

UNIVERSITAS INDONESIA

**IDENTIFIKASI SUATU SEKUEN KONSENSUS PADA
SEKUEN DNA MENGGUNAKAN PEMROGRAMAN LINIER**

SKRIPSI

**DIAN NURHAYATI
0806452154**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI SARJANA MATEMATIKA
DEPOK
2012**



UNIVERSITAS INDONESIA

**IDENTIFIKASI SUATU SEKUEN KONSENSUS PADA
SEKUEN DNA MENGGUNAKAN PEMROGRAMAN LINIER**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana sains

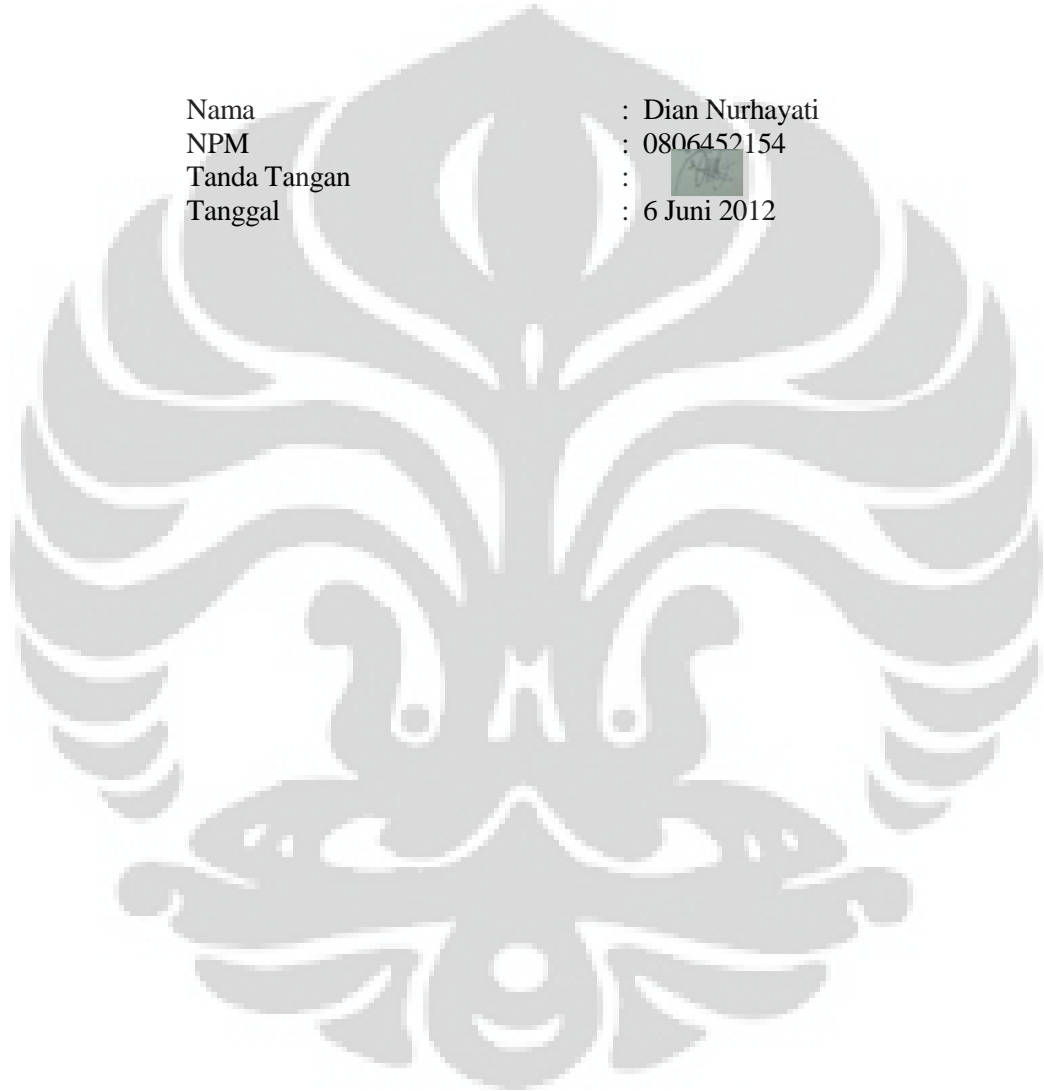
**DIAN NURHAYATI
0806452154**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
PROGRAM STUDI SARJANA MATEMATIKA
DEPOK
2012**

HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.

Nama : Dian Nurhayati
NPM : 0806452154
Tanda Tangan : 
Tanggal : 6 Juni 2012



HALAMAN PENGESAHAN

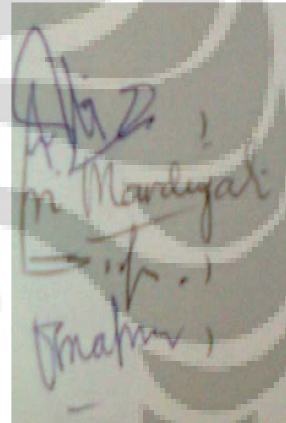
Skripsi ini diajukan oleh

Nama : Dian Nurhayati
NPM : 0806452154
Program Studi : Sarjana Matematika
Judul Skripsi : Identifikasi suatu Sekuen Konsensus Pada Sekuen DNA menggunakan Pemrograman Linier

Telah berhasil diertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi S1 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Alhadi Bustamam, PhD ()
Penguji : Dr. Sri Mardiyati, M.Kom ()
Penguji : Dr. rer. nat. Hendri Murfi, M.Kom ()
Penguji : Bevina D Handari, PhD ()



Ditetapkan di : Depok
Tanggal : 6 Juni 2012

KATA PENGANTAR

Alhamdulillah, puji syukur penulis panjatkan kepada Allah SWT, karena atas semua rahmat dan karunia yang telah Dia berikan sehingga penulis dapat menyelesaikan tugas akhir ini. Penulis sadar bahwa penyelesaian tugas akhir ini tidak terlepas dari bantuan, bimbingan dan dukungan dari berbagai pihak. Oleh karena itu, pada kesempatan ini penulis ingin mengucapkan terima kasih kepada pihak-pihak yang telah berjasa dalam penulisan tugas akhir ini maupun selama penulis kuliah. Ucapan terima kasih terhatur kepada:

- (1) Alhadi Bustamam, PhD., selaku pembimbing penulis yang telah meluangkan waktu, membimbing, memberi masukan serta memberikan bantuan kepada penulis dalam menyelesaikan skripsi ini.
- (2) Dra. Saskya Mary Soemartojo M.Si., selaku pembimbing akademis penulis selama menjalani masa kuliah.
- (3) Seluruh dosen pengajar Departemen Matematika FMIPA UI yang telah memberikan ilmu dan bimbingan dalam belajar.
- (4) Seluruh karyawan di Departemen Matematika FMIPA UI atas bantuan yang diberikan.
- (5) Bapak, ibu dan kakak-kakak penulis yang telah memberikan dukungan, doa serta bantuan kepada penulis.
- (6) S. W. Wulandari yang telah menemani masa-masa kuliah penulis selama 4 tahun.
- (7) Vika dan Icha yang sudah mendengarkan curhatan penulis dan maaf sering merepotkan.
- (8) Januarina yang telah menjadi teman dekat penulis di masa kuliah.
- (9) Teman-teman penulis di Departemen Matematika angkatan 2008 : Ade, Agnes, Maimun, Andy, Awe, Anisah, Puput, Arief, Arkies, Arman, Asri, Oline, Cindy, Citra, Danis, Dede, Dhea, Dheni, Agy, Eka, Emy, Dhila, Siwi, Ega, Hendry, Hindun, Dhewe, Nora, Janu, Qiqi, Luthfa, Adhi, Ifah, Maulia, Mayta, Mei, Nadia, Nita, Vika, Numa, Restie, Sita, Risya, Bowo, Ines, Tuti, Wulan, Ucil, Fani,

Icha, Uchida, Lian, Umbu, Yulial. Terima kasih atas kebersamaan yang telah terjalin.

- (10) Teman-teman Matematika UI.
- (11) Kakak yang telah menjalani kebersamaan selama 1,5 tahun. Terima kasih atas semangat, doa dan dukungan yang diberikan. Semoga kebersamaan kita selalu terjaga.

Penulis juga ingin mengucapkan terimakasih kepada seluruh pihak yang tidak dapat disebutkan satu per satu, yang telah membantu dalam penyusunan skripsi ini. Akhir kata, penulis mohon maaf jika terdapat kesalahan atau kekurangan dalam skripsi ini. Penulis berharap semoga skripsi ini bermanfaat bagi pembaca.

Penulis

2012

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

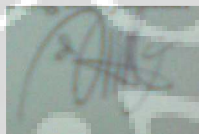
Nama : Dian Nurhayati
NPM : 0806452154
Program Studi : S1 Matematika
Departemen : Matematika
Fakultas : Matematika dan Ilmu Pengetahuan Alam
Jenis karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty Free Right*) atas karya ilmiah saya yang berjudul :

Identifikasi suatu Sekuen Konsensus pada Sekuen DNA menggunakan Pemrograman Linier beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok
Pada tanggal : 6 Juni 2012
Yang menyatakan



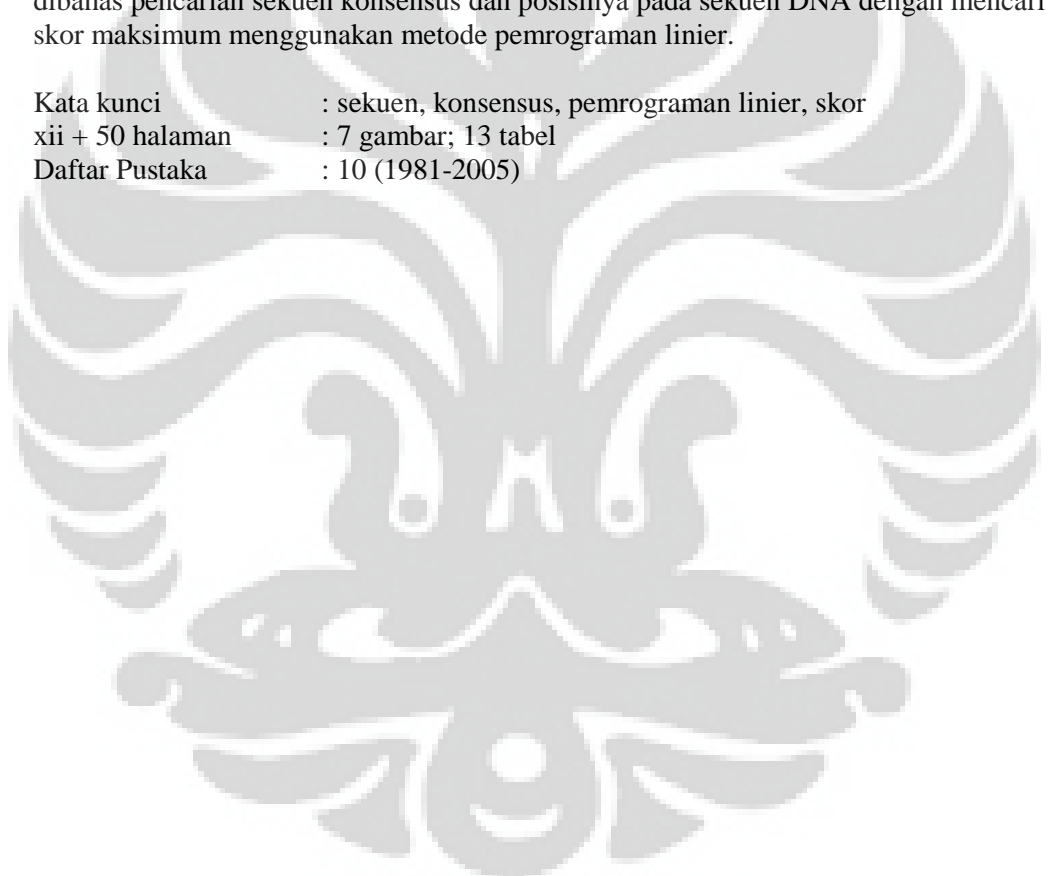
(Dian Nurhayati)

ABSTRAK

Nama : Dian Nurhayati
Program Studi : Matematika
Judul : Identifikasi suatu Sekuen Konsensus pada Sekuen DNA menggunakan Pemrograman Linier

Sekuen konsensus adalah subsekuen yang paling sering muncul pada sekuen DNA. Sekuen ini berguna untuk menentukan letak dari protein. Dalam mencari sekuen konsensus harus memperhatikan posisi tiap basa dari sekuen DNA. Terdapat pola tertentu untuk menentukan sekuen ini. Pencarian sekuen konsensus pada umumnya dilakukan dengan cara menyejajarkan subsekuen DNA dan memberikan skor, sekuen yang memberikan skor maksimum akan dijadikan sekuen konsensus. Pencarian skor maksimum ini dapat dilakukan dengan pemrograman linier. Dalam skripsi ini akan dibahas pencarian sekuen konsensus dan posisinya pada sekuen DNA dengan mencari skor maksimum menggunakan metode pemrograman linier.

