BAB 1

PENDAHULUAN

1.1 Latar Belakang

Penjadwalann produksi merupakan aktivitas yang tidak terpisahkan dalam suatu perusahaan *manufakturing*. Penjadwalan (*scheduling*) sendiri didefinisikan sebagai suatu proses pengalokasian sumber daya atau mesin-mesin yang ada untuk melaksanakan tugas-tugas yang ada dalam suatu waktu tertentu (Baker, 1974). Sedangkan yang dimaksud dengan proses produksi adalah serangkaian langkah-langkah yang digunakan untuk mentransformasikan *Input* menjadi *Output*.

Proses penjadwalan *Flow Shop* adalah salah satu metode penjadwalan produksi di mana urutan mesin yang digunakan untuk setiap proses dalam seluruh pekerjaan harus sama. Dalam penelitian - penelitian penjadwalan sebelumnya hanya difokuskan pada satu kriteria saja (*single*) namun pada penelitian kali ini akan menggunakan lebih dari satu kriteria (*multiple*). Banyak algoritma yang dapat digunakan untuk menentukan urutan pengerjaan pekerjaan dalam proses penjadwalan produksi *Flow Shop*. Salah satu algoritma yang dapat digunakan dalam proses penjadwalan produksi *Multi Objective Flow Shop* adalah algoritma *Ant Colony Optimization*. Algoritma *Ant Colony Optimization* adalah algoritma yang mengadopsi perilaku koloni semut yang dikenal sebagai sistem semut. Algoritma ini menyelesaikan permasalahan berdasarkan tingkah laku semut dalam sebuah koloni yang sedang mencari sumber makanan.

Penelitian ini dibuat untuk mempelajari, mengaplikasikan, serta mengukur kinerja Algoritma *Ant Colony Optimization* pada proses penjadwalan *Multi Objective Flow Shop Scheduling* (MOFSP). Pada skripsi ini juga akan dibuat perangkat lunak yang dapat menerima n job yang masing-masing terdiri atas m buah operasi dan m buah mesin. Setiap operasi hanya ditangani oleh sebuah mesin dan setiap mesin hanya bisa menangani satu operasi. Urutan operasi dari setiap job adalah sama.

1.2 Rumusan Masalah

1. Apa itu penjadwalan *Multi Objective Flowshop Scheduling* (MOFSP) ?
2. Apa itu algoritma *Ant Colony Optimization* (ACO) ?
3. Bagaimana cara kerja dan implementasi algoritma *Ant Colony Optimization* (ACO) dalam menyelesaikan permasalahan MOFSP ?
4. Bagaimana kinerja algoritma *Ant Colony Optimization* (ACO) dalam menyelesaikan permasalahan MOFSP ?

1.3 Tujuan

1. Menjelaskan penjadwalan *Multi Objective Flowshop Scheduling* (MOFSP).
2. Menjelaskan algoritma *Ant Colony Optimization* (ACO) .

3. Menampilkan cara kerja dan implementasi algoritma *Ant Colony Optimization* (ACO) dalam menyelesaikan permasalahan MOFSP .

4. Mengetahui kinerja algoritma *Ant Colony Optimization* (ACO) dalam menyelesaikan permasalahan MOFSP dengan bantuan *benchmark* tertentu.

1.4 Batasan Masalah

Batasan dan asumsi untuk penelitian ini adalah :

1. Waktu proses dari setiap pekerjaan telah diketahui dan bernilai tetap
2. Semua penyelesaian proses dari pekerjaan mengikuti alur proses yang sama dan sistematis
3. Pengerjaan proses dari pekerjaan-pekerjaan yang ada tidak dapat saling mendahului
4. Eksperimen dilakukan dengan sampel data kasus milik *Tailard Benchmark* flow shop dengan jumlah mesin yang beragam.
5. Pengukuran tingkat performansi dari algoritma menggunakan perbandingan nilai makespan dan nilai idle mesin dalam menjalankan suatu job.

