

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Wireless Sensor Network (WSN) adalah suatu jaringan nirkabel yang terdiri dari kumpulan *sensor node* dengan kemampuan deteksi (*sensing*), komputasi, dan komunikasi yang tersebar pada suatu tempat. Setiap sensor akan mengumpulkan data dari area yang dideteksi seperti temperatur, suara, getaran, tekanan, gerakan, kelembaban udara dan deteksi lainnya tergantung kemampuan sensor tersebut. Data yang diterima ini kemudian akan diteruskan ke *base station* untuk diolah sehingga memberikan suatu informasi. *Wireless Sensor Network* dapat diimplementasikan pada berbagai bidang kehidupan manusia diantaranya bidang militer untuk deteksi musuh, bidang pertanian untuk pemantauan pertumbuhan tanaman, bidang kesehatan, deteksi bahaya dan bencana alam, bidang pembangunan dan tata kota, dan bidang pendidikan.

Terdapat dua macam arsitektur *Wireless Sensor Network*, yaitu kluster dan flat. Pada arsitektur kluster, *node-node* akan disusun secara hierarki dan memiliki peran sebagai *cluster head*, *child node*, atau *parent node*. *Cluster head* berfungsi sebagai pengatur beberapa *child node*. Sedangkan pada arsitektur flat hanya terdapat dua macam *node* secara fungsional, yaitu *source node* dan *sink node*. Semua *sensor node* (*source node*) akan mengirim data ke satu tujuan akhir yaitu *sink node*.

Dalam praktiknya pengiriman data merupakan suatu hal yang penting pada *Wireless Sensor Network*. Data yang didapat dari sensor harus dapat sampai ke *base station* dengan utuh dan akurat (*reliable*). Data yang *reliable* ini sangat penting karena hasil pengukuran dan tindakan selanjutnya yang akan diambil akan bergantung pada data-data tersebut. Terdapat beberapa cara untuk memastikan transfer data *reliable* yaitu dengan *Link-Layer Retransmission*, *End-to-End Retransmission*, dan *Erasure Code*.

Pada skripsi ini dibangun aplikasi untuk transfer data pada *Wireless Sensor Network*. *Wireless Sensor Network* yang dibuat juga dapat melakukan transfer data ke *sensor node* tetangganya hingga sampai ke *sensor node* yang berperan sebagai *base station*. Karena *sensor node* memiliki kapasitas penyimpanan yang kecil dan data yang *reliable* sangat dibutuhkan untuk menentukan tindakan selanjutnya, maka akan dibangun juga *Wireless Sensor Network* yang memiliki sifat *reliable* tersebut.

1.2 Rumusan Masalah

- Bagaimana cara membangun aplikasi transfer data dari setiap *wireless sensor node* pada *Wireless Sensor Network*?
- Bagaimana cara membangun aplikasi transfer data yang *reliable* pada *Wireless Sensor Network*?

1.3 Tujuan

- Membangun aplikasi transfer data yang *reliable* pada *Wireless Sensor Network*.

1.4 Batasan Masalah

Penelitian ini dibuat berdasarkan batasan-batasan sebagai berikut:

1. Sensor yang digunakan sebagai penelitian hanya sensor untuk mengukur temperatur, kelembapan, getaran, dan tekanan udara.
2. Arsitektur yang digunakan untuk membangun *Wireless Sensor Network* ini adalah flat dan kluster.

1.5 Metodologi

Berikut adalah metode penelitian yang digunakan dalam penelitian ini:

1. Melakukan studi literatur mengenai *Wireless Sensor Network*.
2. Mempelajari protokol transfer data yang biasa pada *Wireless Sensor Network*.
3. Mempelajari prinsip *Reliable Data Transfer* pada *Wireless Sensor Network*.
4. Mempelajari pemrograman pada *Wireless Sensor Network* dengan Bahasa Pemrograman JAVA.
5. Melakukan perancangan perangkat lunak.
6. Mengimplementasi rancangan perangkat lunak pada *Wireless Sensor Network*.

1.6 Sistematika Pembahasan

Setiap bab dalam penelitian ini memiliki sistematika penulisan yang dijelaskan ke dalam poin-poin sebagai berikut:

1. Bab 1: Pendahuluan yaitu 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, yaitu membahas teori-teori yang mendukung berjalannya penelitian ini. Berisi tentang *Wireless Sensor Network*,
3. Bab 3: Analisis,
4. Bab 4: Perancangan,
5. Bab 5: Implementasi dan Pengujian,
6. Bab 6: Kesimpulan,

BAB 2

LANDASAN TEORI

Pada bab ini dijelaskan dasar-dasar teori mengenai *Wireless Sensor Network*, transfer data pada wireless sensor network, prinsip *reliable data transfer* pada Wireless Sensor Network, dan pengembangan pemrograman pada WSN.

