

# SKRIPSI

«JUDUL BAHASA INDONESIA»



Prayogo Cendra

NPM: 2014730033

PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNOLOGI INFORMASI DAN SAINS  
UNIVERSITAS KATOLIK PARAHYANGAN

«tahun»



# UNDERGRADUATE THESIS

«JUDUL BAHASA INGGRIS»



Prayogo Cendra

NPM: 2014730033

DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY AND SCIENCES  
PARAHYANGAN CATHOLIC UNIVERSITY

«tahun»



# LEMBAR PENGESAHAN

«JUDUL BAHASA INDONESIA»

Prayogo Cendra

NPM: 2014730033

Bandung, «tanggal» «bulan» «tahun»

Menyetujui,

Pembimbing Utama

Pembimbing Pendamping

Claudio Franciscus, M.T.

«pembimbing pendamping/2»

Ketua Tim Penguji

Anggota Tim Penguji

«penguji 1»

«penguji 2»

Mengetahui,

Ketua Program Studi

Mariskha Tri Adithia, P.D.Eng



## PERNYATAAN

Dengan ini saya yang bertandatangan di bawah ini menyatakan bahwa skripsi 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 Rp. 6000
---------------------

Prayogo Cendra  
NPM: 2014730033





## 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 . . . . .	2
<b>2 LANDASAN TEORI</b>	<b>5</b>
2.1 Linear Programming . . . . .	5
2.1.1 Karakteristik . . . . .	5
2.1.2 Daerah Solusi dan Solusi Optimal . . . . .	6
2.1.3 Bentuk Standar . . . . .	8
2.1.4 Variabel Basis dan Non Basis . . . . .	10
2.1.5 Metode Simplex . . . . .	10
2.2 Template Skripsi FTIS UNPAR . . . . .	13
2.2.1 Tabel . . . . .	13
2.2.2 Kutipan . . . . .	14
2.2.3 Gambar . . . . .	14
<b>A KODE PROGRAM</b>	<b>17</b>
<b>B HASIL EKSPERIMEN</b>	<b>19</b>



## DAFTAR GAMBAR

2.1	Titik-titik sudut dalam daerah solusi . . . . .	7
2.2	Daerah solusi yang dibatasi . . . . .	7
2.3	Daerah solusi yang tidak dibatasi . . . . .	8
2.4	Tidak memiliki daerah solusi . . . . .	8
2.5	Gambar <i>Serpentes</i> dalam format png . . . . .	15
2.6	Ular kecil . . . . .	16
2.7	<i>Serpentes</i> betina . . . . .	16
B.1	Hasil 1 . . . . .	19
B.2	Hasil 2 . . . . .	19
B.3	Hasil 3 . . . . .	19
B.4	Hasil 4 . . . . .	19



## DAFTAR TABEL

2.1	Kerangka awal tabel simplex . . . . .	11
2.2	Tabel simplex pada iterasi ke-1 . . . . .	11
2.3	Proses <i>pivoting</i> . . . . .	12
2.4	Proses pembaharuan baris basic . . . . .	12
2.5	Tabel simplex iterasi ke-2 . . . . .	12
2.6	Tabel simplex iterasi ke-3 . . . . .	12
2.7	Tabel simplex iterasi ke-4 . . . . .	13
2.8	Tabel contoh . . . . .	13
2.9	Tabel bewarna(1) . . . . .	14
2.10	Tabel bewarna(2) . . . . .	14



# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Kamera merupakan alat/komponen optik yang digunakan untuk mengambil citra/gambar. Salah satu penggunaan kamera dalam kehidupan sehari-hari adalah kamera CCTV (*closed-circuit television*). Kamera CCTV digunakan untuk memantau suatu lokasi dengan tujuan pengawasan dan keamanan. Kamera CCTV pada umumnya dipasang pada tempat strategis sehingga memiliki tingkat jangkauan yang baik. Kamera CCTV bekerja dengan cara merekam lokasi dalam jangkauannya secara terus menerus dan menyimpan hasil rekamannya dalam media penyimpanan. Rekaman ini biasanya digunakan oleh petugas keamanan untuk memantau lokasi tersebut dari tempat yang berbeda sehingga petugas tidak perlu memantau lokasi tersebut dengan datang secara langsung. Petugas hanya perlu mendatangi lokasi tersebut apabila mendapati hal-hal yang mencurigakan berdasarkan hasil rekaman CCTV.

