



TUGAS AKHIR - KI091391

# IMPLEMENTASI ARTIFICIAL BEE COLONY UNTUK PEMILIHAN TITIK PUSAT PADA ALGORITMA K-MEANS

ARIO BAGUS NUGROHO  
NRP 5110 100 056

Dosen Pembimbing I  
Diana Purwitasari, S.Kom., M.Sc.

Dosen Pembimbing II  
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.

JURUSAN TEKNIK INFORMATIKA  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016





**TUGAS AKHIR - KI091391**

# **IMPLEMENTASI ARTIFICIAL BEE COLONY UNTUK PEMILIHAN TITIK PUSAT PADA ALGORITMA K-MEANS**

**ARIO BAGUS NUGROHO  
NRP 5110 100 056**

**Dosen Pembimbing I  
Diana Purwitasari, S.Kom., M.Sc.**

**Dosen Pembimbing II  
Dr.Eng. Chastine Fatichah, S.Kom., M.Kom.**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2016**

*(Halaman ini sengaja dikosongkan)*



**UNDERGRADUATE THESES - KI091391**

**IMPLEMENTATION OF ARTIFICIAL BEE  
COLONY TO ELECT CENTROID IN K-MEANS  
ALGORITHM**

**ARIO BAGUS NUGROHO  
NRP 5110 100 056**

**First Advisor  
Diana Purwitasari, S.Kom., M.Sc.**

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

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA 2014**

*(Halaman ini sengaja dikosongkan)*

# LEMBAR PENGESAHAN

## IMPLEMENTASI ARTIFICIAL BEE COLONY UNTUK PEMILIHAN TITIK PUSAT PADA ALGORITMA K- MEANS

### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Cerdas Visual  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh

**ARIO BAGUS NUGROHO**

**NRP : 5110 100 056**

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Diana Purwitasari, S.Kom., M.Kom. ....  
NIP: 197804102003122001 (Pembimbing 1)
2. Dr.Eng. Chastine Fatchah, S.Kom., M.Kom. ....  
NIP: 197512202001122002 (Pembimbing 2)

**SURABAYA  
MEI, 2016**

*(Halaman ini sengaja dikosongkan)*



# IMPLEMENTASI ARTIFICIAL BEE COLONY UNTUK PEMILIHAN TITIK PUSAT PADA ALGORITMA K-MEANS

Nama Mahasiswa : ARIO BAGUS NUGROHO  
NRP : 5110100056  
Jurusan : Teknik Informatika FTIF-ITS  
Dosen Pembimbing 1 : Diana Purwitasari, S.Kom., M.Sc.  
Dosen Pembimbing 2 : Dr.Eng. Chastine Faticah, S.Kom., M.Kom.

## Abstrak

Klastering merupakan metode yang digunakan untuk membagi data menjadi beberapa kelompok bagian. K-means (KM) merupakan algoritma yang sering digunakan dalam klastering, hanya saja hasil dari KM sering kali terjebak di lokal optima. *Artificial Bee Colony* (ABC) merupakan algoritma yang bekerja berdasarkan cara lebah mencari makan, ABC terkenal mampu lolos dari jebakan lokal optima dengan mengenali mana hasil yang terbaik dari serangkaian hasil optimal.

Menggabungkan ABCKM dimulai dengan memilih sumber makanan awal secara acak dan menggunakan KM untuk menyelesaikan semua permasalahan klastering pada setiap langkah ABC berikutnya serta menyimpan sumber makanan terbaik disetiap iterasinya. Sumber-sumber terbaik tersebut akan dipilih sumber makanan terbaiknya berdasarkan probabilitas kecocokannya masing-masing.

Hasil dari implementasi algoritma ABCKM ini adalah data yang telah dibagi berdasarkan sumber terbaik. Setelah di evaluasi menggunakan algoritma *silhouette* dapat dibuktikan bahwa rata-rata nilai koefisien pada 5 buah *dataset* adalah 0.65 yang berarti data telah di klaster dengan baik.

***Kata Kunci:*** *K-Means, Artificial Bee Colony, KM, ABC, klastering.*

*(Halaman ini sengaja dikosongkan)*

# IMPLEMENTATION OF ARTIFICIAL BEE COLONY TO ELECT CENTROID IN K-MEANS ALGORITHM

**Student's Name** : ARIO BAGUS NUGROHO  
**Student's ID** : 5110 100 056  
**Department** : Teknik Informatika FTIF-ITS  
**First Advisor** : Diana Purwitasari, S.Kom., M.Sc.  
**Second Advisor** : Dr.Eng. Chastine Fatichah, S.Kom.,  
M.Kom.

## Abstract

*Clustering is a method that used to divide data into multiple groups of parts. K-means (KM) is an algorithm that is frequently used in clustering only the outcome of KM often trapped in a local optima. Artificial Bee Colony (ABC) is an algorithm that inspired by the way of bees foraging, ABC is famous to be able to escape from the trap of local optima by identifying which result is the best result from a set of optimal results.*

*Combining ABCKM starts with selecting random initial food source and use KM to solve all the clustering problems at every next step of ABC and save the best food source in each iteration. The best food source will be selected based on probability suitability of each.*

*The results of the implementation of ABCKM algorithm is a data that has been clustered according to the optimum food source. The clustered data will be evaluated using silhouette algorithm and proven to be having a mean coefficient value for five dataset 0.65, which means that all the data is well clustered.*

**Keywords:** *K-Means, Artificial Bee Colony, KM, ABC, Clustering.*

*(Halaman ini sengaja dikosongkan)*

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
Abstrak .....	vii
Abstract .....	ix
KATA PENGANTAR .....	xxi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xvii
DAFTAR KODE SUMBER .....	xxi
DAFTAR NOTASI.....	xxiii
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Batasan Masalah.....	3
1.4 Tujuan Tugas Akhir.....	5
1.5 Manfaat.....	5
1.6 Metodologi .....	6
1.6.1 Penyusunan Proposal Tugas Akhir.....	6
1.6.2 Studi Literatur.....	6
1.6.3 Implementasi Perangkat Lunak .....	6
1.6.4 Pengujian dan Evaluasi .....	7
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	7
BAB II TINJAUAN PUSTAKA.....	9
2.1 <i>Artificial Bee Colony</i> .....	9
2.2 <i>K-means</i> .....	16
2.3 Optimasi pada <i>Artificial Bee Colony K-means</i> .....	17
2.4 <i>Silhouette</i> .....	22
BAB III DESAIN DAN PERANCANGAN PERANGKAT LUNAK.....	25
3.1 Metode ABCKM Secara Umum.....	25
BAB IV IMPLEMENTASI.....	33
4.1 Lingkungan Implementasi .....	33

4.2 Implementasi.....	33
4.2.1 Implementasi Program Utama .....	33
4.2.2 Implementasi Inisialisasi Data Masukan .....	34
4.2.3 Implementasi Inisialisasi Sumber Makanan .....	35
4.2.4 Implementasi Algoritma <i>K-means</i> .....	37
4.2.5 Implementasi ABCKM pada Fase <i>Employed Bee</i> .....	40
4.2.6 Implementasi ABCKM pada Fase <i>Onlooker Bee</i> .....	42
4.2.7 Implementasi ABCKM pada Fase <i>Scout Bee</i> .....	45
4.2.8 Implementasi <i>Artificial Bee Colony</i> menggunakan <i>K-Means</i> .....	46
BAB V HASIL UJI COBA DAN EVALUASI.....	49
5.1 Lingkungan Pengujian .....	49
5.2 Data Pengujian.....	49
5.3 Skenario Uji Coba.....	50
5.4 Uji Kinerja .....	51
5.4.1 Hasil Uji Kinerja.....	51
BAB VI KESIMPULAN DAN SARAN.....	57
6.1 Kesimpulan .....	57
6.2 Saran .....	57
Daftar Pustaka .....	59
Lampiran A Hasil Klustering Data .....	61
Lampiran B Visualisasi ABCKM untuk Data <i>Birth and Death Rates</i> .....	69
BIODATA PENULIS.....	75

## DAFTAR GAMBAR

Gambar 3.1 Diagram Alir ABCKM.....	26
Gambar 3.2 Diagram Alir Fase Inisialisasi Sumber Makanan .....	27
Gambar 3.3 Diagram Alir Fase <i>Employed Bee</i> .....	28
Gambar 3.4 Diagram Alir Fase <i>Onlooker Bee</i> .....	29
Gambar 3.5 Diagram Alir Fase <i>Scout Bee</i> .....	30
Gambar 3.6 Diagram Alir Fase Optimasi ABCKM.....	31
Gambar Lampiran B1 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Makanan Awal Xij.....	69
Gambar Lampiran B2 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Makanan Awal Xij setelah di KM .....	70
Gambar Lampiran B3 Grafik Data <i>Birth and Death Rates</i> terhadap Tetangga Sumber Makanan Awal Vij setelah di KM .....	70
Gambar Lampiran B4 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Makanan Hasil KM Gabungan dari Xij dan Vij.....	71
Gambar Lampiran B5 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Makanan yang akan ditinggalkan.....	71
Gambar Lampiran B6 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Terbaik pada Iterasi Satu yang akan disimpan .....	72
Gambar Lampiran B7 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Makanan Optimal Hasil ABCKM.....	72
Gambar Lampiran B8 Grafik Data <i>Birth and Death Rates</i> terhadap Sumber Makanan Awal yang dibandingkan dengan Sumber Optimal Hasil ABCKM .....	73

*(Halaman ini sengaja dikosongkan)*



## DAFTAR TABEL

Tabel 2.1 Contoh Data <i>iris</i> .....	11
Tabel 2.2 Sumber Makanan Awal Xij .....	11
Tabel 2.3 Sumber Makanan Awal Xij setelah di KM .....	12
Tabel 2.4 Sumber Makanan Awal Vij .....	13
Tabel 2.5 Sumber Makanan Awal Vij setelah di KM .....	13
Tabel 2.6 Sumber Makanan Xij baru .....	13
Tabel 2.7 Sumber Makanan Vij baru .....	14
Tabel 2.8 Sumber Makanan Terbaik Saat Ini .....	14
Tabel 2.9 Sumber Makanan Terburuk Saat Ini .....	15
Tabel 2.10 Sumber Makanan Optimal .....	15
Tabel 2.11 Hasil Klasifikasi data <i>iris</i> .....	16
Tabel 2.12 Contoh Data <i>Wholesale</i> .....	17
Tabel 2.13 Sumber Makanan Awal Xij .....	18
Tabel 2.14 Sumber Makanan Awal Xij setelah di KM .....	18
Tabel 2.15 Sumber Makanan Awal Vij .....	19
Tabel 2.16 Sumber Makanan Awal Vij setelah di KM .....	19
Tabel 2.17 Sumber Makanan Xij baru .....	20
Tabel 2.18 Sumber Makanan Vij baru .....	20
Tabel 2.19 Sumber Makanan Terbaik Saat Ini .....	21
Tabel 2.20 Sumber Makanan Terburuk Saat Ini .....	21
Tabel 2.21 Sumber Makanan Optimal .....	22
Tabel 2.22 Hasil Klasifikasi data <i>wholesale</i> .....	22
Tabel 5.1 Tabel Karakteristik Kelima <i>Dataset</i> .....	50
Tabel 5.2 Uji Perbandingan Nilai Parameter Jumlah Klaster pada <i>dataset crimerate.txt</i> .....	51
Tabel 5.3 Uji Perbandingan Nilai Parameter Jumlah Klaster pada <i>dataset dowjonesindex.txt</i> .....	52
Tabel 5.4 Uji Perbandingan Nilai Parameter Jumlah Klaster pada <i>dataset wholesale.txt</i> .....	53
Tabel 5.5 Uji Perbandingan Nilai Parameter Jumlah Klaster pada <i>dataset birthanddeathrates.txt</i> .....	54

Tabel 5.6 Uji Perbandingan Nilai Parameter Jumlah Klaster pada <i>dataset iris.txt</i> .....	54
Tabel 5.7 Hasil Pengujian <i>Confusion Matrices</i> Data <i>Iris</i> .....	55
Tabel Lampiran A1 Hasil Klastering data <i>crimerate</i> Klaster 2 .....	61
Tabel Lampiran A2 Hasil Klastering data <i>crimerate</i> Klaster 3 .....	61
Tabel Lampiran A3 Hasil Klastering data <i>crimerate</i> Klaster 4 .....	61
Tabel Lampiran A4 Hasil Klastering data <i>crimerate</i> Klaster 6 .....	62
Tabel Lampiran A5 Hasil Klastering data <i>dowjonesindex</i> Klaster 2.....	62
Tabel Lampiran A6 Hasil Klastering data <i>dowjonesindex</i> Klaster 3.....	62
Tabel Lampiran A7 Hasil Klastering data <i>dowjonesindex</i> Klaster 4.....	63
Tabel Lampiran A8 Hasil Klastering data <i>dowjonesindex</i> Klaster 5.....	63
Tabel Lampiran A9 Hasil Klastering data <i>dowjonesindex</i> Klaster 6.....	63
Tabel Lampiran A10 Hasil Klastering data <i>wholesale</i> Klaster 2 .....	64
Tabel Lampiran A11 Hasil Klastering data <i>wholesale</i> Klaster 3 .....	64
Tabel Lampiran A12 Hasil Klastering data <i>wholesale</i> Klaster 4 .....	64
Tabel Lampiran A13 Hasil Klastering data <i>wholesale</i> Klaster 5 .....	64
Tabel Lampiran A14 Hasil Klastering data <i>wholesale</i> Klaster 6 .....	65
Tabel Lampiran A15 Hasil Klastering data <i>birthanddeathrates</i> Klaster 2.....	65
Tabel Lampiran A16 Hasil Klastering data <i>birthanddeathrates</i> Klaster 3.....	65

Tabel Lampiran A17 Hasil Klastering data <i>birthanddeathrates</i> Klaster 4 .....	66
Tabel Lampiran A18 Hasil Klastering data <i>birthanddeathrates</i> Klaster 5 .....	66
Tabel Lampiran A19 Hasil Klastering data <i>birthanddeathrates</i> Klaster 6 .....	66
Tabel Lampiran A20 Hasil Klastering data <i>iris</i> Klaster 2.....	67
Tabel Lampiran A21 Hasil Klastering data <i>iris</i> Klaster 3.....	67
Tabel Lampiran A22 Hasil Klastering data <i>iris</i> Klaster 4.....	67
Tabel Lampiran A23 Hasil Klastering data <i>iris</i> Klaster 5.....	67
Tabel Lampiran A24 Hasil Klastering data <i>iris</i> Klaster 6.....	68

*(Halaman ini sengaja dikosongkan)*

## DAFTAR KODE SUMBER

Kode Sumber 4.1 Implementasi Program Utama.....	33
Kode Sumber 4.2 Implementasi Inisialisasi Data Masukan..	34
Kode Sumber 4.3 Implementasi Inisialisasi Sumber Makanan .....	36
Kode Sumber 4.4 Implementasi <i>K-means</i> untuk Sekali Iterasi .....	37
Kode Sumber 4.5 Implementasi <i>K-means</i> .....	39
Kode Sumber 4.6 Implementasi ABCKM pada Fase <i>Employed Bee</i> .....	40
Kode Sumber 4.7 Implementasi Probabilitas untuk Fase <i>Onlooker Bee</i> .....	42
Kode Sumber 4.8 Implementasi ABCKM pada Fase <i>Onlooker Bee</i> .....	44
Kode Sumber 4.9 Implementasi ABCKM pada Fase <i>Scout Bee</i> .....	45
Kode Sumber 4.10 Implementasi <i>Artificial Bee Colony</i> menggunakan <i>K-means</i> .....	46