1.5 Metodologi

Metodologi penyelesaian masalah dalam penelitian ini adalah :

1. Studi Literatur

   Mencari referensi dari sumber-sumber tertentu untuk memperdalam pemahaman mengenai cara kerja proses penjadwalan flow shop dan cara kerja algoritma ant colony agar kemudian dapat mengaplikasikan algoritma tersebut pada proses penjadwalan flow shop.

2. Analisa Kasus

   Menentukan cara pengaplikasian algoritma ant colony untuk optimisasi penjadwalan flow shop. Menentukan data masukan dan data keluaran yang dibutuhkan oleh perangkat lunak. Menentukan fungsi-fungsi apa saja yang dibutuhkan perangkat lunak.

3. Pengembangan perangkat lunak

   Membentuk struktur kelas dari perangkat lunak. Mendesain interface yang sesuai untuk perangkat lunak. Membangun perangkat lunak untuk optimisasi penjadwalan flow shop dengan algoritma ant colony. Melakukan pengujian fungsional pada perangkat lunak.

4. Eksperimen

   Melakukan proses optimisasi dengan menggunakan perangkat lunak pada beberapa sampel kasus flow shop. Mencatat dan mengolah data hasil proses optimisasi untuk mengukur performa perangkat lunak. Mengukur tingkat keoptimalan dari proses optimisasi yang dilakukan oleh perangkat lunak.

5. Pengambilan kesimpulan

   Mengambil kesimpulan-kesimpulan yang bisa didapatkan dari hasil eksperimen. Melakukan dokumentasi dari skripsi ini.

1.6 Sistematika Pembahasan

Sistematika penulisan karya tulis ini adalah sebagai berikut :

1. Bab 1 : Pendahuluan untuk mendefinisikan masalah yang akan dibahas, alasan pemilihan topik dan usulan solusi terhadap masalah yang ada.
2. Bab 2 : Dasar teori yang digunakan dalam penelitian ini. Pembahasan permasalahan flow shop. Pembahasan cara kerja dari algoritma ant colony. Pembahasan cara pengaplikasian algoritma ant colony untuk optimisasi proses penjadwalan flow shop.
3. Bab 3 : Analisis masalah yang akan dilakukan pada penelitian ini. Pembahasan cara penerapan algoritma ant colony dan rancangan awal dari perangkat lunak.
4. Bab 4 : Perancangan perangkat lunak. Detil informasi mengenai perangkat lunak yang telah dibuat. Struktur kelas dan desain antarmuka grafis dari perangkat lunak yang telah dibuat.
5. Bab 5 : Implementasi dan pengujian. Hasil implementasi algoritma ant colony pada perangkat lunak. Penjelasan cara penggunaan perangkat lunak. Hasil pengujian dan eksperimen kasus flow shop pada perangkat lunak
6. Bab 6 : Kesimpulan dan saran. Hal-hal yang dapat disimpulkan dari penelitian ini. Saran pengembangan yang dapat dilakukan pada penelitian selanjutnya.

BAB 2

LANDASAN TEORI

2.1 Penjadwalan Secara Umum

Secara umum penjadwalan menurut Baker (1974) didefinisikan sebagai proses pengalokasian sumber-sumber dalam jangka waktu tertentu untuk melakukan sekumpulan pekerjaan. Definisi ini mengandung dua arti yang berbeda, yaitu :

1. Penjadwalan merupakan fungsi pengambilan keputusan, yaitu menentukan jadwal.
2. Penjadwalan merupakan suatu teori, yaitu sekumpulan prinsip-prinsip dasar, model-model, teknik-teknik, dan kesimpulan-kesimpulan logis dalam proses pengambilan keputusan yang memberikan dalam fungsi penjadwalan (nilai konseptual).

