



**MODIFIKASI ALGORITMA *MULTI-INSTANCE EXTREME
LEARNING MACHINE* MENGGUNAKAN ALGORITMA
K-MEANS UNTUK MENGATASI MASALAH BESARAN
DIMENSI PADA *HIDDEN LAYER* KE DUA**

SKRIPSI

RACHMAT PRABOWO

1210511039

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAKARTA
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
2016**



**MODIFIKASI ALGORITMA *MULTI-INSTANCE EXTREME
LEARNING MACHINE* MENGGUNAKAN ALGORITMA
K-MEANS UNTUK MENGATASI MASALAH BESARAN
DIMENSI PADA *HIDDEN LAYER* KE DUA**

SKRIPSI

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Komputer**

RACHMAT PRABOWO

1210511039

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” JAKARTA
FAKULTAS ILMU KOMPUTER
PROGRAM STUDI TEKNIK INFORMATIKA
2016**

PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya sendiri, dan semua sumber yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.

Nama : Rachmat Prabowo

NRP : 1210511039

Tanggal : 5 Agustus 2016

Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan saya ini, maka saya bersedia dituntut dan diproses sesuai dengan ketentuan yang berlaku.

Jakarta, 5 Agustus 2016

Yang Menyatakan,



(Rachmat Prabowo)

PERNYATAAN PERSETUJUAN PUBLIKASI SKRIPSI UNTUK KEPENTINGAN AKADEMIS

Sebagai civitas akademik Universitas Pembangunan Nasional “Veteran” Jakarta, saya yang bertanda tangan di bawah ini :

Nama : Rachmat Prabowo
NRP : 1210511039
Fakultas : Ilmu Komputer
Program Studi : S1 Teknik Informatika

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Pembangunan Nasional “Veteran” Jakarta Hak Bebas Royalti Non eksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul:

**“Modifikasi Algoritma *Multi-Instance Extreme Learning Machine*
Menggunakan Algoritma *K-Means* Untuk Mengatasi Masalah Besaran
Dimensi Pada *Hidden Layer* Ke Dua”**

Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti ini Universitas Pembangunan Nasional “Veteran” Jakarta berhak menyimpan, mengalih media/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan Skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Jakarta
Pada tanggal : 5 Agustus 2016
Yang menyatakan,



(Rachmat Prabowo)

PENGESAHAN

Skripsi diajukan oleh:

Nama : Rachmat Prabowo
NRP : 1210511039
Program Studi : S1 - Teknik Informatika
Judul Skripsi : Modifikasi Algoritma *Multi-Instance Extreme Learning Machine* Menggunakan Algoritma *K-Means* Untuk Mengatasi Masalah Besaran Dimensi Pada *Hidden Layer* Ke Dua

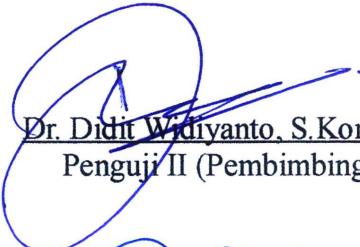
Telah berhasil dipertahankan di hadapan Tim Pengujian dan diterima sebagai bagian persyaratan untuk memperoleh gelar Sarjana Komputer pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Pembangunan Nasional "Veteran" Jakarta.



Dr. Nidjo Sandjojo, M.Sc.
Ketua Pengujian


Bambang Triwahyono, S.Kom., M.Si.
Pengujii I


Dr. Nidjo Sandjojo, M.Sc.
Dekan



Dr. Didit Widiyanto, S.Kom., M.Si.
Pengujii II (Pembimbing)



Vini Indriasari, S.T., M.Sc., Ph.D.
Ka. Prodi

Ditetapkan di : Jakarta
Tanggal Ujian : 27 Juli 2016

MODIFIKASI ALGORITMA *MULTI-INSTANCE EXTREME LEARNING MACHINE* MENGGUNAKAN ALGORITMA *K-MEANS* UNTUK MENGATASI MASALAH BESARAN DIMENSI PADA *HIDDEN LAYER* KE DUA

Rachmat Prabowo

Abstrak

Multiple Instance Learning (MIL) merupakan solusi dari permasalahan klasifikasi *supervised learning*, di mana objek yang diklasifikasikan memiliki beragam bentuk yang berbeda dan menyebabkan keambiguan pada saat pembelajaran. *Multi-Instance Extreme Learning Machine* (MI-ELM) merupakan salah satu metode yang diusulkan untuk penyelesaian permasalahan *Multiple-Instance Classification* (MIC). Permasalahannya, penggunaan fungsi *gaussian similarity* pada proses transofrmasi *feature* pada MI-ELM menyebabkan jumlah *hidden node* pada *hidden layer* ke dua menjadi sebanyak jumlah *bag* yang dipelajari. Hal ini membuat MI-ELM menjadi tidak efisien saat melakukan pembelajaran dengan jumlah *bag* yang besar. Metode yang diusulkan dalam penelitian ini adalah menerapkan algoritma *clustering k-means* ke dalam MI-ELM, yang disebut *K-Means* MI-ELM. Hal ini bertujuan agar jumlah *hidden node* pada *hidden layer* ke dua dapat ditetapkan. Pada penelitian ini percobaan dilakukan dengan menggunakan tiga *data set* klasifikasi citra, TIGER, FOX, dan ELEPHANT. Untuk melihat efesiensi dari metode usulan, dilakukan perbandingan dengan MI-ELM dengan melakukan pembelajaran menggunakan jumlah *bag* yang berbeda-beda. Hasil dari percobaan yang dilakukan memperlihatkan, nilai efisiensi dan akurasi metode usulan *K-Means* MI-ELM lebih baik dibandingkan MI-ELM.

Kata Kunci : *Multiple-Instance Learning, Multiple-Instance Classification, Extreme Learning Machine, Multi-Instance Extreme Learning Machine, K-Means*

MODIFIKASI ALGORITMA *MULTI-INSTANCE EXTREME LEARNING MACHINE* MENGGUNAKAN ALGORITMA *K-MEANS* UNTUK MENGATASI MASALAH BESARAN DIMENSI PADA *HIDDEN LAYER* KE DUA

Rachmat Prabowo

Abstract

Multiple Instance Learning (MIL) is the solution for supervised learning classification problem, where the object which being classification target contains several featured section and make some ambiguous in the learning process. *Multi-Instance Extreme Learning Machine* (MI-ELM) is one of some methods that being proposed for resolve the Multiple-Instance Classification (MIC) problem. The problem is, the utilization function of gaussian similarity in the transformation feature process on MI-ELM cause the number of the hidden node in second hidden layer become as many as number of the bag which being learned. This case makes the MI-ELM did not efficient for learning with a big number of bag. The method that proposed for this research is apply the clustering k-means algorithm into MI-ELM, called K-Means MI-ELM. This case intend to determine the number of the hidden node in second hidden layer. In this research the experimentation being trial used three data set classification image as, TIGER, FOX, and ELEPHANT. For obtain the efficiency of the proposed method, make a comparison using MI-ELM with takes some leaning of several number of the bag. The results of the research showed that the efficiency and accuracy of proposed method K-Means MI-ELM was better then MI-ELM.

Keywords : *Multiple-Instance Learning, Multiple-Instance Classification, Extreme Learning Machine, Multi-Instance Extreme Learning Machine, K-Means*

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah SWT atas segala karuniannya sehingga penelitian ini berhasil diselesaikan. Penelitian ini dilaksanakan sejak bulan Januari 2016 hingga Juli 2016 dengan judul **Modifikasi Algoritma Multi-Instance Extreme Learning Machine Menggunakan Algoritma K-Means Untuk Mengatasi Masalah Besaran Dimensi Pada Hidden Layer Ke Dua**

Dalam penulisan skripsi ini tentunya tidak lepas dari bantuan pembimbing, petunjuk dan dorongan baik secara materiil maupun secara moril dari berbagai pihak. Dalam kesempatan ini, penulis bermaksud menyampaikan rasa terima kasih kepada :

1. Bapak Dr. Didit Widiyanto, S.Kom., M.Si. selaku Dosen Pembimbing yang telah memberikan banyak bimbingan dan saran yang sangat bermanfaat.
2. Ibu Vini Indriasari, S.T., M.Sc., Ph.D. selaku Kepala Jurusan S1 Teknik Informatika.
3. Bapak Dr. Nidjo Sandjojo, M.Sc. selaku Dekan Fakultas Ilmu Komputer Universitas Pembangunan Nasional “Veteran” Jakarta.
4. Kedua orang tua yang telah memberikan dukungan, semangat, dan yang tentunya tidak pernah berhenti mendoakan kesuksesan dan kelancaran dalam penyusunan skripsi ini.
5. Ibu Yayuk Purwadi selaku Ketua Yayasan Kencana Lestari yang telah memberikan penulis kesempatan untuk mendapatkan bewasiswa dalam mengampu pendidikan di Universitas Pembangunan Nasional “Veteran” Jakarta dari awal hingga lulus.
6. Zaid Arif D., ‘Ainurrofiqurohman, Rendi Prasetyo, Witandari P. dan teman-teman seperjuangan Program Studi S1 Teknik Informatika angkatan 2012 yang tentunya tidak dapat disebutkan satu persatu.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna, untuk itu penulis mengharapkan saran dan kritik yang bersifat membangun dalam menyempurnakan penelitian ini. Akhir kata dari penulis, semoga skripsi ini dapat bermanfaat bagi para pembaca khususnya mahasiswa FIK UPN "Veteran" Jakarta.

Jakarta, 5 Agustus 2016

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
PERNYATAAN ORISINALITAS.....	ii
PERNYATAAN PERSETUJUAN PUBLIKASI.....	iii
PENGESAHAN.....	iv
ABSTRAK.....	v
ABSTRACT.....	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR TABEL.....	xi
DAFTAR GAMBAR	xii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
I.1 Latar Belakang.....	1
I.2 Rumusan Masalah.....	2
I.3 Ruang Lingkup Penelitian.....	2
I.4 Tujuan Penelitian.....	3
I.5 Luaran dan Manfaat Penelitian.....	3
I.6 Sistematika Penulisan.....	3
BAB II STUDI LITERATUR.....	5
II.1 Penelitian Terdahulu.....	5
II.1.1 Guang-Bin, H, <i>et. al.</i> (2006).....	5
II.1.2 Qiang, L, <i>et. al.</i> (2016).....	9
II.1.3 Qing, H, <i>et. al.</i> (2014).....	15
II.2 Landasan Teori.....	16
II.2.1 <i>Multiple-Instance Learning</i>	16
II.2.2 Algoritma <i>K-means</i>	19
II.2.3 <i>K-Fold Cross Validation</i>	19
BAB III METODOLOGI PENELITIAN	22
III.1 Pengambilan Data	23
III.2 Pra Proses.....	23
III.3 Membentuk Kelompok 10 Fold Cross Validation.....	24
III.4 Desain Metode Usulan.....	25
III.5 Evaluasi Performa.....	28
BAB IV HASIL DAN PEMBAHASAN.....	29
IV.1 Pra Proses.....	29
IV.1.1 Seleksi <i>Feature</i>	29

IV.1.2	Normalisasi.....	29
IV.2	Evaluasi Performa.....	29
IV.2.1	Evaluasi Akurasi.....	30
IV.2.2	Evaluasi <i>Learning Time</i>	39
BAB V	PENUTUP	48
V.1	Kesimpulan.....	48
V.2	Saran	49
DAFTAR PUSTAKA.....		50
RIWAYAT HIDUP		
LAMPIRAN		

DAFTAR TABEL

Tabell1 Ringkasan Informasi *Data set* yang digunakan22

DAFTAR GAMBAR

Gambar 1 Arsitektur SLFNs Extreme Learning Machine.....	7
Gambar 2 Framework of Supervised MI Classification.....	10
Gambar 3 Alur Kerja Dari MI-ELM.....	11
Gambar 4 Contoh Pembentukan Instance dengan SBN.....	17
Gambar 5 Contoh Pembentukan Bag dari Sebuah Citra.....	18
Gambar 6 Ilustrasi Pembagian Data menjadi Beberapa Kelompok.....	20
Gambar 7 Ilustrasi Rotasi Training Data Set dengan 4-Fold Cross	21
Gambar 8 Metodologi Penelitian.....	22
Gambar 9 Contoh Pembagian Bag Menjadi 10 Fold Cross Validation.....	24
Gambar 10 Alur Kerja K-Means MI-ELM.....	27
Gambar 11 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set TIGER.....	30
Gambar 12 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set TIGER.....	31
Gambar 13 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set TIGER.....	31
Gambar 14 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set TIGER.....	32
Gambar 15 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set FOX.....	32
Gambar 16 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set FOX	33
Gambar 17 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set FOX.....	34
Gambar 18 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set FOX.....	34
Gambar 19 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set ELEPHANT.....	35
Gambar 20 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 140 pada data set ELEPHANT.....	36
Gambar 21 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set ELEPHANT.....	36
Gambar 22 Grafik Perbandingan Akurasi K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 200 pada data set ELEPHANT.....	37
Gambar 23 Grafik Perbandingan Rata-Rata akurasi K-Means MI-ELM Dan MI- ELM Pada Data Set TIGER.....	37
Gambar 24 Grafik Perbandingan Rata-Rata Akurasi K-Means MI-ELM dan MI- ELM Pada Data Set FOX.....	38
Gambar 25 Grafik Perbandingan Rata-Rata Akurasi K-Means MI-ELM dan MI- ELM Pada Data Set ELEPHANT.....	38
Gambar 26 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set TIGER.....	39

Gambar 27 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set TIGER.....	40
Gambar 28 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set TIGER.....	40
Gambar 29 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set TIGER.....	41
Gambar 30 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set FOX.....	41
Gambar 31 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set FOX.....	42
Gambar 32 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set FOX.....	42
Gambar 33 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set FOX.....	43
Gambar 34 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set ELEPHANT.....	44
Gambar 35 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set ELEPHANT.....	44
Gambar 36 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set ELEPHANT.....	45
Gambar 37 Grafik Perbandingan Learning Time K-Means MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set ELEPHANT.....	45
Gambar 38 Grafik Perbandingan Rata-Rata Learning Time K-Means MI-ELM dan MI-ELM Pada Data Set TIGER.....	46
Gambar 39 Grafik Perbandingan Rata-Rata Learning Time K-Means MI-ELM dan MI-ELM Pada Data Set FOX.....	47
Gambar 40 Grafik Perbandingan Rata-Rata Learning Time K-Means MI-ELM dan MI-ELM Pada Data Set ELEPHANT.....	47

DAFTAR LAMPIRAN

Lampiran 1 Tabel Contoh Data Set TIGER

Lampiran 2 Tabel Contoh Data Set FOX

Lampiran 3 Tabel Contoh Data Set ELEPHANT

Lampiran 4 Tabel Hasil Pra Proses (Seleski Feature) Pada Data Set TIGER

Lampiran 5 Tabel Hasil Pra Proses (Seleski Feature) Pada Data Set FOX

Lampiran 6 Tabel Hasil Pra Proses (Seleski Feature) Pada Data Set ELEPHANT

Lampiran 7 Tabel Hasil Pra Proses (Normalisasi) Pada Data Set TIGER

Lampiran 8 Tabel Hasil Pra Proses (Normalisasi) Pada Data Set FOX

Lampiran 9 Tabel Hasil Pra Proses (Normalisasi) Pada Data Set ELEPHANT

BAB I

PENDAHULUAN

I.1 Latar Belakang

Multiple-Instance Learning (MIL) menjadi suatu jenis pembelajaran dalam bidang pengenalan pola yang menarik untuk dibahas. MIL merupakan solusi dari permasalahan klasifikasi suatu objek yang memiliki beberapa bentuk yang berbeda, misalnya adalah molekul suatu *drug* dapat memiliki beberapa bentuk (konformasi) molekul yang berbeda. Dalam MIL objek direpresentasikan oleh *bag*, di mana *bag* terdiri dari beberapa *feature vector* (dalam terminologi MIL disebut dengan istilah *instance*) yang merepresentasikan setiap bentuk dari objek tersebut. Permasalahan klasifikasi dalam MIL disebut sebagai *Multiple-Instance Classification* (MIC).

Hingga saat ini banyak metode yang telah diusulkan untuk menangani permasalahan MIC, beberapa di antaranya adalah *Axis Parallel Rectangle* (Dietterich, TG, *et. al.* 1997), MI-SVM (Stuart, A, *et. al.* 2003) dan MI-ELM (Qiang, L, *et. al.* 2016). MI-ELM menjanjikan proses pembelajaran model yang lebih cepat dibandingkan algoritma sebelumnya yang bersifat iteratif, seperti MI SVM. Hal ini karena MI-ELM berbasiskan algoritma ELM yang tidak melakukan iterasi dalam penentuan bobot di antara setiap *layer*-nya.

Merujuk pada penelitian yang dilakukan oleh Nan-Ying, L *et. al.* (2006) tentang penerapan *online sequence learning* pada algoritma ELM, penulis tertarik untuk menerapkan *online sequence learning* pada algoritma MI-ELM. Pada *online sequence learning*, pembelajaran dilakukan dengan mempelajari *training data set* satu persatu atau sebagian demi sebagian. Permasalahannya, proses transformasi *feature* pada MI-ELM yang menggunakan fungsi *gaussian similarity* menyebabkan jumlah *hidden node* pada *hidden layer* ke dua menjadi sebanyak jumlah *bag* yang dipelajari. Dengan demikian MI-ELM dengan *online sequence*

learning akan menjadi tidak efisien. Hal ini dikarenakan seiring bertambahnya *bag* yang dipelajari, maka jumlah *hidden node* pada *hidden layer* ke dua juga semakin bertambah. Di samping itu, hal ini juga tidak efisien untuk MI-ELM dengan pembelajaran konvensional (*batch learning*) ketika *bag* yang digunakan saat proses pembelajaran berjumlah banyak. Oleh karena itu, sebelum ke tahap pengembangan MI-ELM dengan pembelajaran secara *online sequence learning*, maka perlu dilakukan penelitian untuk mengatasi masalah besaran dimensi pada *hidden layer* ke dua pada algoritma MI-ELM terlebih dahulu.

Pada permasalahan besaran dimensi di *hidden layer* ke dua, agar jumlah *hidden node* pada *hidden layer* ke dua di MI-ELM dapat diatur, maka diperlukan metode untuk mentransformasi *feature* berbasis pengukuran jarak dengan titik pusat yang dapat ditentukan jumlahnya, dalam penelitian ini menggunakan algoritma *k-means* sebagai metode untuk mentransformasi *feature*.

I.2 Rumusan Masalah

Berdasarkan latar belakang masalah di atas, perumusan masalah yang diteliti adalah bagaimana menerapkan algoritma *k-means* pada proses transformasi *feature* di algoritma MI-ELM?

I.3 Ruang Lingkup Penelitian

Adapun ruang lingkup dalam penelitian ini adalah sebagai berikut :

- a. Penelitian ini belum membahas sampai ke proses pembelajaran MI-ELM secara *online sequence learning*.
- b. Untuk mensimulasikan proses pembelajaran secara *online sequence learning*, dalam penelitian ini dilakukan beberapa kali pembelajaran model menggunakan MI-ELM dengan jumlah *bag* pada *training data set* yang terus bertambah.
- c. *Data set* yang digunakan dalam penelitian ini adalah *data set* MIC yang bersumber dari situs <http://www.miproblem.org>.

- d. Fokus dari penelitian ini adalah melakukan modifikasi proses transformasi *feature* pada algoritma MI-ELM dengan menggunakan algorima *k-means*.
- e. Klasifikasi yang dibahas dalam penelitian ini yaitu berupa *binary classification*.

I.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah mendapatkan metode untuk mengatasi permasalahan MIC berbasis algortima MI-ELM dengan jumlah *hidden node* (pada *hidden layer* ke dua) yang dapat ditentukan serta memiliki performa komputasi yang lebih baik dibandingkan MI-ELM.

I.5 Luaran dan Manfaat Penelitian

Luaran dan manfaat (kontribusi) dari penelitian ini adalah suatu metode usulan, yakni *K-Means* MI-ELM yang dapat mengatasi permasalahan MIC dengan performa komputasi yang lebih baik dibandingkan algortima MI-ELM itu sendiri.

I.6 Sistematika Penulisan

Skripsi ini ditulis dan diuraikan dalam lima (5) bagian besar, yaitu BAB I sampai dengan BAB V. Adapun isi bab-bab tersebut adalah sebagai berikut :

BAB I PENDAHULUAN

Bab ini memberikan gambaran secara garis besar mengenai isi skripsi sehingga pembaca dapat memahaminya dengan mudah. Isi dari bab ini adalah latar belakang, rumusan masalah, ruang lingkup penelitian, tujuan penelitian, luaran dan manfaat, serta sistematika penulisan yang digunakan untuk menyusun laporan ini.

BAB II STUDI LITERATUR

Bab ini berisi tentang penjelasan mengenai penelitian terdahulu, yaitu tentang *Extreme Learning Machine* (ELM), *Multi-Instance Extreme Learning Machine* (MI-ELM), dan *Clustering In Extreme Learning*

Machine Feature Space. Pada bab ini juga berisi tentang landasan teori mengenai *Multiple-Instance Learning* (MIL), Algoritma *K-means*, dan *K-Fold Cross Validation*.

BAB III METODE PENELITIAN

Bab ini berisi tentang penjelasan mengenai metodologi yang digunakan dalam penelitian ini.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisikan penjelasan hasil uji coba yang telah dilakukan dalam penelitian ini, serta pembahasan atau analisa dari hasil uji coba tersebut.

BAB V PENUTUP

Bab ini berisi tentang penjelasan mengenai kesimpulan dan saran yang dapat membangun serta meningkatkan maksud dan tujuan dari penelitian ini menuju arah yang lebih bermanfaat untuk banyak orang.

DAFTAR PUSTAKA

DAFTAR RIWAYAT HIDUP

LAMPIRAN

BAB II

STUDI LITERATUR

II.1 Penelitian Terdaulu

Berikut ini akan dipaparkan hasil penelitian terdahulu yang berhubungan dengan penelitian yang akan dilakukan peneliti sekaligus dijadikan sebagai rujukan dalam penelitian ini.

