

**IMPLEMENTASI  
ALGORITMA *BUBBLE SORT* DAN *SELECTION SORT*  
MENGUNAKAN *ARRAYLIST* MULTIDIMENSI  
PADA PENGURUTAN DATA MULTI PRIORITAS**

**(Skripsi)**

**Oleh  
Roma Rio Sitepu  
(1317051060)**



**JURUSAN ILMU KOMPUTER  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
2017**

## **ABSTRACT**

### **IMPLEMENTATION OF *BUBBLE SORT* AND *SELECTION SORT* USING MULTIDIMENSIONAL *ARRAYLIST* ON MULTI PRIORITY DATA**

**By**

**ROMA RIO SITEPU**

Ordering is a process of sorting data with a certain rule, so it will be arranged regularly in accordance with the rules. Basically, there are two kinds of commonly used sequencing rules: ascending and descending. In sorting data that consists of many criteria, there will be priority criteria on which the data is sorted. Priority criteria are sorted by initial criteria which are the most preferred criteria until the last criterion which is an additional criterion. In this study, researchers developed an algorithm that can help sort data that has many priorities. The algorithm used is Bubble Sort and Selection Sort, developed using multidimensional ArrayList to sort multi-priority data. The test was conducted on five times experiment and the data sorted by descending. The data used consist of four fictitious data and one real data taken from the list of all criteria of PPA 2017 scholarship recipient of MIPA faculty, University of Lampung. From the results of the tests performed, the best solution is found in the Selection Sort algorithm if compared with the Bubble Sort algorithm for multi priority data sorting.

**Keywords:** *ArrayList, Bubble Sort, Selection Sort, Multi Priority, Multidimensional.*

## **ABSTRAK**

### **IMPLEMENTASI ALGORITMA *BUBBLE SORT* DAN *SELECTION SORT* MENGUNAKAN *ARRAYLIST* MULTIDIMENSI PADA PENGURUTAN DATA MULTI PRIORITAS**

**Oleh**

**ROMA RIO SITEPU**

Pengurutan merupakan suatu proses mengurutkan data dengan suatu aturan tertentu, sehingga tersusun secara teratur sesuai dengan aturan tersebut. Pada dasarnya ada dua macam aturan pengurutan yang biasa digunakan yaitu *ascending* (menaik) dan *descending* (menurun). Pada pengurutan data yang terdiri dari banyak kriteria, terdapat kriteria prioritas yang menjadi dasar data tersebut diurutkan. Kriteria prioritas diurutkan berdasarkan kriteria awal yang merupakan kriteria yang paling diunggulkan sampai kriteria terakhir yang merupakan kriteria tambahan. Dalam penelitian ini, peneliti mengembangkan algoritma yang dapat membantu mengurutkan data yang memiliki banyak prioritas. Algoritma yang digunakan adalah *Bubble Sort* dan *Selection Sort*, yang dikembangkan menggunakan *ArrayList* multidimensi untuk mengurutkan data multi prioritas. Pengujian dilakukan sebanyak lima kali percobaan dan data diurutkan secara *descending*. Data yang digunakan terdiri dari empat data fiktif dan satu data real yang diambil dari daftar seluruh kriteria calon penerima beasiswa PPA 2017 Fakultas MIPA Universitas Lampung. Dari hasil pengujian yang dilakukan, solusi terbaik terdapat pada algoritma *Selection Sort* jika dibandingkan dengan algoritma *Bubble Sort* pada pengurutan data multi prioritas.

**Kata Kunci :** *ArrayList, Bubble Sort, Selection Sort, Multi Prioritas, Multidimensi.*

**IMPLEMENTASI  
ALGORITMA *BUBBLE SORT* DAN *SELECTION SORT*  
MENGUNAKAN *ARRAYLIST* MULTIDIMENSI  
PADA PENGURUTAN DATA MULTI PRIORITAS**

**Oleh:  
ROMA RIO SITEPU**

**Skripsi**

Sebagai Salah Satu Syarat untuk Memperoleh Gelar  
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer  
Fakultas Matematika dan Ilmu Pengetahuan Alam



**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS LAMPUNG  
BANDAR LAMPUNG  
2017**

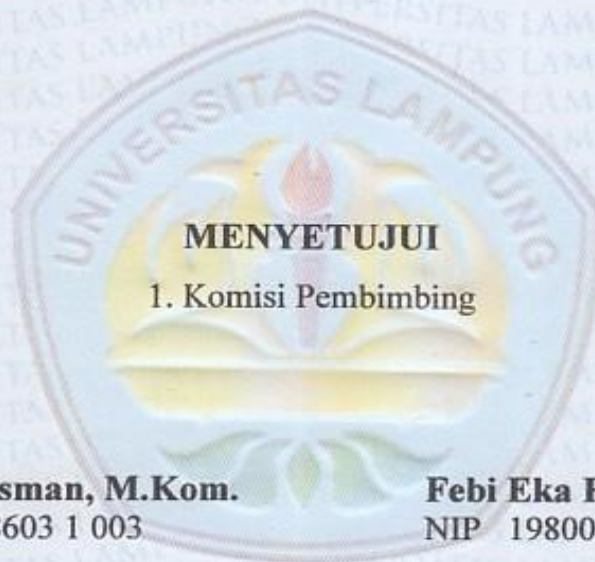
Judul Skripsi : **IMPLEMENTASI ALGORITMA *BUBBLE SORT*  
DAN *SELECTION SORT* MENGGUNAKAN  
*ARRAYLIST* MULTIDIMENSI PADA  
PENGURUTAN DATA MULTI PRIORITAS**

Nama Mahasiswa : **Roma Rio Sitepu**

No. Pokok Mahasiswa : 1317051060

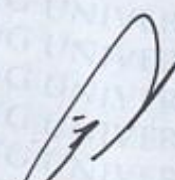
Jurusan : Ilmu Komputer

Fakultas : Matematika dan Ilmu Pengetahuan Alam



**MENYETUJUI**

**1. Komisi Pembimbing**

  
**Ir. Machudor Yusman, M.Kom.**  
NIP 19570330 198603 1 003

  
**Febi Eka Febriansyah, M.T.**  
NIP 19800219 200604 1 001

**2. Mengetahui**  
Ketua Jurusan Ilmu Komputer  
FMIPA Universitas Lampung

  
**Dr. Ir. Kurnia Muludi, M.S.Sc.**  
NIP 19640616.198902 1 001



## MENGESAHKAN

### 1. Tim Penguji

Ketua : **Ir. Machudor Yusman, M.Kom.** .....

Sekretaris : **Febi Eka Febriansyah, M.T.** .....

Penguji  
Bukan Pembimbing : **Dra. Wamiliana, M.A., Ph.D.** .....

### 2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



**Prof. Warsito, S.Si. D.E.A., Ph.D.**  
NIP. 19710212 199512 1 001

Tanggal Lulus Ujian Skripsi : **26 Juli 2017**

## PERNYATAAN

Saya yang bertandatangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul **"Implementasi Algoritma Bubble Sort Dan Selection Sort Menggunakan ArrayList Multidimensi Pada Pengurutan Data Multi Prioritas"** merupakan karya saya sendiri dan bukan hasil karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerima sanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 2 Agustus 2017



ROMA RIO SITEPU  
NPM. 1317051060

## **RIWAYAT HIDUP**



Penulis dilahirkan di Kabanjahe (Sumatera Utara) pada tanggal 11 Maret 1995, sebagai anak keempat dari empat bersaudara dengan Ayah Bagus Sitepu dan Ibu Erlina Br Ginting. Penulis memulai pendidikan di Taman Kanak-Kanak Sint. Xaverius Kabanjahe, pada tahun 2000.

Penulis terdaftar sebagai siswa di SD Sint. Yoseph Kabanjahe pada tahun 2001. Setelah menyelesaikan pendidikan tingkat sekolah dasar, penulis melanjutkan pendidikan di Sekolah Menengah Pertama (SMP) Negeri 1 Kabanjahe. Pada tahun 2010, penulis lulus dari SMP dan melanjutkan pendidikan di Sekolah Menengah Atas (SMA) Swasta Katolik 1 Kabanjahe. Pada tahun 2013, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung. Selama menjadi mahasiswa penulis mengikuti beberapa kegiatan yang dilakukan penulis antara lain:

1. Anggota POM MIPA 2013/2014.
2. Anggota Himakom Bidang Keilmuan 2014/2015.



3. Pada bulan Januari 2014 penulis mengikuti Karya Wisata Ilmiah di Desa Mulyosari, Kecamatan Tanjung Bintang, Kabupaten Lampung Selatan .
4. Anggota Panitia Bidang P3D2T Natal POM MIPA 2014.
5. Anggota Eksekutif Panitia Karya Wisata Ilmiah 2015.
6. Pada bulan Januari 2015 penulis menjadi eksekutif panitia Karya Wisata Ilmiah (KWI) di Pekon Sidokaton, Kecamatan Gisting, Kabupaten Tanggamus.
7. Panitia Propti Jurusan Ilmu Komputer Bidang K-3 2015.
8. Koordinator Bidang P3D2T Paskah POM MIPA 2015.
9. Pada Februari 2016 penulis mengikuti kerja praktek di BBPOM Bandar Lampung.
10. Pada bulan Juli 2016 penulis mengikuti Kuliah Kerja Nyata (KKN) di desa Sri Busono, Kecamatan Way Seputih, Kabupaten Lampung.