*(Halaman ini sengaja dikosongkan)*

## DAFTAR NOTASI

$SN$	Jumlah sumber makanan
$D$	Jumlah dimensi data
$K$	Jumlah klaster
$k$	$\{1,2,\dots,K\}$
$i$	$\{1,2,\dots,SN\}$
$j$	$\{1,2,\dots,D\}$
$x_{i,j}$	Sumber makanan awal- $i$ pada dimensi ke- $j$
$l_{jk}$	Nilai bawah dari tiap nilai yang ada di dimensi- $j$ untuk tiap klaster
$u_{jk}$	Nilai atas dari tiap dimensi- $j$ untuk tiap klaster
$rand(0,1)$	Bangkitkan angka acak dengan distribusi normal 0 sampai 1.
$fit_i$	<i>Fitness</i> dari sumber makanan.
$f_i$	Fungsi biaya dari permasalahan klaster.
$D_{Train} =$	Jumlah banyaknya data (sumber makanan) yang akan digunakan untuk menormalisasi penjumlahan yang akan digolongkan.
$p_i^{CL_{known}(x_j)}$	Mendefinisikan kelas instansi data.
$x_j$	Data ke- $j$ .
$v_{ij}$	Kandidat posisi makanan baru berdasarkan posisi yang lama dalam memori. $k \in \{1,2, \dots, SN\}$ dan $j \in \{1,2, \dots, D\}$ adalah indeks yang dipilih acak, namun $k$ harus berbeda indeks dengan $i$ .
$\phi_{ij}$	Angka acak diantara $\{-1,1\}$
$z_i$	Sumber makanan yang di tinggalkan <i>onlooker bee</i> dan $j \in \{1,2, \dots, D\}$ .
$\tau_{max}$	Nilai zat <i>pheromones</i> maksimal.
$X_i$	Vektor nilai <i>input</i> $\{x_1, x_2, \dots, x_n\}$ .
$\bar{Y}_j$	Vektor nilai <i>centroid</i> $\{y_1, y_2, \dots, y_n\}$ .
$\mu_k$	<i>Centroid</i> baru dari klaster ke- $k$ .
$N_k$	Banyaknya data pada klaster ke- $k$ .
$x_i$	Vektor nilai input ke- $i$ pada klaster ke- $k$ .
$i$	Obyek data ke- $i$

- a Rata-rata jarak  $i$  terhadap obyek lain pada klaster yang sama.
- b Nilai minimum dari rata-rata jarak  $i$  terhadap obyek lain pada klaster yang berbeda.
- si Nilai koefisien *silhouette*



# **BAB I**

## **PENDAHULUAN**

Bab ini akan menjelaskan beberapa hal dasar yang meliputi: latar belakang, tujuan, manfaat permasalahan, batasan permasalahan, metodologi serta sistematika penulisan Tugas Akhir. Latar belakang berisi mengenai hal-hal yang melatarbelakangi pemilihan judul Tugas Akhir. Rumusan masalah memuat hal-hal yang harus diselesaikan. Batasan masalah berisi batasan-batasan yang melingkupi pembuatan Tugas Akhir ini. Tujuan berisi tujuan dari pembuatan Tugas Akhir ini. Metodologi membahas mengenai metode pengerjaan Tugas Akhir. Sistematika penulisan membahas sistematika penulisan buku Tugas Akhir sebagai laporan dari pengerjaan Tugas Akhir. Penjelasan tentang hal-hal tersebut diharapkan dapat memberikan gambaran umum mengenai permasalahan sehingga penyelesaian masalah dapat dipahami dengan baik.

### **1.1 Latar Belakang**

Klastering digunakan untuk membagi data menjadi kelompok yang homogen, banyak aplikasi dari klastering yang telah dirasakan manfaatnya seperti segmentasi citra, mengenali segmentasi pasar dalam bisnis, temu kembali informasi dan juga dalam merangkum data [1].

Segmentasi citra memiliki peranan penting dalam banyak hal khususnya di bidang biomedik, banyak studi yang melakukan riset dan pengaplikasian segmentasi citra untuk mengenali karakteristik dan bentuk morfologi penyakit melalui data citra, berdasarkan pada keserupaan tertentu antara tingkat keabuan suatu piksel dengan tingkat keabuan piksel-piksel tetangganya. Sedangkan implementasi klastering untuk segmentasi pasar dalam bisnis digunakan untuk membagi kelompok konsumen berdasarkan kebutuhan, perilaku dan karakteristik masing-masing kelompok konsumen agar pelaku bisnis dapat memasarkan produknya dengan lebih efektif.

*K-means* (KM) merupakan salah satu algoritma klusterisasi yang umum digunakan karena kemudahan dan relatif cepatnya waktu yang dibutuhkan dalam menjalankan pembelajaran. Permasalahannya hasil kluster KM kerap mengacu pada solusi lokal optima [3]. Yaitu solusi yang optimal (baik maksimal ataupun minimal) pada kandidat solusi di tetangga terdekatnya saja bukan keseluruhan dari semua solusi yang ada atau yang biasa disebut global optima.

Maka dari itu diusulkan agar mengimplementasikan algoritma *Artificial Bee Colony* (ABC) yang biasa digunakan untuk pencarian global seperti permasalahan *Travelling Salesman Problem* (TSP) yang mencari rute termurah dan efisien, untuk mencapai tujuan para sales yang sangat banyak. Cara kerja ABC meniru cara lebah mencari makan (*foregaging*) nektar. Dimulai dari terbangnya *employed bee* mencari sumber makanan yang kemudian informasi letak/posisinya akan diberikan kepada *onlooker bee* dengan cara menari.

*Onlooker bee* akan membandingkan nilai probabilitas masing-masing letak sumber makanannya dan mencari sumber makanan disekitar sumber yang dipilih, hingga seketika sumber makanan yang ditemukan saat ini lebih banyak dari sumber sebelumnya maka lebah tersebut akan melupakan informasi tentang sumber makanan terbanyak sebelumnya. Sumber makanan yang habis atau tidak berubah, sumber makanan tersebut akan ditinggalkan dan dilupakan serta lebah tersebut akan menjadi *scout bee* yang akan mengintai sebuah sumber makanan baru yang akan diciptakan di ruang pencarian. Langkah-langkah tersebut akan dilakukan hingga ditemukan solusi global optima [2].

Pengaplikasian lain dari ABC adalah penjadwalan produksi barang, yang akan digunakan untuk menjaga siklus produksi, yaitu serangkaian aktifitas bisnis seperti pencatatan order dari pelanggan, pencatatan bahan mentah, juga pencatatan gaji karyawan, dan kegiatan pengolahan data secara kontinu. Peran ABC pada pengolahan data siklus bisnis akan sangat signifikan dimana akan dibutuhkan perencanaan jumlah produksi dan

persediaan, menjadwalkan proses produksi dengan meminimalkan jumlah waktu proses yang dibutuhkan untuk menyelesaikan seluruh proses produksi, dan juga mengoptimalkan biaya pokok produksi.

Metode klustering yang ditawarkan menggabungkan kedua algoritma tersebut menjadi *Artificial Bee Colony K-means* (ABCKM). Metode ini akan melengkapi cara kerja algoritma KM pada pemilihan *centroid* dan dengan sifat pencarian global dari ABC, kumpulan sumber makanan yang telah disimpan akan dibandingkan dan dipilih sehingga didapatkan sumber makanan optimal untuk membagi datanya.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam Tugas Akhir ini dapat dipaparkan sebagai berikut:

1. Melakukan implementasi algoritma *K-means*.
2. Melakukan implementasi algoritma *Artificial Bee Colony*.
3. Melakukan implementasi metode *K-means* pada setiap langkah algoritma *Artificial Bee Colony*.
4. Melakukan evaluasi terhadap gabungan algoritma *Artificial Bee Colony* dan *K-means*.

## 1.3 Batasan Masalah

Tugas Akhir ini mempunyai banyak permasalahan, tetapi tidak semua permasalahan tersebut akan terselesaikan karena terdapat keterbatasan sumber daya. Maka karena itu, dibuat batasan-batasan permasalahan yang ada, antara lain batasannya sebagai berikut:

1. Implementasi algoritma yang diusulkan menggunakan bahasa pemrograman Matlab.
2. Implementasi algoritma akan digunakan untuk mengkluster beberapa *dataset* yaitu :

- 2.1.1. *Wholesale Customer* yang merupakan data penjualan kebutuhan sehari-hari pada suatu distributor *Wholesale* yang memiliki 6 atribut numerikal yaitu penjualan makanan dan kebutuhan rumah tangga seperti *fresh, milk, grocery, frozen, detergents and paper*, dan *delicatessen*, di dua kota besar dan satu kota komplementer yang merupakan gabungan kota-kota lain, baik melalui restoran, hotel, cafe maupun toko eceran. Yang di dapatkan dari <https://archive.ics.uci.edu/ml/datasets/Wholesale+customers>,
- 2.1.2. *Crime Rate America* yang merupakan data banyaknya kota dan kejahatannya disetiap populasi 100.000 pada 1970 di amerika yang memiliki 16 baris nama kota dan 7 atribut numerik yang terdiri dari kejahatan pembunuhan, pemerkosaan, perampokan, pencurian, penyerangan, pencurian mobil, penggarongan. Yang didapatkan dari <http://people.sc.fsu.edu/~jburkardt/datasets/hartigan/file03.txt>
- 2.1.3. *Birth and Death Rates* yang merupakan data banyaknya Negara terhadap angka kehidupan dan kematiannya disetiap 1000 orang pada tahun 1966 yang didapatkan dari <http://people.sc.fsu.edu/~jburkardt/datasets/hartigan/file26.txt>
- 2.1.4. *Iris dataset* yang merupakan data yang mencatat ukuran 3 jenis kelas tanaman yaitu *Iris Setosa, Iris Versicolour, Iris Virginica* berdasarkan lebar, tinggi *sepal*-nya dan lebar, tinggi *petal*-nya. Yang di dapatkan dari <https://archive.ics.uci.edu/ml/datasets/Iris>

2.1.5. *Dow Jones Index* yang merupakan data yang akan digunakan untuk memprediksi harga saham yang dikumpulkan pada suatu periode waktu, dimana pada data tersebut setiap barisnya mencatat datanya setiap minggu, sehingga tujuan dari penggunaan klustering pada data ini adalah dapat menentukan saham mana yang akan menghasilkan tingkat terbesar kembali untuk menjadi pertimbangan berinvestasi di minggu berikutnya. Yang di dapatkan dari <https://archive.ics.uci.edu/ml/datasets/Dow+Jones+Index>

#### **1.4 Tujuan Tugas Akhir**

Tugas Akhir ini memiliki tujuan sebagai berikut:

1. Melakukan implementasi algoritma ABCKM untuk menyelesaikan suatu permasalahan klustering.
2. Melakukan evaluasi algoritma ABCKM dengan cara melakukan implementasi dan uji coba.

#### **1.5 Manfaat**

Algoritma *Artificial Bee Colony K-means* ini diharapkan mampu memberikan hasil klustering yang lebih baik, dan didapatkan solusinya sehingga dapat diterapkan pada kasus-kasus dan permasalahan klustering di lingkup yang lebih besar.

#### **1.6 Metodologi**

Pembuatan Tugas Akhir ini terbagi menjadi beberapa tahap, adapun tahapan-tahapan yang dilakukan dalam pengerjaan Tugas Akhir ini adalah sebagai berikut:

### **1.6.1 Penyusunan Proposal Tugas Akhir**

Tahap awal pengerjaan Tugas Akhir adalah menyusun proposal Tugas Akhir. Gagasan yang diajukan pada proposal Tugas Akhir ini adalah menggabungkan algoritma *Artificial Bee Colony* dengan *K-means*.

### **1.6.2 Studi Literatur**

Tahap berikutnya adalah mencari informasi, pengidentifikasian masalah, pencarian studi literatur, perumusan masalah, dan penetapan tujuan penelitian. Studi literatur digunakan dengan tujuan untuk mencari, mengumpulkan, mempelajari, dan memahami informasi dari berbagai sumber literatur yang diperlukan untuk pembuatan aplikasi penyelesaian permasalahan klustering. Dasar informasi yang diperlukan diantaranya mengenai karakteristik algoritma ABC dan KM, algoritma ABCKM yang akan dikembangkan untuk menyelesaikan permasalahan pembagian data hingga mencapai hasil optimal. Informasi dan literatur didapatkan dari buku, jurnal ilmiah, internet, maupun sumber-sumber informasi lain yang berhubungan.

### **1.6.3 Implementasi Perangkat Lunak**

Tahapan ini melakukan pembangunan dari rancangan aplikasi yang dibuat. Membuat suatu realisasi dari apa yang terdapat pada tahapan perancangan sehingga menjadi sebuah aplikasi yang sesuai dengan apa yang telah direncanakan sebelumnya. Implementasi penyelesaian permasalahan klustering dengan menggunakan algoritma ABCKM. Pelaksanaan implementasi tersebut dilakukan dengan perangkat lunak Matlab. Hasil dari aplikasi yang akan diimplementasikan diharapkan mampu didapat solusi optimal dari setiap permasalahan klustering yang diberikan.

### 1.6.4 Pengujian dan Evaluasi

Tahap pengujian dan evaluasi dilakukan proses uji coba terhadap aplikasi yang telah dibuat. Pengujian dan evaluasi akan dilakukan dengan melihat kesesuaian dengan rancangan awal serta *output* yang dihasilkan. Tahap ini dilakukan juga untuk mengevaluasi jalannya aplikasi, mencari *bug* yang mungkin muncul pada aplikasi, dan memperbaiki aplikasi apabila terdapat kesalahan dalam proses eksekusinya. Diharapkan dengan melakukan evaluasi tersebut maka program dapat disempurnakan.

### 1.7 Sistematika Penulisan Laporan Tugas Akhir

Sistematika penulisan laporan Tugas Akhir adalah sebagai berikut:

1. Bab I. Pendahuluan  
Bab ini berisi penjelasan mengenai latar belakang masalah, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu juga turut disertakan pula rumusan permasalahan, batasan permasalahan, dan sistematika penulisan pada bab ini.
2. Bab II. Tinjauan Pustaka  
Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang untuk mendukung pembuatan Tugas Akhir. Dasar ilmu tersebut antara lain algoritma *Artificial Bee Colony*, *K-means*.
3. Bab III. Desain dan Implementasi  
Bab ini berisi penjelasan mengenai desain metode dan implementasi dari algoritma yang telah diusulkan untuk Tugas Akhir ini.
4. Bab IV. Implementasi  
Bab ini berisi tentang penjelasan dan pembahasan mengenai implementasi program yang telah dilakukan.
5. Bab V. Pengujian dan Evaluasi  
Bab ini berisi penjelasan mengenai data hasil percobaan dan pembahasan mengenai hasil percobaan yang telah dilakukan.