Penempatan kamera CCTV di ruangan yang berbentuk sederhana (persegi panjang) tidaklah sulit. Kamera CCTV yang dibutuhkan biasanya berjumlah dua buah dan dipasang di kedua sudut ruangan yang merupakan satu diagonal. Namun, jika ruangan berukuran besar, maka tujuan penggunaan kamera CCTV pun tidak hanya untuk mendeteksi adanya orang, tetapi juga mengenali orang tersebut. Hal ini tentunya menyebabkan kesulitan dalam menentukan jumlah minimum dan lokasi penempatan kamera CCTV. Terdapat beberapa pendekatan yang dapat digunakan untuk menyelesaikan masalah ini, seperti dengan cara memasang kamera CCTV pada daerah-daerah yang dapat dimasuki orang. Tetapi pada kasus terburuk, orang bisa saja masuk melewati jalur-jalur yang tidak diduga, seperti tembok, atap, bawah tanah, dsb. Oleh karena itu, alangkah baiknya pemasangan kamera CCTV dilakukan hingga seluruh daerah di ruangan tersebut tercakup sepenuhnya.

Penempatan kamera CCTV dapat dilakukan dalam berbagai lokasi dan berbagai arah pandang. Apabila penempatan kamera CCTV dilakukan tanpa adanya perhitungan, maka terdapat kemungkinan jumlah kamera yang terlalu banyak dan/atau seluruh lokasi yang tidak tercakup sepenuhnya. Pada penempatan kamera CCTV terdapat perhitungan tingkat *overlap* (penumpukan jangkauan) dan tingkat *out of bound* (jarak pandang terpotong). Penempatan kamera CCTV akan semakin baik apabila memiliki tingkat *overlap* dan tingkat *out of bound* yang semakin rendah.

Pada skripsi ini, akan dibuat sebuah perangkat lunak yang akan menghasilkan jumlah minimum beserta dengan penempatan-penempatan kamera CCTV berdasarkan ukuran ruangan, jarak pandang efektif kamera CCTV, dan sudut pandang kamera CCTV. Penempatan kamera CCTV akan terdiri atas lokasi penempatan dan juga arah pandangnya. Perangkat lunak juga akan memberikan visualisasi guna membantu pengguna memahami penempatan setiap kamera CCTV pada ruangan yang dimasukkan. Hasil dari perangkat lunak ini dapat dipastikan menjadi hasil yang paling optimal yang berarti bahwa hasil akan memiliki jumlah kamera CCTV yang paling minimum yang mencakup seluruh isi ruangan.

## 1.2 Rumusan Masalah

Berdasarkan deskripsi masalah yang dibahas sebelumnya, maka ditetapkan rumusan masalah sebagai berikut:

- Bagaimana cara menentukan jumlah minimum kamera CCTV dalam suatu ruangan?
- Bagaimana cara memvisualisasikan penempatan kamera-kamera CCTV dalam suatu ruangan?

## 1.3 Tujuan

Berdasarkan rumusan masalah, maka tujuan dalam skripsi ini adalah sebagai berikut:

- Mempelajari cara menentukan jumlah minimum kamera CCTV dalam suatu ruangan.
- Membangun perangkat lunak yang dapat memvisualisasikan penempatan kamera-kamera CCTV dalam suatu ruangan.

## 1.4 Batasan Masalah

Untuk mempermudah pembuatan template ini, tentu ada hal-hal yang harus dibatasi, misalnya saja bahwa template ini bukan berupa style  $\text{\LaTeX}$  pada umumnya (dengan alasannya karena belum mampu jika diminta membuat seperti itu)

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

## 1.5 Metodologi

Tentunya akan diisi dengan metodologi yang serius sehingga templatennya terkesan lebih serius.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

## 1.6 Sistematika Pembahasan

Rencananya Bab 2 akan berisi petunjuk penggunaan template dan dasar-dasar  $\text{\LaTeX}$ . 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.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales



---

eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetur odio sem sed wisi.