II.1.1 Guang-Bin, H., et. al. (2006)

Melakukan penelitian yang berjudul “*Extreme Learning Machine: Theory and Application*”. Penelitian ini dilatar belakangi oleh luasnya penggunaan *feedforward neural network* yang karena kemampuannya dalam hal: 1) melakukan aproksimasi pemetaan persoalan nonlinear yang kompleks berdasarkan *input sample*, 2) dapat memberikan beragam model untuk fenomena nyata maupun buatan dalam skala besar yang sulit untuk ditangani oleh teknik paramterik klasik. Namun di sisi lain, ada kekurangan yakni berupa tidak dimilikinya algoritma pembelajaran yang lebih cepat untuk *neural newtorks*. Pada model algoritma pembelajaran tradisional, umumnya melakukan pembelajaran dengan lambat dan membuat kita harus menunggu lama hingga proses pembelajaran tersebut selesai. Tidak heran pula jika mendapati proses pembelajaran yang membutuhkan waktu berjam-jam, berhari-hari dan mungkin lebih.

Algoritma pembelajaran yang diusulkan pada penelitian ini adalah sebuah algoritma pembelajaran yang sederhana, yakni model pembelajaran *feedforward newtwork* dengan berbasiskan *Single Layer Feedforward Networks* (SLFNs). SLFNs merupakan *neural networks* yang hanya memiliki satu buah *hidden layer*. Komponen pada SLFNs yaitu berupa *input layer*, *hidden layer*, dan *output layer* seperti yang digambarkan pada Gambar 1. Secara matematis persamaan SLFNs dapat dimodelkan sebagai berikut:

$$\sum_{i=1}^{\tilde{N}} \beta_i g(x_j) = \sum_{i=1}^{\tilde{N}} \beta_i g(w_i \cdot x_j + b_i) = \mathbf{o}_j \quad (1)$$

$$j=1, \dots, N$$

di mana $w_i = [w_1, w_2, w_3, \dots, w_n]^T$ adalah vektor bobot *input* yang menghubungkan antara *input layer* dengan *hidden layer*, sedangkan b adalah bias pada *hidden node* ke- i . Sedangkan $\beta_i = [\beta_{i1}, \beta_{i2}, \beta_{i3}, \dots, \beta_{im}]^T$ adalah vektor bobot yang menghubungkan antara *hidden layer* dengan *output layer* dan $g(x)$ merupakan fungsi aktifasi yang umum digunakan, seperti *sigmoid*, *radial basis*, *sine*, *cosine*, *exponential*, dsb. Sebagai contoh pada kasus regresi bisa menggunakan fungsi aktifasi *sigmoid biner* dapat dilihat pada persamaan (2) dan untuk kasus klasifikasi bisa menggunakan fungsi *sigmoid bipolar (Hyperbolic tangent function)* dapat dilihat pada persamaan (3) (Gao, H. 2015, hlm. 34).

$$g(x) = \frac{1}{(1+\exp^{-x})} \quad (2)$$

$$g(x) = \frac{(1-\exp^{-x})}{(1+\exp^{-x})} \quad (3)$$

Pada persamaan (1) di atas secara matematis dapat dituliskan menjadi sebagai berikut.

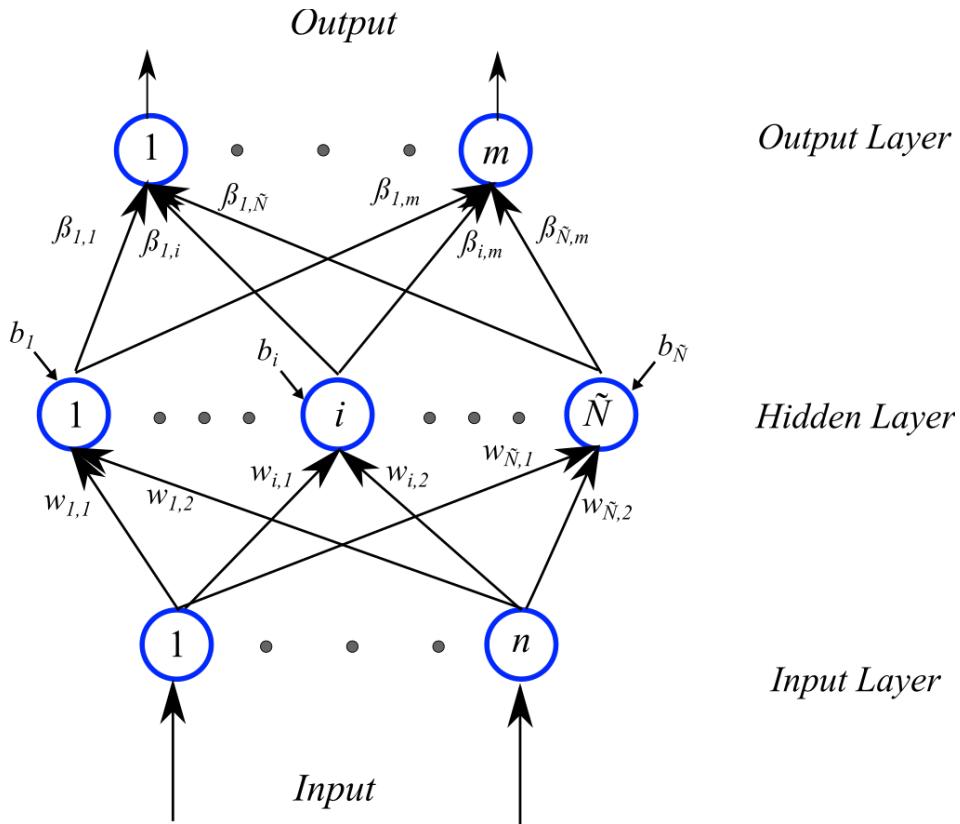
$$H\beta = T \quad (4)$$

di mana,

$$H = \begin{bmatrix} g(w_1 \cdot x_1 + b_1), & \dots & g(w_{\tilde{N}} \cdot x_1 + b_{\tilde{N}}) \\ \vdots & \ddots & \vdots \\ g(w_1 \cdot x_N + b_1), & \dots & g(w_{\tilde{N}} \cdot x_N + b_{\tilde{N}}) \end{bmatrix} \quad (5)$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad \text{dan} \quad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_{\tilde{N}}^T \end{bmatrix}_{\tilde{N} \times m} \quad (6)$$

Pada SLFNs, *training set* terbentuk dari N sembarang data sampel yang berupa data aktual. *Training set* di sini kita notasikan sebagai berikut (x_i, t_i) , di mana $x_i = [x_1, x_2, x_3, \dots, x_n]^T \in \mathbb{R}^n$ merupakan vektor *feature* dan $t_i = [t_1, t_2, t_3, \dots, t_m]^T \in \mathbb{R}^m$ merupakan vektor target atau label.



Gambar 1 Arsitektur SLFNs *Extreme Learning Machine*

a. Algoritma Pembelajaran ELM

Secara singkat algoritma pembelajaran ELM dapat dijelaskan dalam beberapa langkah, yaitu sebagai berikut:

Input : inisialisasi *training set* $\mathcal{D} = \{(x_i, t_i) | x_i \in \mathbb{R}^n, t_i \in \mathbb{R}^m\}$, fungsi aktivasi berupa $g(x)$ dan jumlah *hidden node* sebanyak \tilde{N} node.

Output : Bobot *output* β

- 1) Inisialisasi bobot *input* w_i dan b_i secara acak, di mana $i=1, \dots, \tilde{N}$.

- 2) Ekstraksi *feature* : Dapatkan nilai *output hidden layer* (H) dengan menggunakan persamaan (5).
- 3) Hitung bobot *output* (β) : Dari persamaan (4) dapat diperoleh persamaan untuk mendapatkan nilai β yaitu sebagai berikut:

$$\beta = H^\dagger \cdot T \quad (7)$$

di mana H^\dagger adalah *Moore-Penrose generalized inverse* dari matrix H .

b. Evaluasi Performa ELM

Berbeda dengan algoritma *Backpropagation* di mana terdapat langkah *backward propagation* dan pembaharuan bobot w , ELM tidak memiliki kedua langkah tersebut, hal inilah yang menyebabkan semakin tingginya tingkat *learning speed* pada algoritma ELM. Pengujian performa ELM pada penelitian ini dilakukan pada beberapa kasus, salah satunya adalah regresi. Pada kasus regresi ini Guang-bin Huang, *et.al.* mencoba melakukan regresi terhadap dua jenis *data set*, yang pertama adalah fenomena buatan yakni berupa aproksimasi fungsi ‘sinC’ sebagai berikut.

$$y(x) = \begin{cases} \sin(x)/x, & x \neq 0 \\ 1, & x = 0 \end{cases} \quad (8)$$

dan ke dua adalah fenomena aktual yakni menggunakan 13 *data set* yang merupakan *real-world benchmark datasets*¹. Algoritma pembelajaran yang dijadikan sebagai pembanding algoritma ELM adalah *Backpropagation* (BP) dan *Support Vector Machine for Regression* (SVR). Pada pengujian pertama, ELM mampu mengestimasi fungsi ‘sinC’ dengan *learning speed* jauh lebih cepat dibanding BP dan SVR. Pada tahap *training* ELM hanya membutuhkan waktu 0.125 detik dengan RMSE sebesar 0.1148. Pada tahap *testing* ELM membutuhkan waktu 0.031 detik dengan RMSE sebesar 0.0097. Sedangkan algoritma BP dan SVR pada tahap *training* masing-masing membutuhkan waktu 21.26 detik dan 1273.4 detik dengan RMSE masing-masing sebesar 0.1196 dan

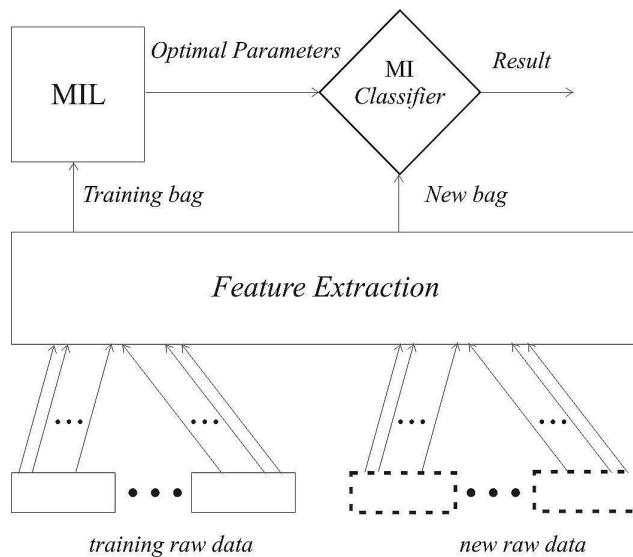
¹ http://www.niaad.liacc.up.pt/#ltorgo/Regression/ds_menu.html

0.1149. Pada tahap *testing* BP dan SVR masing-masing membutuhkan waktu 0.032 detik dan 5.9087 dengan RMSE masing-masing sebesar 0.0159 dan 0.0130. Pada pengujian ke dua, hasil yang diperoleh dari percobaan tersebut adalah ELM dapat mengestimasi 7 buah *data set* lebih akurat dari BP dan 5 buah *data set* lebih akurat dari SVR. Ketentuan yang digunakan pada percobaan ke dua ini adalah suatu algoritma dikatakan lebih unggul dari algoritma lainnya jika nilai RMSE yang diperoleh algoritma tersebut bernilai 0.005 lebih rendah dibanding RMSE yang diperoleh algoritma pembandingnya. Dari pengujian pertama dan ke dua menunjukkan bahwa ELM terbukti memiliki *learning speed* yang jauh lebih cepat dibanding BP dan SVR, selain *learning speed* ELM juga unggul dalam hal akurasi yang ditunjukkan oleh rendahnya nilai RMSE yang diperoleh pada tahap *training* maupun *testing*.

II.1.1.1 Qiang, L, et. al. (2016)

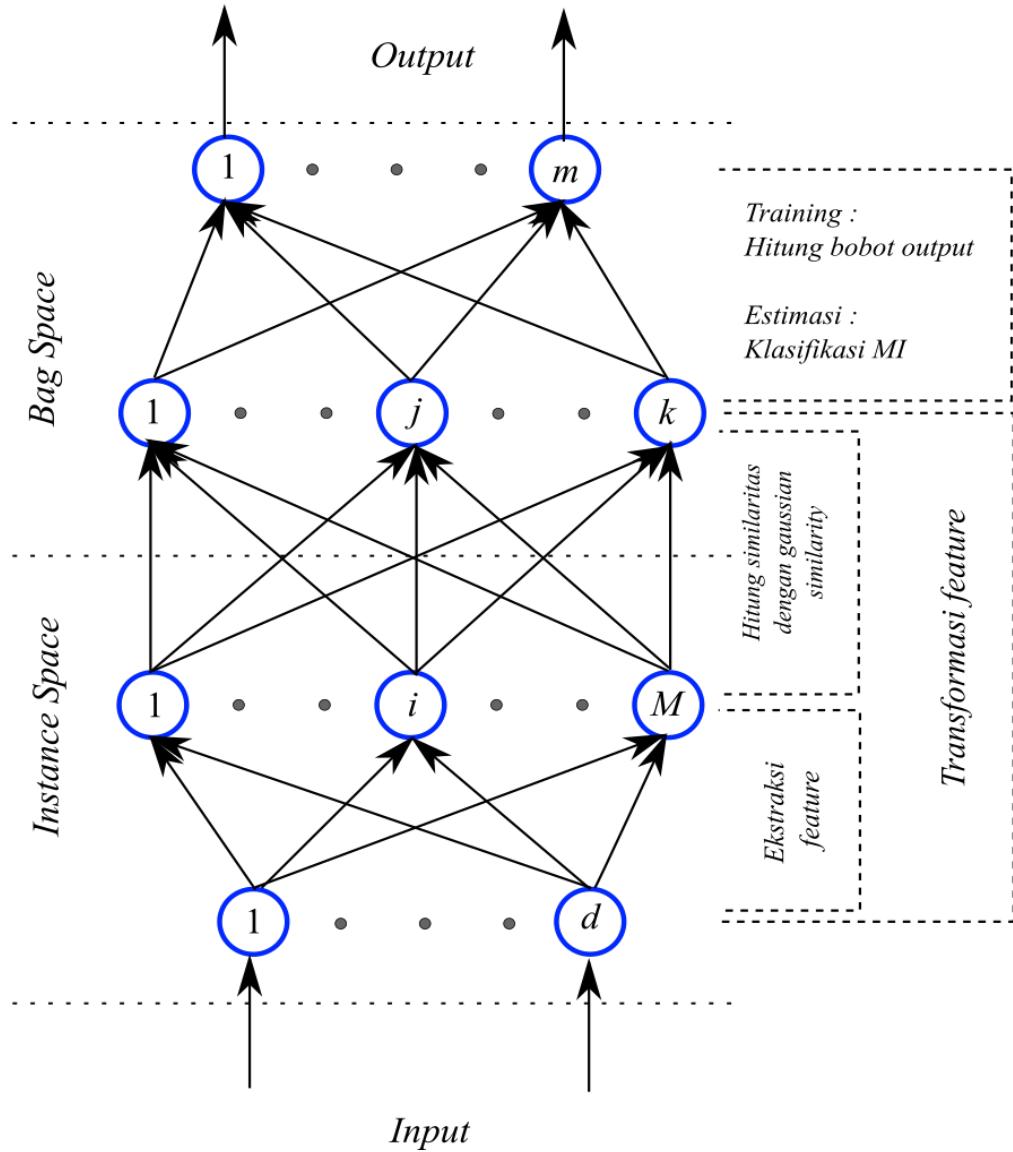
Melakukan penelitian berjudul “*MI-ELM: Highly efficient multi-instance learning based on hierarchical extreme learning machine*”. Tujuan penelitian ini adalah mencoba meningkatkan performa estimasi pada kasus *Multi-Instance Classification* (MIC). Qiang, L. et. al. merujuk pada penelitian yang dilakukan oleh Fu, Z. et. al. (2013). Fu, Z. et. al. (2013) melakukan penelitian berjudul *Learning Sparse Kernel Classifier for Multi-Instance Classification* dengan berfokus pada tingkat keakurasaian dan keefesiensian dalam melakan klasifikasi pada permasalahan MIC.

Namun, metode yang diusulkannya adalah suatu metode yang *iterative*, yakni metode yang melakukan perulangan dalam penentuan parameternya, sehingga membutuhkan waktu yang lama dalam proses pembelajarannya. Kemudian Qiang, L. *et.al.* mencoba mencari suatu alternatif yaitu berupa metode lain yang *non-iterative* guna mengurangi waktu yang dibutuhkan dalam proses pembelajaran.



Gambar 2 Framework of Supervised MI Classification

Qiang, L., *et.al.* (2016, hlm. 1047-1048) kemudian menemukan suatu metode yang diusulkan oleh Huang-bin Guang yaitu *Extreme Learning Machine* (ELM). Terinspirasi dari karakteristik yang dimiliki ELM, Qiang Liu *et.al.* mencoba mengusulkan suatu metode pembelajaran yang sangat efisien yaitu *Multi-Instance Extreme Learning Machine* (MI-ELM). Metode yang diusulkannya ini diilustrasikan pada Gambar 3. MI-ELM melakukan sedikit modifikasi pada ELM, yakni menambahkan sebuah *single-layer* pada ELM, hal ini menjadikan MI-ELM berarsitektur *Double-hidden Layer Feedforward Networks* (DLFNs).



Gambar 3 Alur Kerja MI-ELM

Bentuk umum dari *training data set* pada MI-ELM adalah terdiri dari L bag $D' = \{(\chi_i, t_i) | t_i \in \mathbb{R}^m, i=1, \dots, L\}$ di mana $t_i = [t_{i,1}, \dots, t_{i,m}]^T$ yang merupakan vektor label dari bag ke- i . Setiap bag terdiri dari n_i instance, misal $\chi_i = \{x_{1,i}, \dots, x_{i,n_i}\}, x_{i,j} \in \mathbb{R}^n, j=1, \dots, n_i$. Dan N adalah jumlah keseluruhan

instance pada *training data set*. Dengan demikian $\sum_i^n n_i = N$

a. Algoritma MI-ELM

Pada Gambar 3 terlihat bahwa terdapat dua fase dalam proses pembelajaran MI-ELM, yaitu sebagai berikut:

1) Fase Transformasi *Feature*

Fase transformasi adalah proses ekstraksi *feature* pada MI-ELM dengan cara mentransformasi *instance-level feature* menjadi *bag-level feature*. *Instance-level feature* adalah sekumpulan dari beberapa vektor *feature* dalam suatu *bag* yang terdapat pada data *input*. Sedangkan *bag-level feature* adalah sebuah vektor *feature* yang merupakan nilai *output* pada *hidden layer* ke dua. Proses transformasi *feature* ini dilakukan dalam dua langkah, yaitu sebagai berikut.

a) Ekstraksi *Feature*

Ekstraksi *feature* dilakukan untuk memperoleh nilai *output* pada *hidden layer* pertama (H) dengan menggunakan persamaan (5). Di sini matriks H disebut sebagai *new training set* dan dinotasikan dengan simbol χ' dengan $x' \in \chi'$ merupakan vektor *feature* pada *hidden layer* pertama. χ' .

b) Pembentukan *Bag-level Feature*

Bag-level feature diperoleh dengan menghitung nilai rata-rata dari setiap *output instance-level classifier*. Berikut adalah persamaan untuk memperoleh *bag-level feature*.

$$F(\chi_i) = \frac{1}{n_{i,j}} \sum_{j=1}^{n_i} f(x'_{i,j}), i=1, \dots, L \quad (9)$$

di mana fungsi f adalah *instance classifier* yang berbasis pengukuran jarak (similaritas). Dalam penelitian ini fungsi similaritas dinotasikan dengan $g(x_a, x_b)$ yang berarti melakukan pengukuran similaritas antara x_a dengan x_b . Untuk mengukur similaritas dalam penelitian ini menggunakan fungsi *gaussian similarity*, $g^r(\|x - x_c\|) = \exp\{-\|x - x_c\|^2 * r / (2 \cdot \sigma^2)\}$, di mana x_c adalah *center sample point* yang diperoleh dari nilai rata-rata

semua *instance* yang terdapat pada *bag* χ' ke- l dengan $l=(1,\dots,L)$. Secara keseluruhan *instance classifier* pada

persamaan (9) adalah $f(\chi_i)=\sum_{l=1}^L \beta_l g^r(\mathbf{x}'_{i,j}, \mathbf{y}'_l)$ di mana

$\mathbf{y}'_l (l=1,\dots,L)$ adalah *center point* suatu *bag* dari setiap *bag*.

Nilai *center point* diperoleh dengan menghitung nilai rata-rata setiap *new instance* pada suatu *bag* dari semua *bag*. Dengan demikian unutk memperoleh vektor *feature* untuk setiap *bag* menggunakan persamaan (10). Sekumpulan vektor *feature* inilah yang kemudian disebut sebagai *bag-level feature*.

$$s(\chi_i)=\frac{1}{n_i}[\sum_{j=1}^{n_i} g^r(\mathbf{x}'_{i,j}, \mathbf{y}'_1), \dots, \sum_{j=1}^{n_i} g^r(\mathbf{x}'_{i,j}, \mathbf{y}'_L)] \quad (10)$$

2) Fase Klasifikasi *Bag-level Feature*

Setelah *instance-level feature* ditransformasi menjadi *bag-level feature* langkah selanjutnya adalah melakukan klasifikasi dengan persamaan sebagai berikut.

$$F(\chi_i)=\sum_{j=1}^{n_i} \beta_j s^j(\chi_i) \quad (11)$$

di mana $s^j(\chi_i)$ adalah nilai *bag feature* ke- j pada *bag* ke- i dengan kata lain $s^j(\chi_i)$ dapat diartikan sebagai nilai similaritas *bag* ke- i terhadap *bag* ke- j .

