

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Pada umumnya pendekatan yang digunakan untuk melakukan penaksiran suatu parameter dalam model ekonometrik linier adalah memfokuskan fungsi tujuan (*objective function*), misalnya meminimumkan *sum of square function* atau memaksimumkan *likelihood function* terhadap parameter yang tidak diketahui. Bila diberikan suatu sampel data set, penaksiran parameter dapat dilakukan berdasarkan nilai optimal *objective function*. Pendekatan model linier untuk menggambarkan hubungan variabel ekonomi dapat diterima dengan alasan bahwa pada umumnya realitas situasi perekonomian dapat dilakukan pendekatan secara linier, ataupun model linier yang dapat ditransformasikan ke dalam bentuk non-linier. Namun demikian, kondisi linieritas tidak selalu dapat diaplikasikan sehingga spesifikasi nonlinier tidak dapat dihindarkan. Misalkan dilakukan penaksiran parameter fungsi produksi *Cobb-Douglas* dari suatu data sampel yang dipengaruhi oleh tenaga kerja (L) serta kapital/modal (K), dan mempunyai bentuk fungsi $Y = a \cdot L^{\alpha} \cdot K^{\beta} + \epsilon$ maka bentuk fungsi *Cobb-Douglas* merupakan model fungsi non-linier yang tidak bisa lagi ditransformasikan ke dalam bentuk linier (*Gujarati, 2003*).

Untuk menyelesaikan masalah non-linier dapat diselesaikan dengan beberapa macam metode seperti algoritma *gauss-newton*, *marquardt-levenberg*, *newton-raphson*, *hill-climbing*, dan sebagainya. Penaksiran model Regresi Non-Linier dengan menggunakan metode iterasi konvensional seperti algoritma *Gauss-*

Newton dalam mengestimasi parameter fungsi produksi *Cobb-Dauglas*, masih belum memberikan jaminan terjadinya konvergensi dan optimum global, sedangkan pada algoritma genetika sudah memberikan jaminan mengenai optimum global.. Selain itu, metode iterasi tersebut juga sangat sulit untuk menentukan nilai awal yang mencapai konvergensi dan diperlukan *trial error*.

Sejak tahun 1960-an, peningkatan minat para ilmuwan yang memfokuskan kepada bidang pengimitasian kehidupan telah menghasilkan berbagai teknik atau pendekatan optimisasi yang dapat disebut juga sebagai algoritma evolusioner, sehingga memungkinkan untuk menyelesaikan permasalahan-permasalahan optimisasi yang kompleks secara lebih mudah dan akurat. Ada tiga wilayah utama yang tercakup di dalam algoritma evolusioner : algoritma genetik (GAs), pemrograman evolusioner (EPs), dan strategi evolusi (ESs). Di antara ketiga pendekatan ini, algoritma genetik yang lebih populer digunakan pada berbagai permasalahan optimisasi (*Fadlisyah, dkk, 2009*).

Algoritma Genetika adalah suatu algoritma pencarian yang bertujuan untuk mencari solusi dari suatu masalah, baik dengan satu variabel ataupun dengan multivariabel. Metode ini meniru mekanisme dari genetika alam, yaitu untuk menemukan susunan-susunan gen yang terbaik di dalam tubuh makhluk hidup. Algoritma Genetika diusulkan pertama kali oleh John Holland dan teman-temannya di universitas Michigan untuk aplikasi seluler otomata. Teknik ini menjadi populer diantara saintis dan rekayasawan di seluruh dunia untuk memecahkan masalah optimasi mereka. Aplikasi algoritma genetika antara lain adalah : *job shop scheduling*, pembelajaran pengendali *neuro-fuzzy*, pemrosesan citra serta optimasi kombinatorial, dan sebagainya.

Dasar dari algoritma genetika adalah teori evolusi Darwin, yang menjelaskan prinsip dasar terciptanya banyak spesies makhluk hidup yang ada di dunia sekarang ini. Makhluk hidup yang dapat beradaptasi dengan lebih baik terhadap lingkungannya akan mempunyai kesempatan yang lebih besar untuk bertahan hidup dan bereproduksi sehingga mempengaruhi jumlah populasi spesies yang bersangkutan di waktu-waktu selanjutnya. Individu-individu tersebut dapat dikatakan berhasil menempatkan diri pada lingkungannya lebih baik dari yang lainnya.

Dalam perkembangannya metode ini banyak dipakai dalam berbagai disiplin ilmu, misalnya untuk menyelesaikan kasus-kasus yang mempunyai banyak solusi, dimana tidak ada kepastian solusi mana yang terbaik sehingga dalam penyelesaian masalah tersebut akan membutuhkan waktu yang sangat lama. Setiap solusi pada algoritma genetika diwakili oleh suatu individu atau satu kromosom. Keuntungan dari algoritma genetika adalah sifat metode *search*nya yang lebih optimal, tanpa terlalu memperbesar ruang pencarian dan tanpa kehilangan *completeness*. Sudah banyak permasalahan optimasi yang telah diselesaikan dengan menggunakan algoritma genetika, dan hasil yang diperoleh biasanya lebih baik dari metode sebelumnya, walaupun tidak menutup kemungkinan bahwa ada juga hasil akhir yang kurang begitu memuaskan (*Desiani dan Muhammad, 2006*).