## BAB 2

### LANDASAN TEORI

#### 2.1 Linear Programming

Linear programming adalah teknik dalam ilmu matematika untuk menyelesaikan masalah yang berhubungan dengan optimasi. Teknik ini biasa digunakan dalam bidang produksi, manajemen operasional, penjadwalan, dan bidang-bidang lainnya yang melibatkan pengambilan keputusan. Pada bidang produksi, terdapat masalah ketika menentukan produk yang akan dihasilkan. Biasanya terdapat beberapa faktor yang menjadi pertimbangan seperti jumlah ketersediaan sumber daya, jumlah pekerja, dan besar keuntungan yang didapatkan. Dengan adanya faktor-faktor tersebut, penentuan terhadap produk yang akan dihasilkan tentunya harus dilakukan dengan perhitungan yang tepat. Linear programming merupakan teknik yang cocok untuk menyelesaikan masalah tersebut. Dengan memodelkan masalah tersebut secara tepat, maka masalah tersebut dapat diselesaikan sehingga menghasilkan solusi terbaik tanpa mengabaikan setiap faktor yang ada.

##### 2.1.1 Karakteristik

Setiap masalah optimasi dapat dimodelkan ke bentuk linear programming yang memiliki ciri-ciri sebagai berikut:

- Variabel keputusan

Variabel keputusan merupakan variabel yang akan dicari nilainya. Variabel ini juga menunjukkan keputusan yang akan diambil.

$x_1$  = jumlah produk A yang diproduksi

$x_2$  = jumlah produk B yang diproduksi

- Fungsi Objektif

Dalam masalah linear programming, terdapat tujuan yang ingin dimaksimalkan atau diminimalkan. Tujuan atau hasil ini ditentukan oleh variabel-variabel keputusan. Dalam kasus produksi, pembuat keputusan menginginkan jumlah pendapatan terbesar. Bila pendapatan dihitung berdasarkan jumlah dan harga dari setiap produk, maka fungsi objektif akan memaksimalkan setiap pendapatan dari masing-masing produk.

$$\text{maximize } z = 3x_1 + 2x_2$$

- Batasan

Fungsi objektif ditentukan oleh variabel-variabel keputusan. Dalam fungsi objektif yang memaksimalkan hasil, nilai dari setiap variabel keputusan dapat ditentukan sedemikian mungkin sehingga menghasilkan hasil sebesar-besarnya. Tentunya hal ini tidak mungkin terjadi karena terdapat faktor-faktor yang mencegah variabel keputusan untuk ditentukan

sedemikian mungkin. Faktor-faktor ini berperan sebagai batasan bagi variabel keputusan sehingga nilai variabel keputusan tidak melebihi batas. Dengan adanya batasan ini, nilai dari fungsi objektif tidak akan bernilai tak hingga karena variabel-variabel keputusannya dibatasi.

$$2x_1 + x_2 \leq 100$$

$$x_1 + x_2 \leq 80$$

$$x_1 \leq 40$$

- Tanda Pembatas

Jika variabel keputusan  $x_i$  tidak dapat bernilai negatif maka perlu ditambahkan tanda pembatas  $x_i \geq 0$ . Namun, apabila nilai variabel keputusan tidak dibatasi atau dapat bernilai negatif, maka variabel keputusan tersebut merupakan variabel yang tidak dibatasi oleh tanda (*unrestricted in sign*). Dalam kasus produksi, jumlah barang yang diproduksi tidak dapat bernilai negatif sehingga variabel jumlah barang harus diberikan tanda pembatas.

$$x_1 \geq 0$$

$$x_2 \geq 0$$

Dengan adanya keempat karakteristik di atas, setiap masalah optimasi dapat dimodelkan ke bentuk linear programming. Berikut contoh pemodelan masalah dalam bentuk linear programming:

$$\begin{aligned} &\text{maximize } z = 3x_1 + 2x_2 \\ &\text{subject to } 2x_1 + x_2 \leq 100 \\ &\quad x_1 + x_2 \leq 80 \\ &\quad x_1 \leq 40 \\ &\quad x_1 \geq 0 \\ &\quad x_2 \geq 0 \end{aligned}$$

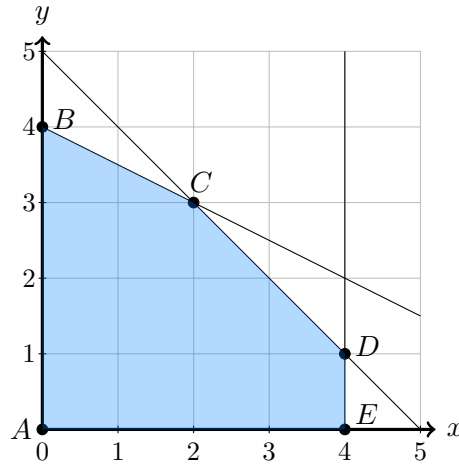
### 2.1.2 Daerah Solusi dan Solusi Optimal