## **PERSEMBAHAN**

Puji syukur kepada Tuhan Yang Maha Esa, dengan segala kerendahan  
dan ketulusan hatiku,  
kupersembahkan karya kecilku untuk

Kedua orang tua tercinta, Bapak dan Ibu yang selalu mendoakan dan  
menyebut nama putra putrinya disetiap langkahnya, yang selalu  
melimpahkan kasih sayang yang tidak ternilai berharganya, yang selalu  
mengajari dan membimbing kedua putra-putrinya, serta telah  
mengorbankan seluruh jiwa dan raga demi kebahagiaan putra dan  
putrinya.

Dan untuk,

Kakak-kakakku tercinta Aprilita, Pujiati, Yustina, dan sahabat-sahabatku  
yang selama ini memberi kebahagiaan, motivasi, dukungan, dan  
kebersamaan yang takkan pernah terlupakan.

Dan untuk Almamater tercinta, UNIVERSITAS LAMPUNG.

## **MOTO**

“Takut akan TUHAN adalah permulaan pengetahuan, tetapi orang bodoh  
menghina hikmat dan didikan.”

(Amsal 1:7)

“Stay hungry stay foolish”

(Steve Jobs)

“Perjuanganku lebih mudah karena mengusir  
penjajah, perjuanganmu akan lebih sulit karena  
melawan bangsamu sendiri”

(Ir. Soekarno)

## SANWACANA

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa atas segala rahmat, dan kesehatan yang diberikan kepada penulis sehingga dapat menyelesaikan penulisan skripsi ini.

Skripsi ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer, Fakultas Matematika dan Ilmu Pengetahuan Universitas Lampung. Judul dari skripsi ini adalah “Implementasi Algoritma *Bubble Sort* Dan *Selection Sort* Menggunakan *ArrayList* Multidimensi Pada Pengurutan Data Multi Prioritas”. Dari hasil pengujian yang dilakukan, solusi yang lebih baik terdapat pada algoritma *Selection Sort* jika dibandingkan dengan algoritma *Bubble Sort* pada pengurutan data multi priotitas.

Dalam penyusunan skripsi ini, penulis banyak menghadapi kesulitan. Namun, berkat bantuan dan dorongan dari berbagai pihak, akhirnya penulis dapat menyelesaikan skripsi ini. Untuk itu pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Kedua orang tua, Bapak Bagus Sitepu dan Erlina Br Ginting, kakak-kakakku tersayang Aprilita, Yustina, Pujiati, serta keluarga besar yang selalu memberikan doa, motivasi dan kasih sayang yang tak terhingga.
2. Bapak Ir. Machudor Yusman, M.Kom. sebagai pembimbing I, yang telah membimbing penulis dan memberikan ide, kritik serta saran sehingga penulisan skripsi ini dapat diselesaikan.



3. Bapak Febi Eka Febriansyah, ST, MT sebagai pembimbing II, yang telah memberikan saran, bantuan dan membimbing penulis dalam pembuatan skripsi ini.
4. Ibu Dra. Wamiliana, Ph.D. sebagai pembahas, yang telah memberikan masukan-masukan yang bermanfaat dalam perbaikan skripsi ini.
5. Bapak dan Ibu Dosen serta seluruh *staff* Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang telah memberikan ilmu dan pengalaman dalam hidup untuk menjadi lebih baik.
6. Seluruh mahasiswa angkatan 2013 yang berjuang bersama dalam suka maupun duka, saling membantu, saling bertanya dalam menyelesaikan karya-karyanya selama di Universitas Lampung. Kebersamaan yang telah dilalui menjadi pengalaman berharga bagi penulis.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi sedikit harapan semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan terutama bagi teman-teman Ilmu Komputer.

Bandar Lampung, 2 Agustus 2017

Roma Rio Sitepu

## DAFTAR ISI

	Halaman
<b>DAFTAR ISI</b> .....	xiv
<b>DAFTAR GAMBAR</b> .....	xvii
<b>DAFTAR TABEL</b> .....	xviii
<b>DAFTAR KODE</b> .....	xix
<b>DAFTAR LAMPIRAN</b> .....	xx
<b>I. PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	5
1.3 Batasan Masalah.....	5
1.4 Tujuan.....	5
1.5 Manfaat.....	6
<b>II. TINJAUAN PUSTAKA</b> .....	7
2.1 Data .....	7
2.2 Struktur data dan Algoritma .....	7
2.3 Array.....	8
2.4 <i>ArrayList</i> .....	8
2.5 Pengurutan.....	8
2.6 Pengurutan Data .....	9
2.7 Teknik Pengurutan Data .....	10
2.8 <i>Bubble Sort</i> .....	11

2.9	Konsep <i>Bubble Sort</i> .....	11
2.10	Algoritma <i>Bubble Sort</i> .....	12
2.11	Simulasi <i>Bubble Sort</i> .....	12
2.12	Efisiensi Pada <i>Bubble Sort</i> .....	14
2.13	Algoritma <i>Selection Sort</i> .....	15
2.14	Konsep <i>Selection Sort</i> .....	15
2.15	Algoritma <i>Selection Sort</i> .....	16
2.16	Efisiensi Pada <i>Selection Sort</i> .....	16
2.17	Simulasi <i>Selection Sort</i> .....	17
<b>III.</b>	<b>METODE PENELITIAN</b> .....	19
3.1	Tempat Penelitian .....	19
3.2	Tahapan Penelitian .....	19
3.3	Studi Kepustakaan .....	19
3.4	Analisis Kebutuhan .....	20
3.4.1	Analisis Software .....	21
3.4.2	Analisis Hardware .....	21
3.5	Metode Pengembangan Aplikasi .....	22
3.5.1	Analisis .....	22
3.5.2	Perancangan .....	23
3.5.3	Implementasi .....	30
3.5.4	Pengujian .....	30
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN</b> .....	34
4.1	Data yang Digunakan .....	34
4.2	Penentuan Jumlah Iterasi dan Waktu proses Looping .....	35
4.3	Penjelasan Program .....	36
4.4	Hasil Program .....	42

<b>V.</b>	<b>SIMPULAN DAN SARAN .....</b>	<b>48</b>
5.1	Simpulan.....	48
5.2	Saran .....	49
	<b>DAFTAR PUSTAKA .....</b>	<b>50</b>
	<b>LAMPIRAN.....</b>	<b>51</b>



## DAFTAR GAMBAR

Gambar	Halaman
2.1 Simulasi <i>Bubble Sort</i> .....	14
2.2 Simulasi <i>Selection Sort</i> .....	18
3.1 Tahapan Rencana Penelitian .....	20
3.2 Alur Metode Pengembangan Waterfall.....	22
3.3 <i>Flowchart</i> Algoritma <i>Bubble Sort</i> Menggunakan <i>ArrayList</i> Mutidimesi.....	25
3.4 <i>Flowchart</i> Algoritma <i>Selection Sort</i> Menggunakan <i>ArrayList</i> Mutidimensi .....	26
3.5 <i>Flowchart</i> Tahapan Penulisan Program Pada Bahasa Pemrograman <i>Java</i> .....	27
3.6 Gambar Rancangan Tampilan Awal .....	28
3.7 Gambar Rancangan Tampilan <i>Bubble Sort</i> .....	28
3.8 Gambar Rancangan Tampilan <i>Bubble Sort</i> .....	29
4.1 Tampilan Awal Program .....	42
4.2 Tampilan Formulir <i>Bubble Sort</i> .....	43
4.3 Tampilan Formulir <i>Selection Sort</i> .....	44
4.4 Diagram Perbandingan Jumlah Iterasi Algoritma <i>Bubble Sort</i> dan <i>Selection Sort</i> .....	46
4.5 Diagram Perbandingan Waktu Eksekusi Algoritma <i>Bubble Sort</i> dan <i>Selection Sort</i> .....	47

## DAFTAR TABEL

<b>Tabel</b>	<b>Halaman</b>
<b>3.1</b> Simulasi Algoritma Buble Sort Pada Array Multidimensi .....	32
<b>3.2</b> Simulasi Algoritma Selection Sort Pada Array Multidimensi .....	33
<b>4.1</b> Rincian Jumlah <i>Record</i> Pengujian Data.....	35
<b>4.2</b> Hasil Pengujian Data Dengan Algoritma <i>Bubble Sort</i> .....	45
<b>4.3</b> Hasil Pengujian Data Dengan Algoritma <i>Selection Sort</i> .....	45