1.2 PERUMUSAN MASALAH

Dalam penulisan tugas akhir ini, permasalahan yang dibahas adalah menyelesaikan persoalan estimasi parameter fungsi regresi non linear, dengan menggunakan algoritma genetika.

1.3 PEMBATAAN MASALAH

Penulisan tugas akhir ini dibatasi pada permasalahan estimasi parameter fungsi *cobb-douglas* menggunakan algoritma genetika, dengan data yang digunakan adalah mengenai hasil produksi untuk ekonomi rakyat Mexico pada tahun 1955-1974, dengan *GDP* (y) dan input *Labour* () serta *Capital* ()

1.4 TUJUAN PENULISAN

Tujuan penulisan tugas akhir ini adalah untuk menentukan solusi optimal dari permasalahan estimasi parameter-parameter fungsi produksi *cobb-douglas* dengan meminimumkan fungsi tujuan *least square* dan memaksimumkan fungsi tujuan *likelihood*, dan menggunakan algoritma genetika.

1.5 MANFAAT PENULISAN

Manfaat penulisan tugas akhir ini adalah agar dapat digunakan untuk memberikan gambaran umum tentang struktur serta tahapan-tahapan penyelesaian masalah menggunakan algoritma genetika, dan untuk mengetahui nilai parameter terbaik dari fungsi produksi *cobb-douglas*.

1.6 SISTEMATIKA PENULISAN

Sistematika penulisan pada Tugas Akhir dengan judul "Estimasi Parameter Fungsi *Cobb-Douglas* dengan Algoritma Genetika" meliputi empat bab. BAB I merupakan Pendahuluan yang menjelaskan tentang Latar Belakang, Perumusan Masalah, Pembatasan Masalah, Tujuan, Manfaat dan Sistematika Penulisan. BAB II merupakan Teori Penunjang yang membahas beberapa konsep dasar Algoritma Genetika, fungsi produksi *cobb-douglas*, dan Non-Linear *Least Square* serta *Maximum Likelihood*, yang akan digunakan sebagai acuan dalam memahami dan memecahkan permasalahan yang dibahas. BAB III berisi pembahasan mengenai prosedur-prosedur yang digunakan atau tahapan eksperimen dalam memecahkan masalah estimasi parameter fungsi *cobb-douglas* dengan menggunakan Algoritma Genetika. Sedangkan BAB IV berisi Penutup, yaitu diberikan beberapa kesimpulan dan saran dari penulisan tugas akhir.

BAB II

TINJAUAN PUSTAKA

2.1 Algoritma Genetika

2.1.1 Sejarah Algoritma Genetika

Algoritma genetika pertama kali diperkenalkan oleh John Holland dari Universitas Michigan pada awal 1970-an di New York, Amerika Serikat. John Holland bersama murid-murid serta rekan kerjanya lalu menghasilkan buku yang berjudul "*Adaption in Natural and Artificial Systems*" pada tahun 1975, yang cara kerjanya berdasarkan pada seleksi dan genetika alam. Konsep yang dipergunakan dalam algoritma genetika adalah mengikuti apa yang dilakukan oleh alam (*Desiani dan Muhammad, 2006*).

Algoritma Genetik khususnya diterapkan sebagai simulasi komputer dimana sebuah populasi representasi abstrak ([kromosom](#)) dari solusi-solusi calon (individual) pada sebuah masalah optimisasi akan berkembang menjadi solusi-solusi yang lebih baik. Secara tradisional solusi-solusi tersebut dilambangkan dalam biner sebagai string '0' dan '1', walaupun dimungkinkan juga penggunaan [penyandian \(*encoding*\)](#) yang berbeda. Evolusi dimulai dari sebuah populasi individual acak yang lengkap dan terjadi dalam generasi-generasi. Dalam tiap generasi kemampuan keseluruhan populasi dievaluasi, kemudian *multiple individuals* dipilih dari populasi sekarang (*current*) secara *stochastic* (berdasarkan kemampuan mereka) lalu dimodifikasi (dengan mutasi atau rekombinasi) menjadi bentuk populasi baru yang menjadi populasi sekarang (*current*) pada [iterasi](#) berikutnya dari algoritma.

2.1.2 Aplikasi Algoritma Genetika

Sejak pertama kali dirintis oleh John Holland, Algoritma Genetika telah dipelajari, diteliti dan diaplikasikan secara luas pada berbagai bidang. Algoritma Genetika banyak digunakan pada masalah praktis yang berfokus pada pencarian parameter-parameter yang optimal. Hal ini membuat banyak orang mengira bahwa algoritma genetika hanya digunakan untuk menyelesaikan masalah optimasi saja. Namun demikian, pada kenyataannya algoritma genetika juga memiliki performansi yang bagus untuk masalah-masalah selain optimasi.

