

BAB 1

PENDAHULUAN

Pada bab ini, akan dibahas mengenai latar belakang penelitian, rumusan masalah, tujuan, batasan masalah, dan sistematika pembahasan yang dilakukan pada penelitian ini.

1.1 Latar Belakang

Online judge merupakan sebuah sistem yang dirancang untuk melakukan evaluasi dari sumber kode algoritma yang dikirimkan oleh pengguna [1]. Konsep dari *online judge* adalah dengan asumsi pengguna mengirimkan solusi berupa kode program, atau bahkan pengguna mengirimkan *file* yang dapat dijalankan dimana tahap selanjutnya kode atau *file* tersebut akan dievaluasi yang sering kali menggunakan infrastruktur berbasis *cloud*. Dalam perancangan *online judge* harus dipastikan bahwa *online judge* harus dapat menanggulangi berbagai macam serangan seperti memaksakan waktu kompilasi yang tinggi, atau mengakses sumber daya yang dibatasi selama proses evaluasi berlangsung.

Sharif Judge merupakan salah satu *online judge* yang gratis untuk bahasa pemrograman C, C++, Java dan Python. Perangkat lunak ini diciptakan oleh Mohammad Javad Naderi pada tahun 2014 dan bersifat *open source*. Antarmuka Sharif Judge ditulis menggunakan bahasa pemrograman PHP (framework CodeIgniter) dan backend menggunakan BASH¹.

Sharif Judge biasa digunakan untuk mempermudah evaluasi kode program. dan salah satu universitas yang menggunakannya adalah Universitas Katolik Parahyangan (UNPAR) pada jurusan Informatika (IF). Namun Sharif Judge telah dimodifikasi dan berubah namanya menjadi SharIF-Judge. SharIF-Judge ini digunakan pada beberapa kuliah di IF, seperti Dasar Pemrograman dan Algoritma Struktur Data. Tujuan SharIF-Judge digunakan pada perkuliahan, adalah agar dapat mempermudah kegiatan belajar mengajar berlangsung. Mahasiswa dapat dengan mudah melakukan pengiriman kode program atau *file* ke SharIF-Judge dan dapat langsung melihat nilainya. Pengajar yang bertanggung jawab atas kegiatan belajar mengajar tersebut tidak akan kesusahan dalam mengumpulkan hasil pekerjaan para pelajar.

Alur dari penggunaan SharIF-Judge ini diawali dengan pengajar yang membuat soal terlebih dahulu. Setelah soal disiapkan, pengajar dapat membuat *assignment* dengan click tombol *add assignment*. Didalamnya, pengajar diwajibkan untuk memasukan nama *assignment*, waktu dimulainya pengerjaan, waktu selesai pengerjaan, waktu tambahan pengerjaan, daftar peserta, deskripsi soal, dan kunci jawaban dari soal yang sudah dibuat oleh pengajar. Meskipun SharIF-Judge dapat membaca berbagai bahasa pemrograman, JavaScript bukan salah satu yang dapat dibaca oleh SharIF-Judge.

JavaScript adalah bahasa skrip sisi klien yang berjalan sepenuhnya di dalam *browser web*[2]. Sebagai contoh dapat dilihat *menu drop-down* yang mencolok, memindahkan teks, dan mengubah konten yang sekarang tersebar luas di situs web. Semua interaksi tersebut diaktifkan melalui JavaScript. Berdasarkan survey dari <https://insights.stackoverflow.com/survey/2021>, selama 9 tahun berturut-turut, JavaScript merupakan bahasa pemrograman yang paling sering digunakan. 64,96% orang di dunia menggunakannya.

