

«SKRIPSI/TUGAS AKHIR»

«JUDUL BAHASA INDONESIA»



«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

PROGRAM STUDI «MATEMATIKA/FISIKA/TEKNIK INFORMATIKA»  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN  
«tahun»



«FINAL PROJECT/UNDERGRADUATE THESIS»

«JUDUL BAHASA INGGRIS»



«Nama Lengkap»

NPM: «10 digit NPM UNPAR.»

DEPARTMENT OF «MATHEMATICS/PHYSICS/INFORMATICS»  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY  
«tahun»



# LEMBAR PENGESAHAN

«JUDUL BAHASA INDONESIA»

«Nama Lengkap»

NPM: «10 digit NPM UNPAR»

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

«pembimbing utama/1»

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

«Ketua Program Studi»



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa «skripsi/tugas akhir» dengan judul:

**«JUDUL BAHASA INDONESIA»**

adalah benar-benar karya saya sendiri, dan saya tidak melakukan penjiplakan atau pengutipan dengan cara-cara yang tidak sesuai dengan etika keilmuan yang berlaku dalam masyarakat keilmuan.

Atas pernyataan ini, saya siap menanggung segala risiko dan sanksi yang dijatuhkan kepada saya, apabila di kemudian hari ditemukan adanya pelanggaran terhadap etika keilmuan dalam karya saya, atau jika ada tuntutan formal atau non-formal dari pihak lain berkaitan dengan keaslian karya saya ini.

Dinyatakan di Bandung,  
Tanggal «tanggal» «bulan» «tahun»

|                     |
|---------------------|
| Meterai<br>Rp. 6000 |
|---------------------|

«Nama Lengkap»  
NPM: «10 digit NPM UNPAR»





## ABSTRAK

«Tuliskan abstrak anda di sini, dalam bahasa Indonesia»

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

**Kata-kata kunci:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Indonesia»



## ABSTRACT

«Tuliskan abstrak anda di sini, dalam bahasa Inggris»

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

**Keywords:** «Tuliskan di sini kata-kata kunci yang anda gunakan, dalam bahasa Inggris»



*«kepada siapa anda mempersembahkan skripsi ini...?»*



## KATA PENGANTAR

«Tuliskan kata pengantar dari anda di sini ...»

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetur adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Bandung, «bulan» «tahun»

Penulis





## DAFTAR ISI

|                                      |             |
|--------------------------------------|-------------|
| <b>KATA PENGANTAR</b>                | <b>xv</b>   |
| <b>DAFTAR ISI</b>                    | <b>xvii</b> |
| <b>DAFTAR GAMBAR</b>                 | <b>xix</b>  |
| <b>DAFTAR TABEL</b>                  | <b>xxi</b>  |
| <b>1 PENDAHULUAN</b>                 | <b>1</b>    |
| 1.1 Latar Belakang . . . . .         | 1           |
| 1.2 Rumusan Masalah . . . . .        | 2           |
| 1.3 Tujuan . . . . .                 | 2           |
| 1.4 Batasan Masalah . . . . .        | 2           |
| 1.5 Metodologi . . . . .             | 2           |
| 1.6 Sistematika Pembahasan . . . . . | 3           |
| <b>2 LANDASAN TEORI</b>              | <b>5</b>    |
| 2.1 WebSockets . . . . .             | 5           |
| 2.1.1 Interface . . . . .            | 5           |
| 2.1.2 WebSocket Events . . . . .     | 5           |
| 2.1.3 WebSocket Methods . . . . .    | 5           |
| 2.2 Socket.io . . . . .              | 5           |
| 2.2.1 Connection . . . . .           | 6           |
| 2.2.2 Messages . . . . .             | 6           |
| 2.3 Node.js . . . . .                | 6           |
| 2.4 Express.js . . . . .             | 6           |
| 2.5 Canvas API . . . . .             | 6           |
| 2.5.1 Tabel . . . . .                | 6           |
| 2.5.2 Kutipan . . . . .              | 7           |
| 2.5.3 Gambar . . . . .               | 7           |
| <b>A KODE PROGRAM</b>                | <b>11</b>   |
| <b>B HASIL EKSPERIMEN</b>            | <b>13</b>   |