Kata kunci : sekuen, konsensus, pemrograman linier, skor
xii + 50 halaman : 7 gambar; 13 tabel
Daftar Pustaka : 10 (1981-2005)

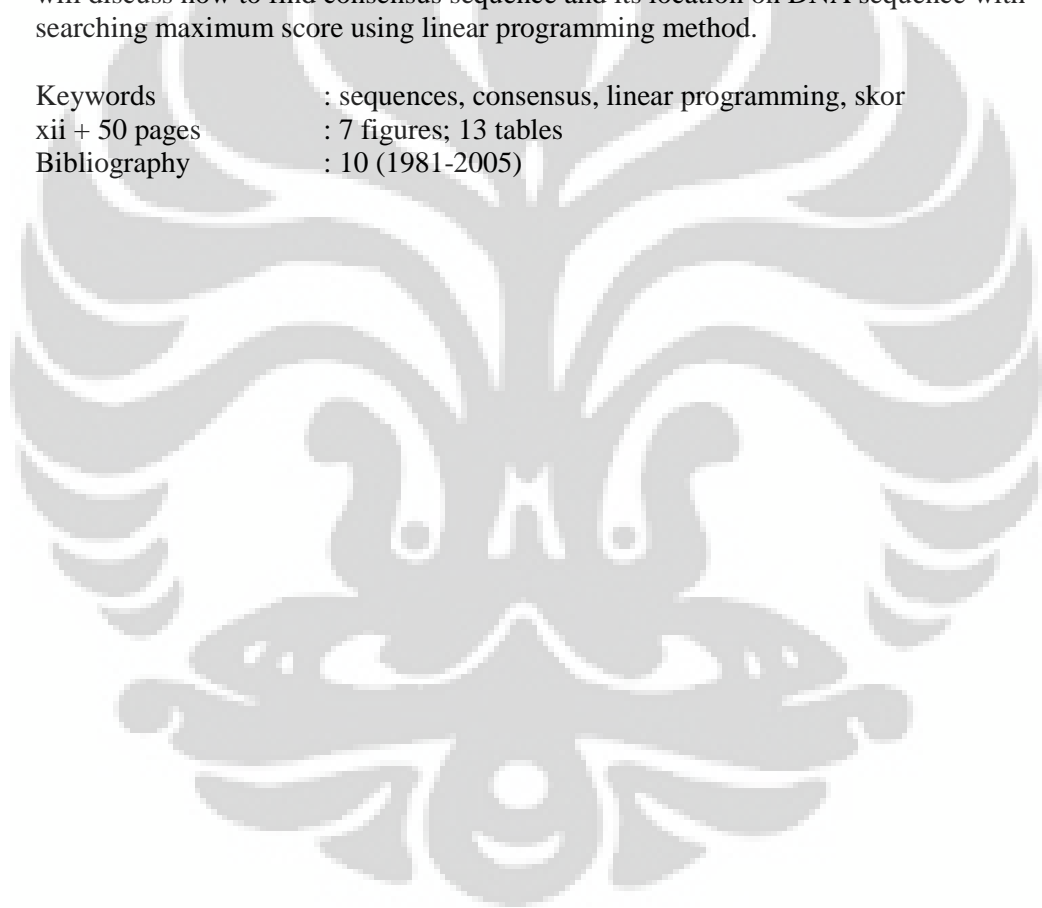


ABSTRACT

Name : Dian Nurhayati
Study Program : Mathematics
Title : Identification of a Consensus Sequence on DNA Sequences using Linear Programming

Consensus sequence is the most frequent subsequences at DNA sequences. This sequence used to find binding site of protein. To searching consensus sequence it is important to consider location every nucleotides on DNA sequences. There is specific pattern to find this sequence. Searching consensus sequence commonly done by aligning DNA subsequences and giving score, a subsequence which give maximum score will be consensus sequence. Searching maximum score can use linear programming. In this paper will discuss how to find consensus sequence and its location on DNA sequence with searching maximum score using linear programming method.

Keywords : sequences, consensus, linear programming, skor
xii + 50 pages : 7 figures; 13 tables
Bibliography : 10 (1981-2005)



DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xi
DAFTAR LAMPIRAN.....	xii
1. PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah dan Ruang Lingkup.....	2
1.3 Tujuan Penulisan.....	3
1.4 Jenis Penelitian dan Metode yang digunakan	3
2. LANDASAN TEORI.....	4
2.1 Teori Dasar Genetika	4
2.2 Teori Dasar Pemodelan Matematis	7
2.3 Teori Dasar Pemrograman Linier.....	8
2.4 Metode Simpleks.....	11
2.5 Analisa Kompleksitas.....	12
3. IDENTIFIKASI SEKUEN KONSENSUS PADA SEKUEN DNA MENGUNAKAN PEMROGRAMAN LINIER.....	14
3.1 Deskripsi Masalah.....	14
3.1 Formulasi Matematika	15
3.3 Penyelesaian Masalah	21
4. IMPLEMENTASI PROGRAM.....	24
4.1 Simulasi Program.....	26
4.2 Analisa Kompleksitas.....	27
5. PENUTUP.....	31
5.1 Kesimpulan	31
5.2 Saran.....	31
DAFTAR PUSTAKA	32
LAMPIRAN.....	33

DAFTAR GAMBAR

Gambar 2.1 Heliks ganda dari sekuen DNA.....	5
Gambar 2.2 Skema dogma sentral	5
Gambar 2.3 Tabel simpleks.....	13
Gambar 4.1 Tampilan masukan dan keluaran untuk dua sekuen DNA	26
Gambar 4.2 Grafik waktu periterasi terhadap banyaknya iterasi	27
Gambar 4.3 Tampilan masukan dan keluaran untuk dua sekuen DNA <i>E.Coli</i>	29
Gambar 4.4 Tampilan masukan dan keluaran untuk tiga sekuen DNA <i>E.Coli</i>	29

DAFTAR TABEL

Tabel 3.1 Ringkasan nilai fungsi objektif dari tabel akhir simpleks.....	23
Tabel 4.1 Estimasi waktu <i>running</i> program.....	28
Tabel Lampiran 1.1 Tabel awal simpleks untuk $z_{1,1} = 1$ dan $z_{2,1} = 1$	33
Tabel Lampiran 1.2 Tabel kedua simpleks untuk $z_{1,1} = 1$ dan $z_{2,1} = 1$	34
Tabel Lampiran 1.3 Tabel akhir simpleks untuk $z_{1,1} = 1$ dan $z_{2,1} = 1$	35
Tabel Lampiran 1.4 Tabel akhir simpleks untuk $z_{1,1} = 1$ dan $z_{2,2} = 1$	36
Tabel Lampiran 1.5 Tabel akhir simpleks untuk $z_{1,1} = 1$ dan $z_{2,3} = 1$	37
Tabel Lampiran 1.6 Tabel akhir simpleks untuk $z_{1,2} = 1$ dan $z_{2,1} = 1$	38
Tabel Lampiran 1.7 Tabel akhir simpleks untuk $z_{1,2} = 1$ dan $z_{2,2} = 1$	39
Tabel Lampiran 1.8 Tabel akhir simpleks untuk $z_{1,2} = 1$ dan $z_{2,3} = 1$	40
Tabel Lampiran 1.9 Tabel akhir simpleks untuk $z_{1,3} = 1$ dan $z_{2,1} = 1$	41
Tabel Lampiran 1.10 Tabel akhir simpleks untuk $z_{1,3} = 1$ dan $z_{2,2} = 1$	42
Tabel Lampiran 1.11 Tabel akhir simpleks untuk $z_{1,3} = 1$ dan $z_{2,3} = 1$	43

DAFTAR LAMPIRAN

Lampiran 1 Tabel simpleks.....	33
Lampiran 2 Daftar sekuen DNA <i>E.Coli</i>	44
Lampiran 3 <i>Listing</i> Program	46



BAB 1

PENDAHULUAN

1.1 Latar Belakang

DNA (*deoxyribonucleid acid*) merupakan dasar hereditas yang disusun oleh empat nukleotida, yaitu adenin (A), guanin (G), sitosin (C) dan timin (T). Ketika keempat nukleotida tersebut tersusun menjadi polimier linier berdimensi satu, akan terbentuk sekuen DNA (Yi-Ping, 1998). Schneider (2002), menjelaskan bahwa dalam sekuen DNA terdapat sekuen dasar yang sering muncul di sekuen lainnya setelah sekuen-sekuen DNA itu disejajarkan. Sekuen dasar ini disebut sekuen konsensus. Sekuen konsensus ini dapat digunakan untuk menemukan *binding site* yang merupakan tempat melekatnya protein. Dalam aplikasinya, *binding site* digunakan sebagai target aksi untuk meletakkan molekul baru pada perancangan obat (Yi-Ping, 1998).

Salah satu kesulitan dalam mencari sekuen konsensus menurut Schneider (2002) adalah perbedaan sekuen konsensus pada setiap sekuen DNA. Setiap sekuen DNA akan mempunyai sekuen konsensusnya sendiri, akan tetapi sekuen konsensus yang diperoleh dari satu sekuen DNA mungkin saja berbeda dengan yang diperoleh di sekuen DNA lainnya. Misalkan pada tiga sekuen DNA, akan dicari sekuen konsensus terdiri dari enam basa. Ketiga sekuen konsensus yang diperoleh yaitu TATAAT, GATAAT dan TATAAT. Sekuen DNA pertama dan kedua mempunyai sekuen konsensus yang berbeda, sedangkan sekuen DNA pertama dan ketiga mempunyai sekuen konsensus yang sama. Sekuen konsensus yang akan kita cari adalah sekuen konsensus yang paling sering muncul. Untuk tiga sekuen DNA tersebut akan diperoleh sekuen konsensus TATAAT. Kesulitan lainnya dalam menemukan sekuen konsensus yaitu posisi sekuen konsensus pada setiap sekuen DNA berbeda (Schneider, 2002).

Prinsip dasar untuk mencari sekuen konsensus adalah dengan menyejajarkan sekuen-sekuen DNA untuk dibandingkan dan dihitung skor kemiripannya. Ada beberapa metode yang pernah digunakan untuk mencari sekuen konsensus, diantaranya adalah metode *Maximum Likelihood* oleh Brazma

(1998) dan metode Pemrograman Nonlinier oleh Ecker (2002). Namun dua metode tersebut mempunyai kekurangan karena hanya menemukan nilai yang merupakan optimum lokal bukan optimum global (Han-Lin dan Chang-Jui, 2005). Sehingga dalam skripsi ini akan digunakan metode lain yang dapat menemukan nilai optimum global dari skor kemiripan tersebut. Salah satu metode yang dapat digunakan adalah dengan menggunakan pemrograman linier. Solusi optimal dari pemrograman linier ini akan menghasilkan nilai optimum global (Schrage, 1999).

Pemrograman linier adalah suatu teknik matematika dalam memilih program terbaik dari sekumpulan daerah layak. Dalam pemrograman linier terdapat beberapa variabel yang di gunakan yaitu variabel terikat berupa fungsi tujuan, serta variabel bebas berupa variabel keputusan. Metode ini disebut linier karena hubungan antar variabelnya bersifat linier. Persoalan pemrograman linier adalah bagaimana mengoptimalkan (maksimum atau minimum) variabel terikat (fungsi tujuan dari variabel keputusan) terhadap kendala linier. Solusi optimal dari pemrograman linier adalah nilai-nilai dari variabel keputusan yang dapat dimasukkan ke dalam fungsi tujuan sedemikian sehingga fungsi tujuan akan menghasilkan nilai yang optimal (Wu dan Coppins, 1981). Dalam skripsi ini akan dilakukan identifikasi sekuen konsensus dari sekuen DNA menggunakan metode pemrograman linier.

1.2 Perumusan Masalah dan Ruang Lingkup

Bagaimanakah menemukan sekuen konsensus dan posisinya pada sekuen - sekuen DNA yang telah diketahui menggunakan pemrograman linier. Permasalahan yang dibahas dalam skripsi ini hanya mencari sekuen konsensus dengan pola $L_1L_2L_3L_4L_5xxxxxxL_6L_7L_8L_9L_{10}$, dimana $L_i, x \in \{A, C, T, G\}$ dan x adalah gen promotor, dengan $i = 1, 2, \dots, 10$. Serta minimal terdapat dua sekuen DNA yang telah diketahui.

1.3 Tujuan Penulisan

Dari rumusan masalah diatas tugas akhir ini bertujuan untuk menemukan sekuen konsensus dan posisinya pada sekuen DNA yang telah diketahui menggunakan pemrograman linier.

1.4 Jenis Penelitian dan Metode yang Digunakan

Penelitian ini dilakukan dengan menggunakan studi pustaka serta simulasi program.



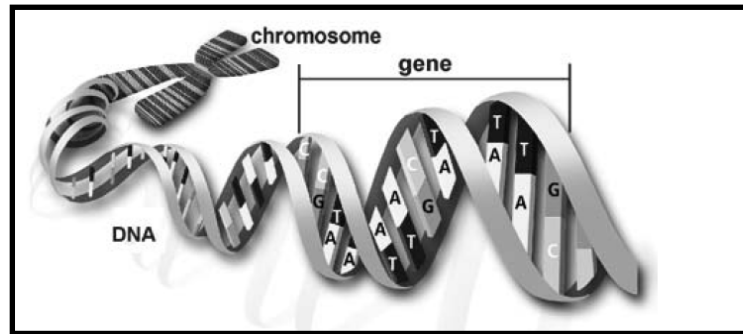
BAB 2 LANDASAN TEORI

Pada bab ini akan dijelaskan teori-teori yang akan digunakan dalam bab selanjutnya. Teori-teori tersebut mencakup teori dasar genetika, pemodelan matematis, pemrograman linier, metode simpleks serta analisa kompleksitas.

2.1 Teori Dasar Genetika

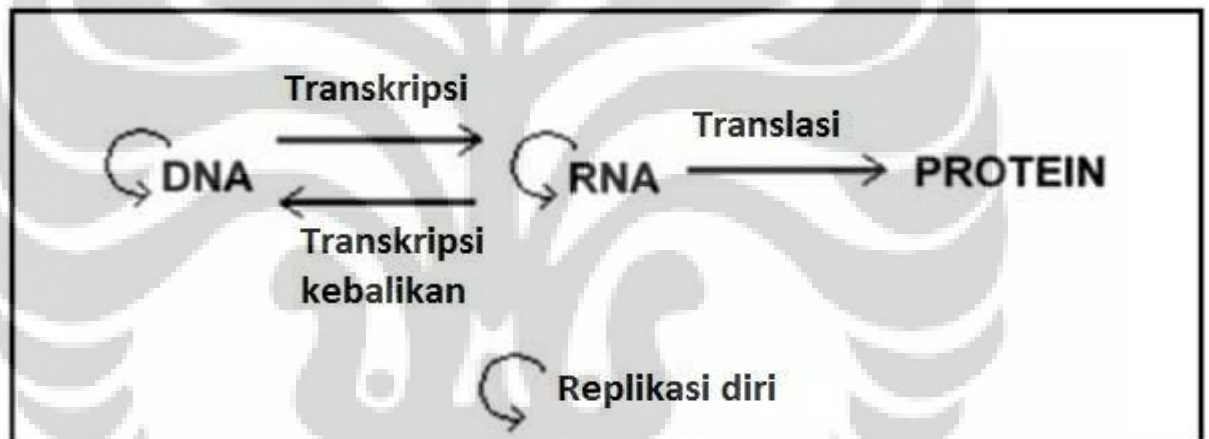
DNA (*deoxyribonucleid acid*) merupakan sebuah polimer yang terbentuk dari molekul kecil yang disebut **nukleotida** dan berperan sebagai dasar hereditas. Informasi mengenai hereditas ini digambarkan dalam sekuen DNA mereka. **Sekuen DNA** adalah polimer linier berdimensi satu yang disusun oleh empat nukleotida berbeda, yaitu adenin (A) sitosin (C) guanin (G) dan Timin (T). Dengan kata lain sekuen DNA adalah sekuen yang disusun oleh 4 huruf A, C, T dan G, selanjutnya 4 huruf ini akan disebut sebagai **basa nukleotida** atau **basa**. Identifikasi sekuen DNA tersebut dapat memberikan informasi mengenai perbedaan spesies yang satu dengan yang lainnya, walaupun mereka mempunyai nukleotida yang sama. DNA organisme ditentukan dengan proses yang disebut *sequencing*. *DNA sequencing* meliputi proses penentuan empat basa nukleotida A, C, G dan T hingga membentuk sekuen DNA. Proses *sequencing* yang sering digunakan biasanya dilakukan dengan metode pemisahan fragmen DNA dengan *gel electrophoresis* (Yi-Ping, 1998).