6. Bab VI. Kesimpulan dan Saran  
Bab ini berupa hasil penelitian yang menjawab permasalahan atau yang berupa konsep, program, dan rancangan. Selain itu pada bab ini diberikan saran-saran yang berisi hal-hal yang berhubungan dengan penelitian yang telah dilakukan untuk dikembangkan lebih lanjut atau berisi masalah-masalah yang dialami pada proses pengerjaan Tugas Akhir.
7. Daftar Pustaka  
Bab ini berisi daftar pustaka yang dijadikan literatur dalam pengerjaan Tugas Akhir.
8. Lampiran Tugas Akhir  
Dalam lampiran terdapat tabel-tabel data hasil uji coba dan kode sumber program keseluruhan.



## BAB II TINJAUAN PUSTAKA

Bab ini membahas tentang teori dasar yang menunjang pengerjaan serta penyusunan Tugas Akhir mengenai penggunaan algoritma *Artificial Bee Colony* yang telah dimodifikasi dengan algoritma *K-means* sehingga menjadi *Artificial Bee Colony K-means*.

### 2.1 Artificial Bee Colony

*Artificial Bee Colony* (ABC) merupakan salah satu algoritma optimasi yang berdasarkan cara lebah mencari makanan. Seketika suatu tempat bernektar telah ditemukan, *Employed bee* akan menari (memberi tanda) agar nektar tersebut dapat dipanen oleh *onlooker bee*. Kemudian *Onlooker bee* akan menentukan mana saja sumber makanan yang baik untuk dipanen, dan meninggalkan sumber yang habis dan berubah menjadi *scout bee*. *Scout bee* bertugas mencari sumber baru yang dibuat secara acak pada ruang pencarian, dimana informasi posisi nektar yang lebih sedikit sebelumnya akan dilupakan oleh lebah tersebut sampai ditemukan posisi sumber makanan yang terbaik. [4].

Berikut langkah yang digunakan dalam ABC [4]:

1. Menentukan jumlah sumber makanan awal atau calon-calon solusi (SN) yang akan disebut  $X_{ij}$ .
2. Mengevaluasi kecocokan  $f_i$  (kualitas sumber makanan) pada populasi karena semakin banyak nektarnya semakin besar probabilitas sumber tersebut dipilih *onlooker bee*.
3. Masuk ke fase *Employed Bee*, yaitu menghitung  $V_{ij}$  yang merupakan tetangga disekitar  $X_{ij}$ .
4. Masuk ke fase *Onlooker Bee* yang akan mengerjakan langkah kedua dengan tujuan menggabungkannya menjadi sumber dengan probabilitas terbaik untuk setiap klasternya yang akan disebut  $X_{ij}$  baru. Sumber tersebut akan diulang langkah

*employed Bee*, tetapi sumber terbaiknya akan disimpan di memori sebagai sumber terbaik pada iterasi ini.

5. Masuk ke fase *Scout Bee*, fase ini membangkitkan sumber baru dengan kondisi kumpulan sumber makanan belum mencapai batas limit *Maximum Cycle Number*. Fase ini akan menjembatani iterasi pertama ke iterasi kedua dan berikutnya.
6. Kumpulan sumber tersebut akan dibandingkan probabilitas kecocokannya hingga ditemukan sumber optimal.

$$x_{i,j} = l_{jk} + rand(0,1)(u_{jk} - l_{jk}) \quad (2.1)$$

Dimana :

- SN = Jumlah sumber makanan
- D = Jumlah dimensi data
- K = Jumlah klaster
- k =  $\{1,2,\dots,K\}$
- i =  $\{1,2,\dots,SN\}$
- j =  $\{1,2,\dots,D\}$
- $x_{i,j}$  = Sumber makanan awal-i pada dimensi ke-j
- $l_{jk}$  = nilai bawah dari tiap nilai yang ada di dimensi-j untuk tiap klaster
- $u_{jk}$  = nilai atas dari tiap dimensi-j untuk tiap klaster
- $rand(0,1)$  = membangkitkan angka acak dengan distribusi normal 0 sampai 1.

Pada *dataset iris* memiliki matriks 150x4 seperti Tabel 2.1 yang akan dibagi menjadi 3 klaster seperti Tabel 2.2 yang berarti:

- SN = 3
- D = 4
- K = 3
- $l_{jk}$  =  $\{1,50,100\}$
- $u_{jk}$  =  $\{50,100,150\}$

**Tabel 2.1 Contoh Data *iris***

No	Sepal length	Sepal Width	Petal length	Petal width
1	5.1	3.5	1.4	0.2
2	4.9	3.0	1.4	0.2
3	4.7	3.2	1.3	0.2
4	4.6	3.1	1.5	0.2
5	5.0	3.6	1.4	0.2

**Tabel 2.2 Sumber Makanan Awal Xij**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.3	2.0	1.0	0.1
2	6.4	2.9	2.9	1.6
3	7.3	4.2	6.9	2.6

Menghitung *fitness* dengan cara:

$$fit_i = \frac{1}{1+f_i} \quad (2.2)$$

$$f_i = \frac{1}{D_{Train}} \sum_{j=1}^{D_{Train}} d(x_j, p_i^{CL_{known}(x_j)}) \quad (2.3)$$

Dimana:

- $fit_i$  = *Fitness* dari sumber makanan.
- $f_i$  = Fungsi biaya dari permasalahan klaster.
- $D_{Train}$  = Jumlah banyaknya data (sumber makanan) yang akan digunakan untuk menormalisasi penjumlahan yang akan digolongkan.
- $p_i^{CL_{known}(x_j)}$  = Mendefinisikan kelas instansi data.
- $x_j$  = Data ke-j.

Pada *Dataset iris* maupun kasus *dataset* lain fungsi biaya permasalahan klaster dikerjakan dengan *K-means* dimana hasilnya

$f_i$  akan dicari *fitness*-nya pada masing-masing baris data seperti Tabel 2.3.

**Tabel 2.3 Sumber Makanan Awal Xij setelah di KM**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	2.78	1.59	0.30
2	5.99	3.08	4.50	1.46
3	6.87	3.33	5.79	2.11

**Memulai siklus iterasi  $cycle = 1$  hingga MCN.**

Untuk setiap *employed bee* hitung solusi  $v_i$  baru dengan cara:

$$v_{ij} = z_{ij} + \phi_{ij}(z_{ij} - z_{kj}) \quad (2.4)$$

Dimana:

- $v_{ij}$  = Kandidat posisi makanan baru berdasarkan posisi yang lama dalam memori.  $k \in \{1,2, \dots, SN\}$  dan  $j \in \{1,2, \dots, D\}$  adalah indeks yang dipilih acak, namun  $k$  harus berbeda indeks dengan  $i$ .
- $\phi_{ij}$  = Angka acak diantara  $\{-1,1\}$

Indeks acak pada persamaan tersebut bertujuan untuk mengontrol produksi sumber makanan tetangga di sekitar  $z_{i,j}$  dan merepresentasikan perbandingan dua posisi sumber makanan yang terlihat pada lebah. Langkah berikutnya adalah Hitung nilai  $f_i$ , dan lakukan proses pemilihan sumber makanan secara *greedy* di setiap klasternya.

Pada *dataset iris* sumber makanan yang telah di *k-means* akan di carikan sumber makanan tetangganya/di sekitarnya yang akan di *k-means* juga dan dibandingkan hasil *fitness*-nya disetiap klaster. Langkah ini akan dijelaskan pada Tabel 2.4, dan Tabel 2.5 untuk sumber makanan berdasarkan hasil KM yang sudah tidak berubah.

**Tabel 2.4 Sumber Makanan Vij**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	1.18	0.26	0.98
2	5.73	3.22	4.04	0.96
3	6.43	3.09	5.86	0.35

**Tabel 2.5 Sumber Makanan Vij setelah di KM**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	2.78	1.59	0.30
2	5.99	3.08	4.50	1.46
3	6.87	3.33	5.79	2.11

Untuk setiap *onlooker bee* akan menentukan solusi  $z_i$  berdasarkan  $p_i$ . Kemudian hitung nilai  $v_i$  sesuai langkah *Employed Bee*. Bandingkan kedua sumber makanan tersebut secara *greedy*. Berdasarkan hasil probabilitas, solusi terbaik pada iterasi ini akan disimpan.

Pada *dataset iris* sumber makanan yang didapatkan dari *employed bee* akan di *k-means* dan dicarikan sumber makanan tetangganya/di sekitarnya yang akan di *k-means* juga dan dibandingkan hasil *fitness*-nya sekali lagi seperti Tabel 2.6 dan Tabel 2.7.

**Tabel 2.6 Sumber Makanan Xij baru**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	2.78	1.59	0.30
2	5.99	3.08	4.50	1.46
3	6.87	3.33	5.79	2.11

**Tabel 2.7 Sumber Makanan Vij baru**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	2.72	1.63	0.85
2	6.11	2.90	4.46	4.81
3	5.66	4.45	5.31	3.16

Berdasarkan hasil probabilitas, solusi yang paling buruk ditinggalkan *onlooker bee* dan diganti dengan solusi baru yang secara acak didapatkan dari :

$$z_i^j = z_{min}^j + rand(0,1)(z_{max}^j - z_{min}^j) \quad (2.5)$$

Dimana:

- $z_i$  = Sumber makanan yang di tinggalkan *onlooker bee* dan  $j \in \{1,2, \dots, D\}$ .

Pada *dataset iris* setelah setiap kandidat posisi sumber  $v_{i,j}$  di produksi dan di evaluasi oleh *scout bee* pada iterasi berikutnya, performanya akan dibandingkan dengan sebelumnya, jika sumber tersebut sama atau lebih banyak maka posisi dan jalurnya akan disimpan dalam memori dan sebaliknya, disinilah letak ABC melepaskan diri dari jeratan lokal optima. Sumber makanan terbaik akan disimpan seperti Tabel 2.8. Sedangkan Sumber makanan hasil dari fase scout ditunjukkan pada Tabel 2.9

**Tabel 2.8 Sumber Makanan Terbaik Saat Ini**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	2.78	1.59	0.30
2	5.99	3.08	4.50	1.46
3	6.87	3.33	5.79	2.11

**Tabel 2.9 Sumber Makanan untuk Iterasi Berikutnya**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	4.30	2.00	2.00	0.10
2	5.40	2.90	2.90	0.60
3	7.30	4.20	6.90	1.60

Ulangi siklus hingga *Maximum Cycle Number* (MCN). *Maximum Cycle Number* untuk ABCKM dihitung dengan persamaan berikut :

$$\text{limit} = SN * D \quad (2.6)$$

Sumber makanan optimal didapatkan dari perbandingan nilai probabilitas kecocokan kumpulan sumber-sumber makanan tersebut. Tabel 2.10 merupakan tabel yang menunjukkan sumber makanan optimal yang lepas dari jebakan lokal optima. Tabel 2.11 merupakan tabel hasil klasifikasi data iris menggunakan sumber makanan optimal tersebut.

**Tabel 2.10 Sumber Makanan Optimal**

Klaster	Sepal length	Sepal Width	Petal length	Petal width
1	5.00	2.78	1.59	0.30
2	5.99	3.08	4.50	1.46
3	6.87	3.33	5.79	2.11

**Tabel 2.11 Hasil Klasifikasi data iris**

Klaster	Jumlah masing-masing klasifikasi data iris
1	54
2	61
3	35

## 2.2 K-means

*K-means* (KM) merupakan suatu algoritma kluster data yang mengelompokkan data berdasarkan data yang tidak berlabel kelas. Berikut langkah-langkah yang dilakukan dalam KM [3]:

1. Menentukan jumlah k (klaster).
2. Bangkitkan k *centroid* (titik pusat klaster) awal secara acak.
3. Hitung jarak setiap data ke setiap *centroid*-nya menggunakan rumus korelasi antar dua objek yaitu *Euclidean Distance* pada setiap klaster.
4. Kelompokkan setiap data berdasarkan jarak terdekat antara data dengan *centroid*-nya.
5. Perbarui nilai *centroid* dengan nilai *centroid*
6. Ulangi langkah 3-5 hingga nilai *centroid* tidak berubah.

Jarak antar input dengan *centroid* dihitung dengan cara:

$$d(\bar{X}_i, \bar{Y}_j) = \sqrt{(x_1 - y_1)^2 + \dots + (x_n - y_n)^2} \quad (2.7)$$

Dimana:

- $\bar{X}_i$  = Vektor nilai input  $\{x_1, x_2, \dots, x_n\}$ .
- $\bar{Y}_j$  = Vektor nilai *centroid*  $\{y_1, y_2, \dots, y_n\}$ .

Memperbarui nilai *centroid* dengan nilai *centroid* baru dilakukan dengan cara:

$$\mu_k = \frac{1}{N_k} \sum_{i=1}^{N_k} x_i \quad (2.8)$$

Dimana:

- $\mu_k$  = *Centroid* baru dari klaster ke-k.



- $N_k$  = Banyaknya data pada kluster ke-k.
- $x_i$  = Vektor nilai input ke-i pada kluster ke-k.

Memilih titik pusat suatu kluster merupakan tugas yang rumit karena diatur maupun diacak, apabila nilai inisialisasi yang dilakukan kurang baik maka pengelompokan data bisa kurang optimal [3].

### 2.3 Optimasi pada *Artificial Bee Colony K-Means*

Algoritma *K-means* akan digunakan pada setiap langkah *Artificial Bee Colony* untuk menyelesaikan permasalahan klustering untuk setiap sumber makanan pada suatu data.

Berikut Visualisasi data dan perhitungan pada siklus iterasi pertama pada ABCMKM:

**Tabel 2.12 Contoh data *Wholesale***

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	12669	9656	7561	214	2674	1338
2	7057	9810	9568	1762	3293	1776
3	6353	8808	7684	2405	3516	7844
4	13265	1196	4221	6404	507	1788
5	22615	5410	7198	3915	1777	5185

Tabel 2.12 diatas merupakan sampel dari data *Wholesale.txt* yang merupakan matriks 440x6 yang merupakan data penjualan barang-barang rumah tangga yang dijual pada gerai *Wholesale* dalam satuan banyaknya barang yang dijual. Dimensi pada data ini adalah berbagai macam barang yang dijual yaitu Makanan Segar, Susu, Bahan Makanan, Makanan Beku, Detergen, dan *Delicatessen*.

Fase pertama Inisialisasi Sumber Makanan dijelaskan melalui Tabel 2.13 berikut yang merupakan hasil dari persamaan (1) yaitu  $x_{i,j}$  yang diberi nama SMA atau Sumber Makanan Awal bersama dengan K atau klaster dan MCN yang merupakan *limit* untuk banyaknya iterasi yang akan dikerjakan pada ABCKM.

**Tabel 2.13 Sumber Makanan Awal Xij**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	1746	715	1573	293	99	135
2	3269	1961	2562	775	219	619
3	7957	4012	3747	1521	732	837
4	15315	5422	8066	4137	3537	1443
5	28175	25999	65115	40803	22816	24792

**Tabel 2.14 Sumber Makanan Awal Xij setelah di KM**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8585	3158	3989	2742	1153	1037
2	4896	11912	19647	1572	8808	1819
3	31472	5504	6407	4391	1278	2333
4	63831	16958	13427	26234	1578	11321
5	29863	53081	60016	3262	27942	3082

Pada Tabel 2.14 merupakan hasil fungsi inisialisasi sumber makanan yang kemudian di *k-means* untuk menghasilkan *centroid*/sumber makanan awal baru yang telah di-KM, dan hasil KM berupa *cost function*-nya.