## DAFTAR GAMBAR

|     |  |    |
|-----|--|----|
| 2.1 | Gambar <i>Serpentes</i> dalam format png . . . . . | 9  |
| 2.2 | Ular kecil . . . . .                               | 9  |
| 2.3 | <i>Serpentes</i> betina . . . . .                  | 10 |
| B.1 | Hasil 1 . . . . .                                  | 13 |
| B.2 | Hasil 2 . . . . .                                  | 13 |
| B.3 | Hasil 3 . . . . .                                  | 13 |
| B.4 | Hasil 4 . . . . .                                  | 13 |



## DAFTAR TABEL

|     |                            |   |
|-----|----------------------------|---|
| 2.1 | Tabel contoh . . . . .     | 7 |
| 2.2 | Tabel bewarna(1) . . . . . | 7 |
| 2.3 | Tabel bewarna(2) . . . . . | 7 |



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

*WebSockets* adalah teknologi yang memungkinkan *web browser* pengguna dan *web server* membuka sesi komunikasi interaktif satu sama lain. Teknologi *WebSockets* didesain untuk diimplementasikan pada *web browser* dan *web server*, tetapi dapat juga digunakan oleh setiap aplikasi *client* maupun *server*. *WebSockets* memiliki standar yang menyediakan cara agar *web server* dapat mengirim konten ke *web browser* tanpa diminta oleh *client*, dan memungkinkan agar pesan dikirimkan berulang-ulang dengan tetap menjaga koneksi yang terbuka. Oleh karena itu, protokol *WebSockets* memungkinkan interaksi antara *web browser* dan *web server* dengan *overhead* yang rendah, dan juga memfasilitasi transfer data *realtime* dari *server* maupun menuju *server*.

Salah satu teknologi yang memanfaatkan protokol *WebSockets* adalah *Socket.io*. Teknologi ini memungkinkan untuk melakukan komunikasi secara *realtime*, dan dua arah antara *client* dan *server*. *Socket.io* memiliki dua bagian: *client-side library* yang berjalan didalam *web browser*, dan *server-side library* yang berjalan pada *Node.js*. *Socket.io* memiliki fitur-fitur yang beragam, seperti melakukan broadcast ke beberapa *sockets*, dan menyimpan data yang berhubungan dengan masing-masing *client*. Teknologi ini sangat berguna untuk membantu membangun sebuah aplikasi yang membutuhkan koneksi *realtime* seperti dalam aplikasi *chatting* maupun *game*.

Untuk memanfaatkan protokol *WebSockets* dalam membangun aplikasi permainan, akan dibutuhkan beberapa teknologi yang dapat membantu pembangunan aplikasinya. Salah satu teknologi tersebut yaitu *HTML5 Canvas*. Teknologi ini merupakan bagian dari *HTML5 element* yang dapat digunakan untuk menggambar suatu grafis melalui *JavaScript* secara *on the fly*. *HTML5 Canvas* dapat juga digunakan untuk membuat komposisi foto, membuat animasi, dan membuat *real-time video processing* atau *rendering*. Oleh karena itu, fungsi-fungsi yang ada pada *HTML5 Canvas* akan membantu pembangunan aplikasi permainan terutama pada bagian pengembangan grafis pada aplikasinya.

Teknologi lain yang dapat membantu membangun aplikasi permainan dalam memanfaatkan protokol *WebSockets* yaitu *Node.js*. Teknologi ini merupakan sebuah *platform* yang didesain untuk mengembangkan aplikasi berbasis web pada bagian *web server*. *Node.js* ditulis dalam sintaks bahasa pemrograman *JavaScript* dan menggunakan *V8* yang merupakan *engine JavaScript* milik perusahaan *Google* untuk mengeksekusi *JavaScript* pada *web server*. *Node.js* memiliki sifat *non-blocking*, yang berarti *Node.js* tidak akan menunggu untuk mengerjakan *request* selanjutnya. *Node.js* pun sangat cepat dalam mengeksekusi suatu kode karena menggunakan *engine JavaScript V8*. Fitur-fitur yang dimiliki oleh *Node.js* akan sangat membantu untuk membangun aplikasi permainan yang membutuhkan koneksi *real-time*.