Tujuan penjadwalan secara umum menurut Baker (1974) adalah :

1. Meningkatkan produktivitas mesin, yaitu dengan mengurangi waktu menganggur mesin.
2. Mengurangi terhadap persediaan barang setengah jadi, dengan mengurangi rata-rata pekerjaan yang menunggu dalam antrian karena mesin sibuk oleh pekerjaan lain.
3. Mengurangi keterlambatan (tardiness). Dalam banyak hal, beberapa atau semua pekerjaan mempunyai batas waktu penyelesaian (duedate). Apabila suatu pekerjaan melewati batas waktu tersebut, maka akan dikenai pinalti. Keterlambatan dapat diperkecil dengan mengurangi maksimal tardiness atau mengurangi pekerjaan yang terlambat (number of tardy job).

Menurut Baker pula masalah penjadwalan muncul karena keterbatasan :

- Waktu
- Tenaga Kerja
- Jumlah Mesin
- Sifat dan syarat pekerja.

2.1.1 Istilah-Istilah dalam Penjadwalan

Bedworth [7] menyebutkan beberapa istilah dalam penjadwalan yang perlu dijelaskan adalah sebagai berikut:

- Waktu Pemrosesan (*Processing Time*)
  Lamanya waktu yang dibutuhkan untuk menyelesaikan satu aktivitas pekerjaan.
- Makespan
  Keseluruhan waktu yang dibutuhkan untuk menyelesaikan aktivitas pekerjaan.

- Waktu Aliran (Flow Time)
  Rentang waktu antara satu pekerjaan dapat dimulai sampai saat dimana pekerjaan tersebut selesai dikerjakan. Sehingga waktu aliran sama dengan waktu pemrosesan ditambah dengan waktu menunggu sebelum diproses.

- Waktu Penyelesaian (Completion Time)
  Waktu dimana tugas terakhir dari satu pekerjaan tertentu selesai dikerjakan.

- Batas Waktu (Due Time)
  Batas waktu penyelesaian untuk suatu pekerjaan (order) dan jika suatu pekerjaan (order) melewati batas waktu tersebut akan dikenakan denda (penalty)

2.1.2 Klasifikasi Masalah Penjadwalan

2.2 Penjadwalan Flow Shop

2.3 Template Skripsi FTIS UNPAR

Akan dipaparkan bagaimana menggunakan template ini, termasuk petunjuk singkat membuat referensi, gambar dan tabel. Juga hal-hal lain yang belum terpikir sampai saat ini.

Nulla in ipsum. Praesent eros nulla, congue vitae, euismod ut, commodo a, wisi. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nonummy magna non leo. Sed felis erat, ullamcorper in, dictum non, ultricies ut, lectus. Proin vel arcu a odio lobortis euismod. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Proin ut est. Aliquam odio. Pellentesque massa turpis, cursus eu, euismod nec, tempor congue, nulla. Duis viverra gravida mauris. Cras tincidunt. Curabitur eros ligula, varius ut, pulvinar in, cursus faucibus, augue.

Nulla mattis luctus nulla. Duis commodo velit at leo. Aliquam vulputate magna et leo. Nam vestibulum ullamcorper leo. Vestibulum condimentum rutrum mauris. Donec id mauris. Morbi molestie justo et pede. Vivamus eget turpis sed nisl cursus tempor. Curabitur mollis sapien condimentum nunc. In wisi nisl, malesuada at, dignissim sit amet, lobortis in, odio. Aenean consequat arcu a ante. Pellentesque porta elit sit amet orci. Etiam at turpis nec elit ultricies imperdiet. Nulla facilisi. In hac habitasse platea dictumst. Suspendisse viverra aliquam risus. Nullam pede justo, molestie nonummy, scelerisque eu, facilisis vel, arcu.

2.3.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh LaTeX, perhatikan contoh di file bab2.tex untuk melihat bagaimana cara memaksa tabel ditempatkan sesuai keinginan kita.