Optimalisasi terhadap MI-ELM pun dilakukan oleh Qiang Liu *et.al.* (2016) agar *output* dari algoritma ini bisa sedekat mungkin dengan label (target). Dari optimalisasi yang dilakukan diperoleh fungsi untuk menghitung nilai bobot *output* β yaitu dengan menggunakan persamaan (12). Sedangkan persamaan untuk memperoleh nilai *output* dari MI-ELM adalah dengan menggunakan persamaan (13).

$$\beta=G^T \left(\frac{\mathbf{I}}{C} + GG^T \right)^{-1} T^T \quad (12)$$

$$F(\chi) = g(\chi) G^T \left(\frac{I}{C} G G^T \right)^{-1} T^T \quad (13)$$

di mana $T = [t_1, \dots, t_L]$ adalah matriks yang terdiri dari vektor label setiap *bag* pada *training data set*. Sedangkan $G = [g(\chi_1), \dots, g(\chi_L)]$ dengan $g(\chi_i) = s(\chi_i)$.

Secara singkat algoritma pembelajaran MI-ELM dapat dijelaskan dalam beberapa langkah, yaitu sebagai berikut :

Input : inisialisasi *training set* yang terdiri dari L *bag* $D' = \{(\chi_i, t_i) | t_i \in \mathbb{R}^m, i=1, \dots, L\}$, fungsi aktivasi berupa $g(x)$ dan jumlah *hidden node* sebanyak M *node*.

Output : Bobot *output* β

- 1) Inisialisasi bobot *input* w_i dan b_i secara acak, di mana $i=1, \dots, M$.
- 2) Transformasi *feature* : Hitung nilai *output* dari *hidden layer* pertama berdimensi M menggunakan persamaan (5). Kemudian hitung nilai *output* dari *hidden layer* ke dua (G) mengikuti persamaan (10).
- 3) Hitung bobot *output* β mengikuti persamaan (12)

b. Evaluasi Performa MI-ELM

Pengujian performa dilakukan dengan menggunakan beberapa MI *data set*, yaitu MUSK1 dan MUSK2 merupakan *data set* prediksi terhadap aktivitas suatu obat, TIGER, FOX dan ELEPHANT merupakan *data set* pengklasifikasian citra, dan TST1-4, TST7, TST9, dan TST10 merupakan *data set* pengklasifikasian teks. Sebagai perbandingan metode yang diusulkan, MI-ELM dibandingkan dengan SVM for MI *classification*. Hasil perbandingan antara ke dua metode pembelajaran tersebut menunjukkan bahwa kecepatan klasifikasi MI-ELM lebih cepat ± 10 kali lebih cepat dibandingkan SVM. Demikian juga dalam hal akurasi, MI-ELM memiliki nilai ± 10 poin di atas akurasi yang diperoleh SVM.

II.1.2 Qing, H, et. al. (2014)

Melakukan penelitian berjudul '*Clustering In Extreme Learning Machine Feature Space*'. Pada penelitian ini Qing, H, et. al. melakukan imporvisasi terhadap algortima ELM yaitu dengan menjadikan ELM sebagai *learning machine* yang dapat digunakan untuk melakukan *clustering* data. Algoritma ELM adalah *learning machine* yang dimodelkan untuk pengenalan pola pada kasus *supervised learning*, baik dalam permasalahan estimasi regresi maupun klasifikasi. Berbeda dengan kasus *supervised learning*, di mana *training data* yang digunakan memiliki nilai target atau label. *Clustering* masuk ke dalam kasus *unsupervised learning*, di mana *training data* yang digunakan tidak memiliki nilai target atau label. Di sini, Qing, H et. al. memodifikasi algoritma ELM agar dapat menyelesaikan permasalahan pada kasus *unsupervised learning* dengan menggunakan algoritma *k-means* pada *feature space (hidden layer)*.

Nilai *output hidden layer* pada ELM diperoleh dengan menggunakan persamaan (3). Menurut Qing, H, et. al. melihat aproksimasi ELM yang universal dan kapabilitas klasifikasi dari ELM, penggunaan jumlah *hidden node* yang cukup besar akan menjamin data akan terpisahkan secara linear pada *feature space*. Setelah nilai *output hidden layer* sudah didapatkan, metode *clustering* tradisional seperti *k-means* dapat digunakan.

a. Algoritma ELM *K-means*

Input : Inisialisasi secara vektor bobot *input w* dan *bias b* acak.

k : jumlah *cluster*, *L* : jumlah *hidden nodes* pada ELM, dan *D* : *training data* berisikan *m* objek

Output : kelompok data yang sudah terbagi-bagi menjadi *k cluster*.

- 1) Hitung nilai *output hidden layer H* : Ekstraksi *feature* ELM dengan persamaan (5).
- 2) Tentukan secara acak sejumlah *k* objek sebagai nilai *centeroid* awal.
- 3) Kelompokan setiap objek ke dalam *cluster* yang memiliki nilai jarak *centeroid* terdekat terhadap objek tersebut.

- 4) Perbaharui nilai *centeroid* dengan cara menghitung nilai *mean* dari semua objek yang ada pada setiap *cluster*.
- 5) Ulangi langkah 3-4 hingga tidak ada lagi perpindahan objek dari *cluster* satu dengan *cluster* lainnya (dengan kata lain, nilai *centeroid* sudah tidak berubah lagi) atau sampai batas iterasi maksimum yang telah ditentukan.

II.2 Landasan Teori

II.2.1 *Multiple-Instance Learning*

Multiple-Instance Learning (MIL) adalah salah satu dari varian *supervised learning*, di mana *task* dari MIL tersebut ialah untuk mempelajari pola suatu *data set* yang terdiri dari beberapa *bag*. Dalam terminologi MIL, *bag* adalah sekumpulan *feature vector* yang merepresentasikan suatu objek, menggantikan *feature vector* tunggal seperti pada kasus *supervised learning* konvensional pada umumnya. *Feature vector* pada suatu *bag* disebut dengan *instance*. Sehingga *multiple-instance* dapat diartikan sebagai sekumpulan *feature vector* yang terdapat pada *bag* (Amores, J. 2013, hlm. 81).

Untuk memberikan gambaran mengenai *data set* berbentuk *multiple-instance* dapat dilihat pada contoh permasalahan klasifikasi *drug* (Dietterich, T. G., *et.al.* 1997, hlm. 36-38). Permasalahan di sini adalah melakukan klasifikasi *drug* yang bagus dan *drug* yang tidak bagus. Pada umumnya *drug* merupakan suatu molekul kecil yang bekerja dengan cara melekat pada target (*binding site*) berupa molekul protein dengan ukuran jauh lebih besar dari ukuran molekul *drug* itu sendiri, misalnya seperti enzim. Suatu *drug* dikatakan bagus apabila memiliki bentuk molekul (konformasi) yang mendekati bentuk dari rongga yang ada pada molekul target. Dengan demikian, *drug* tersebut dapat melekat dengan baik pada molekul target.

Permasalahannya, suatu *drug* dapat memiliki lebih dari satu konformasi dan hanya ada satu atau beberapa konformasi saja yang dapat melekat pada target dengan baik. Di samping itu, jumlah konformasi dari suatu molekul *drug* dapat

berbeda-bebeda antara *drug* yang satu dengan *drug* lainnya. Dalam kasus ini, untuk merepresentasikan informasi *drug* ke dalam bentuk *data set multiple-instance* adalah *drug* direpresentasikan ke dalam bentuk *bag* $X = [x_1, \dots, x_n]$, di mana $x_i, i=1, \dots, n$ adalah *feature vektor (instance)* yang merepresentasikan konformasi ke-*i* dari molekul *drug* tersebut, sedangkan *n* adalah jumlah konformasi yang dimiliki *drug* tersebut.

Contoh lainnya adalah pada kasus pengklasifikasian citra. Seperti yang dijelaskan oleh Maron, O & Ratan, A. L (2001, hlm. 426-428) bahwa setiap citra direpresentasikan ke dalam bentuk *bag*. Sedangkan lokal area dari citra tersebut direpresentasikan ke dalam bentuk *instance*. Misalnya, jika *bag* berlabel “*waterfall*” (positif) maka setidaknya ada satu dari beberapa lokal area dalam citra tersebut yang mengandung konten air terjun. Jika *bag* berlabel “*non-waterfall*” (negatif) maka tidak ada satu pun lokal area yang mengandung konten air terjun.

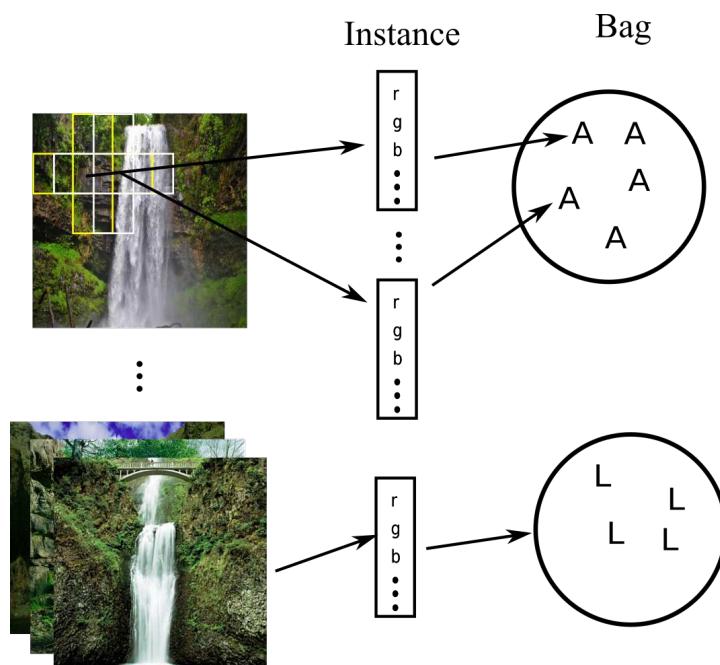


Gambar 4 Contoh Pembentukan *Nstance* Dengan SBN. (a) Pembentukan Citra Menjadi 8x8 Piksel. (b) Contoh SB Berukuran 2x2 Piksel. (c) Contoh SBN

Setiap *instance* merepresentasikan *feature* dari setiap lokal area. Sebagai contoh, dalam pembahasan ini nilai *feature* dari setiap lokal area diperoleh dengan menggunakan metode *Single Blob with Neighbors* (SBNs) sebagai pembangkit *Bag* dari suatu citra. Pada SBN, sebuah lokal area adalah sebuah *Single Blob* (SB) berukuran 2x2 piksel dari suatu citra dengan empat buah *blob* yang bertetanggan

pada sisi atas, kiri, bawah dan kanannya. Pada Gambar 4 ditunjukkan pembentukan *instance* pada suatu citra dengan jumlah piksel 8x8.

Setiap *Instance* terbentuk dari nilai rata-rata setiap kanal warna (Merah, Hijau, Biru) pada SB tengah dan selisih dari rata-rata setiap kanal warna pada empat SB tetangga terhadap SB tengah. Dengan demikian setiap *instance* akan berbentuk vektor $x_i = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{i15}]$, di mana x_i adalah *instance* ke-*i*. Sedangkan x_{i1}, x_{i2}, x_{i3} adalah nilai rata-rata kanal warna SB tengah, x_{i3}, x_{i4}, x_{i5} nilai selisih dari rata-rata setiap kanal warna pada empat SB tetangga terhadap SB tengah, dst. Sedangkan *bag* dibentuk dari sekumpulan beberapa *instance* seperti yang ditunjukkan pada Gambar 5. Sehingga citra dapat direpresentasikan sebagai *bag* $X = [x_1, \dots, x_n]$, di mana $x_i, i=1, \dots, n$ adalah *instance* ke-*i* dan *n* adalah jumlah *instance* (lokal area dari citra tersebut).



Gambar 5 Contoh Pembentukan *Bag* Dari Sebuah Citra

Dalam melakukan pembangkitan *bag* dari suatu citra, selain menggunakan metode SBN seperti contoh di atas, dapat juga dilakukan dengan beberapa metode lainnya. Xiu-Shen, W. & Zhi-Hua, Z. (2016, hlm. 5) melakukan survei terhadap tujuh metode pembangkitan *bag*. Beberapa di antaranya adalah *Row*, *Single Blob*,

Single Blob with Neighbors dan *Blobworld* yang dikategorikan ke dalam pembangkit *bag* berbasis non-segmentasi. Sedangkan tiga pembangkit *bag* lainnya adalah *k-meansSeg*, *WavSeg* dan *JSEG-bag* dikategorikan ke dalam pembangkit *bag* berbasis segmentasi.

Dalam permasalahan MIC seperti contoh di atas, banyak metode yang telah diusulkan. Beberapa di antaranya adalah *Axis Parallel Rectangle* (Dietterich, TG, et. al. 1997), MI-SVM (Stuart, A, et. al. 2003) dan MI-ELM (Qiang, L, et. al. 2016).

II.2.2 Algoritma *K-means*

K-means Clustering merupakan metode *non-heirarchical* yang pada awalnya mengambil sebagian data dari populasi untuk dijadikan pusat *cluster* awal secara acak. Berikutnya *k-means* menghitung jarak setiap data dengan pusat *cluster* menggunakan metode pengukuran *euclidean distance*. Tahap selanjutnya data akan dikelompokkan ke dalam *cluster* yang memiliki jarak pusat *cluster* terdekat. Pusat *cluster* yang baru diperoleh dengan menghitung nilai rata-rata semua data pada masing-masing *cluster*. Proses pengelompokan data ke dalam *cluster* dan pembaruan nilai pusat *cluster* dilakukan berulang hingga sudah tidak ada lagi data yang berpindah *cluster*; nilai pusat *cluster* ke-*t* dengan pusat *cluster* ke-*t*+1 tidak berubah atau sudah mencapai batas iterasi yang ditentukan (McLachlan, G.J. and Peel, D., 2000).

II.2.3 *K-Fold Cross Validation*

Cross Validation merupakan salah satu teknik yang dapat digunakan untuk mengevaluasi *output* model pembelajaran, yaitu berupa nilai estimasi. Hal ini bertujuan untuk mendapatkan parameter yang optimal bagi model pembelajaran tersebut.

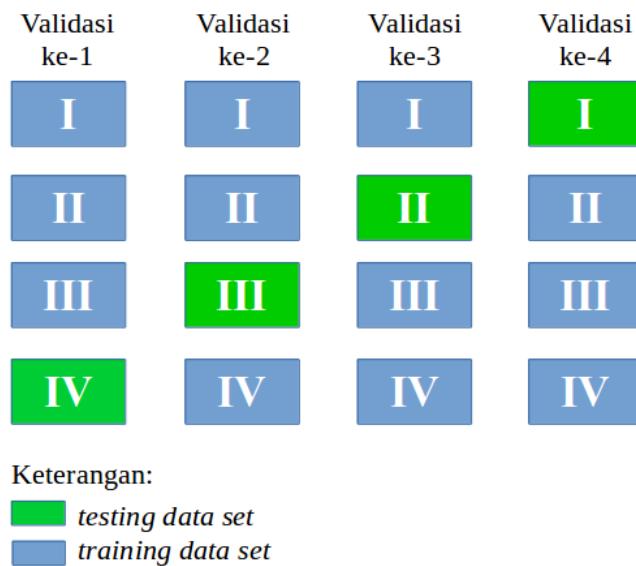


Gambar 6 Ilustrasi Pembagian Data menjadi Beberapa Kelompok

Pada umumnya, teknik yang digunakan untuk mengevaluasi suatu model pembelajaran adalah dengan membagi *data set* menjadi dua bagian, yaitu 70% sebagai *training data set* dan 30% sebagai *testing data set*. Kemudian tingkat kesalahan estimasi pada *testing data set* dihitung menggunakan teknik pengukuran kesalahan, semisal *Rooted Mean Square Error*. Namun, teknik konvensional ini kurang merepresentasikan kehandalan suatu model pembelajaran dalam melakukan estimasi. Hal ini disebabkan karena bisa saja sebaran data pada *training data set* tidak bagus. Oleh karena itu, penggunaan teknik *crossing* (pertukaran) *trainig data set* dengan *testing data set* dianggap sebagai cara yang tepat (Grossman, R, Seni, G, Joh, Agrwal, N, Huan, 2010).

Pada *k-fold cross validation*, *data set* dipecah menjadi *k* kelompok. Di mana kelompok pertama hingga kelompok ke $k - 1$ menjadi *training data set* dan kelompok ke *k* menjadi *testing data set*. Validasi dilakukan sebanyak *k* kali. Di mana, pada setiap perulangannya dilakukan pertukaran *training data set* dengan salah satu kelompok pada *training data set*. Pertukaran ini dilakukan sampai seluruh kelompok menjadi *training set* masing-masing sebanyak satu kali. Ilustrasi pembentukan kelompok dan pertukaran (rotasi) *training data set* masing-masing ditunjukkan pada Gambar 6 dan Gambar 7. Evaluasi terhadap model dilakukan dengan melihat pada validasi ke berapa nilai *error* terendah diperoleh (McLachlan, G.J., Do, Kim-Anh; Ambroise, Christophe, 2004). Dengan demikian

parameter yang optimal untuk pembentukan model dapat diperoleh dari hasil pembelajaran tersebut.

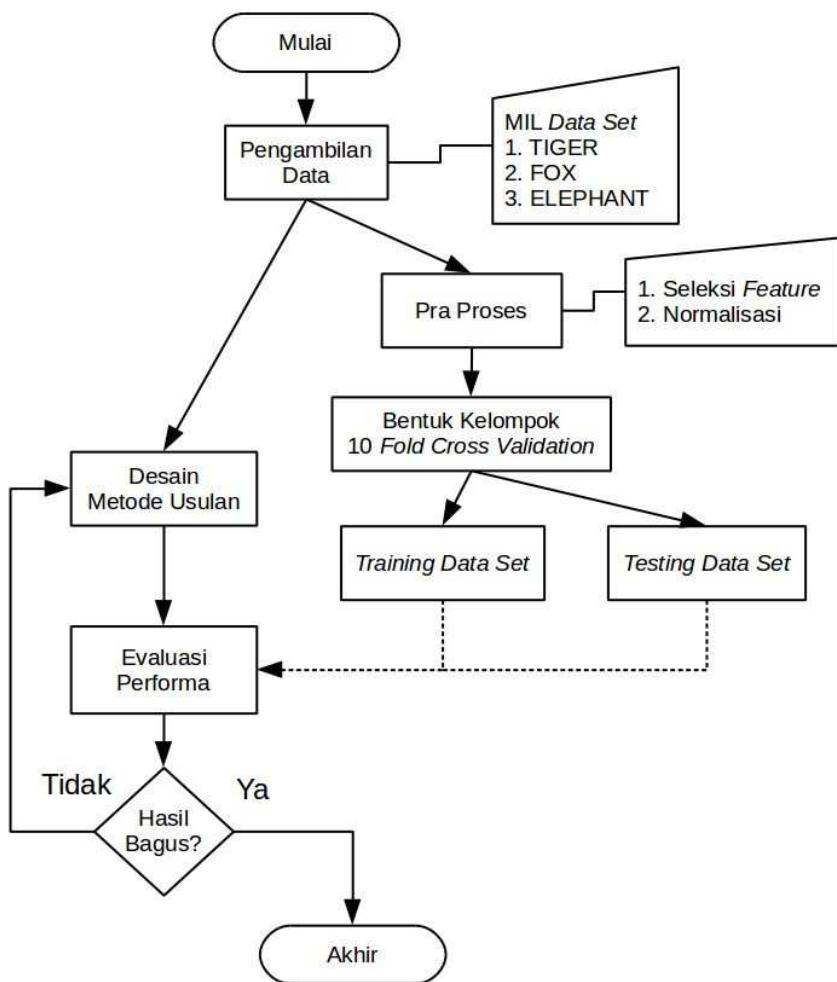


Gambar 7 Ilustrasi Rotasi *Training Data Set* dengan 4-Fold Cross Validation

BAB III

METODOLOGI PENELITIAN

Untuk mendapatkan hasil yang tepat dan akurat, maka perlu dirumuskan suatu metodologi penelitian berupa prosedur yang dilakukan dalam penelitian. Adapun metodologi penelitian yang dirumuskan dalam penelitian ini ditunjukkan pada Gambar 8.



Gambar 8 Metodologi Penelitian

III.1 Pengambilan Data

Data yang digunakan dalam penelitian ini adalah tiga buah *data set* kategorisasi citra, yaitu TIGER, FOX dan ELEPHANT seperti yang diperkenalkan oleh Stuart, A., *et. al.* (2003). Data tersebut diambil dari situs <http://www.miproblem.org>.

Tabel 1 Ringkasan Informasi *Data Set* yang Digunakan

Nama <i>Data Set</i>	Jumlah <i>Bag</i>	Jumlah <i>Instance</i>	Jumlah <i>Feature</i>
TIGER	200	1220	230 (143 <i>non-zero</i>)
FOX	200	1320	230 (143 <i>non-zero</i>)
ELEPHANT	200	1391	230 (143 <i>non-zero</i>)

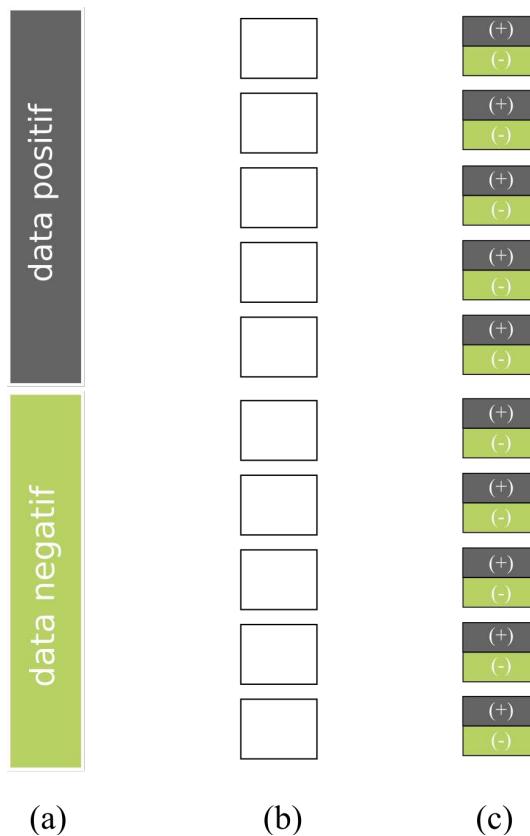
Stuart, A, *et. al.* (2003) memperkenalkan MI *data set* dalam penelitiannya pada kasus kategorisasi citra. Terdapat tiga kategori dalam *data set* yang diperkenalkannya, yaitu TIGER, FOX, dan ELEPHANT. Setiap *data set* terdiri dari 100 *bag* positif dan 100 *bag* negatif. Adapun ringkasan informasi tentang *data set* yang digunakan dalam penelitian ini dapat dilihat pada Tabel 1. Contoh *data set* yang digunakan dalam penelitian ini dapat dilihat pada Lampiran 1, 2 dan 3 untuk masing-masing *data set* TIGER, FOX dan ELEPHANT.