Solusi dari linear programming merupakan solusi yang tidak melanggar setiap batasan-batasannya. Dengan diketahuinya hal ini, bisa saja maka linear programming memiliki banyak solusi dengan tidak melanggar batasan manapun. Dalam linear programming, solusi-solusi ini berada dalam daerah yang disebut dengan daerah solusi (feasible region). Daerah ini berbentuk *convex polytope* yang terdiri dari banyak titik sudut. Titik-titik sudut ini merupakan perpotongan dari setiap batasan dengan batasan-batasan lainnya. Dengan adanya daerah ini, pencarian solusi menjadi dibatasi sehingga tidak perlu memeriksa seluruh kemungkinan yang ada.

Bila diketahui bahwa titik sudut dibentuk dari perpotongan antar batasan, maka hal ini tidak berlaku sebaliknya. Titik potong tidak selalu menjadi titik sudut bagi daerah solusi. Titik sudut merupakan titik potong yang tidak melanggar batasan-batasan. Apabila daerah solusi divisualisasikan dalam bentuk grafik atau lainnya, daerah solusi akan berbentuk *convex polytope*.

Bila daerah solusi sudah ditemukan, solusi optimal dapat dicari dengan memeriksa setiap kemungkinan yang berada di dalam daerah solusi. Walaupun demikian, pemeriksaan ini masih dianggap tidak efisien. Bila solusi optimal dapat berupa bilangan pecahan, maka pemeriksaan tidak dapat berakhir karena bilangan pecahan yang berada di antara dua batas berjumlah tak hingga.

Tentunya seluruh pemeriksaan ini dapat dihindari karena solusi optimal dari linear programming berada pada salah satu titik sudut di daerah solusi. Dengan diketahuinya hal ini, maka solusi optimal dapat dicari dengan mengecek setiap titik sudut yang berjumlah terhingga (*finite*).

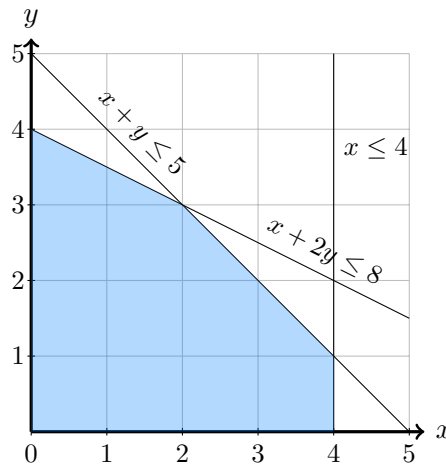


Gambar 2.1: Titik-titik sudut dalam daerah solusi

Dengan diketahuinya daerah solusi, maka terdapat 3 jenis solusi optimal dalam linear programming, yaitu:

1. Daerah solusi yang dibatasi (*bounded feasible region*)

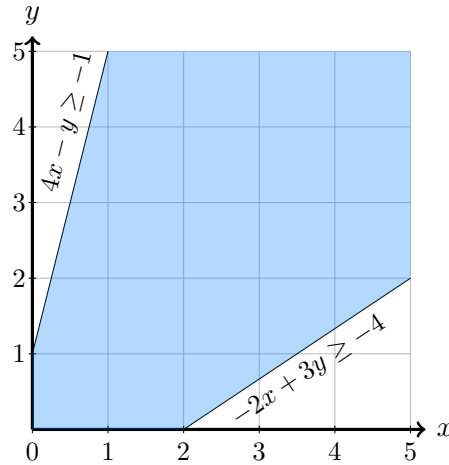
Batasan-batasan menghasilkan daerah solusi yang berbentuk *convex polytope*. Solusi optimal linear programming terdapat di salah satu titik sudut.