**Tabel 2.15 Sumber Makanan Vj**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8585	4832	7405	2668	5673	477
2	3211	10643	26410	3424	8924	2358
3	14900	7199	7944	4400	3920	1416
4	77082	9134	21519	22725	3648	12357
5	40120	8031	28342	4877	3460.2	2047

**Tabel 2.16 Sumber Makanan Vj setelah di KM**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8960	3032	3798	2782	1026	1029
2	5589	11583	18071	1593	7945	1972
3	34858	5124	6011	6282	977	2171
4	74499	36789	19159	26639	2594	28247
5	25603	43461	61472	2636	29974	2709

Pada Tabel 2.15 telah dikerjakan fase *employed bee* yaitu mencari sumber makanan tetangga disekitar Xij berdasarkan persamaan (4) dan perubahannya setelah di KM pada Tabel 2.16

Pada Fase *Onlooker Bee* dikerjakan fungsi probabilitas yang membandingkan nilai *fitness* terhadap *cost function*  $x_{ij}$  dan  $v_{ij}$  disetiap dimensinya. Kemudian ketika gabungan *centroid*-nya telah dibandingkan, *centroid* tersebut akan menjadi  $x_{ij}$  baru seperti pada Tabel 2.17 yang akan dikerjakan fungsi *Employed Bee* lagi yaitu mencari  $v_{ij}$  seperti pada Tabel 2.18 dan akan dibandingkan lagi probabilitas nilai *fitness*-nya.

**Tabel 2.17 Sumber Makanan Xij baru**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8585	3158	3989	2742	1153	1037
2	4896	11912	19647	1572	8808	1819
3	31472	5504	6407	4381	1278	2333
4	74499	36789	19159	26639	2594	28247
5	29863	53081	60016	3262	27942	3082

**Tabel 2.18 Sumber Makanan Vij baru**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8859	3092	6049	906	2486	4168
2	5336	12477	9042	4357	8651	905
3	12441	6663	7603	6810	2467	8358
4	71871	29367	12664	17177	114	26100
5	32007	42250	86340	3448	10069	504

Fase fase keempat yaitu *Scout Bee* Hasil centroid gabungan yang terbaik adalah *centroid* yang akan disimpan pada SumberMakananTerbaikSaatIni seperti Tabel 2.19.

Pada fase *scout bee* menerima masukan CMJ yang merupakan *centroid* yang memiliki probabilitas *fitness* yang buruk seperti pada Tabel 2.20. *Centroid*/Sumber makanan tersebut akan ditinggalkan dan diganti dengan hasil fungsi inialisasi sumber makanan baru pada iterasi berikutnya hingga jumlah sumber makanan terbaik mencapai MCN.

**Tabel 2.19 Sumber Makanan terbaik saat ini**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8859	3177	4063	2716	1171	1048
2	5336	12131	19620	1631	8732	1976
3	34644	5205	6147	6223	1039	2234
4	73547	20365	14507	19627	1679	14954
5	29863	53081	60016	3262	27942	3082

**Tabel 2.20 Sumber Makanan terburuk saat ini**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	8908	3004	3731	2807	1001	1027
2	5121	11231	17915	1528	7939	1764
3	32804	5741	6306	5811	1119	2464
4	74499	36789	19159	26639	2594	28247
5	25603	43461	61472	2636	29974	2709

Optimasi merupakan suatu proses untuk mendapatkan solusi yang paling optimal dari suatu permasalahan yang memiliki nilai tujuan maksimal atau minimal, tergantung mana yang diinginkan, tanpa melanggar batasan yang sebelumnya telah ditentukan. Penggunaan KM kerap menghasilkan solusi yang optimal pada kandidat solusi di tetangga terdekatnya saja bukan keseluruhan dari semua solusi yang ada atau yang biasa disebut global optima. Sehingga dengan menggunakan ABC akan membantu agar tidak terjebak dalam lokal optima seperti pada Tabel 2.21.

**Tabel 2.21 Sumber Makanan Optimal**

K	Fresh	Milk	Grocery	Frozen	Detergents Paper	Delicatessen
1	10296.9	2626.8	3078.9	3111.6	669.5	1004.4
2	6991.7	7762.1	12107.1	1371.3	4868.2	1588.8
3	42221.7	6968.1	6450.5	8972.8	895.6	3979.3
4	6500.9	17108.1	26124.8	1896.4	11880.7	2465.1
5	25603	43460.6	61472.2	2636	29974.2	2708.8

Pada algoritma ABCKM akan dihasilkan sumber-sumber makanan terbaik yang telah disimpan dari fase *Onlooker* pada setiap iterasinya yang kemudian akan dipilih sumber optimalnya dengan cara yang sama yaitu berdasarkan probabilitas nilai *fitness* yang terbesar. Sehingga hasil terbaik tersebut akan di-KM untuk terakhir kalinya untuk mendapatkan hasil pembagian data/klasteringnya seperti pada Tabel 2.22.

**Tabel 2.22 Hasil Klasifikasi data Wholesale**

Klaster	Jumlah masing-masing klasifikasi data <i>wholesale</i>
1	256
2	111
3	36
4	32
5	5

## 2.4 Silhouette

Algoritma *Silhouette* berperan besar dalam klastering, *Silhouette* merupakan algoritma yang mengenali kualitas pembagian data. Nilai koefisien *silhouette* berada pada rentang -1 hingga 1 dimana semakin mendekati -1 berarti hasil pembagian

data tersebut buruk, dan sebaliknya. Nilai koefisien *silhouette* didapatkan dari

$$s_i = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (2.9)$$

Dimana:

- $i$  = Obyek data ke- $i$
- $a$  = Rata-rata jarak  $i$  terhadap obyek lain pada klaster yang sama.
- $b$  = Nilai minimum dari rata-rata jarak  $i$  terhadap obyek lain pada klaster yang berbeda.
- $s_i$  = Nilai koefisien *silhouette*

*(Halaman ini sengaja dikosongkan)*



### BAB III DESAIN DAN PERANCANGAN PERANGKAT LUNAK

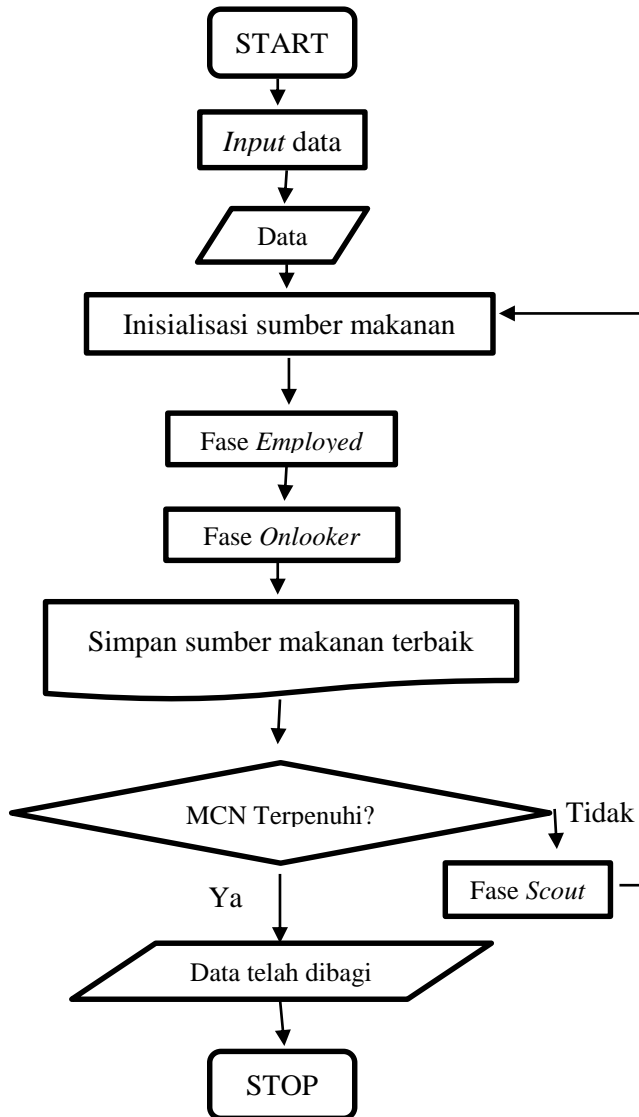
Bab ini akan membahas mengenai perancangan dan pembuatan sistem perangkat lunak (*software*). Sistem perangkat lunak yang dikembangkan dalam Tugas Akhir ini adalah implementasi penyelesaian permasalahan pembagian data dengan menggunakan metode *Artificial Bee Colony* dan *K-means* (ABCKM).

#### 3.1 Metode ABCKM Secara Umum

Pada ABCKM, *K-means* akan mengerjakan semua permasalahan pembagian data pada setiap fase yang dikerjakan, mulai dari setelah tahap inialisasi, kemudian ketiga fase *ABC* yaitu *Employed*, *Onlooker*, juga *Scout Bee*, serta menyimpan sumber makanan terbaik disetiap iterasinya hingga mencapai limit *Maximum Cycle Number*.

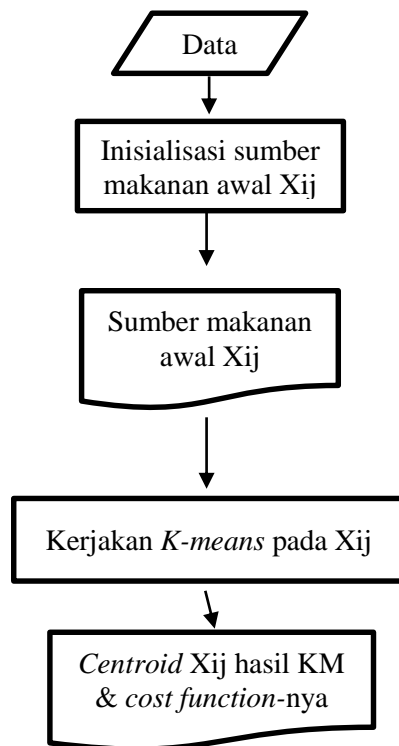
Hasil dari ABCKM adalah sekumpulan sumber makanan yang akan dibandingkan probabilitas nilai kecocokannya, sumber makanan dengan nilai probabilitas tertinggi akan menjadi sumber makanan optimal. Sumber makanan optimal tersebut akan di *K-means* terakhir kali untuk mengetahui dimanakah suatu data terbagi pada tiap-tiap klasternya.

Berikut akan dijelaskan metode ABCKM dengan menggunakan diagram alir yang akan mengilustrasikan langkah-langkah *Artificial Bee Colony K-means* secara umum yang akan dijelaskan pada Gambar 3.1. Seperti yang diketahui ABCKM memiliki 4 fase utama selain masukan dan hasil keluarannya yaitu fase inialisasi sumber makanan, *employed bee*, *onlooker bee* dan *scout bee* keempat fase tersebut akan dijelaskan secara terpisah dan terperinci di diagram berikutnya.



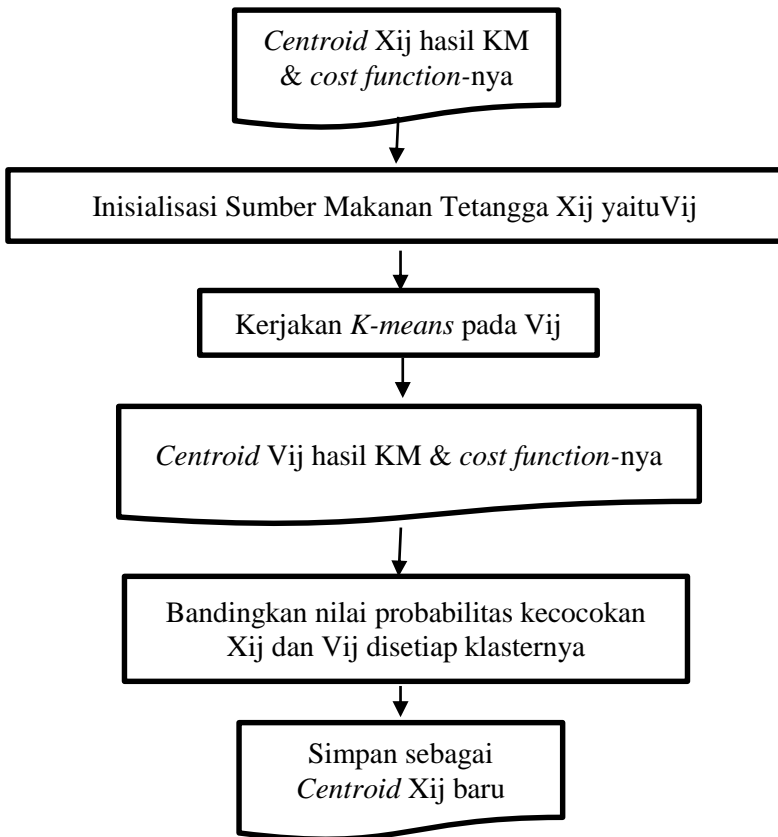
**Gambar 3.1 Diagram Alir ABCMK**

Proses P1 atau fase inisialisasi sumber makanan menerima masukan data dan jumlah klaster, fase ini mengerjakan penentuan inisialisasi titik pusat, berdasarkan upper dan lower bound pada tiap dimensi. Nilai batas pada setiap dimensi dan klasternya akan dikalikan bilangan acak untuk menjadi sumber makanan  $X_{ij}$  sesuai persamaan 1. Langkah selanjutnya adalah menghitung hasil KM dan menyelesaikan permasalahan klustering pada  $X_{ij}$ . Fase ini dijelaskan seperti Gambar 3.2



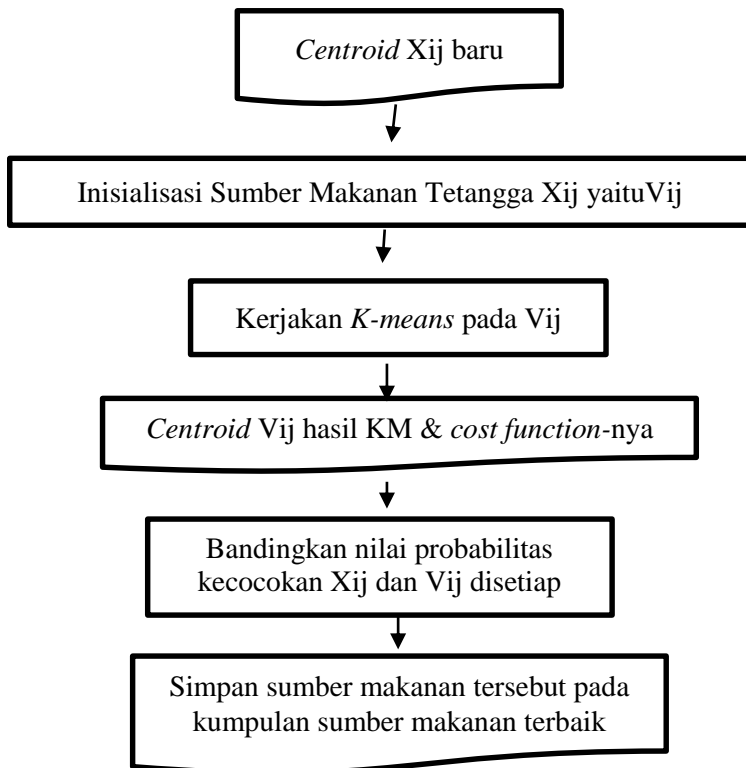
**Gambar 3.2 Diagram Alir Fase Inisialisasi Sumber Makanan**