Perhatikan bawa berbeda dengan penempatan judul gambar gambar, keterangan tabel harus diletakkan di atas tabel!! Lihat Tabel 2.1 berikut ini:

Tabel 2.1: Tabel contoh

	v_{start}	$\mathcal{S}_1$	v_{end}
τ_1	1	12	20
τ_2	1		20
τ_3	1	9	20
τ_4	1		20

Tabel 2.2 dan Tabel 2.3 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.2: Tabel bewarna(1)

	v_{start}	$\mathcal{S}_2$	$\mathcal{S}_1$	v_{end}
τ_1	1	5	12	20
τ_2	1	8		20
τ_3	1	2/8/17	9	20
τ_4	1			20

Tabel 2.3: Tabel bewarna(2)

	v_{start}	$\mathcal{S}_1$	$\mathcal{S}_2$	v_{end}
τ_1	1	12	5	20
τ_2	1		8	20
τ_3	1	9	2/8/17	20
τ_4	1			20

2.3.2 Kutipan

Berikut contoh kutipan dari berbagai sumber, untuk keterangan lebih lengkap, silahkan membaca file referensi.bib yang disediakan juga di template ini. Contoh kutipan:

- Buku: [1]
- Bab dalam buku: [2]
- Artikel dari Jurnal: [3]
- Artikel dari prosiding seminar/konferensi: [4]
- Skripsi/Thesis/Disertasi: [5] [6] [7]
- Technical/Scientific Report: [8]
- RFC (Request For Comments): [9]
- Technical Documentation/Technical Manual: [10] [11] [12]
- Paten: [13]
- Tidak dipublikasikan: [14] [15]
- Laman web: [16]
- Lain-lain: [17]

2.3.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen LaTeX tidak dapat dilakukan melalui proses *drag and drop*. Perhatikan contoh pada file bab2.tex untuk melihat bagaimana cara menempatkan gambar. Beberapa hal yang harus diperhatikan pada saat menempatkan gambar:

- Setiap gambar **harus** diacu di dalam teks (gunakan *field* LABEL)
- *Field* CAPTION digunakan untuk teks pengantar pada gambar. Terdapat dua bagian yaitu yang ada di antara tanda [dan] dan yang ada di antara tanda { dan }. Yang pertama akan muncul di Daftar Gambar, sedangkan yang kedua akan muncul di teks pengantar gambar. Untuk skripsi ini, samakan isi keduanya.
- Jenis file yang dapat digunakan sebagai gambar cukup banyak, tetapi yang paling populer adalah tipe PNG (lihat Gambar 2.1), tipe JPG (Gambar 2.2) dan tipe PDF (Gambar 2.3)
- Besarnya gambar dapat diatur dengan *field* SCALE.

Gambar 2.1: Gambar *Serpentes* dalam format png

- Penempatan gambar diatur menggunakan *placement specifier* (di antara tanda [dan] setelah deklarasi gambar. Yang umum digunakan adalah **H** untuk menempatkan gambar **sesuai** penempatannya di file .tex atau **h** yang berarti "kira-kira" di sini.
  Jika tidak menggunakan *placement specifier*, LaTeX akan menempatkan gambar secara otomatis untuk menghindari bagian kosong pada dokumen anda. Walaupun cara ini sangat mudah, hindarkan terjadinya penempatan dua gambar secara berurutan.
  - Gambar 2.1 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
  - Gambar 2.2 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
  - Gambar 2.3 ditempatkan menggunakan *specifier* **h**.

Curabitur tellus magna, porttitor a, commodo a, commodo in, tortor. Donec interdum. Praesent scelerisque. Maecenas posuere sodales odio. Vivamus metus lacus, varius quis, imperdiet quis, rhoncus a, turpis. Etiam ligula arcu, elementum a, venenatis quis, sollicitudin sed, metus. Donec nunc pede, tincidunt in, venenatis vitae, faucibus vel, nibh. Pellentesque wisi. Nullam malesuada. Morbi ut tellus ut pede tincidunt porta. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam congue neque id dolor.