DNA terdapat di untai ganda dan setiap basa dalam 2 untai tersebut saling komplemen. Misalnya A berpasangan dengan T dan G berpasangan dengan C, mereka saling dipasangkan dengan ikatan hidrogen. Untai ganda ini berbentuk heliks, sehingga lebih dikenal dengan sebutan **heliks ganda** (*double helix*) (Yi-Ping, 1998).



Gambar 2.1 Heliks ganda dari skuen DNA [Yi-Ping, 1998].

Informasi genetik yang dibutuhkan oleh organisme dibawa oleh DNA. Alur perjalanan informasi dimulai dari DNA sampai diterima oleh sel, diringkas dalam **Dogma Sentral** (Yi-Ping, 1998). Dapat digambarkan dengan skema sebagai berikut:



Gambar 2.2 Skema dogma sentral.

Langkah awal dari DNA untuk sintesis protein disebut transkripsi. Transkripsi yaitu proses penyalinan informasi di DNA menjadi salinan yang disebut RNA. Jika bagian dari sekuen DNA yang disalin untuk dijadikan RNA berfungsi menerjemahkan protein, RNA yang terbentuk disebut *messenger*, atau mRNA. Selain itu juga terdapat transfer RNAs (tRNAs) dan ribosomal RNAs (rRNAs), yang dibutuhkan pada saat translasi. Dalam transkripsi, basa nukleotida DNA A, C, T, G berturut-turut dituliskan ke dalam basa nukleotida RNA menjadi A, C, U (urasil (U) menggantikan timin (T)) dan G. (Yi-Ping, 1998).

Transkripsi melalui 3 tahapan proses : (1) Inisiasi, dimulai dengan enzim yang disebut RNA polimerase menempel pada barisan di hulu (*upstream*) gen promotor. **Gen promotor** merupakan gen yang sering muncul pada sekuen DNA, tetapi gen promotor ini hanya mempunyai satu fungsi yaitu sebagai titik awal proses transkripsi; (2) Elongasi (pemanjangan), setelah promotor RNA polimerase melekat pada enzim tersebut akan terus bergerak sepanjang molekul DNA, mengurai dan meluruskan heliks tersebut; (3) Terminasi (Penghentian), dimana polimerase RNA mencapai barisan terminator di DNA dan kemudian transkripsi berhenti. Selama transkripsi, untai heliks ganda DNA yang pertama berpisah dengan memutus ikatan hidrogen pasangan nukleotida. Salah satu untai dari DNA ini kemudian digunakan untuk menjadi *template* untai tunggal RNA (Yi-Ping, 1998).

Tahap akhir dalam skema Gambar 2.2 adalah translasi, yaitu proses penerjemahan RNA menjadi protein. Informasi yang telah diterjemahkan di mRNA digunakan untuk menyusun asam amino untuk membentuk protein. Protein merupakan komposisi dari 20 asam amino berbeda. Masing-masing asam amino ini diterjemahkan dari komposisi 3 nukleotida. Terdapat 4 basa nukleotida pada saat translasi (A, G, U dan C), sehingga terbentuk $4^3 = 64$ kombinasi basa nukleotida untuk menghasilkan asam amino. Satu asam amino yang sama dapat diterjemahkan dari beberapa komposisi basa nukleotida yang berbeda, seperti asam amino *Leucine (leu)* dapat diterjemahkan dari 6 komposisi basa nukleotida yang berbeda. Selain itu terdapat 3 komposisi nukleotida UAA, UAG dan UGA yang tidak diterjemahkan menjadi asam amino, komposisi basa nukleotida ini berfungsi untuk menghentikan proses translasi.

Dalam sekuen DNA juga terdapat sekuen dasar yang sering muncul di sekuen lainnya setelah sekuen-sekuen DNA itu disejajarkan. Sekuen dasar ini disebut **sekuen konsensus**. Sekuen konsensus ini dapat digunakan untuk menemukan *binding site* yang merupakan tempat melekatnya protein (Schneider, 2002). *Binding site* ini sebagai tempat faktor transkripsi, dimana faktor transkripsi merupakan elemen pengatur untuk melaksanakan transkripsi sampai terbentuknya protein. Salah satu kegunaan *binding site* adalah sebagai lokasi target aksi untuk penempatan molekul baru pada perancangan obat (Yi-Ping, 1998).

2.2 Teori Dasar Pemodelan Matematis

Model adalah suatu objek atau konsep yang digunakan untuk merepresentasikan sesuatu. Hal yang akan dimodelkan tersebut diubah ke dalam ukuran dan bentuk yang mudah dipahami. Model matematika adalah suatu model yang bagian-bagiannya menggunakan konsep matematika, seperti konstanta, variabel, fungsi, pertidaksamaan dan lain-lain (Meyer, 1985).

Menurut Meyer (1985) model matematika tidak dapat dibentuk menggunakan alat atau program. Pembentukan model matematika ini dilakukan secara manual. Ada tiga tahapan yang dilalui untuk membentuk model matematika, yaitu : Formulasi, Manipulasi Matematika dan Evaluasi.

1. Formulasi

- Menentukan pertanyaan. Dalam persoalan yang akan di buat model matematikanya, pertanyaan di dalamnya biasanya masih kurang jelas, sehingga kita harus memperjelasnya, sedangkan jika dalam masalah tersebut pertanyaannya terlalu luas, yang harus dilakukan adalah membagi pertanyaan tersebut kedalam beberapa subbagian.
- Tentukan faktor-faktor yang penting dalam masing-masing pertanyaan dan hubungannya.
- Deskripsi matematika. Masing-masing perhitungan di representasikan ke dalam bentuk matematika yang cocok, contohnya variabel, fungsi, dan bentuk geometris. Masing-masing hubungan variabel atau hasil penghitungan dapat direpresentasikan menggunakan persamaan, pertidaksamaan atau asumsi matematika lainnya yang cocok.

2. Manipulasi Matematika

Formulasi matematika jarang memberikan jawaban secara langsung. Perlu digunakan proses matematika untuk menyelesaikannya. Proses ini mengandung perhitungan, menyelesaikan persamaan, membuktikan teorema dan lain-lain.

3. Evaluasi

Dalam memutuskan apakah suatu model baik atau tidak, ada banyak hal yang dapat kita hitung. Pertanyaan paling penting adalah apakah model

tersebut memberikan jawaban yang baik atau tidak. Jika jawabannya tidak cukup akurat, atau model ini mempunyai kekurangan, kita harus berusaha mengidentifikasi kekurangannya. Mungkin kesalahan terjadi ketika manipulasi matematika, tetapi dalam kebanyakan kasus, yang dibutuhkan adalah formulasi baru. Sebagai contoh, terdapat beberapa perhitungan atau hubungan yang kita abaikan padahal sesungguhnya penting. Setelah terbentuk formulasi baru, kita akan melakukan manipulasi matematika baru dan evaluasi baru. Sehingga model matematika baru terbentuk.

2.3 Teori Dasar Pemrograman Linier

Pemrograman linier adalah suatu teknik matematika dalam memilih program terbaik dari sekumpulan daerah layak. Metode ini disebut linier karena hubungan antar variabelnya bersifat linier (Wu dan Coppins, 1981).

Definisi 2.1 (Winston, 1995). Fungsi $f(x_1, x_2, \dots, x_n)$ dari x_1, x_2, \dots, x_n adalah fungsi linier jika dan hanya jika untuk beberapa konstanta c_1, c_2, \dots, c_n , $f(x_1, x_2, \dots, x_n) = c_1x_1 + c_2x_2 + \dots + c_nx_n$.

Definisi 2.2 (Winston, 1995). Untuk fungsi linier $f(x_1, x_2, \dots, x_n)$ dan sebuah nilai b , maka pertidaksamaan $f(x_1, x_2, \dots, x_n) \leq b$ dan $f(x_1, x_2, \dots, x_n) \geq b$ adalah pertidaksamaan linier.

Menurut Winston (1995), dalam pemrograman linier ini terdapat :

1. Variabel keputusan yaitu variabel bersangkutan yang mendeskripsikan tujuan.
2. Fungsi objektif atau fungsi tujuan yaitu fungsi yang bertujuan untuk memaksimalkan atau meminimumkan.
3. Kendala yaitu syarat yang harus dipenuhi oleh variabel keputusan.

Persoalan pemrograman linier adalah bagaimana mengoptimalkan (maksimum atau minimum) variabel terikat (fungsi linier dari variabel bebas) terhadap kendala linier. Solusi optimal dari pemrograman linier adalah nilai-nilai dari variabel keputusan yang dapat dimasukkan ke fungsi tujuan sehingga akan menghasilkan nilai yang optimal (Wu dan Coppins, 1981). Secara umum pemrograman linier dapat ditulis sebagai berikut :

Optimasi	$f = c_1x_1 + c_2x_2 + \dots + c_nx_n = \sum_{j=1}^n c_jx_j$
Kendala	$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n \leq b_1$ $a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n \leq b_2$ $a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n \leq b_m$
Atau	$\sum_{j=1}^n a_{ij}x_j \leq b_j, \quad i = 1, 2, \dots, m$ $x_1, x_2, \dots, x_n \geq 0$

Dimana :

x_1, x_2, \dots, x_n = variabel keputusan

f = fungsi tujuan

c_1, c_2, \dots, c_n = koefisien fungsi tujuan

$a_{i1}, a_{i2}, \dots, a_{in}$ = koefisien variabel keputusan pada kendala ke- i

b_i = konstanta dari kendala ke- i

(Wu dan Coppins, 1981)

Solusi yang ditemukan dari bentuk model pemrograman linier adalah optimum global (Schrage, 1999). Suatu nilai x_0 dikatakan optimum lokal pada $f(x)$ apabila $f(x_0) \geq f(x)$ ketika memaksimumkan dan $f(x_0) \leq f(x)$ ketika meminimumkan, untuk nilai x yang berada pada sebuah selang kecil yang berpusat di x_0 , sedangkan optimum global terjadi apabila pertidaksamaan tersebut berlaku untuk semua nilai x (dalam domain yang didefinisikan $f(x)$) (Wu dan Coppins, 1981).

Teorema 2.1 (Bronson, 1988). Jika $f(x)$ kontinu pada selang tertutup dan terbatas $[a, b]$, maka $f(x)$ memiliki optimum global (baik maksimum maupun minimum) pada selang ini.

Teorema 2.2 (Bronson, 1988). jika $f(x)$ memiliki sebuah optimum lokal di x_0 dan jika $f(x)$ dapat didefinisikan pada sebuah selang kecil yang berpusat di x_0 , maka $f'(x_0) = 0$.

Teorema 2.3 (Bronson, 1988). jika $f(x)$ dapat didefinisikan dan diferensialkan dua kali pada sebuah selang kecil yang berpusat di x_0 , dan jika $f'(x_0) = 0$ dan $f''(x_0) > 0$, maka $f(x)$ memiliki sebuah minimum lokal di x_0 . Sedangkan jika $f'(x_0) = 0$ dan $f''(x_0) < 0$, maka $f(x)$ memiliki sebuah maksimum lokal di x_0 .

Contoh model pemrograman linier:

Sebuah pabrik mebel yang memproduksi meja dan kursi, memerlukan dua jenis kayu yang dibutuhkan oleh masing-masing produk. Setiap meja memerlukan 5 m kayu jenis A dan 2 m kayu jenis B serta 4 jam waktu pengerjaannya. Sedangkan setiap kursi memerlukan 2 m kayu jenis A dan 3 m kayu jenis B serta 2 jam waktu pengerjaannya. Dalam penjualan produknya, pabrik ini akan memperoleh keuntungan Rp. 12.000 untuk setiap meja dan Rp. 8.000 untuk setiap kursi. Namun dalam seminggu hanya terdapat persediaan 150 m kayu jenis A dan 100 m kayu jenis B serta 80 jam pengerjaan. Pabrik ini ingin mengetahui berapa banyak produk yang harus diproduksi untuk memaksimalkan keuntungan.

Variabel keputusan

x_1 : Banyaknya meja yang diproduksi

x_2 : Banyaknya kursi yang diproduksi

1. Fungsi objektif

Pabrik ingin memaksimalkan keuntungan. Setiap penjualan meja memberikan keuntungan Rp. 12.000 sehingga total keuntungan yang diperoleh dari penjualan meja adalah $12.000x_1$. Sedangkan setiap penjualan kursi akan memberikan keuntungan sebesar Rp. 8000, maka total keuntungan yang diperoleh dari penjualan kursi adalah $8.000x_2$. Fungsi f merupakan fungsi objektif yang ingin memaksimalkan total keuntungan yang diperoleh dari penjualan kedua produk tersebut, sehingga fungsi tersebut adalah :

$$f = 12000x_1 + 8000x_2$$

2. Kendala

Terdapat tiga kendala pada masalah ini. Persediaan kayu jenis A dalam seminggu sebanyak 150 m, untuk membuat meja dibutuhkan 5 m dan untuk membuat kursi diperlukan 2 m. sehingga kendala untuk kayu jenis A adalah:

$$5x_1 + 2x_2 \leq 150$$

Persediaan kayu jenis B selama seminggu sebanyak 100 m. untuk membuat meja memerlukan 2 m kayu jenis B dan untuk pembuatan kursi memerlukan 3 m kayu jenis B. sehingga kendala untuk kayu jenis B adalah :

$$2x_1 + 3x_2 \leq 100$$

Produksi meja dan kursi tersebut hanya dapat dilakukan selama 80 jam dalam seminggu. Sedangkan untuk pembuatan meja memerlukan 4 jam pengerjaan serta untuk pembuatan kursi memerlukan 2 jam pengerjaan. Sehingga kendala untuk waktu pengerjaan adalah :

$$4x_1 + 2x_2 \leq 80$$

Masing-masing variabel keputusan harus nonnegatif, kendalanya dapat ditulis sebagai berikut :

$$x_1 \geq 0 \text{ dan } x_2 \geq 0$$

Masalah pabrik mebel tersebut dapat diringkas menjadi:

Memaksimumkan $f = 12000x_1 + 8000x_2$

Dengan syarat $5x_1 + 2x_2 \leq 150$ kayu jenis A

$2x_1 + 3x_2 \leq 100$ kayu jenis B

$4x_1 + 2x_2 \leq 80$ waktu pengerjaan

$x_1, x_2 \geq 0$ nonnegativitas

Contoh diatas merupakan contoh pemodelan masalah menjadi pemrograman linier. Masalah pencarian sekuen konsensus juga dapat dimodelkan menjadi bentuk pemrograman linier yang akan dibahas dalam bab selanjutnya.

2.4 Metode Simpleks

Untuk menyelesaikan pemrograman linier dapat menggunakan salah satu metode yaitu metode simpleks. Metode simpleks dikembangkan oleh Prof. George B. Dantzig untuk menyelesaikan masalah pemrograman linier (Wu dan Coppins, 1981). Metode ini terdiri dari beberapa tahapan yaitu.

Tahap 1 : Bentuk masalah pemrograman linier kedalam bentuk standar.