Proses P2 atau fase *employed bee* menerima masukan sumber makanan awal yang akan dicari sumber baru disekitar sumber tersebut sesuai persamaan 4, yang kemudian akan di KM keduanya untuk dibandingkan nilai probabilitasnya seperti persamaan 2 dan 3 dan dikirim informasinya untuk dikerjakan *Onlooker bee* pada fase berikutnya. Fase ini dijelaskan seperti Gambar 3.3



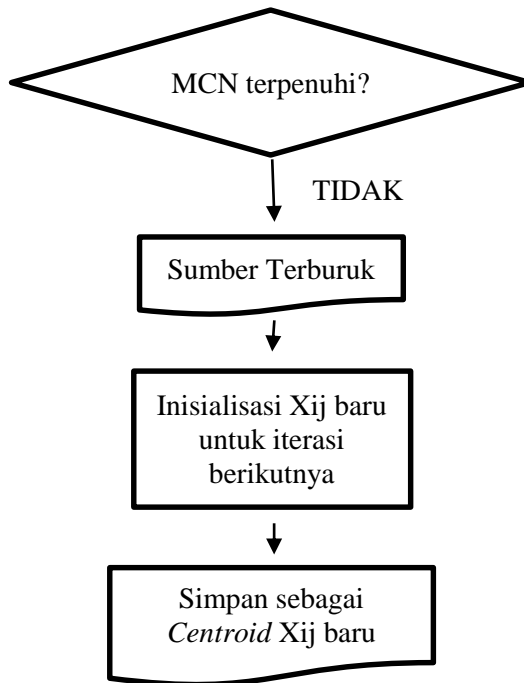
Gambar 3.3 Diagram Alir Fase *Employed Bee*

Proses P3 atau fase *onlooker bee* menerima masukan sumber makanan dengan nilai probabilitas *fitness*-nya yang paling baik pada setiap klasternya, yang kemudian sumber ini akan dicari sumber tetangganya sama seperti yang dilakukan pada fase *employed bee* hanya saja hasil sumber makanan berprobabilitas tertinggi pada tiap klasternya akan disimpan dan sumber buruknya akan menjadi masukan fase berikutnya. Fase ini dijelaskan seperti Gambar 3.4



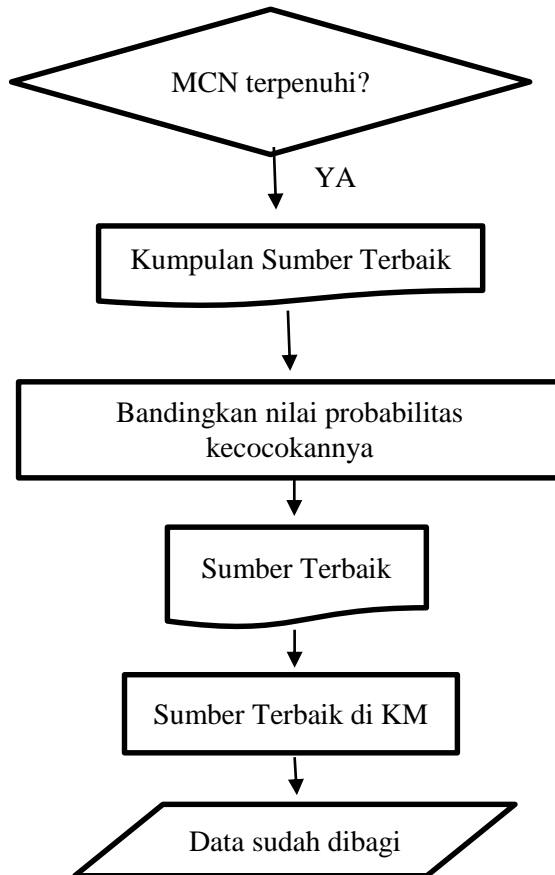
**Gambar 3.4 Diagram alir fase *Onlooker Bee***

Proses P4 atau fase *scout bee* menerima masukan sumber makanan buruk berdasarkan apa yang dikerjakan *onlooker bee*, fase ini hanya dikerjakan apabila kumpulan sumber makanan terbaik atau siklus iterasi belum mencapai *Maximum Cycle Number*, fungsi fase ini adalah mengerjakan fungsi inialisasi sumber makanan baru untuk dikerjakan pada iterasi berikutnya. Fase ini dijelaskan seperti Gambar 3.5



**Gambar 3.5** Diagram alir fase *Scout Bee*

Setetika limit telah dipenuhi hasil dari ABCKM adalah sumber-sumber makanan terbaik sejumlah *limit* seperti persamaan 6, kumpulan sumber tersebut akan dipilih nilai probabilitas kecocokan terbesarnya yang akan menjadi sumber terbaik dari yang paling baik. Langkah optimasi ABCKM ini dijelaskan pada Gambar 3.6



**Gambar 3.6 Diagram alir Optimasi ABCKM**

*(Halaman ini sengaja dikosongkan)*



## **BAB IV IMPLEMENTASI**

Bab ini menjelaskan mengenai implementasi Tugas Akhir yang meliputi algoritma dan kode sumber yang digunakan untuk melakukan proses penyelesaian permasalahan pembagian data dengan menggunakan algoritma ABCKM pada perangkat MATLAB.

### **4.1. Lingkungan Implementasi**

Implementasi penyelesaian permasalahan pembagian data dengan menggunakan algoritma ABCKM, menggunakan perangkat keras komputer dengan prosesor yang mempunyai spesifikasi Intel(R) Core(TM) i5 – 2467M CPU @ 1.60GHz (4 CPUs), ~1.60GHz yang mempunyai memori sebesar 4.0 GB. Sistem operasi yang digunakan adalah Windows 8.1 Pro 64-bit, dalam melakukan implementasi program menggunakan perangkat lunak Matlab R2008a.

### **4.2 Implementasi**

Implementasi sistem perangkat lunak penyelesaian permasalahan pembagian data dengan menggunakan algoritma ABCKM dilakukan sesuai dengan proses yang sebelumnya telah dijelaskan pada bab-bab sebelumnya. Program utama ABCKM akan dijelaskan dan dijabarkan lebih lanjut pada subbab berikutnya.

#### **4.2.1 Implementasi Program Utama**

Tahapan ini dilakukan fungsi utama dari implementasi ABCKM yaitu menentukan nama data *file* dengan format TXT yang terdiri dari sebuah kolom *header* diatas semua data yang akan di kerjakan. ditunjukkan oleh Kode Sumber 4. 1.

1	<code>%% inisialisasi data masukan</code>
2	<code>Data = inputDatas(namaFile);</code>
3	<code>%% inisialisasi sumber makanan awal</code>
4	<code>SumberMakananAwal = inisialisasi(Data, JumlahKlaster);</code>
5	<code>%% pengerjaan K-means biasa untuk dibandingkan hasilnya dengan ABCKM</code>
6	<code>[hasilKMT, hasilCentroid, hasilIndexKlasterO ptimal] = kMeans(Data, SumberMakananAwal, JumlahKlaster);</code>
7	<code>%% pengerjaan ABCKM dengan memulai iterasi 1 dst didalam fungsi</code>
8	<code>[SumberTerbaik, probabilitasTerbaik] = panggilABC(Data, SumberMakananAwal, JumlahKlaster);</code>
9	<code>%% evaluasi hasil pembagian data ABCKM</code>
10	<code>si = siluet(Data, probabilitasTerbaik, SumberTerbaik, JumlahKlaster);</code>
11	<code>%% menampilkan hasil siluet</code>
12	<code>hasilSiluet = si;</code>
13	<code>hasilSiluet</code>

### Kode Sumber 4. 1 Implementasi Program Utama

#### 4.2.2 Implementasi Inisialisasi Data Masukan

Tahapan ini dilakukan inisialisasi data awal, yaitu untuk menentukan data masukan awal, beserta praprosess datanya, ditunjukkan oleh Kode Sumber 4. 2.

1	<code>%% inisialisasi data masukan</code>
	<code>function [dataBagus] = inputDatas(namaFile)</code>
2	<code>disp('Kerjakan fungsi input Data');</code>
3	<code>%input data dan inisialisasinya</code>
4	<code>namaDsata = namaFile;</code>
5	<code>delimiterIn = ',';</code>
6	<code>headerline=1;</code>

7	bukaData = importdata (namaData,delimiterIn,headerline);
8	% mengambil data saja tanpa header
9	bD = bukaData.data;
10	[sizeB sizeK] = size(bD);
11	% mengambil data tanpa kolom id/hasil/kategorikal
12	if strcmp(namaFile,'wholesale.txt')==1
13	bD = bD(:,3:sizeK);
14	elseif strcmp(namaFile,'iris.txt')==1
15	bD = bD(:,1:sizeK);
16	elseif strcmp(namaFile,'dow jones index.txt')==1
17	bD = bD(2:sizeB,4:sizeK);
18	elseif strcmp(namaFile,'birthanddeathrates.txt') ==1   strcmp(namaFile,'crimerateamerica.txt') ==1
19	bD = bD(2:sizeB,2:sizeK);
20	end
21	% merubah nilai kosong menjadi 0
22	bD(isnan(bD))= 0;
23	% simpan data
24	dataBagus = bD;
25	end

**Kode Sumber 4. 2 Implementasi Inisialisasi Data**

### 4.2.3 Implementasi Inisialisasi Sumber Makanan

Tahapan ini dilakukan inisialisasi sumber makanan, pada kasus ini adalah sumber makanan awal. Perhitungan dimulai dengan membagi total data dengan jumlah klaster setelah diurutkan pada masing-masing dimensi data. Langkah berikutnya adalah menjumlahkan batas terbawah pada tiap klaster untuk masing-masing dimensi data dengan angka acak yang dibangkitkan secara

distribusi normal 0 hingga 1 yang sebelumnya akan dikalikan dengan pengurangan dari batas teratas dan batas terbawahnya. Ditunjukkan pada Kode Sumber 4.3.

1	<code>function sumberMakananAwal = inisialisasi(data, jumlahKlaster)</code>
2	<code>disp('Kerjakan fungsi inisialisasi');</code>
3	<code>[m n] = size(data);</code>
4	<code>a=[];</code>
5	<code>k=[];</code>
6	<code>% Mod digunakan untuk mengetahui batas baris masing-masing kolom</code>
7	<code>batasCentroid = ceil(m./jumlahKlaster);</code>
8	<code>for s = 1:m</code>
9	<code>    if mod(s, batasCentroid)</code>
10	<code>        continue</code>
11	<code>    end</code>
12	<code>    a=[a;s];</code>
13	<code>end</code>
14	<code>a=[1;a;m];</code>
15	<code>a=unique(a, 'rows');</code>
16	<code>% inisialisasi titik pusat berdasarkan upper dan lower bound tiap dimensi data pada masing-masing klasternya</code>
17	<code>for j=1:n</code>
18	<code>    bagiKlaster = data(:, j);</code>
19	<code>    bagiKlaster = sortrows(bagiKlaster);</code>
20	<code>    i=1;</code>
21	<code>    while i&lt;length(a)</code>
22	<code>        r = rand();</code>
23	<code>        c = bagiKlaster(a(i))+round(r*(bagiKlaster(a(i+1))-bagiKlaster(a(i))));</code>
24	<code>        i=i+1;</code>
25	<code>        k=[k;c];</code>
26	<code>    end</code>
27	<code>end</code>

28	CentroidBaru=reshape(k, [jumlahKlaster,n]) ;
29	% hasil inisialisasi berupa sumber makanan baru yang terdiri dari titik pusat untuk setiap dimensi data pada masing-masing klaster
30	sumberMakananAwal = CentroidBaru;
31	end

### Kode Sumber 4. 3 Implementasi Inisialisasi Sumber Makanan

#### 4.2.4 Implementasi Algoritma K-means

Tahapan ini akan dilakukan perhitungan untuk menentukan hasil pembagian data menggunakan algoritma K-means. Hasil dari fungsi ini adalah hasil KM yang merupakan nilai jarak Euclidean antara data dengan titik pusatnya, kemudian hasil sumber makanan baru dan jumlah masing-masing index data terhadap titik pusatnya. Implementasinya dapat dilihat pada Kode Sumber 4.4 dan Kode Sumber 4.5

	<code>function</code>
1	<code>[hasilKM, sumberMakananAwal, indexKlaster] = kMeans (data, sumberMakananAwal, K)</code>
2	<code>disp('Kerjakan fungsi Kmeans');</code>
3	<code>[sizeR sizeD] = size(data);</code>
4	<code>[sizeB sizeN] = size(sumberMakananAwal);</code>
5	<code>costFunction = [];</code>
6	<code>calonCentroid=[];</code>
7	<code>indexK=[];</code>
8	<code>% Menghitung nilai jarak euclidean atau cost function untuk K-means</code>
9	<code>for j = 1:sizeB</code>
10	<code>    for k = 1:sizeR</code>
11	<code>        a = data(k, :) - sumberMakananAwal(j, :);</code>
12	<code>        a = sqrt(sum(a, 2).^2);</code>

13	costFunction=[costFunction;a];
14	end
15	end
16	costFunction = reshape(costFunction, [sizeR,K]);
17	barisKosongC = zeros(size(sumberMakananAwal(1,1:sizeN)));
18	% Mencatat di klaster mana setiap data terletak setelah K-means
19	for j = 1:K
20	for i = 1:sizeR
21	x = min(costFunction(i,:));
22	if costFunction(i,j)~=x
23	indexK = [indexK;0];
24	calonCentroid = [calonCentroid;barisKosongC];
25	else
26	indexK = [indexK;i];
27	calonCentroid = [calonCentroid;data(i,:)];
28	end
29	end
30	end
31	indexK = reshape(indexK, [sizeR,K]);
32	% menjaga keutuhan dan keteraturan calon sumber makanan baru terhadap yang sumber yang lama
33	calonCentroid = reshape (calonCentroid, [sizeR,sizeD*K]);
34	jumlahIndex = (sum(indexK~=0)');
35	ji = ones(size(1:sizeD));
36	jumlahIndex = jumlahIndex*ji;
37	jumlahIndex = reshape(jumlahIndex, [1,sizeD*K]);
38	centroid = sum(calonCentroid);
39	centroid(isnan(centroid))= 0;
40	centroid = (centroid./jumlahIndex);
41	centroid(isnan(centroid))= 0;

42	centroid = reshape (centroid, [K, sizeD]);
43	x=[];
44	for j=1:sizeD
45	temp = centroid(:,j);
46	temp = sortrows(temp);
47	x=[x;temp];
48	end
49	centroid = reshape (x, [K, sizeD]);
50	hasilKM = costFunction;
51	sumberMakananAwal = centroid;
52	indexKlaster = indexK;
53	end

**Kode Sumber 4.4 Implementasi *K-means*  
untuk Sekali Iterasi**

1	<code>function</code> [HasilKMA, HasilCentroid, IndexKlaster] = panggilKM(data, centroidAsli, jumlahK)
2	<code>% Bandingkan apabila nilai titik pusat yang dihasilkan sudah tidak berubah lagi</code>
3	[HasilKM, centroidLama] = kMeans (data, centroidAsli, jumlahK);
4	i=1;
5	<code>while</code> 1
6	[HasilKMAkhir, centroidBaru, indexK] = kMeans (data, centroidLama, jumlahK);
7	if centroidLama - centroidBaru == 0
8	break
9	else
10	centroidLama = centroidBaru;
11	i=i+1;
12	end
13	<code>end</code>
14	HasilKMA = HasilKMAkhir;
15	HasilCentroid = centroidBaru;
16	IndexKlaster = indexK;

**Kode Sumber 4.5 Implementasi *K-means***

#### 4.2.5 Implementasi *ABCKM* pada Fase *Employed Bee*

Fase tahapan ini dimulai dengan mengetahui jumlah separuh koloni adalah *Employed Bee* yang bertugas mencari sumber makanan disekitar tetangga dari sumber makanan yang sedang digunakan saat ini pada tiap klasternya. Dimana hasilnya adalah sumber makanan baru yang telah di-KM beserta hasil KM-nya yang juga akan dikelola oleh fase berikutnya. Fase ini dapat dilihat pada Kode Sumber 2.6.