Pada skripsi ini, akan dibuat sebuah aplikasi permainan yang memanfaatkan protokol *WebSockets*, dimana dalam penggunaan protokol tersebut akan dibantu dengan teknologi *Socket.io*. Selain itu, aplikasi yang dibuat akan memanfaatkan *personal computer (PC)* dan *smartphone* untuk pengembangan aplikasinya. Para pemain akan mengkoneksikan *smartphone* miliknya pada suatu *PC*, dimana *smartphone* tersebut akan berfungsi sebagai *controller* untuk memainkan permainannya. Oleh karena itu, protokol *WebSockets* akan digunakan sebagai koneksi antara *smartphone* dan *PC*.

dalam aplikasi permainan yang akan dibangun. Aplikasi permainan akan menggunakan teknologi berbasis web, sehingga untuk memainkannya, *client* bisa mengakses melalui *web browser* tanpa harus berada di satu jaringan lokal yang sama.

## 1.2 Rumusan Masalah

1. Bagaimana membangun aplikasi permainan berbasis web dengan memanfaatkan protokol *WebSockets* untuk penggunaan *smartphone* sebagai pengendali permainan berbasis web ?
2. Berapa *latency* yang dihasilkan berdasarkan penggunaan protokol *WebSockets* ?

## 1.3 Tujuan

1. Mengetahui cara membangun aplikasi permainan berbasis web dengan memanfaatkan protokol *WebSockets* untuk penggunaan *smartphone* sebagai pengendali permainan berbasis web.
2. Mengetahui jumlah *latency* yang dihasilkan berdasarkan pemanfaatan protokol *WebSockets*.

## 1.4 Batasan Masalah

Batasan masalah yang dibuat terkait dengan pengerjaan skripsi ini adalah sebagai berikut:

1. Aplikasi permainan yang dibuat merupakan permainan *multiplayer* yang hanya bisa dimainkan oleh dua orang saja.

## 1.5 Metodologi

Metodologi yang dilakukan dalam pengerjaan skripsi ini adalah sebagai berikut:

1. Studi literatur mengenai :
  - *WebSockets* yang akan digunakan untuk koneksi antara *smartphone* dan *PC*.
  - *Socket.io* sebagai teknologi yang akan menggunakan *WebSockets* dalam pembangunan aplikasi.
  - *HTML5 Canvas* yang akan digunakan untuk antarmuka permainan.
  - *NodeJs* sebagai *web server* dalam pembangunan aplikasi.
2. Menganalisis aplikasi sejenis.
3. Merancang antarmuka permainan pada *PC* dan *smartphone*. Antarmuka pada *PC* akan berbeda dengan yang ada di *smartphone*, karena *smartphone* akan bekerja sebagai *controller* dan *PC* akan bekerja sebagai *console*.
4. Menyusun cara bermain aplikasi permainan yang dibangun.
5. Mengimplementasi program aplikasi permainan berbasis web.
6. Menganalisis *latency* yang dihasilkan pada aplikasi.
7. Melakukan eksperimen dan pengujian yang melibatkan responden.



## 1.6 Sistematika Pembahasan

Setiap bab dalam skripsi ini memiliki sistematika penulisan yang dijelaskan kedalam poin-poin sebagai berikut:

1. Bab 1 : Pendahuluan  
Membahas mengenai gambaran umum penelitian ini. Berisi tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metode penelitian, dan sistematika penulisan.
2. Bab 2 : Dasar Teori  
Membahas mengenai teori-teori yang mendukung berjalannya penelitian ini. Berisi tentang *WebSockets*, *Socket.io*, *NodeJs*, dan *HTML5Canvas*.
3. Bab 3 : Analisis  
Membahas mengenai analisa masalah.
4. Bab 4 : Perancangan  
Membahas mengenai perancangan yang dilakukan sebelum melakukan tahapan implementasi.
5. Bab 5 : Implementasi dan Pengujian  
Membahas mengenai implementasi dan pengujian yang telah dilakukan.
6. Bab 6 : Kesimpulan dan Saran  
Membahas hasil kesimpulan dari keseluruhan penelitian ini dan saran-saran yang dapat diberikan untuk penelitian berikutnya.



