hasil kinerja lebih baik.

## 2. 2. Kesalahan Perangkat Lunak

Kesalahan Perangkat Lunak (*software defect*) merupakan salah satu bagian dalam sebuah sistem yang dikembangkan dan memerlukan perhatian khusus. Bentuk hasil kesalahan perangkat lunak didapatkan dari proses pengujian. Tingginya kompleksitas dan ukuran suatu perangkat lunak, memungkinkan untuk memunculkan kesalahan kesalahan yang mengakibatkan kualitas perangkat lunak menurun. Dengan adanya prediksi kesalahan perangkat lunak akan membantu dalam mengurangi sumber daya yang diperlukan dalam menjaga kualitas perangkat lunak (Le et al., 2015; Mause et al., 2011; Perera, 2020; Yadav, 2020).

Proses prediksi perangkat lunak dibentuk dengan memanfaatkan perhitungan matrik pengujian yang kemudian akan dilakukan pembentukan model prediksi. Model tersebut kemudian akan digunakan dalam melakukan prediksi pada data perhitungan matrik pengujian suatu fitur baru yang telah dikembangkan pada perangkat lunak tersebut.

## 2. 3. NASA MDP dan AEEEM

Dalam melakukan penelitian ini dibutuhkan data untuk memudahkan eksperimen *software defect prediction*. Sulit untuk mendapatkan data terkait *software defect* pada berbagai perusahaan. Jarang sekali sebuah perusahaan melakukan pengukuran kesalahan program. Jika perusahaan memiliki data tersebut, belum tentu akan dipublikasikan secara gratis dan perusahaan akan menyimpan data tersebut untuk kepentingan internal saja (Bowes & Sun, 2012).

NASA MDP merupakan penyedia data repositories secara public dan dapat dengan mudah diakses melalui internet. *Repository* memiliki berbagai macam dataset seperti PC1, PC2, KC1, etc. data tersebut tidak bisa digunakan secara langsung dalam melakukan prediksi, perlu adanya *preprocessing* dan analisis lebih lanjut untuk mendapatkan hasil yang maksimal dan sesuai dengan harapan.

Setiap dataset yang ada pada NASA MDP berisikan *static code metric* seperti *Halstead* dan *McCabe*. Data yang diperlukan dalam menghitung *metrics* tersebut didapatkan dari *Defect tracking system* yang menghasilkan sejumlah *report error*. Setiap dataset NASA MDP memiliki dua kelas yaitu *defect* dan *non-defect*. Dari kedua kelas tersebut mayoritas data berada pada kelas *non-defect*.

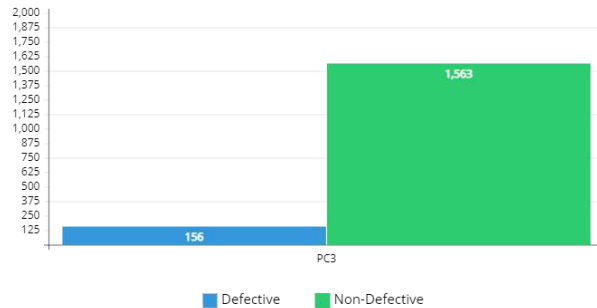
AEEEM merupakan salah satu data *repositories* yang juga dapat diakses secara *public*. Dataset ini biasa digunakan untuk melakukan prediksi kesalahan perangkat lunak berisikan 34 versi rilis dari 10 proyek yang dikembangkan menggunakan bahasa pemrograman java (Ambros et al., 2010; Ambros & Lanza, 2012). Setiap datasetnya memiliki dua kelas seperti NASA MDP, yaitu kelas *defect* dan *non-defect*. Kelas ini dihasilkan dari jumlah *defect* yang didapatkan. Mayoritas data yang ada pada kelas tersebut berada pada kelas *non-defect*.

Mayoritas data pada kedua jenis dataset tersebut menyebabkan ketidakseimbangan pada kelas *defect* dikarenakan jumlah yang cenderung sedikit.

#### 2. 4. *Oversampling dan undersampling*

Dataset yang memiliki perbedaan jumlah kelas yang sangat jauh disebut dengan imbalanced data. Dataset dengan kasus ketidakseimbangan dapat dengan mudah dijumpai pada permasalahan *medical diagnosis*, *network intrusion detection*, *fraud detection*, *software defect prediction* etc. (Mahmood et al., 2015; Miao & Zhao, 2011; Padmaja et al., 2007; Zi, 2014), NASA MDP merupakan salah

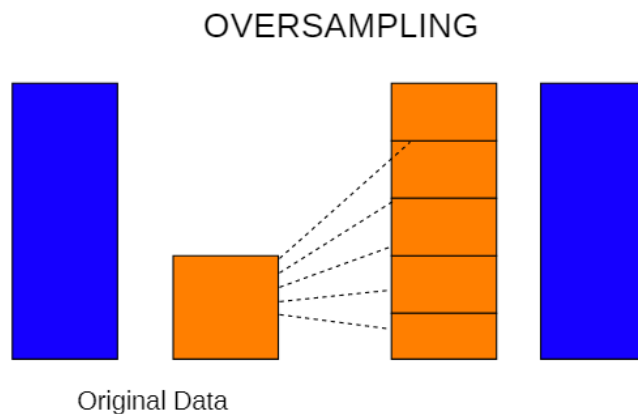
satu dataset yang digunakan dalam melakukan *software defect prediction*, salah satu dataset yang ditunjukkan pada Gambar 2.1 yang diberi nama PC3. Data tersebut memiliki perbedaan jumlah kelas yang sangat mencolok dan condong pada satu sisi.



Gambar 2.1 Distribusi *kelas* pada NASA MDP

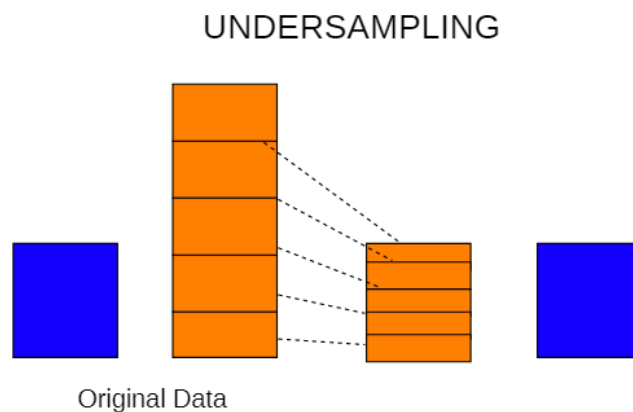
Ketidakseimbangan dataset tidak bisa dibiarkan saja ketika ingin digunakan untuk pembuatan model *machine learning*. Seperti yang dituliskan pada penelitian (Beyan & Fisher, 2015; Malhotra et al., 2017) model *machine learning* yang digunakan akan cenderung melakukan prediksi pada kelas mayoritas dan secara otomatis akan mengabaikan kelas minoritas. Dengan mengabaikan kelas minoritas, pada kebanyakan permasalahan yang ada ketika dilakukan prediksi, model menjadi lebih tidak akurat disisi lain klasifikasi kelas minoritas menjadi lebih penting dibandingkan dengan kelas mayoritas (Malhotra et al., 2017; Seliya & Khoshgoftaar, 2011). Maka untuk menyelesaikan permasalahan tersebut digunakan teknik *oversampling* atau *undersampling*.

*Oversampling* merupakan sebuah cara untuk mengambil data dengan kelas minoritas kemudian membuat persebaran dengan jumlah yang sama dengan kelas mayoritas. Gambar 2.2 menunjukkan representasi teknik *oversampling*.



Gambar 2.2 Ilustrasi Teknik *Oversampling*

Undersampling merupakan sebuah cara yang berkebalikan *oversampling*, teknik ini akan mengurangi sejumlah data dengan kelas mayoritas hingga jumlahnya sama dengan kelas minoritas. Gambar 2.3 merupakan representasi dari teknik *undersampling*.



Gambar 2.3 Ilustrasi Teknik *Undersampling*

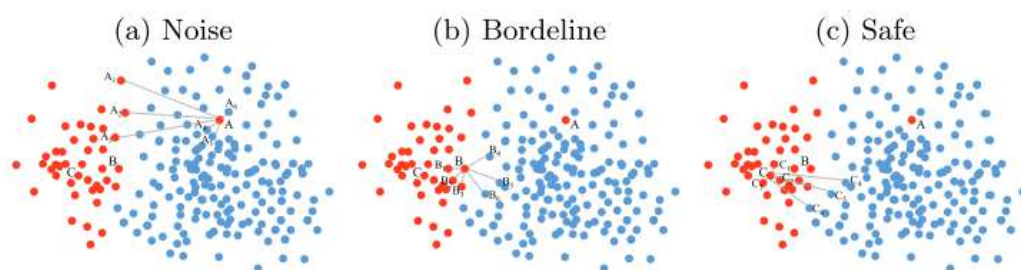
Selain ketidakseimbangan data, sering sekali diiringi dengan gangguan data seperti *noise*, ambigu dan duplikasi data. Data noisy merupakan data dengan yang berisikan informasi tidak berarti. Data ini biasanya berasal dari kesalahan manusia ketika memberikan label dan masalah eksternal ketika pengumpulan data. Data ini biasanya berada pada data dengan kelas yang berbeda atau heterogen. Hampir sama dengan data *noisy*, data ambigu merupakan data yang sama atau identik tetapi label yang diberikan berbeda. sedangkan duplikasi data merupakan data yang sama dan identik tersimpan dalam kumpulan data lebih dari sekali (Du, 2013; Elmagarmid & Member, 2007).

## 2. 5. *Robust Synthetic Minority Oversampling Technique (RSMOTE)*

Pada dataset yang digunakan pada model prediksi terdapat satu masalah yaitu ketidakseimbangan *data*. Diketahui secara umum bahwa Teknik machine learning akan secara otomatis mengabaikan data dengan kelas minoritas yang dapat memberikan kinerja buruk. Padahal data minoritas bisa menjadi kelas terpenting dalam meningkatkan akurasi yang dihasilkan model prediksi. Terdapat tiga kategori *minority samples* antara lain *noise samples locating*, *borderline samples locating* dan *safe sample locating*. Untuk menyelesaikan permasalahan tersebut, berbagai macam pendekatan digunakan antarlain dengan melakukan duplikasi data pada kelas minoritas sehingga jumlahnya setara dengan jumlah kelas mayoritas meski hasilnya tidak menambahkan informasi yang bisa meningkatkan kualitas informasi pada model prediksi. Pada beberapa penelitian saat ini, metode yang dikembangkan hanya berfokus pada kategori tertentu saja untuk menghasilkan sample yang lebih baik dan mendukung kinerja sebuah model prediksi (Bunhumpornpat et al., 2009; Han et al., 2005; H. He et al., 2008). Dengan adanya permasalahan tersebut dikembangkan sebuah metode sintesis data pada kelas minoritas yang disebut SMOTE (Chawla et al., 2002). Penggunaan SMOTE mampu meningkatkan nilai metric evaluasi yang diproduksi model prediksi menggunakan machine learning.

Saat ini SMOTE memiliki berbagai macam variasi yang telah dikembangkan. Beberapa variasi populer yang digunakan adalah, Borderline-SMOTE (Han et al., 2005), Adaptive Synthetic Sampling (ADASYN) (H. He et al., 2008) dan RSMOTE (Chen et al., 2021), etc. Dari penelitian yang dilakukan oleh (Baiyu Chen, 2021) beberapa varian smote dibandingkan dengan metode usulan yaitu RSMOTE menunjukkan hasil yang optimal pada penerapan di beberapa metode *machine learning* untuk model prediksi. Nilai paling optimal diperoleh ketika RSMOTE digunakan pada metode *Decission Tree*. RSMOTE terbentuk karena melihat keberhasilan k-means SMOTE dalam melakukan *oversampling* (Douzas et al., 2018). Metode ini akan melakukan kalkulasi sederhana dalam membedakan *borderline-samples* dan *safe-samples*. *Borderline samples* akan terbentuk dengan mengukur *k-nearest* kelas sejenis (homogen) dan tidak sejenis (heterogen) yang terdekat. *Safe samples*, akan terbentuk dengan mengukur nilai rata-rata dua titik dalam *clustering*. Dari dua hasil tersebut kemudian akan diukur

kembali dengan kelas mayoritas menggunakan *k-nearest neighbors* setiap kelas minoritas. Sehingga hasil dari pengukuran tersebut akan menjadi lebih banyak *safe samples* dibandingkan dengan *gray samples*, hal tersebut bertujuan untuk benar benar memisahkan antar kelas yang ada dengan jumlah yang seimbang. Algoritma 2.1 Menunjukkan pseudocode dari RSMOTE. Pada Gambar 2.4 menunjukkan jarak antara setiap jenis sample. Metode ini akan memperhitungkan jarak kedekatan kelas mayoritas dan minoritas, jika diasumsikan  $k=3$  maka RSMOTE akan melakukan perhitungan kedekatan antara tiga tetangga dengan kelas heterogenya dan tiga tetangga dengan kelas homogenya. Untuk menentukan *cluster* data tersebut dilakukan perbandingan dimana ketika jarak tetangga heterogenya dijumlah kemudian dibagi dengan jarak total tetangga homogen bernilai lebih kecil dari satu maka data tersebut akan masuk pada kelas mayoritas begitu juga sebaliknya jika nilai perhitungan tersebut lebih dari satu maka data tersebut akan dianggap minoritas. Perhitungan tersebut merupakan persamaan *relative density* yang bertujuan untuk membedakan jenis sampel minoritas. Sehingga dari hasil perhitungan itu akan digunakan untuk menjadi penentu lokasi yang aman dalam memproduksi data baru untuk kelas minoritas (Xia et al., 2015).

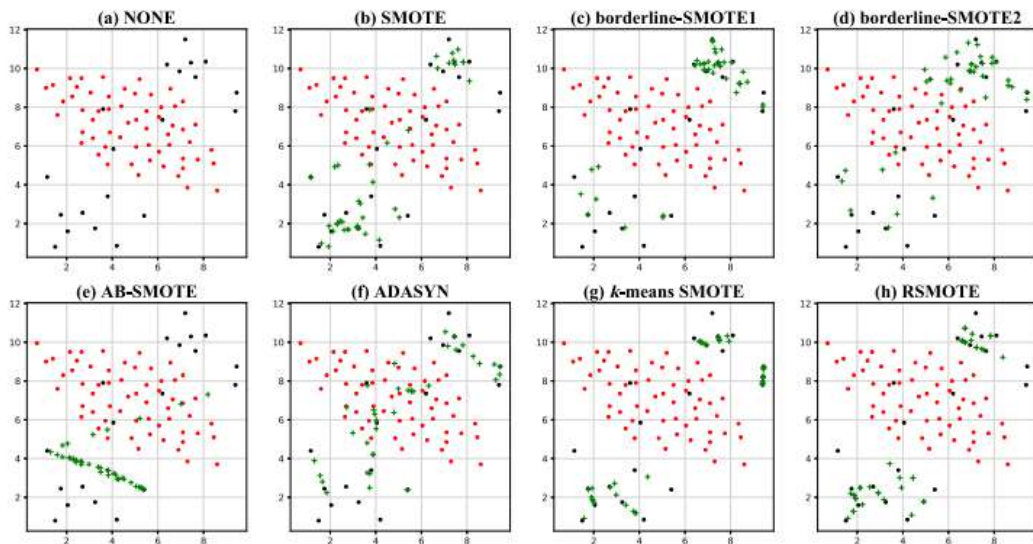


Gambar 2.4 Tiga bagian utama pada RSMOTE (Chen et al., 2021)

Pada Algoritma 2.1 dapat diamati bahwa RSMOTE ini dibagi menjadi tiga tahapan utama yang pertama adalah melakukan filterisasi sample mayoritas, tahap kedua membagi menjadi dua *cluster* dan ketiga merupakan generate data pada area minoritas *class*. Pada tahap pertama akan dilakukan perhitungan *relative density* untuk setiap sample kelas minoritas yang ada sehingga didapatkan jarak suatu data terhadap tetangga-tetangga terdekatnya di kelas homogen dan heterogenya seperti pada penjelasan sebelumnya. Pada tahapan kedua, data akan dibagi menjadi dua cluster dimana nilai *relative density* yang kecil akan dikelompokkan menjadi *cluster*



pada border area dan density yang besar akan dikelompokkan menjadi *cluster* pada *safe area*. Ditahapan terakhir, akan melakukan pembuatan data baru, dimana data baru yang diciptakan akan lebih banyak pada *cluster safe area*. Sehingga RSMOTE mencoba untuk mempertahankan sebanyak mungkin data dengan kelas minoritas, kecuali untuk data yang memungkinkan untuk menimbulkan *noise*. Hal ini ditunjukkan pada tahap pertama dimana data dengan label *noise* tidak akan diikuti sertakan pada sampel minoritas (Chen et al., 2021). Dengan diproduksinya data lebih banyak pada area aman, membuat jarak antar kelas menjadi lebih kuat. Seperti yang ditunjukkan pada Gambar 2.5 pada bagian RSMOTE diperlihatkan bahwa data yang mengganggu/*noise* akan dibersihkan jika dibandingkan dengan hasil varian SMOTE lainnya.



Gambar 2.5 Reprsentasi RSMOTE

Algoritma 2.1 Pseudocode RSMOTE

No	Pseudocode
	Input: Training set $D = P \cup Q$ , $P$ : sample minoritas dan $Q$ : sample mayoritas; $IR$ : ratios ekspektasi ketidakseimbangand $k$ : total <i>nearest neighbors</i> Output: <i>Syn</i> .
1	// Jumlah total sampel minoritas yang perlu disintasi
2	// Step 1: Filter sample mayoritas yang memiliki
	<b>For</b> every minority sample $x_i \sim P$ <b>do</b>
3	Compute $k$ <i>nearest neighbors</i>
4	Count number $m$ of majority samples in $k$ <i>nearest neighbors</i>
5	If $m > k$ then
6	Filter $x_i$ from $P$
7	Else

8	$P' \leftarrow P' \cup \{x_i\};$
9	$RD(x_i) \leftarrow$ Calculate the relative density of $x_i$
10	End if
11	End for
12	// Step 2: Naturally divide $P'$ into two clusters based on the 2 means clustering result of $RD(P')$ .
13	2-clusters $C_A, C_B \leftarrow$ 2-means( $RD(P')$ ), where $c_A$ and $c_B$ represent the centers of $C_A$ and
14	$C_B$ respectively, and $c_A \geq c_B$
15	$P'_A, P'_B \leftarrow$ minority samples correspond to $C_A$ and $C_B$ respectively
16	$Syn$
17	// Step 3: reweight the number generated by $x_i$ based on the number $m$ of majority
18	samples in its $k$ nearest neighbors and generate new samples within each cluster,
19	respectively
20	For each cluster do
21	Number of minority samples need to be synthesized in $P'_j$
22	For each samples $x_i$ in $P'_j$ do
23	$w = \frac{k-m}{k+1}$
24	$\bar{w} = \frac{w}{\sum w}$
25	$N_i = \bar{w} \times N_j$ Populate( $N_i, I, narray$ )
	End for
	$Syn_j \leftarrow Syn \cup$ generated samples by every minority sample in $P'_j$
	End for
	Return generated samples $Syn = Syn_A \cup Syn_B$

Pada penelitian yang telah dilakukan (Chen et al., 2021), menunjukkan hasil penggunaan metode RSMOTE pada berbagai metode klasifikasi seperti Adaboost, Decision Tree, KNN, Linear Regression dan Gradient Boosting memberikan peningkatan nilai pada 5 matrix evaluasi yaitu GMean, F1, AUC, Recall dan Precision. Dari penelitian tersebut penggunaan RSMOTE dan DT pada setiap matrix evaluasi memberikan nilai yang sangat baik dibandingkan dengan versi SMOTE lainnya.

## 2. 6. Edited Nearest Neighbors (ENN)

Edited Nearest Neighbors merupakan algoritma yang digunakan untuk menemukan data ambigu dan data noise yang ada pada sebuah dataset. Algoritma ini menggunakan tiga *nearest neighbors* untuk menemukan data tersebut yang bisa mengakibatkan kesalahan dalam klasifikasi dan kemudian menerapkan 1 *nearest neighbors* untuk membuat keputusan dalam menghapus data tersebut. Algoritma 2.2 menunjukkan alur yang digunakan ENN. Pada penggunaannya dalam metode undersampling, setiap instance pada dataset akan dilakukan komputasi menggunakan *three nearest neighbors*, jika *instance* tersebut adalah kelas mayoritas dan menyebabkan kesalahan dalam klasifikasi maka akan dihapus dari

dataset. Tetapi, jika instance tersebut adalah kelas minoritas dan menyebabkan kesalahan dalam klasifikasi maka nearest neighbors disekitarnya akan dihapus.

Algoritma 2.2 Alur ENN

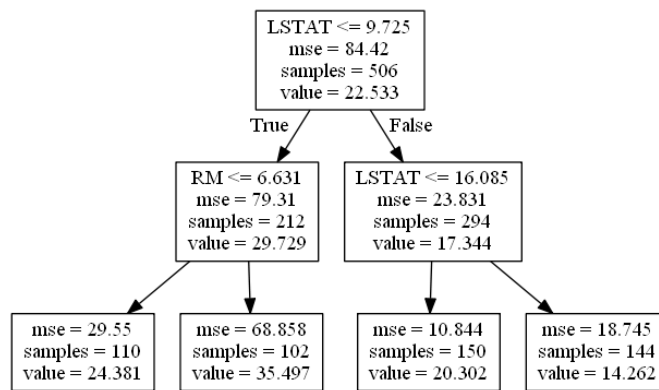
No	Pseudocode
Input: D = dataset Output: D' = cleaned dataset	
1	let X = D
2	let D' = D
3	for each xi in X do
4	if make miskelasified using k-nn
5	if xi is majority kelas
6	discard xi from D'
7	end if
8	if xi is minority
9	discard nearby xi from D'
10	end if
11	end if
12	end for
13	return D'

Pada penelitian (Alejo et al., 2010) melakukan eksperimen menggunakan beberapa dataset dengan *neural network* sebagai training modelnya. Dataset yang digunakan diambil dari *UCI Machine Learning Database Repository* antara lain seperti *Diabetes, German, Heart, Liver, Phoneme, Sonar, Coyo, E-coli, Fetwell dan Satimage*. Dataset tersebut memiliki variasi jumlah kelas, fitur dan sample training maupun test. Penggunaan ENN mampu memberikan 26.1% persentase *size reduction*. Hasilnya metode ENN mampu meningkatkan akurasi saat melakukan klasifikasi menggunakan *neural network*.

## 2. 7. Fitur Seleksi menggunakan *Random Forest* (RF)

RF merupakan salah satu algoritma machine learning yang populer digunakan. Menggunakan algoritma ini mampu menciptakan model dengan performa yang baik dan mudah untuk digunakan. Karena akurasi yang dihasilkan sangat baik, dengan algoritma ini memungkinkan untuk mengetahui tingkat kepentingan sebuah *attribute* pada model yang dibuat. Teknik ini menjadi penentu kualitas model yang dibuat. Dataset dengan semua fitur yang digunakan bisa saja memberikan hasil evaluasi yang tidak terlalu signifikan disbanding dengan penggunaan beberapa fitur yang optimal (Thapa et al., 2020).

RF yang digunakan pada teknik pemilihan fitur akan melakukan ekstraksi fitur secara acak. Kemudian hasil observasi akan mengurangi tingkat *overfitting* pada model. Setiap tree akan hanya memiliki dua percabangan *yes-no* dan hingga mampu membuat kombinasi pertanyaan berdasarkan *tree* yang terhubung. Pembentukan cabang *yes-no* ini dibuat berdasarkan nilai *error* yang dihasilkan seperti nilai MSE atau *Gini Impurity*. Untuk mengurutkan *variable* yang paling penting akan diurutkan berdasarkan seberapa kecil *error* yang muncul Ketika *attribute* atau fitur tersebut digunakan. Sehingga Ketika direpresentasikan akan seperti Gambar 2.4. (Irmina et al., 2020)



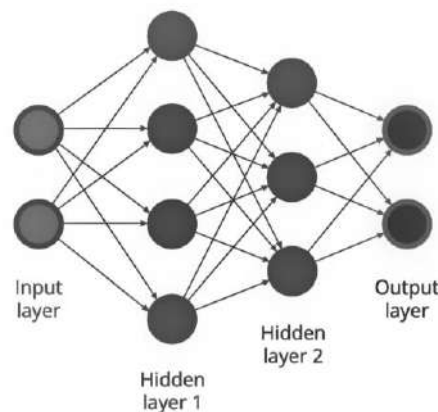
Gambar 2.6 Contoh Alur Metode RF (Sumber: www.deepai.org)

Tetapi teknik ini memiliki beberapa kekurangan, salah satunya fitur-fitur yang berkorelasi bisa memiliki tingkat kepentingan yang sama sehingga memungkinkan fitur untuk tidak terseleksi dan penggunaan RF memberikan fitur dengan kardinalitas yang tinggi pada model yang dibuat. RF mampu memberikan hasil prediksi atau klasifikasi sangat baik pada berbagai permasalahan atau obyek (Irmina et al., 2020). Seperti pada penelitian *fraud detection* (Shamsudin et al., 2020), penelitian tersebut menerapkan RF pada fitur seleksi dan mampu memberikan hasil yang baik pada tiga dataset yang digunakan. Hasil evaluasi akurasi yang dihasilkan sekitar lebih dari 80%.

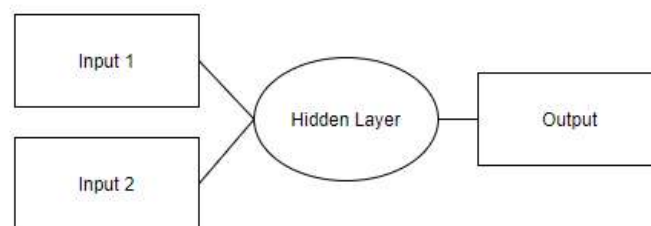
## 2. 8. Artificial Neural Network (ANN)

ANN merupakan algorithm yang membantu dalam mengenali kumpulan data melalui proses peniruan otak manusia. Algorithm ini mampu beradaptasi dengan perubahan input dan mampu memberikan hasil yang lebih baik. ANN merupakan salah satu varian paling sederhana pada neural network karena

pemrosesannya hanya berlangsung satu arah. Setiap inputan akan dilewatkan kedalam berbagai node input hingga menuju node output. Terdapat tiga komponen utama pada algorithm ini, input layer, hidden layer dan output layer seperti yang ditunjukkan pada Gambar 2.5. Pada input layer berfungsi untuk membaca data input dan data yang dibaca kemudian diteruskan pada layer selanjutnya. Hidden layer merupakan bagian yang ada diantara input dan output layer. Lapisan ini akan melakukan transformasi non-linier dari input yang masuk ke dalamnya. Dan output layer merupakan bagian yang menampilkan hasil (Alparslan et al., 2021; M. Mishra & Srivastava, 2014).



Gambar 2.7 Komponen artificial neural network (Sumber: www.medium.com)

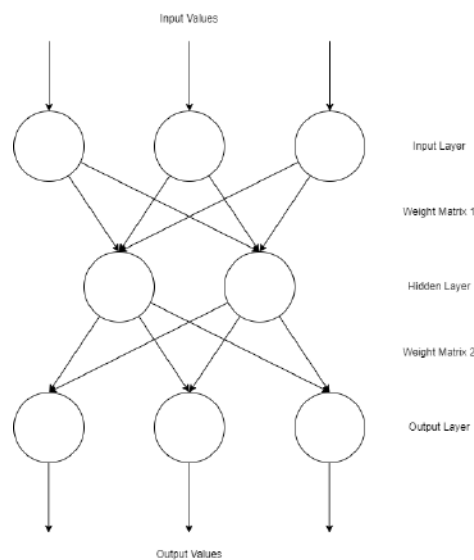


Gambar 2.8 Contoh artificial neural network

Jika terdapat dua input data seperti yang ditunjukkan pada Gambar 2.6 maka input tersebut akan dikalikan dengan bobot yang kemudian dijumlahkan dengan nilai bias seperti ditunjukkan pada rumus 1 kemudian dilakukan aktivasi pada fungsi sehingga input tak terbatas menjadi output yang memiliki bentuk yang dapat diprediksi, umumnya fungsi aktivasi ini menggunakan fungsi sigmoid.

MLP merupakan salah satu pemodelan ANN yang memiliki karakteristik penilaian bobot yang lebih baik sehingga mampu memberikan hasil klasifikasi

yang lebih akurat. Metode ini diperkenalkan pertama kali oleh S. Papert dan M. Minsky pada tahun 1969. Seperti gambar 2.7 merupakan gambaran lapisan yang ada pada MLP. Pada gambar tersebut ditunjukkan bahwa MLP terdiri dari satu input layer, beberapa hidden layer dan satu output layer. Karena terdiri dari beberapa macam layer metode ini masuk kedalam Teknik Deep Learning. Setiap neuron yang ada saling terhubung sehingga neuron tersebut akan memproses semua input yang kemudian menghitung nilai aktivasi (Gardner & Dorling, 1998). Dalam penelitian yang dilakukan oleh (Desai & Shah, 2020) menunjukkan bahwa penerapan MLP memberikah hasil yang tidak jauh berbeda dengan CNN tetapi tetap lebih unggul akurasi yang diberikan ketika menggunakan CNN. Tapi ketika dibandingkan dengan ensembled method seperti Decission Tree (DT), dalam penelitian (Greco et al., 2021) memberikan hasil yang lebih efektif.