Keuntungan penggunaan algoritma genetika sangat jelas terlihat dari kemudahan implementasi dan kemampuannya untuk menemukan solusi yang bagus dan bisa diterima secara cepat untuk masalah-masalah berdimensi tinggi. Algoritma Genetika sangat berguna dan efisien untuk masalah dengan karakteristik sebagai berikut (*Suyanto, 2005*) :

- a. Ruang masalah sangat besar, kompleks, dan sulit dipahami,
- b. Kurang atau bahkan tidak ada pengetahuan yang memadai untuk merepresentasikan masalah ke dalam ruang pencarian yang lebih sempit,
- c. Tidak tersedianya analisis matematika yang memadai,
- d. Ketika metode-metode konvensional sudah tidak mampu menyelesaikan masalah yang dihadapi,
- e. Solusi yang diharapkan tidak harus paling optimal, tetapi cukup “bagus” atau bisa diterima,
- f. Terdapat batasan waktu, misalnya dalam *real time system* atau sistem waktu nyata.

Algoritma Genetika juga telah banyak diaplikasikan untuk berbagai macam permasalahan, antara lain (Suyanto, 2005) :

a. Optimasi

Algoritma Genetika untuk optimasi antara lain untuk optimasi numerik dan optimasi kombinatorial seperti *Traveling Salesmen Problem* (TSP), Perancangan *Integrated Circuit* atau IC, *Job Scheduling*, Optimasi video dan suara.

b. Pemrograman Otomatis

Algoritma Genetika untuk pemrograman otomatis antara lain untuk melakukan proses evolusi terhadap program komputer dalam merancang struktur komputasional, seperti *cellular automata* dan *sorting networks*.

c. Machine Learning

Algoritma Genetika telah berhasil diaplikasikan untuk memprediksi struktur protein. Algoritma Genetika juga berhasil diaplikasikan dalam perancangan *neural networks* (jaringan syaraf tiruan) untuk melakukan proses evolusi terhadap aturan-aturan pada *learning classifier system* atau *symbolic production system*. Algoritma Genetika juga dapat digunakan untuk mengontrol robot.

d. Model Ekonomi

Dalam bidang ekonomi, algoritma genetika digunakan untuk memodelkan proses-proses inovasi dan pembangunan *bidding strategies*.

e. Model Sistem Imunisasi

Contoh penggunaan algoritma genetika dalam bidang ini untuk memodelkan berbagai aspek pada sistem imunisasi alamiah, termasuk *somatic mutation* selama kehidupan individu dan menemukan keluarga dengan gen ganda (*multi gen families*) sepanjang waktu evolusi.

f. Model Ekologis

Algoritma genetika juga dapat digunakan untuk memodelkan fenomena ekologis seperti *host-parasite co evolutions*, simbiosis dan aliran sumber di dalam ekologi.

2.1.3 Struktur Umum Algoritma Genetika

Algoritma genetika memiliki perbedaan yang mendasar dengan metode pencarian solusi optimum model matematika kalkulus, perbedaan tersebut dijelaskan pada tabel 2.1 berikut :

Tabel 2.1 Perbedaan Algoritma Genetika dengan Model Matematika Kalkulus

Algoritma Genetika	Model Matematika Kalkulus
Pencarian solusi menggunakan informasi langsung dari hasil transfer tiap-tiap parameternya ke suatu fungsi yang dapat mewakili tujuan dari proses optimasi yang sedang dilakukan.	Pencarian solusi menggunakan prosedur-prosedur matematis dan prosedur-prosedur turunan
Proses pencarian solusi dilakukan pada sekumpulan titik pencarian dengan titik acuan yang sembarang.	Proses pencarian solusi dilakukan pada satu titik pencarian dengan titik acuan yang sudah ditentukan
Bersifat probabilistik	Bersifat deterministik

Sumber : *Fadlisyah, dkk., 2009*

Secara garis besar algoritma genetika memiliki banyak kesamaan dengan mekanisme genetika alami dan seleksi alam, baik dalam tahapan prosesnya maupun definisi dari istilah-istilah atau terminologi yang digunakan. Terminologi algoritma genetika dan genetika alami bisa dilihat pada tabel 2.2 berikut :

Tabel 2.2 Terminologi Algoritma Genetika

Algoritma Genetika	Genetika Alami
String	kromosom
Posisi string	Lokus
karakter	Gen
Nilai karakter	Alel
Struktur	Genotip
Kode struktur	Fenotip

Secara alami semua organisme terdiri dari sel yang terdiri dari sekumpulan kromosom. Kromosom terbentuk dari sekumpulan *gen*, membuat satu kesatuan yang tersusun dalam rangkaian linier. Setiap gen mempunyai letak tersendiri di dalam kromosom, disebut *lokus*. Gen tersusun dari DNA yang membawa sifat keturunan. Setiap gen menyandikan protein tertentu suatu sifat. Contoh : gen warna mata binatang dengan posisi lokus 10. Bagian tertentu dari gen di dalam *genome* disebut *genotip*. Beberapa sifat individu yang menunjukkan perbedaan gen dan berada pada bagian yang berbeda disebut *alel*.