## DAFTAR KODE

Kode	Halaman
2.1 Algoritma <i>Bubble Sort</i> .....	12
2.2 Algoritma <i>Selection Sort</i> .....	16
3.1 Algoritma <i>Bubble Sort</i> Menggunakan <i>ArrayList</i> Multidimensi .....	23
3.2 Algoritma <i>Selection Sort</i> Menggunakan <i>ArrayList</i> Multidimensi.....	24
4.1 Kode Program Membaca File CSV dengan bahasa pemrograman <i>Java</i> .....	36
4.2 Kode Program Memisah Data Menjadi Bentuk Baris Dan Kolom.....	37
4.3 Kode Program Mengurutkan Data Dengan Algoritma <i>Bubble Sort</i> .....	38
4.4 Kode Program Mengurutkan Data Dengan Algoritma <i>Selection Sort</i> .....	40
4.5 Kode Membuka File Hasil <i>Running</i> Program.....	42

## DAFTAR LAMPIRAN

Lampiran	Halaman
1 Lampiran Data 1 Dan Hasil.....	51
2 Lampiran Data 2 Dan Hasil.....	52
3 Lampiran Data 3 Dan Hasil.....	54
4 Lampiran Data 4 Dan Hasil.....	55
5 Lampiran Data 5 Dan Hasil.....	57



# **I. PENDAHULUAN**

## **1.1 Latar Belakang**

Pengurutan (*sorting*) merupakan suatu proses mengurutkan data dengan suatu aturan tertentu, sehingga tersusun secara teratur sesuai dengan aturan tersebut. Pada dasarnya ada dua macam aturan pengurutan yang biasa digunakan yaitu *ascending* dan *descending*. *Ascending* adalah proses pengurutan data dari data yang paling kecil sampai data yang paling besar. *Descending* adalah proses mengurutkan data dari yang paling besar sampai data yang paling kecil. Proses yang terjadi dalam pengurutan data adalah proses perbandingan data dan pertukaran data (Utami, dkk,2007).

Data yang telah terurut mempermudah dalam proses pencarian, proses pemeriksaan, dan proses perbaikan data jika terjadi kesalahan. Selain itu, dengan mengurutkan data mempermudah dalam penyisipan data ataupun penggabungan data. Informasi-informasi penting seperti nilai tertinggi dan terendah dalam suatu data sangat mudah didapat setelah dilakukan proses pengurutan.

Proses pengurutan tidak hanya membantu dalam pengolahan data yang berupa angka, data berupa kata-kata juga sering diolah dan diurutkan

berdasarkan urutan huruf alfabet untuk memudahkan pencarian data. Contoh pengurutan data berupa kata-kata sering didapat pada kamus-kamus bahasa. Kata-kata pada kamus diurutkan untuk mempermudah dalam pencarian kata yang dicari maknanya.

Pada dunia pendidikan proses pengurutan sering digunakan dalam membuat daftar absensi siswa. Daftar absensi siswa biasanya diurutkan berdasarkan urutan huruf alphabet dari nama masing-masing siswa. Selain untuk membuat daftar absensi, proses pengurutan juga sering digunakan dalam mengurutkan nilai siswa dari terbesar ke terkecil untuk mendapatkan ranking I, II, dan III. Ranking I, II, dan III didapatkan dengan menjumlahkan seluruh nilai dari kriteria mata pelajaran dan diurutkan secara *descending* berdasarkan kriteria jumlah total dari seluruh kriteria mata pelajaran.

Jika data terdiri dari banyak kriteria, maka harus ada kriteria prioritas yang menjadi dasar data tersebut diurutkan. Contohnya dalam kasus perangkingan, dimana kriteria jumlah total menjadi satu-satunya prioritas yang menjadi dasar pengurutan data. Kriteria jumlah total biasanya didapatkan dengan menjumlahkan nilai dari kriteria-kriteria lainnya. Permasalahan akan muncul ketika pada proses perangkingan, kriteria jumlah total yang merupakan satu-satunya prioritas memiliki nilai yang sama. Hal ini menjadi masalah karena akan sulit melakukan perangkingan secara adil jika hanya memiliki satu prioritas. Mengatasi permasalahan ini dibutuhkan prioritas lain agar data dapat terurut secara adil, dan perlu dibuatkan suatu

algoritma yang dapat mengurutkan data multi kriteria yang memiliki lebih dari satu prioritas.

Saat ini ada begitu banyak algoritma yang ditemukan dalam kasus pengurutan data, seperti *Bubble Sort*, *Insertion Sort*, *Merge Sort*, *Quick Sort*, *Selection Sort*, dan *Shell Sort*. Tetapi, dalam penelitian ini akan menjelaskan bagaimana pengurutan data dengan menggunakan metode *Bubble Sort* dan *Selection Sort*, kemudian diterapkan pada *ArrayList* multidimensi untuk membantu pengurutan data banyak kriteria dan memiliki lebih dari satu kriteria prioritas. *ArrayList* merupakan sebuah kelas yang dapat menyimpan data berupa list objek berbentuk array yang ukurannya dapat berubah secara dinamis sesuai dengan jumlah data yang dimasukkan. *ArrayList* sangat cocok diimplementasikan dalam pengurutan data multi kriteria yang ukuran datanya bersifat dinamis.

*Bubble Sort* adalah metode pengurutan dengan cara melakukan penukaran data dengan data yang tepat berada disebelahnya secara terus menerus sampai bisa dipastikan dalam satu iterasi tertentu tidak ada lagi perubahan. Metode *Bubble Sort* yang ditemukan pada tahun 1965 (Kochar dan Agrawal, 2014) diinspirasi oleh gelembung sabun yang berada dipermukaan air. Berat jenis gelembung sabun lebih ringan daripada berat jenis air, maka gelembung sabun selalu terapung ke atas permukaan. Prinsip tersebut sebagai dasar yang dipakai dalam pengembangan algoritma *Bubble Sort*.

Algoritma *Bubble Sort* merupakan algoritma sorting paling sederhana dan mudah dipahami. *Bubble Sort* sangat cocok digunakan untuk mengurutkan data dengan elemen kecil, sedangkan untuk data dengan elemen besar algoritma ini memiliki kinerja yang lambat jika dibandingkan dengan algoritma sorting lain seperti *Insertion Sort* ( Fauzi dan Indrayana, 2005).

*Selection Sort* merupakan salah satu algoritma pengurutan yang sederhana. Untuk *sorting ascending* (menaik), elemen yang paling kecil di antara elemen-elemen yang belum urut, disimpan indeksnya, kemudian dilakukan pertukaran nilai elemen dari indeks yang disimpan tersebut dengan elemen yang paling depan yang belum urut. Sebaliknya, untuk *sorting descending* (menurun), elemen yang paling besar disimpan indeksnya kemudian ditukar dengan elemen yang paling depan yang belum urut.

Selain untuk mengurutkan data multi kriteria berdasarkan prioritas, implementasi dari algoritma *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi juga akan membandingkan mana metode yang lebih disarankan untuk digunakan dalam pengurutan.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas, maka rumusan masalah dalam penelitian ini adalah bagaimana mengimplementasikan algoritma *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi pada pengurutan data multi kriteria dan memiliki lebih dari satu kriteria prioritas.

## 1.3 Batasan Masalah

Batasan masalah dalam penelitian ini antara lain sebagai berikut

1. Algoritma yang digunakan adalah algoritma *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi khususnya pada *ArrayList* dua dimensi.
2. Data yang diurutkan memiliki tipe data yang sama yaitu Double .
3. Bahasa pemrograman yang digunakan adalah bahasa pemrograman *Java*.

## 1.4 Tujuan

Tujuan dari dilakukannya penelitian ini adalah:

1. Mengimplementasikan algoritma *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi.
2. Mengurutkan data multi kriteria yang memiliki lebih dari satu kriteria prioritas.
3. Memberikan contoh implementasi algoritma *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi pada data multi kriteria yang memiliki lebih dari satu kriteria prioritas.

4. Membandingkan waktu eksekusi dan jumlah perulangan algoritma *Bubble Sort* dan *Selection Sort* pada pengurutan data multi prioritas.

### **1.5 Manfaat**

Manfaat dari dilakukannya penelitian ini adalah:

1. Membantu dalam pengurutan data multi kriteria yang memiliki lebih dari satu prioritas.
2. Memudahkan dalam hal pengambilan kesimpulan dari data yang memiliki banyak kriteria.
3. Membantu memudahkan penentuan nilai terendah dan tertinggi dari suatu data yang diurutkan berdasarkan kriteria prioritas.

## **II. TINJAUAN PUSTAKA**

### **2.1 Data**

Data adalah kata jamak (*plural*) dan kata tunggalnya (*singular*) adalah datum. Kata tersebut berasal dari bahasa Latin yang berarti fakta, kenyataan, kejadian, atau peristiwa. Jadi data adalah suatu kebenaran, karena itu ciri atau karakteristik utama data adalah benar. Pada pekerjaan manajemen sistem informasi data adalah bahan utama. Tanpa data pekerjaan informasi tidak akan pernah ada (Amsyah, 2005).