III.2 Pra Proses

Pra proses yang dilakukan terhadap *data set* yang digunakan dalam penelitian ini ialah melakukan seleksi *feature* dan normalisasi. Seleksi *feature* bertujuan untuk memilih *feature* yang representatif sebagai parameter dalam melakukan pemodelan. Dalam penelitian ini *feature* yang tidak representatif seperti *zero feature* dan *feature* yang tidak memiliki keragaman nilai akan dihapus. Sedangkan normalisasi *data set* adalah proses penskalaan nilai *feature*, hal ini dilakukan untuk membuat nilai *feature* menjadi berada di dalam *range* yang sama. Dalam penelitian ini, nilai *feature* dinormalisasikan ke dalam *range* -1 sampai 1. Adapun fungsi normalisasi yang digunakan adalah fungsi *sigmoid bipolar* seperti yang ditunjukkan pada persamaan (3).

III.3 Membentuk Kelompok 10 Fold Cross Validation

Dalam penelitian ini menggunakan *10 fold cross validation* yang bertujuan untuk mengevaluasi model pada saat melakukan pembelajaran. Untuk menerapkan *10 fold cross validation*, *data set* dikelompokkan menjadi 10 kelompok. Sebelum *data set* dikelompokkan, *data set* diacak terlebih dahulu pada setiap kategorinya. Agar data yang digunakan pada proses pembelajaran seimbang, maka setiap kelompok memiliki jumlah *bag* positif yang sama dengan jumlah *bag* negatif.



Gambar 9 Pembagian *Bag* Menjadi 10 Fold Cross Validation.

Dengan demikian dari 200 *bag* pada *data set* (TIGER, FOX dan ELEPHANT), setiap kelompok berisi 20 *bag* yang terdiri dari 10 *bag* positif dan 10 *bag* negatif. Pada Gambar 9 diilustrasikan pembagian *data set* menjadi 10 kelompok. Pada Gambar 9 (a) menunjukkan bentuk awal *data set* yang terdiri dari dua kategori, yakni *bag* positif dan *bag* negatif. Pada Gambar 9 (b)

mengilustrasikan pembentukan 10 kelompok data dan Gambar 9 (c) mengilustrasikan kelompok data yang berisi *bag* positif dan *bag* negatif dengan jumlah yang sama banyak.

III.4 Desain Metode Usulan

Dari permasalahan yang telah diuraikan di atas, dalam penelitian ini malakukan modifikasi algoritma MI-ELM yaitu dengan mengubah fungsi *gaussian similarity* dengan menggunakan *clustering k-means*, disebut *K-Means* MI-ELM. Alur kerja *K-Means* MI-ELM dapat dilihat pada Gambar 10. Berikut desain metode yang diusulkan.

a. Transformasi *Feature*

1) Ekstraksi *Feature*

Proses ekstraksi *feature* hitung nilai *output* pada *hidden layer* pertama χ' menggunakan persamaan (5). Di mana χ' adalah *new training set* berdimensi $N \times M$ dengan N adalah jumlah *instance* dan M adalah jumlah *hidden node* pada *hidden layer* pertama.

2) Pembentukan *Bag-level Feature*

Pada tahap ini akan melakukan transformasi χ' menjadi matrik G sebagai *output* pada *hidden layer* ke dua. Untuk mentransformasikan *feature* pada χ' adalah dengan cara menghitung jarak setiap *bag* terhadap nilai *centeroid* m di mana $m = m_1, \dots, m_k$ dan k adalah jumlah *cluster* yang telah ditentukan.

Jarak antara *bag* dengan *centeroid* diperoleh dengan menghitung nilai selisih antara $\chi_c^{(i)}$ dengan m , di mana adalah $\chi_c^{(i)}$ rata-rata *instance-level feature* pada *bag* ke- i . Berikut adalah langkah-langkah dalam melakukan *clustering* χ' untuk mendapatkan *bag-level feature*.

- a) Langkah awal pada tahap ini adalah menghitung $\chi_c^{(i)}$ dengan persamaan berikut.

$$\chi_c^{(i)} = \frac{1}{n_i} \left[\sum_{j=1}^{n_i} x'_{j1}, \dots, \sum_{j=1}^{n_i} x'_{jM} \right] \quad (14)$$

di mana $i=1, \dots, L$, n_i adalah jumlah *instance* yang ada pada *bag* ke- i dan $x'_{j\cdot}$ adalah *instance* ke- j pada *bag* ke- i .

- b) Langkah selanjutnya adalah membuat sejumlah k kelompok *cluster* S , di mana $S=S_1, \dots, S_k$ dan $k \leq L$. Kemudian mengalokasikan χ_c ke dalam kelompok *cluster* S secara sembarang, di mana $\chi_c = \chi_c^{(1)}, \dots, \chi_c^{(L)}$.
- c) Hitung nilai *centeroid* m awal yang diperoleh dengan menghitung rata-rata χ_c yang ada pada setiap *cluster* S .
- d) Kelompokan ulang χ_c ke dalam *cluster* S yang memiliki *centeroid* terdekat. Pengukuran jarak dilakukan dengan menggunakan *euclidean distance* seperti yang ditunjukkan pada persamaan (15).

$$d(\chi_c^{(i)}, m_j) = \sqrt{\sum_{n=1}^M (\chi_{c,n}^{(i)} - m_{j,n})^2} \quad (15)$$

di mana, $i=1, \dots, L$ dengan L adalah jumlah *bag* dan $j=1, \dots, k$ dengan k adalah jumlah *cluster*.

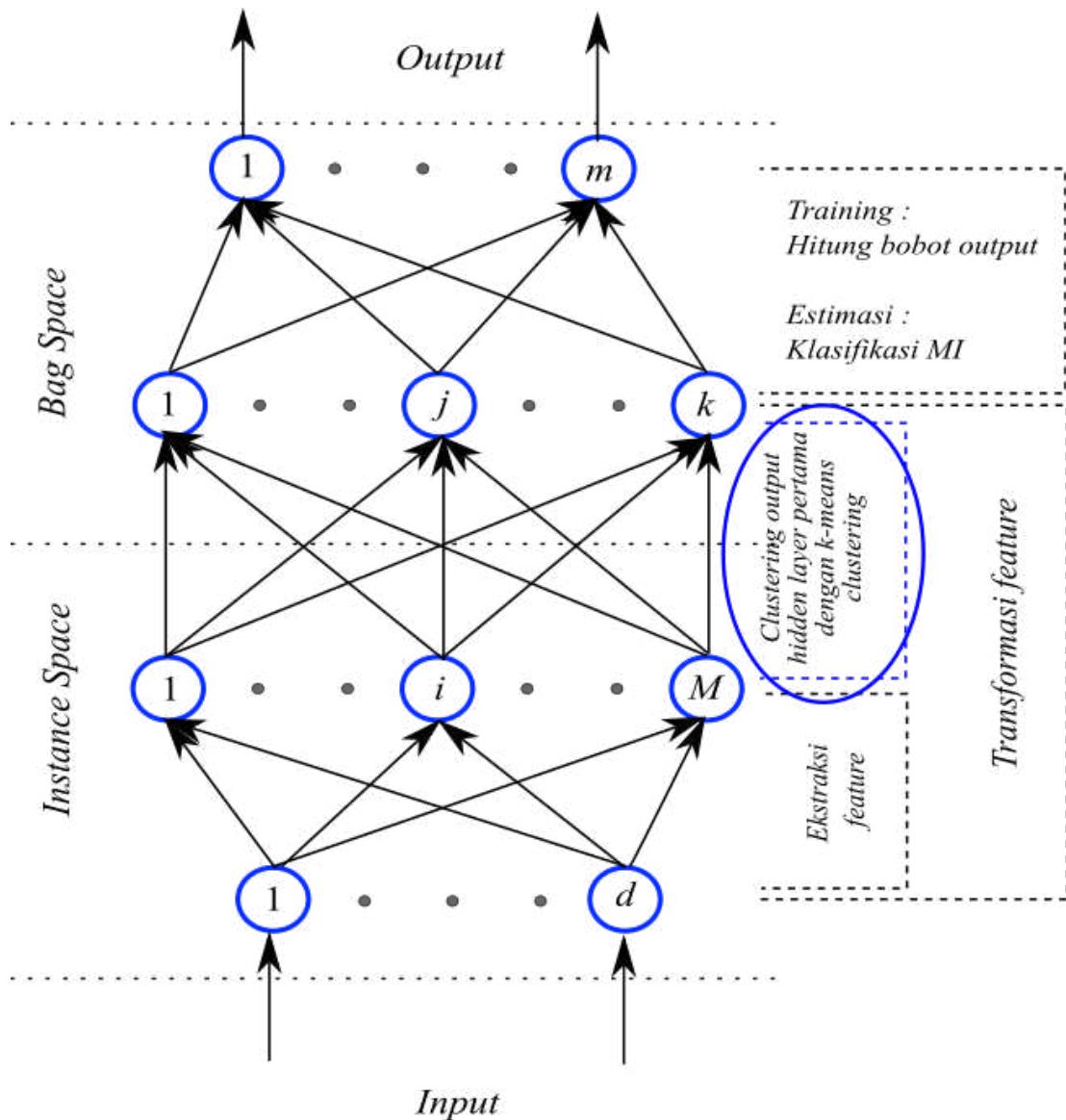
- e) Lakukan pembaruan nilai *centeroid* dengan menghitung nilai rata-rata χ_c pada setiap *cluster*.
- f) Ulangi langkah d) - e) hingga tercapai kondisi tidak ada lagi perpindahan data χ_c dari *cluster* satu ke *cluster* lainnya atau telah mencapai batas iterasi maksimal yang telah ditentukan.
- g) Setelah iterasi terhenti maka didapatkan *centeroid* yang optimum. Nilai *centeroid* inilah yang akan digunakan untuk melakukan pembentukan *bag-level feature*. Pembentukan *bag-level feature* ini adalah dengan cara menghitung jarak *euclidean distance* antara

χ_c dengan nilai *centeroid* m , berikut adalah persamaan untuk memperoleh *bag-level feature*.

$$G = [g(\chi'_1), \dots, g(\chi'_L)] \quad (16)$$

di mana $g(\chi'_i)$ adalah

$$g(\chi'_i) = [d(\chi_c^{(i)}, m_1), \dots, d(\chi_c^{(i)}, m_k)] \quad (17)$$



Gambar 10 Alur Kerja *K-Means MI-ELM*

b. Fase Klasifikasi *Bag-level Feature*

Untuk mengklasifikasi hal pertama yang dilakukan adalah proses pembelajaran model untuk mendapatkan bobot $output \beta$. Bobot $output \beta$ dapat diperoleh dengan menggunakan persamaan (12). Sedangkan untuk melakukan klasifikasi pada permasalahan MIC menggunakan persamaan (13).

III.5 Evaluasi Performa

Pada tahap ini, performa *K-Means* MI-ELM dibandingkan dengan MI-ELM. *K-Means* MI-ELM dan MI-ELM dibangun dengan menggunakan bahasa pemrograman Python versi 3.5.1. Pengujian dijalankan pada perangkat keras dengan prosesor Intel® Core™ i5-2520M CPU @ 2.50GHz × 4 dan RAM 4GB. *K-Means* MI-ELM dan MI-ELM diuji menggunakan *data set* kategorisasi citra (TIGER, FOX dan ELEPHANT). *K-Means* MI-ELM dan MI-ELM menggunakan jumlah *hidden node* (pada *hidden layer* pertama) yang ditentukan dalam penelitian ini, yaitu 1000 *hidden node*. Adapun kriteria penilaian performa dalam evaluasi ini adalah nilai akurasi dan *learning time* dari kedua metode tersebut. Formula

yang digunakan untuk menghitung nilai akurasi adalah $(1 - \frac{Error}{L_{test}}) \times 100$.

Untuk membandingkan performa *K-Means* MI-ELM dengan MI-ELM, dilakukan 30 kali percobaan dengan melakukan pengujian terhadap kedua metode tersebut pada empat kasus dengan jumlah *bag* yang berbeda-beda. Adapun jumlah *bag* pada setiap kasusnya masing-masing ditetapkan sebagai berikut : 100, 140, 180 dan 200. Kemudian dihitung rata-rata dan simpangan baku dari hasil 30 kali percobaan tersebut. Adapun hasil pada setiap percobaan adalah rata-rata akurasi dan *learning time* dari 10 *fold cross validation*.

BAB IV

HASIL DAN PEMBAHASAN

IV.1 Pra Proses

IV.1.1 Seleksi *Feature*

Melihat pada Tabel 1, bahwa *data set* TIGER, FOX dan ELEPHANT masing-masing memiliki jumlah *feature* yang sama, yakni 230 dengan 143 *nonzero-feature*. Dengan demikian 87 *zero-feature* yang terdapat pada ketiga *data set* tersebut akan dihapus. Selain itu pada ketiga *data set* tersebut masing-masing (TIGER, FOX dan ELEPHANT) terdapat 19, 31 dan 33 *feature* yang akan dihapus karena tidak representatif (*feature* bernilai homogen).

Pada akhirnya *feature* yang terpilih untuk menjadi parameter dalam melakukan pemodelan pada ketiga *data set* tersebut masing-masing berjumlah 124, 122, dan 110 *feature*. Contoh *data set* yang sudah terseleksi *feature*-nya untuk ketiga *data set* tersebut masing-masing ditunjukkan pada Lampiran 4, 5 dan 6.

IV.1.2 Normalisasi

Nilai *feature* dari ketiga *data set* (TIGER, FOX dan ELEPHANT) dinormalisasikan menggunakan fungsi *sigmoid bipolar*. Pada Lampiran 7, 8, dan 9 ditunjukkan hasil normalisasi pada ketiga *data set* tersebut dan dapat dilihat bahwa nilai *feaeture* sudah berada di dalam *range* -1 sampai 1.

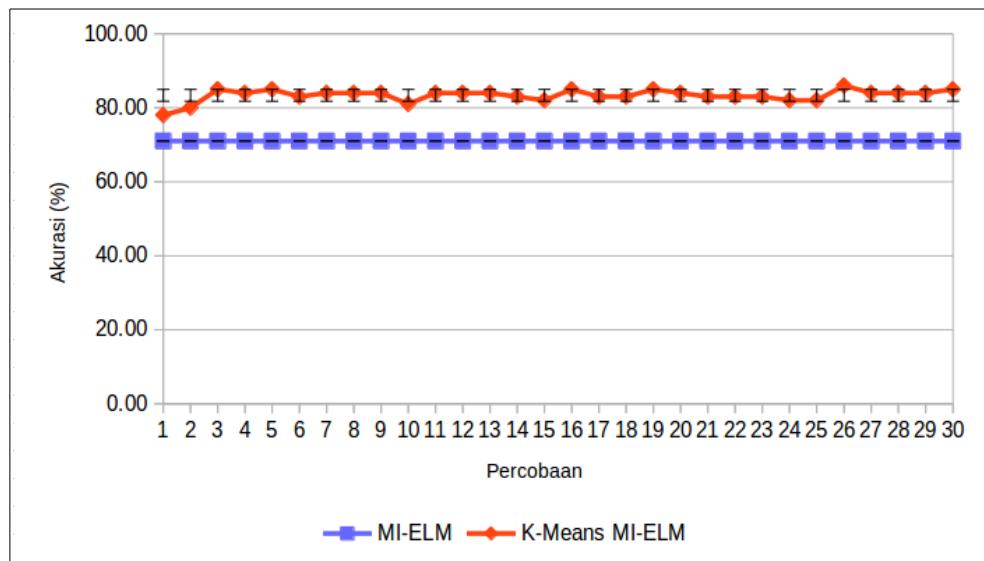
IV.2 Evaluasi Performa

Setelah melewati tahap pra proses, maka semua *data set* sudah siap digunakan untuk mengevaluasi serta membandingkan performa *K-Means* MI-ELM dengan MI-ELM. Pada kedua metode tersebut dilakukan 30 kali percobaan pada kasus jumlah *bag* yang berbeda-beda, yakni 100, 140, 180 dan 200. Pada *K*-

Means MI-ELM digunakan jumlah *cluster* sebanyak 26, dengan demikian akan terbentuk 26 *hidden node* pada *hidden layer* ke dua pada *K-Means* MI-ELM. Adapun pembahasan evaluasi performa yang telah dilakukan adalah sebagai berikut.

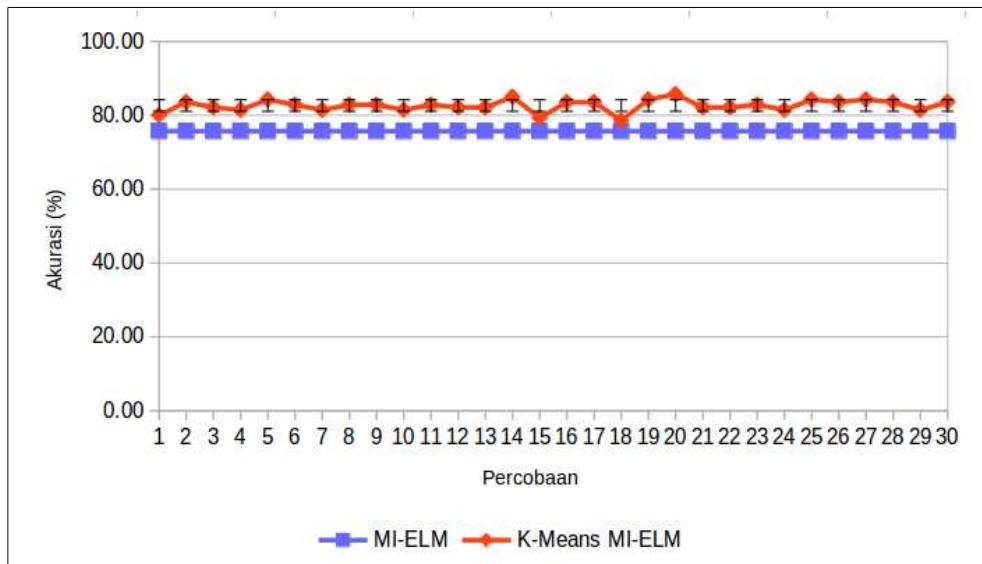
IV.2.1 Evaluasi Akurasi

Pada Gambar 11, 12, 13, dan 14 ditunjukkan grafik perbandingan akurasi yang diperoleh *K-Means* MI-ELM dan MI-ELM dengan jumlah *bag* masing-masing sebanyak 100, 140, 180, dan 200 pada *data set* TIGER. Pada percobaan dengan jumlah *bag* sebanyak 100 (Gambar 11) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 83.37% dengan simpangan baku sebesar 1.629. Sedangkan pada MI-ELM diperoleh rata-rata akurasi 71.00% dengan simpangan baku sebesar 0.000.

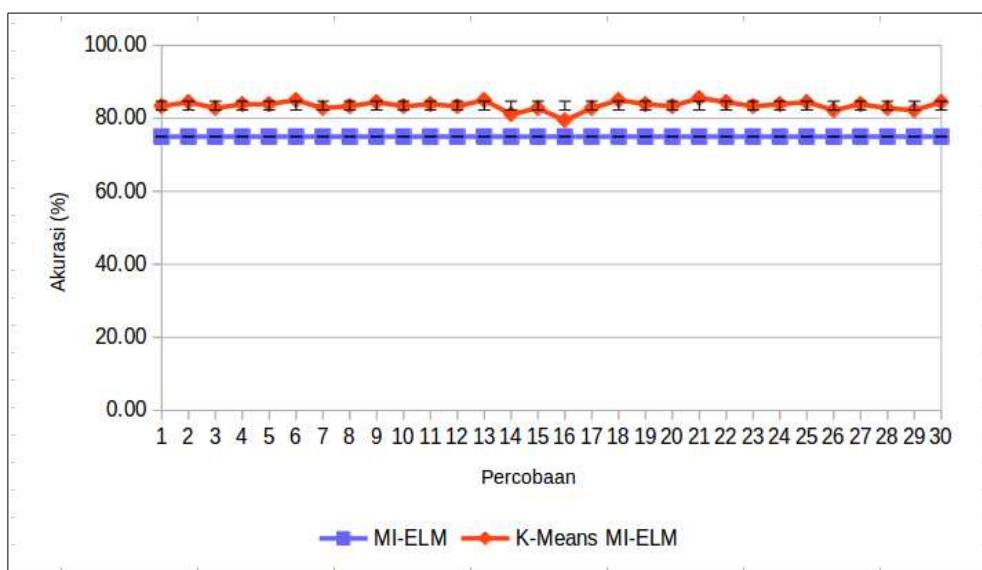


Gambar 11 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada *Data Set* TIGER

Pada percobaan dengan jumlah *bag* sebanyak 140 (Gambar 12) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 82.64% dengan simpangan baku sebesar 1.604. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 75.71% dengan simpangan baku sebesar 0.000.