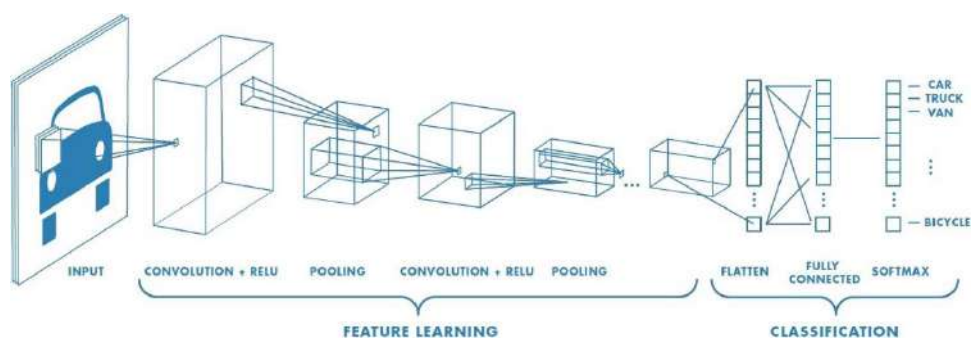


Gambar 2.9 Skema ANN MLP

## 2. 9. Convolutional Neural Network (CNN)

Deep learning merupakan salah satu bagian metode machine learning, yang berpusat dalam pengembangan neural network multiple-layer. Deep learning sering kali berhasil menyelesaikan tugas seperti deteksi objek, pengenalan suara, terjemahan bahasa dan lain sebagainya. Perbedaan deep learning dari machine learning adalah pada tahapan dimana deep learning secara otomatis melakukan representasi data tanpa perlu adanya aturan dan knowledge base. Beberapa algoritma deep learning yang populer seperti Deep Neural Network, Artificial

Neural Network, Recurrent Neural Network dan Convolutional Neural Network (Jian Li et al., 2017). Pada penelitian ini akan menggunakan Convolutional Neural Network dalam pembuatan model prediksi Software Defect. Convolutional Neural Network adalah salah satu jenis neural network yang biasanya sering kali digunakan pada pemrosesan data image. CNN sendiri tidak jauh beda dengan neural network biasanya. Komposisi juga tidak jauh berbeda. Yang membedakan adalah arsitektur dibagi menjadi dua Fully Connected Layer (MLP) dan Feature Extraction Layer seperti ditunjukkan pada Gambar 2.8



Gambar 2.10 Ilustrasi Metode CNN

## 2. 10. Recurrent Neural Network (RNN)

Pada tahun 1980, RNN dikenalkan dan menjadi salah satu algoritma tertua yang dikembangkan. Algoritma ini penting karena mampu menciptakan internal memori tersendiri. Kemudian pada tahun 1990, banyak modifikasi dari RNN salah satunya adalah Long Short-Term Memory (LSTM) (Santana et al., 2018; Sunder et al., 2020). Dengan adanya internal memori, membuat algoritma ini mampu mengingat input dan memberikan output yang sesuai karena mampu memprediksi hal hal berikutnya. Sehingga algoritma ini sering digunakan dalam deret waktu, mesin pencarian baik berupa teks, audio dan video. Ide di balik terciptanya algoritma ini adalah bagaimana cara sebuah struktur data sekuensial yang ada dapat dieksploitasi secara keseluruhan. Sehingga satu data yang digunakan sebagai input akan diproses secara berulang.

Cara kerja dari algoritma ini adalah dengan melakukan pengolahan input dan kemudian akan diproses pada berbagai informasi yang sudah pernah diperoleh sebelumnya. Sehingga hasil yang ditentukan akan dipengaruhi dengan informasi yang sudah pernah ada dikarenakan memori internal yang ada pada algoritma ini.

Data input yang diterima tidak hanya melewati satu siklus seperti algoritma lain, tetapi akan berada pada sebuah *loop* yang mengandung beberapa informasi yang telah didapatkan sebelumnya. Salah satu penerapan pada algoritma ini yang bisa dirasakan adalah pada rekognisi suara pada produk Apple dan Google.

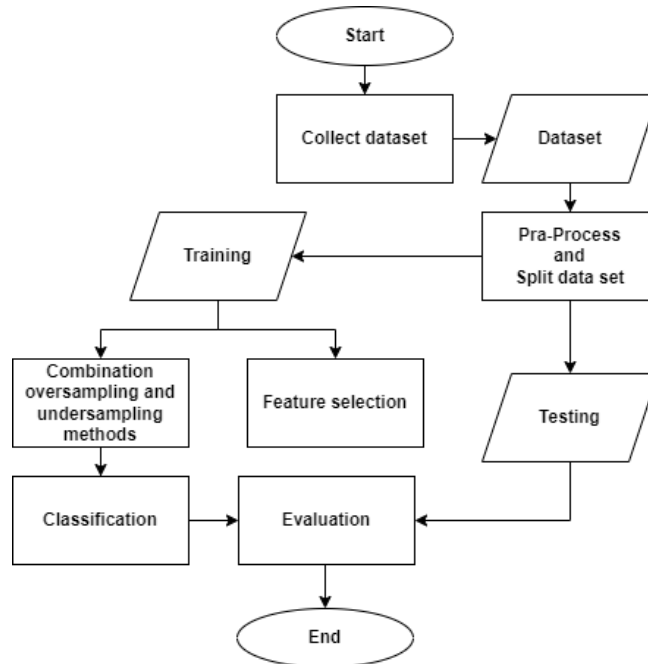
Adapun beberapa yang perlu diperhatikan dalam menggunakan algoritma ini, seperti *hidden\_state* yang digunakan sebagai memori dari jaringan yang dibentuk. Terkadang *hidden\_state* yang digunakan dapat menyebabkan hilangnya informasi saat melakukan prediksi karena banyak sekali step yang telah dilalui. Selain itu jika dibandingkan dengan deep neural network, layer yang digunakan dapat dibentuk secara berbeda. sehingga RNN akan menjalankan Langkah yang sama pada setiap stepsnya dan membagi sama rata terhadap parameter yang digunakan. dan yang terakhir merupakan perlu memastikan kembali hasil yang didapatkan. Hal ini dikarenakan karena Langkah yang sama dilakukan secara berulang dan input yang digunakan merupakan input hasil dari tahapan yang dilakukan sebelumnya.



*[Halaman ini sengaja dikosongkan]*

## BAB 3 METODOLOGI PENELITIAN

Bab ini akan memaparkan tentang metodologi penelitian yang digunakan pada penelitian ini,



Gambar 3.1 Alur Metodologi Penelitian

Penjelasan tahapan metode penelitian pada Gambar 3.1 akan diterangkan secara terperinci pada sub bab berikut.

### 3. 1. Pengumpulan Dataset

Data yang digunakan pada penelitian terkait dengan *software defect*. Data tersebut antara lain NASA MDP dan AEEEM yang bisa bersifat *open access*. NASA MDP berisikan *metrics* based on complexity perangkat lunak, metric tersebut antara lain ditunjukkan pada Table 3.1.

Tabel 3.1 Metric used on MDP Dataset

Code Metrics	Metric Name	Description
Halstead	HALSTEAD-CONTENT	Berisikan nilai pengukuran pada perangkat lunak dan relasinya.
	HALSTEAD-EFFORT	
	HALSTEAD ERROR-SET	
	HALSTEAD-DIFFICULTY	

	NUM-OPERATORS	
	NUM-UNIQUE-OPERATORS	
	NUM-UNIQUE-OPERANDS	
	NUM-OPERANDS	
McCabe	CYCLOMATIC-COMPLEXITY	Menunjukkan tingkat kompleksitas pada sebuah program.
	LOC-TOTAL	
	ESSENTIAL-COMPLEXITY	
	DESIGN-COMPLEXITY	
Line of Code	LOC-COMMENTS	Berisikan seluruh jumlah baris yang ada pada proyek
	HALSTEAD-VOLUME	
	HALSTEAD-LEVEL	
	LOC-CODE-AND-COMMENT	
	LOC-BLANK	
	BRANCH-COUNT	

Dengan adanya metric tersebut akan membantu dalam melakukan proses prediksi *software defect*. Beberapa dataset yang digunakan antara lain ditunjukkan pada Table 3.2,

Tabel 3.2 Daftar MDP dataset yang digunakan

<b>Name</b>	<b>Total Attributes</b>	<b>Total Instances</b>	<b>Total Non-Defect</b>	<b>Total Defect</b>
PC1	40	1107	1030	77
PC4	40	1458	1280	178
PC5	39	17186	16670	516
AR1	31	121	112	9
AR3	31	63	55	8
AR4	31	107	87	20
KC1	21	2109	1783	326
KC2	22	522	415	107
KC3	40	194	158	36
MC1	39	1988	1942	46

MC2	40	125	81	44
JM1	21	10878	8776	2102
CM1	40	498	449	49
MW1	40	253	226	27

Berdasarkan Tabel 3.2, terdapat 13 dataset yang digunakan dan setiap dataset memiliki jumlah *attribute* serta instances yang berbeda. Data pada setiap *attribute* bertipe nominal. Untuk *attribute* yang ada didalam dataset kurang lebih seperti yang ditunjukkan pada Tabel 3.1. Karena dataset yang digunakan bertipe binary, sehingga hanya terdapat 2 kelas yaitu *defect* dan *non-defect*. Data tersebut tergabung dari beberapa metrik dan belum dilakukan aktivitas pre-processing sehingga dapat diketahui beberapa permasalahan yang timbul dari data tersebut, seperti ketidakseimbangan kelas dan juga fitur yang dependensi.

Dataset AEEEM yang digunakan pada penelitian ini berjumlah 5 proyek yang ditunjukkan pada Tabel 3.3. Setiap dataset tersebut memiliki jumlah *attribute* 61 dengan berbagai jumlah data yang ada. Attributes yang ada pada dataset tersebut ditunjukkan pada table 3.4

Tabel 3.3 Dataset AEEEM yang digunakan

No	Proyek	Total Attributes	Total Instances	Total Non-defect	Total Defect
1	EQ	61	324	195	129
2	JDT	61	997	791	206
3	Lucene	61	691	627	64
4	Mylyn	61	1862	1617	245
5	PDE	61	1497	1288	209

Tabel 3.4 *Attributes* dari dataset AEEEM

<i>Attributes</i>	
ck_oo_numberOfPrivateMethods	WCHU_dit
LDHH_lcom	ck_oo_lcom
LDHH_fanIn	WCHU_numberOfAttributesInherited
numberOfNonTrivialBugsFoundUntil:	ck_oo_rfc

WCHU_numberOfPublicAttributes	LDHH_wmc
WCHU_numberOfAttributes	LDHH_numberOfAttributes
CvsWEntropy	LDHH_numberOfLinesOfCode
LDHH_numberOfPublicMethods	WCHU_fanOut
WCHU_fanIn	WCHU_lcom
LDHH_numberOfPrivateAttributes	ck_oo_cbo
CvsEntropy	WCHU_rfc
LDHH_numberOfPublicAttributes	ck_oo_numberOfAttributes
WCHU_numberOfPrivateMethods	numberOfHighPriorityBugsFoundUntil:
WCHU_numberOfMethods	ck_oo_numberOfPrivateAttributes
ck_oo_numberOfPublicAttributes	numberOfMajorBugsFoundUntil:
ck_oo_noc	WCHU_numberOfPublicMethods
numberOfCriticalBugsFoundUntil:	LDHH_dit
ck_oo_wmc	WCHU_cbo
LDHH_numberOfPrivateMethods	CvsLinEntropy
WCHU_numberOfPrivateAttributes	WCHU_numberOfMethodsInherited
CvsLogEntropy	numberOfBugsFoundUntil:
WCHU_noc	LDHH_fanOut
LDHH_numberOfAttributesInherited	LDHH_numberOfMethodsInherited
WCHU_wmc	LDHH_rfc
ck_oo_fanOut	ck_oo_numberOfMethodsInherited
ck_oo_numberOfLinesOfCode	ck_oo_numberOfPublicMethods
ck_oo_numberOfAttributesInherited	LDHH_cbo
ck_oo_numberOfMethods	WCHU_numberOfLinesOfCode
ck_oo_dit	CvsExpEntropy
ck_oo_fanIn	LDHH_numberOfMethods
LDHH_noc	

### 3. 2. Praproses dan Pembagian Dataset

Pra-proses merupakan tahapan untuk memastikan data yang dimiliki telah layak untuk digunakan. Pertama yang dilakukan adalah kita lakukan cleaning data dengan mencari apakah terdapat missing value pada instances yang ada. Kemudian untuk lakukan scalling menggunakan Teknik scaling sehingga data yang dimiliki

menjadi data yang berada pada rentan 0-1. Pada penelitian ini untuk scaling menggunakan metode min max.

Tabel 3.5 Contoh Hasil Normalisasi Pada Dataset MDP

No	LOC_BLANK	LOC BLANK'
1	16	1
2	2	0.067
3	1	0
4	8	0.467
5	1	0

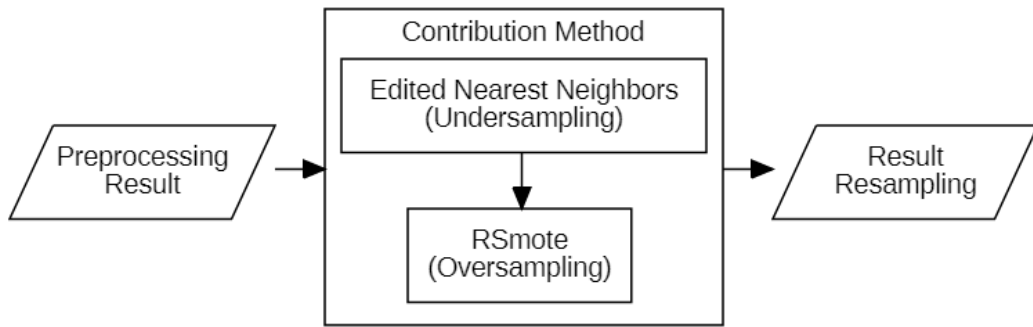
Pada Tabel 3.5 merupakan contoh data dan hasil proses normalisasi menggunakan metode *min-max*. Setiap kolom akan dicarikan nilai maksimal dan nilai minimal kemudian nilai tersebut akan digunakan untuk melakukan perhitungan pada setiap baris data pada kolom LOC\_BLANK yang rumusnya ditunjukkan pada nomor 3.1

$$y = (x - \min) / (\max - \min) \tag{3.1}$$

Setelah dataset dinormalisasi menggunakan rumus 3.1, dataset akan dibagi menjadi dua bagian, data 80 persen training dan 20 persen data testing. Data training akan digunakan untuk implementasi kombinasi *oversampling* dan *undersampling*, sedangkan untuk data testing akan digunakan sebagai pembanding saat melakukan klasifikasi dan evaluasi pada penelitian ini. Tujuan dilakukan *splitting* diawal sebelum melakukan over/undersampling untuk menghindari data yang direkayasa oleh metode over/undersampling menjadi data test, jika hal tersebut terjadi memungkinkan data yang diprediksi akan menjadi tidak realistis.

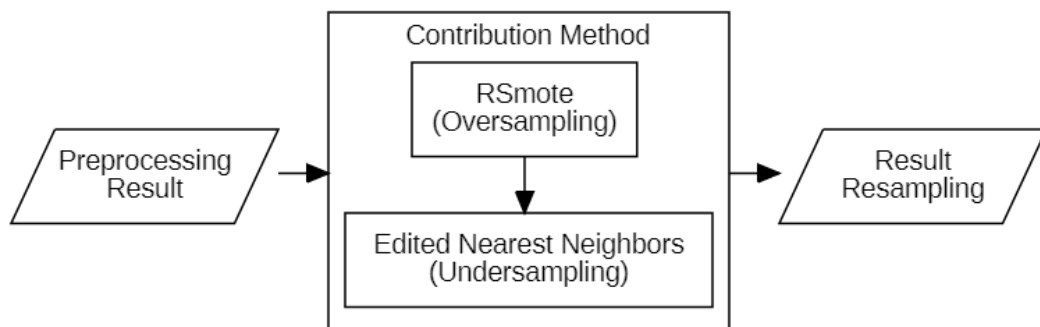
### 3. 3. Model Kombinasi *Oversampling* dan *undersampling*

Tahapan ini akan melakukan proses oversampling dan undersampling menggunakan dua teknik oversampling dan undersampling. Untuk Teknik oversampling yang digunakan adalah RSmote yang merupakan pengembangan terbaru dari SMOTE dan teknik undersampling menggunakan ENN. Penelitian ini menciptakan tiga model kombinasi yang ditunjukkan pada Gambar 3.2 hingga 3.4.



Gambar 3.2 Combination Undersampling-Oversampling (FCM)

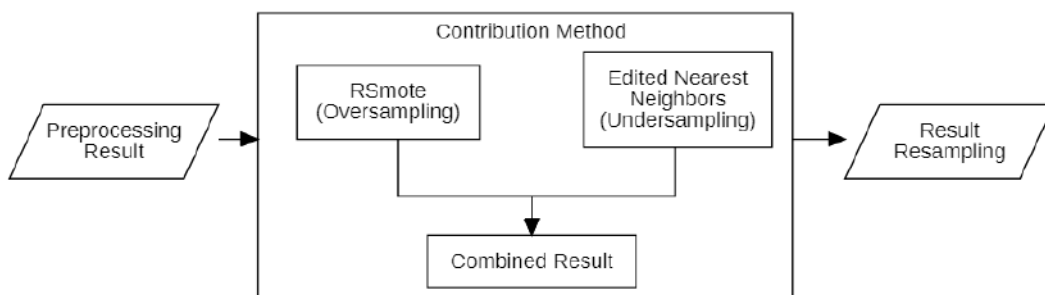
Model FCM (First Combination Model) yang merupakan model pertama yang ditunjukkan pada Gambar 3.2, model tersebut akan menjalankan teknik undersampling terlebih dahulu kemudian menjalankan oversampling. Hal ini dilakukan untuk mencari jawaban kenapa penelitian yang ada tidak melakukan undersampling terlebih dahulu kemudian menerapkan teknin oversampling. Berdasarkan penelitian (Shamsudin et al., 2020) dia menggunakan model pada Gambar 3.2 dalam penerapan kombinasi metode *oversampling* dan *undersampling*. Tujuan dalam penelitian tersebut adalah untuk menghilangkan data yang bisa menyebabkan kesalahan pada data aslinya menggunakan *undersampling* kemudian dilakukan penyeimbangan menggunakan *oversampling*. Dari hasil penelitian tersebut menunjukkan hasil yang lebih baik dengan penerapan kedua kombinasi tersebut sehingga penelitian ini tertarik untuk membandingkan model ini dengan model lainnya.



Gambar 3.3 Combination Oversampling-Undersampling (SCM Model)

Model SCM (*Second Combination Model*) merupakan model kedua yang ditunjukkan pada Gambar 3.2, model tersebut akan menjalankan teknik oversampling terlebih dahulu kemudian menjalankan undersampling. Tujuannya

untuk melakukan sampling pada kelas minoritas sehingga pada kasus binary klasifikasi, kelas pada dataset menjadi seimbang. Kemudian undersampling bertujuan untuk membersihkan data yang terindikasi noisy dan duplicated dengan cara menghapus atau menggabungkan pada data dengan kelas mayoritas. Berdasarkan penelitian (Park & Park, 2020) berhasil melakukan kombinasi metode oversampling dan undersampling dengan model seperti Gambar 3.2. pada penelitian tersebut dia menerapkan oversampling beserta validasi kemudian jika sudah tercapai maka akan dilakukan undersampling hingga overfitting muncul. Hasil dari penelitian tersebut mampu memberikan hasil evaluasi terbaik dibandingkan dengan metode oversampling ataupun undersampling lainnya seperti smote, b-smote, adasyn dan Tabular GAN.



Gambar 3.4 Merge of Oversampling and Undersampling (TCM Model)

Model TCM (*Third Combination Model*) merupakan model ketiga yang ditunjukkan pada Gambar 3.4. model ini akan menjalankan dua teknik oversampling dan undersampling secara terpisah. Hasil kedua teknik tersebut akan digabungkan dan menghasilkan jumlah data yang lebih banyak dibandingkan dengan jumlah data dari original data. Untuk penelitian selanjutnya kemungkinan untuk model ini tidak dilanjutkan karena bisa menyebabkan berbagai macam permasalahan selain itu metode penggabungan seperti dijelaskan tidak ditemukan referensi dimanapun. Tetapi bisa jadi skema untuk model tiga akan diubah seperti penelitian yang dilakukan oleh (Koziarski, 2021). Pada penelitian tersebut melakukan penerapan metode oversampling hanya pada kelas minoritas dan undersampling hanya pada kelas mayoritas. Skema tersebut nantinya dapat menutupi kesalahan pada perencanaan model ketiga sebelumnya.



### 3. 4. Fitur Seleksi

Penggunaan fitur seleksi bertujuan untuk mendapatkan *attribute* yang relevant dan mempermudah dalam melakukan klasifikasi. Selain itu, fitur seleksi mampu mengurangi waktu *training* pada suatu model klasifikasi, meningkatkan akurasi dan juga mengurangi adanya *overfitting* saat melakukan prediksi atau klasifikasi. Metode fitur seleksi yang digunakan pada penelitian ini adalah *Random Forest* (RF). RF merupakan *bagging algorithm* yang mengumpulkan sejumlah *decision tree*. Setiap tree berisikan node-node saling terhubung satu arah dan setiap ujung node yang dihasilkan RF berisikan informasi terpenting pada serangkaian dataset.

Pada penelitian ini setiap dataset akan diterapkan perilaku yang sama. Setiap hasil penilaian yang dilakukan menggunakan metode RF akan diurutkan dari skor tertinggi. Kemudian dari urutan tersebut nantinya akan diambil beberapa fitur yang digunakan untuk melakukan klasifikasi.

### 3. 5. Klasifikasi atau Pembuatan Model Prediksi

Tahapan ini akan dilakukan pembuatan model klasifikasi untuk melakukan prediksi pada *software defect*. Model klasifikasi dihasilkan menggunakan metode *deep learning*. Metode yang digunakan adalah *Recurrent Neural Network*, *Convolutional Neural Network* dan *Artificial Neural Network*. Penggunaan deep learning dikarenakan pada penelitian (Jian Li et al., 2017; Santana et al., 2018; Zhang et al., 2020) menunjukkan metode tersebut mampu memberikan hasil yang lebih baik dibandingkan *ensembling method* pada *machine learning* ketika digunakan untuk melakukan aktifitas prediksi. Adapun konfigurasi yang digunakan pada model prediksi yang digunakan pada penelitian ini ditunjukkan pada Tabel 3.6 sebagai berikut.

Tabel 3.6 Konfigurasi Model Prediksi *Deep Learning*

No	Metode	Konfigurasi
1	ANN (Desai & Shah, 2020)	optimizer: 'adam' loss: 'binary_crossentropy' batch_size: 10 epochs: 25 layer_1:

		<p>type: Dense  units: 15  kernel_initializer: 'glorot_uniform',  activation = 'relu'</p> <p>layer_2:  type: Dense  units: 8  kernel_initializer: 'glorot_uniform',  activation = 'relu'</p> <p>layer_3:  type: Dense  units: 5  kernel_initializer: 'glorot_uniform',  activation = 'relu'</p> <p>layer_4:  type: Dense  units: 1  kernel_initializer: 'glorot_uniform',  activation = 'sigmoid'</p>
2	CNN (Singh, 2020)	<p>optimizer: 'adam',  loss: 'binary_crossentropy',  batch_size: 10,  epochs: 25,  layer_1:  type: Dense  units: 64,  kernel_size: 1,  activation: 'relu'</p> <p>layer_2:  type: Dense  units: 32,  kernel_size: 1,  activation: 'relu'</p> <p>layer_3:  type: Dense  units: 16,  kernel_size: 1,  activation = 'relu'</p> <p>layer_4: flatten</p> <p>layer_5:  type: Dense  units: 8,  activation = 'relu'</p> <p>layer_6:  type: Dense  units: 1,  activation = 'sigmoid'</p>

3	RNN (Ahamasaleh, 2022)	optimizer: 'rmsprop', loss: 'mse', batch_size: 10, epochs: 25, layer_1: type: LSTM units: 80 recurrent_activation: 'hard_sigmoid', activation: 'tanh' layer_2: type: Dense units: 1 activation: 'tanh'
---	---------------------------	--

Seperti yang ditunjukkan pada Tabel 3.6, ketiga model prediksi diterapkan epochs dan batch\_size yang sama dengan nilai 25 dan 10. Pada model ANN, diinisiasi dengan empat buah *layer* dengan jumlah unit setiap layernya yaitu 15, 8, 5 dan 1. Aktifasi pada layer pertama, kedua dan ketiga menggunakan *relu* yaitu Rectified Linear Unit yang mampu mempercepat proses konvergensi yang dilakukan dengan membuat pembatas (*threshold*) pada bilangan nol. Sedangkan untuk layer keempat sebagai layer output menggunakan *sigmoid* dan setiap layer yang ada mendapatkan pengaturan tambahan pada kernel menggunakan *glorot\_uniform* yang merupakan pendekatan standart untuk inisiasi berat sebuah *layer* pada *neural network* (Gardner & Dorling, 1998; Massoudi et al., 2021; M. Mishra & Srivastava, 2014).

Pengaturan pada model CNN mengikuti referensi pada *public repository* github. Terdapat enam *layer* yang digunakan pada model ini. Model FCM, SCM hingga TCM memiliki pengaturan yang sama dengan ukuran kernel sebesar satu dan aktifasi menggunakan *relu*. Untuk ukuran unit yang digunakan pada *layer* pertama hingga kedua sebesar 64, 32 dan 16. Pada *layer* keempat dibentuk menggunakan *flatten* bertujuan untuk data yang dihasilkan pada *layer* sebelumnya akan dibentuk menjadi 1-dimensi yang disebut dengan *fully-connected layer*. *Layer* tersisa yaitu kelima dan keenam memiliki ukuran unit sebesar 8 dan 1, dengan aktifasi pada *layer* kelima menggunakan *relu* dan *layer* terakhir sebagai *layer output* menggunakan *sigmoid* (Singh, 2020).

Pada model prediksi ketiga yaitu RNN (Ahamasaleh, 2022; Sunder et al., 2020). Model ini dibentuk menggunakan optimasi *rmsprop* yang adalah *Propagasi Root Mean Square*, bertujuan untuk mempertahankan *learning rate* per-parameter yang diadaptasi berdasarkan rata-rata besaran gradien terbaru untuk suatu bobot. Terdapat dua layer yang digunakan pada model ini. Pada *layer* pertama dibuat menggunakan LSTM dan *layer* kedua menggunakan Dense. Untuk jumlah unit yang digunakan pada *layer* pertama bernilai 80 dan *layer* kedua bernilai 1. Aktifasi yang digunakan pada masing masing *layer* adalah *tanh* yang merupakan *sigmoidal* yang karakteristiknya seperti *logistic sigmoid* tetapi lebih baik.

### 3. 6. Analisis Hasil

Data yang dihasilkan dari aktivitas prediksi menggunakan neural network berdasarkan hasil kombinasi oversampling dan undesampling dapat dianalisis hasilnya. Pada penelitian ini cara yang digunakan untuk mengetahui hasilnya dengan cara:

1. Melakukan analisis hasil evaluasi *Receiver Operating Curve (ROC)*, dengan ini mampu menghasilkan nilai prediksi yang bisa digunakan untuk melakukan ranking. ROC digunakan untuk menunjukkan *tradeoff* antara *sensitivity* dan *specifity*
2. Melakukan analisis hasil evaluasi *F-Measure*. Tujuannya untuk mendapatkan nilai Precision dan Recall pada satu hitungan.
3. Akurasi prediksi

Evaluasi ROC dilakukan dengan cara memperhitungkan *confusion matrix* antara tiap rate false positif dan rate true positif. Dari hasil perhitungan tersebut akan didapatkan jika kurva mendekati titik [0, 1] maka model yang digunakan memiliki kinerja yang bagus sedangkan mendekati titik [0, 0] maka model tersebut menghasilkan kinerja yang kurang maksimal.

Untuk perhitungan *F-Measure* akan dilakukan dengan mendapatkan nilai *Precision* dan recall dari Persamaan 3.3 dan 3.4. Kedua nilai dari persamaan tersebut akan dikalikan untuk mendapatkan nilai  $x$ , kemudian kedua nilai persamaan tersebut dijumlahkan untuk mendapatkan nilai  $y$ . Dari nilai  $x$  dan  $y$  yang telah didapatkan kemudian akan dilakukan pembagian  $x$  dibagi  $y$ , hasil pembagian

kemudian dikalikan dengan angka dua seperti yang ditunjukkan pada Persamaan 3.2. x

$$F = 2 \frac{Precision \times Recall}{Precision + Recall} \quad (3.2)$$

Nilai *precision* didapatkan dengan menghitung ketepatan informasi yang diminta dengan jawaban yang diberikan oleh system. Untuk dapat menghitungnya diperlukan data ratio prediksi benar positif ( $TP'$ ) yang dibandingkan dengan keseluruhan hasil yang bersifat positif ( $TP + FP$ ). Perhitungan tersebut ditunjukkan pada Persamaan 3.3

$$Precision = \frac{TruePositif (TP')}{TruePositif (TP) + FalsePositif (FP)} \quad (3.3)$$

Nilai *recall* didapatkan dengan menghitung jumlah sample positif ( $TP'$ ) dibagi dengan total sample dalam testing set berkategori positif ( $TP + FN$ ) yang ditunjukkan pada Persamaan 3.4.

$$Recall = \frac{TruePositif (TP')}{TruePositif (TP) + FalseNegatif (FN)} \quad (3.4)$$

*[Halaman ini sengaja dikosongkan]*

## BAB 4

### HASIL DAN PEMBAHASAN

Bab ini dijelaskan tentang hasil implementasi, uji coba dan evaluasi pada metode yang diusulkan pada penelitian ini. Terdapat tiga pokok sub bahasan yang ada pada bab ini yaitu implementasi penelitian, perancangan uji coba dan analisis hasil

#### 4. 1. Implementasi Model Kombinasi (FCM, SCM dan TCM)

Hasil oversampling dan undersampling pada penelitian ini dapat dilihat pada Tabel 4.1 hingga Tabel 4.3. Hasil tersebut didapatkan setelah dataset dilakukan *preprocessing* sehingga mendapatkan nilai yang *normalized*. Kemudian hasil data *normalized* digunakan dalam melakukan oversampling dan undersampling pada ketiga model yang diusulkan pada penelitian ini.