### **2.2 Struktur data dan Algoritma**

Istilah struktur data dan algoritma biasa hadir tidak terpisahkan. Jika kita membicarakan struktur data maka tidak bisa terlepas dari algoritma. Struktur data adalah pengaturan data di dalam memori komputer atau terkadang di dalam disk dengan tujuan agar data dapat diakses secara efisien. Yang termasuk dalam struktur data antara lain adalah antrian, tumpukan, pohon biner. Algoritma adalah suatu langkah atau prosedur yang ditunjukkan untuk memanipulasi data. Contoh algoritma diperlukan untuk mengurutkan data untuk melihat data terurut dari nilai terbesar ke nilai terendah ataupun sebaliknya (Kadir, 2011).

### 2.3 Array

Array atau sering juga disebut larik adalah jenis variabel yang mampu menyimpan beberapa nilai data bertipe sama dalam urutan tertentu. Isi atau nilai data yang berada dalam array tersebut disebut dengan elemen array. Letak urutan masing-masing elemen array ditandai dengan indeks elemen array. Indeks array pada umumnya secara otomatis selalu dimulai dari 0 (Kadir, 2011).

### 2.4 *ArrayList*

Objek *arraylist* adalah perkembangan dari larik (*array*). Objek *ArrayList* mendukung untuk melakukan akses elemen secara acak, dimana setiap elemen dapat diakses dalam waktu yang konstan. Tidak seperti larik (*array*), objek *ArrayList* ukurannya dapat dikelola secara otomatis saat program sedang berjalan (Widiantoro, 2010).

### 2.5 Pengurutan

Pengurutan adalah proses merangkai deretan data menjadi teratur secara naik atau menurun. Pengurutan tidak hanya dilakukan pada data numerik, namun dapat pula dilakukan pada data non numerik. Pengurutan data secara non numeric contohnya adalah mengurutkan suatu rangkaian data secara alfabel. Pengurutan seringkali dilakukan agar perintah pencarian dan perintah lainnya dapat dilakukan dengan lebih mudah. Buku-buku di perpustakaan diurutkan menurut subjek/nama agar pencarian terhadap suatu buku tertentu lebih mudah dilakukan (Kadir, 2011).



## 2.6 Pengurutan Data

Pengurutan data merupakan salah satu proses yang banyak dijumpai dalam aplikasi menggunakan komputer. Pengurutan membuat data disusun dalam keadaan urut menurut aturan tertentu, misalnya dari nilai terkecil menuju ke nilai terbesar untuk bilangan atau A ke Z untuk string. Sebagai contoh, data yang telah diurutkan dalam suatu laporan memudahkan untuk dibaca dan dipahami oleh pemakai. Selain itu, data yang terurut akan memudahkan pencarian. Dalam prakteknya banyak kita jumpai hal-hal yang menggunakan pengurutan data,

- a. *Yellow Pages* berisi informasi telepon yang telah diurutkan menurut nama perusahaan atau perorangan.
- b. Kamus berisi kata-kata yang telah diurutkan tanpa membedakan huruf kecil dan huruf besar.
- c. Laporan penjualan disusun berdasarkan produk yang terlaris hingga yang paling sedikit terjual.
- d. *File-file* dalam direktori ditampilkan urut berdasarkan atribut tertentu (bias nama, tanggal dibuat, dan lain-lain).
- e. Indeks buku berisi daftar istilah yang memudahkan pembaca untuk mendapatkan lokasi halaman yang berisi istilah tersebut.
- f. *Glossary* dalam buku teks berisi istilah dan definisinya, dengan nama-nama istilah disusun urut.

Pengurutan bias dibedakan menjadi dua jenis yaitu pengurutan naik (*ascending*) dan pengurutan turun (*descending*) (Kristanto, 2009).

## 2.7 Teknik Pengurutan Data

Ada beberapa cara yang digunakan untuk mengurutkan data. Teknik terkenal untuk mengurutkan data adalah *Bubble Sort*, *Insertion Sort*, *Selection Sort*, *Quick Sort*, *Merge Sort*, *Heap Sort*, dan *Binary Sort*. Tiga yang pertama yaitu *Bubble Sort*, *Insertion sort*, *Selection Sort* dikenal sebagai cara pengurutan dasar.

Secara umum masing-masing teknik pengurutan data memiliki keunggulan dan kelemahan tersendiri. Sangat sulit untuk mengatakan teknik mana yang paling baik. Namun tentu ada beberapa pertimbangan yang bisa dipakai untuk memilih teknik yang akan dipakai. Jumlah data, kompleksitas waktu, dan lama waktu untuk membuat program terkadang menjadi acuan untuk memilih teknik pengurutan data. Sebagai contoh, untuk jumlah data yang sedikit (misalnya ratusan), *Bubble Sort* bisa digunakan.

Teknik pengurutan *Bubble Sort* merupakan teknik pengurutan yang paling mudah jika dibandingkan dengan teknik pengurutan lainnya. Teknik ini sering digunakan untuk mengenalkan dasar dari suatu pengurutan data karena mudah untuk dimengerti dan mudah untuk dijelaskan. Namun untuk data yang jutaan jumlahnya, penggunaan teknik tersebut tidak tepat karena memakan waktu lama untuk mengurutkan data (Kadir, 2011).

## **2.8 *Bubble Sort***

Pengurutan yang menggunakan algoritma *Bubble Sort* dilakukan dengan cara membandingkan elemen yang diseleksi dengan elemen yang berikutnya. Jika elemen yang diseleksi lebih besar dari elemen berikutnya maka elemen tersebut ditukar. Setiap langkah dari algoritma ini seolah-olah menggeser satu per satu elemen dari kanan ke kiri. Jika barisan bilangan tidak disusun horizontal melainkan vertikal, maka terlihat seperti gelembung-gelembung (*bubble*) yang naik dari dasar akuarium. Oleh karena itu algoritma ini disebut *Bubble Sort* (Kristanto, 2009).

## **2.9 Konsep *Bubble Sort***

*Bubble Sort* merupakan cara pengurutan yang sederhana. Konsep dari ide dasarnya adalah seperti “gelembung air” untuk elemen struktur data yang semestinya berada pada posisi awal. Cara kerja dari algoritma ini adalah mengulang proses perbandingan antara tiap-tiap elemen array dan menukarnya apabila urutannya salah. Perbandingan elemen-elemen ini akan terus diulang hingga tidak perlu dilakukan penukaran lagi. (Yahya, 2014).

## 2.10 Algoritma *Bubble Sort*

Algoritma *Bubble Sort* ditunjukkan pada Kode 2.1.

```

BubbleSort(data,n):
//data adalah array dengan n data
  FOR i ← 0 TO n-2
    FOR j ← 0 TO n-(i+2)
      IF data[j]<data[j+1]
        temp←data[j]
        data[j]←data[j+1]
        data[j+1]←temp
      END IF
    END FOR
  END-FOR

```

**Kode 2.1** Algoritma *Bubble Sort*

Kunci pengurutan secara *ascending* terletak pada perintah IF  $\text{data}[j] > \text{data}[j+1]$ , sementara untuk pengurutan secara *descending*, perintahnya menjadi IF  $\text{data}[j] < \text{data}[j+1]$  (Kadir, 2011).

## 2.11 Simulasi *Bubble Sort*

Berikut ini adalah contoh simulasi algoritma *sorting* terhadap larik (*array*) dengan panjang 5 elemen. Misalnya untuk larik dengan elemen

8	4	5	2	7
---	---	---	---	---

akan diurutkan dengan algoritma *Bubble Sort* secara terurut dari terbesar ke terkecil (*descending*).

Proses perbandingan di dalam setiap *looping* digambarkan dengan warna coklat. Jika ternyata diperlukan pertukaran nilai, maka akan ditulis “tukar” di bawah elemen larik yang dibandingkan tersebut. Hasil penukaran digambarkan di tabel simulasi berikutnya.

Untuk membedakan dengan elemen tabel yang lain, bagian tabel yang telah diurutkan digambarkan dengan warna biru muda. Simulasi *Bubble Sort* ditunjukkan pada Gambar 2.1.

*Looping i* pertama:

8	4	5	2	7
---	---	---	---	---

8	4	5	2	7
---	---	---	---	---

**tukar**

8	5	4	2	7
---	---	---	---	---

8	5	4	2	7
---	---	---	---	---

**tukar**

8	5	4	7	2
---	---	---	---	---

*Looping i* kedua:

8	5	4	7	2
---	---	---	---	---

**tukar**

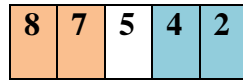
8	5	7	4	2
---	---	---	---	---

**tukar**

8	7	5	4	2
---	---	---	---	---

8	7	5	4	2
---	---	---	---	---

*Looping i ketiga*



*Looping i keempat*



**Gambar 2.1** Simulasi *Bubble Sort*

(Kadir,2011)

## 2.12 Efisiensi Pada *Bubble Sort*

Berdasarkan algoritma *Bubble Sort*, jumlah perbandingan selama pengurutan data sebanyak:

$$(n-1) + (n-2) + (n-3) + \dots + 1 = n(n-1)/2$$

atau

$$\sum_{i=1}^{n-1} i = \frac{(n-1)n}{2}$$

Efisiensi ini berlaku untuk *best case*, *worst case* maupun *average case*.