Tahap 2 : Pilih m variabel (m adalah banyaknya kendala) untuk membentuk basis awal yang layak, variabel ini disebut variabel dasar. Semua variabel bukan dasar nilainya dibuat sama dengan nol.

Tahap 3 : Evaluasi variabel bukan dasar untuk melihat variabel mana yang memperbaiki fungsi objektif. Jika tidak ada yang memperbaiki maka solusi sudah

optimal dan variabel yang terdapat dalam basis adalah variabel yang memaksimalkan dengan nilai masing-masing variabel terdapat pada kolom b_i . Jika ada yang memperbaiki maka variabel yang memberikan perbaikan terbesar per unit pada fungsi tujuan dipilih untuk masuk basis dan lanjutkan ke tahap 4.

Tahap 4 : Pilih salah satu variabel dasar untuk meninggalkan basis, dengan tetap menjaga ketidaknegativan, kemudian lanjutkan ke tahap 5.

Tahap 5 : Lakukan perubahan basis dengan *pivoting*. Variabel baru yang masuk kedalam basis telah dipilih pada tahap 3 sedangkan variabel yang keluar dari basis dipilih pada tahap 4. Kemudian evaluasi fungsi objektif.

Tahap 6. Kembali ke tahap 3.

			Baris fungsi objektif						
i	c_B	x_B	x_1	x_2	...	x_{n-1}	x_n	b_i	b_i/c_B
1	Variabel							Nilai dari variabel basis	Variabel mana yang akan meninggalkan basis
2	basis dan		Koefisien substitusi						
:	profitnya								
m									
	z_j		Variabel mana yang akan masuk					f	
	$c_j - z_j$		basis						

Gambar 2.3 Tabel simpleks [Wu dan Coppins, 1981].

Pada bab selanjutnya akan dijelaskan pencarian sekuen konsensus pada sekuen DNA dengan mencari skor maksimum menggunakan pemrograman linier. Sekuen-sekuen DNA ini akan dimodelkan kedalam bentuk pemrograman linier.

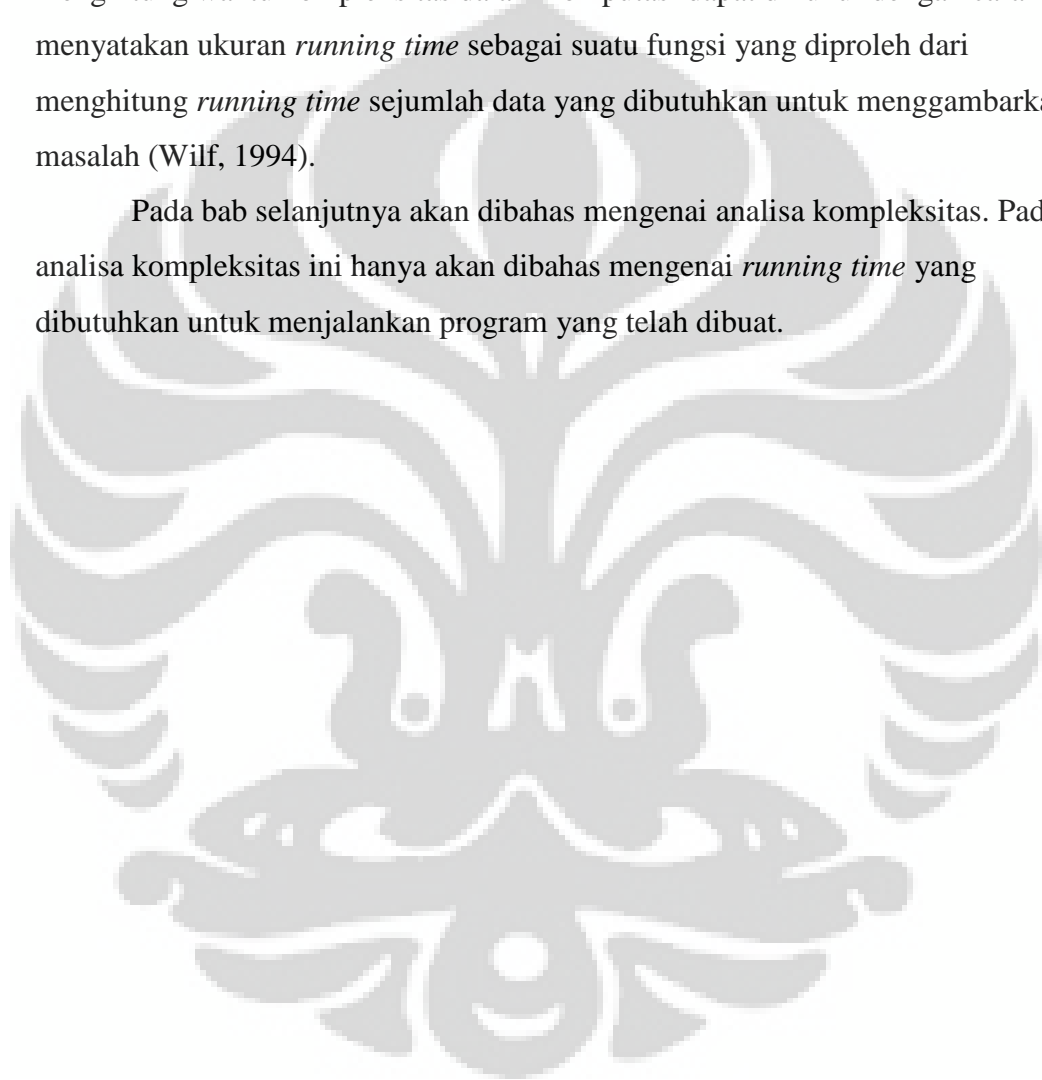
2.5 Analisa Kompleksitas

Algoritma merupakan susunan proses yang digunakan untuk menyelesaikan masalah menggunakan komputer. Dan kompleksitas suatu algoritma adalah biaya, lamanya *running time*, memori yang dipakai, atau semua

unit yang berhubungan dengan penggunaan algoritma untuk menyelesaikan persoalan menggunakan komputer (Wilf, 1994).

Proses komputasi memerlukan waktu untuk menyelesaikan masalah. Beberapa masalah kadang memerlukan banyak waktu, namun ada juga yang dapat diselesaikan dalam waktu yang cepat. Waktu yang digunakan dalam komputasi ini akan berbeda-beda tergantung dari jenis masalah yang digunakan. Untuk menghitung waktu kompleksitas dalam komputasi dapat diukur dengan cara menyatakan ukuran *running time* sebagai suatu fungsi yang diperoleh dari menghitung *running time* sejumlah data yang dibutuhkan untuk menggambarkan masalah (Wilf, 1994).

Pada bab selanjutnya akan dibahas mengenai analisa kompleksitas. Pada analisa kompleksitas ini hanya akan dibahas mengenai *running time* yang dibutuhkan untuk menjalankan program yang telah dibuat.



BAB 3

IDENTIFIKASI SEKUEN KONSENSUS PADA SEKUEN DNA MENGUNAKAN PEMROGRAMAN LINIER

Pada bab ini akan dibahas pencarian sekuen konsensus dari sekuen-sekuen DNA. Pada bab sebelumnya telah dijelaskan bahwa sekuen konsensus adalah sekuen dasar yang sering muncul di sekuen lainnya setelah sekuen-sekuen DNA itu disejajarkan. Pencarian sekuen konsensus ini menggunakan prinsip pencarian skor maksimum. Untuk mencari skor maksimum tersebut akan digunakan metode pemrograman linier yang telah dibahas dalam bab sebelumnya.

Pada bab ini akan dijelaskan deskripsi masalah yang diberikan pada Subbab 3.1. Selanjutnya dibuat formulasi matematis yang kemudian dibentuk menjadi model pemrograman linier yang akan dibahas pada Subbab 3.2. Menggunakan contoh sekuen DNA yang diberikan, akan dibahas penyelesaian masalah dengan model pemrograman linier menggunakan metode simpleks yang akan dijelaskan pada Subbab 3.3.

3.1 Deskripsi Masalah

Dari bagian tubuh suatu organisme, diambil sejumlah p sekuen DNA. Masing-masing sekuen DNA tersebut terdiri dari q buah basa. Kemudian akan dicari sekuen konsensus dari sekuen-sekuen DNA tersebut. Sekuen konsensus yang dicari ini terdiri dari 16 basa serta mempunyai pola $L_1L_2L_3L_4L_5xxxxxxL_6L_7L_8L_9L_{10}$. Dengan $L_i, x \in \{A, C, T, G\}$ untuk $i = 1, 2, \dots, 10$. x merupakan gen promotor, sehingga keberadaan x ini dapat diabaikan dari pembentukan sekuen konsensus. Sekuen DNA yang terdiri dari q basa ini dapat dibagi menjadi beberapa kandidat sekuen konsensus yang masing-masing mempunyai 16 basa. Pencarian sekuen konsensus ini menggunakan teknik pemberian skor, sehingga untuk mendapatkan sekuen konsensus ini akan dicari skor maksimumnya. Dari masalah diatas, definisikan:

- p = banyaknya sekuen DNA
- q = banyaknya basa pada tiap sekuen DNA

- x = gen promotor $\in \{A, C, T, G\}$
 L_i = basa pada posisi i dalam sekuen konsensus, $L_i \in \{A, C, T, G\}$ untuk
 $i = 1, 2, \dots, 10$
 “site” = letak kandidat sekuen konsensus pada sekuen DNA
 l = posisi sekuen DNA
 n = posisi basa pada sekuen DNA
 s = posisi awal kandidat sekuen konsensus pada sekuen DNA

3.2 Formulasi Matematika

Sebanyak p sekuen DNA yang masing-masing mempunyai q buah basa, dapat direpresentasikan dalam bentuk matriks sebagai berikut :

$$D = \begin{bmatrix} b_{1,1} & b_{1,2} & \dots & b_{1,q} \\ b_{2,1} & b_{2,2} & \dots & b_{2,q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{p,1} & b_{p,2} & \dots & b_{p,q} \end{bmatrix}$$

Dimana $b_{l,n}$ adalah basa di posisi n pada sekuen DNA ke- l . Dengan $l = 1, 2, \dots, p$ dan $n = 1, 2, \dots, q$.

Contoh 1:

Misalkan terdapat dua sekuen DNA yang masing-masing terdiri dari 18 basa.

AAGACTGTTTTTTTGATC

GATTATTTGCACGGCGTC

Sekuen-sekuen DNA ini dapat direpresentasikan sebagai matriks menjadi:

$$D = \begin{bmatrix} A & A & G & A & C & T & G & T & T & T & T & T & T & G & A & T & C \\ G & A & T & T & A & T & T & T & G & C & A & C & G & G & C & G & T & C \end{bmatrix}$$

Untuk mencari sekuen konsensus ini, terlebih dahulu akan ditentukan kandidat-kandidat sekuen konsensus yang mungkin dari setiap sekuen DNA. Masing-masing kandidat tersebut harus terdiri dari 16 basa. Misalkan dengan menggunakan sekuen DNA AAGACTGTTTTTTTGATC, akan dicari banyaknya kandidat sekuen konsensus. Posisi awal sekuen konsensus dimulai dari basa paling kiri pada sekuen DNA. Selanjutnya posisi awal sekuen konsensus berikutnya dimulai dari satu basa di sebelah kanan basa yang digunakan sebagai posisi awal sekuen konsensus sebelumnya. Berikut ini sekuen konsensus yang

diperoleh dengan menggunakan basa pertama, kedua dan ketiga dari sekuen DNA sebagai posisi awal kandidat sekuen konsensus.

AAGACTGTTTTTTTGATC

AAGACTGTTTTTTTGATC

AAGACTGTTTTTTTGATC

Ketika menggunakan basa ketiga di sekuen DNA sebagai posisi awal dari sekuen konsensus, basa terakhir sekuen DNA menjadi bagian kandidat sekuen konsensus tersebut. Akibatnya, ketika posisi awal sekuen konsensus di letakkan pada basa yang berada di sebelah kanan basa ketiga sekuen DNA, banyaknya kandidat sekuen konsensus yang terbentuk akan kurang dari 16, yaitu hanya mempunyai 15 basa. Sehingga secara umum banyaknya kandidat yang mungkin pada setiap sekuen DNA adalah $(q - 16) + 1 = q - 15$ kandidat.

Berdasarkan matriks D diatas, akan dibentuk matriks D' yang berisi kandidat-kandidat sekuen konsensus pada masing-masing sekuen DNA. Untuk memudahkan penulisan dan penghitungan, maka 6 gen promotor dapat diabaikan dan tidak ditulis dalam kandidat sekuen konsensus. Sehingga setiap kandidat sekuen konsensus hanya akan mempunyai masing-masing 10 buah basa. Matriks D' yang dimaksud adalah:

$$D' = \begin{bmatrix} d_{1,1}^1 & \dots & d_{1,1}^{10} & d_{1,2}^1 & \dots & d_{1,2}^{10} & \dots & d_{1,q-16+1}^1 & \dots & d_{1,q-16+1}^{10} \\ d_{2,1}^1 & \dots & d_{2,1}^{10} & d_{2,2}^1 & \dots & d_{2,2}^{10} & \dots & d_{2,q-16+1}^1 & \dots & d_{2,q-16+1}^{10} \\ \vdots & & \vdots & \vdots & & \vdots & & \vdots & & \vdots \\ d_{p,1}^1 & \dots & d_{p,1}^{10} & d_{p,2}^1 & \dots & d_{p,2}^{10} & \dots & d_{p,q-16+1}^1 & \dots & d_{p,q-16+1}^{10} \end{bmatrix}$$

Dimana

$$d_{l,s}^i = \begin{cases} b_{l,i+s-1} & (\text{untuk } i = 1,2, \dots, 5) \\ b_{l,i+s+5} & (\text{untuk } i = 6,7, \dots, 10) \end{cases}$$

Dan $s = 1,2, \dots, q - 15$ merupakan posisi awal dari kandidat sekuen konsensus.