Pada Tabel 4.1 merupakan hasil kombinasi oversampling dan undersampling menggunakan model FCM. Model tersebut menerapkan undersampling terlebih dahulu pada semua dataset yang digunakan pada penelitian ini untuk menghilangkan data *noise* atau ambigu sehingga lingkup masing masing kelas pada dataset menjadi lebih sempit. Kemudian hasil undersampling digunakan untuk melakukan oversampling untuk mendapatkan keseimbangan jumlah data pada masing masing kelas. Dari sembilan belas dataset yang digunakan, dataset MC2 dari MDP dan EQ dari AEEEM ketika dilakukan oversampling dan undersampling menggunakan FCM, jumlah data yang dihasilkan menjadi lebih sedikit. Untuk dataset MC2 mengalami penyusutan sebesar 30,71% dengan total data asli sebanyak 127 menjadi 88 data dan dataset EQ mengalami penyusutan sebesar 20,37% dengan total data asli sebanyak 324 menjadi 257 data. Dilihat keseluruhan dataset hasil oversampling dan undersampling yang dihasilkan dari penggunaan FCM, memberikan jumlah yang seimbang antar kelas yang ada pada dataset tersebut.

Tabel 4.1 Hasil Oversampling dan undersampling dengan FCM

Dataset	Jumlah Data						Persentase (%)
	Original	D	ND	NEW	D'	ND'	
PC1	759	61	698	1208	604	604	59,16

PC4	1399	178	1221	1952	976	976	39,53
PC5	17001	503	16498	32050	16025	16025	88,52
AR1	121	9	112	188	94	94	55,37
AR3	63	8	55	88	44	44	39,68
AR4	107	20	87	118	59	59	10,28
KC1	2096	325	1771	2844	1422	1422	35,69
KC2	522	107	415	610	305	305	16,86
KC3	200	36	164	228	114	114	14,00
MC1	9277	68	9209	18302	9151	9151	97,28
MC2	127	44	83	88	44	44	-30,71
CM1	344	42	302	434	217	217	26,16
JM1	9593	1759	7834	10506	5253	5253	9,52
MW1	264	27	237	378	189	189	43,18
EQ	324	129	195	258	129	129	-20,37
JDT	997	206	791	1230	615	615	23,37
Lucene	691	64	627	1086	543	543	57,16
Mylyn	1862	245	1617	2544	1272	1272	36,63
PDE	1497	209	1288	2040	1020	1020	36,27
<b>D: Defect    ND: Non-Defect    D': New Defect    ND': New Non-Defect</b>							

Berdasarkan data uji coba yang terdiri dari 14 dataset MDP dan 5 dataset AEEM didapatkan berbagai macam variasi jumlah data dan selisih perbedaan antara kelas *defect* dan *non-defect* dengan jumlah data mulai dari yang terkecil sebanyak 127 data hingga paling terbesar sebanyak 17001 data dan jumlah tersebut belum dibagi menjadi data training dan data testing. Untuk semua dataset MDP memiliki karakteristik yang sama dimana fitur yang disediakan sama dan begitu juga semua dataset AEEM tetapi MDP dan AEEM memiliki fitur yang berbeda sehingga hasil yang diberikan bisa saja akan berbeda.

Pada dataset PC1 ketika dilakukan perbandingan keenam diagram distribusi seperti pada Gambar 4.1 didapati ketika dilakukan penerapan metode RSMOTE dan SCM data yang memungkinkan memiliki informasi penting tetap dipertahankan seperti yang ditunjukkan pada lingkaran hijau. Akan tetapi tidak terbatas hanya pada lingkaran hijau, bisa diamati data yang dihasilkan lebih

Pada Tabel 4.2 merupakan hasil oversampling dan undersampling menggunakan model SCM. Model tersebut menerapkan oversampling terlebih dahulu untuk menghasilkan keseimbangan data antar kelas yang ada pada dataset kemudian hasil oversampling digunakan untuk melakukan undersampling. Tujuan



untuk melakukan undersampling setelah melakukan oversampling untuk membersihkan noise data dan duplikasi data yang ada pada dataset baik yang dihasilkan setelah oversampling atau data yang sudah ada pada dataset. Dari sembilan belas dataset yang digunakan, dataset EQ dari AEEEM ketika dilakukan oversampling dan undersampling menggunakan model SCM, jumlah data yang dihasilkan menjadi lebih sedikit. Dataset EQ mengalami penyusutan sebesar 8,95% dengan total data asli sebanyak 324 menjadi 295 data.

Dari keseluruhan dataset hasil oversampling dan undersampling yang dihasilkan dari penggunaan model SCM, tidak memberikan jumlah yang sama antar kelas seperti penggunaan model FCM. Tetapi, selisih jumlah data yang dihasilkan tidak terlalu besar. Jika diamati pada Tabel 4.2 hanya kelas Defective yang mengalami perubahan.

Tabel 4.2 Hasil Oversampling dan undersampling dengan Model SCM

Dataset	Jumlah Data						Persentase (%)
	Original	D	ND	NEW	D'	ND'	
PC1	759	61	698	1345	647	698	77,21
PC4	1399	178	1221	2316	1095	1221	65,55
PC5	17001	503	16498	32834	16336	16498	93,13
AR1	121	9	112	208	96	112	71,90
AR3	63	8	55	108	53	55	71,43
AR4	107	20	87	156	69	87	45,79
KC1	2096	325	1771	3297	1526	1771	57,30
KC2	522	107	415	746	331	415	42,91
KC3	200	36	164	299	135	164	49,50
MC1	9277	68	9209	18350	9141	9209	97,80
MC2	127	44	83	133	50	83	4,72
CM1	344	42	302	562	260	302	63,37
JM1	9593	1759	7834	14196	6362	7834	47,98
MW1	264	27	237	448	211	237	69,70
EQ	324	129	195	295	100	195	-8,95
JDT	997	206	791	1459	668	791	46,34
Lucene	691	64	627	1210	583	627	75,11
Mylyn	1862	245	1617	3050	1433	1617	63,80
PDE	1497	209	1288	2417	1129	1288	61,46
<b>D: Defect</b>		<b>ND: Non-Defect</b>		<b>D': New Defect</b>		<b>ND': New Non-Defect</b>	

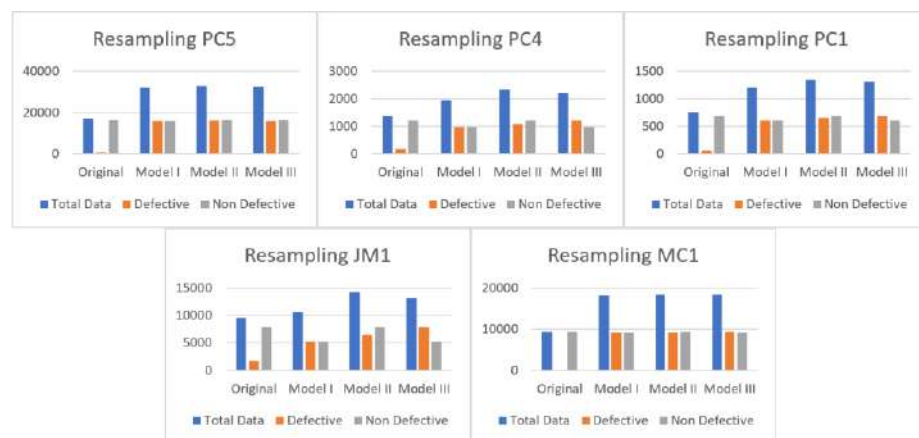
Pada Tabel 4.3 merupakan hasil oversampling dan undersampling menggunakan model TCM. Model tersebut melakukan dua proses bersamaan berupa undersampling dan oversampling pada suatu dataset. Dari kedua proses yang dilakukan kemudian akan mengambil semua data kelas mayoritas pada hasil undersampling dan mengambil semua data kelas minoritas pada hasil oversampling. Kedua data baru yang dihasilkan tersebut kemudian digabungkan menjadi satu. Hasil oversampling dan undersampling tersebut menghasilkan data kelas mayoritas yang lebih bersih dari data *noise* dan pada kelas minoritas memberikan data baru yang lebih rapat persebarannya. Dari sembilan belas dataset yang digunakan, dataset MC2 dari MDP dan EQ dari AEEEM ketika dilakukan oversampling dan undersampling menggunakan model FCM, jumlah data yang dihasilkan menjadi lebih sedikit. Untuk dataset MC2 mengalami penyusutan sebesar 3,94% dengan total data asli sebanyak 127 menjadi 122 data dan dataset EQ mengalami penyusutan sebesar 4,94% dengan total data asli sebanyak 324 menjadi 308 data.

Tabel 4.3 Hasil Oversampling dan undersampling dengan Model TCM

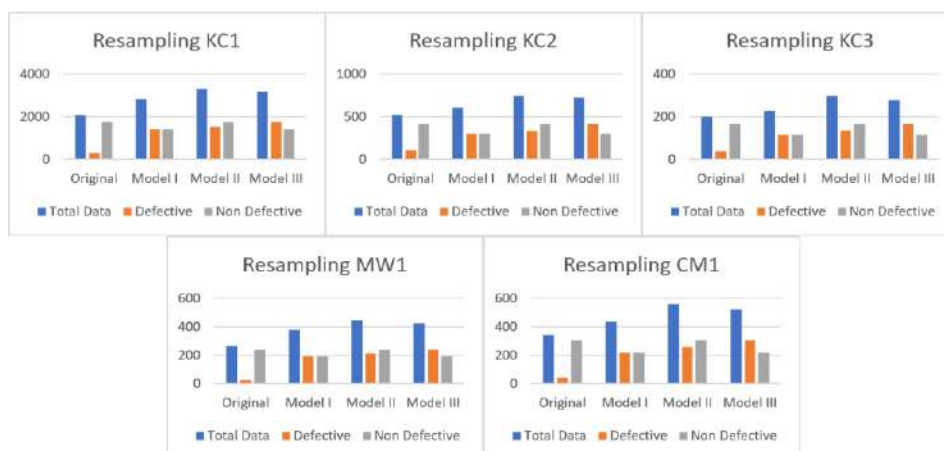
Dataset	Jumlah Data						Persentase (%)
	Original	D	ND	NEW	D'	ND'	
PC1	759	61	698	1302	698	604	71,54
PC4	1399	178	1221	2197	1221	976	57,04
PC5	17001	503	16498	32523	16025	16498	91,30
AR1	121	9	112	206	94	112	70,25
AR3	63	8	55	99	44	55	57,14
AR4	107	20	87	146	59	87	36,45
KC1	2096	325	1771	3193	1771	1422	52,34
KC2	522	107	415	720	415	305	37,93
KC3	200	36	164	278	164	114	39,00
MC1	9277	68	9209	18360	9209	9151	97,91
MC2	127	44	83	122	83	39	-3,94
CM1	344	42	302	519	302	217	50,87
JM1	9593	1759	7834	13087	7834	5253	36,42
MW1	264	27	237	426	237	189	61,36
EQ	324	129	195	308	195	113	-4,94
JDT	997	206	791	1406	791	615	41,02
Lucene	691	64	627	1170	543	627	69,32
Mylyn	1862	245	1617	2889	1272	1617	55,16

PDE	1497	209	1288	2308	1020	1288	54,18
<b>D: Defect</b>	<b>ND: Non-Defect</b>	<b>D': New Defect</b>	<b>ND': New Non-Defect</b>				

Pada Gambar 4.1 hingga 4.4 menunjukkan grafik perubahan dataset hasil oversampling dan undersampling terdiri dari data original, data hasil oversampling dan undersampling menggunakan model FCM, data hasil oversampling dan undersampling menggunakan model SCM dan hasil oversampling dan undersampling menggunakan model TCM. Setiap grafik menunjukkan total data, total data defective dan total data non defective. Dapat diamati ketika diterapkan model kombinasi oversampling dan undersampling, setiap dataset mengalami peningkatan jumlah data dan kelas yang dihasilkan menjadi lebih seimbang dibanding dengan data original.



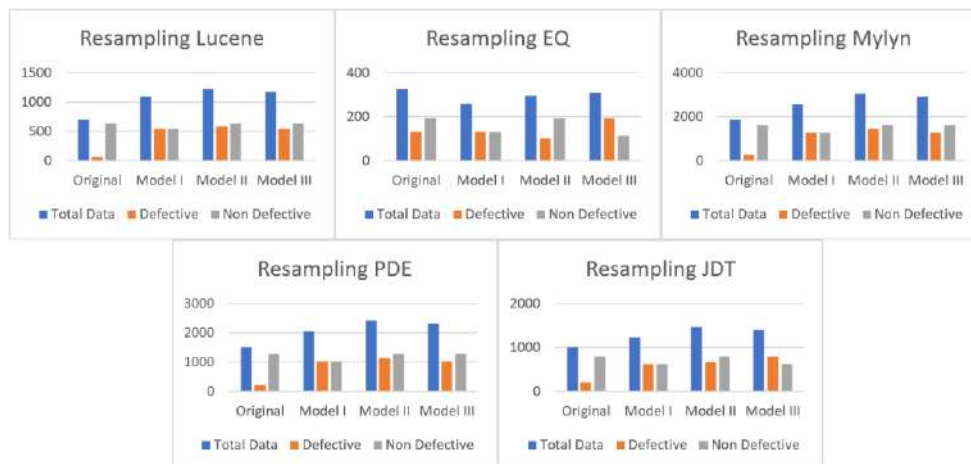
Gambar 4.1 Perbandingan dataset PC5, PC4, PC1, JM1 dan MC1



Gambar 4.2 Perbandingan dataset KC1, KC2, KC3, MW1 dan CM1



Gambar 4.3 Perbandingan dataset AR1, AR2, AR4 dan MC2



Gambar 4.4 Perbandingan dataset Lucene, EQ, Mylyn, PDE dan JDT

Terdapat perbedaan yang cukup signifikan ketika metode yang diusulkan diterapkan pada 19 dataset yang digunakan pada penelitian ini. Seperti yang ditunjukkan pada Tabel 4.1, merupakan hasil implementasi metode ENN, RSMOTE dan ketiga model usulan berupa kombinasi *oversampling* dan *undersampling* menggunakan RSMOTE dan ENN. Kelima model tersebut dibentuk seperti yang telah dituliskan pada bab dasar teori dan metodologi penelitian berdasarkan literatur yang tersedia pada internet. Hasil dari penerapan metode tersebut mampu menyelesaikan permasalahan *imbalanced data*, kecuali ketika menggunakan metode ENN dimana data yang didapatkan masih belum seimbang dan cenderung hanya mengurangi beberapa jumlah data pada kelas *non-defect*.

Tabel 4.4 Perbandingan hasil resampling menggunakan  
ENN, RSMOTE, FCM, SCM dan TCM

Data	Total									
	ENN		RSMOTE		FCM		SCM		TCM	
	D	ND	D	ND	D	ND	D	ND	D	ND
PC1	61	604	698	698	604	604	647	698	698	604
PC4	178	976	1221	1221	976	976	1095	1221	1221	976
PC5	503	16498	16025	16025	16025	16025	16336	16498	16025	16498
AR1	9	112	94	94	94	94	96	112	94	112
AR3	8	55	44	44	44	44	53	55	44	55
AR4	20	87	59	59	59	59	69	87	59	87
KC1	325	1422	1771	1771	1422	1422	1526	1771	1771	1422
KC2	107	305	415	415	305	305	331	415	415	305
KC3	36	114	164	164	114	114	135	164	164	114
MC1	68	9151	9209	9209	9151	9151	9141	9209	9209	9151
MC2	44	39	83	83	44	44	50	83	83	39
CM1	42	217	302	302	217	217	260	302	302	217
JM1	1759	5253	7834	7834	5253	5253	6362	7834	7834	5253
MW1	27	189	237	237	189	189	211	237	237	189
EQ	129	113	195	195	129	129	100	195	195	113
JDT	206	615	791	791	615	615	668	791	791	615
Lucene	64	627	543	543	543	543	583	627	543	627
Mylyn	245	1617	1272	1272	1272	1272	1433	1617	1272	1617
PDE	209	1288	1020	1020	1020	1020	1129	1288	1020	1288

**D: Defect      ND: Non-Defect**

Pada penerapan RSMOTE dan FCM data kedua kelas yang ada memiliki jumlah yang sama disetiap dataset yang digunakan. hal ini dikarenakan pada RSMOTE hanya bertujuan menciptakan data baru pada kelas minoritas hingga jumlahnya sama dengan kelas mayoritas tanpa mengubah data yang ada pada kelas mayoritas, sedangkan metode usulan FCM yang karakteristiknya berfokus membersihkan data terlebih dahulu sebelum melakukan produksi data baru. Sehingga sementara RSMOTE dan FCM mampu memberikan hasil yang lebih baik dibanding dalam menyelesaikan permasalahan *imbalanced data* dimana ratio perbedaan yang didapatkan sama dengan nol.

Pada penerapan SCM jumlah data tetap cenderung lebih banyak pada kelas mayoritas akan tetapi selisih dengan kelas minoritasnya berhasil ditekan seperti pada dataset PC1 dimana selisih yang didapatkan hanya 51 data sehingga kurang lebih ratio perbedaan jumlah data pada kedua kelas sebesar 7%. Tetapi tidak menutup kemungkinan juga terdapat dataset yang masih memiliki ratio perbedaan sangat besar seperti dataset EQ senilai 48%. Hal ini terjadi dikarenakan data originalnya sudah memiliki perbedaan jumlah data antar kelasnya yang tidak terlalu besar dan metode ENN yang ada pada SCM membuat data yang *noisy* yang ada

pada dataset mengalami pembersihan yang awalnya data minoritasnya terdapat 129 menjadi 100.

Pada penerapan metode TCM seperti halnya penerapan metode SCM, data yang didapatkan tidak ada yang seimbang seperti yang dihasilkan metode RSMOTE dan FCM. Ratio yang diberikan metode ini cukup bervariasi, dan terhitung memiliki perbedaan yang cukup besar pada dataset yang digunakan. hal ini dikarenakan hasil penerapan metode ini membuat data baru yang diciptakan pada kelas minoritas lebih banyak dan tidak mengalami penurunan akibat penerapan undersampling pada kelas mayoritas, sedangkan pada kelas mayoritas yang didapatkan lebih bersih dari *noisy*.

Penerapan ENN, cenderung mendapatkan jumlah yang tidak seimbang pada semua dataset yang digunakan, dikarenakan metode ini hanya membersihkan data pada kelas mayoritas. Sehingga ENN tidak terlalu efektif untuk permasalahan data ini dimana data originalnya memiliki ratio perbedaan jumlah data yang sangat besar antara setiap kelas.

#### 4. 2. Implementasi Fitur Seleksi

Pada hasil implementasi fitur seleksi menggunakan RF dapat dilihat pada Tabel 4.5 hingga Tabel 4.23. Pada tabel tersebut ditampilkan fitur fitur yang digunakan saat klasifikasi. Dari skor yang dihasilkan pada setiap fitur, penelitian ini hanya menggunakan fitur yang memberikan nilai diatas 0,03. Hal ini dikarenakan semua dataset ketika diterapkan nilai batas tersebut jumlah fitur yang didapatkan tidak terlalu banyak. Penentuan secara otomatis jumlah fitur optimal menggunakan *KneeLocator* tidak memberikan hasil yang baik karena kembalian yang diberikan menandakan setiap dataset hanya ada satu fitur saja yang dianggap penting, sehingga penelitian ini menentukan nilai minimum pada fitur seleksi.

Tabel 4.5 Hasil Fitur Seleksi Dataset PC1

<b>Nama Fitur</b>	<b>Skor</b>
LOC_COMMENTS	0,095600989
NUMBER_OF_LINES	0,086723031
HALSTEAD_CONTENT	0,073579366
LOC_CODE_AND_COMMENT	0,071130626
LOC_BLANK	0,060082395

PERCENT_COMMENTS	0,059665321
NUM_UNIQUE_OPERANDS	0,058314576
CYCLOMATIC_DENSITY	0,034817364
NORMALIZED_CYLOMATIC_COMPLEXITY	0,031244234
LOC_COMMENTS	0,095600989
NUMBER_OF_LINES	0,086723031
HALSTEAD_CONTENT	0,073579366
LOC_CODE_AND_COMMENT	0,071130626
LOC_BLANK	0,060082395
PERCENT_COMMENTS	0,059665321
NUM_UNIQUE_OPERANDS	0,058314576
CYCLOMATIC_DENSITY	0,034817364
NORMALIZED_CYLOMATIC_COMPLEXITY	0,031244234

Pada Tabel 4.5 merupakan fitur dataset PC1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur, terpilih sembilan fitur dengan nilai tertinggi pada fitur LOC\_COMMENTS senilai 0,095600989 dan NORMALIZED\_CYLOMATIC\_COMPLEXITY sebagai fitur terakhir dengan skor 0,031244234.

Tabel 4.6 Hasil Fitur Seleksi Dataset PC4

Name Attribute	Score
LOC_CODE_AND_COMMENT	0,293846626
LOC_COMMENTS	0,059633525
CYCLOMATIC_DENSITY	0,058589384
LOC_BLANK	0,05101523
PERCENT_COMMENTS	0,050610603
HALSTEAD_CONTENT	0,039369075
NUM_UNIQUE_OPERATORS	0,037398144
HALSTEAD_DIFFICULTY	0,031753158

Pada Tabel 4.6 merupakan fitur dataset PC4 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur, terpilih delapan fitur dengan nilai tertinggi pada fitur LOC\_CODE\_AND\_COMMENT senilai 0,293846626 dan HALSTEAD\_DIFFICULTY sebagai fitur terakhir dengan skor 0,031753158.

Tabel 4.7 Hasil Fitur Seleksi Dataset PC5

Name Attribute	Score
NUM_OPERANDS	0,16088713

DESIGN_COMPLEXITY	0,113990744
HALSTEAD_VOLUME	0,070104535
HALSTEAD_CONTENT	0,054288898
LOC_CODE_AND_COMMENT	0,043093933
PERCENT_COMMENTS	0,040057388
HALSTEAD_DIFFICULTY	0,034109057
LOC_COMMENTS	0,03087203

Pada Table 4.7 merupakan fitur dataset PC5 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur, terpilih delapan fitur dengan nilai tertinggi pada fitur NUM\_OPERANDS senilai 0,16088713 dan LOC\_COMMENTS sebagai fitur terakhir dengan skor 0,03087203.

Tabel 4.8 Hasil Fitur Seleksi Dataset AR1

Name Attribute	Score
unique_operators	0,19073181
decision_density	0,075806602
halstead_effort	0,066867475
halstead_time	0,063718307
design_density	0,059356828
total_loc	0,048065238
total_operators	0,035654842
halstead_difficulty	0,032913996
multiple_condition_count	0,03254887
cyclomatic_complexity	0,031597162
comment_loc	0,031494371
halstead_volume	0,030866188
halstead_length	0,030176682

Pada Tabel 4.8 merupakan fitur dataset AR1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari tiga puluh fitur, terpilih tiga belas fitur dengan nilai tertinggi pada fitur *unique\_operators* senilai 0,19073181 dan *halstead\_length* sebagai fitur terakhir dengan skor 0,030176682.

Tabel 4.9 Hasil Fitur Seleksi Dataset AR3

Name Attribute	Score
halstead_volume	0,105333487
halstead_error	0,10160672
code_and_comment_loc	0,098270221



cyclomatic_complexity	0,063377563
condition_count	0,058466978
comment_loc	0,057524571
total_loc	0,055901856
branch_count	0,042916945
total_operators	0,039020129
cyclomatic_density	0,037767898
call_pairs	0,037482159
normalized_cyclomatic_complexity	0,035848174
unique_operands	0,033287332

Pada Tabel 4.9 merupakan fitur dataset AR3 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari tiga puluh fitur, terpilih tiga belas fitur dengan nilai tertinggi pada fitur *halstead\_volume* senilai 0,105333487 dan *unique\_operands* sebagai fitur terakhir dengan skor 0,033287332.

Tabel 4.10 Hasil Fitur Seleksi Dataset AR4

<b>Name Attribute</b>	<b>Score</b>
unique_operators	0,223288447
comment_loc	0,089415891
halstead_vocabulary	0,08515935
executable_loc	0,065775456
cyclomatic_density	0,05427623
total_loc	0,045143551
total_operands	0,045076672
total_operators	0,041734421
decision_density	0,038628322
unique_operands	0,032585621

Pada Tabel 4.10 merupakan fitur dataset AR4 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari tiga puluh fitur, terpilih sepuluh fitur dengan nilai tertinggi pada fitur *unique\_operators* senilai 0,223288447 dan *unique\_operands* sebagai fitur terakhir dengan skor 0,032585621.

Tabel 4.11 Hasil Fitur Seleksi Dataset KC1

<b>Name Attribute</b>	<b>Score</b>
NUM_OPERANDS	0,099273436
HALSTEAD_CONTENT	0,097196062

HALSTEAD_DIFFICULTY	0,093255141
LOC_TOTAL	0,086819832
LOC_EXECUTABLE	0,079805518
HALSTEAD_VOLUME	0,066932636
NUM_OPERATORS	0,056413658
NUM_UNIQUE_OPERATORS	0,047592439
LOC_BLANK	0,044748128
HALSTEAD_PROG_TIME	0,043122135
HALSTEAD_EFFORT	0,040120211
NUM_UNIQUE_OPERANDS	0,040090362
HALSTEAD_LENGTH	0,038624714
LOC_COMMENTS	0,037272825

Pada Tabel 4.11 merupakan fitur dataset KC1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari sembilan belas fitur, terpilih empat belas fitur dengan nilai tertinggi pada fitur NUM\_OPERANDS senilai 0,099273436 dan LOC\_COMMENTS sebagai fitur terakhir dengan skor 0,037272825.

Tabel 4.12 Hasil Fitur Seleksi Dataset KC2

Name Attribute	Score
total_Opnd	0,190383533
IOBlank	0,09934934
uniq_Opnd	0,08977634
evg	0,070659598
i	0,07050626
loc	0,06755436
total_Op	0,048572801
IOCode	0,042958253
uniq_Op	0,039962173
IOComment	0,037447815
d	0,035344211
v	0,030272951

Pada Table 4.12 merupakan fitur dataset KC2 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari dua puluh fitur, terpilih dua belas fitur dengan nilai tertinggi pada fitur *total\_Opnd* senilai 0,190383533 dan *v* sebagai fitur terakhir dengan skor 0,030272951.

Tabel 4.13 Hasil Fitur Seleksi Dataset KC3

Name Attribute	Score
LOC_CODE_AND_COMMENT	0,12851138
PERCENT_COMMENTS	0,074288286
MAINTENANCE_SEVERITY	0,0574465
HALSTEAD_CONTENT	0,042919502
NUM_UNIQUE_OPERANDS	0,042479483
LOC_COMMENTS	0,04226785
BRANCH_COUNT	0,04150566
CYCLOMATIC_DENSITY	0,040318035
HALSTEAD_DIFFICULTY	0,03594958
NUM_UNIQUE_OPERATORS	0,035079278

Pada Table 4.13 merupakan fitur dataset KC3 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih sepuluh fitur dengan nilai tertinggi pada fitur *LOC\_CODE\_AND\_COMMENT* senilai 0,12851138 dan *NUM\_UNIQUE\_OPERATORS* sebagai fitur terakhir dengan skor 0,035079278.

Tabel 4.14 Hasil Fitur Seleksi Dataset MC1

Name Attribute	Score
NUMBER_OF_LINES	0,075766543
ESSENTIAL_DENSITY	0,061142486
CALL_PAIRS	0,051635146
PERCENT_COMMENTS	0,049568348
HALSTEAD_CONTENT	0,04946776
LOC_CODE_AND_COMMENT	0,046148435
CYCLOMATIC_DENSITY	0,040811694
HALSTEAD_VOLUME	0,040450105
NUM_OPERATORS	0,039995756
HALSTEAD_DIFFICULTY	0,039387499
EDGE_COUNT	0,035821041

Pada Table 4.14 merupakan fitur dataset MC1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari tiga puluh sembilan fitur terpilih, sebelas fitur dengan nilai tertinggi pada fitur *NUMBER\_OF\_LINES* senilai 0,075766543 dan *EDGE\_COUNT* sebagai fitur terakhir dengan skor 0,035821041.

Tabel 4.15 Hasil Fitur Seleksi Dataset MC2

Name Attribute	Score
HALSTEAD_DIFFICULTY	0,08775117
PARAMETER_COUNT	0,053640861
NORMALIZED_CYLOMATIC_COMPLEXITY	0,050163739
LOC_COMMENTS	0,048058442
NODE_COUNT	0,046989526
EDGE_COUNT	0,045985603
HALSTEAD_PROG_TIME	0,044069366
PERCENT_COMMENTS	0,042388584
MAINTENANCE_SEVERITY	0,039201127
DESIGN_DENSITY	0,038675626
ESSENTIAL_DENSITY	0,033775267
GLOBAL_DATA_DENSITY	0,033627278
CALL_PAIRS	0,031743058
CYCLOMATIC_DENSITY	0,031425768
HALSTEAD_EFFORT	0,030661802

Pada Tabel 4.15 merupakan fitur dataset MC2 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih delapan fitur dengan nilai tertinggi pada fitur *HALSTEAD\_DIFFICULTY* senilai 0,08775117 dan *HALSTEAD\_EFFORT* sebagai fitur terakhir dengan skor 0,030661802.