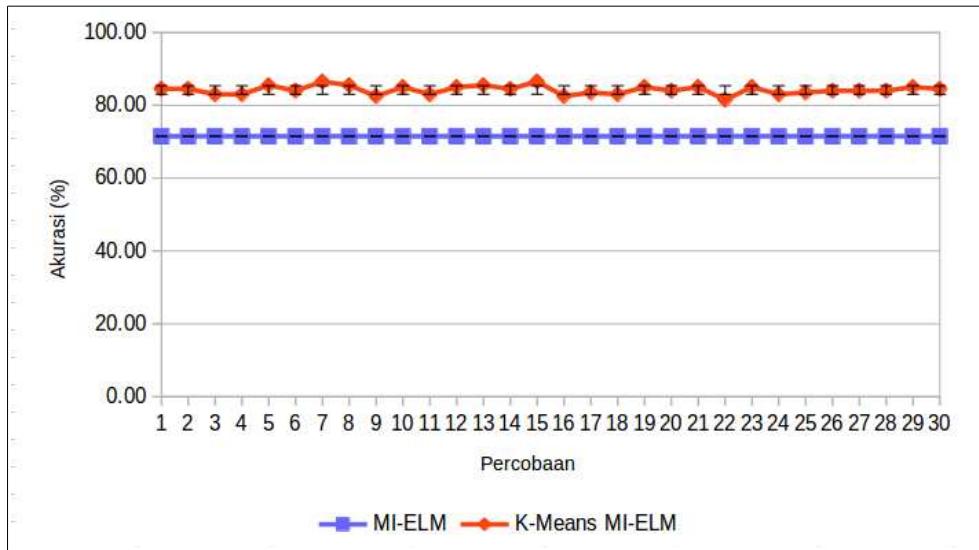


Gambar 12 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set TIGER



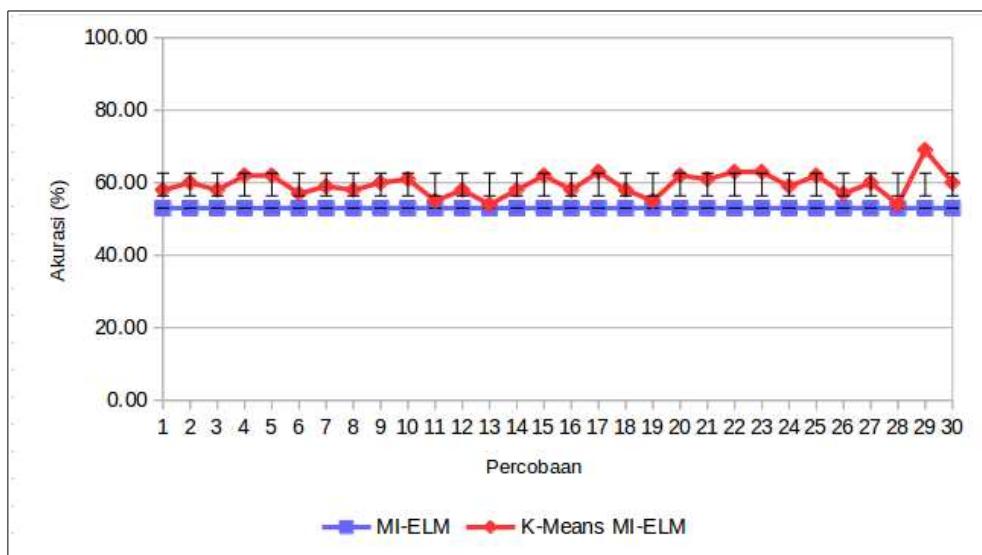
Gambar 13 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set TIGER

Pada percobaan dengan jumlah *bag* sebanyak 180 (Gambar 13) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 83.50% dengan simpangan baku sebesar 1.239. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 75.00% dengan simpangan baku sebesar 0.000.



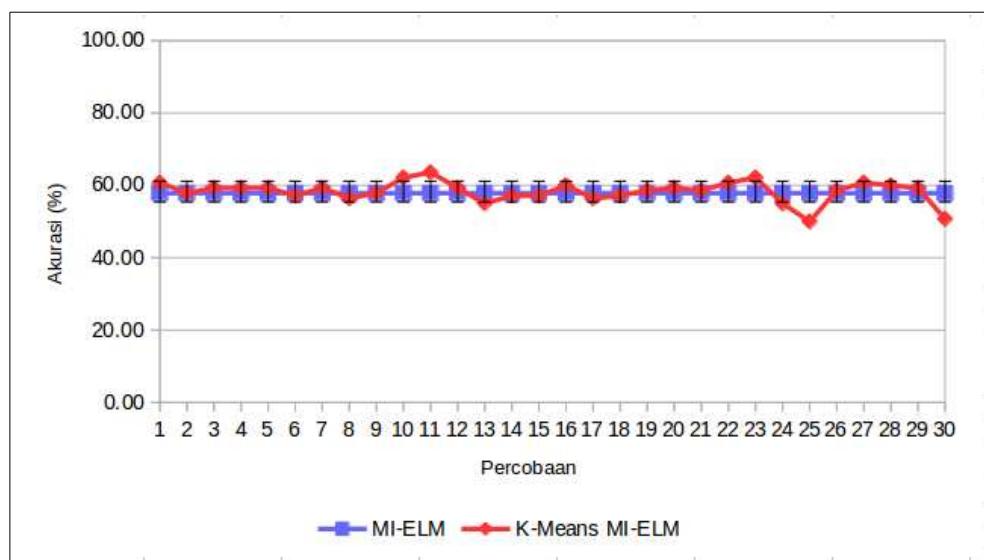
Gambar 14 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set TIGER

Sedangkan pada percobaan dengan jumlah *bag* sebanyak 200 (Gambar 14) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 84.20% dengan simpangan baku sebesar 1.193. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 71.50% dengan simpangan baku sebesar 0.000.



Gambar 15 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set FOX

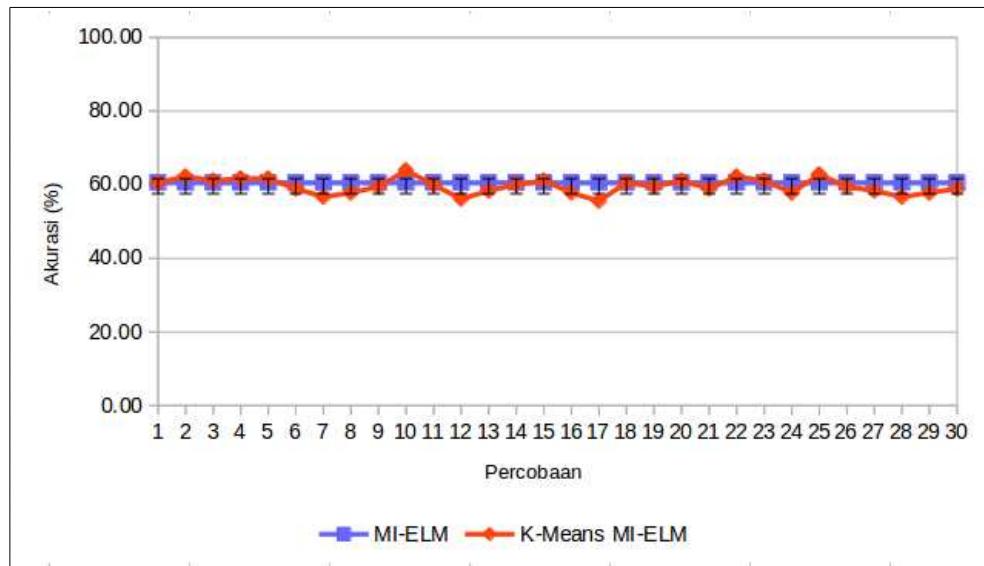
Untuk grafik perbandingan akurasi yang diperoleh *K-Means* MI-ELM dan MI-ELM pada *data set* FOX ditunjukkan pada Gambar 15, 16, 17 dan 18 dengan menggunakan jumlah *bag* masing-masing sebanyak 100, 140, 180 dan 200. Pada percobaan dengan jumlah *bag* sebanyak 100 (Gambar 15), rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 59.53% dengan simpangan baku sebesar 3.181. Sedangkan rata-rata yang diperoleh MI-ELM adalah 53.00% dengan simpangan baku sebesar 0.000.



Gambar 16 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada *Data Set* FOX

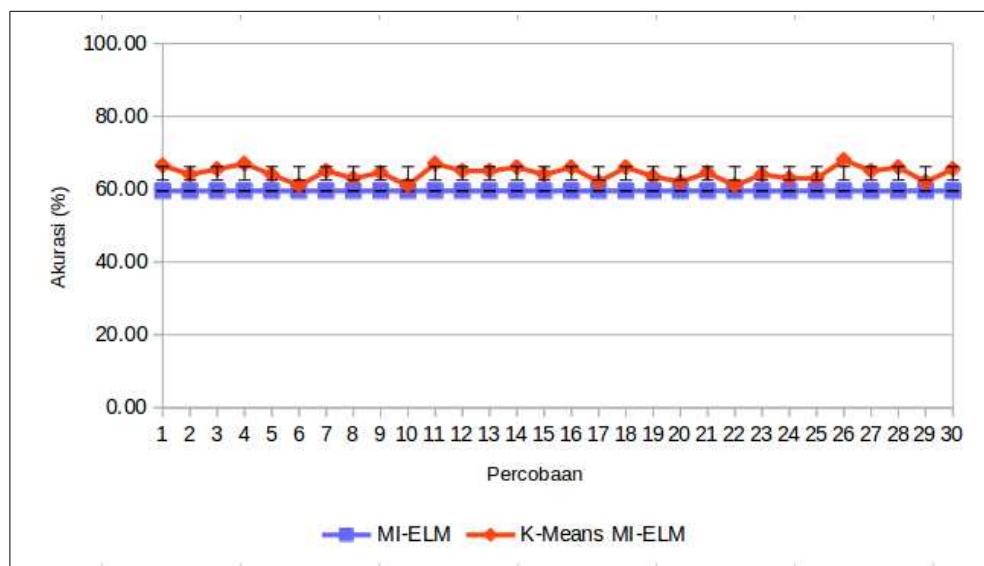
Pada percobaan dengan jumlah *bag* sebanyak 140 (Gambar 16) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 58.29% dengan simpangan baku sebesar 2.928. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 57.86% dengan simpangan baku sebesar 0.000.

Pada percobaan dengan jumlah *bag* sebanyak 180 (Gambar 17) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 59.59% dengan simpangan baku sebesar 2.083. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 60.56% dengan simpangan baku sebesar 0.000.



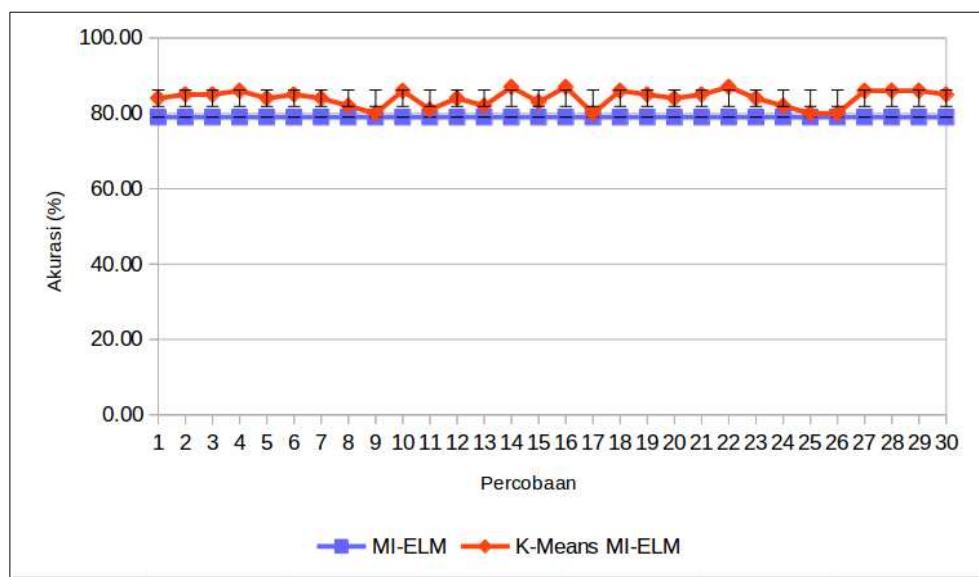
Gambar 17 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set FOX

Pada percobaan dengan menggunakan 200 bag (Gambar 18) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 64.33% dengan simpangan baku sebesar 1.895. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 59.50% dengan simpangan baku sebesar 0.000.



Gambar 18 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set FOX

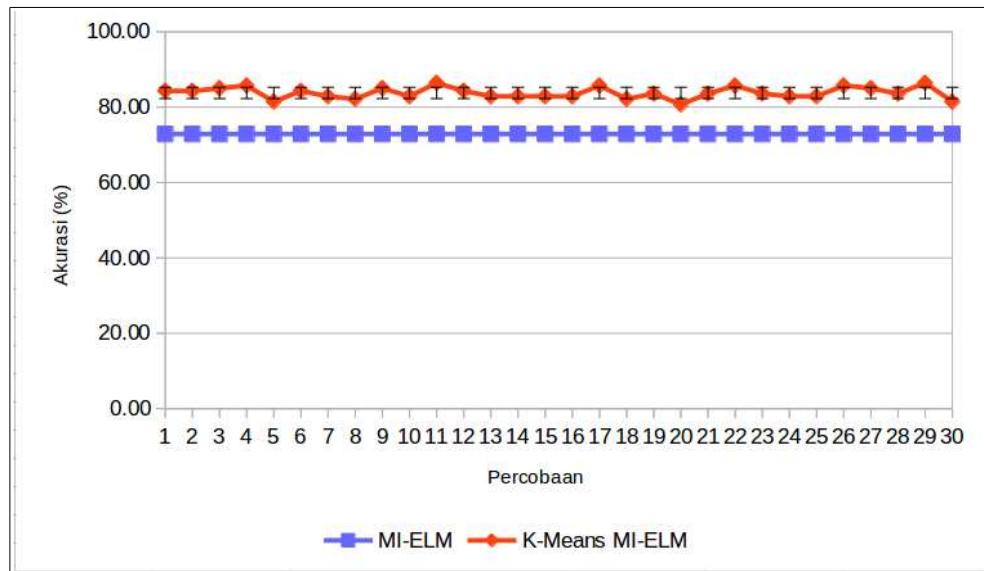
Grafik perbandingan akurasi yang diperoleh *K-Means* MI-ELM dan MI-ELM pada *data set* ELEPHANT ditunjukkan pada Gambar 19, 20, 21 dan 22 dengan menggunakan jumlah *bag* masing-masing sebanyak 100, 140, 180 dan 200. Pada percobaan dengan jumlah *bag* sebanyak 100 (Gambar 19), rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 84.03% dengan simpangan baku sebesar 2.220. Sedangkan rata-rata yang diperoleh MI-ELM adalah 79.00% dengan simpangan baku sebesar 0.000.



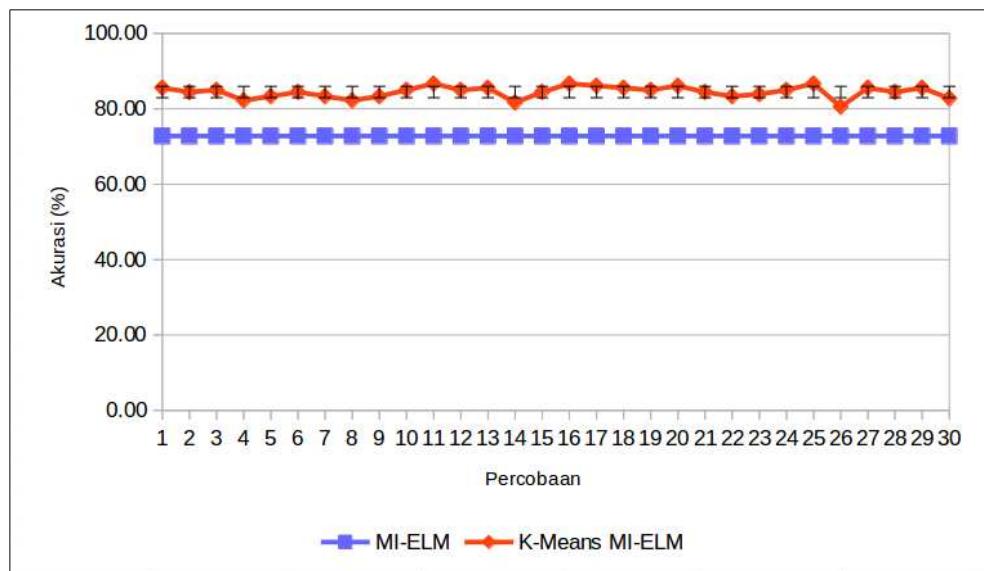
Gambar 19 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada *Data Set* ELEPHANT

Pada percobaan dengan jumlah *bag* sebanyak 140 (Gambar 20) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 83.76% dengan simpangan baku sebesar 1.523. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 72.86% dengan simpangan baku sebesar 0.000.

Pada percobaan dengan jumlah *bag* sebanyak 180 (Gambar 21) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 84.46% dengan simpangan baku sebesar 1.540. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 72.78% dengan simpangan baku sebesar 0.000.

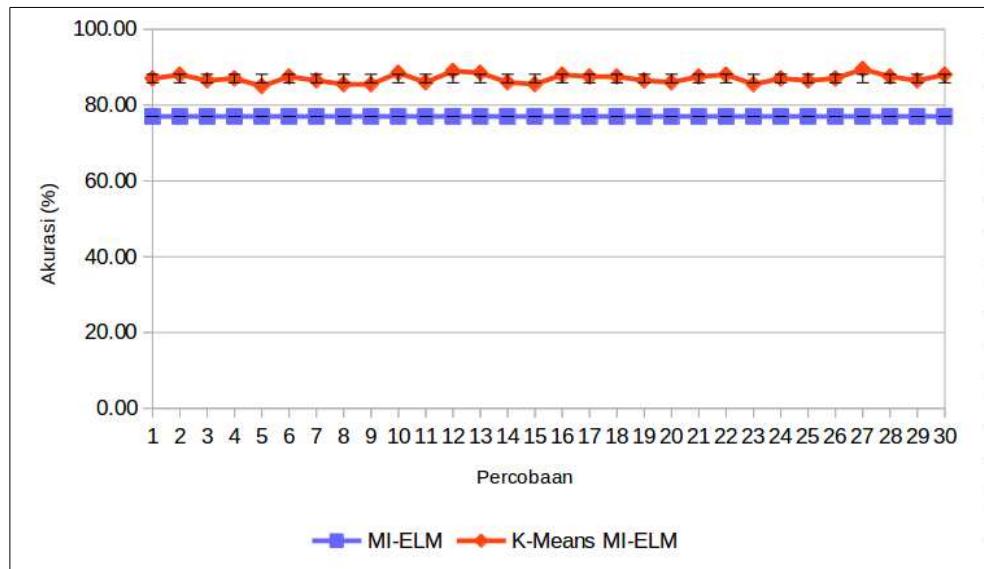


Gambar 20 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 140 pada *data set* ELEPHANT

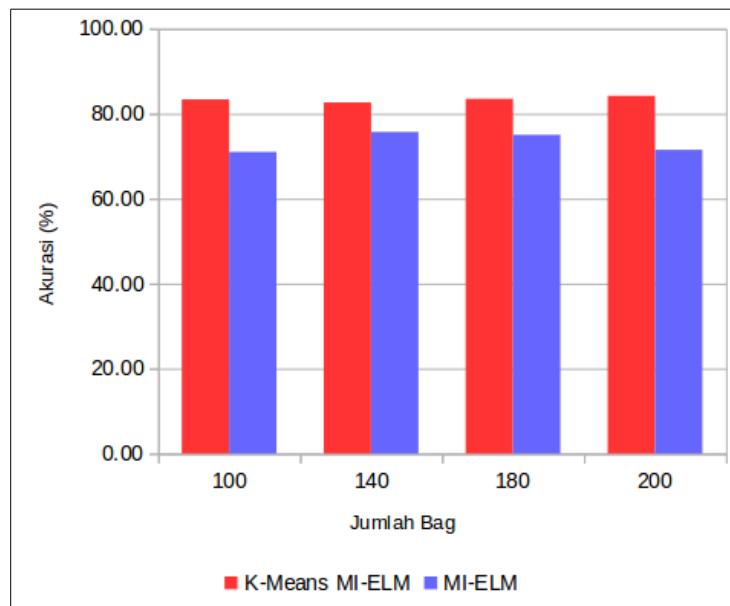


Gambar 21 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada *Data Set* ELEPHANT

Pada percobaan dengan jumlah *bag* sebanyak 200 (Gambar 22) rata-rata akurasi yang diperoleh *K-Means* MI-ELM adalah 87.02% dengan simpangan baku sebesar 1.133. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 77.00% dengan simpangan baku sebesar 0.000.