## BAB 2

### LANDASAN TEORI

Pada bab ini akan dijelaskan landasan teori mengenai *WebSockets API*, *Socket.io*, *Node.js*, *Express.js*, dan *Canvas API*.

## 2.1 WebSockets

### 2.1.1 Interface

- *Websocket*

### 2.1.2 WebSocket Events

*WebSockets API* bekerja berdasarkan *events*(kejadian). Kode-kode yang ada pada aplikasi akan memperhatikan suatu *events* pada objek *WebSocket* untuk mengatasi apabila ada data-data yang masuk dan perubahan pada status koneksi. Aplikasi pada *client* tidak perlu melakukan *poll* terhadap *server* untuk memperbarui data. *Events* dan pesan-pesan lainnya akan diterima secara bersamaan saat server mengirimnya.

Sebuah objek *WebSocket* dapat mengirimkan empat *events* yang berbeda, yaitu :

- Open
- Message
- Error
- Close

### 2.1.3 WebSocket Methods

Suatu objek *WebSocket* memiliki dua *method*, yaitu *send()* dan *close()*.

- *send()*  
Setelah aplikasi melakukan koneksi menggunakan *WebSocket*, maka aplikasi tersebut dapat mengirimkan metode *send()* selama koneksi tetap terbuka. Sebuah aplikasi akan menggunakan *send()* untuk mengirimkan pesan dari *client* menuju *server*.
- *close()*

## 2.2 Socket.io

*Socket.io* merupakan *JavaScript library* yang digunakan pada aplikasi web untuk melakukan koneksi secara *realtime*. Teknologi ini memiliki dua bagian *library*: bagian *client* yang dijalankan pada *web browser*, dan bagian *server* yang dijalankan untuk *Node.js*. *Socket.io* memiliki fitur-fitur yang beragam, seperti melakukan *broadcast* ke beberapa *sockets*, dan menyimpan data yang berhubungan dengan masing-masing *client*.

### 2.2.1 Connection

Koneksi yang dimulai oleh *Socket.io* dilakukan dengan cara *handshake*. Hal tersebut merupakan bagian yang penting dalam protokol. *Handshake* hanya dilakukan hanya pada saat memulai koneksi, pesan-pesan atau hal lain dalam protokol akan dikirimkan melalui *socket*.

### 2.2.2 Messages

Apabila koneksi telah dilakukan, seluruh komunikasi antara *client* dan *server* akan menggunakan pesan(*message*) melalui *socket*. Pesan yang akan dikirimkan harus diubah kedalam format yang sudah dispesifikasi oleh *socket.io*.

Format yang sudah dispesifikasi oleh *socket.io* bertujuan untuk menentukan jenis pesan dan data yang dikirimkan dalam pesan tersebut. Format pesan yang sudah dispesifikasi yaitu seperti berikut :

[type] : [id] : [endpoint] (: [data])

- *type*, merupakan satu digit angka integer yang menunjukkan jenis pesan yang akan dikirim.
- *id*, merupakan identitas pesan yang terdiri dari beberapa digit angka.
- *endpoint*, merupakan *socket* tujuan yang akan menerima pesan yang sedang dikirim. Apabila tidak ada *endpoint*, maka pesan akan dikirimkan ke *default socket*.
- *data*, merupakan data yang akan dikirim ke *socket* tertentu. Pada kasus *messages*, data akan dikirimkan dalam bentuk *plain text*, sementara pada kasus *events* akan dikirimkan dalam bentuk *JSON*.

## 2.3 Node.js

## 2.4 Express.js

Express.js merupakan sebuah *framework* aplikasi web untuk *Node.js*. Express.js menyediakan fitur-fitur yang membuat pengembangan aplikasi web dapat bertahan lama. Teknologi ini pun merupakan modul *node package manager (npm)* yang menjadi ketergantungan dalam suatu aplikasi. Agar dapat berjalan, seluruh *file* yang dimiliki oleh *framework* ini harus berada pada *node\_modules* lokal dalam suatu projek tertentu.