Untuk  $n = 5$  maka jumlah perbandingan data sebanyak  $((5 \times 4)/2) = 10$  kali.

Dengan demikian jika dinyatakan dengan Big O maka kompleksitas *Bubble Sort* berupa  $O(n(n-1)/2)$  atau sama dengan  $O(n^2)$ . *Best case* terjadi kalau data yang diurutkan sudah dalam keadaan urut (Kadir, 2011).

### 2.13 Algoritma Selection Sort

*Selection Sort* adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya sampai ke elemen yang terakhir. Jika ditemukan elemen lain yang lebih kecil dari elemen sekarang maka dicatat posisinya dan langsung ditukar. Konsep proses *Selection Sort* adalah mencari (memilih) nilai terkecil dan menukarnya dengan elemen paling awal (paling kiri) pada setiap tahap. Proses Sort dilakukan tahap per tahap (Yahya, 2014).

### 2.14 Konsep Selection Sort

Konsep *Selection Sort* secara *descending* adalah melakukan pemilihan dari suatu nilai yang terbesar dan kemudian menukarnya dengan elemen paling awal, lalu membandingkan dengan elemen yang sekarang dengan elemen berikutnya sampai dengan elemen terakhir, perbandingan dilakukan terus sampai tidak ada lagi pertukaran data (Agustia, 2016).

### 2.15 Algoritma *Selection Sort*

Algoritma *Selection Sort* ditunjukkan pada Kode 2.2.

```

Selection Sort(data,n):
//data adalah array dengan n data
  FOR i ← 0 TO n-2
    awal=i
    FOR j ← i+1 TO n-(i+2)
      IF (data [awal] > data[j])
        awal = j;
      END IF
    END FOR
    temp←data[i]
    data[i]←data[awal]
    data[awal]←temp
  END FOR

```

**Kode 2.2** Algoritma *Selection Sort*

(Kadir, 2011).

### 2.16 Efisiensi Pada *Selection Sort*

Kompleksitas waktu pada *Selection Sort* diukur dengan jumlah perbandingan. Pada tahap pertama terdapat  $N-1$  perbandingan. Pada tahap kedua terdapat  $N-2$  perbandingan. Pada tahap ketiga terdapat  $N-3$  perbandingan, dan seterusnya. Dengan demikian, total perbandingan sebesar:

$$(N-1)+(N-2)+\dots+2+1 = N(N-1)/2$$

Oleh karena itu, kompleksitas waktu berupa  $O(N^2)$ .

*Worst case* diperoleh ketika data dalam keadaan urut terbalik. Perbandingan yang terjadi  $N(N-1)/2$ . Dengan demikian, *worst case* berupa  $O(N^2)$  (Kadir, 2011).



### 2.17 Simulasi *Selection Sort*

Berikut ini adalah contoh simulasi algoritma *sorting* terhadap larik (*array*) dengan panjang 5 elemen. Misalnya untuk larik dengan elemen

8	4	5	2	7
---	---	---	---	---

akan diurutkan dengan algoritma *Selection Sort* secara terurut dari terbesar ke terkecil (*descending*).

Proses perbandingan di dalam setiap *looping* digambarkan dengan warna coklat. Jika ternyata diperlukan pertukaran nilai, maka akan ditulis “tukar” di bawah elemen larik yang dibandingkan tersebut. Hasil penukaran digambarkan di tabel simulasi berikutnya.

Untuk membedakan dengan elemen tabel yang lain, bagian tabel yang telah diurutkan digambarkan dengan warna biru muda. Simulasi *Selection Sort* ditunjukkan pada Gambar 2.2.

*Looping i* pertama:

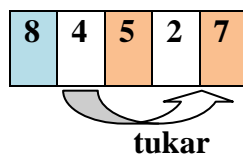
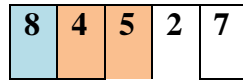
8	4	5	2	7
---	---	---	---	---

8	4	5	2	7
---	---	---	---	---

8	4	5	2	7
---	---	---	---	---

8	4	5	2	7
---	---	---	---	---

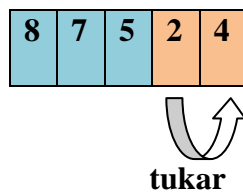
*Looping i* kedua:



*Looping i* ketiga:



*Looping i* keempat:



**Gambar 2.2** Simulasi *Selection Sort*

(Kadir, 2011)

### **III. METODE PENELITIAN**

#### **3.1 Tempat Penelitian**

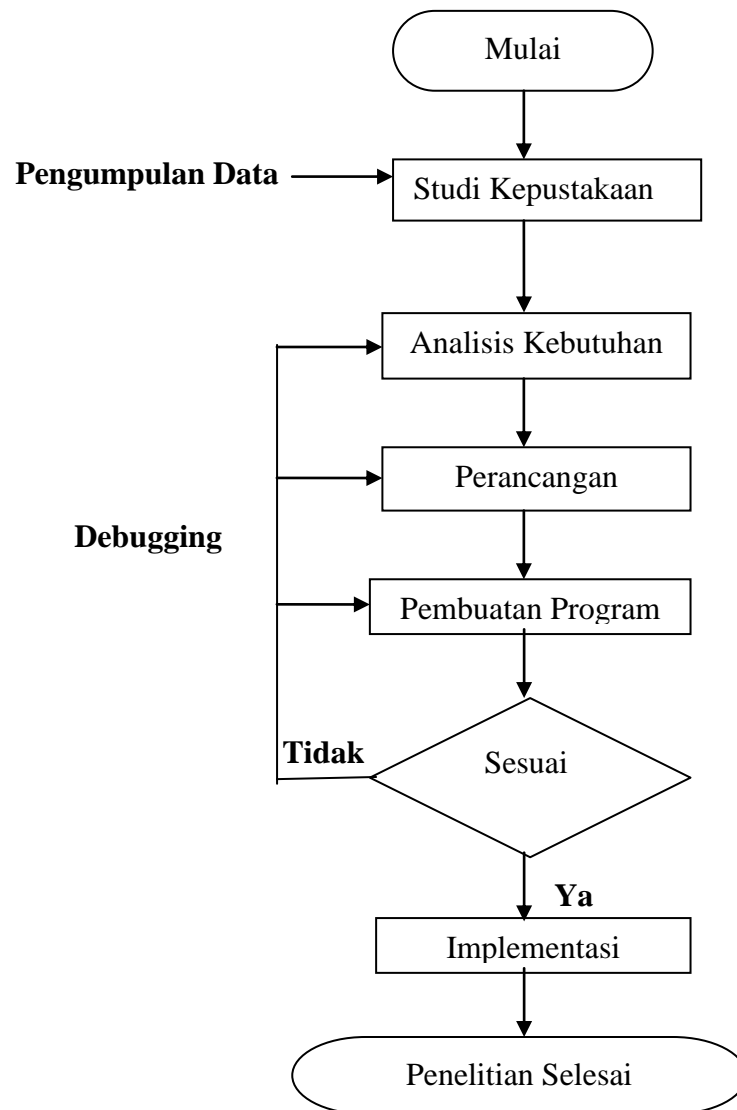
Penelitian ini dilakukan di Laboratorium Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung.

#### **3.2 Tahapan Penelitian**

Penelitian dilakukan melalui tahapan-tahapan kegiatan dengan mengikuti kerangka pikir yang meliputi metode pengumpulan data dan metode pengembangan aplikasi ditunjukkan pada Gambar 3.1.

#### **3.3 Studi Kepustakaan**

Studi kepustakaan merupakan suatu teknik yang digunakan untuk mengumpulkan materi tentang metode pengurutan *Bubble Sort dan Selection Sort* dengan menggunakan *ArrayList* multidimensi. Materi didapat melalui referensi-referensi atau literatur ilmiah berupa jurnal-jurnal penelitian, buku, karya tulis ataupun hasil pencarian melalui internet. Data yang diperoleh dijadikan sebagai basis pengetahuan dalam implementasi algoritma *sorting bubble sort* pada array multidimensi.



**Gambar 3.1** Tahapan Rencana Penelitian

### 3.4 Analisis Kebutuhan

Sebelum suatu aplikasi dibuat perlu adanya suatu rumusan dan perencanaan yang jelas, sehingga dapat ditentukan sasaran dari aplikasi yang dibuat. Untuk mendukung pembuatan aplikasi tersebut, perlu adanya dukungan aplikasi komputer memadai, baik dari segi perangkat lunak (*software*) maupun perangkat keras (*hardware*).