2.1 Wireless Sensor Network

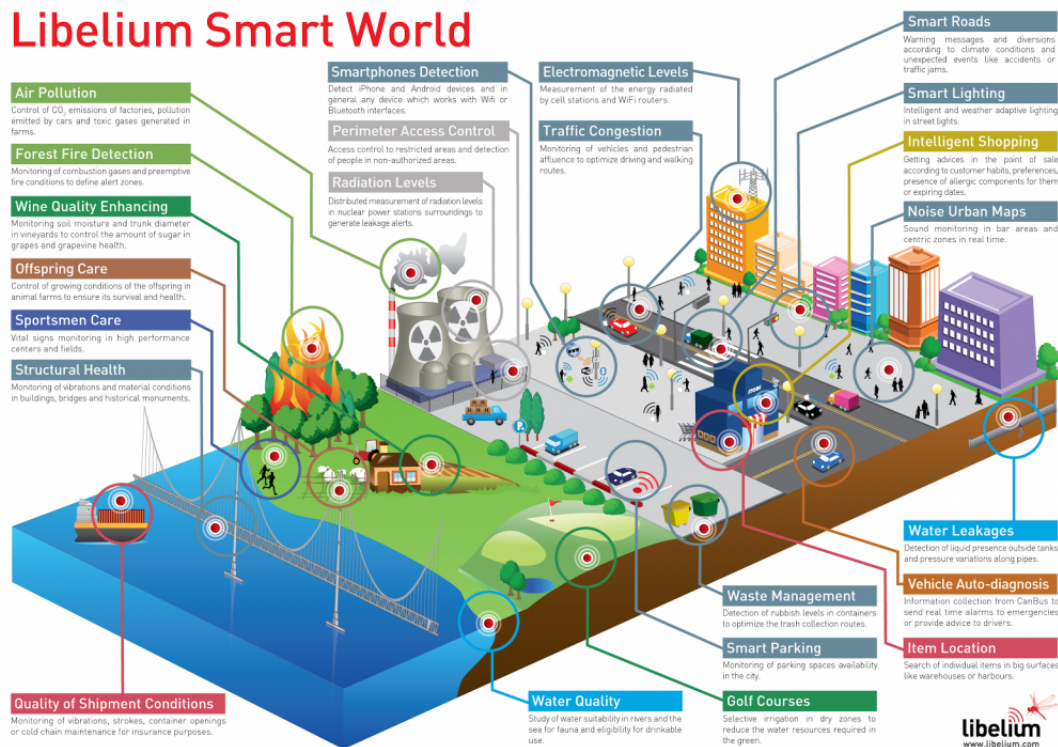
Wireless Sensor Network merupakan jaringan nirkabel yang terdiri dari sekumpulan *sensor node* yang diletakan pada suatu tempat dan memiliki kemampuan untuk mengukur kondisi lingkungan sekitar (*sensing*), komputasi dan dilengkapi dengan alat komunikasi *wireless* untuk komunikasi antar *sensor node*. Sensor ini akan mengumpulkan data dari kondisi lingkungannya, seperti: cahaya, suara, kelembapan, getaran, gerakan, temperatur, tekanan udara, kualitas air, komposisi tanah, dan lain-lain. Data ini kemudian dikirimkan ke *sensor node* tetangganya hingga sampai ke *base station* sebagai pusat untuk dikelola.

2.1.1 Penerapan Wireless Sensor Network

Pada awalnya *sensor network* (jaringan sensor) digunakan dalam teknologi militer untuk mendeteksi musuh di laut dan di darat. Semakin lama sensor node ini banyak dikembangkan untuk membantu berbagai bidang kehidupan manusia. Pemanfaatan Wireless Sensor Network pada kehidupan manusia dapat dilihat pada ilustrasi Gambar 2.1. Berikut adalah beberapa penerapan wireless sensor network pada berbagai bidang kehidupan manusia:

- Bidang militer
Wireless Sensor Network digunakan sebagai bagian dari komunikasi pada bidang militer dan deteksi target atau musuh.
- Monitoring area
Pada monitoring area, sensor node akan disebar pada suatu tempat yang akan di monitoring. Saat sensor node mendeteksi suatu kejadian (panas, tekanan, dan lain-lain) pada tempat, data akan dikirimkan ke base station untuk ditentukan tindakan selanjutnya.
- Transportasi
Pada bidang transportasi wireless sensor network digunakan untuk mendeteksi lalu lintas secara aktual yang nantinya akan disampaikan kepada pengendara tentang kejadian di depan mereka seperti kemacetan lalu lintas.
- Kesehatan
Beberapa aplikasi kesehatan seperti membantu pada disabilitas, monitoring pasien, diagnosis, pengaturan penggunaan obat, dan pelacakan dokter dan pasien di rumah sakit.
- Deteksi lingkungan
Deteksi lingkungan yang dapat dilakukan antara lain deteksi gunung berapi, polusi udara, kebakaran hutan, efek rumah kaca, dan deteksi longsor.