Gambar 2.2: Daerah solusi yang dibatasi

2. Daerah solusi yang tidak dibatasi (*unbounded feasible region*)

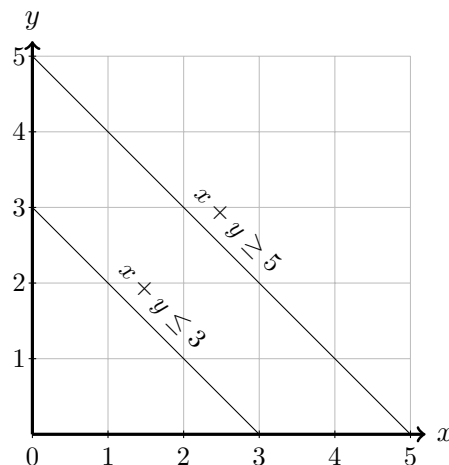
Batasan-batasan menghasilkan daerah solusi, namun daerah tersebut memiliki bagian yang tidak tertutup sehingga tidak berbentuk *convex polytope*. Hal ini dapat menyebabkan jumlah solusi yang tak terhingga. Dalam kasus ini, linear programming memiliki solusi namun solusi optimal tidak dapat ditemukan.



Gambar 2.3: Daerah solusi yang tidak dibatasi

### 3. Tidak memiliki daerah solusi (*infeasible region*)

Batasan-batasan tidak membentuk daerah solusi akibat adanya dua atau lebih persamaan batasan yang saling bertentangan. Pada kasus ini, linear programming tidak memiliki solusi dan tidak dapat diselesaikan.



Gambar 2.4: Tidak memiliki daerah solusi

### 2.1.3 Bentuk Standar

Setiap model linear programming dapat dimodelkan ke dalam bentuk standar berikut ini:

$$\begin{array}{ll}
 \text{maximize} & c_1x_1 + c_2x_2 + \dots + c_nx_n \\
 \text{subject to} & a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\
 & a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\
 & \vdots \\
 & a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m \\
 & x_1 \geq 0, x_2 \geq 0, \dots, x_n \geq 0
 \end{array}$$

Bentuk tersebut merupakan bentuk standar linear programming yang terdiri dari  $n$  buah variabel  $x$  dan  $m$  buah persamaan batasan. Variabel  $b_i, c_i, a_{ij}$  merupakan konstanta dan  $x_i$  merupakan variabel keputusan. Bentuk standar tersebut dapat dibentuk dalam notasi matriks berikut:

$$\begin{array}{ll} \text{maximize} & c^T x \\ \text{subject to} & Ax = b \text{ and } x \geq 0 \end{array}$$

Pada notasi matriks,  $x$  adalah matriks kolom berdimensi  $n$ ,  $C^T$  adalah matriks baris berdimensi  $n$ ,  $A$  adalah matriks berdimensi  $m \times n$ , dan  $b$  adalah matriks kolom berdimensi  $m$ . Matriks  $x \geq 0$  menunjukkan bahwa setiap variabel keputusan  $x_i$  tidak dapat bernilai negatif.

Langkah-langkah untuk mengubah *linear program* ke bentuk standar:

1. Ubah objektif menjadi maksimalisasi

Apabila objektif berupa minimalisasi, maka ubah objektif menjadi maksimalisasi dengan cara mengalikan fungsi objektif dengan -1. Contoh:

$$\text{minimize } z = x_1 + 2x_2$$

menjadi

$$\text{maximize } -z = -x_1 - 2x_2$$

2. Ubah pertidaksamaan pada batasan ke dalam bentuk persamaan

Perubahan pertidaksamaan pada batasan dibedakan menjadi 2 berdasarkan jenis tandanya, yaitu:

- (a) Pertidaksamaan lebih kecil ( $\leq$ )

Pertidaksamaan dengan tanda lebih kecil diubah ke bentuk persamaan dengan cara menambahkan variabel baru bernilai positif yang dinamakan *slack*. Selain itu ditambahkan juga tanda pembatas bagi variabel *slack* tersebut. Contoh:

$$x_1 + 2x_2 \leq 40$$

menjadi

$$\begin{aligned} x_1 + 2x_2 + s_1 &= 40 \\ s_1 &\geq 0 \end{aligned}$$

- (b) Pertidaksamaan lebih besar ( $\geq$ )