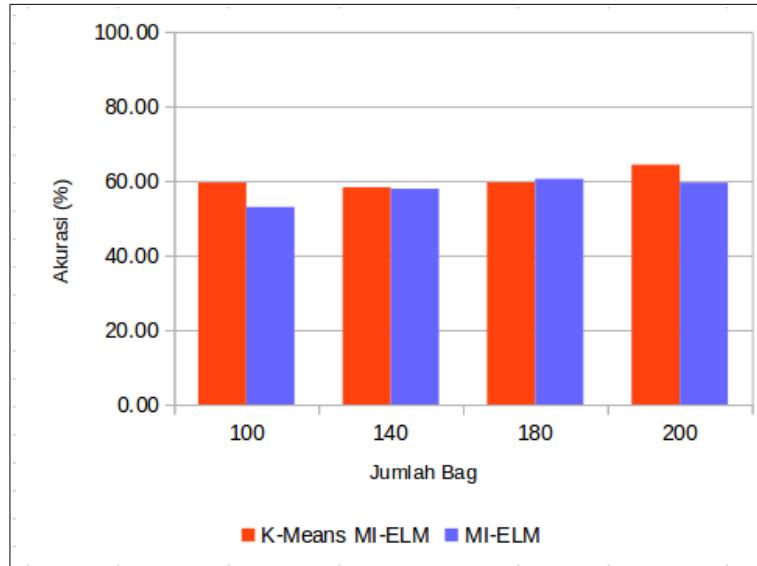
Gambar 22 Grafik Perbandingan Akurasi *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 200 pada *data set* ELEPHANT



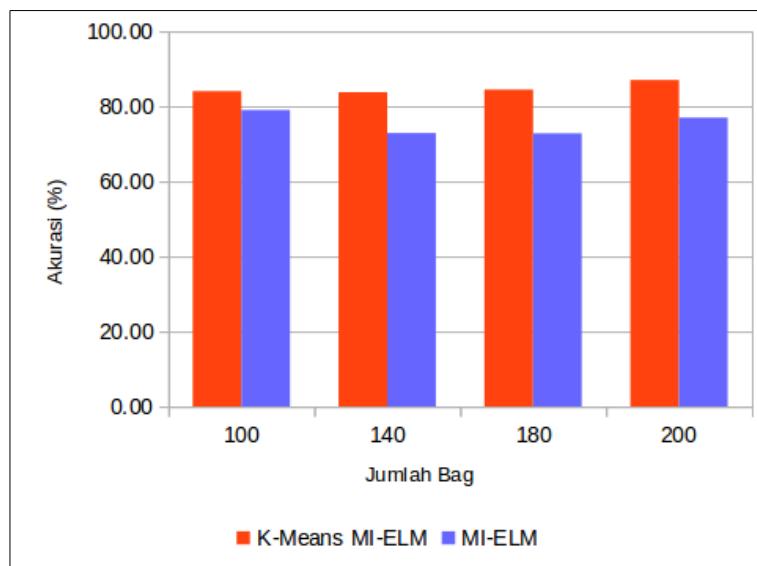
Gambar 23 Grafik Perbandingan Rata-Rata akurasi *K-Means* MI-ELM Dan MI-ELM Pada *Data Set* TIGER

Dari hasil percobaan yang telah diuraikan di atas dapat dilihat bahwa akurasi yang diperoleh MI-ELM bernilai konstan, ini menunjukan MI-ELM sangat stabil dibandingkan dengan *K-Means* MI-ELM. Namun, secara keseluruhan rata-rata akurasi yang diperoleh *K-Means* MI-ELM berada di atas

MI-ELM. Hal ini dapat dilihat pada Gambar 23, 24 dan 25 untuk grafik perbandingan rata-rata akurasi yang diperoleh *K-Means* MI-ELM dengan MI-ELM pada *data set* TIGER, FOX dan ELEPHANT.



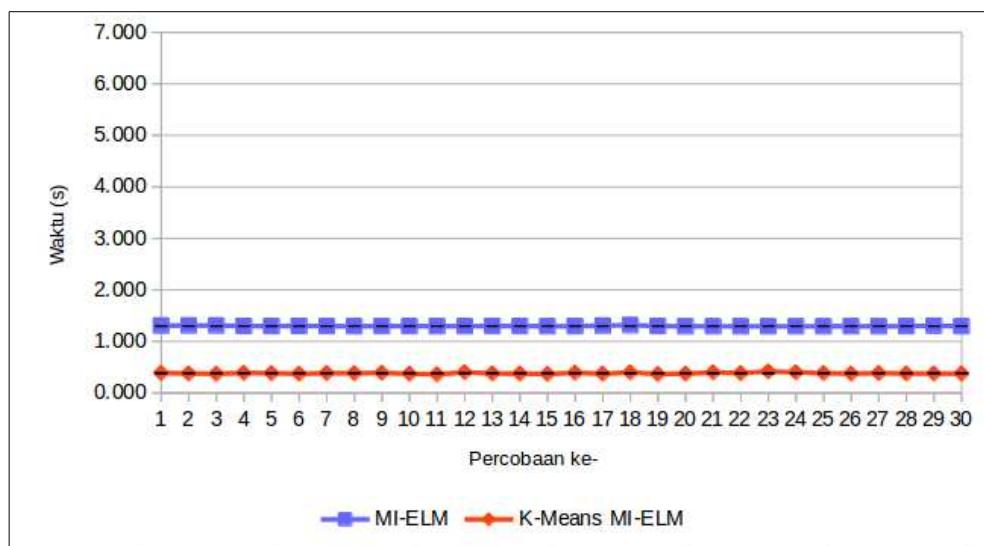
Gambar 24 Grafik Perbandingan Rata-Rata Akurasi *K-Means* MI-ELM dan MI-ELM Pada *Data Set* FOX



Gambar 25 Grafik Perbandingan Rata-Rata Akurasi *K-Means* MI-ELM dan MI-ELM Pada *Data Set* ELEPHANT

IV.2.2 Evaluasi *Learning Time*

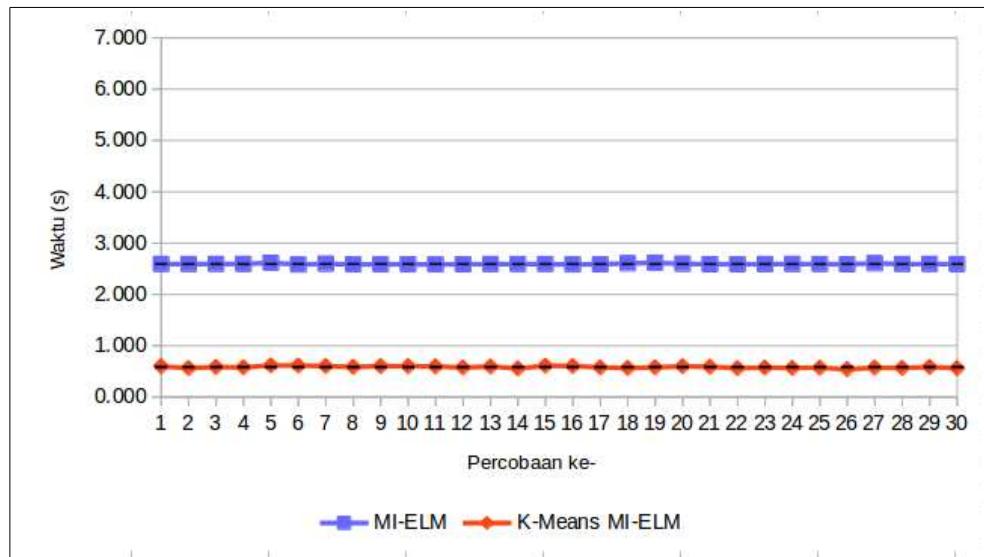
Grafik perbandingan *learning time* *K-Means* MI-ELM dan MI-ELM dari hasil percobaan pada *data set* TIGER dengan jumlah *bag* 100, 140, 180 dan 200 masing-masing ditunjukkan pada Gambar 26, 27, 28 dan 29. Pada percobaan dengan jumlah *bag* sebanyak 100 (Gambar 26), rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.383 detik dengan simpangan baku sebesar 0.012. Sedangkan rata-rata *learning time* yang diperoleh MI-ELM adalah 1.298 detik dengan simpangan baku sebesar 0.006.



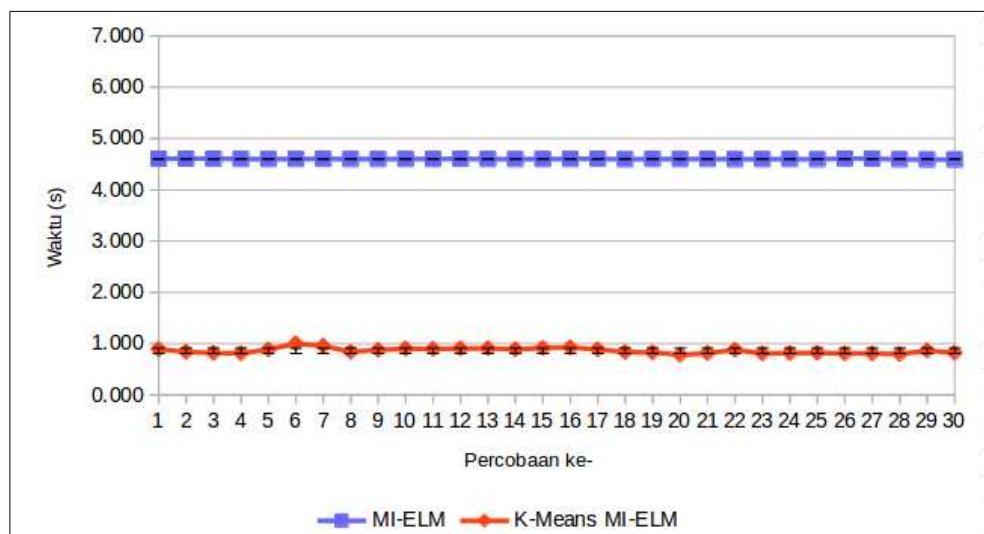
Gambar 26 Grafik Perbandingan *Learning Time* *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada *Data Set* TIGER

Pada percobaan dengan jumlah *bag* sebanyak 140 (Gambar 27) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.584 detik dengan simpangan baku sebesar 0.019. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 2.593 detik dengan simpangan baku sebesar 0.008.

Pada percobaan dengan jumlah *bag* sebanyak 180 (Gambar 28) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.861 detik dengan simpangan baku sebesar 0.053. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 4.600 detik dengan simpangan baku sebesar 0.005.

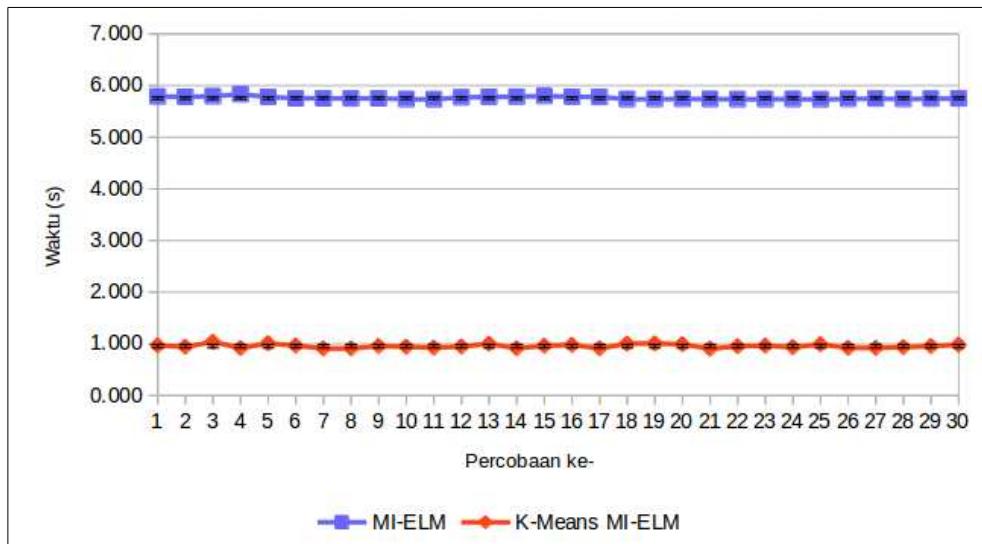


Gambar 27 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set TIGER

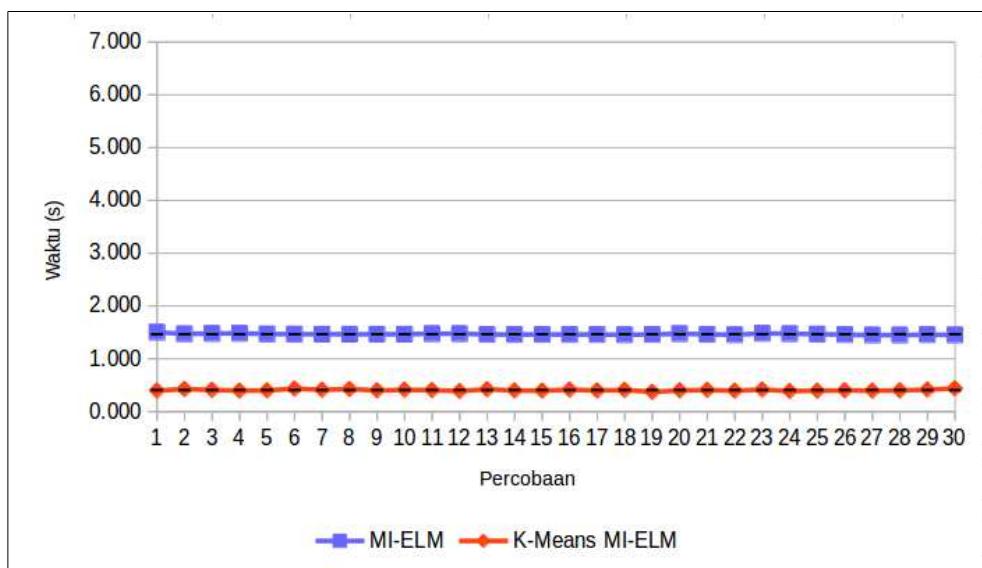


Gambar 28 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set TIGER

Pada percobaan dengan jumlah *bag* sebanyak 200 (Gambar 29) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.956 detik dengan simpangan baku sebesar 0.036. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 5.758 detik dengan simpangan baku sebesar 0.025.



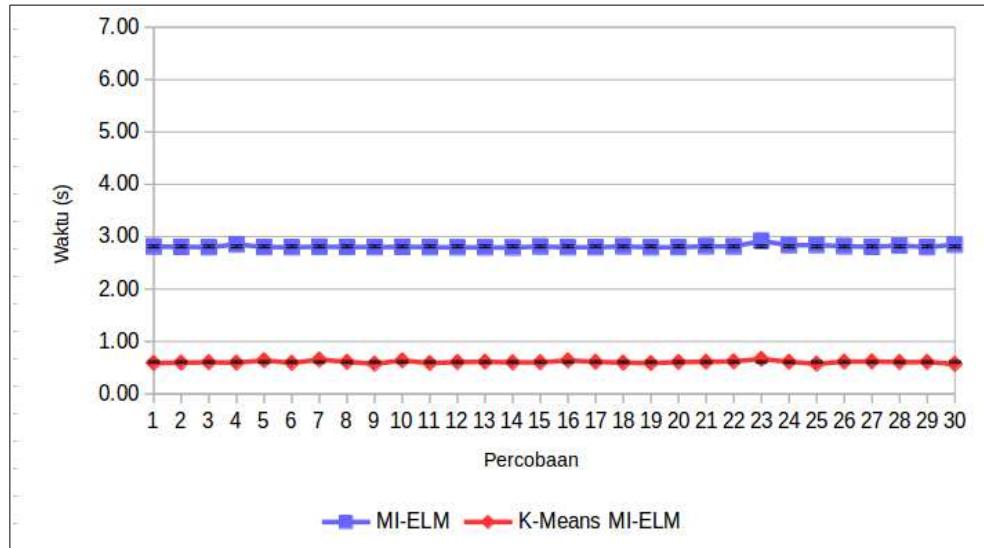
Gambar 29 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set TIGER



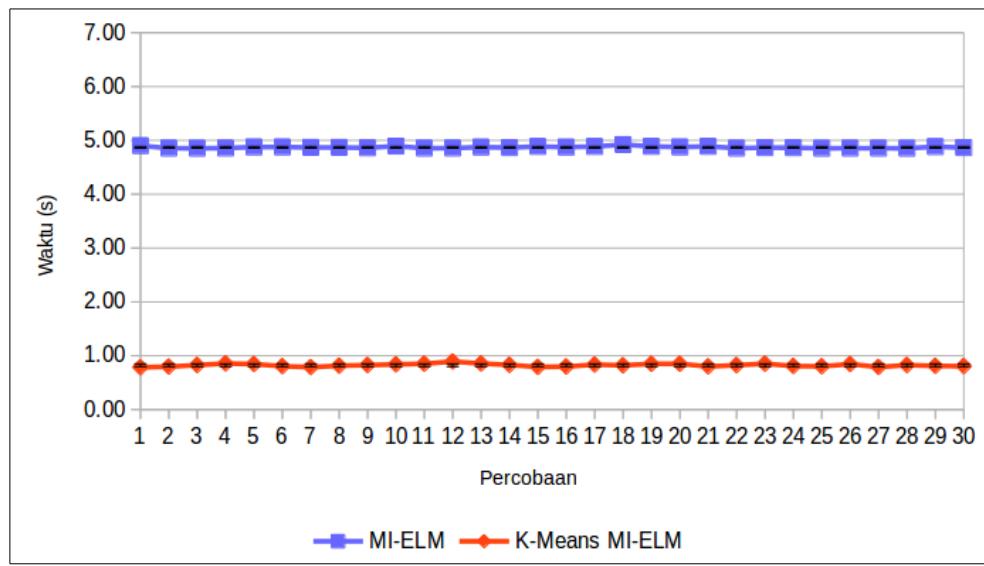
Gambar 30 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set FOX

Grafik perbandingan *learning time K-Means* MI-ELM dan MI-ELM dari hasil percobaan pada *data set FOX* dengan jumlah *bag* 100, 140, 180 dan 200 masing-masing ditunjukkan pada Gambar 30, 31, 32 dan 33. Pada percobaan dengan jumlah *bag* 100 (Gambar 30), rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.406 detik dengan simpangan baku sebesar 0.014.

Sedangkan rata-rata *learning time* yang diperoleh MI-ELM adalah 1.466 detik dengan simpangan baku sebesar 0.011.



Gambar 31 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set FOX

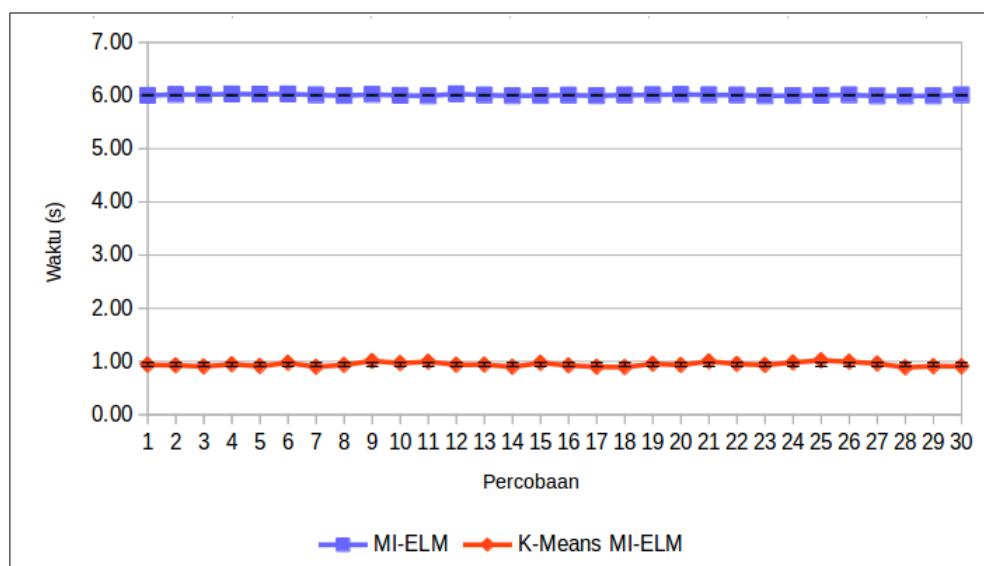


Gambar 32 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set FOX

Pada percobaan dengan jumlah *bag* sebanyak 140 (Gambar 31) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.608 detik dengan

simpangan baku sebesar 0.022. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 2.815 detik dengan simpangan baku sebesar 0.025.

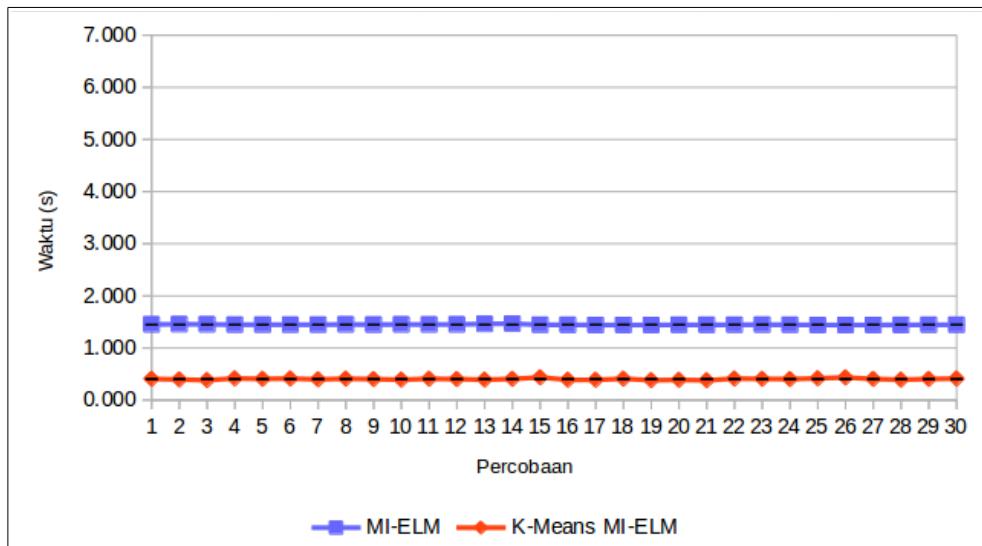
Pada percobaan dengan jumlah *bag* sebanyak 180 (Gambar 32) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.824 detik dengan simpangan baku sebesar 0.022. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 4.871 detik dengan simpangan baku sebesar 0.017.