- Monitoring struktur
Wireless Sensor Network dapat melakukan deteksi pergerakan bangunan dan infrastruktur seperti jembatan, flyover, terowongan dan fasilitas lain tanpa mengeluarkan biaya untuk melakukan dektesi manual dengan mendatangi tempatnya secara langsung.
- Bidang pertanian
Pada bidang pertanian dapat membantu pengelola pertanian untuk mengelola penggunaan air dalam irigasi dan menghasilkan lebih sedikit hasil buangan dari pertanian mereka.



Gambar 2.1: Pemanfaatan *Wireless Sensor Network* diberbagai bidang kehidupan manusia

2.1.2 Struktur Wireless Sensor Network

2.1.3 Arsitektur Wireless Sensor Network

Pada *Wireless Sensor Network* biasanya akan terdapat banyak *sensor node* yang disebar pada suatu tempat atau daerah, dan terdapat satu atau lebih *sink node* atau *base station* dalam area sensing tersebut (Gambar 2.7). Dalam membuat *Wireless Sensor Network* perlu diperhatikan arsitektur yang akan digunakan. Arsitektur jaringan yang biasanya dipakai pada *Wireless Sensor Network* adalah **arsitektur flat** dan **arsitektur hierarki**. Selain itu dalam membangun *Wireless Sensor Network* perlu juga diperhatikan jalur lintasan yang menghubungkan antar *sensor node* saat transfer data. Untuk area *sensing* yang tidak terlalu luas dan hanya menggunakan sedikit *sensor node* dapat menggunakan protokol routing *single hop*. Sedangkan untuk daerah yang luas dan memerlukan banyak *sensor node* dapat menggunakan protokol routing *multi hop*.

Single-Hop dan Multi-Hop

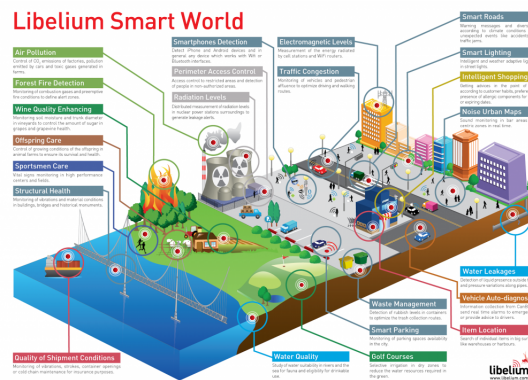
Untuk mengirim data ke sink node setiap sensor node dapat menggunakan single-hop long-distance transmission. Single-hop long-distance ini berarti setiap sensor node akan mengirimkan data ke sink node hanya satu kali lompatan walaupun jarak antara sink node dengan sensor node itu sangat

jauh. Dalam jaringan sensor, penggunaan daya paling besar adalah saat melakukan komunikasi dibandingkan saat sensing. Penggunaan daya akan semakin bertambah jika jarak sink dan sensor node semakin jauh. Untuk menangani masalah tersebut muncul protokol multi-hop.

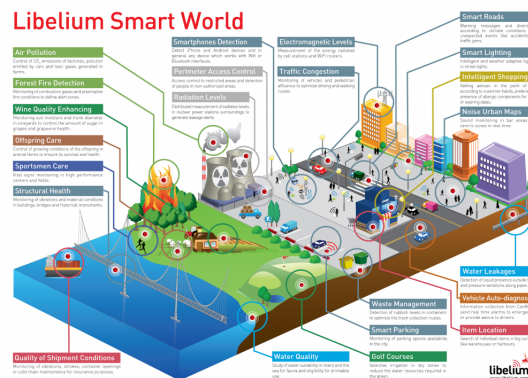
Pada protokol multi-hop sensor node akan disusun saling berdekatan dan terhubung dengan yang lain. Jadi saat akan berkomunikasi dengan sink node, sensor node harus mengirimkan data tersebut ke sensor node tetangganya dan diteruskan hingga sampai ke sink node. Karena jarak yang saling berdekatan maka penggunaan daya dapat efektif.