### 3.4.1 Analisis Software

Kebutuhan perangkat lunak (software) untuk implementasi algoritma

*Bubble Sort* dan *Selection Sort* pada *ArrayList* multidimensi yaitu:

- a. Aplikasi operasi (Windows 7 / Windows 8/ Windows10)
- b. Click Chart Diagram Flowchart
- c. Netbeans

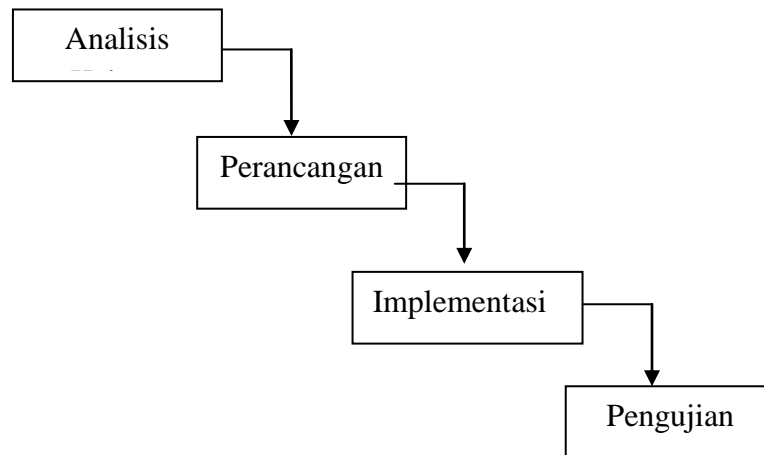
### 3.4.2 Analisis Hardware

Spesifikasi minimal untuk perangkat keras (hardware) adalah sebagai berikut:

- a. Processor Intel Core i3
- b. RAM (Random Access Memory) 2 Giga Byte
- c. Hardisk 320 Giga Byte
- d. Monitor
- e. Keyboard
- f. Mouse

### 3.5 Metode Pengembangan Aplikasi

Metode yang digunakan dalam pembuatan aplikasi adalah metode *Waterfall*. Inti dari metode *Waterfall* adalah pengerjaan dari suatu aplikasi dilakukan secara berurutan atau secara linear (Jogiyanto, 2005). Metode pengembangan aplikasi dapat dilihat pada Gambar 3.2.



**Gambar 3.2** Alur Metode Pengembangan Waterfall

#### 3.5.1 Analisis

Pengumpulan data dalam tahap ini berupa kegiatan penelitian di laboratorium Ilmu Komputer Unila. Dari data yang diperoleh dilakukan analisis terhadap kebutuhan aplikasi, dan kemudian selanjutnya dijadikan acuan untuk menerjemahkan ke dalam bahasa pemrograman.

### 3.5.2 Perancangan

Tahap ini terdiri dari perancangan aplikasi dan pembuatan program. Perancangan aplikasi merupakan perencanaan untuk mencari solusi permasalahan yang diperoleh dari tahap analisis. Pembuatan program merupakan proses penerjemahan desain dalam bahasa yang dikenali oleh komputer atau proses memasukkan kode pada program.

Tahap pertama yang dilakukan dalam perancangan aplikasi adalah membuat algoritma dari teknik *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi untuk melakukan pengurutan berdasarkan prioritas. Algoritma *Bubble Sort* menggunakan *ArrayList* multidimensi ditunjukkan pada Kode 3.1.

```
BubbleSort(data,n):
  FOR i ← 0 TO data.size()-1
    FOR j ← 0 TO data.size()-(i+1)
      FOR k ← 0 TO data.get(j).size()-1
        hasil=data.get(j).get(k)- data.get(j+1).get(k)
        IF hasil !=0
          IF hasil<0
            // Lakukan Penukaran
            Swap(data,j,j+1)
          END IF
          BREAK
        END-IF
      END FOR
    END-FOR
  END-FOR
```

**Kode 3.1** Algoritma *Buble Sort* Menggunakan *ArrayList* Multidimensi

Algoritma *Selection Sort* menggunakan *ArrayList* multidimensi ditunjukkan pada Kode 3.2.

```

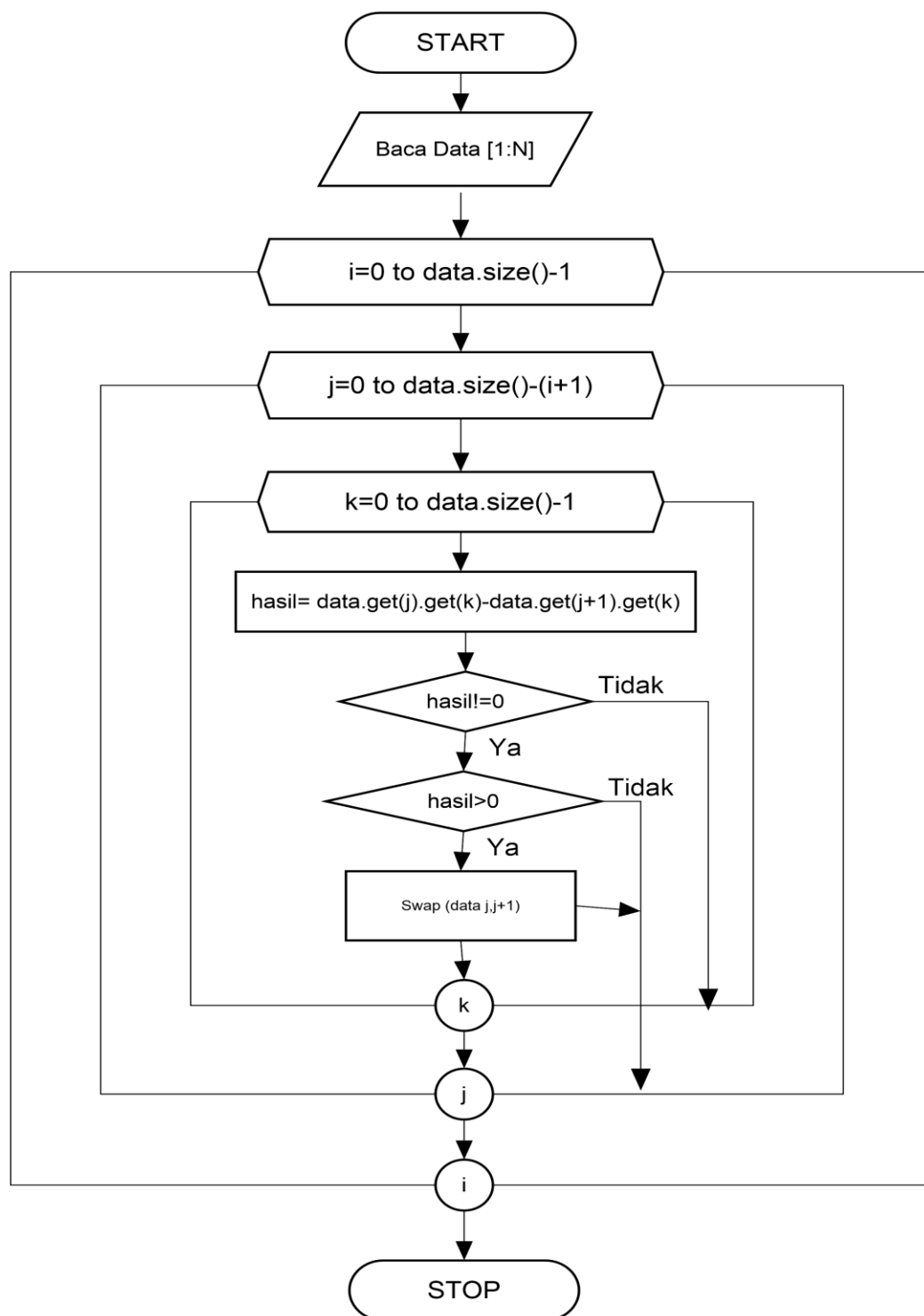
SelectionSort(data,data.size()):
  FOR i ← 0 TO data.size()-1
    awal=i
    FOR j ← i+1 TO data.size()-1
      FOR k← 1 TO data.get(j).size -1
        hasil=data.get(awal).get(k)- data.get(j).get(k)
        IF hasil !=0
          IF hasil<0
            // Lakukan Penukaran
            awal=j
          END IF
          BREAK
        END-IF
      END FOR
    END-FOR
    Swap(data,i,awal);
  END-FOR