¹<https://github.com/mjnaderi/Sharif-Judge>

Berdasarkan data diatas, ditunjukkan bahwa JavaScript termasuk dalam bahasa pemrograman yang sangat populer dan banyak digunakan. Sangat disayangkan jika SharIF Judge tidak dapat membaca bahasa tersebut. Akan kesulitan bagi para pengajar karena harus mengikuti perkembangan zaman, namun pemeriksaannya tidak mudah. Kostumisasi dari SharIF Judge akan dilakukan agar *online judge* tersebut dapat membaca JavaScript. Untuk mempermudah kostumasi, pengerjaan akan dilakukan dengan menggunakan OS Linux. Namun dikarenakan OS yang digunakan adalah Windows, akan digunakan *virtual machine* agar dapat mengerjakan dengan OS Linux pada Windows.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka terbentuklah rumusan masalah penelitian sebagai berikut:

Bagaimana cara agar SharIF Judge dapat memahami bahasa pemrograman JavaScript untuk dievaluasi.

1.3 Tujuan

Berdasarkan rumusan masalah, maka tujuan penelitian ini adalah sebagai berikut:

1. Mempelajari cara kerja dan merancang soal pada Sharif Judge.
2. Mempelajari kode program Sharif Judge agar dapat mengintegrasikan JavaScript.
3. Memodifikasi Sharif Judge agar dapat memasukan soal JavaScript dan dapat mengevaluasi JavaScript.

1.4 Batasan Masalah

Batasan masalah yang terdapat pada penelitian ini adalah:

1. Pembuatan program menggunakan docker sebagai penyambung Linux karena dalam pembuatan program menggunakan Windows.
2. Versi PHP 5.3 atau lebih.

1.5 Metodologi

Metodologi yang digunakan dalam penelitian ini adalah sebagai berikut:

1. melakukan studi literatur terhadap Sharif Judge, JavaScript, Java
2. Mempelajari dan menganalisa cara pembuatan soal pada Sharif Judge dengan menggunakan bahasa pemrograman Java
3. Mempelajari cara membuat tes kasus dan menganalisa cara kerjanya pada Sharif Judge dengan menggunakan bahasa pemrograman Java
4. Membuat kode program agar Sharif Judge dapat membaca soal dari JavaScript dan dapat mengevaluasi berdasarkan tes kasus.
5. Melakukan testing terhadap Sharif Judge
6. Melaporkan hasil pembuatan dalam bentuk dokumen skripsi

1.6 Sistematika Pembahasan

Rencananya Bab 2 akan berisi petunjuk penggunaan template dan dasar-dasar L^AT_EX. Mungkin bab 3,4,5 dapt diisi oleh ketiga jurusan, misalnya peraturan dasar skripsi atau pedoman penulisan, tentu jika berkenan. Bab 6 akan diisi dengan kesimpulan, bahwa membuat template ini ternyata sungguh menghabiskan banyak waktu.

BAB 2

LANDASAN TEORI

2.1 Skripsi

Rencananya akan diisi dengan penjelasan umum mengenai buku skripsi.

Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetur eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetur tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.

2.2 L^AT_EX

Mengapa menggunakan L^AT_EX untuk buku skripsi dan apa keunggulan/kerugiannya bagi mahasiswa dan pembuat template.

Aliquam lectus. Vivamus leo. Quisque ornare tellus ullamcorper nulla. Mauris porttitor pharetra tortor. Sed fringilla justo sed mauris. Mauris tellus. Sed non leo. Nullam elementum, magna in cursus sodales, augue est scelerisque sapien, venenatis congue nulla arcu et pede. Ut suscipit enim vel sapien. Donec congue. Maecenas urna mi, suscipit in, placerat ut, vestibulum ut, massa. Fusce ultrices nulla et nisl.