Algoritma genetika didasari oleh bidang genetik natural dan ilmu komputer, maka istilah-istilah yang digunakan akan berupa campuran dari disiplin kedua ilmu tersebut. Adapun penjelasan dari istilah-istilah yang digunakan dapat dilihat pada tabel 2.3 berikut :

Tabel 2.3 Istilah di dalam algoritma genetika

No	Algoritma Genetika	Penjelasan	Definisi
1	Kromosom (string, individual)	Solusi (pengkodean)	Struktur yang mengkodekan preskripsi yang menspesifikasikan bagaimana organisme dikonstruksikan
2	Gen-gen (bit-bit)	Bagian dari solusi	Bagian dari kromosom yang berupa sejumlah struktur individu
3	locus	Posisi dari gen	
4	Alleles	Nilai gen	
5	Phenotype	Solusi yang diuraikan	Organisme yang dihasilkan dari sekumpulan kromosom
6	Genotype	Solusi yang disandakan	Sekumpulan kromosom-kromosom yang lengkap

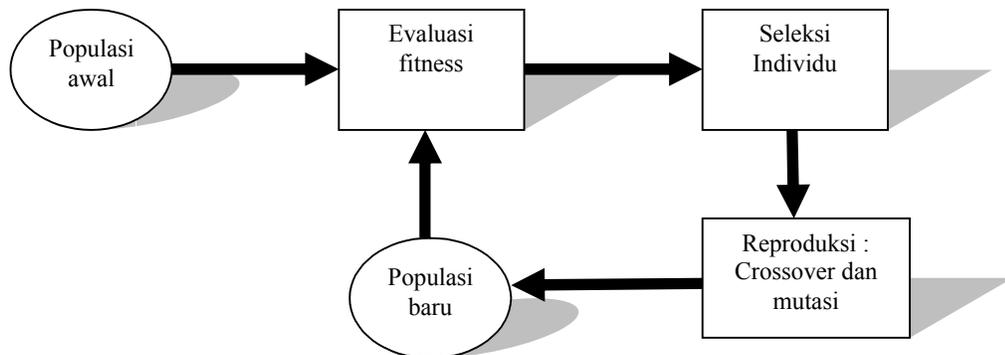
Sumber : Fadlisyah, dkk., 2009

Bentuk lazim algoritma genetik telah dideskripsikan oleh Goldberg. Algoritma genetik merupakan teknik *search stochastic* yang berdasarkan mekanisme seleksi alam dan genetika natural. Yang membedakan algoritma genetik dengan berbagai algoritma konvensional lainnya adalah bahwa algoritma genetik memulai dengan suatu himpunan penyelesaian acak awal yang disebut *populasi*. Setiap individu di dalam populasi disebut *kromosom*, yang merepresentasikan suatu penyelesaian terhadap masalah yang ditangani. Sebuah kromosom terdiri dari sebuah string yang berisi berbagai simbol, dan biasanya, tetapi tidak mutlak, string tersebut berupa sederetan bit-bit biner “0” dan “1”. Sebuah kromosom tumbuh atau berkembang biak melalui berbagai iterasi yang berulang-ulang, dan disebut sebagai *generasi*. Pada setiap generasi, berbagai

kromosom yang dihasilkan akan dievaluasi menggunakan suatu pengukuran *fitness*.

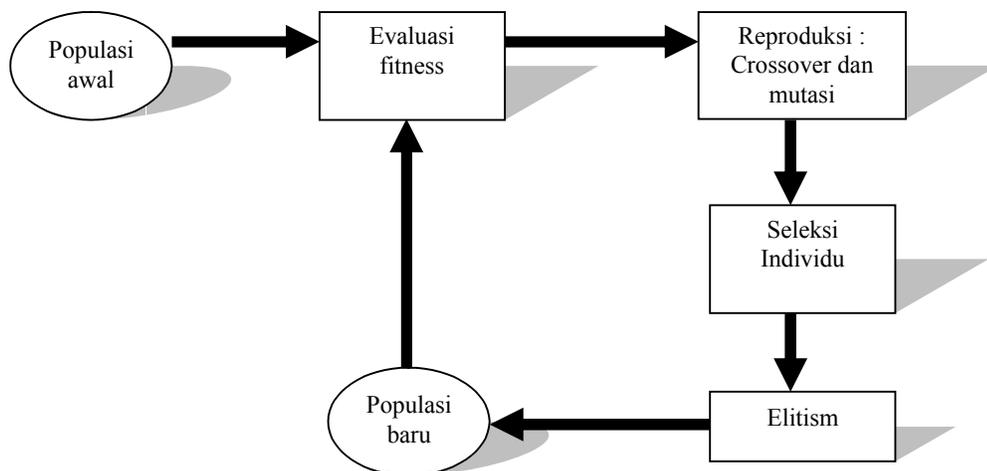
Siklus Algoritma Genetika

Siklus dari algoritma genetika pertama kali diperkenalkan oleh *David Goldberg*, dimana gambaran siklus tersebut adalah sebagai berikut :



Gambar 2.1 Siklus algoritma genetika oleh David Goldberg

Siklus ini kemudian diperbaiki oleh beberapa ilmuwan yang mengembangkan algoritma genetika, yaitu *Zbignew Michalewicz* dengan menambahkan operator *elitism* dan membalik proses seleksi setelah proses reproduksi.