```

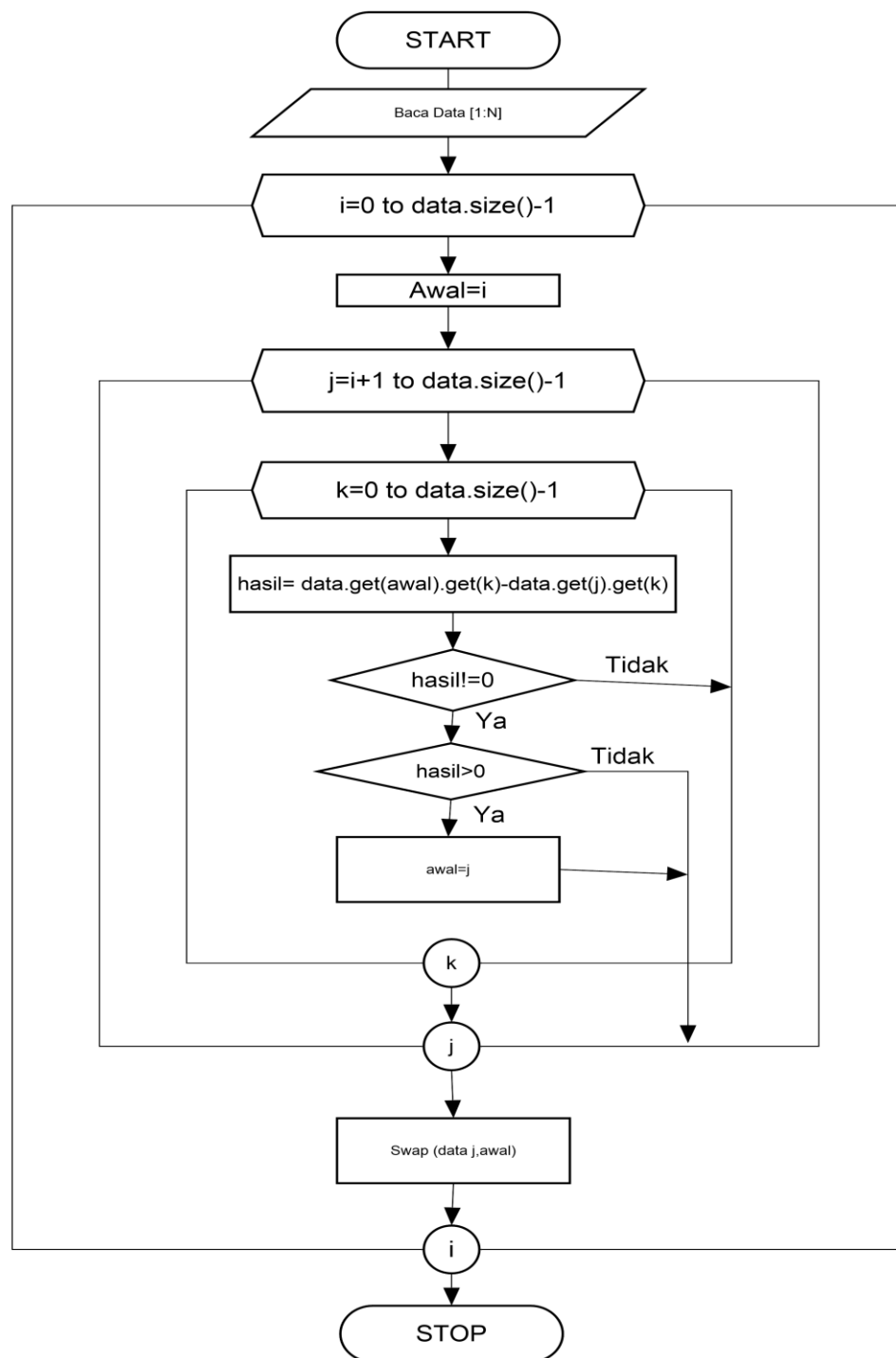
**Kode 3.2** Algoritma *Selection Sort* Menggunakan *ArrayList* Multidimensi

Setelah membuat algoritma, langkah kedua yang dilakukan adalah membuat *flowchart* algoritma *Bubble Sort* pada array multidimensi. *Flowchart* digunakan sebagai acuan dalam menerapkan algoritma *sorting Bubble Sort* pada array multidimensi pada bahasa pemrograman *Java*. *Flowchart* algoritma *Bubble Sort* menggunakan *ArrayList* mutidimensi ditunjukkan pada Gambar 3.3. *Flowchart* algoritma *Selection Sort* menggunakan *ArrayList* mutidimensi ditunjukkan pada Gambar 3.4.



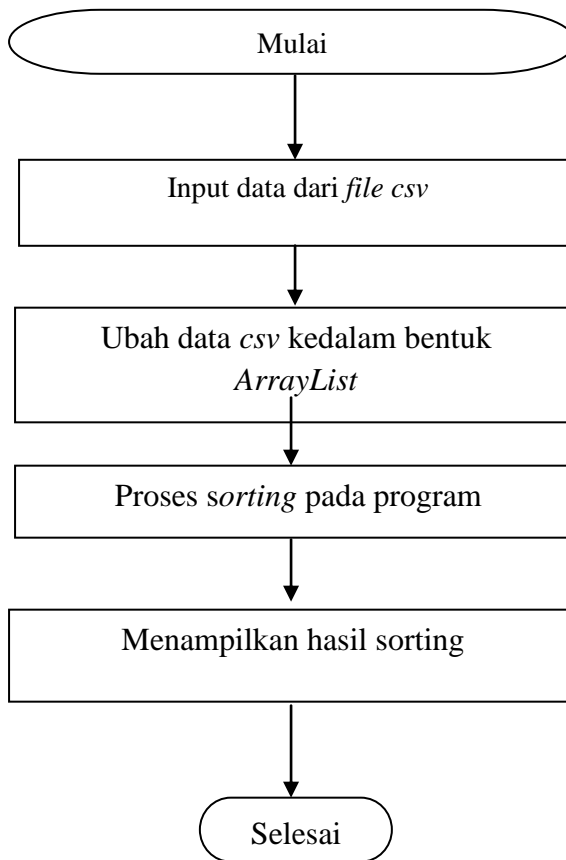


**Gambar 3.3** Flowchart Algoritma Bubble Sort Menggunakan ArrayList Mutidimensi



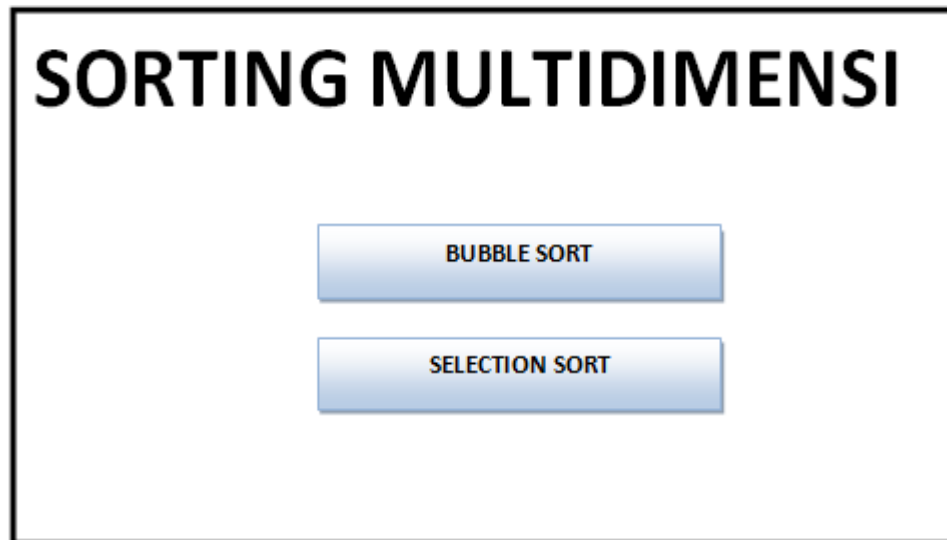
**Gambar 3.4** Flowchart Algoritma Selection Sort Menggunakan ArrayList Mutidimensi

*Flowchart* tahapan penulisan program pada bahasa pemrograman *Java* ditunjukkan pada Gambar 3.5.



**Gambar 3.5** *Flowchart* Tahapan Penulisan Program Pada Bahasa Pemrograman *Java*

Setelah *flowchart* selesai digambarkan, langkah selanjutnya adalah membuat rancangan desain *interface* Aplikasi. Rancangan tampilan awal ditunjukkan pada Gambar 3.6.



**Gambar 3.6** Gambar Rancangan Tampilan Awal

Pada gambar 3.6 terdapat dua tombol yaitu tombol "*Bubble Sort*" dan tombol "*Selection Sort*". Tombol "*Bubble Sort*" berfungsi untuk melakukan proses menuju formulir *Bubble Sort*. Tombol "*Selection Sort*" berfungsi untuk melakukan proses menuju formulir *Selection Sort*. Gambar rancangan tampilan *Bubble Sort* ditunjukkan pada Gambar 3.7.



**Gambar 3.7** Gambar Rancangan Tampilan *Bubble Sort*

Pada gambar 3.7 terdapat tiga tombol yaitu tombol “*Browse*”, tombol “*Bubble Sort*” dan tombol “*OK*”. Terdapat juga 3 *textfield* yaitu *textfield* “nama file”, *textfield* “jumlah” dan *textfield* “waktu”. Tombol “*Browse*” berfungsi untuk membuka *file* yang berisi data yang ingin diurutkan. Tombol “*Bubble Sort*” berfungsi untuk melakukan proses *Bubble Sort* serta menampilkan hasil *sorting*. Tombol “*OK*” berfungsi untuk melakukan perhitungan jumlah iterasi dan waktu eksekusi kemudian menampilkannya di *Textfield* “jumlah” dan *Textfield* “waktu”. Rancangan tampilan *Selection Sort* ditunjukkan pada Gambar 3.8.