Donec et nisl at wisi luctus bibendum. Nam interdum tellus ac libero. Sed sem justo, laoreet vitae, fringilla at, adipiscing ut, nibh. Maecenas non sem quis tortor eleifend fermentum. Etiam id tortor ac mauris porta vulputate. Integer porta neque vitae massa. Maecenas tempus libero a libero posuere dictum. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aenean quis mauris sed elit commodo placerat. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Vivamus rhoncus tincidunt libero. Etiam elementum pretium justo. Vivamus est. Morbi a tellus eget pede tristique commodo. Nulla nisl. Vestibulum sed nisl eu sapien cursus rutrum.

Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. Proin accumsan imperdiet sem. Nunc porta. Donec feugiat mi at justo. Phasellus facilisis ipsum quis ante. In ac elit eget ipsum pharetra faucibus. Maecenas viverra nulla in massa.

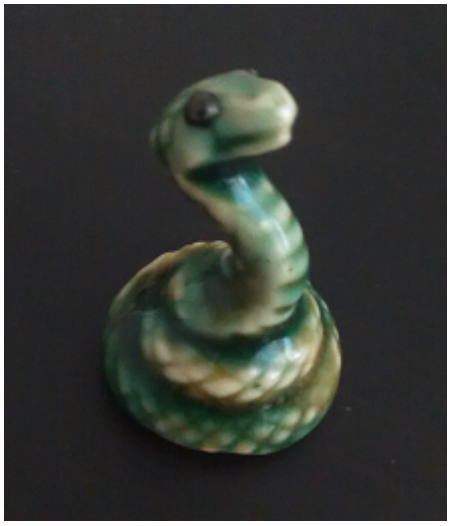
Nulla ac nisl. Nullam urna nulla, ullamcorper in, interdum sit amet, gravida ut, risus. Aenean

ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.

Gambar 2.2: Ular kecil

Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetuer quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.

Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.



Gambar 2.3: *Serpentes* jantan

DAFTAR REFERENSI

[1] de Berg, M., Cheong, O., van Kreveld, M. J., dan Overmars, M. (2008) *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag, Berlin.

[2] van Kreveld, M. J. (2004) Geographic information systems. Bagian dari Goodman, J. E. dan O'Rourke, J. (ed.), *Handbook of Discrete and Computational Geometry*. Chapman & Hall/CRC, Boca Raton.

[3] Buchin, K., Buchin, M., van Kreveld, M. J., Löffler, M., Silveira, R. I., Wenk, C., dan Wiratma, L. (2013) Median trajectories. *Algorithmica*, **66**, 595–614.

[4] van Kreveld, M. J. dan Wiratma, L. (2011) Median trajectories using well-visited regions and shortest paths. *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Chicago, USA, 1-4 November, pp. 241–250. ACM, New York.

[5] Lionov (2002) Animasi algoritma sweepline untuk membangun diagram voronoi. Skripsi. Universitas Katolik Parahyangan, Indonesia.

[6] Wiratma, L. (2010) Following the majority: a new algorithm for computing a median trajectory. Thesis. Utrecht University, The Netherlands.

[7] Wiratma, L. (2022) Coming Not Too Soon, Later, Delay, Someday, Hopefully. Disertasi. Utrecht University, The Netherlands.

[8] van kreveld, M., van Lankveld, T., dan Veltkamp, R. (2013) Watertight scenes from urban lidar and planar surfaces. Technical Report UU-CS-2013-007. Utrecht University, The Netherlands.

[9] Rekhter, Y. dan Li, T. (1994) A border gateway protocol 4 (bgp-4). RFC 1654. RFC Editor, http://www.rfc-editor.org.

[10] ITU-T Z.500 (1997) *Framework on formal methods in conformance testing*. International Telecommunications Union. Geneva, Switzerland.

[11] Version 9.0.0 (2016) *The Unicode Standard*. The Unicode Consortium. Mountain View, USA.