Pertidaksamaan dengan tanda lebih besar diubah ke bentuk persamaan dengan cara menambahkan variabel baru bernilai negatif yang dinamakan *surplus*. Selain itu ditambahkan juga tanda pembatas bagi variabel *surplus* tersebut. Contoh:

$$x_1 + 2x_2 \geq 40$$

menjadi

$$\begin{aligned} x_1 + 2x_2 - e_1 &= 40 \\ e_1 &\geq 0 \end{aligned}$$

### 2.1.4 Variabel Basis dan Non Basis

Jika model linear programming  $Ax = b$  terdiri dari  $m$  buah persamaan batasan dan  $n$  buah variabel, maka solusi basis dapat ditemukan dengan cara membuat  $n - m$  buah variabel bernilai 0 dan menyelesaikan  $m$  buah variabel tersisa lainnya. Sebanyak  $n - m$  buah variabel yang dibuat bernilai 0 merupakan variabel non basis. Sedangkan sebanyak  $m$  buah variabel sisanya merupakan variabel basis karena diselesaikan dengan membuat variabel lainnya menjadi variabel non basis. Berikut contoh sistem persamaan linear:

$$\begin{aligned}x_1 + x_2 &= 3 \\x_2 + x_3 &= -1\end{aligned}$$

Pada sistem persamaan linear di atas, akan dipilih sebanyak  $3 - 2 = 1$  (3 variabel dan 2 persamaan) buah variabel yang akan menjadi variabel non basis dalam. Jika himpunan variabel non basis  $NBV = \{x_3\}$ , maka himpunan variabel basis  $BV = \{x_1, x_2\}$ . Nilai dari variabel basis dapat dicari dengan membuat variabel  $x_3$  menjadi 0 dan menyelesaikan

$$\begin{aligned}x_1 + x_2 &= 3 \\-x_2 &= -1\end{aligned}$$

Didapatkan nilai  $x_1 = 2$  dan  $x_2 = 1$  sehingga  $x_1 = 2$ ,  $x_2 = 1$ ,  $x_3 = 0$  menjadi solusi basis. Solusi basis merupakan solusi yang didapatkan dengan menggunakan variabel basis.

### 2.1.5 Metode Simplex

Metode simplex merupakan salah satu metode yang digunakan untuk mencari solusi optimal dari masalah linear programming. Metode *simplex* bekerja secara beriterasi dan akan menghasilkan solusi yang mendekati optimal pada setiap iterasinya. Iterasi ini akan dilakukan hingga tidak ditemukannya solusi yang lebih optimal. Berikut contoh masalah linear programming:

$$\begin{aligned}&\text{maximize} && 3x_1 + 2x_2 \\&\text{subject to} && 2x_1 + x_2 \leq 18 \\&&& 2x_1 + 3x_2 \leq 42 \\&&& 3x_1 + x_2 \leq 24 \\&&& x_1 \geq 0, x_2 \geq 0\end{aligned}$$

Berikut langkah-langkah untuk menyelesaikan masalah linear programming dengan menggunakan metode simplex:

#### 1. Mengubah ke dalam bentuk standar

Pertidaksamaan-pertidaksamaan pada batasan diubah menjadi bentuk persamaan dengan menambahkan variabel *slack* dan mengubah tanda pertidaksamaan menjadi tanda persamaan.

$$\begin{aligned}&\text{maximize} && 3x_1 + 2x_2 + 0x_3 + 0x_4 + 0x_5 \\&\text{subject to} && 2x_1 + x_2 + x_3 = 18 \\&&& 2x_1 + 3x_2 + x_4 = 42 \\&&& 3x_1 + x_2 + x_5 = 24 \\&&& x_1 \geq 0, x_2 \geq 0, x_3 \geq 0, x_4 \geq 0, x_5 \geq 0\end{aligned}$$

#### 2. Membuat tabel *simplex*

Tabel *simplex* yang terdiri dari 4 kolom utama, yaitu kolom basis, kolom  $z$ , kolom variabel non basis, dan kolom ruas kanan/rhs (*right hand side*). Pada awalnya, baris  $z$  berisi nilai negatif dari konstanta variabel pada fungsi objektif dan  $m$  baris berikutnya berisi konstanta



variabel pada setiap persamaan batasan.  $m$  baris terkahir ini diidentifikasi sebagai baris basis. Berikut bentuk dari tabel simplex:

basic	$z$	$x_1$	$x_2$	$\dots$	$x_n$	rhs
$z$	1	$-c_1$	$-c_2$	$\dots$	$-c_n$	0
$x_{n+1}$	0	$a_{11}$	$a_{12}$	$\dots$	$a_{1n}$	$b_1$
$x_{n+2}$	0	$a_{21}$	$a_{22}$	$\dots$	$a_{2n}$	$b_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$x_{n+m}$	0	$a_{m1}$	$a_{m2}$	$\dots$	$a_{mn}$	$b_m$