Gambar 33 Grafik Perbandingan *Learning Time* *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set FOX

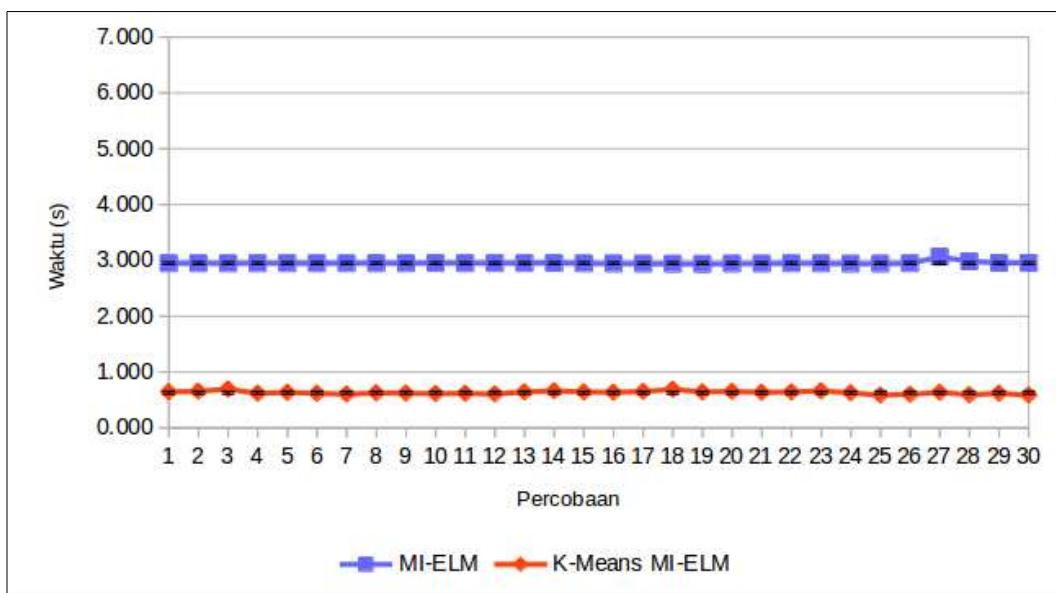
Pada percobaan dengan jumlah *bag* sebanyak 200 (Gambar 33) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.944 detik dengan simpangan baku sebesar 0.037. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 6.009 detik dengan simpangan baku sebesar 0.012.

Sedangkan grafik perbandingan *learning time* *K-Means* MI-ELM dan MI-ELM dari hasil percobaan pada *data set* ELEPHANT dengan jumlah *bag* 100, 140, 180 dan 200 masing-masing ditunjukkan pada Gambar 34, 35, 36 dan 37. Pada percobaan dengan jumlah *bag* 100 (Gambar 34), rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.404 detik dengan simpangan baku sebesar 0.014. Sedangkan rata-rata *learning time* yang diperoleh MI-ELM adalah 1.447 detik dengan simpangan baku sebesar 0.006.



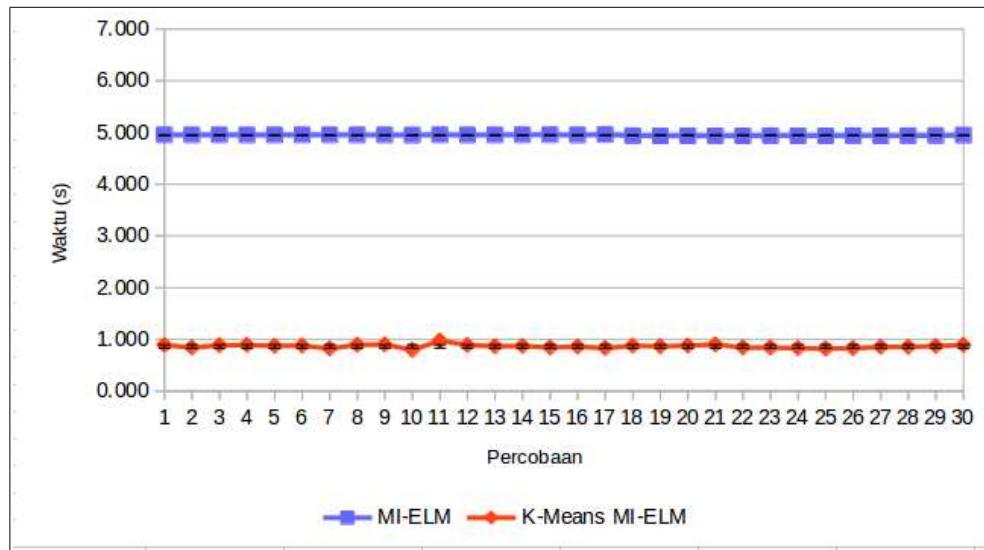
Gambar 34 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 100 Pada Data Set ELEPHANT

Pada percobaan dengan jumlah *bag* sebanyak 140 (Gambar 35) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.631 detik dengan simpangan baku sebesar 0.027. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 2.950 detik dengan simpangan baku sebesar 0.023.

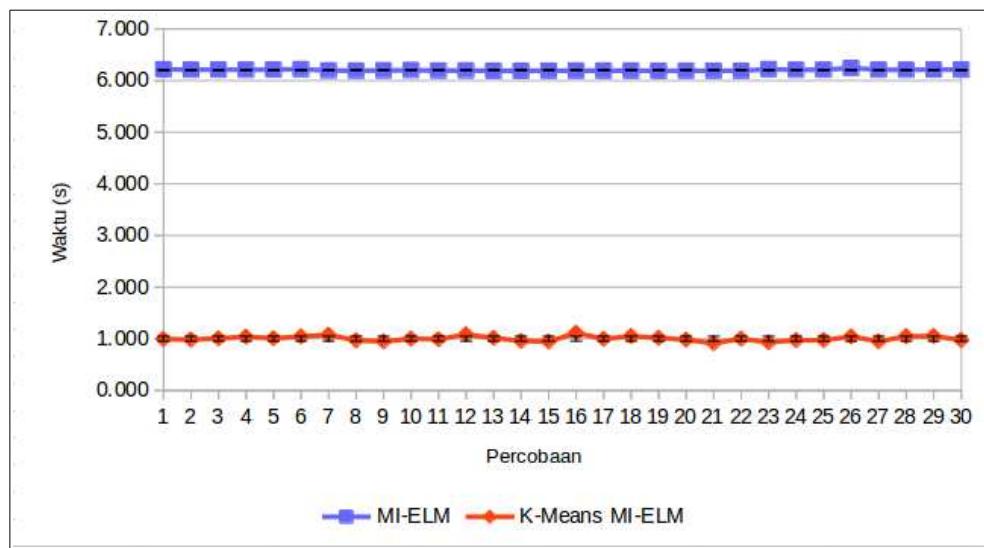


Gambar 35 Grafik Perbandingan *Learning Time K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 140 Pada Data Set ELEPHANT

Pada percobaan dengan jumlah *bag* sebanyak 180 (Gambar 36) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 0.866 detik dengan simpangan baku sebesar 0.034. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 4.945 detik dengan simpangan baku sebesar 0.008.



Gambar 36 Grafik Perbandingan *Learning Time* *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 180 Pada Data Set ELEPHANT

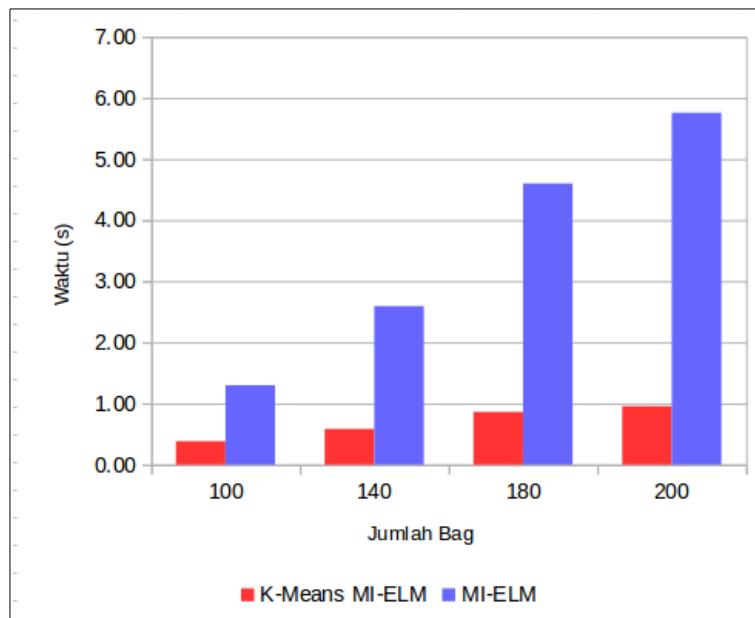


Gambar 37 Grafik Perbandingan *Learning Time* *K-Means* MI-ELM dan MI-ELM Dengan Jumlah Bag 200 Pada Data Set ELEPHANT

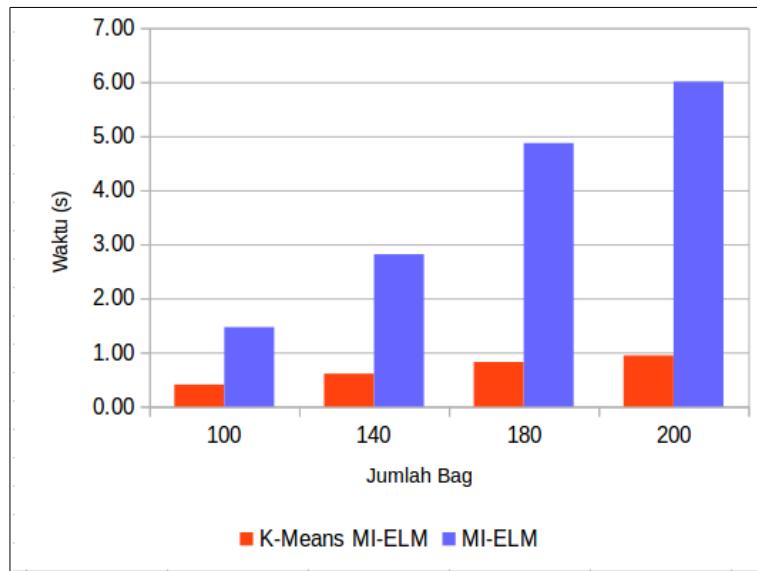
Pada percobaan dengan jumlah *bag* sebanyak 200 (Gambar 37) rata-rata *learning time* yang diperoleh *K-Means* MI-ELM adalah 1.003 detik dengan simpangan baku sebesar 0.048. Sedangkan rata-rata akurasi yang diperoleh MI-ELM adalah 6.203 detik dengan simpangan baku sebesar 0.013.

Dari hasil percobaan yang telah diuraikan di atas dapat dilihat bahwa simpangan baku *learning time* yang diperoleh MI-ELM lebih kecil dibandingkan *K-Means* MI-ELM. Namun simpangan baku yang diperoleh kedua metode tersebut sangat kecil. Dengan kata lain, perolehan *learning time* dari kedua metode tersebut relatif stabil.

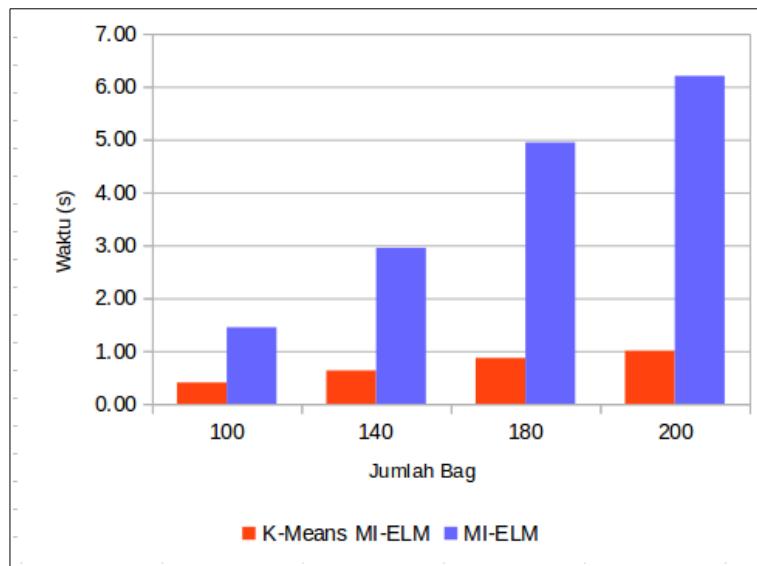
Sedangkan rata-rata *learning time* yang diperoleh MI-ELM mengalami peningkatan yang signifikan dibandingkan dengan rata-rata *learning time* yang diperoleh *K-Means* MI-ELM seiring dengan semakin banyaknya jumlah *bag* yang dipelajari. Grafik perbandingan rata-rata *learning time* dari *K-Means* MI-ELM dan MI-ELM pada *data set* TIGER, FOX dan ELEPHANT masing-masing ditunjukkan pada Gambar 38, 39 dan 40.



Gambar 38 Grafik Perbandingan Rata-Rata *Learning Time* *K-Means* MI-ELM dan MI-ELM Pada *Data Set* TIGER



Gambar 39 Grafik Perbandingan Rata-Rata *Learning Time* K-Means MI-ELM dan MI-ELM Pada *Data Set FOX*



Gambar 40 Grafik Perbandingan Rata-Rata *Learning Time* K-Means MI-ELM dan MI-ELM Pada *Data Set ELEPHANT*

BAB V

PENUTUP

V.1 Kesimpulan

Kesimpulan yang dapat ditarik dari pembahasan dalam penelitian ini adalah sebagai berikut :

- a. Untuk menerapkan algoritma *k-means* pada proses transformasi *feature* di algoritma MI-ELM hal pertama yang dilakukan adalah melakukan *clustering* terhadap nilai rata-rata *instance-level feature* menggunakan algoritma *k-means*. Setelah memperoleh nilai *centeroid* yang optimum, untuk mendapatkan *bag-level feature* adalah dengan menghitung jarak antara nilai rata-rata *instance-level feature* dengan nilai *centeroid* tersebut.
- b. Dengan menerapkan algoritma *k-means clustering* pada MI-ELM, jumlah *hidden node* pada *hidden layer* ke dua menjadi dapat ditentukan sebanyak *k node* dengan $k \leq$ jumlah *bag*.
- c. *K-Menas* MI-ELM lebih cepat 2 sampai 6 kali dibandingkan MI-ELM.
- d. Penerapan algoritma *k-means clustering* pada MI-ELM juga dapat menghasilkan nilai akurasi yang lebih baik dibandingkan nilai akurasi MI-ELM.

V.2 Saran

Dalam penelitian ini masih terdapat banyak kekurangan, oleh karena itu diperlukan saran untuk memperbaikinya. Berikut beberapa saran untuk penelitian berikutnya yang dapat dikemukakan:

- a. Lakukan penelitian untuk menerapkan *clustering* berbasis *Nonnegative Matrix Factorization* (NMF) pada *K-Means* MI-ELM.

- b. Lakukan penelitian ini untuk mengembangkan *K-Means* MI-ELM agar dapat melakukan pembelajaran secara *online sequence learning*.

DAFTAR PUSTAKA

- Amores, J 2013 ‘*Multiple Instance Classification: review, taxonomy and comparative study*’, *Artificial Intelligent*, 201, 81-105.
- Dietterich, TG, Lathrop, R. H., Tomás & Lozano-Pérez 1997 ‘*Solving the multiple instance problem with axis-parallel rectangles*’, *Artificial Intelligent* 89, pp. 31-71.
- Fu, Z, Lu, G, Ting, KM, Zhang, D 2013 ‘*Learning sparse kernel classifiers for multi-instance classification*’, *IEEE Trans. Neural Netw. Learn. Syst.* 24, pp. 1377–1389
- Gao, H, Guang-Bin, H, Shiji, S, Keyou, Y 2015, ‘*Trends in extreme learning machines: A review*’, *Neural Networks* 61, pp. 32-48
- Guang.-Bin, H, Qin-Yu, Z, Chee-Kheong, S 2006, ‘*Extereme learning machine: theory and application*’, *Neurocomputing* 70, pp. 489-501.
- Grossman, R, Seni, G, Joh, Agrwal, N, Huan 2010 ‘*Ensambel Methods in Data Mining: Improving Accuracy Through Combining Predictions*’. Morgan & Claypool.
- Nan-Ying, L, Guang-Bin, H, P. Saratchandran, N. Sundararajan 2006, ‘*A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks*’, *IEEE TRANSACTIONS ON NEURAL NETWORKS*, no. 6, pp.1411-1432.
- Qing H, Xin, J, Changying, D, Fuzhen, Z, Zhongzhi, S 2013, ‘*Clustering in extreme learning machine feature space*’, *Neurocomputing*, no.128, pp.88-95.
- Qiang, L, Sihang, Z, Chengzhang, Z, Xnwang, L, Jianping, Y 2016, ‘*MI-ELM: Highly efficient multi-instance learning based on hierarchical extreme learning machine*’, *Neurocomputing*, no.173, pp.1044-1053
- Maron, O & Ratan, AL 2001, ‘*Multiple-instance learning for natural scene classification*’, *Proceedings of 18th international conference on machine learning*. Williamstown, MA, pp. 425-432.
- McLachlan, GJ, Do, Kim-Anh; Ambroise, Christophe 2004, ‘*Analyzing microarray gene expression data*’. Wiley

McLachlan, GJ and Peel, D 2000, '*Finite Mixture Models*'. New York: Jhon Wiley and Sons.

Stuart, A, Tsochantaridis, I, Hofmann, T 2003, 'Support Vector Machine for Multiple-Instance Learning', *Adv. Newural Inf. Process. Syst.* 15, pp. 561-568

Xiu-Shen, W, & Zhi-Hua, Z 2016, 'An empirical study on image bag generators for multi-instance learning', *Machine Learning*. <http://www/researchgate.net/publication/299375067>

Yixin, C, Jinbo, B, dan James, ZW 2006, 'MILES: Multiple-Instance Learning Embedded Instance Selection', *IEEE Transactions on Pattern Analysis and Machine Learning* no. 12, pp. 1931 - 1947

DAFTAR RIWAYAT HIDUP



Nama	:	Rachmat Prabowo
Tempat/Tanggal Lahir	:	Jakarta, 22 Mei 1994
Jenis Kelamin	:	Laki-laki
Agama	:	Islam
No Telp	:	08963-824-7577
Email	:	praboworachmat@gmail.com

Nama Orang Tua

Ayah	:	Wardani
Ibu	:	Sarminah

PENDIDIKAN FORMAL

1. SDN Jombang 58, Ciputat (2000-2006)
2. SMPN 03 Ciputat, Ciputat (2006-2009)
3. SMK Bina Informatika, Bintaro (2009-2012)
4. Universitas Pembangunan Nasional “Veteran” Jakarta Program Studi S1-Teknik Informatika (2012-2016)

PENGALAMAN ORGANISASI

1. OSIS SMK Bina Informatika (2008-2009)
2. Robotika UPN ‘Veteran’ Jakarta (2012-2016)
3. LDK Uswah UPN ‘Veteran’ Jakarta (2015-2016)

LAMPIRAN

Lampiran 1
Tabel Contoh Data Set TIGER

Id Bag	X₁	X₂	...	X₂₂₀	X₂₂₁	X₂₂₂	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	X₂₂₇	X₂₂₈	X₂₂₉	X₂₃₀	Label	
1	1.613010	1.266250	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	0.062758	-0.172335	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	1.022390	1.828530	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-0.066495	0.127760	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-1.254660	-1.189560	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	2.048050	2.038780	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.769024	1.433180	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-1.097910	-0.973971	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-1.280120	-1.183950	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-0.573995	-0.559183	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
199	-0.777438	-0.755225	...	0.000000	0.000000	0.000000	-0.049855	0.302234	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
199	1.035890	1.487280	...	0.000000	0.000000	0.000000	-0.049855	0.370473	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
199	-1.265890	-1.180670	...	0.000000	0.000000	0.000000	-0.049855	-0.111583	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
199	-1.137790	-1.008810	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
199	-0.578800	-0.440757	...	0.000000	0.000000	0.000000	-0.049855	4.313510	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
199	-0.796740	-0.538850	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
199	-1.172550	-1.055310	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.442055	-0.335707	...	0.000000	0.000000	0.000000	-0.049855	0.243362	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.052340	0.140107	...	0.000000	0.000000	0.000000	-0.049855	-0.065255	0.007736	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.760124	0.645445	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.238350	0.129306	...	0.000000	0.000000	0.000000	-0.049855	-0.050709	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.232650	1.205280	...	0.000000	0.000000	0.000000	-0.049855	2.874990	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.840869	0.483899	...	0.000000	0.000000	0.000000	-0.049855	8.066390	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1

Lampiran 2
Tabel Contoh Data Set FOX

Id Bag	X₁	X₂	...	X₂₂₂	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	X₂₂₇	X₂₂₈	X₂₂₉	X₂₃₀	Label	
1	-1.313750	-1.227930	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	0.100567	0.181302	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.260590	-1.142400	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.313120	-1.223590	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.274250	-1.163020	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.109940	-0.993377	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-0.191282	-0.236270	...	0.000000	-0.049855	12.012900	0.297029	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	1.681580	1.781010	...	0.000000	-0.049855	6.780190	4.047530	7.672440	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.367654	0.476738	...	0.000000	-0.049855	6.778390	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.784899	1.306680	...	0.000000	-0.049855	9.990350	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	1.879210	2.394610	...	0.000000	-0.049855	5.896720	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.926793	1.704860	...	0.000000	-0.049855	20.439100	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.904933	0.875859	...	0.000000	-0.049855	4.430750	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	2.120340	2.411620	...	0.000000	-0.049855	5.411950	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
200	0.442055	-0.335707	...	0.000000	-0.049855	0.243362	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.052340	0.140107	...	0.000000	-0.049855	-0.065255	0.007736	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.760124	0.645445	...	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.238350	0.129306	...	0.000000	-0.049855	-0.050709	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.232650	1.205280	...	0.000000	-0.049855	2.874990	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.840869	0.483899	...	0.000000	-0.049855	8.066390	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1