Dari Contoh 1:

AAGACTGTTTTTTTGATC

GATTATTTGCACGGCGTC

Banyaknya kandidat *site* yang mungkin untuk masing-masing sekuen DNA adalah $q - 15 = 18 - 15 = 3$. Kandidat-kandidat *site* tersebut yaitu :

$l = 1, s = 1$ **AAGACTGTTTTTTTGATC**

$l = 1, s = 2$ **AAGACTGTTTTTTTGATC**

$$\begin{aligned}
l = 1, s = 3 & \text{ AAGACTGTTTTTTTGGATC} \\
l = 2, s = 1 & \text{ GATTATTTGCACGGCGTC} \\
l = 2, s = 2 & \text{ GATTATTTGCACGGCGTC} \\
l = 2, s = 3 & \text{ GATTATTTGCACGGCGTC}
\end{aligned}$$

Dari hasil yang diperoleh diatas, dibentuk matriks D' . Untuk mempermudah penulisan dan penghitungan selanjutnya, 6 gen promotor dari masing-masing kandidat sekuen konsensus tersebut akan diabaikan dan tidak ditulis. Matriks D' yang terbentuk yaitu :

$$D' = \begin{bmatrix} \text{AAGACTTTGA} & \text{AGACTTTGAT} & \text{GACTGTGATC} \\ \text{GATTACGGCG} & \text{ATTATGGCGT} & \text{TTATTGCGTC} \end{bmatrix}$$

Untuk membentuk model pemrograman linier, sebelumnya akan ditentukan variabel keputusan yang akan digunakan. Misalkan L_i adalah basa di posisi i pada sekuen konsensus yang dicari dan jika L_i adalah A , maka $a_i = 1$, jika L_i adalah T maka $t_i = 1$, jika L_i adalah C maka $c_i = 1$, serta jika L_i adalah G maka $g_i = 1$, untuk $i = 1, 2, \dots, 10$. Selanjutnya $z_{l,s}$ merupakan *site* sekuen konsensus pada posisi s di sekuen ke- l dengan $z_{l,s} \in \{0,1\}$. $z_{l,s} = 1$ apabila sekuen konsensus pada *site* tersebut digunakan, sebaliknya $z_{l,s} = 0$ jika sekuen konsensus pada *site* tersebut tidak digunakan. Sehingga a_i, t_i, c_i, g_i untuk $i = 1, 2, \dots, 10$, dan $z_{l,s}$ untuk $l=1, 2, \dots, p$ dan $s=1, 2, \dots, q-15$, akan digunakan sebagai variabel keputusan pada pemrograman linier.

Kemudian akan dibentuk fungsi objektif untuk menghasilkan skor maksimum. Nilai optimal diperoleh dengan memaksimumkan hasil penjumlahan dari semua $skor_l$. $skor_l$ adalah skor pada sekuen DNA ke- l .

$$\text{skor akhir} = \text{maks } skor_total = \text{maks } \sum_{l=1}^p skor_l \quad (3.1)$$

Berdasarkan Contoh 1:

$$\begin{aligned}
skor_1 &= z_{1,1}(a_1 + a_2 + g_3 + a_4 + c_5 + t_6 + t_7 + t_8 + g_9 + a_{10}) \\
&\quad + z_{1,2}(a_1 + g_2 + a_3 + c_4 + t_5 + t_6 + t_7 + g_8 + a_9 + t_{10}) \\
&\quad + z_{1,3}(g_1 + a_2 + c_3 + t_4 + g_5 + t_6 + g_7 + a_8 + t_9 + c_{10}) \\
skor_2 &= z_{2,1}(g_1 + a_2 + t_3 + t_4 + a_5 + c_6 + g_7 + g_8 + c_9 + g_{10}) \\
&\quad + z_{2,2}(a_1 + t_2 + t_3 + a_4 + t_5 + g_6 + g_7 + c_8 + g_9 + t_{10}) \\
&\quad + z_{2,3}(t_1 + t_2 + a_3 + t_4 + t_5 + g_6 + c_7 + g_8 + t_9 + c_{10})
\end{aligned}$$

$$\begin{aligned}
\text{skor_total} = & a_1(z_{1,1} + z_{1,2} + z_{2,2}) + g_1(z_{1,3} + z_{2,1}) + t_1(z_{2,3}) \\
& + a_2(z_{1,1} + z_{1,3} + z_{2,1}) + g_2(z_{1,2}) + t_2(z_{2,2} + z_{2,3}) + g_3(z_{1,1}) \\
& + a_3(z_{1,2} + z_{2,3}) + c_3(z_{1,3}) + t_3(z_{2,1} + z_{2,2}) + a_4(z_{1,1} + z_{2,2}) \\
& + c_4(z_{1,2}) + t_4(z_{1,3} + z_{2,1} + z_{2,3}) + c_5(z_{1,1}) \\
& + t_5(z_{1,2} + z_{2,2} + z_{2,3}) + g_5(z_{1,3}) + a_5(z_{2,1}) \\
& + t_6(z_{1,1} + z_{1,2} + z_{1,3}) + c_6(z_{2,1}) + g_6(z_{2,2} + z_{2,3}) \\
& + t_7(z_{1,1} + z_{1,2}) + g_7(z_{1,3} + z_{2,1} + z_{2,2}) + c_7(z_{2,3}) + t_8(z_{1,1}) \\
& + g_8(z_{1,2} + z_{2,1} + z_{2,3}) + a_8(z_{1,3}) + c_8(z_{2,2}) + g_9(z_{1,1} + z_{2,2}) \\
& + a_9(z_{1,2}) + t_9(z_{1,3} + z_{2,3}) + c_9(z_{2,1}) + a_{10}(z_{1,1}) \\
& + t_{10}(z_{1,2} + z_{2,2}) + c_{10}(z_{1,3} + z_{2,3}) + g_{10}(z_{2,1})
\end{aligned}$$

Sehingga fungsi objektif untuk mengidentifikasi sekuen konsensus dapat ditulis menjadi:

$$\text{skor akhir} = \text{maks} \sum_i^{10} \{ a_i \sum_{(l,s) \in SA_i} z_{l,s} + t_i \sum_{(l,s) \in ST_i} z_{l,s} + c_i \sum_{(l,s) \in SC_i} z_{l,s} + g_i \sum_{(l,s) \in SG_i} z_{l,s} \} \quad (3.2)$$

Dimana $SA_i = \{(l, s) | d_{l,s}^i = A\}$, $ST_i = \{(l, s) | d_{l,s}^i = T\}$, $SC_i = \{(l, s) | d_{l,s}^i = C\}$ dan $SG_i = \{(l, s) | d_{l,s}^i = G\}$, dengan $i = 1, 2, \dots, 10$, $l = 1, 2, \dots, p$ dan $s = 1, 2, \dots, q - 15$.

Fungsi objektif pada (3.2) masih mengandung persamaan nonlinier dimana $q_i \sum z_{l,s}$ untuk $q_i \in \{A, C, T, G\}$. Bentuk perkalian ini dapat dilinierisasikan dengan proposisi berikut.

Proposisi 3.1. Bentuk perkalian $\lambda_i = q_i \sum z_{l,s}$ dengan λ_i untuk memaksimumkan dan $q_i \in \{a_i, c_i, g_i, t_i\}$ serta $z_{l,s}$, $q_i \in \{0, 1\}$ dapat dilinierisasikan sebagai berikut :

$$\lambda_i \geq 0 \quad (3.3)$$

$$\lambda_i \leq \sum z_{l,s} \quad (3.4)$$

$$\lambda_i \leq M q_i \quad (3.5)$$

untuk $i = 1, 2, \dots, 10$, $l = 1, 2, \dots, p$, dan $s = 1, 2, \dots, q - 15$. Dimana M adalah konstanta yang lebih besar atau sama dengan banyaknya sekuen DNA.

Bukti.

Diketahui $q_i, z_{l,s} \in \{0, 1\}$, sehingga untuk $\lambda_i = q_i \sum z_{l,s}$, jika $q_i = 1$ maka $\lambda_i = \sum z_{l,s}$, selain itu $q_i = 0$ mengakibatkan $\lambda_i = 0$. ■

Selanjutnya notasikan :

$$z(a_i) = a_i \sum_{(l,s) \in SA_i} z_{l,s} \quad (3.6)$$

$$z(t_i) = t_i \sum_{(l,s) \in ST_i} z_{l,s} \quad (3.7)$$

$$z(c_i) = c_i \sum_{(l,s) \in SC_i} z_{l,s} \quad (3.8)$$

$$z(g_i) = g_i \sum_{(l,s) \in SG_i} z_{l,s} \quad (3.9)$$

Berdasarkan Proposisi 3.1 substitusi persamaan (3.6), (3.7), (3.8) dan (3.9) ke dalam persamaan (3.3) dan (3.5) pada Proposisi 3.1 menjadi :

$$z(a_i) \leq M a_i \quad (3.10)$$

$$z(t_i) \leq M t_i \quad (3.11)$$

$$z(c_i) \leq M c_i \quad (3.12)$$

$$z(g_i) \leq M g_i \quad (3.13)$$

$$z(a_i), z(t_i), z(c_i), z(g_i) \geq 0 \quad (3.14)$$

Fungsi objektif yang diperoleh adalah :

skor akhir = maks skor_{total}

$$= \text{maks} \sum_{l=1}^p \text{skor}_l = \text{maks} \sum_{i=1}^{10} (z(a_i) + z(t_i) + z(c_i) + z(g_i)) \quad (3.15)$$

Dengan syarat

$$z(a_i) \leq M a_i$$

$$z(t_i) \leq M t_i$$

$$z(c_i) \leq M c_i$$

$$z(g_i) \leq M g_i$$

$$z(a_i), z(t_i), z(c_i), z(g_i) \geq 0$$

Setelah ditentukan variabel keputusan dan fungsi objektif, kemudian tentukan fungsi-fungsi kendala yang diperlukan.

1. Fungsi kendala yang berhubungan dengan posisi basa pada sekuen konsensus. Setiap basa sekuen konsensus L_i hanya dapat ditempati oleh satu basa. Misalkan jika basa sekuen konsensus $L_i = A$ maka $a_i = 1$ dan $c_i = t_i = g_i = 0$. Sehingga akan memberika syarat :

$$a_i + c_i + t_i + g_i = 1 \quad (3.16)$$

Untuk $i = 1, 2, \dots, 10$.

2. Fungsi kendala yang berhubungan dengan *site* pada sekuen DNA untuk menentukan kandidat sekuen konsensus yang akan digunakan. Dalam menentukan sekuen konsensus akan dihitung skor dari tiap-tiap sekuen DNA. Setiap sekuen DNA hanya akan mempunyai satu sekuen konsensus, sehingga untuk menghasilkan $skor_l$ pada setiap sekuen ke l hanya satu *site* yang dipilih dan tidak ada *site* lain selain *site* tersebut yang digunakan.

$$skor_l = \sum_{s=1}^{q-15} z_{l,s} (\theta_{l,s}^1 + \theta_{l,s}^2 + \dots + \theta_{l,s}^{10}) \quad (3.17)$$

Dimana

$$\theta_{l,s}^i = \begin{cases} a_i, & \text{jika } d_{l,s}^i = A \\ t_i, & \text{jika } d_{l,s}^i = T \\ c_i, & \text{jika } d_{l,s}^i = C \\ g_i, & \text{jika } d_{l,s}^i = G \end{cases}$$

Misalkan *site* ke k dipilih sehingga $z_{l,k} = 1$ dan $z_{l,s} = 0$ untuk $k \in \{1, 2, \dots, q - 15\}$ dan $s \in \{1, 2, \dots, q - 15\}$ serta $s \neq k$. Sehingga menghasilkan kendala :

$$\begin{aligned} \sum_{s=1}^{q-15} z_{l,s} &= 1 \\ z_{l,s} &\in \{0,1\} \end{aligned} \quad (3.18)$$

Untuk $l = 1, 2, \dots, p$ dan $s = 1, 2, \dots, q - 15$, juga mengakibatkan $0 \leq skor_l \leq 10$.

Dari variabel keputusan, fungsi kendala dan fungsi objektif di persamaan (3.10) – (3.15), (3.16) dan (3.18), diperoleh model pencarian sekuen konsensus yang berupa pemrograman linier, yaitu :

Fungsi objektif

$$skor\ total = maks \sum_{l=1}^p skor_l = \sum_{i=1}^{10} (z(a_i) + z(t_i) + z(c_i) + z(g_i))$$

Dengan syarat

$$z(a_i) \leq Ma_i$$

$$z(t_i) \leq Mt_i$$

$$z(c_i) \leq Mc_i$$

$$z(g_i) \leq Mg_i$$

$$z(a_i), z(t_i), z(c_i), z(g_i) \geq 0$$

$$\sum_{s=1}^{q-15} z_{l,s} = 1$$

$$z_{l,s} \in \{0, 1\}$$

untuk $l = 1, 2, \dots, p$ dan $s = 1, \dots, q - 15$

$$a_i + t_i + c_i + g_i = 1$$

$$a_i, t_i, c_i, g_i \in \{0, 1\}$$

untuk $i = 1, 2, \dots, 10$

Nilai optimal diperoleh ketika fungsi objektif bernilai maksimum. Variabel-variabel keputusan yang memaksimalkan fungsi objektif merupakan solusi dari pemrograman linier. Sekuen konsensus ditentukan dari nilai variabel keputusan yang didapatkan.

3. 3 Penyelesaian Masalah

Masalah pemrograman linier dapat dapat diselesaikan dengan beberapa metode, salah satunya metode simpleks. Berdasarkan contoh sebelumnya, akan dicari sekuen konsensus menggunakan pemrograman linier dan akan diselesaikan dengan metode simpleks.

Dengan menggunakan sekuen DNA pada Contoh 1, yaitu:

AAGACTGTTTTTTTGATC

GATTATTTGCACGGCGTC

Kandidat sekuen konsensus tersebut diberikan dalam matriks D' .

$$D' = \begin{bmatrix} \text{AAGACTTTGA} & \text{AGACTTTGAT} & \text{GACTGTGATC} \\ \text{GATTACGGCG} & \text{ATTATGGCGT} & \text{TTATTGCGTC} \end{bmatrix}$$

Kemudian bentuklah persamaan pemrograman linier berdasarkan nilai $z_{l,s}$ pada masing-masing sekuen DNA.

Untuk $z_{1,1} = 1$ dan $z_{2,1} = 1$,

AAGACTGTTTTTTTGATC

GATTATTTGCACGGCGTC

$l = 1$ dan $s = 1$ AAGACTTTGA

$l = 2$ dan $s = 1$ GATTACGGCG

Ubahlah sekuen diatas menjadi model pemrograman linier.

$$\begin{aligned} \text{Maks } \quad \text{skor} &= a_1 + a_2 + g_3 + a_4 + c_5 + t_6 + t_7 + t_8 + g_9 + a_{10} + g_1 + \\ & a_2 + t_3 + t_4 + a_5 + c_6 + g_7 + g_8 + c_9 + g_{10} \\ & a_i + c_i + t_i + g_i = 1 \\ & a_i, g_i, t_i, c_i \in \{0,1\} \\ & \text{untuk } i = 1,2, \dots, 10 \end{aligned}$$

Tahap 1, bentuk masalah pemrograman linier kedalam bentuk standar dengan menambahkan sejumlah m variabel artifisial. Karena fungsi kendala berbentuk sama dengan, maka variabel artifisial mempunyai koefisien $-M$.

$$\begin{aligned} \text{Maks } \quad \text{skor} &= a_1 + a_2 + g_3 + a_4 + c_5 + t_6 + t_7 + t_8 + g_9 + a_{10} + g_1 + a_2 + \\ & t_3 + t_4 + a_5 + c_6 + g_7 + g_8 + c_9 + g_{10} - Mx_1 - Mx_2 - \\ & Mx_3 - Mx_4 - Mx_5 - Mx_6 - Mx_7 - Mx_8 - Mx_9 - Mx_{10} \\ & a_i + c_i + t_i + g_i - Mx_i = 1 \\ & a_i, c_i, t_i, g_i \in \{0,1\} \\ & \text{untuk } i = 1,2, \dots, 10 \end{aligned}$$

Tahap 2, terdapat 10 kendala sehingga $m = 10$, gunakan variabel *slack* $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9$ dan x_{10} untuk membentuk basis awal solusi layak, sedangkan variabel yang bukan dasar bernilai nol. Sehingga $x_1 = 1, x_2 = 1, x_3 = 1, x_4 = 1, x_5 = 1, x_6 = 1, x_7 = 1, x_8 = 1, x_9 = 1$ dan $x_{10} = 1$. Tabel dapat dilihat di Tabel Lampiran 1.1.