Etiam ac leo a risus tristique nonummy. Donec dignissim tincidunt nulla. Vestibulum rhoncus molestie odio. Sed lobortis, justo et pretium lobortis, mauris turpis condimentum augue, nec ultricies nibh arcu pretium enim. Nunc purus neque, placerat id, imperdiet sed, pellentesque nec, nisl. Vestibulum imperdiet neque non sem accumsan laoreet. In hac habitasse platea dictumst. Etiam condimentum facilisis libero. Suspendisse in elit quis nisl aliquam dapibus. Pellentesque auctor sapien. Sed egestas sapien nec lectus. Pellentesque vel dui vel neque bibendum viverra. Aliquam porttitor nisl nec pede. Proin mattis libero vel turpis. Donec rutrum mauris et libero. Proin euismod porta felis. Nam lobortis, metus quis elementum commodo, nunc lectus elementum mauris, eget vulputate ligula tellus eu neque. Vivamus eu dolor.

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 L^AT_EX, 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: [3]
- Bab dalam buku: [4]
- Artikel dari Jurnal: [?]
- Artikel dari prosiding seminar/konferensi: [5]
- Skripsi/Thesis/Disertasi: [6] [7] [8]
- Technical/Scientific Report: [9]
- RFC (Request For Comments): [10]
- Technical Documentation/Technical Manual: [11] [12] [13]

- Paten: [14]
- Tidak dipublikasikan: [15] [16]
- Laman web: [17]
- Lain-lain: [18]

2.3.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen L^AT_EX 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*.
 - 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*, L^AT_EX 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, consectetur 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



Gambar 2.1: Gambar *Serpentes* dalam format png

1 at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque
 2 scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur adipiscing elit.
 3 Nunc ut metus. Ut metus justo, auctor at, ultrices eu, sagittis ut, purus. Aliquam aliquam.



Gambar 2.2: Ular kecil

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

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

18 2.3.4 Kode Program

19 Kode program dalam bahasa tertentu seringkali harus ditulis di dalam bab, bukan hanya dilampirkan
 20 di bagian Lampiran. Kode 2.1 menampilkan penggunaan karakter-karakter yang umum digunakan
 21 dalam sebuah program yang ditulis dengan bahasa C.

Gambar 2.3: *Serpentes* jantan

Kode 2.1: Kode untuk menampilkan karakter-karakter aneh

```

1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11                 char = 'g';
12                 b = 2^n + ~right_size - leftSize * MAX_SIZE;
13                 c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14                 strcpy(a,"hello_$@?");
15             }
16             count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf

```

2.3.5 Notasi

Simbol-simbol (matematika) yang sering digunakan sepanjang penulisan skripsi, dapat dimasukkan ke dalam “Daftar Notasi”. Daftar ini ada di halaman depan sebelum Bab 1. Cara memasukkan sebuah simbol ke dalam Daftar Notasi adalah menggunakan perintah `\nomenclature`. Contoh:

```
\nomenclature[]{$A$}{luas kandang ular}
```

Argumen opsional digunakan untuk mengurutkan notasi. Silahkan lihat sendiri dokumentasi package `nomenc1`

DAFTAR REFERENSI

- [1] Kurnia, A., Lim, A., dan Cheang, B. (2001) Online judge. *Computers & Education*, **36**, 299–315.
- [2] Crockford, D. (2008) *JavaScript: The Good Parts: The Good Parts*. " O'Reilly Media, Inc."
- [3] de Berg, M., Cheong, O., van Kreveld, M. J., dan Overmars, M. (2008) *Computational Geometry: Algorithms and Applications*, 3rd edition. Springer-Verlag, Berlin.
- [4] 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.
- [5] 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.
- [6] Lionov (2002) Animasi algoritma sweepline untuk membangun diagram voronoi. Skripsi. Universitas Katolik Parahyangan, Indonesia.
- [7] Wiratma, L. (2010) Following the majority: a new algorithm for computing a median trajectory. Thesis. Utrecht University, The Netherlands.
- [8] Wiratma, L. (2022) Coming Not Too Soon, Later, Delay, Someday, Hopefully. Disertasi. Utrecht University, The Netherlands.
- [9] 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.
- [10] Rekhter, Y. dan Li, T. (1994) A border gateway protocol 4 (bgp-4). RFC 1654. RFC Editor, <http://www.rfc-editor.org>.
- [11] ITU-T Z.500 (1997) *Framework on formal methods in conformance testing*. International Telecommunications Union. Geneva, Switzerland.
- [12] Version 9.0.0 (2016) *The Unicode Standard*. The Unicode Consortium. Mountain View, USA.
- [13] Version 7.0 Nougat (2016) *Android API Reference Manual*. Google dan Open Handset Alliance. Mountain View, USA.
- [14] Webb, R., Daruca, O., dan Alfadian, P. (2012) *Method of optimizing a text message communication between a server and a secure element*. Paten no. EP2479956 (A1). European Patent Organisation. Munich, Germany.
- [15] Wiratma, L. (2009) Median trajectory. Report for GMT Experimentation Project at Utrecht University.