Gambar 2.2 Siklus algoritma genetika yang diperbarui oleh Zbignew Michalewicz

2.1.4 Komponen-komponen utama Algoritma Genetika

Ada 6 komponen utama yang terdapat di dalam algoritma genetika, yaitu (Kusumadewi, 2003) :

2.1.4.1 Teknik Penyandian (Pengkodean)

Teknik penyandian disini meliputi penyandian gen dari kromosom. Gen merupakan bagian dari kromosom, dimana satu gen biasanya akan mewakili satu variabel. Gen dapat direpresentasikan dalam bentuk: string bit, pohon, array bilangan real, daftar aturan, elemen permutasi, elemen program dan lain-lain.

Contoh dari representasi kromosom antara lain sebagai berikut :

- 1) String bit : 10011, 11101, dst
- 2) Bilangan Real : 65.65, 562.88, dst
- 3) Elemen Permutasi : E2, E10, dst
- 4) Daftar Aturan : R1, R2, R3, dst
- 5) Elemen Program : pemrograman genetika, dst
- 6) Struktur lainnya

Misalkan ingin dipecahkan masalah estimasi fungsi produksi *Cobb-Dauglas* yaitu $Y = aL^bK^c$ dengan sampel yang ada untuk L dan K berapa nilai a, b, c , dengan fungsi tujuan meminimumkan *least square* atau memaksimumkan fungsi *likelihood*. Persoalan tersebut dapat diselesaikan dengan algoritma genetika, yaitu ketiga parameter a, b, c dikodekan dalam kromosom yang masing-masing berisi sejumlah gen yang mengkodekan informasi yang disimpan di dalam kromosom. Misalkan untuk memudahkan digunakan *binary encoding* dengan panjang kromosom 12 gen (12 bits), masing-masing parameter

, , dikodekan dengan 4 gen, sehingga dapat diilustrasikan skema pengkodean pada tabel 2.4 berikut :

Tabel 2.4 Skema *Binary Encoding*

Parameter	β_1				β_2				β_3			
Binary number	1	0	1	1	1	1	1	0	0	0	1	1
	g1	g2	g3	g4	g5	g6	g7	g8	g9	g10	g11	g12
decimal number	11				14				3			

Jika nilai parameter yang akan dicari mempunyai *constraint* yaitu $0 < \beta_i < 1$ maka berdasarkan binary encoding, nilai parameter dari β_i dapat diperoleh dengan menggunakan rumus (Fadlisyah, dkk.,2009) :

$$\beta_i = \sum_{g=1}^{n_i} \frac{g}{2^{n_i}} \times \frac{1 - 0}{1 - 0}$$

dimana n_i menyatakan banyaknya bit atau gen, yang pada tabel 2.4 masing-masing berisi empat buah bit untuk setiap parameter dan *constraint* $0 < \beta_i < 1$, sehingga :

$$\beta_1 = 0 + 11 \times \frac{1 - 0}{2^4 - 1} = 0,7333$$

$$\beta_2 = 0 + 14 \times \frac{1 - 0}{2^4 - 1} = 0,9333$$

$$\beta_3 = 0 + 3 \times \frac{1 - 0}{2^4 - 1} = 0,2000$$

Kebutuhan akan presisi berimplikasi terhadap jangkauan domain dari masing-masing variabel yang harus dibagi, sedikitnya sebesar $(\epsilon - \delta) \times 10^6$ jangkauan, sehingga bit-bit yang diperlukan bisa dihitung sebagai berikut (Fadlisyah, dkk.,2009) :

$$2^{\lceil \log_2 \left(\frac{\epsilon - \delta}{\epsilon - \delta} \times 10^6 \right) \rceil} \leq 2^{\lceil \log_2 \left(\frac{\epsilon - \delta}{\epsilon - \delta} \times 10^6 \right) \rceil}$$

Setelah skema pengkodean ditentukan, algoritma genetika diinisialisasi untuk sebuah populasi dengan N kromosom. Gen-gen yang mengisi masing-masing kromosom dibangkitkan secara random. Masing-masing kromosom akan dikodekan menjadi individu dengan nilai *fitness* tertentu, dan kemudian sebuah populasi baru akan dibentuk dengan menggunakan mekanisme seleksi alamiah, yaitu memilih individu-individu secara proporsional terhadap nilai *fitness*-nya, dan genetika alamiah, yakni pindah silang (*crossover*) serta mutasi. Pada algoritma genetika metode yang akan digunakan adalah dengan skema pergantian populasi yang disebut *generational replacement*, artinya, N kromosom dari suatu generasi digantikan sekaligus oleh N kromosom baru hasil pindah silang dan mutasi.