Tabel 4.16 Hasil Fitur Seleksi Dataset CM1

Name Attribute	Score
LOC_COMMENTS	0,143900259
DESIGN_DENSITY	0,084214151
NUM_UNIQUE_OPERATORS	0,06509654
PARAMETER_COUNT	0,05931422
PERCENT_COMMENTS	0,050557731
LOC_CODE_AND_COMMENT	0,047971551
LOC_BLANK	0,044655899
CALL_PAIRS	0,042433816
HALSTEAD_CONTENT	0,040690302
NORMALIZED_CYLOMATIC_COMPLEXITY	0,03941942
NODE_COUNT	0,032282481
HALSTEAD_DIFFICULTY	0,031154562

Pada Tabel 4.16 merupakan fitur dataset CM1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari tiga puluh sembilan

fitur, terpilih dua belas fitur dengan nilai tertinggi pada fitur *LOC\_COMMENTS* senilai 0,143900259 dan *HALSTEAD\_DIFFICULTY* sebagai fitur terakhir dengan skor 0,031154562.

Tabel 4.17 Hasil Fitur Seleksi Dataset JM1

Name Attribute	Score
LOC_TOTAL	0,134292099
HALSTEAD_CONTENT	0,088900483
LOC_EXECUTABLE	0,067920699
HALSTEAD_DIFFICULTY	0,065892986
LOC_BLANK	0,06133813
NUM_UNIQUE_OPERANDS	0,056763759
NUM_OPERATORS	0,054977547
NUM_OPERANDS	0,05219905
NUM_UNIQUE_OPERATORS	0,051583135
HALSTEAD_VOLUME	0,050777657
LOC_COMMENTS	0,039622191
HALSTEAD_EFFORT	0,037130133
HALSTEAD_PROG_TIME	0,037047687
DESIGN_COMPLEXITY	0,036067603
HALSTEAD_LENGTH	0,035437309

Pada Tabel 4.17 merupakan fitur dataset JM1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari dua puluh satu fitur, terpilih lima belas fitur dengan nilai tertinggi pada fitur *LOC\_TOTAL* senilai 0,134292099 dan *HALSTEAD\_LENGTH* sebagai fitur terakhir dengan skor 0,035437309.

Tabel 4.18 Hasil Fitur Seleksi Dataset MW1

Name Attribute	Score
NODE_COUNT	0,152464588
HALSTEAD_CONTENT	0,106445796
CALL_PAIRS	0,087049273
LOC_COMMENTS	0,057756051
NUM_UNIQUE_OPERANDS	0,040245037
NORMALIZED_CYLOMATIC_COMPLEXITY	0,037686425
NUMBER_OF_LINES	0,035618633
NUM_UNIQUE_OPERATORS	0,035387657
LOC_BLANK	0,034500597
EDGE_COUNT	0,034224301

PERCENT_COMMENTS	0,034117844
CYCLOMATIC_DENSITY	0,032376839

Pada Tabel 4.18 merupakan fitur dataset MW1 yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih dua belas fitur dengan nilai tertinggi pada fitur NODE\_COUNT senilai 0,152464588 dan CYCLOMATIC\_DENSITY sebagai fitur terakhir dengan skor 0,032376839.

Tabel 4.19 Hasil Fitur Seleksi Dataset EQ

Name Attribute	Score
CvsExpEntropy	0,210302346
CvsLogEntropy	0,09532467
CvsWEntropy	0,061976902
CvsEntropy	0,052335355
CvsLinEntropy	0,038474631
ck_oo_wmc	0,034488556
ck_oo_numberOfMethodsInherited	0,032267444

Pada Tabel 4.19 merupakan fitur dataset EQ yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih tujuh fitur dengan nilai tertinggi pada fitur *CvsExpEntropy* senilai 0,210302346 dan *ck\_oo\_numberOfMethodsInherited* sebagai fitur terakhir dengan skor 0,032267444.

Tabel 4.20 Hasil Fitur Seleksi Dataset JDT

Name Attribute	Score
CvsEntropy	0,153294384
WCHU_numberOfLinesOfCode	0,089044159
LDHH_rfc	0,068888826
WCHU_wmc	0,062013295
ck_oo_rfc	0,033627245
CvsWEntropy	0,031834971

Pada Tabel 4.20 merupakan fitur dataset JDT yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih enam fitur dengan nilai tertinggi pada fitur *CvsEntropy* senilai 0,16088713 dan *CvsWEntropy* sebagai fitur terakhir dengan skor 0,03087203.

Tabel 4.21 Hasil Fitur Seleksi Dataset Lucene

Name Attribute	Score
CvsLinEntropy	0,139965391
CvsEntropy	0,05970078
CvsExpEntropy	0,045692054
WCHU_numberOfPrivateAttributes	0,04023498
LDHH_wmc	0,038975384
CvsWEntropy	0,035416537

Pada Tabel 4.21 merupakan fitur dataset Lucene yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih enam fitur dengan nilai tertinggi pada fitur *CvsLinEntropy* senilai 0,16088713 dan *CvsWEntropy* sebagai fitur terakhir dengan skor 0,035416537.

Tabel 4.22 Hasil Fitur Seleksi Dataset Mylyn

Name Attribute	Score
ck_oo_fanOut	0,06219171
ck_oo_rfc	0,054166625
ck_oo_numberOfPrivateMethods	0,04808366
LDHH_rfc	0,043541727
LDHH_numberOfLinesOfCode	0,042454974
ck_oo_numberOfLinesOfCode	0,042255014
ck_oo_numberOfAttributes	0,035334977
ck_oo_cbo	0,032854503
ck_oo_wmc	0,031235267

Pada Tabel 4.22 merupakan fitur dataset Mylyn yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur, terpilih sembilan fitur dengan nilai tertinggi pada fitur *ck\_oo\_fanOut* senilai 0,06219171 dan *ck\_oo\_wmc* sebagai fitur terakhir dengan skor 0,031235267.

Tabel 4.23 Hasil Fitur Seleksi Dataset PDE

Name Attribute	Score
CvsWEntropy	0,076301356
numberOfNonTrivialBugsFoundUntil:	0,048891271
CvsExpEntropy	0,04676724
LDHH_rfc	0,044079222
CvsLinEntropy	0,039241012
LDHH_numberOfLinesOfCode	0,036565857
ck_oo_numberOfLinesOfCode	0,036434377

CvsLogEntropy	0,031908418
ck_oo_rfc	0,030991483

Pada Tabel 4.23 merupakan fitur dataset PDE yang terpilih dan yang akan digunakan untuk melakukan prediksi atau klasifikasi. Dari empat puluh fitur terpilih sembilan fitur dengan nilai tertinggi pada fitur *CvsWEntropy* senilai 0,076301356 dan *ck\_oo\_rfc* sebagai fitur terakhir dengan skor 0,030991483. Dari semua hasil seleksi fitur menggunakan metode RF akan digunakan untuk melakukan prediksi ditahap selanjutnya menggunakan ANN, CNN dan RNN. Dengan terpilihnya fitur-fitur tersebut dapat membantu prediksi dalam mengurangi beban kerja sehingga memberikan hasil yang lebih akurat dan

#### 4. 3. Implementasi Model Prediksi ANN

Pada bagian ini, hasil dari oversampling dan undersampling serta fitur seleksi digunakan untuk melakukan prediksi menggunakan metode deep learning ANN. Pada Tabel 4.23 merupakan hasil prediksi menggunakan data hasil oversampling dan undersampling model FCM. Dilihat dari tabel tersebut MC1 memberikan hasil akurasi tertinggi diantara dataset yang lain dan AR4 memberikan hasil akurasi terendah

Tabel 4.24 Hasil evaluasi prediksi ANN menggunakan data FCM

<b>Dataset</b>	<b>F-Measure</b>	<b>Precision</b>	<b>Recall</b>	<b>ROC/AUC</b>	<b>Accuracy</b>
PC1	0,93023	0,89552	0,96774	0,92455	0,92562
PC4	0,95506	0,93407	0,97701	0,96098	0,959184
PC5	0,66486	0,49797	1,00000	0,50000	0,497972
AR1	0,85714	0,78947	0,93750	0,87784	0,868421
AR3	0,75000	0,66667	0,85714	0,79221	0,777778
AR4	0,84211	0,88889	0,80000	0,86429	0,875
KC1	0,81301	0,87719	0,75758	0,83304	0,838596
KC2	0,92308	0,92958	0,91667	0,90833	0,909836
KC3	0,78261	0,81818	0,75000	0,78409	0,782609
MC1	0,98652	0,99891	0,97444	0,98666	0,986346
MC2	0,90909	0,83333	1,00000	0,87500	0,888889
CM1	0,87500	0,91304	0,84000	0,86737	0,863636
JM1	0,77885	0,81818	0,74312	0,78263	0,781161
MW1	0,87180	0,85000	0,89474	0,86842	0,868421
EQ	0,89362	0,87500	0,91304	0,90480	0,90385
JDT	0,81057	0,82883	0,79310	0,82347	0,82520



Lucene	0,86239	0,90385	0,82456	0,86420	0,86239
Mylyn	0,85429	0,88066	0,82946	0,85696	0,85658
PDE	0,85405	0,87778	0,83158	0,86533	0,86765

Pada Tabel 4.24 merupakan hasil prediksi menggunakan data kombinasi oversampling dan undersampling model SCM. Dapat dilihat bahwa prediksi pada AR3 dan MW1 memberikan nilai akurasi dan evaluasi yang sempurna ketika diterapkan pada model klasifikasi ANN.

Tabel 4.25 Hasil evaluasi prediksi ANN menggunakan data SCM

Dataset	F-Measure	Precision	Recall	ROC/AUC	Accuracy
PC1	0,88066	0,91453	0,84921	0,88939	0,89179
PC4	0,90069	0,92857	0,87444	0,90610	0,90733
PC5	0,45153	0,85920	0,30623	0,62763	0,62513
AR1	0,97872	1,00000	0,95833	0,97917	0,97619
AR3	0,94118	0,88889	1,00000	0,96429	0,95455
AR4	0,83333	0,71429	1,00000	0,90476	0,87097
KC1	0,88040	0,94643	0,82298	0,88943	0,89124
KC2	0,35714	0,75000	0,23438	0,58778	0,63758
KC3	0,81633	0,86957	0,76923	0,83916	0,84746
MC1	0,98851	0,98959	0,98742	0,98855	0,98855
MC2	0,66667	0,75000	0,60000	0,74118	0,77778
CM1	0,91057	0,91803	0,90323	0,90354	0,90351
JM1	0,84114	0,90215	0,78786	0,85613	0,86031
MW1	0,91954	0,85106	1,00000	0,93000	0,92222
EQ	0,65000	0,86667	0,52000	0,73059	0,76271
JDT	0,88406	0,86525	0,90370	0,89095	0,89003
Lucene	0,87660	0,84426	0,91150	0,88211	0,88017
Mylyn	0,84848	0,89811	0,80405	0,85834	0,85950
PDE	0,87585	0,91943	0,83621	0,88477	0,88706

Pada Tabel 4.25 merupakan hasil prediksi menggunakan data kombinasi oversampling dan undersampling model TCM. Dari tabel tersebut menunjukkan prediksi pada AR3 memberikan hasil yang sempurna dan pada dataset MC2 memberikan nilai evaluasi terendah dibandingkan dataset lainnya.

Tabel 4.26 Hasil evaluasi prediksi ANN menggunakan data TCM

Dataset	F-Measure	Precision	Recall	ROC/AUC	Accuracy
PC1	0,89394	0,90076	0,88722	0,89283	0,89272
PC4	0,92057	0,90763	0,93388	0,90886	0,91136
PC5	0,83985	0,73585	0,97808	0,81006	0,81168

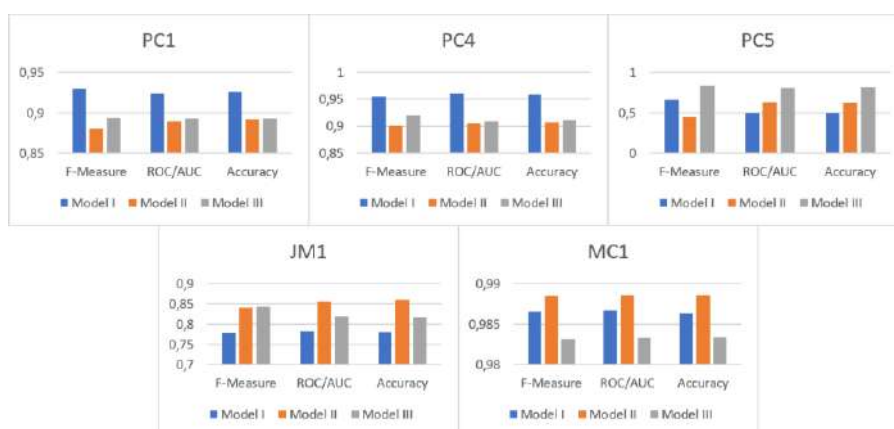
AR1	0,93878	1,00000	0,88462	0,94231	0,92857
AR3	0,54545	0,37500	1,00000	0,64286	0,50000
AR4	0,80000	0,69565	0,94118	0,70136	0,73333
KC1	0,88543	0,89429	0,87675	0,87277	0,87324
KC2	0,90395	0,91954	0,88889	0,87963	0,88194
KC3	0,87671	0,82051	0,94118	0,81150	0,83929
MC1	0,98316	0,98835	0,97803	0,98334	0,98339
MC2	0,72727	0,60000	0,92308	0,62821	0,64000
CM1	0,90323	0,93333	0,87500	0,88750	0,88462
JM1	0,84297	0,87697	0,81151	0,81897	0,81742
MW1	0,96154	0,98039	0,94340	0,95655	0,95349
EQ	0,90141	0,94118	0,86486	0,89243	0,88710
JDT	0,88535	0,91447	0,85802	0,87485	0,87234
Lucene	0,90000	0,87313	0,92857	0,88558	0,88889
Mylyn	0,86371	0,91958	0,81424	0,86202	0,85640
PDE	0,85652	0,91628	0,80408	0,86057	0,85714

Jika dibandingkan tiap modelnya bisa diketahui bahwa penerapan yang memberikan nilai FMeasure terendah ketika menggunakan oversampling dan undersampling model SCM pada dataset MC2. Sedangkan oversampling dan undersampling model SCM dan ketiga mampu memberikan hasil terbaik pada dataset AR3. Secara keseluruhan model TCM mampu memberikan hasil F-Measure lebih baik dibandingkan model lainnya dengan nilai paling kecil 0,8 pada dataset MC2. Perbandingan tersebut ditunjukkan pada Gambar 4.17

Ketika dibandingkan nilai evaluasi ROC/AUC dapat dilihat model SCM mampu memberikan nilai terbaik ditunjukkan ketika menggunakan dataset AR3 dan MW1 model prediksi mampu memberikan nilai evaluasi yang sempurna. Jika dilihat pada model kombinasi oversampling dan undersampling pertama, model tersebut kurang mampu memberikan nilai evaluasi yang maksimal, ketika menggunakan dataset AR4, KC2, dan KC3, model tersebut hanya mampu memberikan nilai evaluasi dibawah 0,75. Diamati secara keseluruhan, model kombinasi oversampling dan undersampling ketiga ketika dilakukan prediksi dibandingkan dengan kedua model yang lain, model ini mampu konsisten memberikan hasil terbaik, bisa dilihat evaluasi terkecil ditunjukkan pada MC2 dengan nilai 0,78571.

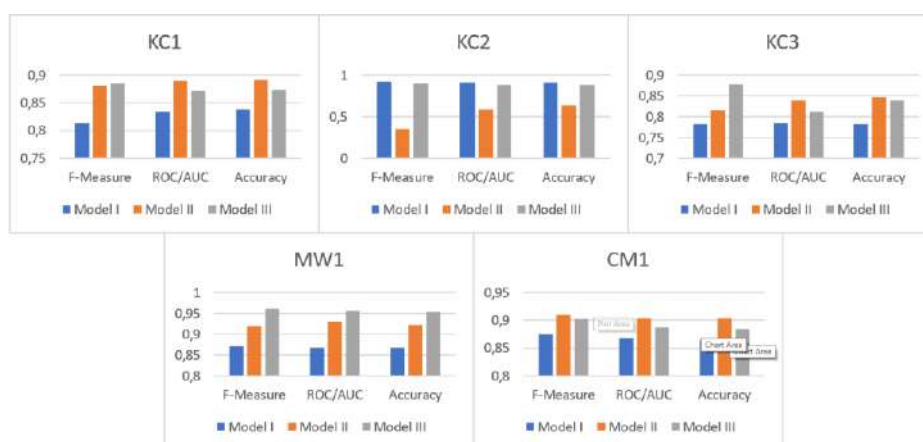
Dan jika dibandingkan pada nilai akurasi, semua model mampu memberikan nilai yang baik, Adapun yang tidak mampu memberikan nilai evaluasi yang maksimal ketika prediksi dilakukan menggunakan data hasil oversampling dan undersampling model FCM pada dataset PC5. Data pada model tersebut hanya mampu memberikan nilai akurasi sebesar 0,497972. Dari semua hasil evaluasi yang ditunjukkan pada hasil prediksi menggunakan data kombinasi oversampling dan undersampling ketiga model yang diusulkan, dapat disimpulkan bahwa model TCM mampu memberikan hasil yang maksimal. Dari semua cara pengukuran yang digunakan (fmeasure, roc/auc dan akurasi) model tersebut ketika digunakan pada model prediksi ANN hanya dataset JM1 dan MC1 yang tidak dapat memberikan hasil diatas 0,90.

Evaluasi yang dihasilkan oversampling dan undersampling pada dataset PC1 memiliki nilai tertinggi ketika menggunakan model FCM, dimana nilai F-Measure dihasilkan 0,93023 dan akurasi yang dihasilkan 0,92562. Hasil evaluasi PC4 sama seperti PC1 dimana hasil evaluasi terbaik diberikan oleh data kombinasi oversampling dan undersampling model FCM dengan nilai F-Measure sebesar 0,95506 dan akurasi sebesar 0,959184. Untuk dataset JM1 dan MC1 memberikan hasil terbaik ketika menggunakan data kombinasi oversampling dan undersampling model SCM. F-Measure yang dihasilkan JM1 dan MC1 sebesar 0,84114 dan 0,98851. Untuk PC5, evaluasi terbaik diberikan ketika menggunakan data kombinasi oversampling dan undersampling model TCM dengan nilai F-Measure sebesar 0,83985 dan akurasi 0,81168.



Gambar 4.5 Perbandingan hasil evaluasi tiap model pada prediksi ANN menggunakan dataset (PC1, PC4, PC5, JM1 dan MC1)

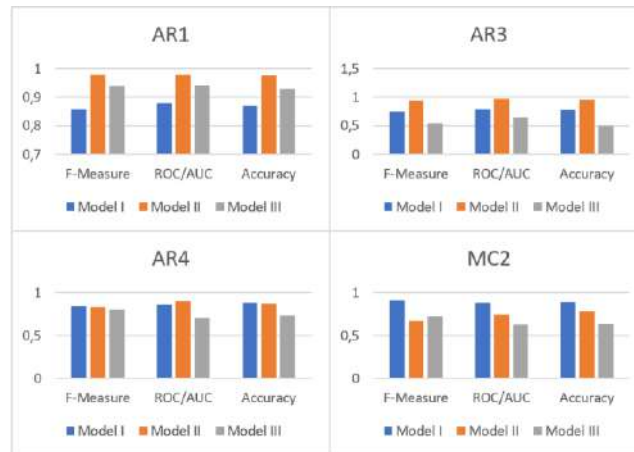
Pada Gambar 4.5 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset KC1, KC2, KC3, MW1 dan CM1. Evaluasi yang dihasilkan oversampling dan undersampling pada dataset KC1 dan CM1 memiliki nilai tertinggi ketika menggunakan model SCM, dimana nilai F-Measure dihasilkan 0,88040 dan 0,91057. Nilai Akurasi yang diberikan kedua dataset tersebut sebesar 0,89124 dan 0,90351. Untuk dataset KC2 ketika dilakukan prediksi saat dilakukan evaluasi memberikan nilai terbaik ketika menggunakan data kombinasi oversampling dan undersampling model FCM, dimana nilai F-Measure yang dihasilkan 0,92308 dan nilai akurasi 0,909836. Pada dataset MW1 dataset model TCM memberikan evaluasi tertinggi dimana nilai F-Measure dan akurasinya sebesar 0,96154 dan 0,95349. Untuk KC3 memiliki keunikan tersendiri dimana nilai F-Measure terbaik diberikan ketika dataset yang digunakan dari model kombinasi oversampling dan undersampling model TCM dan untuk ROC/AUC dan akurasi nilai terbaik diberikan ketika menggunakan data kombinasi oversampling dan undersampling model SCM.



Gambar 4.6 Perbandingan hasil evaluasi tiap model pada prediksi ANN menggunakan dataset (KC1, KC2, KC3, MW1 dan CM1)

Pada Gambar 4.6 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset AR1, AR3, AR4 dan MC2. Evaluasi yang dihasilkan oversampling dan undersampling pada dataset AR1, AR3 dan AR4 memiliki nilai tertinggi ketika menggunakan model SCM. Nilai F-Measure yang dihasilkan AR1 sebesar 0,97872 dan AR3 sebesar 0,94118 dan AR3 senilai 0,83333. Dan untuk dataset MC2 ketika hasil evaluasi prediksi menggunakan ANN memberikan nilai

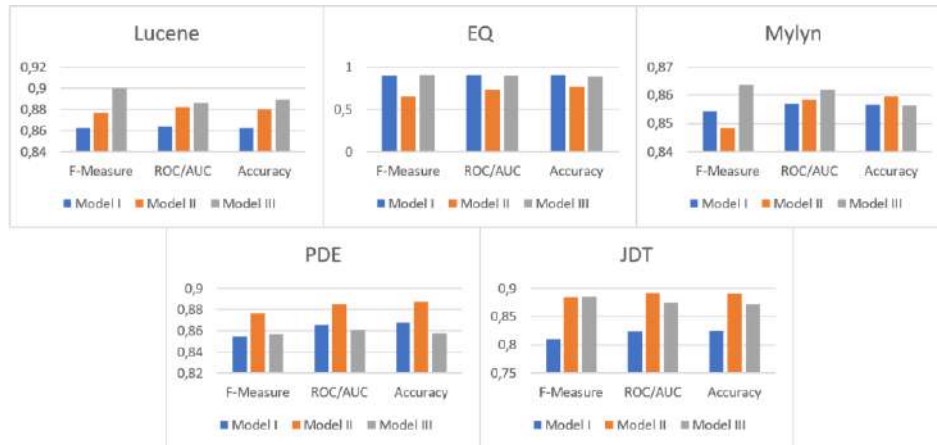
terbaik ketika menggunakan data kombinasi oversampling dan undersampling model FCM. Untuk F-Measure senilai 0,90909 dan evaluasi ROC/AUC senilai 0,8750 dan akurasi sebesar 0,88889.



Gambar 4.7 Perbandingan hasil evaluasi tiap model pada prediksi ANN menggunakan dataset (AR1, AR2, AR3 dan MC2)

Pada Gambar 4.7 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset Lucene, EQ, Mylyn, PDE dan JDT. Evaluasi yang dihasilkan oversampling dan undersampling pada dataset Lucene memiliki nilai tertinggi ketika menggunakan model TCM. Dimana hasil F-Measure yang dihasilkan sebesar 0,9 dan ROC/AUC senilai 0,89243 dan akurasi senilai 0,88710. Untuk dataset EQ memiliki nilai f-measure tertinggi ketika menggunakan data kombinasi oversampling dan undersampling model TCM. Untuk evaluasi ROC/AUC dan Akurasi diberikan tertinggi ketika menggunakan model FCM. Evaluasi F-Measure pada dataset EQ sebesar 0,90141 dan evaluasi ROC/AUC memberikan nilai sebesar 0,90480 dan akurasi sebesar 0,90385. Pada dataset PDE nilai semua evaluasi terbaik dihasilkan ketika menggunakan dataset oversampling dan undersampling model SCM dengan nilai f-measure, roc/auc dan akurasi sebesar 0,8785 dan 0,88477 dan 0,88706. Pada dataset Mylyn perbedaan data kombinasi oversampling dan undersampling yang digunakan ketika melakukan pengukuran akurasi dimana F-measure dan ROC/AUC memberikan hasil terbaik ketika menggunakan model kombinasi oversampling dan undersampling ketiga dan akurasi terbaik ketika dataset yang digunakan berasal dari data kombinasi oversampling dan undersampling model SCM. Untuk nilai F-Measure dan ROC/AUC yang diberikan

ketika menggunakan data model TCM sebesar 0,86371 dan 0,86202. Untuk Akurasi memberikan nilai sebesar 0,85950 selisih 0,00310 dengan akurasi yang dihasilkan ketika melakukan prediksi menggunakan dataset oversampling dan undersampling model TCM.



Gambar 4.8 Perbandingan hasil evaluasi tiap model pada prediksi ANN menggunakan dataset (Lucene, EQ, Mylyn, PDE dan JDT)

Dari semua perbandingan yang telah ditunjukkan pada Gambar 4.1 hingga 4.8. menunjukkan bahwa setiap model tidak ada yang mampu mendominasi ketika dilakukan prediksi menggunakan ANN. Terhitung sebelas dataset oversampling dan undersampling model SCM memberikan nilai F-Measure, ROC/AUC dan Akurasi tertinggi.

#### 4. 4. Implementasi Model Prediksi CNN

Pada bagian ini, hasil dari oversampling dan undersampling serta fitur seleksi digunakan untuk melakukan prediksi menggunakan metode deep learning CNN. Pada Tabel 4.26 merupakan hasil prediksi menggunakan data hasil oversampling dan undersampling model FCM. Dari hasil tersebut dapat diketahui ketika prediksi dilakukan pada dataset AR3, memberikan hasil evaluasi paling kecil dengan nilai F-Measure senilai 0,67. Dan Nilai F-Measure tertinggi didapatkan ketika PC5 digunakan untuk melakukan prediksi dengan nilai sebesar 0,98489.

Tabel 4.27 Hasil evaluasi prediksi CNN menggunakan data FCM

Dataset	F-Measure	Precision	Recall	ROC/AUC	Accuracy
PC1	0,93130	0,88406	0,98387	0,92414	0,92562
PC4	0,92021	0,87817	0,96648	0,92664	0,92327

PC5	0,98489	0,98987	0,97995	0,98500	0,98502
AR1	0,89655	1,00000	0,81250	0,90625	0,92105
AR3	0,66667	1,00000	0,50000	0,75000	0,77778
AR4	0,76191	0,72727	0,80000	0,79286	0,79167
KC1	0,75720	0,82883	0,69697	0,78639	0,79298
KC2	0,90141	0,86486	0,94118	0,87800	0,88525
KC3	0,71429	0,83333	0,62500	0,74432	0,73913
MC1	0,98355	0,97749	0,98969	0,98329	0,98334
MC2	0,90909	0,83333	1,00000	0,87500	0,88889
CM1	0,85714	0,86667	0,84783	0,85074	0,85058
JM1	0,76625	0,82040	0,71881	0,77721	0,77593
MW1	0,94118	0,93023	0,95238	0,93207	0,93421
EQ	0,84615	0,75862	0,95652	0,85757	0,84615
JDT	0,81614	0,85047	0,78448	0,83070	0,83333
Lucene	0,86996	0,88991	0,85088	0,86775	0,86697
Mylyn	0,86747	0,90000	0,83721	0,87080	0,87033
PDE	0,83853	0,90798	0,77895	0,85507	0,86029

Pada penggunaan data kombinasi oversampling dan undersampling model SCM untuk melakukan prediksi, hasil evaluasi ditunjukkan pada Tabel 4.27. Dari tabel tersebut ditunjukkan bahwa dataset EQ ketika dilakukan prediksi, hasil evaluasi F-Measure hanya memberikan nilai sebesar 0,65 sama seperti MC2 memberikan nilai F-Measure terkecil senilai 0,67. Dimana semua dataset lainnya mampu memberikan nilai evaluasi hasil prediksi diatas 0,80. Pada tabel tersebut juga ditunjukkan ketika MC1 digunakan dalam melakukan prediksi, nilai evaluasi F-Measure yang dihasilkan menjadi nilai evaluasi tertinggi dimana nilai yang dihasilkan sebesar 0,98020.

Tabel 4.28 Hasil evaluasi prediksi CNN menggunakan data kombinasi oversampling dan undersampling model SCM

<b>Dataset</b>	<b>F-Measure</b>	<b>Precision</b>	<b>Recall</b>	<b>ROC/AUC</b>	<b>Accuracy</b>
PC1	0,86747	0,87805	0,85714	0,87575	0,87687
PC4	0,91648	0,92273	0,91031	0,91989	0,92026
PC5	0,84041	0,73173	0,98700	0,80974	0,81112
AR1	0,97872	1,00000	0,95833	0,97917	0,97619
AR3	0,61538	0,80000	0,50000	0,71429	0,77273
AR4	0,86957	0,76923	1,00000	0,92857	0,90323
KC1	0,89226	0,87240	0,91304	0,89329	0,89275
KC2	0,80303	0,77941	0,82813	0,82583	0,82550

KC3	0,86957	1,00000	0,76923	0,88462	0,89831
MC1	0,98020	0,97255	0,98797	0,98013	0,98010
MC2	0,66667	0,75000	0,60000	0,74118	0,77778
CM1	0,92800	0,92063	0,93548	0,91967	0,92105
JM1	0,84603	0,87884	0,81559	0,85806	0,86066
MW1	0,93023	0,86957	1,00000	0,94000	0,93333
EQ	0,65000	0,86667	0,52000	0,73059	0,76271
JDT	0,87085	0,86765	0,87407	0,87934	0,87973
Lucene	0,87225	0,86842	0,87611	0,87991	0,88017
Mylyn	0,86598	0,88112	0,85135	0,87066	0,87107
PDE	0,86710	0,87665	0,85776	0,87398	0,87474

Pada penggunaan data kombinasi oversampling dan undersampling model TCM untuk melakukan prediksi, hasil evaluasi ditunjukkan pada Tabel 4.28. Dari tabel tersebut ditunjukkan bahwa dataset AR3 ketika dilakukan prediksi, hasil evaluasi F-Measure hanya memberikan nilai hanya sebesar 0,52174. Hal ini merupakan hasil F-Measure terkecil, dimana hanya dataset AR3 saja yang memberikan hasil evaluasi terkecil. Pada tabel tersebut juga ditunjukkan ketika MC1 digunakan dalam melakukan prediksi, nilai evaluasi F-Measure yang dihasilkan menjadi nilai evaluasi tertinggi dimana nilai yang dihasilkan sebesar 0,96857.