## 2.5 Canvas API

Canvas API merupakan salah satu elemen *HTML5* yang digunakan untuk membuat gambar grafis dalam aplikasi web. Teknologi ini memiliki fitur untuk membuat komposisi foto, membuat animasi, dan membuat *real-time video processing* atau *rendering*. Untuk dapat menggunakan elemen *canvas* harus menambahkan *tag <canvas>* pada suatu halaman *HTML*. *Tag <canvas>* memiliki tiga atribut utama dimana atribut tersebut terdapat didalam kurung lancip pada *HTML tag*. Atribut-atribut tersebut yaitu :

- *id*  
Merupakan nama yang akan digunakan sebagai referensi dalam kode JavaScript. Dimana nantinya nama tersebut akan merujuk ke *tag <canvas>* yang memiliki nama yang sama.
- *width*  
Merupakan lebar dari *canvas* yang dibuat.
- *height*  
Merupakan tinggi dari *canvas* yang dibuat.

### 2.5.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh  $\text{\LaTeX}$ , 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

|          | $v_{start}$ | $\mathcal{S}_1$ | $v_{end}$ |
|----------|-------------|-----------------|-----------|
| $\tau_1$ | 1           | 12              | 20        |
| $\tau_2$ | 1           |                 | 20        |
| $\tau_3$ | 1           | 9               | 20        |
| $\tau_4$ | 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)

|          | $v_{start}$ | $\mathcal{S}_2$ | $\mathcal{S}_1$ | $v_{end}$ |
|----------|-------------|-----------------|-----------------|-----------|
| $\tau_1$ | 1           | 5               | 12              | 20        |
| $\tau_2$ | 1           | 8               |                 | 20        |
| $\tau_3$ | 1           | 2/8/17          | 9               | 20        |
| $\tau_4$ | 1           |                 |                 | 20        |

Tabel 2.3: Tabel bewarna(2)

|          | $v_{start}$ | $\mathcal{S}_1$ | $\mathcal{S}_2$ | $v_{end}$ |
|----------|-------------|-----------------|-----------------|-----------|
| $\tau_1$ | 1           | 12              | 5               | 20        |
| $\tau_2$ | 1           |                 | 8               | 20        |
| $\tau_3$ | 1           | 9               | 2/8/17          | 20        |
| $\tau_4$ | 1           |                 |                 | 20        |

### 2.5.2 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: [?]
- Bab dalam buku: [?]
- Artikel dari Jurnal: [?]
- Artikel dari prosiding seminar/konferensi: [?]
- Skripsi/Thesis/Disertasi: [?] [?] [?]
- Technical/Scientific Report: [?]
- RFC (Request For Comments): [?]
- Technical Documentation/Technical Manual: [?] [?] [?]
- Paten: [?]
- Tidak dipublikasikan: [?] [?]
- Laman web: [?]
- Lain-lain: [?]

### 2.5.3 Gambar

Pada hampir semua editor, penempatan gambar di dalam dokumen L<sup>A</sup>T<sub>E</sub>X 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<sup>A</sup>T<sub>E</sub>X 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.

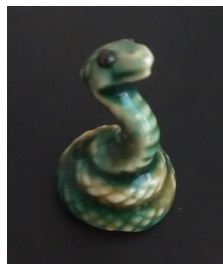
Nulla non mauris vitae wisi posuere convallis. Sed eu nulla nec eros scelerisque pharetra. Nullam varius. Etiam dignissim elementum metus. Vestibulum faucibus, metus sit amet mattis rhoncus, sapien dui laoreet odio, nec ultricies nibh augue a enim. Fusce in ligula. Quisque at magna et nulla commodo consequat. 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

at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque dapibus nibh. Nam enim. Lorem ipsum dolor sit amet, consectetur 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, consectetur 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



# LAMPIRAN A

## KODE PROGRAM

Listing 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

```

Listing 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