2.1.4.2 Prosedur Inisialisasi (membangkitkan populasi awal)

Membangkitkan populasi awal adalah membangkitkan sejumlah individu secara acak atau melalui prosedur tertentu. Ukuran populasi tergantung pada masalah yang akan dipecahkan dan jenis operator genetika yang akan diimplementasikan. Setelah ukuran populasi ditentukan, kemudian harus dilakukan inisialisasi terhadap kromosom yang terdapat pada populasi tersebut. Inisialisasi kromosom dilakukan secara acak, namun demikian harus tetap memperhatikan domain solusi dan kendala permasalahan yang ada.

Teknik dalam membangkitkan populasi awal ini ada beberapa macam, diantaranya adalah sebagai berikut :

1) Random Generator

Inti dari cara ini adalah melibatkan pembangkitan bilangan random untuk nilai setiap gen sesuai dengan representasi kromosom yang digunakan.

Gen nantinya berisi pembulatan dari bilangan random yang dibangkitkan sebanyak N_{ipop} (jumlah populasi) \times N_{bits} (jumlah gen dalam tiap kromosom)

2) Pendekatan tertentu (memasukan nilai tertentu kedalam gen)

Cara ini adalah dengan memasukan nilai tertentu kedalam gen dari populasi awal yang dibentuk.

3) Permutasi gen

Salah satu cara dari pembangkitan populasi awal dengan permutasi gen adalah penggunaan permutasi *Josephus* dalam permasalahan kombinatorial seperti *travelling salesmen problem (TSP)*.

2.1.4.3 Fungsi Evaluasi

Ada tiga langkah dalam proses mengevaluasi nilai *fitness* kromosom, yaitu:

1. Mengganti genotip kromosom menjadi fenotip kromosom, ini berarti mengganti *binary strings* menjadi *real value*
2. Mengevaluasi fungsi objektif
3. Mengganti nilai dari fungsi objektif menjadi nilai *fitness*. Agar nilai *fitness* selalu bernilai positif, maka nilai *fitness* dari setiap kromosom sama dengan memaksimalkan objektif dikurangi objektif yang telah dievaluasi untuk setiap kromosom dalam populasi.

Suatu individu dievaluasi berdasarkan suatu fungsi tertentu sebagai ukuran performansinya. Di dalam evolusi alam, individu yang bernilai *fitness* tinggi yang akan bertahan hidup, sedangkan individu yang bernilai *fitness* rendah akan mati. Pada masalah optimasi, solusi yang akan dicari adalah memaksimalkan sebuah

fungsi *likelihood* dan meminimumkan *least square* untuk fungsi produksi *Cobb-Dauglas*.

2.1.4.4 Seleksi

Seleksi ini bertujuan untuk memberikan kesempatan reproduksi yang lebih besar bagi anggota populasi yang paling fit. Langkah pertama dalam seleksi adalah mencari nilai *fitness*. Masing-masing individu dalam suatu wadah seleksi akan menerima probabilitas reproduksi yang tergantung dari nilai objektif dirinya sendiri terhadap nilai objektif dari semua individu dalam wadah seleksi tersebut. Nilai *fitness* inilah yang nantinya akan digunakan pada tahap seleksi berikutnya (*Kusumadewi, 2003*).

1. Metode Seleksi

Ada beberapa metode seleksi dari induk, antara lain sebagai berikut :

a) Rank-based fitness assignment

Pada *Rank-based fitness*, populasi diurutkan menurut nilai objektifnya. Nilai *fitness* dari tiap-tiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya

b) Roulette wheel selection

Metode seleksi roda roulette ini merupakan metode yang paling sederhana serta paling banyak digunakan, dan sering juga dikenal dengan nama *stochastic sampling with replacement*. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara beraturan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan dengan ukuran *fitness*nya. Sebuah bilangan random akan

dibangkitkan dan individu yang memiliki segmen dalam kawasan bilangan random tersebut akan diseleksi. Proses ini diulang hingga diperoleh sejumlah individu yang diharapkan.

Tabel 2.5 Contoh Populasi dengan 3 kromosom beserta *fitness*-nya

Kromosom	String	Nilai fitness	persentase
A	01101	5	19%
B	10000	12	46%
C	01110	9	35%

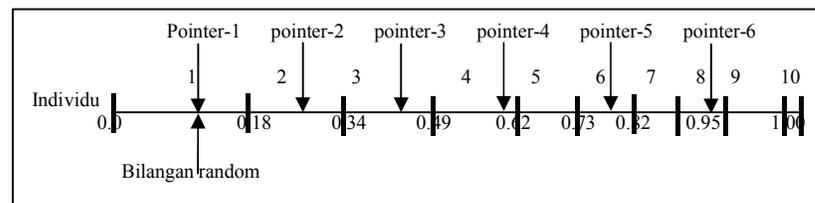
Dari tabel 2.5 kromosom A mempunyai probabilitas 19% untuk dipilih dalam proses reproduksi setiap kali melakukan pemilihan kromosom (setiap roda diputar). Probabilitas masing-masing individu dapat diketahui dengan pembagian *fitness* masing-masing individu dengan total *fitness* dalam populasi