Tahap 3, tentukan variabel yang akan masuk basis. Variabel yang masuk adalah variabel dengan $c_j - z_j$ positif terbesar. Jika tidak ada, solusi sudah optimal. Kolom dari variabel yang akan masuk disebut kolom pivot.

Tahap 4, tentukan variabel yang akan keluar basis. Misalkan θ_j telah dipilih sebagai variabel yang masuk. Variabel dengan $\min b_i/a_{ij}$ dimana b_i adalah konstanta di sebelah kanan fungsi kendala dan a_{ij} adalah koefisien pada kolom θ_j , akan keluar dari basis dan baris ini disebut baris pivot. b_i/a_{ij} untuk suatu i dipertimbangkan jika a_{ij} positif. Perpotongan antara kolom pivot dan baris pivot disebut elemen pivot.

Tahap 5, lakukan sejumlah operasi baris yang mengeliminasi θ^* (variabel masuk) dari seluruh baris kecuali baris pivot, dengan cara membagi baris pivot

dengan elemen pivot, diperoleh baris baru pivot. Untuk baris lainnya (bukan pivot), berlaku:

Baris ke- i baru = baris lama – a_{ij} (baris baru pivot).

Tabel simpleks pada Tabel Lampiran 1.1 setelah mengalami *pivoting* dapat dilihat pada Tabel Lampiran 1.2.

Tahap 6, kembali ke tahap 3, ulangi proses yang telah dilakukan sebelumnya. Lakukan sampai solusi benar-benar optimal. Setelah 10 iterasi, hasil akhir simpleks ditunjukkan pada Tabel Lampiran 1.3. Tabel akhir simpleks menunjukkan bahwa untuk $z_{1,1} = 1$ dan $z_{2,1} = 1$ menghasilkan solusi optimal dengan nilai fungsi objektif 11 dan nilai $a_2 = 2$, serta $a_1, t_3, t_4, c_5, t_6, g_7, g_8, c_9, a_{10} = 1$.

Dengan mengulangi proses diatas untuk $z_{1,1} = 1$ dan $z_{2,2} = 1, z_{1,1} = 1$ dan $z_{2,3} = 1, z_{1,2} = 1$ dan $z_{2,1} = 1, z_{1,2} = 1$ dan $z_{2,2} = 1, z_{1,2} = 1$ dan $z_{2,3} = 1, z_{1,3} = 1$ dan $z_{2,1} = 1, z_{1,3} = 1$ dan $z_{2,2} = 1$, serta $z_{1,3} = 1$ dan $z_{2,3} = 1$, menghasilkan tabel akhir simpleks yang ditunjukkan dalam Tabel Lampiran 1.4 sampai Tabel Lampiran 1.11. Hasilnya diringkas pada tabel berikut :

Tabel 3.1 ringkasan nilai fungsi objektif tabel akhir simpleks

	$z_{1,1}$	$z_{1,2}$	$z_{1,3}$
$z_{2,1}$	11	11	14
$z_{2,2}$	13	13	11
$z_{2,3}$	10	13	13

Berdasarkan Tabel 3.1, nilai fungsi objektif yang paling maksimum diperoleh ketika menggunakan $z_{1,3} = 1$ dan $z_{2,1} = 1$, dengan nilai fungsi objektif 14 dan $g_1 = a_2 = t_4 = g_7 = 2$ serta $c_3 = g_5 = t_6 = g_8 = c_9 = g_{10} = 1$. Sehingga dapat disimpulkan bahwa sekuen konsensus dari sekuen DNA pada Contoh 1 adalah GATTATGATC. Pada sekuen DNA pertama sekuen konsensus terdapat di posisi $s = 3$ dan pada sekuen DNA kedua, sekuen konsensus terdapat di posisi $s = 1$.

BAB 4

IMPLEMENTASI PROGRAM

Dalam memecahkan masalah pencarian sekuen konsensus dengan pemrograman linier, penyelesaian secara manual tidak efisien, karena memerlukan penghitungan yang banyak dan waktu yang cukup lama. Karena itu, menyelesaikan masalah ini dapat dilakukan dengan program yang dibuat. Pada bab ini akan diberikan simulasi penyelesaian masalah pencarian sekuen konsensus menggunakan program MATLAB versi 5.3.1 dan komputer berprosesor intel atom N270 @1,60 GHz RAM 1,00 GB, *listing* program diberikan pada Lampiran 3. Di Bab 3, telah diberikan contoh pencarian sekuen secara manual menggunakan sekuen DNA pada Contoh 1. Pada Subbab 4.1 akan diberikan simulasi program untuk menyelesaikan masalah yang sama. Selanjutnya akan dilakukan analisa terhadap program yang akan dijelaskan pada Subbab 4.2.

Program MATLAB untuk mencari sekuen konsensus dijalankan dengan algoritma :

Tahap 1 Masukkan input berupa sekuen DNA.

Tahap 2 Tentukan kandidat-kandidat sekuen konsensus tanpa gen promotor dari setiap sekuen.

Tahap 3 Bentuk matriks yang berisi kandidat-kandidat sekuen konsensus, matriks berukuran $i \times j$, dimana i banyaknya sekuen dan j banyaknya kandidat pada tiap sekuen.

Tahap 4 Ambil satu kandidat dari setiap sekuen, bentuk fungsi objektifnya dan maksimumkan nilai fungsi objektif.

Tahap 5 bandingkan semua hasil yang diperoleh, dan pilih nilai yang paling maksimum.

Tahap 6 Tentukan sekuen konsensus berdasarkan nilai maksimum yang diperoleh.

Algoritma ini diterapkan pada *pseudocode* berikut

```
% Tahap 1
input masukkan banyaknya sekuen
p = banyaknya sekuen DNA
% Tahap 2
```

```

for j=1:p
    Input masukkan sekuen ke -j
    hitung banyaknya basa tiapa sekuen ke-j
    kolom(j)=banyaknya basa sekuen ke-j
    k(j)=kolom(j)-15
end for
% Tahap 3
for j=1:p
    for i=1:k(j)
        x{i,j}= 5 basa awal kandidat sekuen konsensus
        y(i,j)= 5 basa akhir kandidat sekuen konsensus
        Bentuk matriks yang berisi kandidat-kandidat sekuen konsensus
        D{i,j}=[x{i,j} y(i,j)]
    end for
end for
definisikan
hmax=-9999
% Tahap 4
for z1=1:i
    for z2=1:i
        for z3=1:i
            bentuk fungsi objektif
            f= [(1,z1) (2,z2) (3,z3)]
            pilih basa di tiap-tiap posisi yang memberikan kontribusi terbesar pada
            fungsi objektif
            evaluasi nilai fungsi objektif
            h = nilai fungsi objektif
            % Tahap 5
            if h>=hmaks
                hmaks=h
            else
                hmaks=hmaks
            end if
        end for
    end for
end for
end for
% Tahap 6
hmaks, sekuen konsensus dan posisinya

```

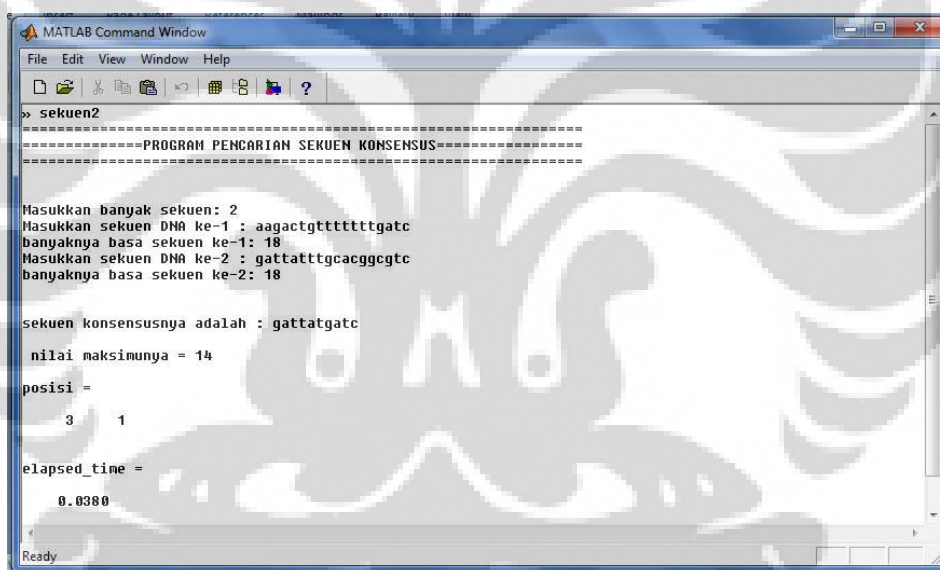
4.1 Simulasi Program

Untuk simulasi awal digunakan data *dummy* pada Contoh 1 di Bab 3 yang berupa 2 sekuen DNA dengan masing-masing 18 basa yaitu :

AAGACTGTTTTTTTGATC
GATTATTTGCACGGCGTC

Kemudian akan dicari sekuen konsensusnya menggunakan program yang telah dibuat.

Untuk menjalankan program ini, buka terlebih dahulu software MATLAB, kemudian ketik “sekuen2” pada *command window*. Tentukan banyaknya sekuen DNA serta masukkan sekuen DNA. Setelah semua sekuen DNA dimasukkan, tekan enter untuk melihat hasilnya.



```

MATLAB Command Window
File Edit View Window Help
-----PROGRAM PENCARIAN SEKUEN KONSENSUS-----
>> sekuen2
-----PROGRAM PENCARIAN SEKUEN KONSENSUS-----
Masukkan banyak sekuen: 2
Masukkan sekuen DNA ke-1 : aagactgttttttgatc
banyaknya basa sekuen ke-1: 18
Masukkan sekuen DNA ke-2 : gattatgttcacggcgtc
banyaknya basa sekuen ke-2: 18

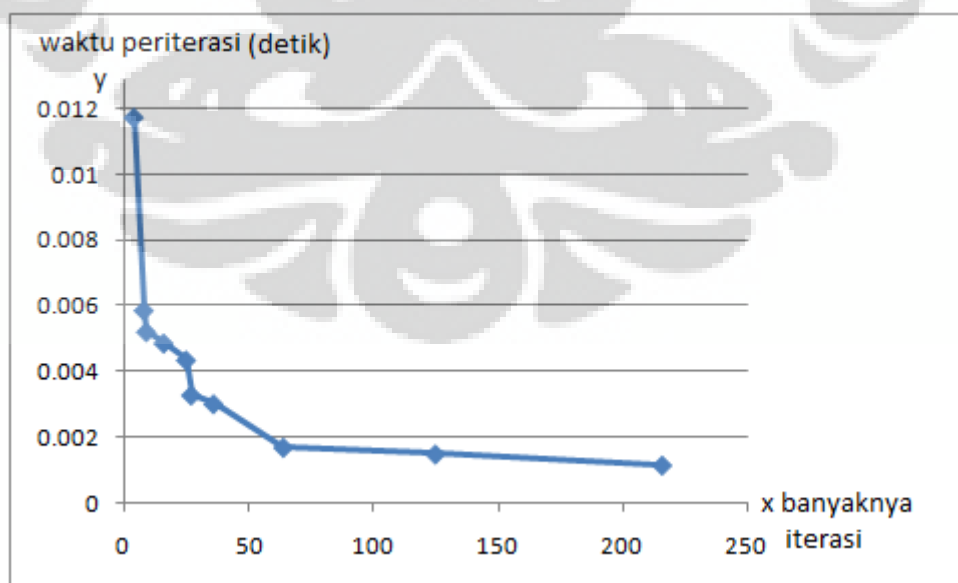
sekuen konsensusnya adalah : gattatgac
nilai maksimumnya = 14
posisi =
     3     1
elapsed_time =
     0.0380
Ready
  
```

Gambar 4.1 Tampilan masukan dan keluaran untuk dua sekuen DNA

Hasil pada Gambar 4.1 menunjukkan bahwa sekuen konsensus yang diperoleh adalah GATTATGATC dengan nilai maksimum 14 serta terletak pada posisi 3 di sekuen pertama dan berada di posisi 1 untuk sekuen kedua. Waktu yang diperlukan untuk memperoleh hasilnya adalah 0,038 detik.

4.2 Analisa Kompleksitas

Di subbab sebelumnya telah dilakukan simulasi program dengan dua sekuen DNA yang masing-masing mempunyai 18 basa. *Running* program ini memerlukan waktu 0,038 detik. Sekuen DNA tersebut merupakan data *dummy* yang digunakan untuk percobaan. Pada subbab ini akan dianalisa *running* program menggunakan data *real* berupa sekuen DNA bakteri *E.Coli*. Data ini terdiri dari 18 sekuen yang masing-masing mempunyai 105 basa. Daftar sekuen DNA dapat dilihat di Lampiran 2. Selanjutnya akan dicari estimasi waktu untuk mendapatkan sekuen konsensus dari sekuen DNA *E.Coli* tersebut. Hal pertama yang akan dilakukan adalah menghitung waktu periterasi berdasarkan banyaknya iterasi. Satu iterasi di hitung sebagai sekali proses penghitungan nilai optimal pemrograman linier dari kombinasi setiap satu kandidat dari masing-masing sekuen DNA yang ada. Proses ini akan dilakukan terus sampai semua kandidat habis dikombinasikan. Sehingga banyaknya iterasi dalam satu kali *running* untuk memperoleh sekuen konsensus adalah $(q - 15)^p$, dengan p adalah banyaknya sekuen DNA dan q adalah banyaknya basa pada tiap sekuen DNA. Waktu rata-rata periterasi dihitung dengan menghitung total waktu yang dibutuhkan untuk sekali mendapatkan sekuen konsensus kemudian dibagi dengan total banyaknya iterasi.