Arsitektur Flat

Pada jaringan flat, setiap *sensor node* memiliki peran / *role* yang sama dalam melakukan *sensing*. Secara fungsional hanya terdapat dua macam *sensor node* pada jaringan flat, yaitu *source node* dan *sink node*. Karena jumlah *sensor node* yang banyak maka tidak mungkin menentukan *global identified* untuk setiap *sensor node* pada jaringan ini. Untuk mendapatkan data dilakukan dengan cara *sink node* melakukan transmit ke semua *sensor node* pada area *sensing* dengan cara *flooding* dan hanya *sensor node* yang sesuai yang akan merespon *sink node*. Setiap *sensor node* mengirimkan data ke *sink node* dengan *multi hop* dan melalui node tetangganya yang terhubung dengannya untuk meneruskan data (Gambar 2.7). Arsitektur flat juga dapat dilakukan dengan *single hop* seperti pada Gambar 2.7.



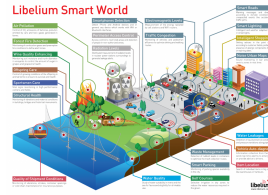
Gambar 2.2: Ilustrasi arsitektur flat pada *Wireless Sensor Network* dengan *multi hop*



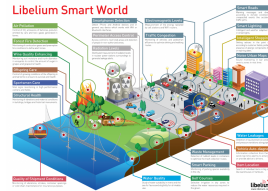
Gambar 2.3: Ilustrasi arsitektur flat pada *Wireless Sensor Network* dengan *single hop*

Arsitektur Hierarki

Pada arsitektur hierarki, semua sensor node dikelompokkan ke dalam cluster-cluster. Terdapat cluster head pada setiap cluster. Cluster head ini yang mengumpulkan data dari setiap sensor node



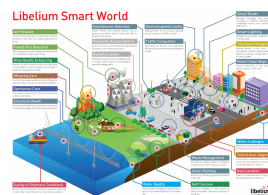
Gambar 2.4: Ilustrasi arsitektur flat pada *Wireless Sensor Network* dengan *multi hop*



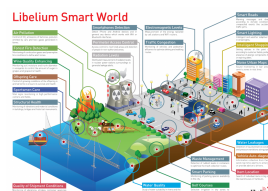
Gambar 2.5: Ilustrasi arsitektur flat pada *Wireless Sensor Network* dengan *single hop*

di bawahnya dan meneruskan data yang telah diterima ke base station atau sink. Hal yang perlu diperhatikan pada arsitektur hierarki adalah pemilihan sensor node sebagai cluster head dan sensor node yang melakukan sensing. Penggunaan daya yang paling besar dalam *Wireless Sensor Network* ini adalah saat melakukan komunikasi yaitu saat mengirimkan data ke sensor node lain. Maka untuk sensor node yang memiliki daya kecil dapat digunakan untuk sensing, karena sensor node sensing ini hanya melakukan komunikasi ke cluster head. Cluster head harus memiliki daya yang lebih banyak, karena cluster head akan bertugas menerima hasil sensing sensor node di bawahnya dan meneruskan data ke sink node.

Masalah yang utama pada clustering ini adalah pemilihan cluster head dan bagaimana cara mengatur setiap cluster. Terdapat beberapa cara untuk membuat clustering ini. Berdasarkan jarak antara cluster head dengan cluster member, dapat dibuat clustering dengan single hop atau multi hop seperti pada Gambar 2.7 dan Gambar 2.7. Sedangkan jika berdasarkan jumlah tier dapat dibangun clustering single tier atau multi tier seperti pada Gambar 2.7 dan Gambar 2.7.



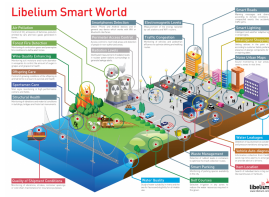
Gambar 2.6: Ilustrasi clustering dengan single tier



Gambar 2.7: Ilustrasi clustering dengan multi tier

2.1.4 Layer pada *Wireless Sensor Network*

Wireless Sensor Network memiliki lima layer protokol: physical layer, data link layer, network layer, transport layer, dan application layer, seperti pada Gambar 2.7.



Gambar 2.8: Layer pada *Wireless Sensor Network*

2.2 Data Transfer pada Wireless Sensor Network

asdf

2.3 Prinsip Reliable Data Transfer pada Wireless Sensor Network

2.4 Pengembangan Pemrograman pada Wireless Sensor Network

2.4.1 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, 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

2.4.2 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: [?]

2.4.3 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 2.8), tipe JPG (Gambar 2.9) dan tipe PDF (Gambar 2.10)
- 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 2.8 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
 - Gambar 2.9 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
 - Gambar 2.10 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 2.9: Gambar *Serpentes* dalam format png

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



Gambar 2.10: Ular kecil

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

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



Gambar 2.11: *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