Tabel 2.1: Kerangka awal tabel simplex

Berikut tabel simplex awal untuk contoh masalah linear programming di atas:

basic	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	rhs
$z$	1	-3	-2	0	0	0	0
$x_3$	0	2	1	1	0	0	18
$x_4$	0	2	3	0	1	0	42
$x_5$	0	3	1	0	0	1	24

Tabel 2.2: Tabel simplex pada iterasi ke-1

### 3. Mengecek solusi optimal

Apabila pada baris  $z$  tidak terdapat variabel dengan nilai negatif, maka solusi optimal sudah ditemukan dan menandakan akhir dari iterasi. Solusi optimal terdapat pada setiap rhs dari setiap baris basis. Jika masih terdapat variabel dengan nilai negatif, maka langkah selanjutnya adalah melakukan proses *pivoting*.

Pada tabel contoh *linear program* iterasi pertama, terdapat nilai negatif di baris  $z$ , yaitu pada variabel  $x_1$  dan  $x_2$ . Maka, akan dilanjutkan dengan proses *pivoting*.

### 4. Melakukan proses *pivoting*

Proses *pivoting* merupakan proses penukaran satu variabel basis dengan satu variabel non basis. Variabel basis yang akan digantikan (keluar dari baris basis) disebut dengan *leaving variable* dan variabel non basis yang akan menggantikan (masuk ke baris basis) disebut *entering variable*. Ketentuan dalam pemilihan variabel yang akan ditukar adalah sebagai berikut:

- *Entering variable* dipilih berdasarkan nilai terkecil dalam baris  $z$ . Kolom dari *entering variable* akan menjadi kolom pivot (*pivot column*).
- *Leaving variable* dipilih berdasarkan rasio terkecil rhs bersangkutan per nilai pada kolom pivot bersangkutan. Baris dari *leaving variable* akan menjadi baris pivot (*pivot row*).
- Pivot merupakan nilai yang berada di perpotongan kolom pivot dan baris pivot.

Berdasarkan contoh,  $x_1$  menjadi *entering variabel* karena bernilai paling kecil di baris  $z$ , yaitu  $-3$ . Sedangkan baris basis  $x_5$  menjadi *leaving variable* karena nilai rasionya yang paling kecil, yaitu 8. Perpotongan baris pivot dan kolom pivot menghasilkan pivot bernilai 3.

*pivot column*  
↓

basic	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	rhs	ratio:
$z$	1	-3	-2	0	0	0	0	$\frac{rhs_i}{pivotCol_i}$
$x_3$	0	2	1	1	0	0	18	$\frac{18}{2} = 9$
$x_4$	0	2	3	0	1	0	42	$\frac{42}{2} = 21$
$x_5$	0	(3)	1	0	0	1	24	$\frac{24}{3} = 8$

*pivot row* →

Tabel 2.3: Proses *pivoting*

## 5. Memperbaharui table simplex

Setiap nilai pada baris basis baru akan dibagi dengan nilai pivot sebelumnya. Sedangkan nilai pada baris basis lainnya dikurangi dengan hasil perkalian antara nilai kolom pivot yang bersangkutan dengan nilai baru pada baris pivot yang bersangkutan. Setelah diperbaharui, langkah selanjutnya adalah kembali pada pengecekan solusi optimal.

Pada contoh linear programming,  $x_1$  menjadi baris basis yang baru. Setiap nilai pada baris tersebut dibagi dengan nilai pivot sebelumnya yang bernilai 3. Setiap nilai pada baris basis lainnya diubah sesuai dengan ketentuan yang sudah dibahas sebelumnya.

basic	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	rhs
$z$	1	$-3 - (-3 \times \frac{3}{3})$	$-2 - (-3 \times \frac{1}{3})$	$0 - (-3 \times 0)$	$0 - (-3 \times 0)$	$0 - (-3 \times \frac{1}{3})$	$0 - (-3 \times \frac{24}{3})$
$x_3$	0	$2 - (2 \times \frac{3}{3})$	$1 - (2 \times \frac{1}{3})$	$1 - (2 \times 0)$	$0 - (2 \times 0)$	$0 - (2 \times \frac{1}{3})$	$18 - (2 \times \frac{24}{3})$
$x_4$	0	$2 - (2 \times \frac{3}{3})$	$3 - (2 \times \frac{1}{3})$	$0 - (2 \times 0)$	$1 - (2 \times 0)$	$0 - (2 \times \frac{1}{3})$	$42 - (2 \times \frac{24}{3})$
$x_1$	3	$\frac{3}{3}$	$\frac{1}{3}$	0	0	$\frac{1}{3}$	$\frac{24}{3}$