**Gambar 3.8** Gambar Rancangan Tampilan *Selection Sort*

Pada gambar 3.8 terdapat tiga tombol yaitu tombol “*Browse*”, tombol “*Bubble Sort*” dan tombol “*OK*”. Terdapat juga 3 *textfield* yaitu *textfield* “nama file”, *textfield* “jumlah” dan *textfield* “waktu”. Tombol “*Browse*” berfungsi untuk membuka *file* yang berisi data yang ingin diurutkan. Tombol “*Selection Sort*” berfungsi untuk melakukan proses *Selection Sort* serta

menampilkan hasil *sorting*. Tombol “OK” berfungsi untuk melakukan perhitungan jumlah iterasi dan waktu eksekusi kemudian menampilkannya di *Textfield* “jumlah” dan *Textfield* “waktu”.

### 3.5.3 Implementasi

Pada tahap ini mengimplementasikan perancangan aplikasi ke dalam bahasa pemrograman *Java*.

### 3.5.4 Pengujian

Tujuan pengujian adalah menemukan kesalahan-kesalahan pada saat mengaplikasikan algoritma dan kemudian dapat diperbaiki. Jika kesalahan ditemukan maka proses analisis, perancangan, dan pembuatan program diperbaiki kembali. Data yang digunakan pada penelitian ini adalah data fiktif dan data daftar seluruh kriteria calon penerima beasiswa PPA 2017 fakultas MIPA Universitas Lampung. Data fiktif digunakan untuk menunjukkan bahwa algoritma *sorting* yang dibuat telah sesuai dan tidak terjadi kesalahan dalam pengurutan multi prioritas. Pengujian menggunakan data data daftar seluruh kriteria calon penerima beasiswa PPA 2017 digunakan untuk memberikan contoh implementasi algoritma yang digunakan. Pengujian pertama menggunakan data 50 baris dengan 5 kolom. Pengujian kedua menggunakan data 100 baris dengan 5 kolom. Pengujian ketiga menggunakan data 50 baris dengan 8 kolom. Pengujian keempat menggunakan data 100 baris dengan 8 kolom. Pengujian kelima menggunakan data PPA dengan 136 baris dan 6 kolom. Pengujian yang

dilakukan akan mengurutkan data multi kriteria secara *descending* berdasarkan prioritas, kemudian menghitung proses looping yang terjadi dan waktu yang dibutuhkan dari proses looping yang berjalan. Waktu proses looping didapat dengan cara menghitung selisih antara waktu akan mulai looping dengan waktu akhir dari proses looping.

Berikut ini adalah contoh simulasi sederhana dari algoritma *sorting* berdasarkan algoritma yang terdapat pada Kode 3.3 dan Kode 3.4 terhadap *ArrayList* multidimensi yang memiliki 4 baris dan 4 kolom. Misalnya untuk *ArrayList* dengan elemen

<b>90</b>	<b>80</b>	<b>70</b>	<b>80</b>
<b>70</b>	<b>80</b>	<b>90</b>	<b>92</b>
<b>92</b>	<b>90</b>	<b>80</b>	<b>80</b>
<b>70</b>	<b>80</b>	<b>90</b>	<b>98</b>

akan diurutkan secara terurut dari terbesar ke terkecil (*descending*).

Data yang dibandingkan di dalam setiap *looping* ditandai dengan cetak tebal (*bold*). Jika ternyata diperlukan pertukaran nilai, maka data yang ditukar akan diberi warna *background* biru. Hasil pertukaran digambarkan di kolom “Hasil Pertukaran”, data yang baru saja ditukarkan ditandai dengan warna *background* hijau pada kolom “Hasil Pertukaran”.

Untuk membedakan dengan elemen yang lain, data yang telah terurut digambarkan dengan warna *background* coklat. Tanda “*Break*” pada kolom looping k bermakna proses looping k diakhiri, sedangkan tanda “k++”

bermakna proses looping k dilanjutkan. Simulasi algoritma *Buble Sort* pada *ArrayList* Multidimensi ditunjukkan pada Tabel 3.1.

**Tabel 3.1** Simulasi Algoritma *Buble Sort* Pada *ArrayList* Multidimensi

i	j	k	Data Sekarang	Hasil	Hasil!=0	Hasil<0	Hasil Pertukaran	k++/ Break	j++	i++
0	0	0	90 80 70 80 70 80 90 92 92 90 80 80 70 80 90 98	20	Ya	Tidak		Break	j++	
0	1	0	90 80 70 80 70 80 90 92 92 90 80 80 70 80 90 98	-22	Ya	Ya	90 80 70 80 92 90 80 80 70 80 90 92 70 80 90 98	Break	j++	
0	2	0	90 80 70 80 92 90 80 80 70 80 90 92 70 80 90 98	0	Tidak	Tidak		k++		
0	2	1	90 80 70 80 92 90 80 80 70 80 90 92 70 80 90 98	0	Tidak	Tidak		k++		
0	2	2	90 80 70 80 92 90 80 80 70 80 90 92 70 80 90 98	0	Tidak	Tidak		k++		
0	2	3	90 80 70 80 92 90 80 80 70 80 90 92 70 80 90 98	-6	Ya	Ya	90 80 70 80 92 90 80 80 70 80 90 92 70 80 90 98	Break		i++
1	0	0	90 80 70 80 92 90 80 80 70 80 90 98 70 80 90 92	-2	Ya	Ya	92 90 80 80 90 80 70 80 70 80 90 98 70 80 90 92	Break	J++	
1	1	0	92 90 80 80 90 80 70 80 70 80 90 98 70 80 90 92	20	Ya	Tidak		Break		i++
2	0	0	92 90 80 80 90 80 70 80 70 80 90 98 70 80 90 92	2	Ya	Tidak		Break		
Hasil Pengurutan				92 90 80 80 90 80 70 80 70 80 90 98 70 80 90 92						





## V. SIMPULAN DAN SARAN

### 5.1 Simpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat diambil kesimpulan sebagai berikut:

- 1 Algoritma *Bubble Sort* dan *Selection Sort* dapat diimplementasikan menggunakan *ArrayList* multidimensi.
- 2 Dengan menggunakan algoritma *Bubble Sort* dan *Selection Sort* yang diimplementasikan menggunakan *ArrayList* multidimensi, dapat mengurutkan data multi kriteria yang memiliki lebih dari satu kriteria prioritas .
- 3 Implementasi algoritma *Bubble Sort* dan *Selection Sort* menggunakan *ArrayList* multidimensi dapat digunakan untuk mengurutkan data daftar mahasiswa yang layak menerima beasiswa PPA.
- 4 Jumlah iterasi *Selection Sort* pada setiap pengujian data lebih sedikit jika dibandingkan dengan *Bubble Sort*, dan waktu eksekusi *Selection Sort* pada setiap pengujian data juga lebih cepat jika dibandingkan dengan waktu eksekusi *Bubble Sort*.

## 5.2 Saran

Dari hasil penelitian yang dilakukan terdapat beberapa saran sebagai berikut:

1. Pengurutan data multi kriteria yang memiliki lebih dari satu kriteria prioritas, dapat dikembangkan dengan menggunakan algoritma pengurutan lainnya untuk mendapatkan hasil yang lebih baik dari *Bubble Sort* dan *Selection Sort*.
2. Jumlah data yang diuji dapat ditambah untuk mendapatkan hasil yang lebih akurat.

## DAFTAR PUSTAKA

- Agustia, Panny 2016. Analisis Perbandingan Kompleksitas Algoritma Pengurutan Nilai (Sorting), *Jurnal Evolusi* Vol. IV No 2.
- Amsyah Z. 2005. *Manajemen Sistem Informasi*. Jakarta: PT Gramedia Pustaka Utama.
- Fauzi, Indrayana, 2005. Perbandingan Kecepatan/ Waktu Komputasi Beberapa Algoritma Pengurutan (Sorting). Bandung: Institut Teknologi Bandung.
- Indrajani dan Martin. 2007. *Pemrograman Berbasis Objek dengan Bahasa Java*. Jakarta: PT Elex Media Komputindo Kelompok Gramedia.
- Jogiyanto. 2005. *Analisis dan Desain Sistem Informasi*. Yogyakarta: Andi.
- Kadir, Abdul. 2011. *Konsep dan Implementasi Struktur Data dalam Pemrograman Delphi*. Yogyakarta: Andi.
- Kristanto, Andri. 2009. *Struktur Data dengan C++*. Yogyakarta: Graha Ilmu.
- Kochar, Gaurav dan Agrawal, Nikita. 2014. Analysis and Review of Sorting Algorithms, *International Journal Of Scientific Engineering And Research (IJSER)*, Vol 2 Issue 3.
- Utami, E., Raharjo, S., dan Sukrisno. 2007. *Struktur Data Konsep & Implementasinya dalam Bahasa C & Free Pascal di GNU/LINUX*. Yogyakarta: Graha Ilmu.
- Widiantoro, Agustinus. 2014. *Peringkasan Teks Otomatis Pada Dokumen Berbahasa Jawa Menggunakan Metode Tf-Idf*. Yogyakarta: Universitas Sanata Dharma.
- Yahya, Sofyansyah, Yusari 2014, Analisa Perbandingan Algoritma *Bubble Sort* dan *Selection Sort* Dengan Perbandingan Eksponensial, *Jurnal Pelita Informatika Budi Darma*, Vol. VI No 3.