Tabel 4.29 Hasil evaluasi prediksi CNN menggunakan data kombinasi oversampling dan undersampling model TCM

<b>Dataset</b>	<b>F-Measure</b>	<b>Precision</b>	<b>Recall</b>	<b>ROC/AUC</b>	<b>Accuracy</b>
PC1	0,88800	0,94872	0,83459	0,89386	0,89272
PC4	0,93920	0,95319	0,92562	0,93503	0,93409
PC5	0,83926	0,73738	0,97381	0,81010	0,81168
AR1	0,92000	0,95833	0,88462	0,91106	0,90476
AR3	0,52174	0,35294	1,00000	0,60714	0,45000
AR4	0,84211	0,76190	0,94118	0,77828	0,80000
KC1	0,87111	0,92453	0,82353	0,86921	0,86385
KC2	0,91111	0,91111	0,91111	0,88148	0,88889
KC3	0,82857	0,80556	0,85294	0,76738	0,78571
MC1	0,96857	0,96409	0,97309	0,96872	0,96868
MC2	0,68421	0,52000	1,00000	0,50000	0,52000
CM1	0,89256	0,94737	0,84375	0,88438	0,87500
JM1	0,84948	0,88000	0,82100	0,82516	0,82429
MW1	0,93204	0,96000	0,90566	0,92253	0,91860



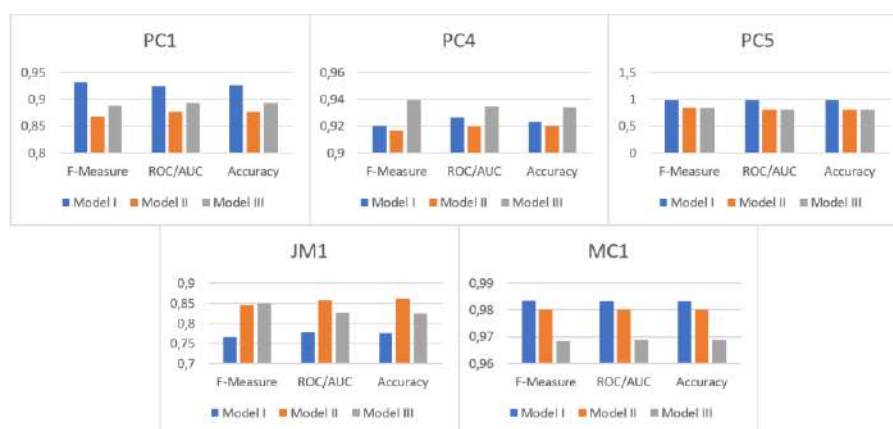
EQ	0,91429	0,96970	0,86486	0,91243	0,90323
JDT	0,88535	0,91447	0,85802	0,87485	0,87234
Lucene	0,90769	0,88060	0,93651	0,89418	0,89744
Mylyn	0,88050	0,89457	0,86687	0,86873	0,86851
PDE	0,87049	0,90708	0,83673	0,86998	0,86797

Hasil evaluasi pada Tabel 4.26 hingga 2.28 kemudian dibandingkan seperti yang ditunjukkan pada Gambar 4.24 hingga 4.26. Ketika dibandingkan semua dataset pada semua model kombinasi oversampling dan undersampling, MC1 pada semua model mampu memberikan nilai F-Measure stabil meski tidak paling tertinggi dimana nilainya berkisar 0,96 hingga 0,98. Pada dataset AR3, ketika pengukuran evaluasi prediksi menggunakan F-Measure dibandingkan, semua dataset oversampling dan undersampling baik model FCM, kedua dan ketiga nilai yang diberikan menjadi nilai terendah dimana nilai tertinggi yang diberikan AR3 ketika menggunakan data kombinasi oversampling dan undersampling model FCM senilai 0,67 dan terkecil diberikan ketika menggunakan model TCM senilai 0,52174.

Nilai yang dibandingkan pada gambar tersebut adalah nilai ROC/AUC. Dataset MC1 pada semua model mampu memberikan nilai F-Measure stabil meski tidak paling tertinggi dimana nilainya berkisar 0,96 hingga 0,98. Pada pengukuran evaluasi ROC/AUC tidak ada dataset yang memberikan nilai evaluasi buruk pada semua model kombinasi oversampling dan undersampling, tetapi pada MC2 ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model TCM, hasil evaluasinya hanya memberikan nilai 0,5 dimana nilai tersebut merupakan nilai terkecil dibandingkan nilai dari semua dataset.

Nilai yang dibandingkan pada gambar tersebut adalah nilai ROC/AUC. Dataset MC1 pada semua model mampu memberikan nilai F-Measure stabil meski tidak paling tertinggi dimana nilainya berkisar 0,96 hingga 0,98. Pada pengukuran evaluasi ROC/AUC tidak ada dataset yang memberikan nilai evaluasi buruk pada semua model kombinasi oversampling dan undersampling, tetapi pada AR3 ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model TCM, hasil evaluasinya hanya memberikan nilai 0,45 dimana nilai tersebut merupakan nilai terkecil dibandingkan nilai dari semua dataset.

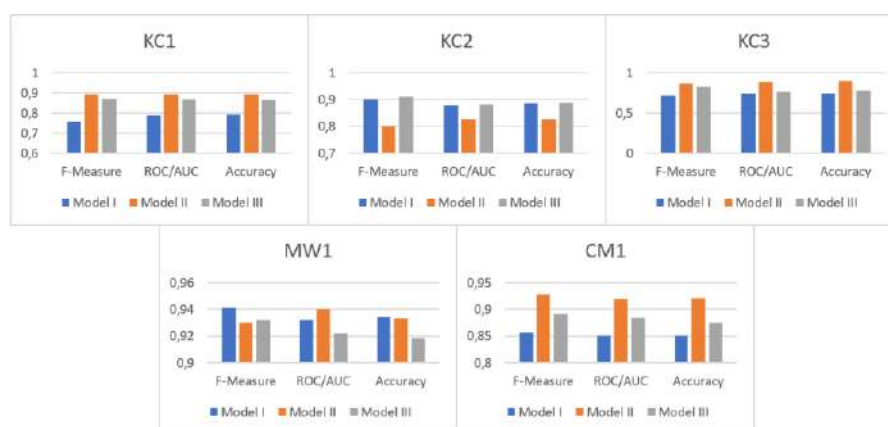
Evaluasi yang dihasilkan oversampling dan undersampling pada dataset PC1, PC5 dan MC1 memberikan nilai tertinggi ketika menggunakan model FCM. Untuk PC1 nilai F-Measure dihasilkan 0,93129. Untuk PC5 nilai F-Measure dihasilkan 0,98488. Untuk MC1 nilai F-Measure dihasilkan 0,983554. Pengukuran akurasi yang dihasilkan pada PC1 senilai 0,92562. Pada PC5, nilai akurasi yang dihasilkan sebesar 0,985023. Dan pada MC1, nilai akurasinya senilai 0,983338. Ketika prediksi dilakukan menggunakan dataset JM1, hasil evaluasinya ketika menggunakan data kombinasi oversampling dan undersampling model SCM mampu menghasilkan yang terbaik kecuali untuk evaluasi F-Measure nilai tertinggi dihasilkan ketika menggunakan oversampling dan undersampling model TCM sebesar 0,849476. Untuk ROC/AUC dan Akurasi memberikan nilai sebesar 0,85806 dan 0,86066. Untuk dataset PC4 semua hasil evaluasinya setelah melakukan prediksi menggunakan dataset oversampling dan undersampling model TCM memberikan nilai terbaik dibandingkan model yang lainnya. Untuk nilai f-measure yang dihasilkan senilai 0,93920 dan akurasi sebesar 0,934091.



Gambar 4.9 Perbandingan hasil evaluasi tiap model pada prediksi CNN menggunakan dataset (PC1, PC4, PC5, JM1 dan MC1)

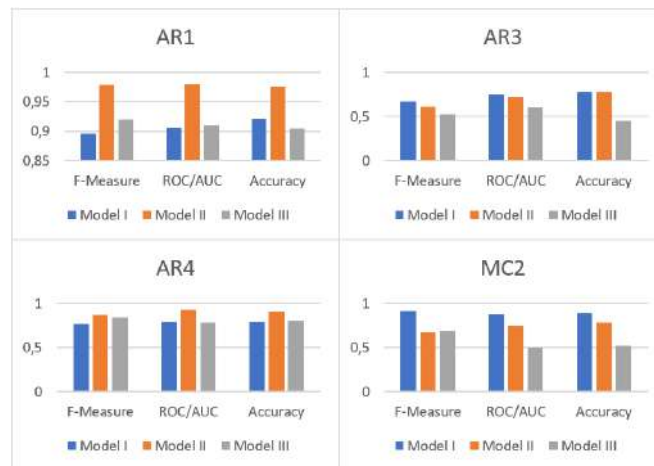
Pada Gambar 4.28 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset KC1, KC2, KC3, MW1 dan CM1. Evaluasi yang dihasilkan oversampling dan undersampling pada dataset KC1, KC3 dan CM1 memberikan nilai tertinggi ketika menggunakan model SCM. Untuk KC1 nilai F-Measure dihasilkan sebesar 0,89226. Untuk KC3 nilai F-Measure dihasilkan sebesar 0,86956. Untuk CM1 nilai F-Measure dihasilkan sebesar 0,928. Dan pengukuran akurasi pada ketiga dataset tersebut, KC1 memberikan nilai 0,892749. KC3

memberikan nilai sebesar 0,898305 dan CM1 memberikan nilai sebesar 0,92105. Pada dataset KC2, nilai evaluasi terbaik dihasilkan ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model TCM dengan F-Measure senilai 0,91111 dan akurasi senilai 0,88889. Untuk MW1, setiap pengukuran evaluasi memberikan hasil yang berbeda beda. Untuk F-Measure dan akurasi, nilai terbaik diberikan ketika prediksi dilakukan menggunakan model FCM senilai 0,941176 dan 0,934211.



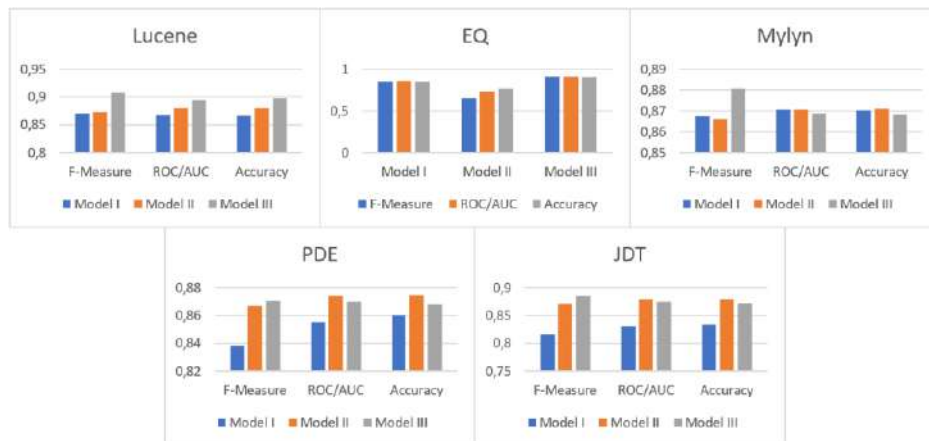
Gambar 4.10 Perbandingan hasil evaluasi tiap model pada prediksi CNN menggunakan dataset (KC1, KC2, KC3, MW1 dan CM1)

Pada Gambar 4.29 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset AR1, AR3, AR4 dan MC2. Evaluasi yang dihasilkan oversampling dan undersampling pada dataset AR1 dan AR4 memberikan nilai evaluasi tertinggi ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model SCM. Nilai F-Measure yang dihasilkan kedua data tersebut senilai 0,97872 dan 0,86956. Untuk nilai akurasi dari kedua data tersebut memberikan sebesar 0,97619 dan 0,903226. Pada dataset AR4 dan MC2, keduanya memberikan hasil evaluasi terbaiknya ketika menggunakan data kombinasi oversampling dan undersampling model FCM. Ketika pengukuran F-Measure dilakukan, AR3 dan MC2 mampu memberikan nilai sebesar 0,67 dan 0,909091. Untuk pengukuran akurasi memberikan nilai sebesar 0,78 dan 0,89.



Gambar 4.11 Perbandingan hasil evaluasi tiap model pada prediksi CNN menggunakan dataset (AR1, AR2, AR3 dan MC2)

Pada Gambar 4.30 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset Lucene, EQ, Mylyn, PDE dan JDT. Evaluasi yang dihasilkan ketika melakukan prediksi menggunakan dataset oversampling dan undersampling Lucene dan EQ memberikan nilai evaluasi tertinggi ketika model kombinasi oversampling dan undersampling ketiga digunakan. Pada pengukuran F-Measure kedua dataset tersebut, Lucene menghasilkan nilai evaluasi sebesar 0,907692 dan EQ memberikan nilai evaluasi sebesar 0,914286. Untuk pengukuran akurasi, Lucene memberikan nilai akurasi sebesar 0,89743 dan EQ dengan nilai sebesar 0,90322. Untuk dataset Mylyn, PDE dan JDT, memiliki karakteristik yang sama terkait pengukuran evaluasi hasil prediksi dimana F-Measure tertinggi dihasilkan ketika data kombinasi oversampling dan undersampling model TCM digunakan dan untuk pengukuran evaluasi lainnya seperti ROC/AUC dan akurasi memberikan hasil terbaik ketika menggunakan data kombinasi oversampling dan undersampling model SCM. Untuk nilai F-Measure pada dataset Mylyn memberikan nilai 0,880503 dan untuk dataset PDE memberikan nilai 0,870488 dan JDT memberikan nilai evaluasi F-Measure sebesar 0,88535. Untuk evaluasi yang dihasilkan Mylyn memberikan nilai sebesar 0,87107 dan PDE memberikan nilai sebesar 0,87474 dan JDI memberikan nilai akurasi sebesar 0,87972.



Gambar 4.12 Perbandingan hasil evaluasi tiap model pada prediksi CNN menggunakan dataset (Lucene, EQ, Mylyn, PDE dan JDT)

Dari semua perbandingan yang telah ditunjukkan pada Gambar 4.27 hingga 4.30. menunjukkan bahwa setiap model tidak ada yang mampu mendominasi ketika dilakukan prediksi menggunakan CNN. Tidak seperti hasil perbandingan prediksi ANN, hanya terdapat delapan dataset yang menggunakan oversampling dan undersampling model TCM yang mampu memberikan nilai F-Measure terbaik. Pada pengukuran ROC/AUC dan Akurasi, terdapat 10 dataset yang memberikan nilai evaluasi terbaik di kedua pengukuran tersebut.

#### 4. 5. Implementasi Model Prediksi RNN

Pada bagian ini data kombinasi oversampling dan undersampling yang menerapkan ketiga model yang diusulkan pada penelitian ini dilakukan prediksi menggunakan RNN. Bentuk hasil prediksi ini sama seperti sub bab sebelumnya akan menampilkan nilai F-Measure, ROC/AUC dan akurasi. Pada Tabel 4.29 merupakan hasil prediksi menggunakan data hasil oversampling dan undersampling model FCM. Dari hasil tersebut dapat diketahui ketika prediksi dilakukan pada dataset KC3, memberikan hasil evaluasi paling kecil dengan nilai F-Measure senilai 0,784321. dan Nilai F-Measure tertinggi didapatkan ketika PC5 digunakan untuk melakukan prediksi dengan nilai sebesar 0,99497.

Tabel 4.30 Hasil Prediksi RNN menggunakan Data kombinasi oversampling dan undersampling Model FCM

Dataset	F-Measure	Precision	Recall	ROC/AUC	Accuracy
PC1	0,94323	0,93913	0,94737	0,94634	0,94628

PC4	0,97727	0,96629	0,98851	0,98049	0,97959
PC5	0,99497	0,99874	0,99123	0,99499	0,99501
AR1	0,88889	0,80000	1,00000	0,90909	0,89474
AR3	0,85714	1,00000	0,75000	0,87500	0,88889
AR4	0,85714	0,81818	0,90000	0,87857	0,87500
KC1	0,86142	0,85185	0,87121	0,87025	0,87018
KC2	0,93793	0,93151	0,94444	0,92222	0,92623
KC3	0,78431	0,74074	0,83333	0,75758	0,76087
MC1	0,99213	0,99240	0,99186	0,99208	0,99208
MC2	0,80000	0,88889	0,72727	0,79221	0,77778
CM1	0,95745	0,93750	0,97826	0,95255	0,95402
JM1	0,82676	0,82601	0,82752	0,81989	0,82017
MW1	0,94737	0,94737	0,94737	0,94737	0,94737
EQ	0,85714	0,80769	0,91304	0,87031	0,86538
JDT	0,82609	0,83333	0,81897	0,83641	0,83740
Lucene	0,88288	0,90741	0,85965	0,88175	0,88073
Mylyn	0,85010	0,90393	0,80233	0,85734	0,85658
PDE	0,84946	0,86813	0,83158	0,86074	0,86275

Pada penggunaan data kombinasi oversampling dan undersampling model SCM untuk melakukan prediksi, hasil evaluasi ditunjukkan pada Tabel 4.30. Dari tabel tersebut ditunjukkan bahwa dataset MC2 memberikan nilai F-Measure terkecil dibandingkan dengan hasil F-Measure lainnya yang dihasilkan dataset lain senilai 0,78. Pada tabel tersebut juga ditunjukkan ketika MC1 digunakan dalam melakukan prediksi, nilai evaluasi F-Measure yang dihasilkan menjadi nilai evaluasi tertinggi dimana nilai yang dihasilkan sebesar 0,99477.

Tabel 4.31 Hasil Prediksi RNN menggunakan Data kombinasi oversampling dan undersampling Model SCM

<b>Dataset</b>	<b>F-Measure</b>	<b>Precision</b>	<b>Recall</b>	<b>ROC/AUC</b>	<b>Accuracy</b>
PC1	0,94615	0,91111	0,98400	0,95062	0,94815
PC4	0,94831	0,93778	0,95909	0,95097	0,95054
PC5	0,84349	0,73664	0,98660	0,81688	0,81691
AR1	0,97872	1,00000	0,95833	0,97917	0,97674
AR3	0,95652	1,00000	0,91667	0,95833	0,95455
AR4	0,83333	0,71429	1,00000	0,90909	0,87500
KC1	0,90000	0,90566	0,89441	0,90309	0,90332
KC2	0,85714	0,82192	0,89552	0,87038	0,86755
KC3	0,93750	0,90909	0,96774	0,93030	0,93220
MC1	0,99477	0,99231	0,99724	0,99485	0,99482

MC2	0,77778	0,63636	1,00000	0,90000	0,85185
CM1	0,95313	0,93846	0,96825	0,94413	0,94690
JM1	0,84954	0,90077	0,80382	0,86410	0,86885
MW1	0,97826	0,95745	1,00000	0,97826	0,97802
EQ	0,80851	0,86364	0,76000	0,83588	0,84746
JDT	0,89474	0,90840	0,88148	0,90228	0,90378
Lucene	0,87764	0,83871	0,92035	0,88266	0,88017
Mylyn	0,90671	0,87937	0,93581	0,90642	0,90579
PDE	0,88073	0,94118	0,82759	0,89026	0,89322

Pada penggunaan data kombinasi oversampling dan undersampling model TCM untuk melakukan prediksi, hasil evaluasi ditunjukkan pada Tabel 4.31. Dari tabel tersebut ditunjukkan bahwa dataset PC5 memberikan nilai F-Measure terkecil dibandingkan dengan hasil F-Measure lainnya yang dihasilkan dataset lain senilai 0,84308. Pada tabel tersebut juga ditunjukkan ketika MC1 digunakan dalam melakukan prediksi, nilai evaluasi F-Measure yang dihasilkan menjadi nilai evaluasi tertinggi dimana nilai yang dihasilkan sebesar 0,98959. Untuk pengukuran nilai akurasi, MC1 memberikan nilai evaluasi tertinggi ketika menggunakan data kombinasi oversampling dan undersampling model TCM yaitu sebesar 0,98965 dan untuk dataset PC5 memberikan nilai evaluasi terkecil sebesar 0,81568.

Tabel 4.32 Hasil Prediksi RNN menggunakan Data kombinasi oversampling dan undersampling Model TCM

<b>Dataset</b>	<b>F-Measure</b>	<b>Precision</b>	<b>Recall</b>	<b>ROC/AUC</b>	<b>Accuracy</b>
PC1	0,93950	0,89189	0,99248	0,93374	0,93487
PC4	0,96356	0,94444	0,98347	0,95638	0,95909
PC5	0,84308	0,73927	0,98082	0,81407	0,81568
AR1	0,92593	0,89286	0,96154	0,88702	0,90476
AR3	0,92308	0,85714	1,00000	0,96429	0,95000
AR4	0,85714	0,83333	0,88235	0,82579	0,83333
KC1	0,89240	0,91471	0,87115	0,88416	0,88263
KC2	0,94382	0,95455	0,93333	0,92963	0,93056
KC3	0,89552	0,90909	0,88235	0,87299	0,87500
MC1	0,98959	0,98742	0,99176	0,98967	0,98965
MC2	0,92308	0,92308	0,92308	0,91987	0,92000
CM1	0,92800	0,95082	0,90625	0,91563	0,91346
JM1	0,85570	0,91137	0,80645	0,84344	0,83575
MW1	0,97087	1,00000	0,94340	0,97170	0,96512
EQ	0,91429	0,96970	0,86486	0,91243	0,90323

JDT	0,89320	0,93878	0,85185	0,88843	0,88298
Lucene	0,90909	0,90551	0,91270	0,90079	0,90171
Mylyn	0,86780	0,95880	0,79257	0,87472	0,86505
PDE	0,86607	0,95567	0,79184	0,87518	0,87013

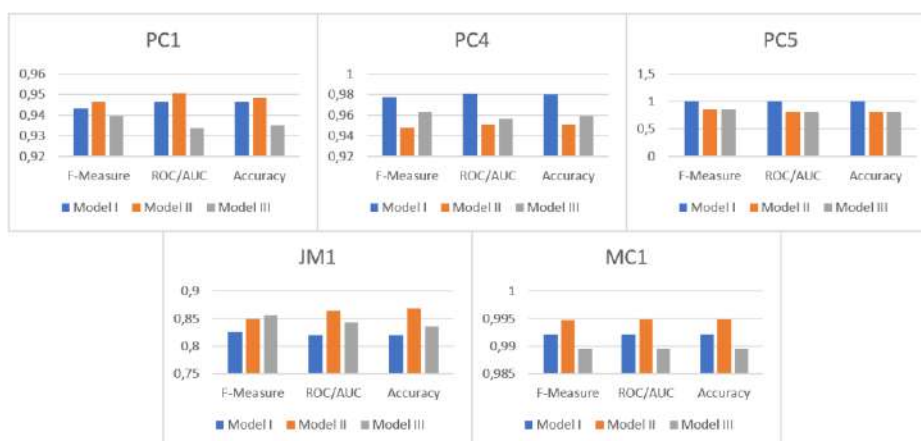
Jika dibandingkan tiap modelnya bisa diketahui bahwa penerapan yang memberikan nilai FMeasure paling stabil pada ketiga model ditunjukkan pada dataset MC1. Pada semua model dataset oversampling dan undersampling yang digunakan dalam prediksi, mampu memberikan hasil evaluasi sama baiknya. Sedangkan oversampling dan undersampling model SCM pada dataset MC2 memberikan hasil terkecil dibandingkan hasil hasil prediksi menggunakan berbagai model kombinasi oversampling dan undersampling.

Pengukuran ROC/AUC yang dihasilkan pada model ini dapat dilihat pada Gambar 4.2 yang menunjukkan bahwa dataset MC1 memberikan hasil evaluasi terbaik seperti ketika pengukuran evaluasi menggunakan F-Measure. Dari semua dataset hasil oversampling dan undersampling berbagai macam model, pengukuran evaluasinya memberikan hasil yang baik dimana nilai paling kecil berada pada nilai 0,75757 yang diberikan oleh dataset KC3 oversampling dan undersampling model FCM. Pengukuran akurasi yang dihasilkan sama seperti pengukuran lainnya dimana memberikan hasil yang baik dan nilai terendah diberikan KC3 senilai 0,76087.

Evaluasi yang dihasilkan ketika melakukan prediksi menggunakan dataset oversampling dan undersampling PC1 dan MC1 memberikan nilai evaluasi tertinggi ketika model kombinasi oversampling dan undersampling kedua digunakan. Dimana nilai F-Measure yang dihasilkan pada dataset PC1 sebesar 0,94615 dan dataset MC1 memberikan hasil F-Measure sebesar 0,99476. Untuk pengukuran akurasi yang dihasilkan PC1 dan MC1 sebesar 0,94814 dan 0,99482. Pada pengukuran evaluasi PC4 dan PC5, metode oversampling dan undersampling yang berhasil memberikan hasil evaluasi ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model FCM. Hasil prediksi menggunakan data kombinasi oversampling dan undersampling model tersebut memberikan nilai F-Measure pada PC4 sebesar 0,97727 dan PC5 memberikan nilai sebesar 0,99496. Untuk pengukuran akurasi, PC4 dan PC5 yang dioversampling dan undersampling menggunakan model FCM setelah melakukan prediksi



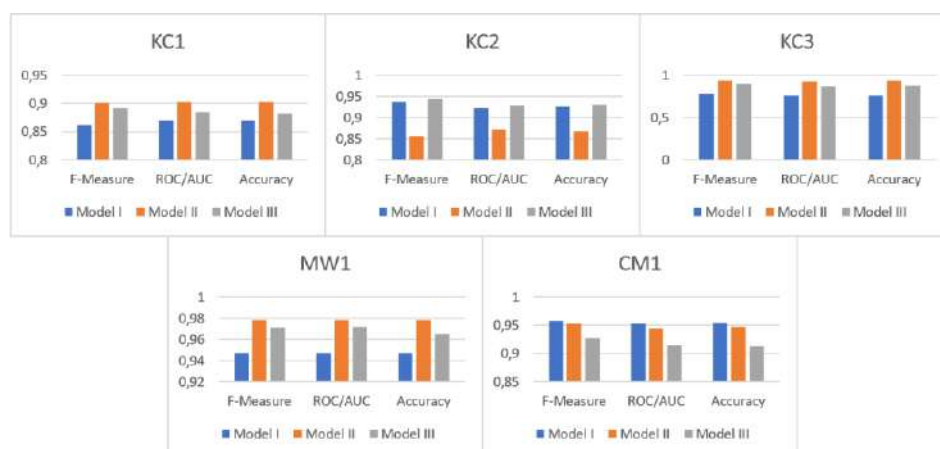
memberikan nilai evaluasi sebesar 0,97959 dan 0,9958. Untuk dataset JM1, hasil setiap pengukuran evaluasi yang digunakan pada penelitian ini memberikan hasil yang berbeda beda. Ketika dilakukan pengukuran F-Measure, Data kombinasi oversampling dan undersampling JM1 menggunakan model TCM memberikan nilai terbaik sebesar 0,8557. Dan pengukuran akurasi pada JM1, hasil terbaik diberikan ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model SCM dengan nilai sebesar 0,86884.



Gambar 4.13 Perbandingan hasil evaluasi tiap model pada prediksi RNN menggunakan dataset (PC1, PC4, PC5, JM1 dan MC1)

Pada Gambar 4.35 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset KC1, KC2, KC3, MW1 dan CM1. Evaluasi yang dihasilkan ketika melakukan prediksi menggunakan dataset oversampling dan undersampling KC1, MW1 dan KC3 memberikan nilai evaluasi tertinggi ketika model kombinasi oversampling dan undersampling kedua digunakan. Ketika prediksi dilakukan pengukuran evaluasi menggunakan F-Measure, KC1 memberikan nilai sebesar 0,9 dan MW1 memberikan nilai sebesar 0,97826 dan KC3 memberikan nilai evaluasi sebesar 0,9375. Untuk pengukuran akurasi pada ketiga data tersebut, pada KC1 memberikan nilai akurasi sebesar 0,9033 dan MW1 memberikan nilai akurasi 0,97802 dan KC3 memberikan nilai akurasi sebesar 0,93220. Pada dataset KC2, ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model TCM, hasil evaluasi pada semua pengukuran memberikan hasil yang baik. Pengukuran F-Measure dari data KC2 dengan menggunakan

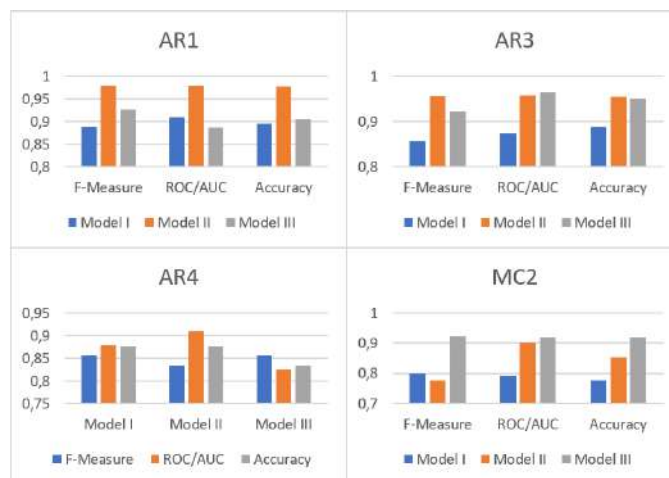
oversampling dan undersampling model TCM, memberikan nilai sebesar 0,94382 dan nilai akurasi sebesar 0,93055.