Skema dengan seleksi roda roulette ini adalah berdasarkan *fitness scale* (skala fitness). Terpilihnya suatu kromosom dalam populasi untuk dapat berkembang biak sebanding dengan *fitness*-nya. *Tradeoff* antara eksplorasi dan eksploitasi terjadi jika terdapat satu atau sekelompok kecil kromosom yang mempunyai *fitness* yang baik, yaitu mengeksplorasi bagian-bagian baru dalam ruang pencarian, atau terus mengeksplorasi informasi yang telah diperoleh. Kecenderungan kromosom yang baik untuk terpelihara terus dapat membawa ke hasil optimum lokal atau konvergensi dini (*premature convergence*) ke suatu hasil yang bukan optimum global. Namun demikian, jika semua kromosom dalam populasi mempunyai *fitness* yang hampir sama, maka seleksi ini akan menjadi seleksi yang bersifat acak.

c) Stochastic universal sampling

Stochastic universal sampling memiliki nilai bias nol dan penyebaran yang minimum. Pada metode ini, individu-individu dipetakan dalam suatu segmen garis secara berurutan sedemikian hingga tiap-tiap segmen individu memiliki ukuran yang sama dengan ukuran *fitness*-nya seperti halnya pada seleksi roda roulette, dan diberikan sejumlah *pointer* sebanyak individu yang diseleksi di garis tersebut. Andaikan N adalah jumlah individu, dan posisi pointer pertama diberikan secara acak pada range $[1, 1/N]$.

Apabila ada 6 individu yang akan diseleksi, maka jarak antar *pointer* adalah $1/6=0.167$ (gambar 2.3), sehingga misalkan bilangan random yang dibangkitkan pada range $[0, 0.167]$ adalah 0, 1, maka hasil yang diperoleh setelah seleksi adalah :



Gambar 2.3 Seleksi *Universal Sampling*

d) Seleksi lokal (*Local selection*)

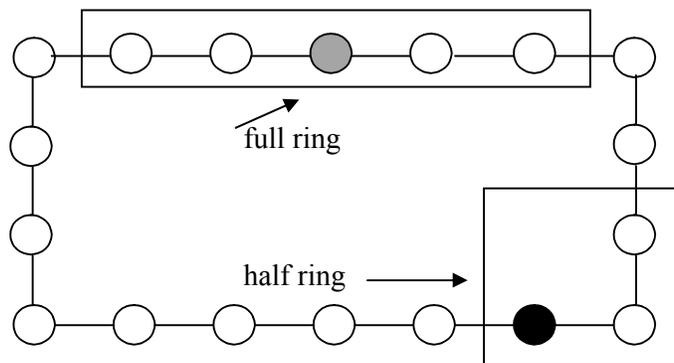
Pada seleksi lokal setiap individu yang berada di dalam *constraint* tertentu disebut dengan nama lingkungan lokal. Interaksi antar individu hanya dilakukan dalam wilayah tersebut. Lingkungan tersebut ditetapkan sebagai struktur dimana populasi tersebut terdistribusi. Lingkungan tersebut juga dipandang sebagai sekelompok pasangan-pasangan yang potensial.

Langkah pertama yang harus dilakukan adalah menyeleksi separuh pertama dari populasi yang berpasangan secara random, kemudian lingkungan baru tersebut diberikan pada setiap individu yang terseleksi.

Struktur lingkungan pada seleksi lokal dapat berbentuk :

- a) Linear : *full ring* dan *half ring*
- b) Dimensi-2: *full cross* dan *half cross*, *full star* dan *half star*
- c) Dimensi-3 dan struktur yang lebih kompleks yang merupakan kombinasi dari kedua struktur diatas.

Jarak antara individu dengan struktur tersebut akan sangat menentukan ukuran lingkungan. Individu yang terdapat dalam lingkungan dengan ukuran yang lebih kecil, akan lebih terisolasi dibandingkan dengan individu yang terletak pada lingkungan dengan ukuran yang lebih besar. Contoh pada gambar 2.4 di bawah ini menunjukkan seleksi lokal dalam lingkungan linier (*full and half ring*).



Gambar 2.4 Lingkungan linier (jarak=2) : *full and half ring*

d) Seleksi dengan pemotongan (*Truncation selection*)

Seleksi dengan pemotongan ini lebih berkesan sebagai seleksi buatan dan biasanya digunakan oleh populasi yang jumlahnya sangat besar. Pada metode ini, individu-individu yang terbaik saja yang akan diseleksi sebagai induk. Parameter yang digunakan dalam metode ini adalah suatu nilai ambang *trunc* yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk yang berkisar antara 50% - 10%. Individu-individu yang ada di bawah nilai ambang ini tidak akan menghasilkan keturunan.

e) Seleksi dengan turnamen (*Tournament selection*)

Pada metode seleksi dengan turnamen ini akan ditetapkan suatu nilai *tour* untuk individu-individu yang dipilih secara acak (random) dari suatu populasi. Individu-individu yang terbaik dalam kelompok ini akan diseleksi sebagai induk. Parameter yang digunakan pada metode ini adalah ukuran *tour* yang bernilai antara 2 sampai N (jumlah individu dalam suatu populasi).