1	<code>function [HasilKMEmployed, sumberBaru] = employed(data,centroidE,jumlahK)</code>
2	<code>disp('Kerjakan fungsi employed Bee');</code>
3	<code>[m n] = size(data);</code>
4	<code>% Jumlah separuh koloni (employed bee) diketahui sejumlah kluster</code>
5	<code>[employedB p] = size(centroidE);</code>
6	<code>% Bangkitkan angka acak sejumlah sumber makanan, berdistribusi uniform -1,1</code>
7	<code>angkaAcak = unifrnd(-1,1,[employedB p]);</code>
8	<code>centroidAcak = [];</code>
9	<code>a = 1;</code>
10	<code>b = employedB;</code>
11	<code>% Bangkitkan angka acak berdistribusi normal 0,1 untuk mendapatkan index tetangga dari sumber makanan</code>
12	<code>j = 1;</code>
13	<code>limit = p*employedB;</code>
14	<code>for i = 1 : employedB</code>
15	<code>    for k = 1:p</code>
16	<code>        while j &lt;= limit</code>
17	<code>            ran = rand();</code>
18	<code>            r = round((b-a).*ran)+a;</code>
19	<code>            c = centroidE(r,k);</code>
20	<code>            if c &lt; centroidE(i,k)</code>
21	<code>                centroidAcak =                 [centroidAcak;c];</code>



22	<code>j=j+1;</code>
23	<code>elseif c &gt; centroidE(i,k)</code>
24	<code>centroidAcak = [centroidAcak;centroidE(i,k)];</code>
25	<code>j=j+1;</code>
26	<code>else</code>
27	<code>centroidAcak = [centroidAcak;centroidE(r,k)];</code>
28	<code>j=j+1;</code>
29	<code>end</code>
30	<code>end</code>
31	<code>end</code>
32	<code>end</code>
33	<code>centroidAcak = reshape(centroidAcak,employedB,p);</code>
34	<code>% Tetangga sumber didapatkan dari menambahkan sumber makanan Employed dengan mengalikan angka acak dengan selisih dari sumber Employed dengan centroid acak</code>
35	<code>tetanggaSumberMakanan = abs(centroidE+(angkaAcak.*(centroidE-centroidAcak)));</code>
36	<code>[HasilKME , centroidBaru] = panggilKM(data,tetanggaSumberMakanan,jumlahK);</code>
37	<code>% didapatkan tetangga dari sumber makanan dan nilai hasil KMnya</code>
38	<code>HasilKMEmployed = HasilKME;</code>
39	<code>sumberBaru = centroidBaru;</code>
40	<code>end</code>

**Kode Sumber 2.6 Implementasi ABCKM pada Fase *Employed Bee***

#### 4.2.6 Implementasi *ABCKM* pada Fase *Onlooker Bee*

Tahapan ini dimulai dengan membandingkan nilai probabilitas kecocokan tiap klaster dari sumber makanan awal dengan sumber yang dihasilkan *employed bee*, nilai probabilitas kecocokan tersebut dihitung berdasarkan hasil *euclidean distance* setiap data pada sumber makanan tersebut. Hasil dari fungsi probabilitas ini yaitu sumber gabungan terbaik yang akan dikerjakan lagi fungsi *employed bee*-nya yaitu mencari tetangga dari sumber makanannya, dimana hasil KM dari probabilitas kedua sumber baru ini akan dibandingkan dan hasil fungsi probabilitas ini adalah nilai probabilitas kecocokan terbaik pada masing-masing klaster. Sumber terbaik pada iterasi inilah yang akan disimpan, dan sumber buruknya akan dibuang. Fase *onlooker bee* memiliki dua fungsi utama yaitu fase *onlooker bee* itu sendiri dan probabilitas dapat dilihat pada Kode Sumber 7 dan Kode Sumber 8.

1	<code>function [SMO, SMJ, probabilitasBagus] = probablitas (cLama, cBaru, HKMBaru, HKMLama)</code>
2	<code>disp('Kerjakan fungsi Probabilitas');</code>
3	<code>[s t] = size(HKMBaru);</code>
4	<code>[u v] = size(HKMLama);</code>
5	<code>pi = [];</code>
6	<code>pj = [];</code>
7	<code>ps = [];</code>
8	<code>% Menghitung nilai kecocokan dari hasil KM yang berupa nilai euclidean distance setiap data pada sumber makanannya</code>
9	<code>k = 1;</code>
10	<code>l = 1;</code>
11	<code>while k &lt;= t &amp;&amp; l &lt;= v</code>
12	<code>    if (HKMLama(l) &gt;= 0    HKMBaru(k) &gt;= 0)</code>
13	<code>        HKMLama(l) = 1/(1+HKMLama(l));</code>
14	<code>        HKMBaru(k) = 1/(1+HKMBaru(k));</code>
15	<code>    else</code>
16	<code>        HKMLama(l) = 1 + abs(HKMLama(l));</code>
17	<code>        HKMBaru(k) = 1 + abs(HKMBaru(k));</code>

18	End
19	k = k+1;
20	l = l+1;
21	End
22	% Menghitung probabilitas sumber baru dengan sumber lama berdasarkan nilai kecocokannya
23	totalHKMLama = sum(HKMLama);
24	totalHKMBaru = sum(HKMBaru);
25	pvi = totalHKMBaru./sum(totalHKMBaru,2)';
26	pxi = totalHKMLama./sum(totalHKMLama,2)';
27	% Membandingkan nilai probabilitas sumber baru dengan sumber yang lama pada tiap klasternya
28	i = 1;
29	j = 1;
30	while i <= t && j <= v
31	if pvi(i)>pxi(j)
32	pi = [pi;cBaru(i,:)];
33	pj = [pj;cLama(j,:)];
34	ps = [ps;pxi(i)];
35	Else
36	pi = [pi;cLama(j,:)];
37	pj = [pj;cBaru(i,:)];
38	ps = [ps;pvi(j)];
39	End
40	i = i+1;
41	j = j+1;
42	End
43	% Mencatat hasil fungsi probabilitas yaitu nilai probabilitas terbaik pada masing-masing klaster
44	probabilitasBagus = ps;
45	sumberMakananProbabilitasTinggi = pi;
46	sumberMakananProbabilitasBuruk = pj;
47	% Mencatat hasil fungsi probabilitas yaitu sumber terbaik yang akan disimpan
48	SMO = sumberMakananProbabilitasTinggi;

49	% Mencatat hasil fungsi probabilitas yaitu sumber terburuk yang akan dibuang
50	SMJ = sumberMakananProbabilitasBuruk;
51	End

**Kode Sumber 4.7 Implementasi probabilitas untuk Fase  
*Onlooker Bee***

1	<code>function [cTerbaik, cNextIteration, pBest] = onlooker (data, cBaru, cEmployed, HKME, HKM, jumlahK)</code>
2	<code>disp('Kerjakan fungsi Onlooker Bee');</code>
3	<code>% Menghitung nilai probabilitas sumber makanan terbaru dengan sumber employed yang terakhir</code>
4	<code>[BeeMOnlooker] = probablitas (cBaru, cEmployed, HKME, HKM);</code>
5	<code>% Hasil probabilitas yang baik akan dikerjakan KM-nya</code>
6	<code>[HKMOnlooker, centroidOnlooker] = panggilKM (data, BeeMOnlooker, jumlahK);</code>
7	<code>% Mengulang fungsi Employed yaitu mencari tetangga dari sumber makanan hasil dari probabilitas terbaik yang diketahui pada iterasi saat ini</code>
8	<code>[HKMEmployed, centroidEmployed] = employed (data, centroidOnlooker, jumlahK);</code>
9	<code>% Probabilitasnya kembali dibandingkan</code>
10	<code>[CentroidMO, CentroidMJ] = probablitas (centroidOnlooker, centroidEmployed, HKMOnlooker, HKMEmployed);</code>
11	<code>[HKMCMO] = panggilKM (data, CentroidMO, jumlahK);</code>
12	<code>[sizeData sizeK] = size (HKMCMO);</code>
13	<code>pc = [];</code>
14	<code>k = 1;</code>
15	<code>% Menghitung nilai kecocokan dari hasil KM yang berupa nilai euclidean distance setiap data pada sumber makanan</code>
16	<code>while k &lt;= sizeK</code>

17	<code>if (HKMCMO (k) &gt;= 0)</code>
18	<code>HKMCMO (k) = 1/(1+HKMCMO (k));</code>
19	<code>Else</code>
20	<code>HKMCMO (k) = 1 + abs (HKMCMO (k));</code>
21	<code>End</code>
22	<code>k = k+1;</code>
23	<code>End</code>
24	<code>totalHKMCMO = sum(HKMCMO);</code>
25	<code>pvi = totalHKMCMO./sum(totalHKMCMO,2)';</code>
26	<code>pc = [pc;pvi];</code>
27	<code>% Probabilitas terbaik tersebut akan disimpan bersama sumber makanan terbaik dan sumber yang sebagian dibuang</code>
28	<code>pBest = pc;</code>
29	<code>cTerbaik = CentroidMO;</code>
30	<code>cNextIteration = CentroidMJ;</code>
31	<code>End</code>

**Kode Sumber 4.8 Implementasi *ABCKM* pada  
Fase *Onlooker Bee***

#### 4.2.7 Implementasi *ABCKM* pada Fase *Scout Bee*

Tahapan ini dimulai dengan melakukan inialisasi sebuah sumber makanan baru untuk dibandingkan dengan sumber makanan yang telah dibuang oleh *onlooker bee*. Fase *scout bee* dapat dilihat pada Kode Sumber 9.

1	<code>function [sumberNextIter] = scout (data, smDitinggalkan, jumlahK)</code>
2	<code>disp('Kerjakan fungsi Scout Bee');</code>
3	<code>d=[];</code>
4	<code>[o p] = size(smDitinggalkan);</code>
5	<code>% Fungsi inialisasi digunakan untuk menghasilkan sumber makanan baru untuk sumber yang ditinggalkan</code>
6	<code>centroidTes = inialisasi (data, jumlahK);</code>

7	<code>% Memilih sumber mana yang akan dibuang dan digunakan untuk iterasi berikutnya</code>
8	<code>for l = 1:o</code>
9	<code>    if smDitinggalkan(l,')==0</code>
10	<code>        d = [d;centroidTes(l, :)];</code>
11	<code>    elseif smDitinggalkan(l, :)~= centroidTes(l, :)</code>
12	<code>        d = [d;centroidTes(l, :)];</code>
13	<code>    Else</code>
14	<code>        d = [d;smDitinggalkan(l, :)];</code>
15	<code>    end</code>
16	<code>end</code>
17	<code>centroidUIterBaru = centroidTes;</code>
18	<code>sumberNextIter = centroidUIterBaru;</code>
19	<code>end</code>

**Kode Sumber 4.9 Implementasi ABCKM pada Fase Scout Bee**

#### 4.2.8 Implementasi *Artificial Bee Colony* menggunakan *K-Means*

Tahapan ini dilakukan implementasi ABCKM pada semua fase, untuk langkah pertama adalah menghitung *limit* sumber makanan terbaik yang akan disimpan yaitu banyaknya dimensi dikalikan banyaknya sumber makanan. Langkah berikutnya adalah memulai iterasi satu yaitu mengerjakan ketiga fase *employed*, *onlooker* dimana hasil terbaik dari fungsi tersebut disimpan dan hasil terburuknya akan diolah pada fase *scout bee*. Iterasi berikutnya dan seterusnya akan dikerjakan hingga sumber terbaik yang disimpan telah mencapai *limit*. Implementasi ABCKM ini dapat dilihat pada 4.10.

1	<code>function [sumberTerbaik,probTerbaik] = panggilABC (data, SMA, K)</code>
2	<code>[sizeK sizeD] = size(SMA);</code>
3	<code>% Tentukan limit banyaknya sumber terbaik yang akan disimpan</code>

4	limit = sizeK * sizeD;
5	% Iterasi 1 ABCKM
6	[HasilKMI, SMAKM] = panggilKM(data, SMA, K);
7	% Kerjakan fungsi Employed Bee
8	[HKMEmployed, centroidEmployed] = employed(data, SMAKM, K);
9	% Kerjakan fungsi Onlooker Bee
10	[CMO, CMJ, Pb] = onlooker(data, SMAKM, centroidEmployed, HKMEmp loyed, HasilKMI, K);
11	ProbabilitasBest = [];
12	ProbabilitasBest = [ProbabilitasBest; Pb];
13	% Simpan sumber makanan terbaik berdasarkan nilai probabilitas nilai kecocokan
14	SumberMakananTerbaikSaatIni = [];
15	CMOs = CMO';
16	CMOs = reshape(CMOs, [1, limit]);
17	SumberMakananTerbaikSaatIni = [SumberMakananTerbaikSaatIni; CMOs];
18	SMCLama = scout(data, CMJ, K);
19	count = 1;
20	i=1;
21	% iterasi berikutnya hingga limit tercapai
22	while i
23	[HasilKMIter, SMCLamaKM] = panggilKM(data, SMCLama, K);
24	[HKMEmployedIter, centroidEmployedIter] = employed(data, SMCLamaKM, K);
25	[CMOIter, CMJIter, PbIter] = onlooker(data, SMCLamaKM, centroidEmployedIte r, HKMEmployedIter, HasilKMIter, K);
26	ProbabilitasBest = [ProbabilitasBest; PbIter];
27	CMOIter = CMOIter';
28	CMOIter = reshape(CMOIter, [1, limit]);
29	% Simpan sumber makanan terbaik hingga jumlah limit untuk sumber makanan tercapai

30	SumberMakananTerbaikSaatIni [SumberMakananTerbaikSaatIni;CMOIter];	=
31	[sizeC                    sized] size(SumberMakananTerbaikSaatIni);	=
32	SMCBaru = scout(data,CMJIter,K);	
33	if sizeC>limit-1	
34	break;	
35	else	
36	SMCLama = SMCBaru;	
37	end	
38	count=count+1;	
39	end	
40	ProbabilitasBest reshape(ProbabilitasBest,[sizeK,sizeD]);	=
41	probTerbaik = ProbabilitasBest;	
42	sumberTerbaik SumberMakananTerbaikSaatIni;	=
43	end	

**Kode Sumber 4.10 Implementasi Artificial Bee Colony  
menggunakan K-means**



## **BAB V**

### **HASIL UJI COBA DAN EVALUASI**

Bab ini akan menjelaskan mengenai rangkaian pengujian dan evaluasi nilai koefisien *silhouette* pada metode ABCKM untuk menyelesaikan permasalahan pembagian data dengan hasil perhitungan *K-means*. Pembahasan ini meliputi lingkungan uji coba, skenario uji coba, hasil uji coba dan evaluasi uji coba yang dilakukan.