[12] Version 7.0 Nougat (2016) *Android API Reference Manual*. Google dan Open Handset Alliance. Mountain View, USA.

[13] Webb, R., Daruca, O., dan Alfadian, P. (2012) *Method of optimizing a text massage communication between a server and a secure element*. Paten no. EP2479956 (A1). European Patent Organisation. Munich, Germany.

[14] Wiratma, L. (2009) Median trajectory. Report for GMT Experimentation Project at Utrecht University.

[15] Lionov (2011) Polymorphism pada C++. Catatan kuliah AKS341 Pemrograman Sistem di Universitas Katolik Parahyangan, Bandung. http://tinyurl.com/lionov. 30 September 2016.

[16] Erickson, J. (2003) CG models of computation? http://www.computational-geometry.org/mailing-lists/compgeom-announce/2003-December/000852.html. 30 September 2016.

[17] AGUNG (2012) Menjajal tango 12. Majalah HAI no 02, Januari 2012.

LAMPIRAN A

KODE PROGRAM

Listing A.1: MyCode.c

```c

// This does not make algorithmic sense,
// but it shows off significant programming characters.

#include<stdio.h>

void myFunction( int input, float* output ) {
    switch ( array[i] ) {
        case 1: // This is silly code
            if ( a >= 0 || b <= 3 && c != x )
                *output += 0.005 + 20050;
            char = 'g';
            b = 2^n + ~right_size - leftSize * MAX_SIZE;
            c = (--aaa + &daa) / (bbb++ - ccc % 2 );
            strcpy(a,"hello $@?");
    }
    count = ~mask | 0x00FF00AA;
}

// Fonts for Displaying Program Code in LATEX
// Adrian P. Robson, nepsweb.co.uk
// 8 October 2012
// http://nepsweb.co.uk/docs/progfonts.pdf
```

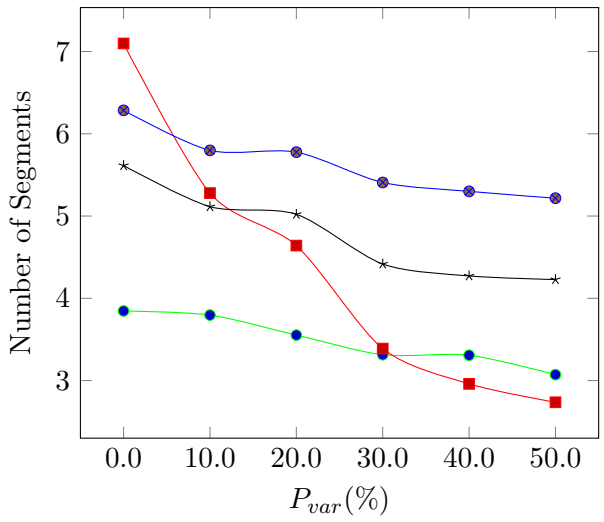
Listing A.2: MyCode.java

```java
import java.util.ArrayList;
import java.util.Collections;
import java.util.HashSet;

//class for set of vertices close to furthest edge
public class MyFurSet {
    protected int id;                                   //id of the set
    protected MyEdge FurthestEdge;                      //the furthest edge
    protected HashSet<MyVertex> set;                    //set of vertices close to furthest edge
    protected ArrayList<ArrayList<Integer>> ordered;    //list of all vertices in the set for each trajectory
    protected ArrayList<Integer> closeID;               //store the ID of all vertices
    protected ArrayList<Double> closeDist;              //store the distance of all vertices
    protected int totaltrj;                             //total trajectories in the set

    /*
     * Constructor
     * @param id : id of the set
     * @param totaltrj : total number of trajectories in the set
     * @param FurthestEdge : the furthest edge
     */
    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
        this.id = id;
        this.totaltrj = totaltrj;
        this.FurthestEdge = FurthestEdge;
        set = new HashSet<MyVertex>();
        ordered = new ArrayList<ArrayList<Integer>>();
        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
        closeID = new ArrayList<Integer>(totaltrj);
        closeDist = new ArrayList<Double>(totaltrj);
        for (int i = 0;i <totaltrj;i++) {
            closeID.add(-1);
            closeDist.add(Double.MAX_VALUE);
        }
    }

}
```