Gambar 4.2 Grafik waktu periterasi terhadap banyaknya iterasi

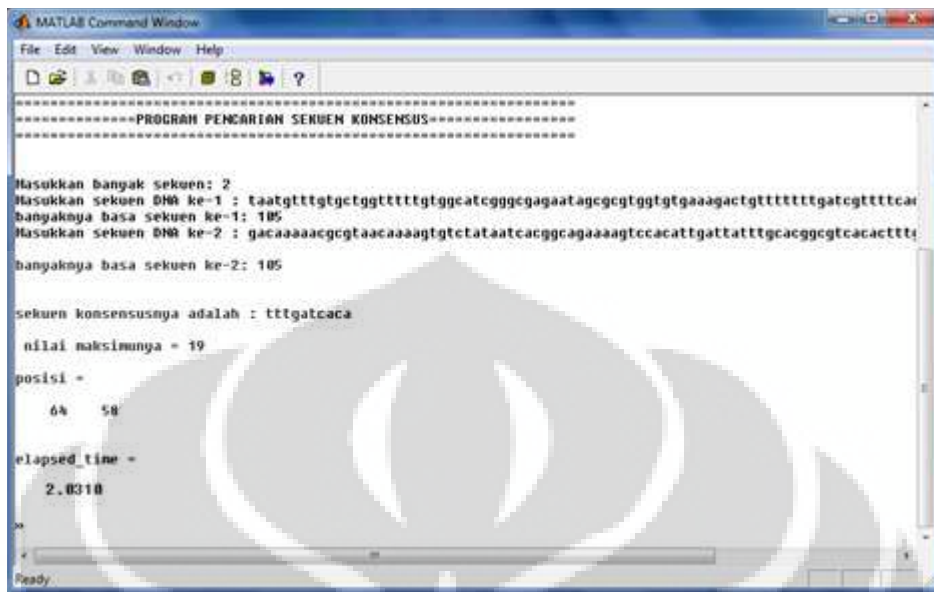
Sumbu x pada Gambar 4.2 menyatakan banyaknya iterasi dan sumbu y menyatakan waktu rata-rata yang diperlukan untuk melakukan sekali iterasi jika iterasi sebanyak x dilakukan. Berdasarkan Gambar 4.2 waktu rata-rata yang dibutuhkan untuk sekali iterasi tidak lebih dari 0,012 detik. Selanjutnya untuk menghitung waktu yang dibutuhkan untuk *running* program dengan data *real* sekuen DNA *E.Coli* pada Lampiran 2, diasumsikan jika waktu rata-rata sekali iterasi adalah 0,012 detik.

Tabel 4.1 Estimasi waktu *running* program

p	q	$q-15$	Total iterasi	Estimasi waktu yang dibutuhkan jika menggunakan p sekuen				
				detik	menit	jam	hari	bulan
2	105	90	8100	2,7				
3	105	90	729000	243	4,05			
4	105	90	$6,56 \times 10^7$	21870	364,5	6,075		
5	105	90	$5,9 \times 10^9$	1968300	32805	546,75	22,78125	
6	105	90	$5,34 \times 10^{11}$	$1,77 \times 10^8$	2952450	49207,5	2050,31	68,34375
7	105	90	$4,78 \times 10^{13}$	$1,59 \times 10^{10}$	$2,66 \times 10^8$	4428675	184528	6150,938
8	105	90	$4,3 \times 10^{15}$	$1,43 \times 10^{12}$	$2,39 \times 10^{10}$	$3,99 \times 10^8$	$1,66 \times 10^7$	$5,5 \times 10^5$
9	105	90	$3,87 \times 10^{17}$	$1,29 \times 10^{14}$	$2,15 \times 10^{12}$	$3,59 \times 10^{10}$	$1,49 \times 10^9$	$4,9 \times 10^7$
10	105	90	$3,49 \times 10^{19}$	$1,16 \times 10^{16}$	$1,94 \times 10^{14}$	$3,23 \times 10^{12}$	$1,35 \times 10^{11}$	$4,4 \times 10^9$
11	105	90	$3,14 \times 10^{21}$	$1,05 \times 10^{18}$	$1,74 \times 10^{16}$	$2,91 \times 10^{14}$	$1,21 \times 10^{13}$	$4,04 \times 10^{11}$
12	105	90	$2,82 \times 10^{23}$	$9,41 \times 10^{19}$	$1,57 \times 10^{18}$	$2,62 \times 10^{16}$	$1,09 \times 10^{15}$	$3,63 \times 10^{13}$
13	105	90	$2,54 \times 10^{25}$	$8,47 \times 10^{21}$	$1,41 \times 10^{20}$	$2,35 \times 10^{18}$	$9,81 \times 10^{16}$	$3,27 \times 10^{15}$
14	105	90	$2,29 \times 10^{27}$	$7,63 \times 10^{23}$	$1,27 \times 10^{22}$	$2,12 \times 10^{20}$	$8,83 \times 10^{18}$	$2,94 \times 10^{17}$
15	105	90	$2,06 \times 10^{29}$	$6,86 \times 10^{25}$	$1,14 \times 10^{24}$	$1,91 \times 10^{22}$	$7,94 \times 10^{20}$	$2,65 \times 10^{19}$
16	105	90	$1,85 \times 10^{31}$	$6,18 \times 10^{27}$	$1,03 \times 10^{26}$	$1,72 \times 10^{24}$	$7,15 \times 10^{22}$	$2,38 \times 10^{21}$
17	105	90	$1,67 \times 10^{33}$	$5,56 \times 10^{29}$	$9,27 \times 10^{27}$	$1,54 \times 10^{26}$	$6,43 \times 10^{24}$	$2,14 \times 10^{23}$
18	105	90	$1,5 \times 10^{35}$	5×10^{31}	$8,34 \times 10^{29}$	$1,39 \times 10^{28}$	$5,79 \times 10^{26}$	$1,93 \times 10^{25}$

Pada Tabel 4.1 dapat dilihat hasil simulasi waktu yang dibutuhkan untuk *running* program apabila kita menggunakan 2 sampai 18 sekuen DNA *E.Coli*. Untuk menyelesaikan masalah pencarian sekuen konsensus menggunakan data yang semakin besar ternyata memerlukan waktu yang semakin lama. Berdasarkan hasil simulasi pada Tabel 4.1, jumlah sekuen yang memungkinkan masih bisa dieksekusi program ini adalah maksimum 5 sekuen DNA dalam sekali *running*.

Berikut ini sebagai contoh akan dicoba *running* program untuk 2 sekuen dan 3 sekuen DNA *E.Coli* dengan waktu *running* tidak terlalu lama.



```

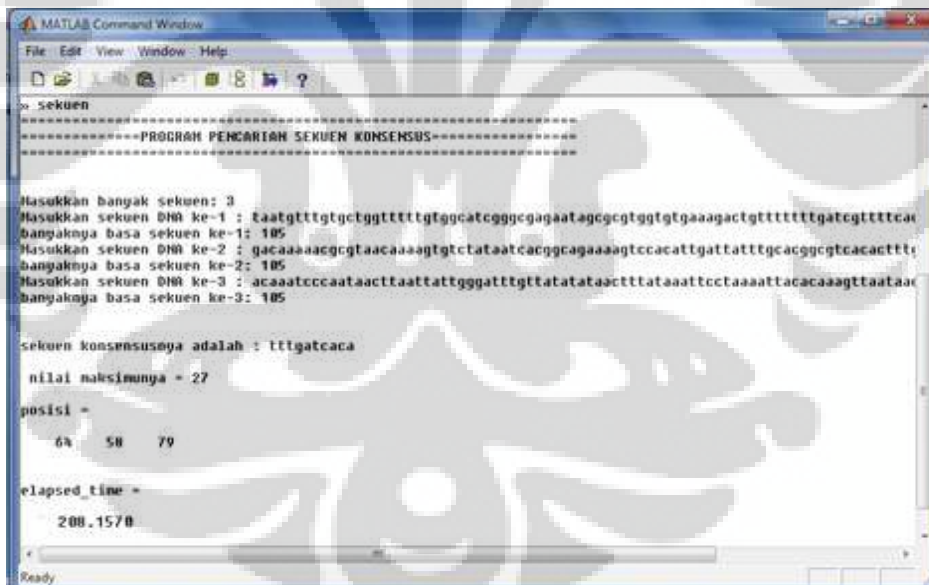
MATLAB Command Window
File Edit View Window Help
-----PROGRAM PENCARIAN SEKUEN KONSENSUS-----
Masukkan banyak sekuen: 2
Masukkan sekuen DNA ke-1 : taatgtttgtcgtggtttttgtgcatcggggcagaatagcgcgtgggtgaaagactgtttttgatcgttttca
Banyaknya basa sekuen ke-1: 105
Masukkan sekuen DNA ke-2 : gacaaaaacgcgtaacaaaagtgtctataatcagggcagaaaaagtcacattgattatttgcacggcgtcacacttt
Banyaknya basa sekuen ke-2: 105

sekuen konsensusnya adalah : tttgatcaca
nilai maksimumnya = 19
posisi =
    64    58

elapsed_time =
    2.0310

```

Gambar 4.3 Tampilan masukkan dan keluaran untuk 2 sekuen DNA *E.coli*



```

MATLAB Command Window
File Edit View Window Help
-----PROGRAM PENCARIAN SEKUEN KONSENSUS-----
>> sekuen
Masukkan banyak sekuen: 3
Masukkan sekuen DNA ke-1 : taatgtttgtcgtggtttttgtgcatcggggcagaatagcgcgtgggtgaaagactgtttttgatcgttttca
Banyaknya basa sekuen ke-1: 105
Masukkan sekuen DNA ke-2 : gacaaaaacgcgtaacaaaagtgtctataatcagggcagaaaaagtcacattgattatttgcacggcgtcacacttt
Banyaknya basa sekuen ke-2: 105
Masukkan sekuen DNA ke-3 : acaaatcccaataacttaattattgggatttggfataataactttataaattcctaaaattacacaaagttaataa
Banyaknya basa sekuen ke-3: 105

sekuen konsensusnya adalah : tttgatcaca
nilai maksimumnya = 27
posisi =
    64    58    79

elapsed_time =
    208.1570

```

Gambar 4.4 Tampilan masukkan dan keluaran untuk 3 sekuen DNA *E.coli*

Berdasarkan Gambar 4.3 dan Gambar 4.4, dengan menggunakan 2 sekuen dan 3 sekuen DNA diperoleh sekuen konsensus yang sama yaitu TTTGATCACA dan dengan posisi yang sama yaitu terletak di posisi 64 untuk sekuen pertama, di posisi 58 untuk sekuen kedua dan di posisi 79 untuk sekuen ketiga. Waktu yang

diperlukan untuk *running* adalah 2,0310 detik untuk 2 sekuen dan 208,1570 detik untuk 3 sekuen. Waktu ini tidak berbeda jauh dari estimasi waktu yang telah diberikan pada Tabel 4.2 yaitu 2,7 detik untuk 2 sekuen dan 243 detik untuk 3 sekuen. Sementara itu berdasarkan hasil estimasi waktu pada Tabel 4.2, untuk mencari sekuen konsensus dengan menggunakan 18 sekuen diperlukan waktu yang sangat lama sehingga tidak dilakukan dalam skripsi ini.



BAB 5

PENUTUP

5.1 Kesimpulan

Dalam skripsi ini pencarian sekuen konsensus dapat dicari menggunakan pemrograman linier. Sekuen konsensus diperoleh ketika nilai fungsi objektif bernilai maksimum. Letak sekuen konsensus diketahui melalui nilai $z_{l,s}$ yang digunakan pada tiap sekuen DNA untuk memaksimumkan nilai fungsi objektif. Serta sekuen konsensus ditentukan melalui nilai a_i, c_i, t_i dan g_i untuk $i = 1, 2, \dots, 10$ yang mengakibatkan nilai fungsi objektif menjadi maksimum.

Pencarian sekuen konsensus ini dapat dilakukan menggunakan program. Dengan menggunakan sekuen DNA pada Bab 3, cara manual maupun menggunakan program memberikan hasil yang sama yaitu diperoleh sekuen konsensus GATTATGATC dengan posisi $s = 3$ untuk sekuen pertama dan $s = 1$ untuk sekuen kedua. Penyelesaian menggunakan program ini memerlukan waktu 0,038 detik. Ketika menggunakan data *real* berupa sekuen DNA *E.Coli*, untuk 2 sekuen DNA, waktu *running* cukup singkat yaitu 2,7 detik. Namun ketika menggunakan 18 sekuen DNA, estimasi waktu *running* yang diperlukan sangat lama yaitu sekitar 5×10^{31} detik. Berdasarkan estimasi waktu pada Tabel 4.1, program ini hanya mampu melakukan *running* sampai 5 sekuen DNA *E.Coli*.

5.2 Saran

Dalam dunia bioinformatika umumnya bekerja dengan data sekuen DNA yang sangat banyak. Penyelesaian masalah sekuen konsensus dengan pemrograman linier untuk data yang besar akan memerlukan waktu *running* yang sangat lama. Ini merupakan kelemahan dari program tersebut karena berjalan secara serial. Salah satu solusi yang bisa digunakan adalah menggunakan teknik komputasi tingkat lanjut berbasis komputasi paralel. Untuk penelitian selanjutnya disarankan menggunakan komputasi paralel sehingga waktu *running* dapat dipersingkat.

DAFTAR PUSTAKA

- Brazma, Alvis, et al.(1998).Approaches to the Automatic Discovery of Patterns in Biosequences.*Journal of Computational Biology* 5, 279-305.
- Bronsosn, Richard.(1998).*Teori dan Soal-soal Operation Research*.(Drs. Hans J. Wospakrik, penerjemah).Jakarta:Erlangga.
- Ecker, J.G. et al.(2002).An Aplication of Non-linear Optimization in Molecular Biology.*Europe Journal of Operational Research* 138, 452-458.
- Han-Lin Li & Chang-Jui Fu.(2005).*A Linear Programming Approach for Identifying a Consensus sequence on DNA Sequences*.Oxford University Press 21, 1838-1845.
- Schrage.(1999).*Optimization Modeling with Lingo*.Chicago:LINDO System Inc.
- Scheider, Thomas.D.(2002).Consensus Sequence Zen.*National Cancer Institute* 1, 111-119.
- Wilf, Herbert, S.*Algorithms and Complexity*.1994. 10 Juni 2012.
www.math.upenn.edu/~Wilf/AlgoComp.pdf
- Winston, Wayne.(1995).*Introduction to Mathematical Programming*. California:Duxbury Press.
- Wu, Nesa & Coppins, Richard.(1981).*Linear Programming and Extensions*.USA:McGraw-Hill.
- Yi-Ping.(1998).*Bioinformatics Technologies*.Australia:Springer.