#### **5.1 Lingkungan Pengujian**

Uji coba dan evaluasi Tugas Akhir ini menggunakan perangkat keras komputer dengan prosesor yang mempunyai spesifikasi Intel(R) Core(TM) i5 – 2467M CPU @ 1.60GHz (4 CPUs), ~1.60GHz yang mempunyai memori sebesar 4.0 GB. Sistem operasi yang digunakan adalah Windows 8.1 Pro 64-bit, dalam melakukan implementasi program menggunakan perangkat lunak Matlab R2008a, serta Microsoft Office Excel 2013 dalam pengolahan beberapa tabel yang terdapat pada Tugas Akhir ini.

#### **5.2 Data Pengujian**

Data masukan yang digunakan pada uji coba Tugas Akhir ini adalah beberapa *dataset* yang didapatkan dari berbagai sumber. Penjelasan mengenai macam-macam dataset yang digunakan telah dijelaskan pada Bab 1.

Data akan direpresentasikan kedalam suatu file dengan format *text* (txt). Pada Tabel 5.1 akan dijelaskan karakteristik masing-masing kelima *dataset* tersebut mulai dari jumlah data, jumlah fitur hingga jenis data seperti berikut:

**Tabel 5.1 Tabel Karakteristik Kelima Dataset**

Nama Dataset	Jumlah Fitur	Jumlah Data	Jenis Data
Wholesale Costumer	440	6	Numerik
Crime Rates	16	7	Numerik
Birth and Death Rates	70	2	Numerik
Iris	150	4	Numerik
Dow Jones Index	750	16	Numerik

### 5.3 Skenario Uji Coba

Bagian ini akan menjelaskan tentang skenario uji coba program yang telah dikembangkan. Skenario dalam uji coba ini adalah perubahan nilai dari parameter jumlah kluster pada setiap data. Tujuan dari uji coba ini adalah untuk melakukan perubahan pada parameter agar diketahui apakah mempengaruhi atau memberikan dampak pada hasil optimasi.

Pada uji coba ABCKM akan dilakukan skenario perubahan nilai parameter jumlah kluster. Perubahan yang terjadi adalah nilai jumlah kluster atau banyaknya kluster yang digunakan dalam membagi data yaitu mulai dua, tiga, empat, lima dan enam kecuali pada *dataset crimerate* karena metode yang digunakan untuk mencari batasan untuk persamaan 1 menggunakan *mod* dimana jumlahnya pembagian 16 data tidak setara untuk kluster 5.

## 5.4 Uji Kinerja

Tahap ini dilakukan pengujian kinerja program untuk permasalahan pembagian data. Uji kinerja ini dilakukan dengan cara mencari nilai koefisien *silhouette* data setelah diklaster oleh ABCKM, dan KM biasa dengan masing-masing nilai dari parameter jumlah klaster yang berubah-ubah.

### 5.4.1 Hasil Uji Kinerja

Subbab ini menunjukkan perbandingan hasil uji coba ABCKM berupa nilai koefisien hasil pembagian data optimal yang akan dibandingkan dengan hasil pembagian data dengan KM yang digunakan. Berdasarkan berbagai *dataset* dengan parameter jumlah klaster, dimana hasil klaster dikatakan baik menurut algoritma *silhouette* adalah memiliki nilai koefisien diatas 0 dalam batas -1 hingga 1.

**Tabel 5.2 Uji Perbandingan Nilai Parameter Jumlah Klaster pada *dataset crimerate.txt***

No	Parameter Jumlah klaster	Nilai koefisien <i>silhouette</i>	
		<i>ABCKM</i>	<i>KM</i>
1	2	0.615	0.615
2	3	0.460	0.460
3	4	0.398	0.416
4	6	0.445	0.414
Rata-rata Nilai Koefisien		0.479	0.476

Tabel 5.2 menunjukkan perbandingan hasil nilai koefisien *silhouette* pada *dataset crimerate.txt* dimana dapat diketahui bahwa nilai koefisien *silhouette* pada jumlah klaster dua untuk ABCKM dan KM memiliki nilai yang sama dan selain itu ABCKM memiliki nilai koefisien yang semakin menurun setiap kali ditambah jumlah klasternya berarti dua klaster adalah jumlah yang disarankan.

Nilai koefisien *silhouette* KM dan ABCKM kerap bernilai diatas 0 dan dibawah 0.5 yang berarti hasil klastering cukup baik, perbedaan selisih nilai koefisien *silhouette* yang dihasilkan ABCKM pada klaster 4 lebih buruk dibandingkan hasil dari KM sedangkan pada klaster 6 lebih baik melewati fase-fase pada ABCKM. Selisih yang cukup besar tersebut memberikan rata-rata nilai *silhouette* yang lebih baik untuk ABCKM, sehingga ABCKM berhasil mengklaster data dengan cukup baik pada *dataset crimerate*.

**Tabel 5.3 Hasil Uji Perbandingan Nilai Parameter Jumlah Klaster pada *dataset dowjonesindex.txt***

No	Parameter Jumlah klaster	Nilai koefisien <i>silhouette</i>	
		ABCKM	KM
1	2	0.835	0.838
2	3	0.827	0.843
3	4	0.840	0.756
4	5	0.772	0.723
5	6	0.740	0.699
Rata-rata Nilai Koefisien		0.802	0.771

Tabel 5.3 menunjukkan perbandingan hasil nilai koefisien *silhouette* pada *dataset dowjonesindex.txt* dimana parameter jumlah klasternya dapat diketahui bahwa nilai koefisien *silhouette* pada seluruh skenario ABCKM memiliki nilai koefisien tertinggi kecuali pada klaster tiga, dan data telah diklaster dengan baik.

Nilai koefisien *silhouette* KM yang dihasilkan mendekati hasil koefisien *silhouette* ABCKM, hal ini menandakan bahwa data memiliki kemiripan dan ketidakmiripan yang cukup kentara. Hasil KM lebih baik dari ABCKM terutama pada klaster kecil dua dan tiga, namun tidak

pada jumlah kluster empat, lima dan enam yang menandakan ABCKM mampu membagi data lebih baik pada kluster yang lebih besar pada *dataset dowjonesindex*.

**Tabel 5.4 Hasil Uji Perbandingan Nilai Parameter Jumlah Kluster pada *dataset wholesale.txt***

No	Parameter Jumlah kluster	Nilai koefisien <i>silhouette</i>	
		<i>ABCKM</i>	<i>KM</i>
1	2	0.849	0.849
2	3	0.708	0.649
3	4	0.546	0.504
4	5	0.561	0.566
5	6	0.561	0.466
Rata-rata Nilai Koefisien		0.645	0.606

Tabel 5.4 menunjukkan perbandingan hasil nilai koefisien *silhouette* pada *dataset wholesale.txt* dimana parameter jumlah klasternya dapat diketahui bahwa nilai koefisien *silhouette* pada seluruh skenario jumlah kluster ABCKM lebih besar dari KM kecuali pada kluster 5.

Berdasarkan tabel 5.4 data telah dikluster dengan baik, kecuali pada kluster empat, lima dan enam data dikluster cukup baik, hal ini berarti bahwa pada kasus dataset *wholesale* sebaiknya menggunakan jumlah kluster kecil, karena pada kluster-kluster besar nampaknya data memiliki kesamaan diantara klasternya dan ketidaksamaan pada kluster lain yang sangat tipis.

Tabel 5.5 menunjukkan perbandingan hasil nilai koefisien *silhouette* pada *dataset birhanddeathrates.txt* dimana dapat diketahui bahwa nilai koefisien *silhouette* pada seluruh skenario jumlah kluster ABCKM memiliki nilai koefisien lebih tinggi kecuali pada kluster dua yang memiliki nilai koefisien *silhouette* sama dengan KM, dan data telah dikluster dengan

sangat baik karena data memiliki kemiripan disesama klasternya dan ketidakmiripan pada klaster lainnya.

**Tabel 5.5 Hasil Uji Perbandingan Nilai Parameter Jumlah Klaster pada *dataset birthdeathrates.txt***

No	Parameter Jumlah klaster	Nilai koefisien <i>silhouette</i>	
		<i>ABCKM</i>	<i>KM</i>
1	2	0.858	0.858
2	3	0.749	0.683
3	4	0.710	0.648
4	5	0.703	0.703
5	6	0.687	0.687
Rata-rata Nilai Koefisien		0.741	0.715

**Tabel 5.6 Hasil Uji Perbandingan Nilai Parameter Jumlah Klaster pada *dataset iris.txt***

No	Parameter Jumlah klaster	Nilai koefisien <i>silhouette</i>	
		<i>ABCKM</i>	<i>KM</i>
1	2	0.847	0.850
2	3	0.735	0.735
3	4	0.667	0.666
4	5	0.502	0.669
5	6	0.555	0.556
Rata-rata Nilai Koefisien		0.6612	0.6848

Tabel 5.6 menunjukkan perbandingan hasil nilai koefisien *silhouette* pada *dataset iris.txt* dimana dapat diketahui bahwa nilai koefisien *silhouette* pada seluruh klaster ABCKM memiliki hasil yang lebih sedikit dibandingkan KM kecuali pada klaster empat dan tiga yang memiliki nilai sama, data telah

diklaster dengan baik, kecuali pada kluster besar lima dan enam, karena memang perhitungan klasifikasi dan klastering berbeda. Hasil Klastering data iris kluster tiga dapat dilihat pada lampiran A 21.

**Tabel 5.7 Hasil Pengujian *Confusion Matrices* Data Iris**

No.	Variabel	Jumlah Kluster			Total
		1	2	3	
1	<i>True Positives (TP)</i>	50	46	35	131
2	<i>False Positives (FP)</i>	4	15	0	19
3	<i>False Negatives (FN)</i>	0	4	15	19
4	<i>True Negatives (TN)</i>	96	85	100	281

Hasil klastering data iris pada kluster tiga akan dibandingkan dengan klasifikasi data aktual pada setiap klasternya yang diilustrasikan pada Tabel 5.7. Berdasarkan hasil *confusion matrices* tersebut, diperoleh nilai akurasi, *error rate*, presisi, dan *recall* sebagai berikut:

$$Akurasi = \frac{131 + 19}{131 + 19 + 19 + 281} * 100\% \quad (5.1)$$

$$Akurasi = 91.6\% \quad (5.2)$$

$$Error\ rate = \frac{19 + 19}{131 + 19 + 19 + 281} * 100\% \quad (5.3)$$

$$Error\ rate = 8.4\% \quad (5.4)$$

$$Presisi = \frac{131}{131 + 19} \quad (5.5)$$

$$Presisi = 0.87 \quad (5.6)$$

$$\text{Recall} = \frac{131}{131 + 19} \quad (5.7)$$

$$\text{Recall} = 0.87 \quad (5.8)$$



## LAMPIRAN A

### HASIL KLASTERING DATA

**Tabel Lampiran A 1 Hasil Klastering data *crimerate*  
Klaster 2**

Klaster	Jumlah masing-masing pembagian data <i>crimerate</i>
1	6
2	10

**Tabel Lampiran A 2 Hasil Klastering data *crimerate*  
Klaster 3**

Klaster	Jumlah masing-masing pembagian data <i>crimerate</i>
1	6
2	6
3	4

**Tabel Lampiran A 3 Hasil Klastering data *crimerate*  
Klaster 4**

Klaster	Jumlah masing-masing pembagian data <i>crimerate</i>
1	2
2	3
3	6
4	5

**Tabel Lampiran A 4 Hasil Klastering data *crimerate*  
Klaster 6**

Klaster	Jumlah masing-masing pembagian data <i>crimerate</i>
1	4
2	6
3	6
4	0
5	0
6	0

**Tabel Lampiran A 5 Hasil Klastering data *dowjonesindex*  
Klaster 2**

Klaster	Jumlah masing-masing pembagian data <i>dowjonesindex</i>
1	647
2	103

**Tabel Lampiran A 6 Hasil Klastering data *dowjonesindex*  
Klaster 3**

Klaster	Jumlah masing-masing pembagian data <i>dowjonesindex</i>
1	594
2	132
3	24

**Tabel Lampiran A 7 Hasil Klastering data *dowjonesindex*  
Klaster 4**

Klaster	Jumlah masing-masing pembagian data <i>dowjonesindex</i>
1	581
2	140
3	24
4	5

**Tabel Lampiran A 8 Hasil Klastering data *dowjonesindex*  
Klaster 5**

Klaster	Jumlah masing-masing pembagian data <i>dowjonesindex</i>
1	532
2	123
3	69
4	24
5	2

**Tabel Lampiran A 9 Hasil Klastering data *dowjonesindex*  
Klaster 6**

Klaster	Jumlah masing-masing pembagian data <i>dowjonesindex</i>
1	488
2	95
3	134
4	2
5	24
6	13

**Tabel Lampiran A 10 Hasil Klastering data *wholesale*  
Klaster 2**

Klaster	Jumlah masing-masing pembagian data <i>wholesale</i>
1	423
2	17

**Tabel Lampiran A 11 Hasil Klastering data *wholesale*  
Klaster 3**

Klaster	Jumlah masing-masing pembagian data <i>wholesale</i>
1	367
2	59
3	14

**Tabel Lampiran A 12 Hasil Klastering data *wholesale*  
Klaster 4**

Klaster	Jumlah masing-masing pembagian data <i>wholesale</i>
1	285
2	48
3	100
4	7

**Tabel Lampiran A 13 Hasil Klastering data *wholesale*  
Klaster 5**

Klaster	Jumlah masing-masing pembagian data <i>wholesale</i>
1	277
2	100
3	54
4	7
5	2

**Tabel Lampiran A 14 Hasil Klastering data *wholesale*  
Klaster 6**

Klaster	Jumlah masing-masing pembagian data <i>wholesale</i>
1	272
2	57
3	98
4	4
5	8
6	1

**Tabel Lampiran A 15 Hasil Klastering data *birthanddeathrates*  
Klaster 2**

Klaster	Jumlah masing-masing pembagian data <i>birthanddeathrates</i>
1	62
2	8

**Tabel Lampiran A 16 Hasil Klastering data *birthanddeathrates*  
Klaster 3**

Klaster	Jumlah masing-masing pembagian data <i>birthanddeathrates</i>
1	49
2	19
3	2

**Tabel Lampiran A 17 Hasil Klastering data  
*birthanddeathrates* Klaster 4**

Klaster	Jumlah masing-masing pembagian data <i>birthanddeathrates</i>
1	37
2	24
3	7
4	2

**Tabel Lampiran A 18 Hasil Klastering data  
*birthanddeathrates* Klaster 5**

Klaster	Jumlah masing-masing pembagian data <i>birthanddeathrates</i>
1	15
2	15
3	2
4	6
5	32

**Tabel Lampiran A 19 Hasil Klastering data  
*birthanddeathrates* Klaster 6**

Klaster	Jumlah masing-masing pembagian data <i>birthanddeathrates</i>
1	11
2	16
3	6
4	23
5	2
6	12