Gambar 4.14 Perbandingan hasil evaluasi tiap model pada prediksi RNN menggunakan dataset (KC1, KC2, KC3, MW1 dan CM1)

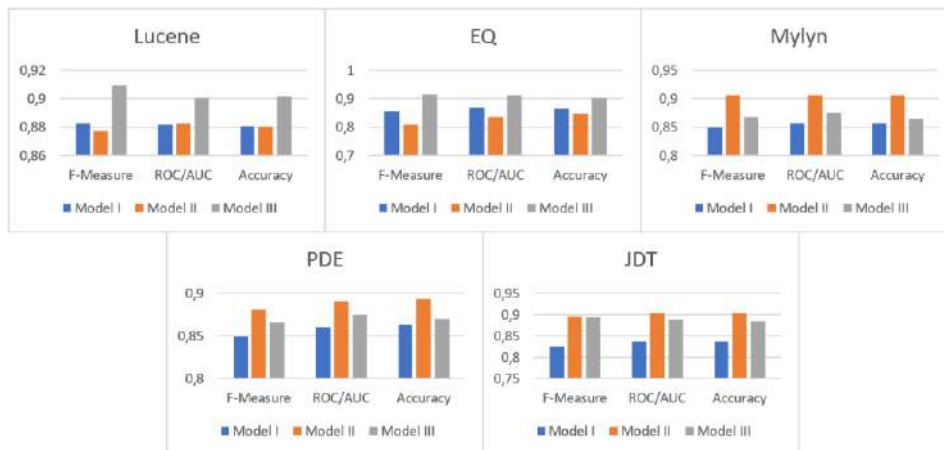
Pada Gambar 4.36 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset AR1, AR3, AR4 dan MC2. Evaluasi yang dihasilkan ketika melakukan prediksi menggunakan dataset oversampling dan undersampling AR1 memberikan nilai evaluasi tertinggi ketika model kombinasi oversampling dan undersampling kedua digunakan. dimana pengukuran F-Measure pada hasil prediksi dataset tersebut memberikan nilai 0,97872 dan pengukuran akurasi memberikan nilai 0,976744. Untuk dataset MC2, ketika prediksi dilakukan menggunakan data kombinasi oversampling dan undersampling model SCM, evaluasi yang dihasilkan dari pengukuran F-Measure memberikan nilai sebesar 0,92307 dan akurasi yang diberikan sebesar 0,92. Dan untuk AR4, hasil evaluasi yang dihasilkan untuk F-measure diberikan nilai terbaik ketika prediksi menggunakan dataset resampling model SCM dengan nilai sebesar 0,85714 dan akurasi terbaik diberikan ketika dataset oversampling dan undersampling model

FCM digunakan. pengukuran akurasi pada prediksi tersebut memberikan nilai sebesar 0,85714.



Gambar 4.15 Perbandingan hasil evaluasi tiap model pada prediksi RNN menggunakan dataset (AR1, AR2, AR3 dan MC2)

Pada Gambar 4.37 merupakan perbandingan evaluasi model FCM, kedua dan ketiga pada dataset Lucene, EQ, Mylyn, PDE dan JDT. Evaluasi yang dihasilkan ketika melakukan prediksi menggunakan dataset oversampling dan undersampling Lucene dan EQ memberikan nilai evaluasi tertinggi ketika model kombinasi oversampling dan undersampling ketiga digunakan saat melakukan prediksi. Hasil pengukuran evaluasi F-Measure pada data Lucene memberikan nilai sebesar 0,90909 dan EQ memberikan nilai sebesar 0,91428. Untuk pengukuran akurasi yang dihasilkan pada kedua data tersebut senilai 0,9017 dan 0,90322. Pada dataset Mylyn, JDT dan PDE, ketika dataset tersebut memiliki karakteristik yang sama dimana hasil evaluasi prediksi memberikan hasil terbaik ketika menggunakan data kombinasi oversampling dan undersampling dataset tersebut menggunakan model SCM. Pengukuran F-Measure yang dilakukan pada ketiga dataset tersebut, untuk Mylyn memberikan nilai evaluasi sebesar 0,906 dan JDT memberikan nilai evaluasi sebesar 0,894 dan PDE memberikan nilai evaluasi sebesar 0,8807. Untuk pengukuran akurasi pada prediksi data tersebut, Mylyn memberikan nilai sebesar 0,9057 dan JDT memberikan nilai akurasi sebesar 0,9037 dan PDE memberikan nilai evaluasi sebesar 0,8932.



Gambar 4.16 Perbandingan hasil evaluasi tiap model pada prediksi RNN menggunakan dataset (Lucene, EQ, Mylyn, PDE dan JDT)

Dari semua perbandingan yang telah ditunjukkan pada Gambar 4.34 hingga 4.37. menunjukkan bahwa setiap model tidak ada yang mampu mendominasi ketika dilakukan prediksi menggunakan RNN. Tidak seperti hasil perbandingan prediksi ANN tetapi sama seperti hasil perbandingan evaluasi prediksi menggunakan CNN, tidak ada model yang mampu mendominasi hasil pengukuran evaluasi. Pada pengukuran F-Measure, terdapat 10 dataset yang memberikan hasil terbaiknya ketika dataset oversampling dan undersampling model TCM saat melakukan prediksi. Pada pengukuran ROC/AUC dan Akurasi terdapat sekitar 10-11 dataset mampu memberikan hasil yang baik ketika dataset yang digunakan menerapkan oversampling dan undersampling model SCM.

#### 4. 6. Perbandingan Hasil Prediksi

Pada sub bab ini membandingkan keenam data yang digunakan dalam prediksi ANN, CNN dan RNN. Data-data ini antara lain adalah data Original yang diberinama Ori, data hasil dari RSMOTE, data hasil dari ENN dan data hasil dari ketiga model kombinasi yang diusulkan pada penelitian ini (FCM, SCM dan TCM). Pada Tabel 4.33 merupakan perbandingan nilai F-Measure yang dihasilkan oleh penggunaan model prediksi ANN. Dari kelima data model yang digunakan dapat dilihat bahwa ketiga metode yang diusulkan pada penelitian ini memberikan hasil yang lebih baik dibandingkan dengan data Original, ENN dan RSMOTE. Penggunaan data Original dan ENN memberikan hasil yang tidak stabil, dimana

pada beberapa dataset seperti PC1, AR1 dan AR4 tidak mampu memberikan hasil evaluasi F-Measure dimana setelah berulang kali diuji sering kali nilai yang diberikan sebesar 0 (*zero*). Meski pada penggunaan data ENN pada KC2 dan EQ mampu memberikan hasil yang baik tapi model ENN tidak mampu memberikan hasil yang stabil pada dataset lain sehingga model ENN tidak direkomendasikan untuk ruang lingkup permasalahan ini. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan ANN memberikan nilai rata-rata hasil pengukuran F-Measure sebesar 0.853, 0,816 dan 0.865 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.312, 0.423 dan 0.807. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.33 Perbandingan Hasil F-Measure pada Model Prediksi ANN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,000	0,308	0,830	0,930	0,881	0,894
PC4	0,538	0,649	0,915	0,955	0,901	0,921
PC5	0,351	0,561	0,830	0,665	0,452	0,840
AR1	0,000	0,000	0,917	0,857	0,979	0,939
AR3	0,000	0,000	0,941	0,750	0,941	0,545
AR4	0,000	0,333	0,815	0,842	0,833	0,800
KC1	0,356	0,468	0,836	0,813	0,880	0,885
KC2	0,526	0,923	0,875	0,923	0,357	0,904
KC3	0,391	0,000	0,818	0,783	0,816	0,877
MC1	0,462	0,476	0,988	0,987	0,989	0,983
MC2	0,000	0,667	0,706	0,909	0,667	0,727
CM1	0,000	0,154	0,855	0,875	0,911	0,903
JM1	0,090	0,474	0,793	0,779	0,841	0,843
MW1	0,153	0,000	0,905	0,872	0,920	0,962
EQ	0,727	0,949	0,815	0,894	0,650	0,901
JDT	0,552	0,725	0,819	0,811	0,884	0,885
Lucene	0,333	0,222	0,275	0,862	0,877	0,900
Mylyn	0,188	0,464	0,866	0,854	0,848	0,864
PDE	0,333	0,526	0,842	0,854	0,876	0,857

Pada Tabel 4.34, menunjukkan hasil perbandingan dari model prediksi CNN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika

dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan CNN memberikan nilai rata-rata hasil pengukuran F-Measure sebesar 0.854, 0,849 dan 0.860 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.210, 0.410 dan 0.840. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.34 Perbandingan Hasil F-Measure pada Model Prediksi CNN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,000	0,211	0,854	0,931	0,867	0,888
PC4	0,571	0,632	0,914	0,920	0,916	0,939
PC5	0,380	0,527	0,838	0,985	0,840	0,839
AR1	0,000	0,000	0,917	0,897	0,979	0,920
AR3	0,000	0,000	0,800	0,667	0,615	0,522
AR4	0,000	0,333	0,786	0,762	0,870	0,842
KC1	0,273	0,500	0,826	0,757	0,892	0,871
KC2	0,526	0,927	0,850	0,901	0,803	0,911
KC3	0,000	0,429	0,818	0,714	0,870	0,829
MC1	0,000	0,000	0,978	0,984	0,980	0,969
MC2	0,000	0,818	0,621	0,909	0,667	0,684
CM1	0,000	0,000	0,864	0,857	0,928	0,893
JM1	0,170	0,496	0,790	0,766	0,846	0,849
MW1	0,000	0,400	0,925	0,941	0,930	0,932
EQ	0,750	0,949	0,769	0,846	0,650	0,914
JDT	0,533	0,693	0,818	0,816	0,871	0,885
Lucene	0,333	0,235	0,884	0,870	0,872	0,908
Mylyn	0,125	0,419	0,870	0,867	0,866	0,881
PDE	0,320	0,473	0,832	0,839	0,867	0,870

Pada Tabel 4.35, menunjukkan hasil perbandingan dari model prediksi RNN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan RNN memberikan nilai rata-rata hasil pengukuran F-Measure sebesar 0.889, 0,901 dan 0.911 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.312, 0.423 dan 0.807. Sehingga bisa

diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.35 Perbandingan Hasil F-Measure pada Model Prediksi RNN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,286	0,286	0,823	0,943	0,946	0,940
PC4	0,000	0,080	0,850	0,977	0,948	0,964
PC5	0,458	0,449	0,836	0,995	0,843	0,843
AR1	0,000	0,000	0,830	0,889	0,979	0,926
AR3	0,500	0,000	0,615	0,857	0,957	0,923
AR4	0,667	0,600	0,800	0,857	0,833	0,857
KC1	0,383	0,529	0,795	0,861	0,900	0,892
KC2	0,526	0,884	0,850	0,938	0,857	0,944
KC3	0,000	0,571	0,776	0,784	0,938	0,896
MC1	0,000	0,000	0,964	0,992	0,995	0,990
MC2	0,400	0,833	0,605	0,800	0,778	0,923
CM1	0,000	0,154	0,815	0,957	0,953	0,928
JM1	0,268	0,450	0,793	0,827	0,850	0,856
MW1	0,500	0,667	0,923	0,947	0,978	0,971
EQ	0,667	0,918	0,815	0,857	0,809	0,914
JDT	0,444	0,639	0,791	0,826	0,895	0,893
Lucene	0,400	0,118	0,887	0,883	0,878	0,909
Mylyn	0,229	0,393	0,755	0,850	0,907	0,868
PDE	0,200	0,473	0,819	0,849	0,881	0,866

Pada Tabel 4.36, menunjukkan hasil perbandingan dari model prediksi ANN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan ANN memberikan nilai rata-rata hasil pengukuran ROC/AUC sebesar 0.849, 0,855 dan 0.848 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.554, 0.613 dan 0.843. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.36 Perbandingan Hasil ROC/AUC pada Model Prediksi ANN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM

PC1	0,500	0,596	0,829	0,925	0,889	0,893
PC4	0,698	0,749	0,916	0,961	0,906	0,909
PC5	0,616	0,720	0,798	0,500	0,628	0,810
AR1	0,500	0,500	0,915	0,878	0,979	0,942
AR3	0,500	0,000	0,944	0,792	0,964	0,643
AR4	0,000	0,600	0,855	0,864	0,905	0,701
KC1	0,608	0,660	0,843	0,833	0,889	0,873
KC2	0,684	0,763	0,883	0,908	0,588	0,880
KC3	0,622	0,500	0,825	0,784	0,839	0,811
MC1	0,650	0,656	0,988	0,987	0,989	0,983
MC2	0,364	0,568	0,716	0,875	0,741	0,628
CM1	0,500	0,533	0,865	0,867	0,904	0,888
JM1	0,521	0,653	0,805	0,783	0,856	0,819
MW1	0,538	0,500	0,905	0,868	0,930	0,957
EQ	0,728	0,940	0,808	0,905	0,731	0,892
JDT	0,711	0,789	0,837	0,823	0,891	0,875
Lucene	0,617	0,568	0,580	0,864	0,882	0,886
Mylyn	0,549	0,669	0,871	0,857	0,858	0,862
PDE	0,615	0,688	0,843	0,865	0,885	0,861

Pada Tabel 4.37, menunjukkan hasil perbandingan dari model prediksi CNN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan CNN memberikan nilai rata-rata hasil pengukuran ROC/AUC sebesar 0.863, 0.869 dan 0.841 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.541, 0.622 dan 0.852. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.37 Perbandingan Hasil ROC/AUC pada Model Prediksi CNN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,486	0,559	0,863	0,924	0,876	0,894
PC4	0,723	0,739	0,916	0,927	0,920	0,935
PC5	0,634	0,722	0,808	0,985	0,810	0,810
AR1	0,500	0,500	0,923	0,906	0,979	0,911
AR3	0,500	0,000	0,857	0,750	0,714	0,607
AR4	0,000	0,600	0,832	0,793	0,929	0,778
KC1	0,578	0,680	0,832	0,786	0,893	0,869



KC2	0,684	0,777	0,861	0,878	0,826	0,881
KC3	0,500	0,619	0,825	0,744	0,885	0,767
MC1	0,500	0,500	0,979	0,983	0,980	0,969
MC2	0,409	0,742	0,704	0,875	0,741	0,500
CM1	0,500	0,500	0,868	0,851	0,920	0,884
JM1	0,543	0,664	0,811	0,777	0,858	0,825
MW1	0,500	0,625	0,927	0,932	0,940	0,923
EQ	0,759	0,940	0,771	0,858	0,731	0,912
JDT	0,705	0,767	0,831	0,831	0,879	0,875
Lucene	0,625	0,573	0,880	0,868	0,880	0,894
Mylyn	0,528	0,641	0,870	0,871	0,871	0,869
PDE	0,611	0,661	0,827	0,855	0,874	0,870

Pada Tabel 4.38, menunjukkan hasil perbandingan dari model prediksi RNN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan RNN memberikan nilai rata-rata hasil pengukuran ROC/AUC sebesar 0.892, 0.914 dan 0.903 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.602, 0.624 dan 0.808. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.38 Perbandingan Hasil ROC/AUC pada Model Prediksi RNN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,611	0,592	0,809	0,946	0,951	0,934
PC4	0,500	0,521	0,845	0,980	0,951	0,956
PC5	0,670	0,652	0,803	0,995	0,817	0,814
AR1	0,458	0,500	0,791	0,909	0,979	0,887
AR3	0,650	0,000	0,722	0,875	0,958	0,964
AR4	0,750	0,709	0,857	0,879	0,909	0,826
KC1	0,619	0,696	0,798	0,870	0,903	0,884
KC2	0,684	0,928	0,861	0,922	0,870	0,930
KC3	0,474	0,698	0,778	0,758	0,930	0,873
MC1	0,500	0,500	0,965	0,992	0,995	0,990
MC2	0,659	0,705	0,475	0,792	0,900	0,920
CM1	0,500	0,533	0,832	0,953	0,944	0,916
JM1	0,572	0,642	0,800	0,820	0,864	0,843
MW1	0,667	0,750	0,927	0,947	0,978	0,972

EQ	0,733	0,888	0,808	0,870	0,836	0,912
JDT	0,652	0,735	0,804	0,836	0,902	0,888
Lucene	0,625	0,523	0,884	0,882	0,883	0,901
Mylyn	0,562	0,629	0,753	0,857	0,906	0,875
PDE	0,556	0,661	0,834	0,861	0,890	0,875

Pada Tabel 4.39, menunjukkan hasil perbandingan dari model prediksi ANN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan ANN memberikan nilai rata-rata hasil pengukuran Akurasi sebesar 0.849, 0.860 dan 0.843 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.809, 0.626 dan 0.839. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.39 Perbandingan Hasil Akurasi pada Model Prediksi ANN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,611	0,592	0,809	0,946	0,951	0,934
PC4	0,500	0,521	0,845	0,980	0,951	0,956
PC5	0,670	0,652	0,803	0,995	0,817	0,814
AR1	0,458	0,500	0,791	0,909	0,979	0,887
AR3	0,650	0,000	0,722	0,875	0,958	0,964
AR4	0,750	0,709	0,857	0,879	0,909	0,826
KC1	0,619	0,696	0,798	0,870	0,903	0,884
KC2	0,684	0,928	0,861	0,922	0,870	0,930
KC3	0,474	0,698	0,778	0,758	0,930	0,873
MC1	0,500	0,500	0,965	0,992	0,995	0,990
MC2	0,659	0,705	0,475	0,792	0,900	0,920
CM1	0,500	0,533	0,832	0,953	0,944	0,916
JM1	0,572	0,642	0,800	0,820	0,864	0,843
MW1	0,667	0,750	0,927	0,947	0,978	0,972
EQ	0,733	0,888	0,808	0,870	0,836	0,912
JDT	0,652	0,735	0,804	0,836	0,902	0,888
Lucene	0,625	0,523	0,884	0,882	0,883	0,901
Mylyn	0,562	0,629	0,753	0,857	0,906	0,875
PDE	0,556	0,661	0,834	0,861	0,890	0,875

Pada Tabel 4.40, menunjukkan hasil perbandingan dari model prediksi CNN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan CNN memberikan nilai rata-rata hasil pengukuran Akurasi sebesar 0.866, 0.875 dan 0.834 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.811, 0.691 dan 0.847. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan.

Tabel 4.40 Perbandingan Hasil Akurasi pada Model Prediksi CNN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,611	0,592	0,809	0,946	0,951	0,934
PC4	0,500	0,521	0,845	0,980	0,951	0,956
PC5	0,670	0,652	0,803	0,995	0,817	0,814
AR1	0,458	0,500	0,791	0,909	0,979	0,887
AR3	0,650	0,000	0,722	0,875	0,958	0,964
AR4	0,750	0,709	0,857	0,879	0,909	0,826
KC1	0,619	0,696	0,798	0,870	0,903	0,884
KC2	0,684	0,928	0,861	0,922	0,870	0,930
KC3	0,474	0,698	0,778	0,758	0,930	0,873
MC1	0,500	0,500	0,965	0,992	0,995	0,990
MC2	0,659	0,705	0,475	0,792	0,900	0,920
CM1	0,500	0,533	0,832	0,953	0,944	0,916
JM1	0,572	0,642	0,800	0,820	0,864	0,843
MW1	0,667	0,750	0,927	0,947	0,978	0,972
EQ	0,733	0,888	0,808	0,870	0,836	0,912
JDT	0,652	0,735	0,804	0,836	0,902	0,888
Lucene	0,625	0,523	0,884	0,882	0,883	0,901
Mylyn	0,562	0,629	0,753	0,857	0,906	0,875
PDE	0,556	0,661	0,834	0,861	0,890	0,875

Pada Tabel 4.41, menunjukkan hasil perbandingan dari model prediksi RNN pada keenam data yaitu Original, ENN, RSMOTE, FCM, SCM dan TCM. Hasilnya menunjukkan bahwa FCM, SCM dan TCM memberikan hasil yang lebih baik jika dibandingkan dengan Ori, ENN dan RSMOTE. Penggunaan FCM, SCM dan TCM pada prediksi menggunakan RNN memberikan nilai rata-rata hasil pengukuran

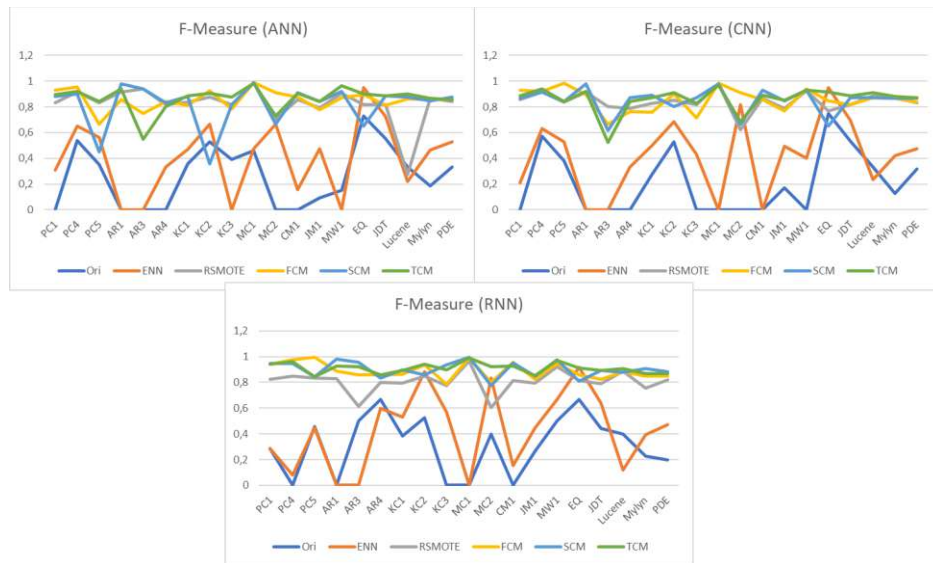
Akurasi sebesar 0.891, 0.910 dan 0.902 sedangkan penggunaan data Ori, ENN, RSMOTE memberikan hasil rata rata sebesar 0.802, 0.798 dan 0.810. Sehingga bisa diamati dimana hasil yang diberikan metode usulan pada penelitian ini memberikan hasil yang lebih baik, meski peningkatan nilai tidak signifikan. Dari ketiga model yang diusulkan, ketika prediksi dilakukan menggunakan data SCM, hasil prediksi memberikan nilai yang lebih baik ditunjukkan hanya PC4, PC5, AR3, MC2, CM1, EQ dan lucene yang memberikan hasil rendah tetapi masih lebih tinggi dari evaluasi hasil prediksi yang dilakukan menggunakan data original, enn dan rsmote.

Tabel 4.41 Perbandingan Hasil Akurasi pada Model Prediksi RNN

Dataset	Data Model					
	Ori	ENN	RSMOTE	FCM	SCM	TCM
PC1	0,611	0,592	0,809	0,946	0,951	0,934
PC4	0,500	0,521	0,845	0,980	0,951	0,956
PC5	0,670	0,652	0,803	0,995	0,817	0,814
AR1	0,458	0,500	0,791	0,909	0,979	0,887
AR3	0,650	0,000	0,722	0,875	0,958	0,964
AR4	0,750	0,709	0,857	0,879	0,909	0,826
KC1	0,619	0,696	0,798	0,870	0,903	0,884
KC2	0,684	0,928	0,861	0,922	0,870	0,930
KC3	0,474	0,698	0,778	0,758	0,930	0,873
MC1	0,500	0,500	0,965	0,992	0,995	0,990
MC2	0,659	0,705	0,475	0,792	0,900	0,920
CM1	0,500	0,533	0,832	0,953	0,944	0,916
JM1	0,572	0,642	0,800	0,820	0,864	0,843
MW1	0,667	0,750	0,927	0,947	0,978	0,972
EQ	0,733	0,888	0,808	0,870	0,836	0,912
JDT	0,652	0,735	0,804	0,836	0,902	0,888
Lucene	0,625	0,523	0,884	0,882	0,883	0,901
Mylyn	0,562	0,629	0,753	0,857	0,906	0,875
PDE	0,556	0,661	0,834	0,861	0,890	0,875

#### 4. 7. Analisis Hasil Penelitian

Dari hasil implementasi prediksi menggunakan ketiga data hasil metode yang diusulkan (FCM, SCM dan TCM) dan tiga data tambahan yaitu data Original, ENN dan RSMOTE, didapatkan bahwa dengan menggunakan metode usulan pada penelitian, prediksi yang dilakukan mengalami peningkatan ditunjukkan pada Gambar 4.2.



Gambar 4.1 Grafik Pengukuran F-Measure pada hasil prediksi menggunakan ANN, CNN dan RNN

Pada gambar tersebut dapat diamati untuk ketiga model usulan FCM berwarna kuning, SCM berwarna biru muda dan TCM berwarna hijau menunjukkan hasil yang lebih baik terutama jika menggunakan model prediksi RNN. Dimana hasil yang diberikan terlihat lebih stabil dibandingkan prediksi menggunakan ketiga data yang lain. Dalam menentukan model yang terbaik diantara FCM, SCM dan TCM, sembilan belas dataset yang digunakan pada penelitian ini, saat melakukan prediksi menggunakan ANN, sebelas dataset yang dihasilkan dengan melakukan oversampling dan undersampling menggunakan model SCM mampu memberikan hasil evaluasi yang baik.

Kombinasi model SCM akan melakukan oversampling terlebih dahulu kemudian melakukan undersampling. Melakukan oversampling pada data kelas minoritas, menjadikan data lebih seimbang jumlahnya dengan data kelas mayoritas, dengan menggunakan rsmote sebagai metode oversampling menciptakan data baru pada kelas minoritas yang lebih aman sehingga jarak (*border*) antar kelas masih tetap aman dan terhindar dari produksi data kelas minoritas (Chen et al., 2021). Hasil oversampling kemudian diterapkan metode undersampling, dengan metode ini menyebabkan data noise, duplikasi dan ambigu yang dihasilkan oleh oversampling akan terhapus sehingga menyebabkan informasi data train menjadi

lebih solid (Junnan Li et al., 2021). Jika dibandingkan dengan model yang lain, meski model SCM memberikan data yang kurang seimbang akan tetapi informasinya data asal masih tetap terjaga, berbeda dengan model kombinasi oversampling dan undersampling pertama.

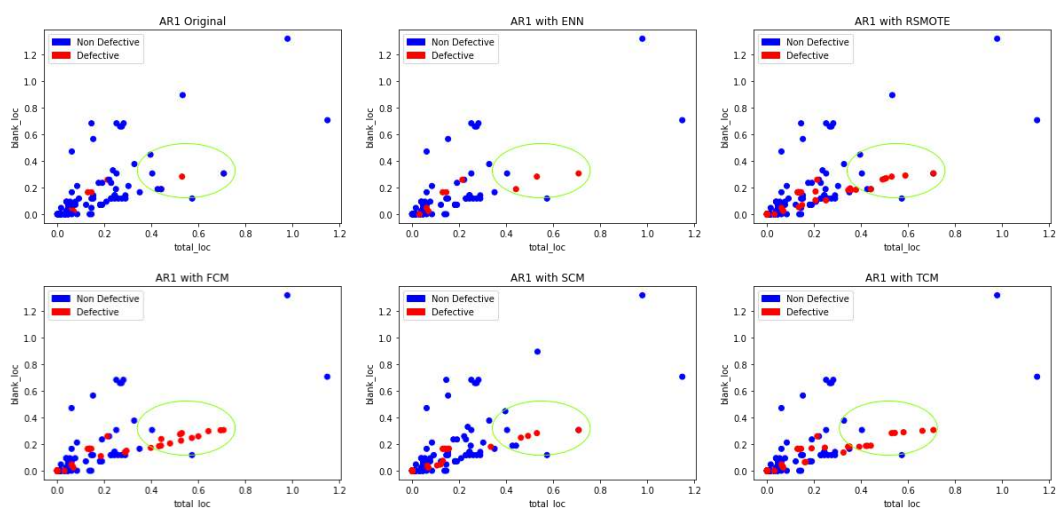
Model kombinasi oversampling dan undersampling pertama akan melakukan undersampling terlebih dahulu baru kemudian data hasil undersampling digunakan untuk melakukan oversampling. Dari alur model tersebut dapat diketahui bahwa data original yang digunakan untuk melakukan prediksi selanjutnya terjadi pengurangan akibat pembersihan data. Dengan penerapan itu menyebabkan data original bisa jadi mengalami kehilangan informasi. Hasil dari undersampling tersebut kemudian dilakukan oversampling untuk menyeimbangkan data. Hasil dari model ini sangat baik dimana jumlah data pada masing masing kelas menjadi seimbang.