Lampiran 3
Tabel Contoh Data Set ELEPHANT

Id Bag	X₁	X₂	...	X₂₂₀	X₂₂₁	X₂₂₂	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	X₂₂₇	X₂₂₈	X₂₂₉	X₂₃₀	Label	
1	2.057730	1.629760	...	0.000000	0.000000	0.000000	-0.049855	-0.099670	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	0.842025	0.453111	...	0.000000	0.000000	0.000000	-0.049855	-0.087956	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-0.077775	-0.516443	...	0.000000	0.000000	0.000000	-0.049855	0.249723	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.313330	-1.239750	...	0.000000	0.000000	0.000000	0.281890	0.086480	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.221680	-1.160640	...	0.000000	0.000000	0.000000	-0.049855	-0.057620	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.108920	-1.134300	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
1	-1.140160	-1.049910	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-0.801718	-0.728592	...	0.000000	0.000000	0.000000	-0.049855	-0.004255	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.729744	0.077972	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	1.293230	1.728680	...	0.000000	0.000000	0.000000	-0.049855	0.139799	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.193597	-0.229656	...	0.000000	0.000000	0.000000	-0.049855	3.767520	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-0.496522	-0.409700	...	0.000000	0.000000	0.000000	-0.049855	0.364099	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	-0.997742	-0.907455	...	0.000000	0.000000	0.000000	-0.049855	1.198680	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	0.534048	0.404399	...	0.000000	0.000000	0.000000	-0.049855	0.209850	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
2	1.325110	2.052350	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
200	0.301678	-0.325895	...	0.000000	0.000000	0.000000	-0.049855	-0.050502	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.922838	0.825981	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.676150	0.714680	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.807393	0.387954	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.261180	0.346004	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	1.441560	0.815060	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1
200	0.119711	-0.218966	...	0.000000	0.000000	0.000000	-0.049855	-0.114025	-0.078862	-0.021452	0.000000	0.000000	-0.014952	-0.021097	-1	1

Lampiran 4

Tabel Hasil Pra Proses (Seleski Feature) Pada Data Set TIGER

Id Bag	X₁	X₂	X₃	X₄	X₅	...	X₂₁₆	X₂₁₇	X₂₁₈	X₂₁₉	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	Label
1	1.613010	1.266250	0.522618	-0.485324	2.211700	...	-0.640696	-0.347397	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
1	0.062758	-0.172335	-0.998067	0.470306	0.352271	...	-0.745974	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
1	1.022390	1.828530	0.383356	0.244013	0.450037	...	-0.742024	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
1	-0.066495	0.127760	1.676100	-1.166280	-0.732014	...	-0.404388	-0.333949	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
2	-1.254660	-1.189560	-0.728301	2.060280	1.632570	...	-0.549021	-0.345993	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
2	2.048050	2.038780	-0.504011	0.514204	1.047370	...	-0.481618	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
2	0.769024	1.433180	0.171997	-0.239974	-0.300419	...	-0.717149	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
2	-1.097910	-0.973971	-1.419130	-0.759106	0.057857	...	-0.746243	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
2	-1.280120	-1.183950	-0.011163	-1.444390	-0.421758	...	-0.746243	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
2	-0.573995	-0.559183	-1.074960	3.032120	-0.837324	...	-0.593662	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1 -1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
199	-0.777438	-0.755225	-0.831155	-0.795944	1.242210	...	0.847917	0.753691	-0.102856	-0.017565	-0.049855	0.302234	-0.078862	-0.021452	-1 1
199	1.035890	1.487280	-0.459538	0.797610	1.039830	...	-0.178279	0.879681	-0.102856	-0.017565	-0.049855	0.370473	-0.078862	-0.021452	-1 1
199	-1.265890	-1.180670	-0.929677	2.367050	0.291485	...	0.693411	1.109350	-0.102856	-0.017565	-0.049855	-0.111583	-0.078862	-0.021452	-1 1
199	-1.137790	-1.008810	0.423986	1.738550	-0.606326	...	-0.064820	1.165670	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1 1
199	-0.578800	-0.440757	0.081456	0.191322	-0.670229	...	-0.319383	0.292166	-0.102856	-0.017565	-0.049855	4.313510	-0.078862	-0.021452	-1 1
199	-0.796740	-0.538850	0.097220	0.853061	-0.817459	...	-0.233251	-0.102772	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1 1
199	-1.172550	-1.055310	-0.170214	2.045560	-0.837039	...	-0.112077	2.986180	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1 1
200	0.442055	-0.335707	0.484455	0.837146	0.424934	...	-0.483517	1.121870	0.200753	-0.017565	-0.049855	0.243362	-0.078862	-0.021452	-1 1
200	1.052340	0.140107	0.629634	-1.162770	1.018270	...	-0.678190	0.196320	-0.101455	-0.017565	-0.049855	-0.065255	0.007736	-0.021452	-1 1
200	0.760124	0.645445	-1.003510	-0.629426	0.202452	...	-0.647166	-0.102736	-0.070970	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1 1
200	0.238350	0.129306	-0.523668	0.198815	0.058170	...	-0.552947	0.718135	-0.102856	-0.017565	-0.049855	-0.050709	-0.078862	-0.021452	-1 1
200	1.232650	1.205280	0.607645	-0.061945	-0.597014	...	-0.110019	0.101500	-0.102856	-0.017565	-0.049855	2.874990	-0.078862	-0.021452	-1 1
200	0.840869	0.483899	1.179020	0.224400	-0.800514	...	0.672206	0.296234	-0.102856	-0.017565	-0.049855	8.066390	-0.078862	-0.021452	-1 1

Lampiran 5

Tabel Hasil Pra Proses (Seleski Feature) Pada Data Set FOX

Id Bag	X₁	X₂	...	X₂₁₆	X₂₁₇	X₂₁₈	X₂₁₉	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	X₂₂₉	Label	
1	-1.313750	-1.227930	...	0.563973	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	1	-1
1	0.100567	0.181302	...	-0.744061	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	1	-1
1	-1.260590	-1.142400	...	-0.746243	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	1	-1
1	-1.313120	-1.223590	...	-0.746243	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	1	-1
1	-1.274250	-1.163020	...	-0.709192	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	1	-1
1	-1.109940	-0.993377	...	-0.687824	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	1	-1
2	-0.191282	-0.236270	...	2.395460	-0.311877	-0.102856	-0.017565	-0.049855	12.012900	0.297029	-0.021452	-0.014952	1	-1
2	1.681580	1.781010	...	1.437960	-0.254573	-0.102856	-0.017565	-0.049855	6.780190	4.047530	7.672440	-0.014952	1	-1
2	0.367654	0.476738	...	1.189130	-0.332656	-0.102856	-0.017565	-0.049855	6.778390	-0.078862	-0.021452	-0.014952	1	-1
2	0.784899	1.306680	...	1.370960	-0.323233	-0.102856	-0.017565	-0.049855	9.990350	-0.078862	-0.021452	-0.014952	1	-1
2	1.879210	2.394610	...	1.264410	-0.156246	-0.102856	-0.017565	-0.049855	5.896720	-0.078862	-0.021452	-0.014952	1	-1
2	0.926793	1.704860	...	1.432140	-0.042327	-0.102856	-0.017565	-0.049855	20.439100	-0.078862	-0.021452	-0.014952	1	-1
2	0.904933	0.875859	...	2.846420	-0.081961	-0.102856	-0.017565	-0.049855	4.430750	-0.078862	-0.021452	-0.014952	1	-1
2	2.120340	2.411620	...	0.833267	-0.374972	-0.102856	-0.017565	-0.049855	5.411950	-0.078862	-0.021452	-0.014952	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
200	0.442055	-0.335707	...	-0.483517	1.121870	0.200753	-0.017565	-0.049855	0.243362	-0.078862	-0.021452	-0.014952	-1	1
200	1.052340	0.140107	...	-0.678190	0.196320	-0.101455	-0.017565	-0.049855	-0.065255	0.007736	-0.021452	-0.014952	-1	1
200	0.760124	0.645445	...	-0.647166	-0.102736	-0.070970	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-0.014952	-1	1
200	0.238350	0.129306	...	-0.552947	0.718135	-0.102856	-0.017565	-0.049855	-0.050709	-0.078862	-0.021452	-0.014952	-1	1
200	1.232650	1.205280	...	-0.110019	0.101500	-0.102856	-0.017565	-0.049855	2.874990	-0.078862	-0.021452	-0.014952	-1	1
200	0.840869	0.483899	...	0.672206	0.296234	-0.102856	-0.017565	-0.049855	8.066390	-0.078862	-0.021452	-0.014952	-1	1

Lampiran 6

Tabel Hasil Pra Proses (Seleski Feature) Pada Data Set ELEPHANT

Id Bag	X₁	X₂	...	X₂₁₀	X₂₁₁	X₂₁₅	X₂₁₆	X₂₁₇	X₂₁₈	X₂₁₉	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	Label	
1	2.057730	1.629760	...	-0.106346	-0.037709	-0.265739	-0.566476	-0.366808	-0.102856	-0.017565	-0.049855	-0.099670	-0.078862	-0.021452	1	-1
1	0.842025	0.453111	...	-0.106346	-0.037709	-0.265739	-0.630833	-0.386125	-0.102856	-0.017565	-0.049855	-0.087956	-0.078862	-0.021452	1	-1
1	-0.077775	-0.516443	...	-0.106346	-0.037709	-0.265360	-0.143848	-0.381302	-0.102856	-0.017565	-0.049855	0.249723	-0.078862	-0.021452	1	-1
1	-1.313330	-1.239750	...	-0.106346	-0.037709	-0.228280	0.413402	-0.409178	-0.102856	-0.017565	0.281890	0.086480	-0.078862	-0.021452	1	-1
1	-1.221680	-1.160640	...	-0.106346	-0.037709	-0.265739	-0.180954	-0.412626	-0.102856	-0.017565	-0.049855	-0.057620	-0.078862	-0.021452	1	-1
1	-1.108920	-1.134300	...	-0.106346	-0.037709	-0.265739	-0.705960	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1	-1
1	-1.140160	-1.049910	...	-0.106346	-0.037709	-0.265739	-0.690063	-0.366933	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1	-1
2	-0.801718	-0.728592	...	2.625790	17.761400	-0.265739	0.020179	0.336096	-0.102856	-0.017565	-0.049855	-0.004255	-0.078862	-0.021452	1	-1
2	0.729744	0.077972	...	0.040247	-0.037709	-0.265739	-0.746168	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1	-1
2	1.293230	1.728680	...	0.801317	-0.037709	-0.265739	0.083608	0.985500	-0.102856	-0.017565	-0.049855	0.139799	-0.078862	-0.021452	1	-1
2	0.193597	-0.229656	...	0.105024	-0.037709	-0.265739	-0.086671	-0.403809	-0.102856	-0.017565	-0.049855	3.767520	-0.078862	-0.021452	1	-1
2	-0.496522	-0.409700	...	-0.106346	-0.037709	-0.265739	0.028718	1.661410	-0.102856	-0.017565	-0.049855	0.364099	-0.078862	-0.021452	1	-1
2	-0.997742	-0.907455	...	2.482370	-0.037709	-0.265739	-0.680328	-0.403235	-0.102856	-0.017565	-0.049855	1.198680	-0.078862	-0.021452	1	-1
2	0.534048	0.404399	...	-0.086800	-0.037709	-0.265739	-0.714768	-0.376805	-0.102856	-0.017565	-0.049855	0.209850	-0.078862	-0.021452	1	-1
2	1.325110	2.052350	...	5.072200	-0.037709	-0.265739	-0.746243	-0.412626	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
200	0.301678	-0.325895	...	2.178160	-0.037709	-0.265739	-0.545461	0.642973	1.344810	-0.017565	-0.049855	-0.050502	-0.078862	-0.021452	-1	1
200	0.922838	0.825981	...	10.051400	-0.037709	-0.265739	-0.698940	1.414710	4.349210	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1	1
200	1.676150	0.714680	...	0.399837	-0.037709	-0.243176	0.031790	0.027152	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1	1
200	0.807393	0.387954	...	1.317660	-0.037709	-0.258399	-0.195215	0.545918	-0.098848	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1	1
200	1.261180	0.346004	...	4.564090	-0.037709	-0.265739	0.229159	1.353350	2.210770	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1	1
200	1.441560	0.815060	...	0.480160	-0.037709	-0.265739	0.412194	0.895675	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1	1
200	0.119711	-0.218966	...	-0.051490	-0.037709	-0.260506	0.623474	-0.135484	-0.102856	-0.017565	-0.049855	-0.114025	-0.078862	-0.021452	-1	1

Lampiran 7

Tabel Hasil Pra Proses (Normalisasi) Pada Data Set TIGER

Id Bag	X₁	X₂	X₃	X₄	X₅	...	X₂₁₆	X₂₁₇	X₂₁₈	X₂₁₉	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	Label	
1	0.667658	0.560200	0.255519	-0.238009	0.802591	...	-0.309822	-0.171972	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
1	0.031369	-0.085955	-0.461357	0.230912	0.174336	...	-0.356602	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
1	0.470876	0.723173	0.189365	0.121405	0.221296	...	-0.354877	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
1	-0.033235	0.063793	0.684775	-0.524944	-0.350494	...	-0.199483	-0.165440	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	-0.556211	-0.533325	-0.348865	0.773964	0.673043	...	-0.267817	-0.171291	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	0.771501	0.769618	-0.246803	0.251583	0.480539	...	-0.236260	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	0.366619	0.614793	0.085787	-0.119414	-0.149090	...	-0.343958	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	-0.499737	-0.451821	-0.610404	-0.362319	0.028920	...	-0.356719	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	-0.564940	-0.531315	-0.005581	-0.618267	-0.207808	...	-0.356719	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	-0.279369	-0.272527	-0.491078	0.908009	-0.395803	...	-0.288410	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
199	-0.370255	-0.360632	-0.393198	-0.378212	0.551897	...	0.400260	0.359965	-0.051383	-0.008782	-0.024922	0.149977	-0.039411	-0.010726	-1	1
199	0.476112	0.631339	-0.225809	0.378926	0.477634	...	-0.088904	0.413512	-0.051383	-0.008782	-0.024922	0.183147	-0.039411	-0.010726	-1	1
199	-0.560077	-0.530137	-0.434020	0.828560	0.144719	...	0.333451	0.504016	-0.051383	-0.008782	-0.024922	-0.055734	-0.039411	-0.010726	-1	1
199	-0.514547	-0.465574	0.208873	0.701006	-0.294205	...	-0.032399	0.524723	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
199	-0.281582	-0.216879	0.040705	0.095370	-0.323109	...	-0.158348	0.145053	-0.051383	-0.008782	-0.024922	0.973581	-0.039411	-0.010726	-1	1
199	-0.378553	-0.263090	0.048572	0.402418	-0.387393	...	-0.116100	-0.051341	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
199	-0.527211	-0.483587	-0.084902	0.770996	-0.395682	...	-0.055980	0.903892	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.217497	-0.166295	0.237599	0.395728	0.209327	...	-0.237156	0.508671	0.100041	-0.008782	-0.024922	0.121084	-0.039411	-0.010726	-1	1
200	0.482448	0.069939	0.304813	-0.523671	0.469271	...	-0.326669	0.097846	-0.050684	-0.008782	-0.024922	-0.032616	0.003868	-0.010726	-1	1
200	0.362761	0.311967	-0.463496	-0.304719	0.100882	...	-0.312743	-0.051323	-0.035470	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.118614	0.064563	-0.256010	0.099081	0.029077	...	-0.269638	0.344392	-0.051383	-0.008782	-0.024922	-0.025349	-0.039411	-0.010726	-1	1
200	0.548564	0.538925	0.294807	-0.030963	-0.289946	...	-0.054954	0.050706	-0.051383	-0.008782	-0.024922	0.893192	-0.039411	-0.010726	-1	1
200	0.397296	0.237336	0.529543	0.111732	-0.380169	...	0.323994	0.147043	-0.051383	-0.008782	-0.024922	0.999372	-0.039411	-0.010726	-1	1

Lampiran 8

Tabel Hasil Pra Proses (Normalisasi) Pada Data Set FOX

Id Bag	X₁	X₂	...	X₂₁₆	X₂₁₇	X₂₁₈	X₂₁₉	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	X₂₂₉	Label	
1	-0.576280	-0.546912	...	0.274743	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	1	-1
1	0.050241	0.090404	...	-0.355766	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	1	-1
1	-0.558255	-0.516240	...	-0.356719	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	1	-1
1	-0.576070	-0.545389	...	-0.356719	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	1	-1
1	-0.562939	-0.523762	...	-0.340445	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	1	-1
1	-0.504236	-0.459509	...	-0.330965	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	1	-1
2	-0.095350	-0.117588	...	0.832961	-0.154687	-0.051383	-0.008782	-0.024922	0.999988	0.147432	-0.010726	-0.007476	1	-1
2	0.686227	0.711643	...	0.616277	-0.126604	-0.051383	-0.008782	-0.024922	0.997730	0.965669	0.999069	-0.007476	1	-1
2	0.181784	0.233955	...	0.533171	-0.164811	-0.051383	-0.008782	-0.024922	0.997726	-0.039411	-0.010726	-0.007476	1	-1
2	0.373470	0.573914	...	0.595070	-0.160224	-0.051383	-0.008782	-0.024922	0.999908	-0.039411	-0.010726	-0.007476	1	-1
2	0.735041	0.832831	...	0.559569	-0.077964	-0.051383	-0.008782	-0.024922	0.994518	-0.039411	-0.010726	-0.007476	1	-1
2	0.432848	0.692337	...	0.614469	-0.021160	-0.051383	-0.008782	-0.024922	1.000000	-0.039411	-0.010726	-0.007476	1	-1
2	0.423924	0.411927	...	0.890267	-0.040958	-0.051383	-0.008782	-0.024922	0.976469	-0.039411	-0.010726	-0.007476	1	-1
2	0.785729	0.835418	...	0.394091	-0.185320	-0.051383	-0.008782	-0.024922	0.991114	-0.039411	-0.010726	-0.007476	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
200	0.217497	-0.166295	...	-0.237156	0.508671	0.100041	-0.008782	-0.024922	0.121084	-0.039411	-0.010726	-0.007476	-1	1
200	0.482448	0.069939	...	-0.326669	0.097846	-0.050684	-0.008782	-0.024922	-0.032616	0.003868	-0.010726	-0.007476	-1	1
200	0.362761	0.311967	...	-0.312743	-0.051323	-0.035470	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-0.007476	-1	1
200	0.118614	0.064563	...	-0.269638	0.344392	-0.051383	-0.008782	-0.024922	-0.025349	-0.039411	-0.010726	-0.007476	-1	1
200	0.548564	0.538925	...	-0.054954	0.050706	-0.051383	-0.008782	-0.024922	0.893192	-0.039411	-0.010726	-0.007476	-1	1
200	0.397296	0.237336	...	0.323994	0.147043	-0.051383	-0.008782	-0.024922	0.999372	-0.039411	-0.010726	-0.007476	-1	1

Lampiran 9

Tabel Hasil Pra Proses (Normalisasi) Pada *Data Set* ELEPHANT

Id Bag	X₁	X₂	...	X₂₁₀	X₂₁₁	X₂₁₅	X₂₁₆	X₂₁₇	X₂₁₈	X₂₁₉	X₂₂₃	X₂₂₄	X₂₂₅	X₂₂₆	Label	
1	0.773453	0.672274	...	-0.053123	-0.018852	-0.132093	-0.275899	-0.181375	-0.051383	-0.008782	-0.024922	-0.049794	-0.039411	-0.010726	1	-1
1	0.397783	0.222757	...	-0.053123	-0.018852	-0.132093	-0.305357	-0.190699	-0.051383	-0.008782	-0.024922	-0.043950	-0.039411	-0.010726	1	-1
1	-0.038868	-0.252631	...	-0.053123	-0.018852	-0.131907	-0.071800	-0.188374	-0.051383	-0.008782	-0.024922	0.124217	-0.039411	-0.010726	1	-1
1	-0.576140	-0.551041	...	-0.053123	-0.018852	-0.113647	0.203807	-0.201782	-0.051383	-0.008782	0.140019	0.043213	-0.039411	-0.010726	1	-1
1	-0.544718	-0.522898	...	-0.053123	-0.018852	-0.132093	-0.090231	-0.203435	-0.051383	-0.008782	-0.024922	-0.028802	-0.039411	-0.010726	1	-1
1	-0.503855	-0.513263	...	-0.053123	-0.018852	-0.132093	-0.339016	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
1	-0.515418	-0.481515	...	-0.053123	-0.018852	-0.132093	-0.331962	-0.181435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	-0.380684	-0.348992	...	0.865006	1.000000	-0.132093	0.010089	0.166484	-0.051383	-0.008782	-0.024922	-0.002127	-0.039411	-0.010726	1	-1
2	0.349498	0.038966	...	0.020121	-0.018852	-0.132093	-0.356686	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
2	0.569387	0.698487	...	0.380512	-0.018852	-0.132093	0.041780	0.456396	-0.051383	-0.008782	-0.024922	0.069786	-0.039411	-0.010726	1	-1
2	0.096497	-0.114326	...	0.052464	-0.018852	-0.132093	-0.043308	-0.199205	-0.051383	-0.008782	-0.024922	0.954825	-0.039411	-0.010726	1	-1
2	-0.243283	-0.202032	...	-0.053123	-0.018852	-0.132093	0.014358	0.680854	-0.051383	-0.008782	-0.024922	0.180065	-0.039411	-0.010726	1	-1
2	-0.461229	-0.424958	...	0.845793	-0.018852	-0.132093	-0.327624	-0.198929	-0.051383	-0.008782	-0.024922	0.536580	-0.039411	-0.010726	1	-1
2	0.260853	0.199488	...	-0.043373	-0.018852	-0.132093	-0.342908	-0.186205	-0.051383	-0.008782	-0.024922	0.104542	-0.039411	-0.010726	1	-1
2	0.580061	0.772370	...	0.987541	-0.018852	-0.132093	-0.356719	-0.203435	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	1	-1
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	
200	0.149705	-0.161520	...	0.796542	-0.018852	-0.132093	-0.266164	0.310850	0.586560	-0.008782	-0.024922	-0.025246	-0.039411	-0.010726	-1	1
200	0.431240	0.391009	...	0.999914	-0.018852	-0.132093	-0.335905	0.609016	0.974495	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.684788	0.342869	...	0.197297	-0.018852	-0.120992	0.015894	0.013575	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.383107	0.191580	...	0.577584	-0.018852	-0.128485	-0.097299	0.266376	-0.049384	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.558458	0.171296	...	0.979376	-0.018852	-0.132093	0.114081	0.589354	0.802425	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.617392	0.386373	...	0.235571	-0.018852	-0.132093	0.203228	0.420120	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1
200	0.059784	-0.109048	...	-0.025739	-0.018852	-0.129521	0.302016	-0.067639	-0.051383	-0.008782	-0.024922	-0.056951	-0.039411	-0.010726	-1	1