Tabel 2.4: Proses pembaharuan baris basic

Berikut tabel simplex pada akhir iterasi ke-2:

basic	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	rhs
$z$	1	0	-1	0	0	1	24
$x_3$	0	0	$\frac{1}{2}$	1	0	$-\frac{2}{3}$	2
$x_4$	0	0	$\frac{7}{3}$	0	1	$-\frac{3}{3}$	26
$x_1$	3	1	$\frac{1}{3}$	0	0	$\frac{1}{3}$	8

Tabel 2.5: Tabel simplex iterasi ke-2

Pada iterasi ke-2 masih terdapat nilai negatif pada baris  $z$ , sehingga iterasi masih dilakukan hingga tidak terdapat nilai negatif pada baris  $z$ . Berikut tabel-tabel simplex pada iterasi yang masih tersisa:

basic	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	rhs
$z$	1	0	0	3	0	-1	30
$x_2$	2	0	1	3	0	-2	6
$x_4$	0	0	0	-7	1	4	12
$x_1$	3	1	0	-1	0	1	6

Tabel 2.6: Tabel simplex iterasi ke-3

basic	$z$	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	rhs
$z$	1	0	0	$\frac{5}{4}$	$\frac{1}{4}$	0	33
$x_2$	2	0	1	$-\frac{1}{2}$	$\frac{1}{2}$	0	12
$x_5$	0	0	0	$-\frac{7}{4}$	$\frac{1}{4}$	1	3
$x_1$	3	1	0	$\frac{3}{4}$	$-\frac{1}{4}$	0	3

Tabel 2.7: Tabel simplex iterasi ke-4

Pada iterasi ke-4 sudah tidak terdapat nilai negatif pada baris  $z$ . Dengan demikian, solusi optimal sudah dicapai, yaitu pada titik  $x_1 = 3$  dan  $x_2 = 12$ .

## 2.2 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.

### 2.2.1 Tabel

Berikut adalah contoh pembuatan tabel. Penempatan tabel dan gambar secara umum diatur secara otomatis oleh L<sup>A</sup>T<sub>E</sub>X, 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 2.8 berikut ini:

Tabel 2.8: 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 2.9 dan Tabel 2.10 berikut ini adalah tabel dengan sel yang berwarna dan ada dua tabel yang bersebelahan.

Tabel 2.9: 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 2.10: 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

### 2.2.2 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: [?]

### 2.2.3 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 2.5), tipe JPG (Gambar 2.6) dan tipe PDF (Gambar 2.7)
- Besarnya gambar dapat diatur dengan *field* SCALE.



Gambar 2.5: 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*, 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 2.5 ditempatkan di bagian atas halaman, walaupun penempatannya dilakukan setelah penulisan 3 paragraf setelah penjelasan ini.
  - Gambar 2.6 dengan skala 0.5 ditempatkan di antara dua buah paragraf. Perhatikan penulisannya di dalam file bab2.tex!
  - Gambar 2.7 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

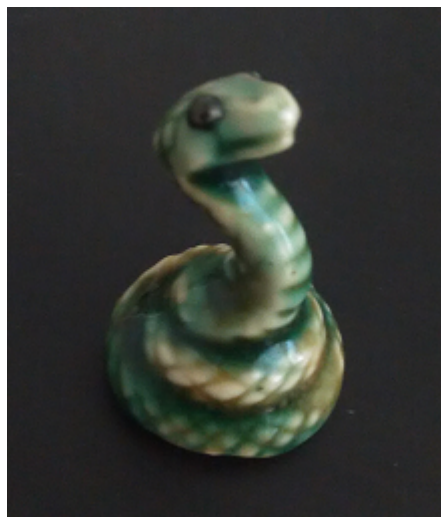
ac enim. In luctus. Phasellus eu quam vitae turpis viverra pellentesque. Duis feugiat felis ut enim. Phasellus pharetra, sem id porttitor sodales, magna nunc aliquet nibh, nec blandit nisl mauris at pede. Suspendisse risus risus, lobortis eget, semper at, imperdiet sit amet, quam. Quisque scelerisque 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 2.6: 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 2.7: *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