Lampiran 1. Tabel Simpleks

Tabel Lampiran 1.1. Tabel awal simpleks untuk $Z_{1,1} = 1$ dan $Z_{2,1} = 1$

Maks	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
b_i	b_1	b_2	b_4	b_5	b_{10}	c_5	c_9	c_3	c_4	c_7	c_8	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	b_i	b_i/a_{ij}									
c_i	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	1	-	-						
x_1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	-	-						
x_2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	-	-						
x_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	-	-	-						
x_4	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	-	-	-							
x_5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	-	-	-	-						
x_6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	-	-	-						
x_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	-	-	-						
x_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	-	-	-					
x_9	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	-	-	-						
x_{10}	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	-	-	-	-						
z_j	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	
$c_j - z_j$	$1+M(2+M)$	$1+M$	$-M$	$-M$	$-M$	$1+M$	$-M$	$-M$	$1+M$	$-M$	$-M$	$1+M$	$1+M$	$-M$	$1+M$	$-M$	$1+M$	$1+M$	$1+M$	$1+M$	$1+M$	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

(lanjutan)

Tabel Lampiran 1.3 Tabel akhir simpleks untuk $z_{1,1} = 1$ dan $z_{2,1} = 1$

Maks	θ_1	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	b_i				
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1		
2	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
3	1	f_3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
4	1	t_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
5	1	c_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
6	1	t_6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
7	1	e_7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
8	1	e_8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
9	1	c_9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
10	1	a_{10}	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
z_j																																				
c_j-z_j																																				

(lanjutan)

Tabel Lampiran 1.4. Tabel akhir simpleks untuk $z_{1,1} = 1$ dan $z_{2,2} = 1$

i	Maks		a_1	a_2	a_3	a_4	a_{10}	c_5	c_8	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	g_5	g_6	g_7	g_8	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	b_i		
	2	1																																		
1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
zj		2	1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	2	1	1	1	1	1	2	1	13
cj-zj		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-M-2	-M-1	-M-1	-M-1	-M-2	-M-1	-M-1	-M-1	-M-1	-M-2	-M-1	-M-1	

(lanjutan)

Tabel Lampiran 1.5. Tabel akhir simpleks untuk $z_{1,1} = 1$ dan $z_{2,3} = 1$

i	Maks	ci	θi	a1	a2	a3	a4	a10	c7	c10	t1	t2	t4	t5	t6	t7	t8	t9	g5	g6	g8	g9	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M	b _i
				a1	a2	a3	a4	a10	c7	c10	t1	t2	t4	t5	t6	t7	t8	t9	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	b _i				
1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
2	1	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
3	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
zj		1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10
cj-zj		0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-M-1	-M-1	-M-1	-M-1	-M-1	-M-1	-M-1	-M-1	-M-1	-M-1

(lanjutan)

Tabel Lampiran 1.7. Tabel akhir simpleks untuk $z_{1,2} = 1$ dan $z_{2,2} = 1$

i	Maks		c _i	θ _i	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	a ₉	a ₁₀	b _i
	z ₁	z ₂													
1	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
z _j		2	1	1	1	1	1	1	1	1	1	1	1	1	1
c _j -z _j		0	0	0	0	0	0	0	0	0	0	0	0	0	0

(lanjutan)

Tabel Lampiran 1.9. Tabel akhir simpleks untuk $z_{1,3} = 1$ dan $z_{2,1} = 1$

i	C _i	Maks	-										g ₁₀	g ₈	g ₇	g ₅	g ₁	t ₉	t ₆	t ₄	t ₃	t ₂	t ₁	1	1	1	1	1	1	-										b _i
			x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀																		x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀	
1	2	g ₁	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
2	2	a ₂	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
3	1	f ₃	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
4	2	t ₄	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
5	1	a ₅	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0			
6	1	t ₆	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
7	2	g ₇	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
8	1	a ₈	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
9	1	f ₉	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
10	1	c ₁₀	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0			
		z _j	2	1	1	1	1	1	1	1	1	2	2	2	2	1	2	1	1	2	1	1	2	1	2	1	1	1	1	1	1	1	1	1	1	1	14			
		c _j -z _j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-M-1			

(lanjutan)

Tabel Lampiran 1.10. Tabel akhir simpleks untuk $z_{1,3} = 1$ dan $z_{2,2} = 1$

i	Maks										x ₁₀	-M												
	θ _i	a ₁	a ₂	a ₃	a ₄	a ₅	a ₆	a ₇	a ₈	a ₉			a ₁₀	x ₁	x ₂	x ₃	x ₄	x ₅	x ₆	x ₇	x ₈	x ₉	x ₁₀	-M
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
z _j	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	II
c _j -z _j	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-M-1

(lanjutan)

Tabel Lampiran 1.11. Tabel akhir simpleks untuk $z_{1,3} = 1$ dan $z_{2,3} = 1$

Maks	1	1	1	1	2	1	1	2	1	1	1	1	1	1	-M	-M	-M	-M	-M	-M	-M	-M	-M	-M							
i	c_i	θ_i	a_1	a_2	a_3	c_3	c_7	c_{10}	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	b_i			
1	1	t_1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1		
2	1	a_2	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	
3	1	c_3	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	
4	2	t_4	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	
5	1	t_5	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	
6	1	t_6	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	
7	1	c_7	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	
8	1	g_8	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	1	
9	2	t_9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	
10	2	c_{10}	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
z_j			1	1	1	1	1	2	1	1	2	1	1	1	1	1	1	1	1	1	1	2	1	1	1	1	2	2	13		
$c_j - z_j$			0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-M-1	-M-1	-M-1	-M-2	-M-1	-M-1	-M-1	-M-1	-M-1	-M-2	-M-2	-M-2		

Lampiran 2. Daftar sekuen DNA *E.coli*

- TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATAGCGCG
TGGTGTGAAAGACTGTTTTTTTGATCGTTTTTCACAAAATGGAA
GTCCACAGTCTTGACAG
- GACAAAACGCGTAACAAAAGTGTCTATAATCACGGCAGAAAA
GTCCACATTGATTATTTGCACGGCGTCACACTTTGCTATGCCAT
AGCATTTTTATCCATAAG
- ACAAATCCCAATAACTTAATTATTGGGATTTGTTATATATAACT
TTATAAATTCCTAAAATTACACAAAGTTAATAACTGTGAGCATG
GTCATATTTTTATCAAT
- CACAAAGCGAAAGCTATGCTAAAACAGTCAGGATGCTACAGTA
ATACATTGATGTACTGCATGTATGCAAAGGACGTCACATTACCG
TGCAGTACAGTTGATAGC
- ACGGTGCTACACTTGTATGTAGCGCATCTTTCTTTACGGTCAAT
CAGCAAGGTGTTAAATTGATCACGTTTTAGACCATTTTTTCGTC
GTGAAACTAAAAAACC
- AGTGAATTATTTGAACCAGATCGCATTACAGTGATGCAAACCTG
TAAGTAGATTTCCCTTAATTGTGATGTGTATCGAAGTGTGTTGCG
GAGTAGATGTTAGAATA
- GCGCATAAAAAACGGCTAAATTCTTGTGTAAACGATTCCACTAA
TTTATTCCATGTCACACTTTTCGCATCTTTGTTATGCTATGGTTA
TTTCATAACCATAAGCC
- GCTCCGGCGGGGTTTTTTGTTATCTGCAATTCAGTACAAAACGT
GATCAACCCCTCAATTTTCCCTTTGCTGAAAAATTTTCCATTGTC
TCCCCTGTAAAGCTGT
- AACGCAATTAATGTGAGTTAGCTCACTCATTAGGCACCCCAGGC
TTTACACTTTATGCTTCCGGCTCGTATGTTGTGTGGAATTGTGAG
CGGATAACAATTTAC
- ACATTACCGCCAATTCTGTAACAGAGATCACACAAAGCGACGG
TGGGGCGTAGGGGCAAGGAGGATGGAAAGAGGTTGCCGTATAA
AGAAACTAGAGTCCGTTTA

- GGAGGAGGCGGGAGGATGAGAACACGGCTTCTGTGAACTAAAC
CGAGGTCATGTAAGGAATTTCTGTATGTTGCTTGCAAAAATCGT
GGCGATTTTATGTGCGCA
- GATCAGCGTCGTTTTAGGTGAGTTGTTAATAAAGATTTGGAATT
GTGACACAGTGCAAATTCAGACACATAAAAAAACGTCATCGCT
TGCATTAGAAAGGTTTCT
- GCTGACAAAAAAGATTAACATACCTTATACAAGACTTTTTTTT
CATATGCCTGACGGAGTTCACACTTGTAAGTTTTCAACTACGTT
GTAGACTTTACATCGCC
- TTTTTTAAACATTAAAATTCTTACGTAATTTATAATCTTTAAAAA
AAGCATTTAATATTGCTCCCCGAACGATTGTGATTTCGATTCACA
TTTAAACAATTCAGA
- CCCATGAGAGTGAAATTGTTGTGATGTGGTTAACCCAATTAGAA
TTCGGGATTGACATGTCTTACCAAAGGTAGAACTTATACGCCA
TCTCATCCGATGCAAGC
- CTGGCTTAACTATGCGGCATCAGAGCAGATTGTAAGTACTGAGAGTGC
ACCATATGCGGTGTGAAATACCGCACAGATGCGTAAGGAGAAA
ATACCGCATCAGGCGCTC
- CTGTGACGGAAGATCACTTCGCAGAATAAATAAATCCTGGTGTC
CCTGTTGATACCGGGAAGCCCTGGGCCAACTTTTGCGGAAAATG
AGACGTTGATCGGCACG
- GATTTTTATACTTTAACTTGTGATATTTAAAGGTATTTAATTGT
AATAACGATACTCTGAAAAGTATTGAAAGTTAATTTGTGAGTGG
TCGCACATATCCTGTT

Lampiran 3. Listing Program

```

p=input('Masukkan banyak sekuen: ');

for j=1:p
    Sekuen{j}=input(['Masukkan sekuen DNA ke-'
int2str(j) ' : '], 's');
    [baris(j) kolom(j)]=size(Sekuen{j});
    k(j)=kolom(j)-15;
    fprintf(['banyaknya basa sekuen ke-' int2str(j) ':
%d \n'],kolom(j));
end

for j=1:p
    for i=1:k(j)
        x{j,i}=Sekuen{j}(i:(i+4));
        y{j,i}=Sekuen{j}((i+11):(i+15));
        d{j,i}=[x{j,i} y{j,i}] ;
    end
end

hmax=-9999;
posisi=[];

for z1=1:i
    for z2=1:i
        for z3=1:i
            e=( [d(1,z1) d(2,z2) d(3,z3)] );
            pos1=z1;
            pos2=z2;
            pos3=z3;
            f=char(e);
            g=f';
            ea1=findstr(g(1,:), 'a');
            [rowa1 cola1]=size(ea1);
            et1=findstr(g(1,:), 't');
            [rowt1 colt1]=size(et1);
            ec1=findstr(g(1,:), 'c');
            [rowc1 colc1]=size(ec1);
            eg1=findstr(g(1,:), 'g');
            [rowg1 colg1]=size(eg1);
            h1=max([cola1 colt1 colc1 colg1]);
            if h1==cola1
                L1='a';
            elseif h1==colt1

```

```

        L1='t';
elseif h1==colc1
        L1='c';
else
        L1='g';
end
ea2=findstr(g(2,:), 'a');
[rowa2 cola2]=size(ea2);
et2=findstr(g(2,:), 't');
[rowt2 colt2]=size(et2);
ec2=findstr(g(2,:), 'c');
[rowc2 colc2]=size(ec2);
eg2=findstr(g(2,:), 'g');
[rowg2 colg2]=size(eg2);
h2=max([cola2 colt2 colc2 colg2]);
if h2==cola2
        L2='a';
elseif h2==colt2
        L2='t';
elseif h2==colc2
        L2='c';
else
        L2='g';
end

ea3=findstr(g(3,:), 'a');
[rowa3 cola3]=size(ea3);
et3=findstr(g(3,:), 't');
[rowt3 colt3]=size(et3);
ec3=findstr(g(3,:), 'c');
[rowc3 colc3]=size(ec3);
eg3=findstr(g(3,:), 'g');
[rowg3 colg3]=size(eg3);
h3=max([cola3 colt3 colc3 colg3]);
if h3==cola3
        L3='a';
elseif h3==colt3
        L3='t';
elseif h3==colc3
        L3='c';
else
        L3='g';
end

ea4=findstr(g(4,:), 'a');
[rowa4 cola4]=size(ea4);
et4=findstr(g(4,:), 't');
[rowt4 colt4]=size(et4);
ec4=findstr(g(4,:), 'c');

```

```

[rowc4 colc4]=size(ec4);
eg4=findstr(g(4,:), 'g');
[rowg4 colg4]=size(eg4);
h4=max([cola4 colt4 colc4 colg4]);
if h4==cola4
    L4='a';
elseif h4==colt4
    L4='t';
elseif h4==colc4
    L4='c';
else
    L4='g';
end

ea5=findstr(g(5,:), 'a');
[rowa5 cola5]=size(ea5);
et5=findstr(g(5,:), 't');
[rowt5 colt5]=size(et5);
ec5=findstr(g(5,:), 'c');
[rowc5 colc5]=size(ec5);
eg5=findstr(g(5,:), 'g');
[rowg5 colg5]=size(eg5);
h5=max([cola5 colt5 colc5 colg5]);
if h5==cola5
    L5='a';
elseif h5==colt5
    L5='t';
elseif h5==colc5
    L5='c';
else
    L5='g';
end

ea6=findstr(g(6,:), 'a');
[rowa6 cola6]=size(ea6);
et6=findstr(g(6,:), 't');
[rowt6 colt6]=size(et6);
ec6=findstr(g(6,:), 'c');
[rowc6 colc6]=size(ec6);
eg6=findstr(g(6,:), 'g');
[rowg6 colg6]=size(eg6);
h6=max([cola6 colt6 colc6 colg6]);
if h6==cola6
    L6='a';
elseif h6==colt6
    L6='t';
elseif h6==colc6
    L6='c';
else

```

```

        L6='g';
    end

    ea7=findstr(g(7,:), 'a');
    [rowa7 cola7]=size(ea7);
    et7=findstr(g(7,:), 't');
    [rowt7 colt7]=size(et7);
    ec7=findstr(g(7,:), 'c');
    [rowc7 colc7]=size(ec7);
    eg7=findstr(g(7,:), 'g');
    [rowg7 colg7]=size(eg7);
    h7=max([cola7 colt7 colc7 colg7]);
    if h7==cola7
        L7='a';
    elseif h7==colt7
        L7='t';
    elseif h7==colc7
        L7='c';
    else
        L7='g';
    end

    ea8=findstr(g(8,:), 'a');
    [rowa8 cola8]=size(ea8);
    et8=findstr(g(8,:), 't');
    [rowt8 colt8]=size(et8);
    ec8=findstr(g(8,:), 'c');
    [rowc8 colc8]=size(ec8);
    eg8=findstr(g(8,:), 'g');
    [rowg8 colg8]=size(eg8);
    h8=max([cola8 colt8 colc8 colg8]);
    if h8==cola8
        L8='a';
    elseif h8==colt8
        L8='t';
    elseif h8==colc8
        L8='c';
    else
        L8='g';
    end

    ea9=findstr(g(9,:), 'a');
    [rowa9 cola9]=size(ea9);
    et9=findstr(g(9,:), 't');
    [rowt9 colt9]=size(et9);
    ec9=findstr(g(9,:), 'c');
    [rowc9 colc9]=size(ec9);
    eg9=findstr(g(9,:), 'g');
    [rowg9 colg9]=size(eg9);

```