- [16] Lionov (2011) Polymorphism pada C++. Catatan kuliah AKS341 Pemrograman Sistem di Universitas Katolik Parahyangan, Bandung. <http://tinyurl.com/lionov>. 30 September 2016.
- [17] Erickson, J. (2003) CG models of computation? <http://www.computational-geometry.org/mailling-lists/compgeom-announce/2003-December/000852.html>. 30 September 2016.
- [18] AGUNG (2012) Menjajal tango 12. Majalah HAI no 02, Januari 2012.

LAMPIRAN A

KODE PROGRAM

Kode A.1: MyCode.c

```
1 // This does not make algorithmic sense,
2 // but it shows off significant programming characters.
3
4 #include<stdio.h>
5
6 void myFunction( int input, float* output ) {
7     switch ( array[i] ) {
8         case 1: // This is silly code
9             if ( a >= 0 || b <= 3 && c != x )
10                 *output += 0.005 + 20050;
11             char = 'g';
12             b = 2^n + ~right_size - leftSize * MAX_SIZE;
13             c = (--aaa + &daa) / (bbb++ - ccc % 2 );
14             strcpy(a,"hello_$@?");
15         }
16         count = ~mask | 0x00FF00AA;
17     }
18 }
19
20 // Fonts for Displaying Program Code in LATEX
21 // Adrian P. Robson, nepsweb.co.uk
22 // 8 October 2012
23 // http://nepsweb.co.uk/docs/progfonts.pdf
```

Kode A.2: MyCode.java

```
1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.HashSet;
4
5 //class for set of vertices close to furthest edge
6 public class MyFurSet {
7     protected int id; //id of the set
8     protected MyEdge FurthestEdge; //the furthest edge
9     protected HashSet<MyVertex> set; //set of vertices close to furthest edge
10    protected ArrayList<ArrayList<Integer>> ordered; //list of all vertices in the set for each trajectory
11    protected ArrayList<Integer> closeID; //store the ID of all vertices
12    protected ArrayList<Double> closeDist; //store the distance of all vertices
13    protected int totaltrj; //total trajectories in the set
14
15    /*
16     * Constructor
17     * @param id : id of the set
18     * @param totaltrj : total number of trajectories in the set
19     * @param FurthestEdge : the furthest edge
20     */
21    public MyFurSet(int id,int totaltrj,MyEdge FurthestEdge) {
22        this.id = id;
23        this.totaltrj = totaltrj;
24        this.FurthestEdge = FurthestEdge;
25        set = new HashSet<MyVertex>();
26        ordered = new ArrayList<ArrayList<Integer>>();
27        for (int i=0;i<totaltrj;i++) ordered.add(new ArrayList<Integer>());
28        closeID = new ArrayList<Integer>(totaltrj);
29        closeDist = new ArrayList<Double>(totaltrj);
30        for (int i = 0;i <totaltrj;i++) {
31            closeID.add(-1);
32            closeDist.add(Double.MAX_VALUE);
33        }
34    }
35
36 }
```


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