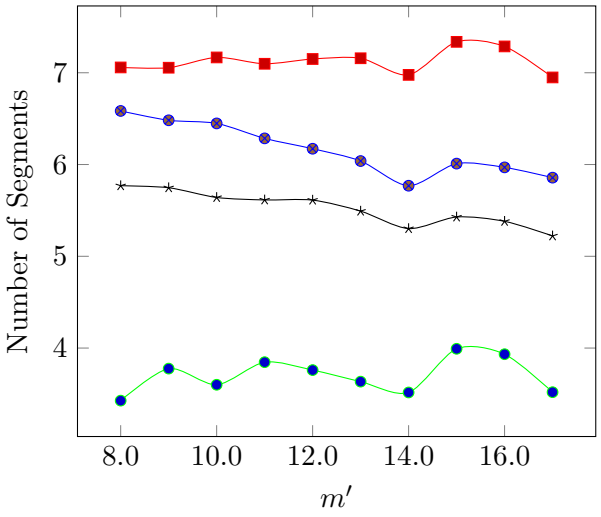
LAMPIRAN B

HASIL EKSPERIMEN

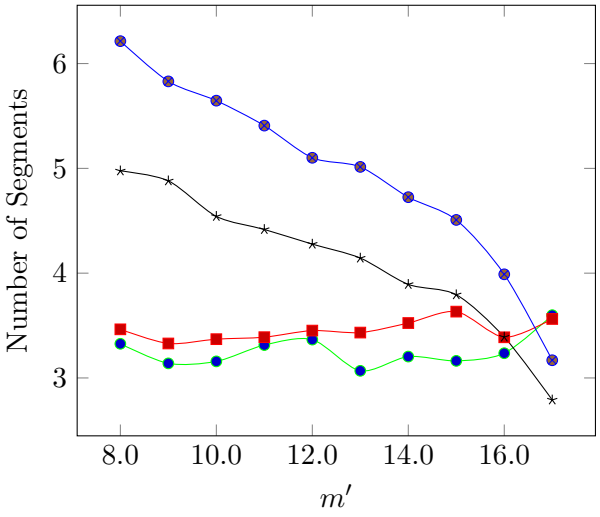
Hasil eksperimen berikut dibuat dengan menggunakan TIKZPICTURE (bukan hasil excel yg diubah ke file bitmap). Sangat berguna jika ingin menampilkan tabel (yang kuantitasnya sangat banyak) yang datanya dihasilkan dari program komputer.



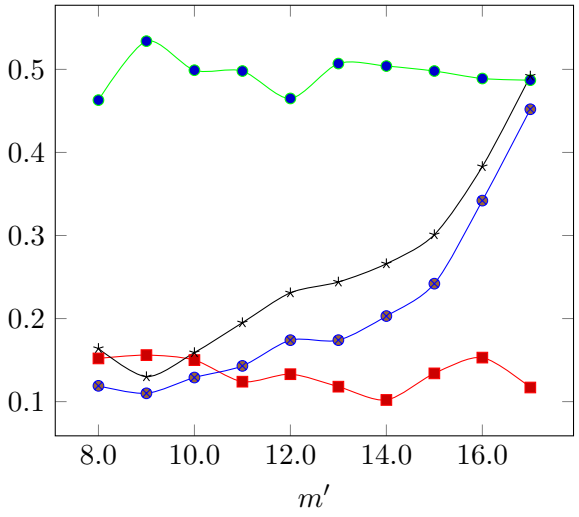
Gambar B.1: Hasil 1



Gambar B.2: Hasil 2



Gambar B.3: Hasil 3



Gambar B.4: Hasil 4