Untuk model kombinasi oversampling dan undersampling ketiga, dataset yang digunakan untuk prediksi akan diterapkan oversampling dan undersampling secara bersamaan tetapi dilakukan terpisah. Hasil dari proses tersebut akan memberikan dua grup data hasil oversampling dan undersampling. Kemudian grup data oversampling dipisahkan dengan kelas mayoritasnya, begitu juga untuk grup data undersampling akan dipisahkan dengan kelas minoritasnya. Hasil pemisahan tersebut kemudian digabungkan sehingga pada dataset yang baru ini berisikan kelas minoritas baru yang dihasilkan proses oversampling dan kelas mayoritas baru yang dihasilkan dari proses undersampling (Park & Park, 2020). Hasil dari penggabungan menyebabkan data menjadi tidak seimbang seperti yang dihasilkan model FCM, tetapi menyebabkan kelas minoritas menjadi lebih banyak sehingga bisa memaksimalkan evaluasi prediksi dalam menentukan data yang bersifat kemungkinan terjadi *defect*.

Dari ketiga teori model kombinasi oversampling dan undersampling tersebut sehingga menjelaskan bahwa terdapat nilai positif dan negative pada masing-masing model. Dari ketiga model tersebut, oversampling dan undersampling model SCM masih memberikan hasil yang baik. tetapi bergantung dengan model prediksi yang digunakan serta tujuan dalam melakukan prediksi. Jika prediksi dilakukan menggunakan RNN, oversampling dan undersampling model SCM merupakan

pilihan terbaik karena mampu memaksimalkan potensi learning dan memberikan hasil yang terbaik berdasarkan pengukuran evaluasi F1-Measure, ROC/AUC dan Akurasi. Jike prediksi menggunakan ANN dan CNN, oversampling dan undersampling model SCM merupakan pilihan baik untuk prediksi untuk menentukan probabilitas dan menentukan label kelas pada suatu data dikarenakan pengukuran ROC/AUC dan Akurasi yang dihasilkan sangat baik. Tetapi pengukuran berdasarkan F1-Measure, oversampling dan undersampling model SCM untuk prediksi menggunakan ANN dan CNN kurang cocok, dan disarankan untuk menggunakan oversampling dan undersampling model TCM.

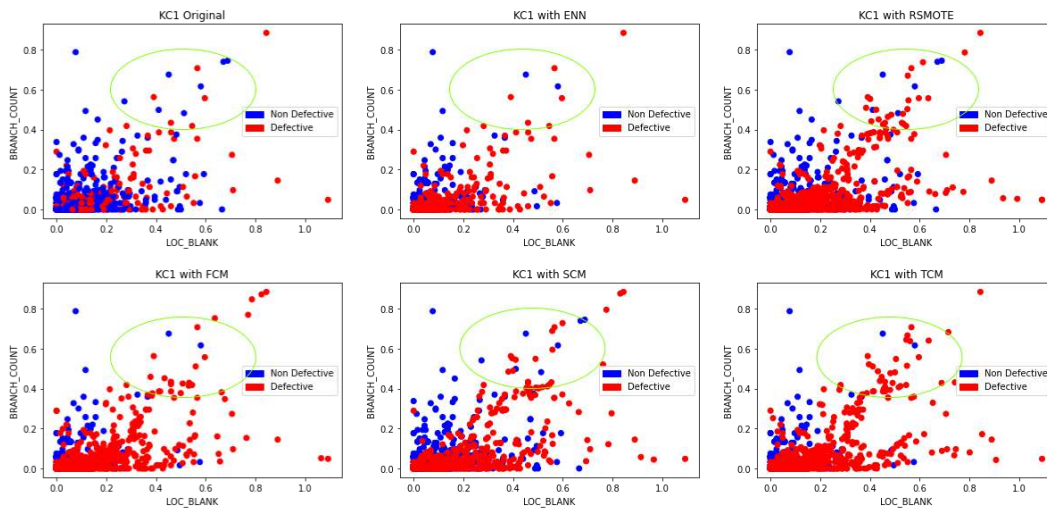
Pada Gambar 4.2 merupakan salah satu dataset (AR1) yang digunakan dalam penelitian ini dan untuk lebih lanjut bisa dilihat pada Lampiran A. Pada gambar tersebut menunjukkan distribusi data berdasarkan jarak antara fitur *total\_loc* dan *blank\_loc*. Didapati pada bagian yang dilingkar hijau didalamnya terdapat kelas *defective* (warna merah) dan *non defective* (warna biru), dengan model kombinasi yang diusulkan informasi kelas mayoritasnya tetap dipertahankan tidak seperti model lainnya dimana data tersebut digantikan dengan kelas minoritasnya.



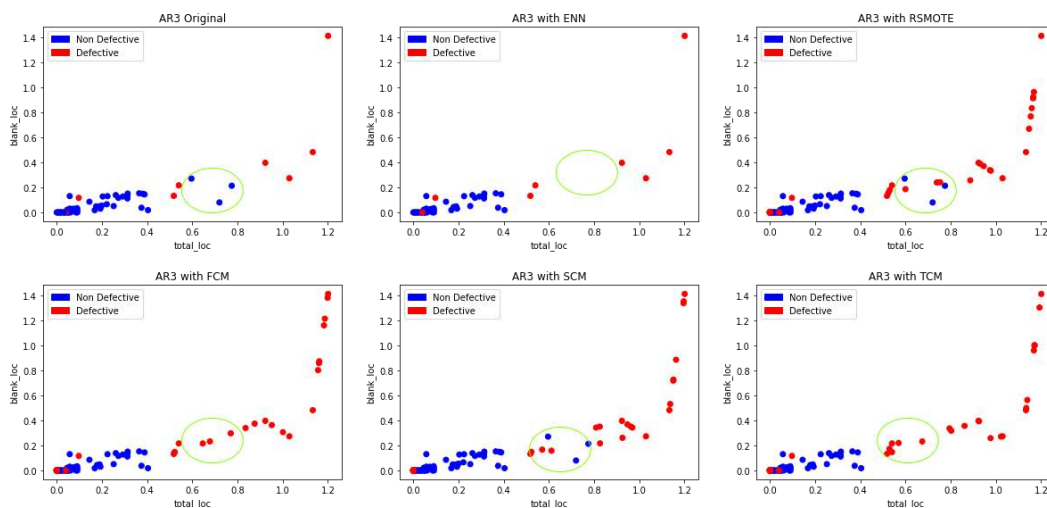
Gambar 4.2 Perbandingan data distribusi pada AR1

Pada dataset KC1 yang ditunjukkan pada Gambar 4.3 juga sama seperti dataset AR1, dimana hasil model SCM seperti hasil model RSMOTE dimana tetap mempertahankan kelas lainnya sebanyak mungkin dan menciptakan data baru disekitarnya dan berusaha untuk menghindari data yagn bersimpangan, akan tetapi dengan kombinasi metode ENN yang diimplementasikan pada model SCM

membuat data yang dihasilkan lebih rapi dan tidak terlalu mendekati dengan titik yang memiliki kelas heterogen. Sama juga dengan AR3 pada Gambar 4.4, dimana data ketika diterapkan metode ENN, TCM dan FCM pada bagian yang berlingkar hijau mengalami penghapusan sedangkan diketahui bahwa pada bagian tersebut di data originalnya ada kelas lain yang mungkin bisa menjadi data dengan informasi yang penting.



Gambar 4.3 Perbandingan data distribusi pada KC1



Gambar 4.4 Perbandingan data distribusi pada AR3



*[Halaman ini sengaja dikosongkan]*

## BAB 5

### KESIMPULAN DAN SARAN

Pada bab ini menjelaskan kesimpulan dari penelitian yang telah dilakukan dan memberikan saran untuk penelitian selanjutnya berdasarkan penelitian yang telah dilakukan.

#### 5. 1. Kesimpulan

Dari hasil penelitian yang telah dilakukan, maka dapat diambil beberapa kesimpulan dan juga untuk menjawab rumusan masalah yang telah dituliskan pada bab pertama, antara lain:

1. Dari ketiga model FCM, SCM dan TCM yang diusulkan pada penelitian ini digunakan untuk menangani permasalahan *imbalanced data*, kesembilan belas dataset yang digunakan mampu memberikan hasil evaluasi prediksi lebih baik dibandingkan dengan menggunakan data original, data hasil dari RSMOTE dan data hasil dari ENN.
2. Data model FCM ketika digunakan dalam melakukan prediksi menggunakan ANN mendapatkan nilai rata-rata f-measure, roc/auc dan akurasi sebesar **0.853, 0.849 dan 0.849**. Ketika prediksi dilakukan menggunakan CNN nilai rata-rata ketiga metric tersebut sebesar **0.854, 0.863 dan 0.866**. Untuk prediksi yang dilakukan menggunakan RNN nilai rata-rata ketiga metric tersebut juga sebesar **0.889, 0.892 dan 0.891**.
3. Data model SCM ketika digunakan dalam melakukan prediksi menggunakan ANN mendapatkan nilai rata-rata f-measure, roc/auc dan akurasi sebesar **0.816, 0.855 dan 0.860**. Ketika prediksi dilakukan menggunakan CNN nilai rata-rata ketiga metric tersebut sebesar **0.849, 0.869 dan 0.875**. Untuk prediksi yang dilakukan menggunakan RNN nilai rata-rata ketiga metric tersebut juga sebesar **0.901, 0.914 dan 0.902**.
4. Data model TCM ketika digunakan dalam melakukan prediksi menggunakan ANN mendapatkan nilai rata-rata f-measure, roc/auc dan akurasi sebesar **0.865, 0.848 dan 0.843**. Ketika prediksi dilakukan menggunakan CNN nilai rata-rata ketiga metric tersebut sebesar **0.860, 0.841 dan 0.834**. Untuk prediksi yang

dilakukan menggunakan RNN nilai rata-rata ketiga metric tersebut juga sebesar **0.911, 0.903 dan 0.902**.

5. Dibandingkan prediksi menggunakan original data, ENN dan RSMOTE. Ketiga model yang diusulkan memberikan hasil yang lebih baik ditunjukkan dari hasil evaluasi pengukuran matrik f-measure, roc/auc dan akurasi dimana terdapat selisih disetiap dataset yang digunakan.

## **5.2. Saran**

Prediksi kecacatan perangkat lunak yang dilakukan pada penelitian ini berdasarkan dengan penelitian sebelumnya yang berusaha untuk meningkatkan kualitas dari prediksi itu sendiri, salah satunya dengan menyelesaikan permasalahan ketidakseimbangan *class*. Berdasarkan hasil pengujian atau evaluasi penelitian ini perlu adanya perbaikan pada penelitian selanjutnya sehingga hasil evaluasi yang diberikan lebih stabil ketika dilakukan prediksi menggunakan *deep learning* untuk *binary classification*. Diperlukannya proyek-proyek baru yang bisa digunakan sebagai dataset sehingga tema penelitian ini lebih berkembang lagi dengan berbagai macam tantangan dan obyek penelitian yang baru.

*[Halaman ini sengaja dikosongkan]*

## DAFTAR PUSTAKA

- Ahamasaleh. (2022). [CODE] *Deep learning for intrusion detection using Recurrent Neural Network*. Github. <https://github.com/Ahamasaleh/Deep-learning-for-intrusion-detection-using-Recurrent-Neural-network-RNN>
- Alejo, R., Sotoca, J. M., Valdovinos, R. M., & Toribio, P. (2010). Edited Nearest Neighbor Rule for Improving Neural Network Classification. *International Symposium on Neural Networks*, 303–310.
- Alparslan, Y., Moyer, E. J., Isozaki, I. M., & Schwartz, D. (2021). Towards Searching Efficient and Accurate Neural Network Architectures in Binary Classification. *ArXiv:2101.06511*.
- Ambros, M. D., & Lanza, M. (2012). Evaluating Defect Prediction Approaches : A Benchmark and an Extensive Comparison. *Empir. Softw. Eng, Dcc*.
- Ambros, M. D., Lanza, M., & Robbes, R. (2010). An Extensive Comparison of Bug Prediction Approaches. *Proc. 7th Int. Working Conf. Mining Software Repositories (MSR '10)*.
- Arshad, A., Riaz, S., Jiao, L., & Murthy, A. (2018). The empirical study of semi-supervised deep fuzzy c-mean clustering for software fault prediction. *IEEE Access*, 6, 47047–47061. <https://doi.org/10.1109/ACCESS.2018.2866082>
- Beyan, C., & Fisher, R. (2015). Classifying imbalanced data sets using similarity based hierarchical decomposition. *Pattern Recognition*, 48(5), 1653–1672. <https://doi.org/10.1016/j.patcog.2014.10.032>
- Bowes, D. G. D., & Sun, N. D. Y. (2012). *Reflections on the NASA MDP data sets. February*. <https://doi.org/10.1049/iet-sen.2011.0132>
- Bunkhumpornpat, C., Sinapiromsaran, K., & Lursinsap, C. (2009). *Safe-Level-SMOTE : Safe-Level-Synthetic Minority Over-Sampling TEchnique*. 475–482.
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16(1), 321–357. <https://doi.org/10.1613/jair.953>
- Chen, B., Xia, S., Chen, Z., Wang, B., & Wang, G. (2021). RSMOTE : A self-adaptive robust SMOTE for imbalanced problems with label noise. *Information Sciences*, 553, 397–428.

<https://doi.org/10.1016/j.ins.2020.10.013>

- Desai, M., & Shah, M. (2020). An anatomization on Breast Cancer Detection and Diagnosis employing Multi-layer Perceptron Neural Network (MLP) and Convolutional Neural Network (CNN). *Clinical EHealth*. <https://doi.org/10.1016/j.ceh.2020.11.002>
- Douzas, G., Bacao, F., & Last, F. (2018). Improving imbalanced learning through a heuristic oversampling method based on k-means and SMOTE. *Information Sciences*, *465*, 1–20. <https://doi.org/10.1016/j.ins.2018.06.056>
- Du, J. (2013). *Modelling Class Noise with Symmetric and Asymmetric Distributions*. 2589–2595.
- Elmagarmid, A. K., & Member, S. (2007). *Duplicate Record Detection : A Survey*. May 2015. <https://doi.org/10.1109/TKDE.2007.250581>
- Fan, S., Liu, C., & Li, Z. (2020). *An Empirical Study on The Impact of The Interaction Between Feature Selection and Sampling in Defect Prediction*.
- Gardner, M. W., & Dorling, S. R. (1998). *ARTIFICIAL NEURAL NETWORKS ( THE MULTILAYER PERCEPTRON )— A REVIEW OF APPLICATIONS IN THE ATMOSPHERIC SCIENCES*. *32*(14), 2627–2636.
- Greco, C., Pace, P., Basagni, S., & Fortino, G. (2021). Jamming detection at the edge of drone networks using Multi-layer Perceptrons and Decision Trees. *Applied Soft Computing*, *111*, 107806. <https://doi.org/10.1016/j.asoc.2021.107806>
- Han, H., Wang, W., & Mao, B. (2005). *Borderline-SMOTE : A New Over-Sampling Method in*. 878–887.
- He, H., Bai, Y., Garcia, E. A., & Li, S. (2008). *ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning*. *3*, 1322–1328.
- He, P., Li, B., Liu, X., Chen, J., & Ma, Y. (2014). *An Empirical Study on Software Defect Prediction with a Simplified Metric Set*.
- Irmina, M., Ulfa, N., & Surendro, K. (2020). Feature selection to increase the random forest method performance on high dimensional data. *International Journal of Advances in Intelligent Informatics*, *6*(3), 303–312. <https://doi.org/https://doi.org/10.26555/ijain.v6i3.471>
- Jia, Y., Kaul, C., Lawton, T., Murray-smith, R., & Habli, I. (2021). Prediction of

- weaning from mechanical ventilation using Convolutional Neural Networks. *Artificial Intelligence In Medicine*, 117(December 2020), 102087. <https://doi.org/10.1016/j.artmed.2021.102087>
- Khadijah, Adorada, A., Wirawan, P. W., & Kurniawan, K. (2020). The Comparison of Feature Selection Methods in Software Defect Prediction. *ICICoS 2020 - Proceeding: 4th International Conference on Informatics and Computational Sciences*, 0–5. <https://doi.org/10.1109/ICICoS51170.2020.9299022>
- Koziarski, M. (2021). CSMOUTE: Combined Synthetic Oversampling and Undersampling Technique for Imbalanced Data Classification. *Computing 103*. <https://doi.org/https://doi.org/10.1007/s00607-020-00854-1>
- Kuhn, M., & Johnson, K. (2013). *Applied Predictive Modeling with Applications in R*. [http://appliedpredictivemodeling.com/s/Applied\\_Predictive\\_Modeling\\_in\\_R.pdf](http://appliedpredictivemodeling.com/s/Applied_Predictive_Modeling_in_R.pdf)
- Kuncheva, L. I., Youse, P., & Almeida, J. (2018). Edited nearest neighbour for selecting keyframe summaries of egocentric videos. *Journal of Visual Communication and Image Representation*, 52(January), 118–130. <https://doi.org/10.1016/j.jvcir.2018.02.010>
- Le, T. D. B., Linares-Vásquez, M., Lo, D., & Poshyvanyk, D. (2015). RCLinker: Automated Linking of Issue Reports and Commits Leveraging Rich Contextual Information. *IEEE International Conference on Program Comprehension, 2015-Augus*, 36–47. <https://doi.org/10.1109/ICPC.2015.13>
- Leslie, H., Spits, H., Gaol, F., & Abdurachman, E. (2017). *Software Metrics for Fault Prediction Using Machine Learning Approaches*. November. <https://doi.org/10.1109/CYBERNETICSCOM.2017.8311708>
- Li, Jian, He, P., Zhu, J., & Lyu, M. R. (2017). Software defect prediction via convolutional neural network. *Proceedings - 2017 IEEE International Conference on Software Quality, Reliability and Security, QRS 2017*, 318–328. <https://doi.org/10.1109/QRS.2017.42>
- Li, Junnan, Zhu, Q., Wu, Q., & Fan, Z. (2021). A novel oversampling technique for class-imbalanced learning based on SMOTE and natural neighbors. *Information Sciences*, 565, 438–455.

<https://doi.org/10.1016/j.ins.2021.03.041>

- Liu, H., & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining* (1st ed.). Springer US. <https://doi.org/10.1007/978-1-4615-5689-3>
- Mahmood, Z., Bowes, D., & Lane, P. C. R. (2015). *What is the Impact of Imbalance on Software Defect Prediction Performance ?*
- Malhotra, R. (2015). A systematic review of machine learning techniques for software fault prediction. *Applied Soft Computing Journal*, 27, 504–518. <https://doi.org/10.1016/j.asoc.2014.11.023>
- Malhotra, R., Agrawal, V., Pal, V., & Agarwal, T. (2021). *Support Vector Based Oversampling Technique for Handling Class Imbalance in Software Defect Prediction*. 1078–1083.
- Malhotra, R., Bahl, L., Sehgal, S., & Priya, P. (2017). *Empirical Comparison of Machine Learning Algorithms for Bug Predictions in Open Source Software*. 40–45.
- Malhotra, R., Nishant, Gurha, S., & Rathi, V. (2021). Application of Praticle Swarm Optimization for Software Defect Prediction Using Object Oriented Metrics. *International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, 88–93.
- Massoudi, M., Jain, N. K., & Bansal, P. (2021). *Software Defect Prediction using Dimensionality Reduction and Deep Learning*. *Icicv*, 884–893.
- Mausa, G., Grbac, T. G., & Basic, B. D. (2011). Software defect prediction with Bug-Code analyzer-A data collection tool demo. *2014 22nd International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2014*, 425–426. <https://doi.org/10.1109/SOFTCOM.2014.7039122>
- Miao, Z., & Zhao, L. (2011). Multi-class Imbalanced Learning Implemented in Network Intrusion Detection. *International Conference on Computer Science and Service System (CSSS)*, 1395–1398.
- Mishra, M., & Srivastava, M. (2014). A View of Artificial neural Network. *IEEE International Conference on Advances in Engineering & Technology Research (ICAETR - 2014)*, 7–9.
- Mishra, S. (2017). *Handling Imbalanced Data : SMOTE vs . Random*



*Undersampling*. 317–320.

- Padmaja, T. M., Dhulipalla, N., Bapi, R. S., & Krishna, P. R. (2007). *Unbalanced Data Classification Using extreme outlier Elimination and Sampling Techniques for Fraud Detection*. 511–516. <https://doi.org/10.1109/ADCOM.2007.74>
- Park, S., & Park, H. (2020). Combined oversampling and undersampling method based on slow-start algorithm for imbalanced network traffic. *Computing*. <https://doi.org/10.1007/s00607-020-00854-1>
- Perera, A. (2020). Using Defect Prediction to Improve the Bug Detection Capability of Search-Based Software Testing. *Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020*, 1170–1174. <https://doi.org/10.1145/3324884.3415286>
- Pralienka, F., Muhamad, B., Siahaan, D. O., & Fatichah, C. (2017). *Perbaikan Prediksi Kesalahan Perangkat Lunak Menggunakan Seleksi Fitur dan Cluster-Based Classification*. 6(3).
- Putri, S. A., & Wahono, R. S. (2015). Integrasi SMOTE dan Information Gain pada Naive Bayes untuk Prediksi Cacat Software. *Journal of Software Engineering*, 1(2), 86–91.
- Qiao, L., Li, X., Umer, Q., & Guo, P. (2020). Deep learning based software defect prediction. *Neurocomputing*, 385, 100–110. <https://doi.org/10.1016/j.neucom.2019.11.067>
- Santana, L. M. Q. De, Santos, R. M., Matos, L. N., & Macedo, H. T. (2018). Deep Neural Networks for Acoustic Modeling in the Presence of Noise. *IEEE Latin America Transactions*, 16(3), 918–925. <https://doi.org/10.1109/TLA.2018.8358674>
- Seliya, N., & Khoshgoftaar, T. M. (2011). *The use of decision trees for cost-sensitive classification : an empirical study in software quality prediction*. 1(October), 448–459. <https://doi.org/10.1002/widm.38>
- Shamsudin, H., Yusof, U. K., Jayalakshmi, A., Nor, M., & Khalid, A. (2020). Combining oversampling and undersampling techniques for imbalanced classification : A comparative study using credit card fraudulent transaction dataset. *IEEE, 2020*. <https://doi.org/10.1109/ICCA51439.2020.9264517>

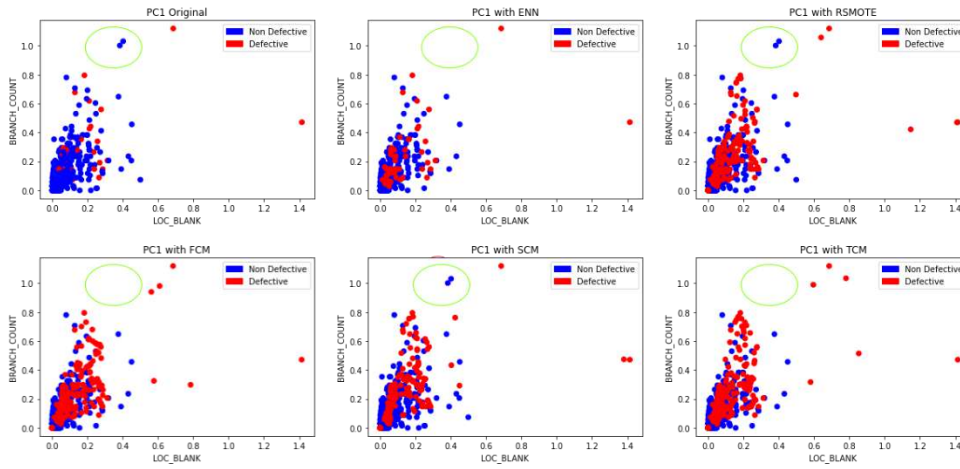
- Singh, R. (2020). *Transfer Learning Code Vectorizer based Machine Learning Models for Software Defect Prediction*. *Lm*, 497–502.
- Sunder, K. S., Reddy, Y. C. A. P., & Mallikarjuna, C. (2020). Materials Today : Proceedings Recurrent neural network based prediction of number of COVID-19 cases in India. *Materials Today: Proceedings*, *xxxx*, 4–7. <https://doi.org/10.1016/j.matpr.2020.11.117>
- Tahir, M. A., Kittler, J., Mikolajczyk, K., & Yan, F. (2009). A multiple expert approach to the class imbalance problem using inverse random under sampling. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5519 LNCS, 82–91. [https://doi.org/10.1007/978-3-642-02326-2\\_9](https://doi.org/10.1007/978-3-642-02326-2_9)
- Thapa, S., Alsadoon, A., Prasad, P. W. C., Al-Dala'In, T., & Rashid, T. A. (2020). Software defect prediction using atomic rule mining and random forest. *CITISIA 2020 - IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, Proceedings*. <https://doi.org/10.1109/CITISIA50690.2020.9371797>
- Wahono, R. S., & Suryana, N. (2013). Combining particle swarm optimization based feature selection and bagging technique for software defect prediction. *International Journal of Software Engineering and Its Applications*, 7(5), 153–166. <https://doi.org/10.14257/ijseia.2013.7.5.16>
- Xia, S., Xiong, Z., Luo, Y., Dong, L., & Xing, C. (2015). Relative density based support vector machine. *Neurocomputing*, 149, 1424–1432. <https://doi.org/https://doi.org/10.1016/j.neucom.2014.08.053>
- Yadav, H. S. (2020). Increasing Accuracy of Software Defect Prediction using 1-dimensional CNN with SVM. *2020 IEEE International Conference for Innovation in Technology, INOCON 2020*, 1–6. <https://doi.org/10.1109/INOCON50539.2020.9298189>
- Yousef, A. H. (2015). Extracting software static defect models using data mining. *Ain Shams Engineering Journal*, 6(1), 133–144. <https://doi.org/10.1016/j.asej.2014.09.007>
- Zhang, Y., Gao, J., & Zhou, H. (2020). Breeds Classification with Deep Convolutional Neural Network. *ACM International Conference Proceeding*

*Series*, 145–151. <https://doi.org/10.1145/3383972.3383975>

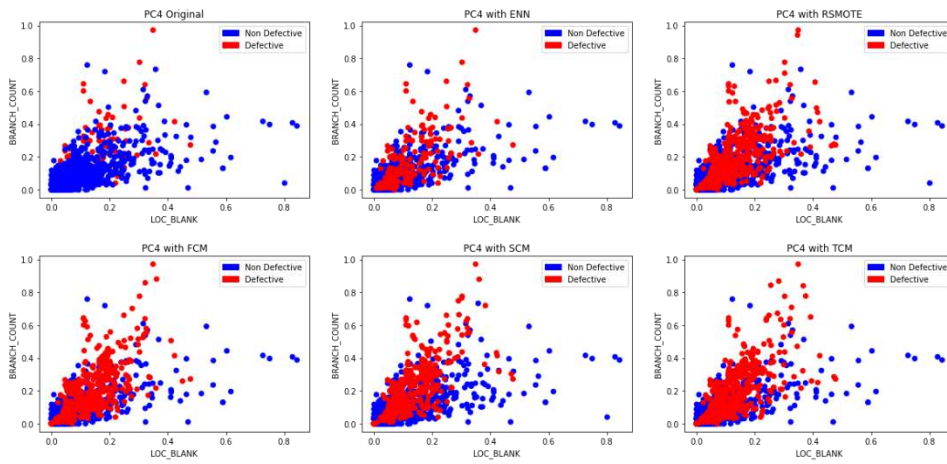
Zhu, Y., Jia, C., Li, F., & Song, J. (2020). Inspector : a lysine succinylation predictor based on edited nearest-neighbor undersampling and adaptive synthetic oversampling ☆. *Analytical Biochemistry*, 593(January), 113592. <https://doi.org/10.1016/j.ab.2020.113592>

Zi, M. (2014). *Service-Oriented Medical System for Supporting Decisions With Missing and Imbalanced Data*. 18(5), 1533–1540.