**Tabel Lampiran A 20 Hasil Klastering data *iris* Klaster 2**

Klaster	Jumlah masing-masing pembagian data <i>iris</i>
1	54
2	96

**Tabel Lampiran A 21 Hasil Klastering data *iris* Klaster 3**

Klaster	Jumlah masing-masing pembagian data <i>iris</i>
1	54
2	61
3	35

**Tabel Lampiran A 22 Hasil Klastering data *iris* Klaster 4**

Klaster	Jumlah masing-masing pembagian data <i>iris</i>
1	50
2	33
3	46
4	21

**Tabel Lampiran A 23 Hasil Klastering data *iris* Klaster 5**

Klaster	Jumlah masing-masing pembagian data <i>iris</i>
1	10
2	40
3	40
4	48
5	12

**Tabel Lampiran A 24 Hasil Klastering data *iris* Klaster 6**

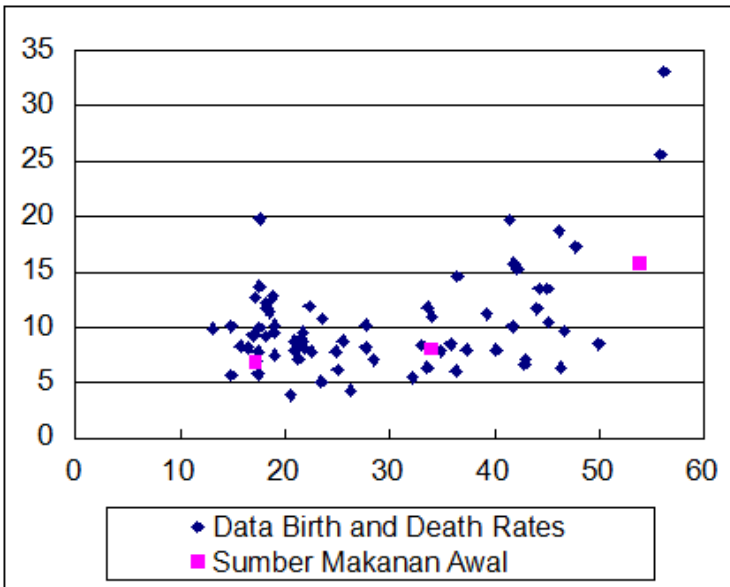
Klaster	Jumlah masing-masing pembagian data <i>birtheaddeathrates</i>
1	23
2	27
3	24
4	24
5	40
6	12



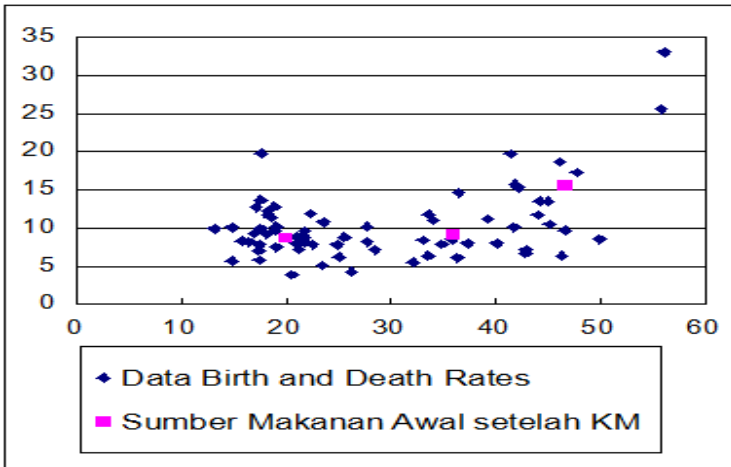
## Lampiran B

### Visualisasi ABCKM untuk Data *Birth and Death Rates*

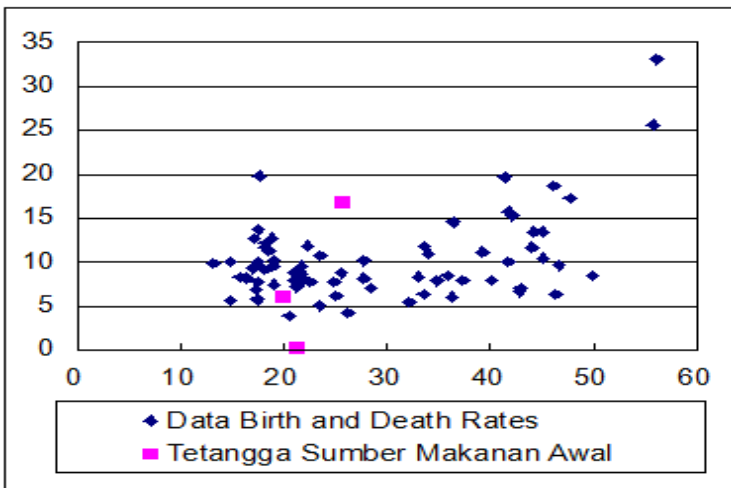
Sumbu X merupakan dimensi 1 yaitu *birth rates*/ angka kelahiran, dan sumbu Y merupakan dimensi 2 yaitu *death rates*/angka kematian.



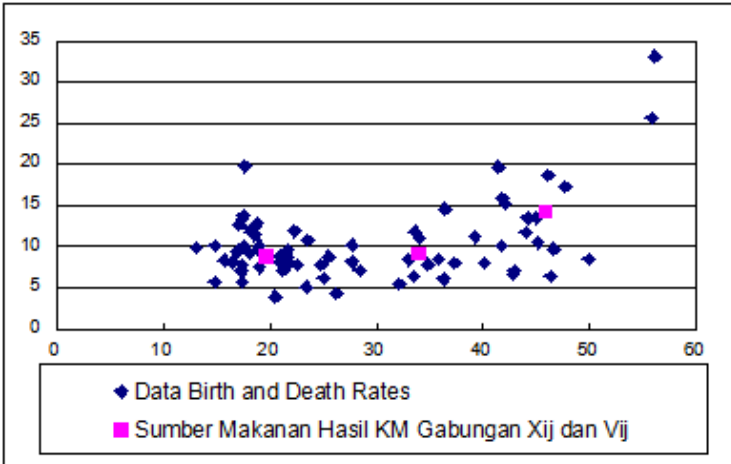
**Gambar B 1** Grafik Data *Birth and Death Rates* terhadap Sumber Makanan Awal Xij



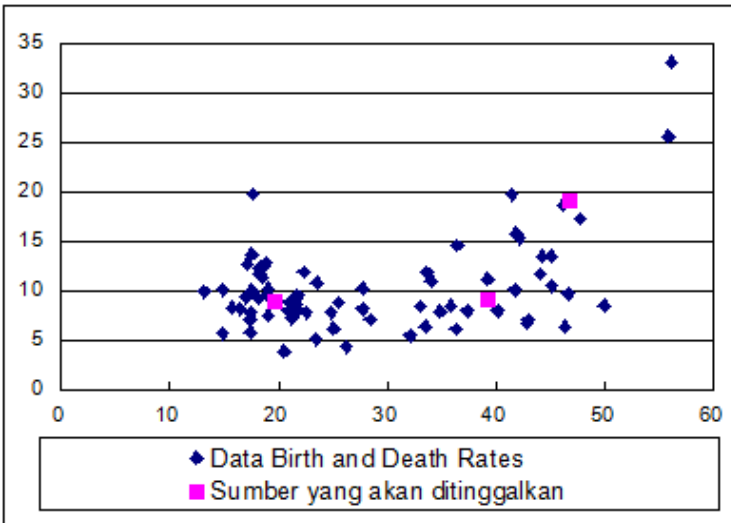
**Gambar B 2 Grafik Data *Birth and Death Rates* terhadap Sumber Makanan Awal Xij setelah di KM**



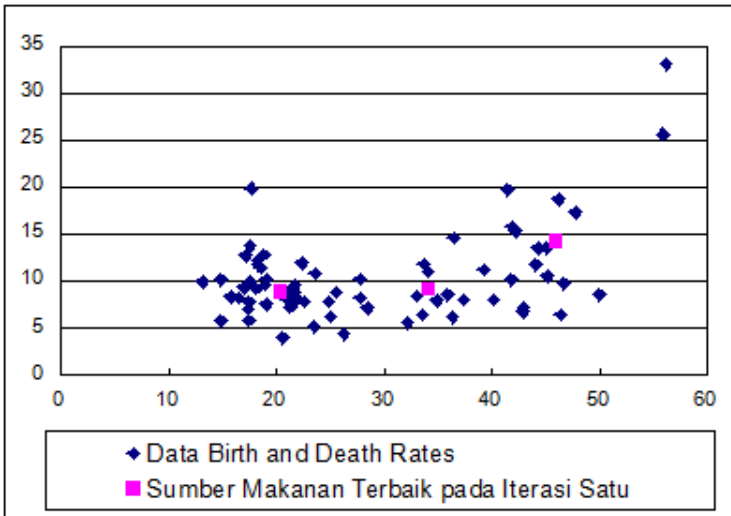
**Gambar B 3 Grafik Data *Birth and Death Rates* terhadap Tetangga Sumber Makanan Awal Vij setelah di KM**



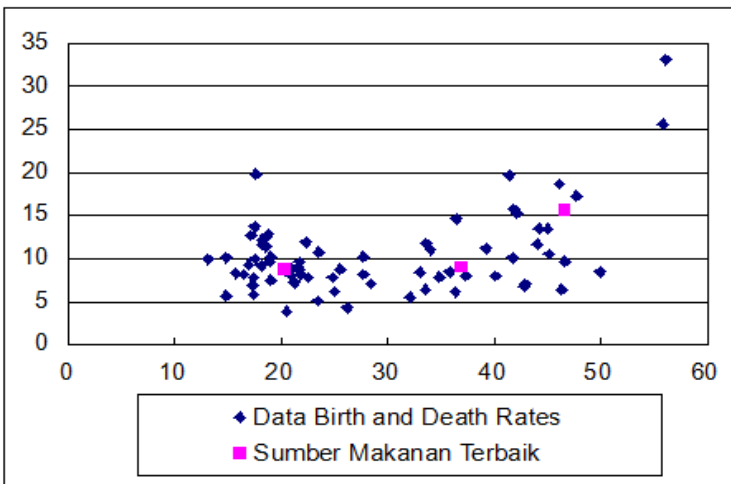
**Gambar B 4** Grafik Data *Birth and Death Rates* terhadap Sumber Makanan Hasil KM Gabungan dari Xij dan Vij



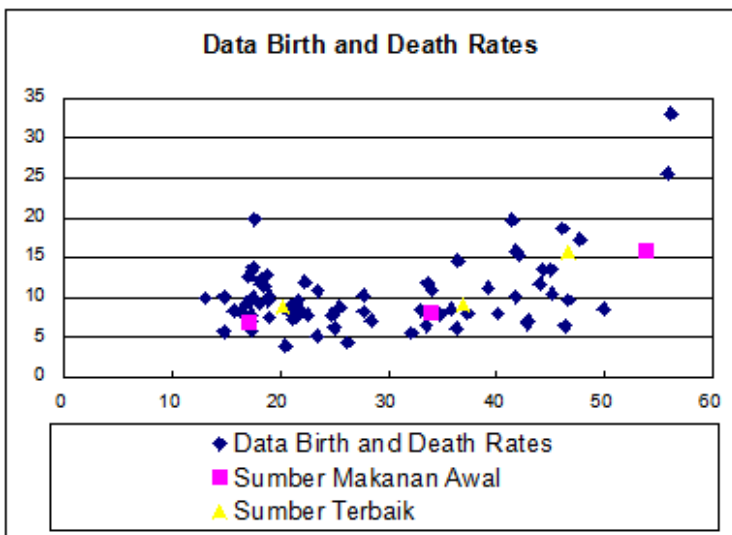
**Gambar B 5** Grafik Data *Birth and Death Rates* terhadap Sumber Makanan yang akan ditinggalkan



**Gambar B 6** Grafik Data *Birth and Death Rates* terhadap Sumber Terbaik pada Iterasi Satu yang akan disimpan



**Gambar B 7** Grafik Data *Birth and Death Rates* terhadap Sumber Makanan Optimal Hasil ABCMKM



**Gambar B 8 Grafik Data *Birth and Death Rates* terhadap Sumber Makanan Awal yang dibandingkan dengan Sumber Optimal Hasil ABCKM**

*(Halaman ini sengaja dikosongkan)*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Pada bab ini berisi kesimpulan mengenai pengujian dan evaluasi yang telah dilakukan. Selain itu, terdapat beberapa saran guna penyempurnaan ke depan.

#### **6.1 Kesimpulan**

Kesimpulan yang diperoleh berdasarkan pengujian dan evaluasi yang telah dilakukan adalah sebagai berikut:

1. Metode ABCKM telah terbukti dapat menyelesaikan permasalahan pembagian data.
2. Berdasarkan hasil koefisien *silhouette* untuk keempat *dataset* pada uji coba, ABCKM memberikan rata-rata nilai koefisien 0.67 yang berarti ABCKM mampu mengklaster data dengan baik, sedangkan KM biasa 0.64.
3. Berdasarkan perhitungan *silhouette* untuk *dataset iris* pada jumlah klaster tiga didapati bahwa ABCKM menghasilkan nilai koefisien *silhouette* 0.73 dan akurasi klasifikasi data yang sangat tinggi yaitu 91.6%.
4. Kinerja algoritma ABCKM sering ditemukan lebih baik dari KM.

#### **6.2 Saran**

Saran yang hendak disampaikan terkait dengan Tugas Akhir ini adalah diperlukannya peninjauan ulang kembali dalam menentukan nilai parameter lain yang akan digunakan, juga dibutuhkan metode pengujian lain untuk permasalahan klustering.

*(Halaman ini sengaja dikosongkan)*



## DAFTAR PUSTAKA

- Pham, D. T., Otri, S., Afifiy, A., Mahmuddin, M., & Al-
- [1] Jabbouli, H. (2007). Data Clustering Using the Bees Algorithm.
  - [2] Kacprzyk, J., & Pedrycz, W. (2015). Springer Handbook of Computational Intelligence.
  - [3] Aggarwal, C. C. (2015). Data Mining: The Textbook. New York.
  - [4] Karaboga, D., & Ozturk, C. (2009). A Novel Clustering Approach: Artificial Bee Colony (ABC) algorithm.

*(Halaman ini sengaja dikosongkan)*

## BIODATA PENULIS



Ario Bagus Nugroho, akrab dipanggil Ario, lahir di Surabaya, 26 Agustus 1992. Penulis merupakan anak kedua dari tiga bersaudara. Penulis menempuh pendidikan formal dimulai dari TK Hang Tuah V (1996-1998), SDN Hang Tuah X, Juanda, Sidoarjo (1998-2004), SMP Al-Falah Sidoarjo (2004-2007), SMA Negeri 15 Surabaya (2007-2010) dan sebagai mahasiswa Teknik Informatika Institut

Teknologi Sepuluh Nopember Surabaya (2010-2016).

Di Teknik Informatika ITS, penulis mengambil bidang minat Komputasi Cerdas dan Visualisasi (KCV), dan aktif berorganisasi di Himpunan Mahasiswa Teknik Computer (HMTC). Selain bidang akademik, penulis suka menghabiskan waktu untuk mendengarkan musik, bermain band, dan bermain game. Penulis dapat dihubungi melalui e-mail: [ariobn56@gmail.com](mailto:ariobn56@gmail.com)