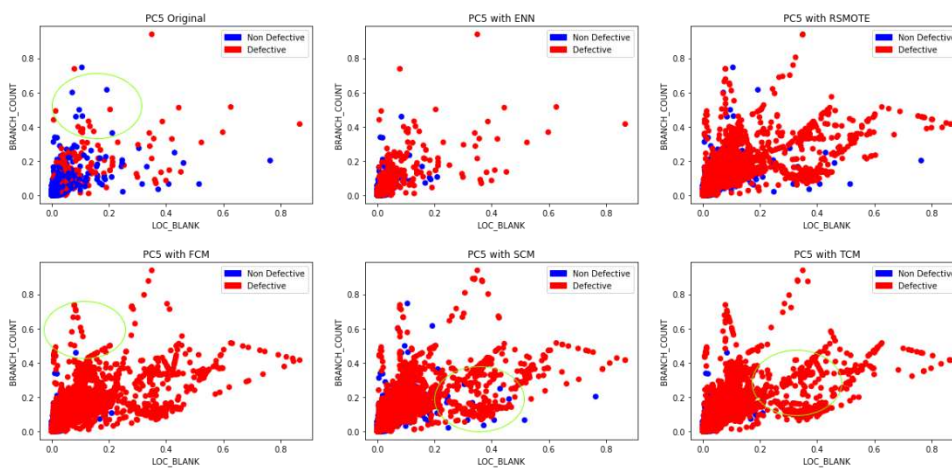
## LAMPIRAN A



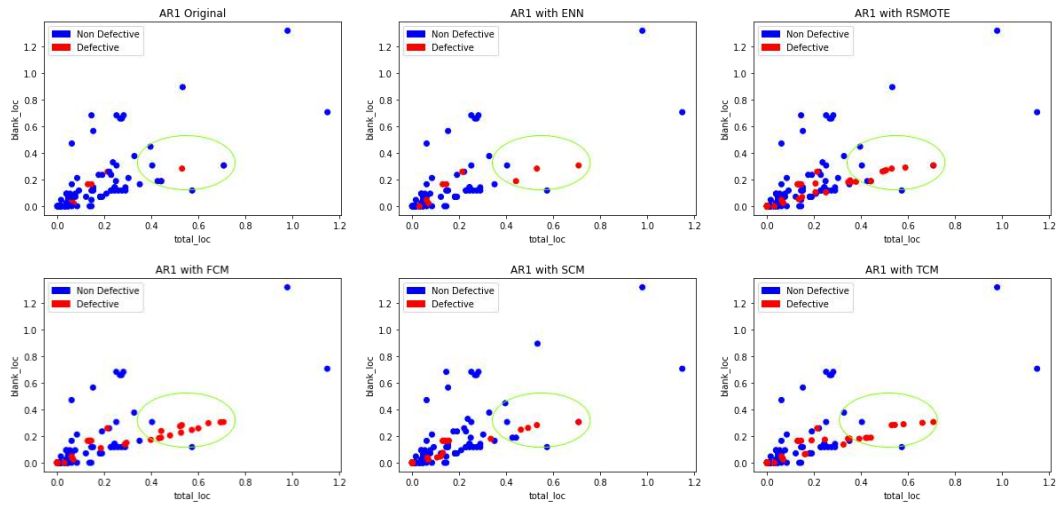
Lampiran A 1 Perbandingan Distribusi data PC1



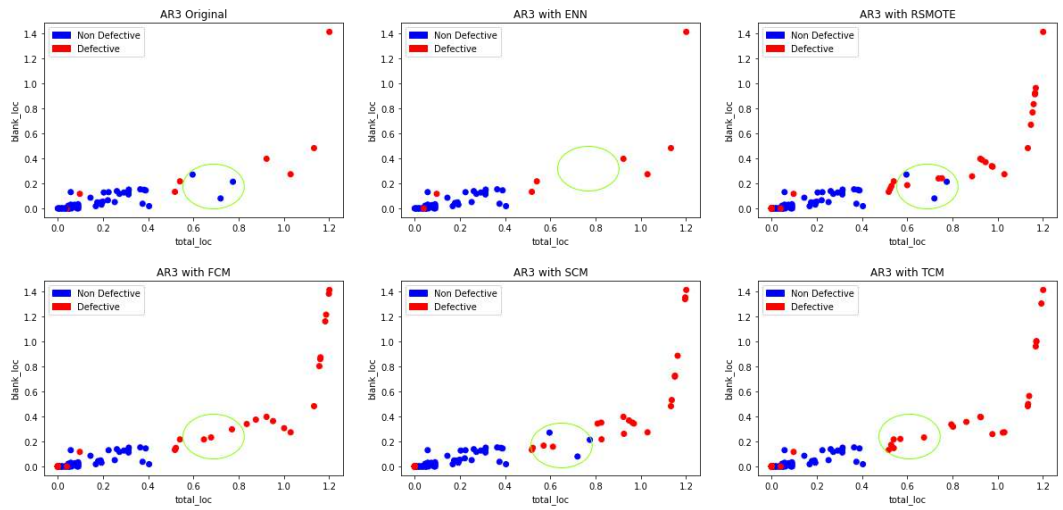
Lampiran A 2 Perbandingan Distribusi data PC4



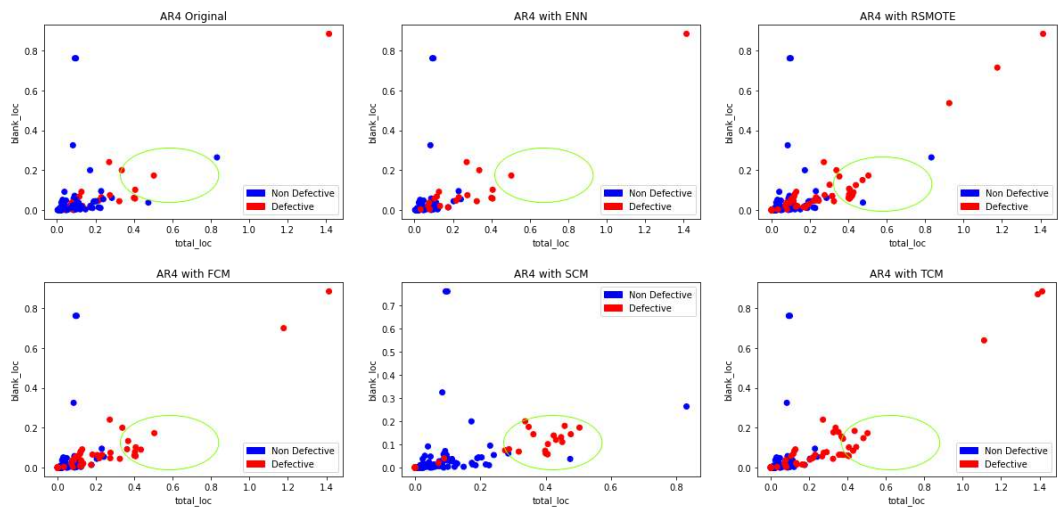
Lampiran A 3 Perbandingan Distribusi data PC5



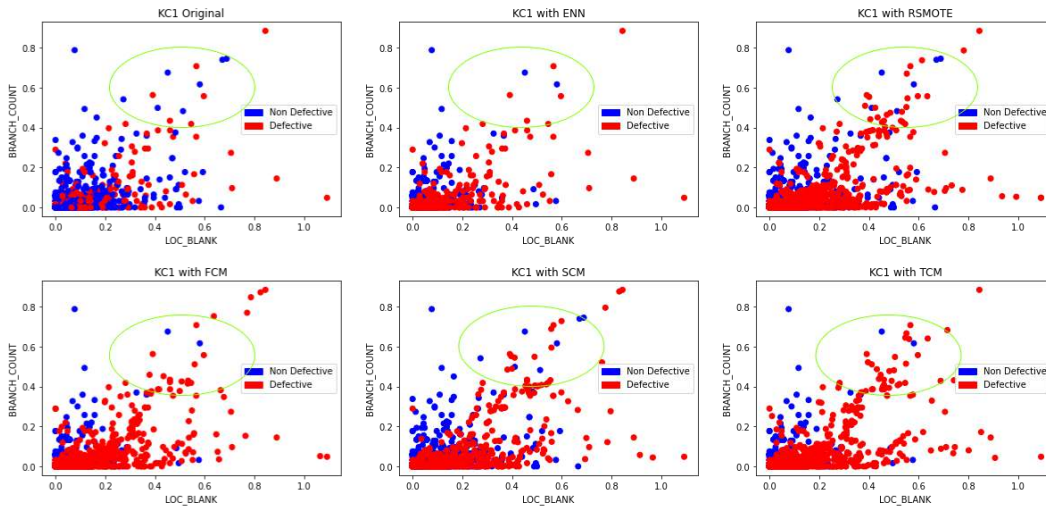
Lampiran A 4 Perbandingan Distribusi data AR1



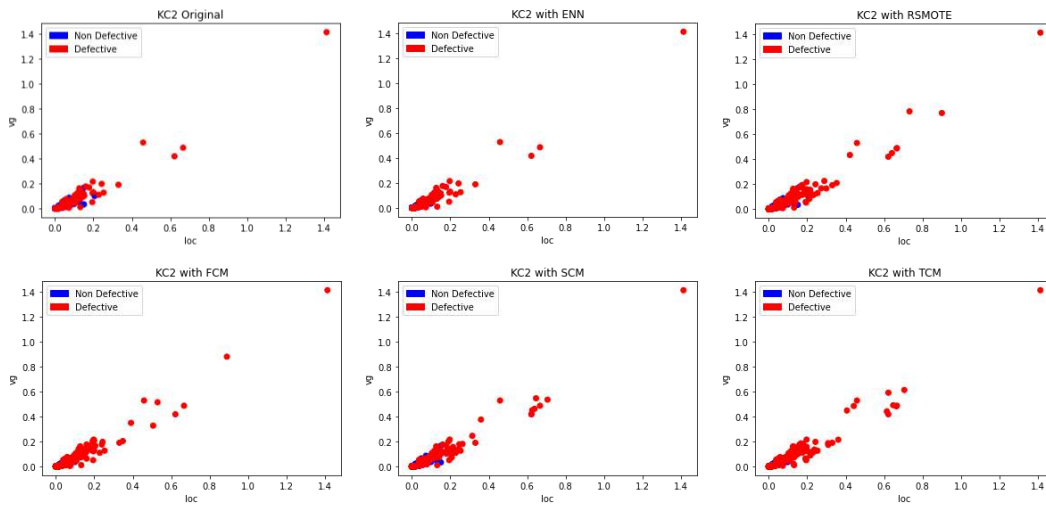
Lampiran A 5 Perbandingan Distribusi data AR3



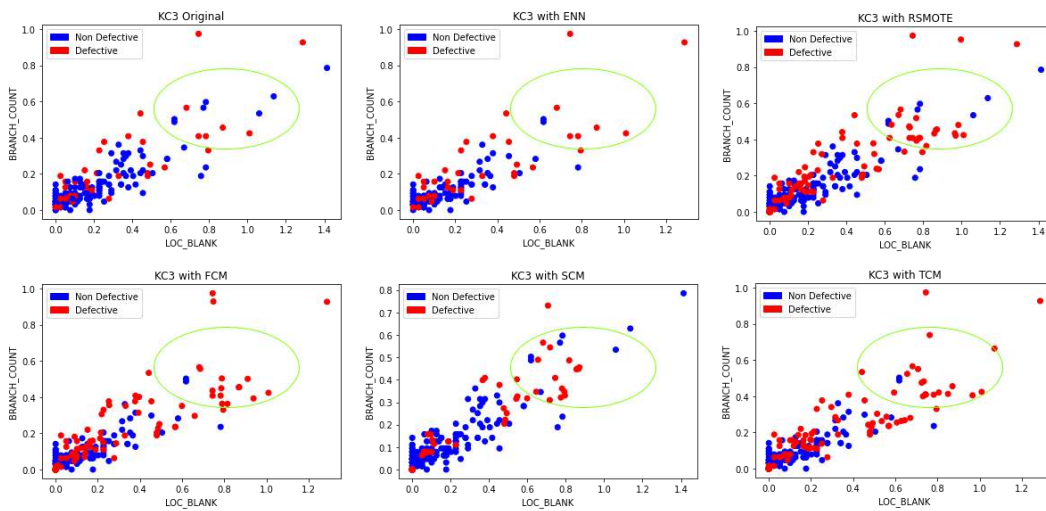
Lampiran A 6 Perbandingan Distribusi data AR4



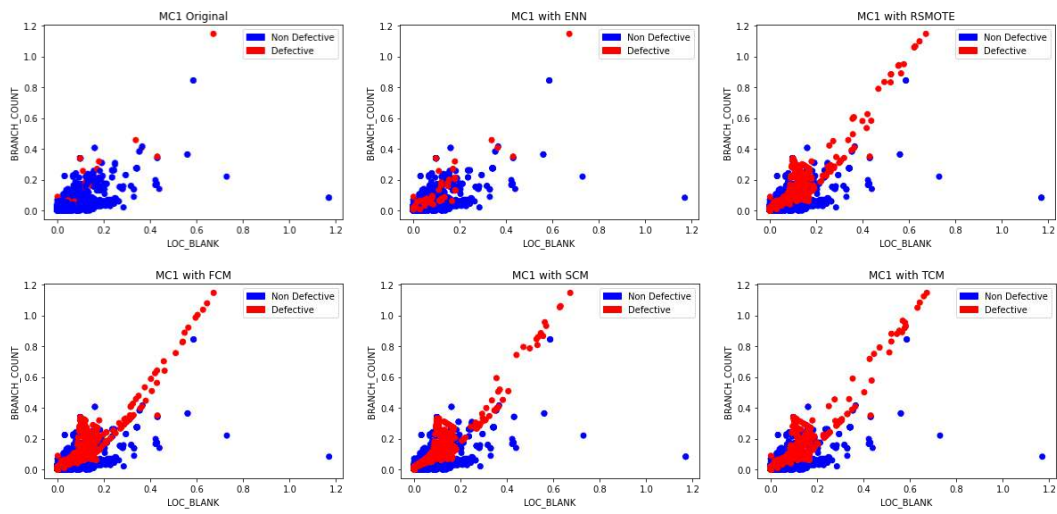
Lampiran A 7 Perbandingan Distribusi data KC1



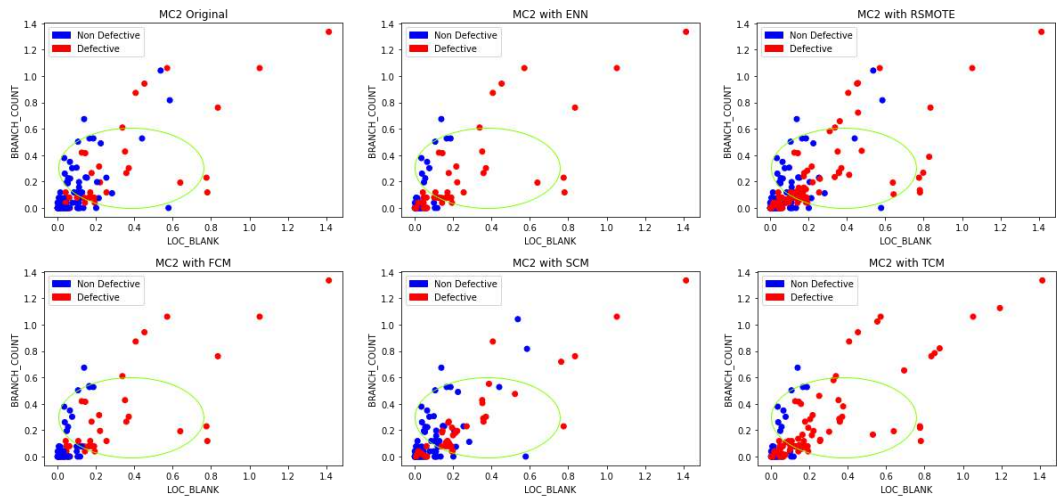
Lampiran A 8 Perbandingan Distribusi data KC2



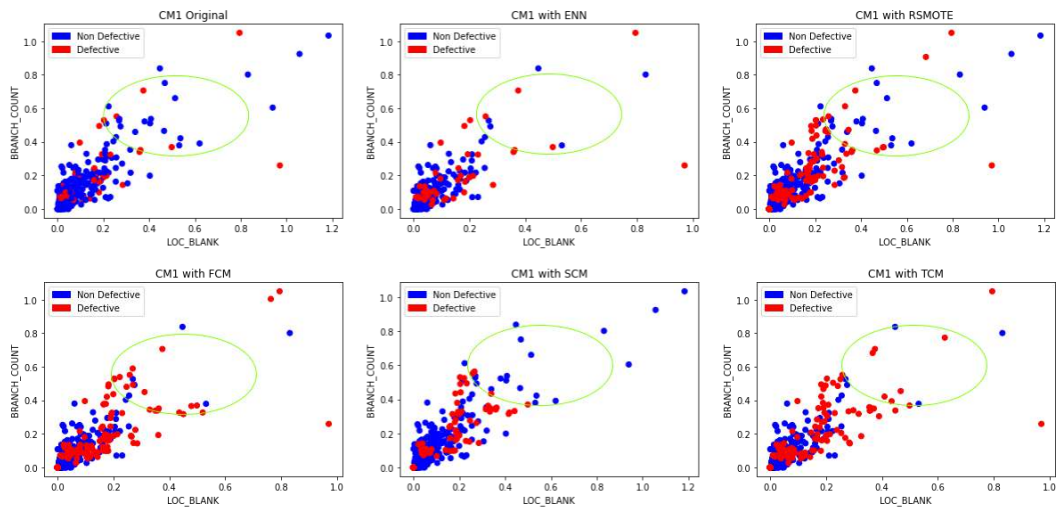
Lampiran A 9 Perbandingan Distribusi data KC3



Lampiran A 10 Perbandingan Distribusi data MC1

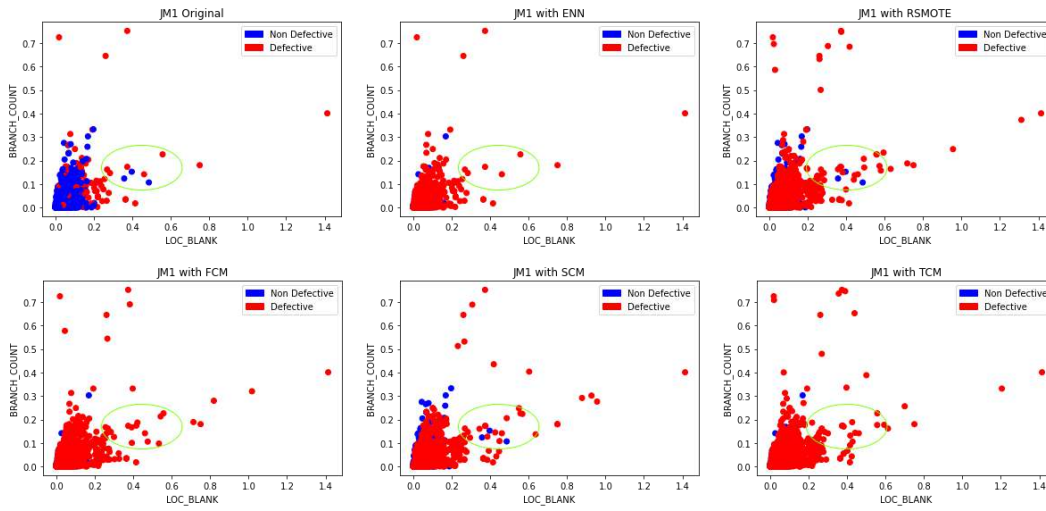


Lampiran A 11 Perbandingan Distribusi data MC2

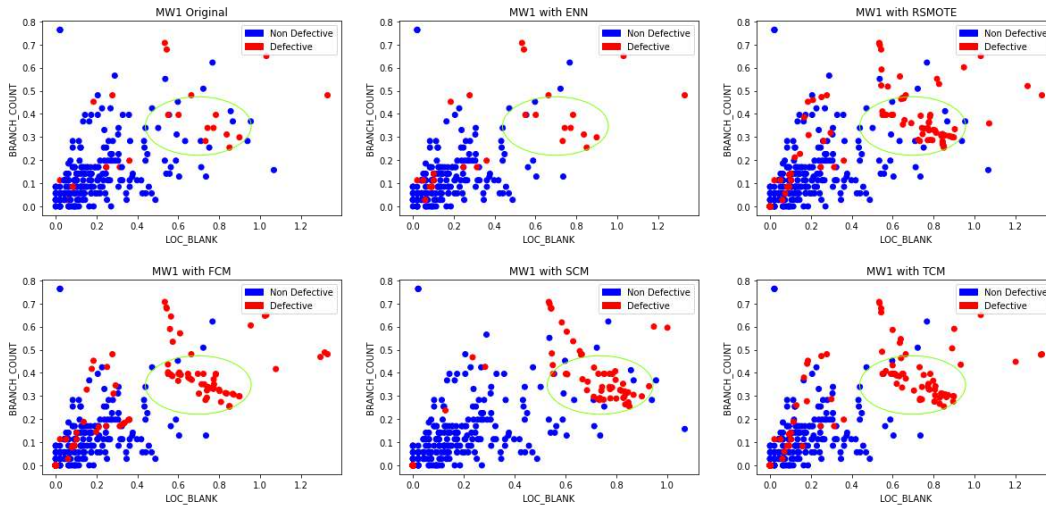


Lampiran A 12 Perbandingan Distribusi data CM1

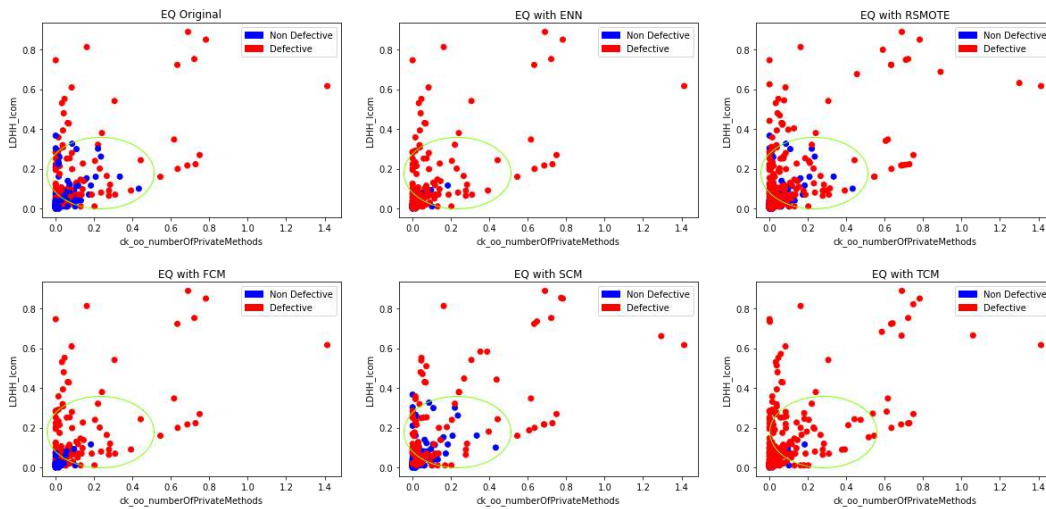




Lampiran A 13 Perbandingan Distribusi data JM1

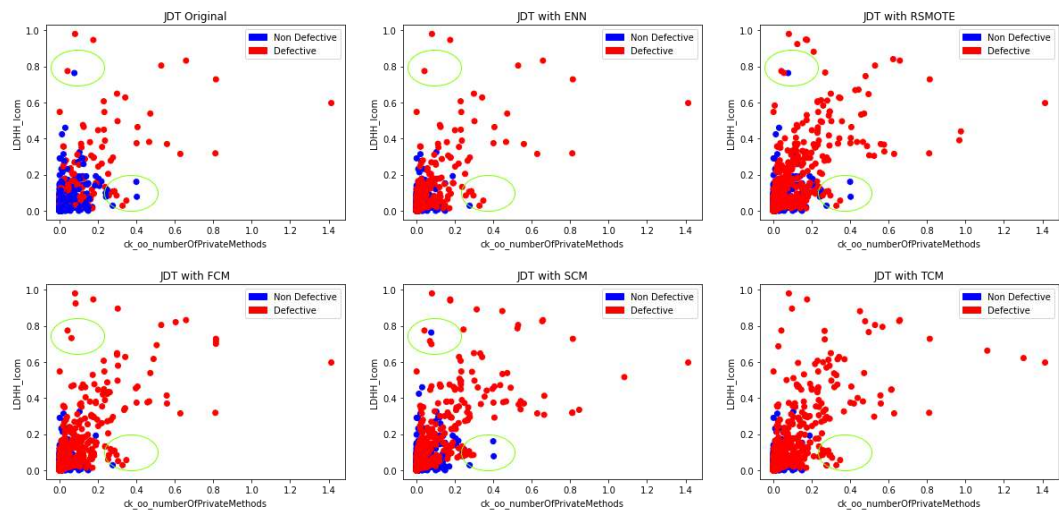


Lampiran A 14 Perbandingan Distribusi data MW1

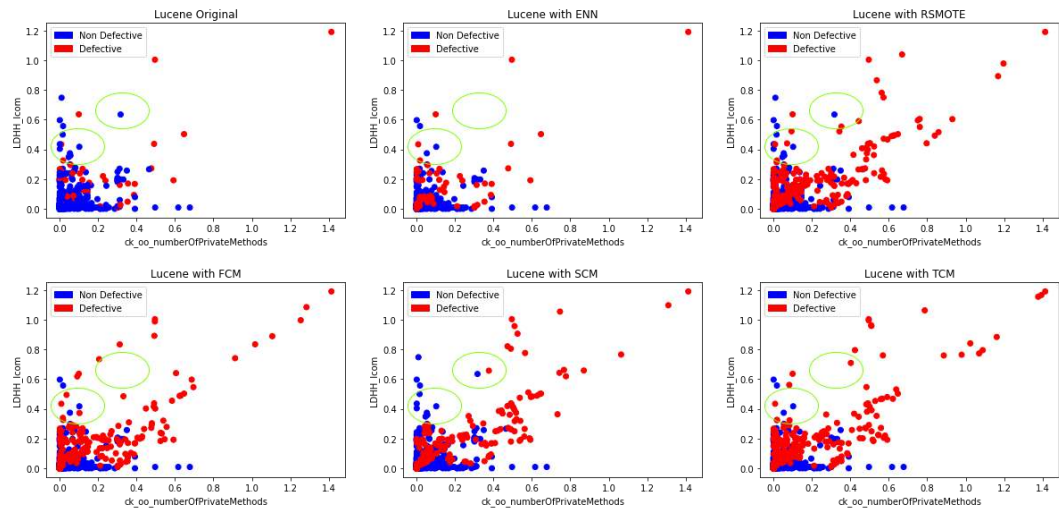


Lampiran A 15 Perbandingan Distribusi data EQ

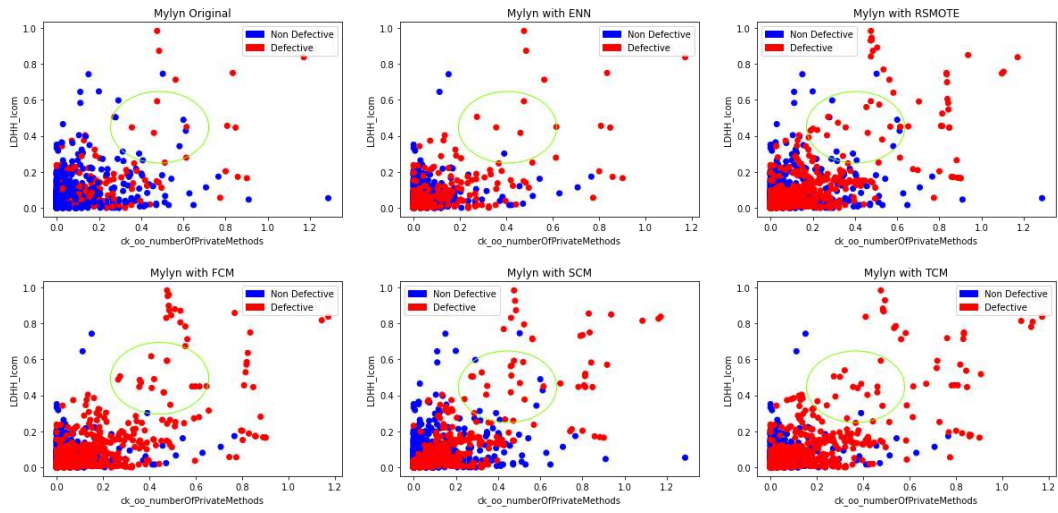




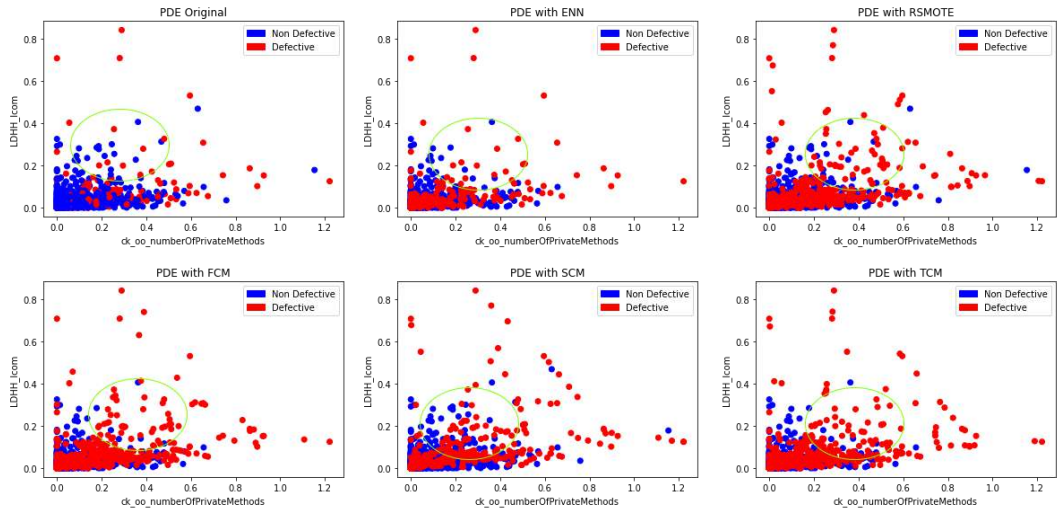
Lampiran A 16 Perbandingan Distribusi data JDT



Lampiran A 17 Perbandingan Distribusi data Lucene



Lampiran A 18 Perbandingan Distribusi data Mylyn



Lampiran A 19 Perbandingan Distribusi data PDE

## **BIODATA PENULIS**



Aizul Faiz Iswafaza, kelahiran 25 Maret 1997 di Malang. Penulis merupakan anak kedua dari dua bersaudara. Pendidikan sekolah dasar ditempuh di Madrasah Ibtidaiyah Negeri 1 Malang lalu dilanjutkan pendidikan tingkat menengah pada Madrasah Tsanawiyah Negeri 1 Malang dan pendidikan tingkat atas pada Madrasah Aliyah Negeri 3 Malang. Kemudian penulis menempuh pendidikan sarjana di Universitas Brawijaya Malang pada

Fakultas Ilmu Komputer dengan Program Studi Teknik Informatika pada tahun 2015 hingga lulus pada tahun 2019. Pendidikan magister ditempuh di Pascasarjana Departemen Teknik Informatika, Fakultas Teknik Elektro dan Informatika Cerdas, Institut Teknologi Sepuluh Nopember, Surabaya.

Saat menempuh pendidikan magister Teknik Informatika penulis memperoleh banyak ilmu dan wawasan penelitian dari para dosen maupun rekan – rekan sesama peneliti. Setelah menempuh S2, penulis akan melanjutkan karir pada bidang Informatika yang telah ditekuni sejak lama. Jika ingin menghubungi penulis dapat melalui email [aizuliswafaza@gmail.com](mailto:aizuliswafaza@gmail.com)