2. Perbandingan antar metode seleksi

Ada beberapa definisi yang biasa digunakan untuk melakukan perbandingan terhadap metode seleksi yang akan digunakan, antara lain :

- a) *Selective Pressure* : probabilitas dari individu terbaik yang akan diseleksi dibandingkan dengan rata-rata probabilitas dari semua individu yang diseleksi.

- b) *Bias* : perbedaan absolut antara *fitness* ternormalisasi dari suatu individu dan probabilitas reproduksi yang diharapkan.
- c) *Spread* : range nilai kemungkinan untuk sejumlah *off-spring* dari suatu individu.
- d) *Loss of diversity* : proporsi dari individu-individu dalam suatu populasi yang tidak terseleksi selama fase seleksi.
- e) *Selection intensity* : nilai *fitness* rata-rata yang diharapkan dalam suatu populasi setelah dilakukan seleksi (menggunakan distribusi Gauss ternormalisasi).
- f) *Selection variance* : variansi yang diharapkan dari distribusi *fitness* dalam populasi setelah dilakukan seleksi (menggunakan distribusi Gauss ternormalisasi).

Penekanan selektif dilakukan dengan memilih jenis seleksi yang sesuai. Umumnya jenis seleksi pada roda roulette paling sering digunakan, terkadang juga metode ranking dan turnamen. Ada beberapa metode lebih lanjut yang mengubah parameter seleksi ini selama eksekusi algoritma genetika. Yang perlu diperhatikan dalam seleksi adalah prinsip *elitism*, yang dilakukan dalam sekali seleksi untuk *update* generasi, biasanya digunakan *steady-state update*. Jadi tujuan utama dari *elitism* ini adalah untuk menjaga agar individu-individu yang bernilai *fitness* tertinggi tidak hilang selama proses evolusi, maka perlu dibuat kopyannya.

2.1.4.5 Operator Genetika

Algoritma genetika merupakan proses pencarian yang *heuristic* dan acak sehingga penekanan pemilihan operator yang digunakan sangat menentukan

keberhasilan algoritma genetika dalam menemukan solusi optimum suatu masalah yang diberikan. Hal yang harus diperhatikan adalah menghindari terjadinya konvergensi prematur, dimana dicapai solusi optimum yang belum waktunya, dalam arti bahwa solusi yang diperoleh adalah hasil optimum lokal.

Ada dua operator genetika, yaitu :

1. Operator untuk melakukan rekombinasi, yang terdiri dari:

a) Rekombinasi bernilai real

i. Rekombinasi diskret

Rekombinasi diskret akan menukar nilai variabel antar kromosom induk. Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1 : 12 25 5

Induk 2 : 123 4 34

Untuk tiap-tiap variabel induk yang menyumbangkan variabelnya ke anak yang dipilih secara random dengan probabilitas yang sama.

sampel 1 : 2 2 1

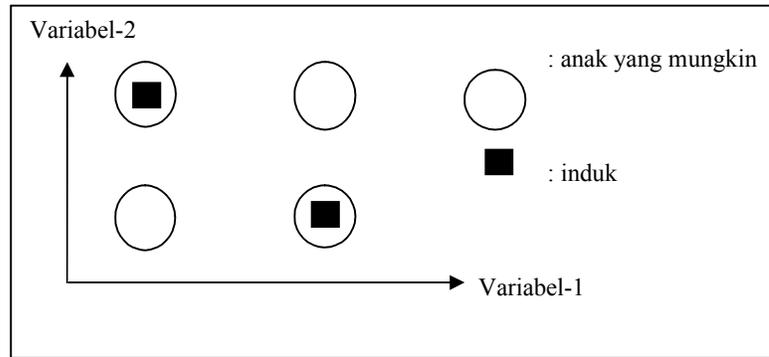
sampel 2 : 1 2 1

Setelah rekombinasi, kromosom-kromosom baru yang terbentuk yaitu :

Anak 1 : 123 4 5

Anak 2 : 12 4 5

Rekombinasi dapat digunakan untuk sembarang variabel (biner, real, atau simbol).



Gambar 2.5 Rekombinasi diskret

ii. Rekombinasi intermediate (menengah)

Rekombinasi intermediate hanya dapat digunakan untuk variabel real (dan variabel yang bukan biner).

Anak dihasilkan menurut aturan sebagai berikut :

$$\text{Anak} = \text{induk 1} + \alpha (\text{induk 2} - \text{induk 1})$$

Dengan alpha adalah faktor skala yang dipilih secara random pada interval $[-d, 1+d]$, biasanya $d=0,25$. Tiap-tiap variabel pada anak merupakan hasil kombinasi variabel-variabel menurut aturan di atas dengan nilai alpha dipilih ulang untuk tiap variabel. Misalkan ada 2 individu dengan 3 variabel, yaitu:

Induk 1 : 12 25 5

Induk 2 : 123 4 34

Misalkan nilai alpha yang terpilih :

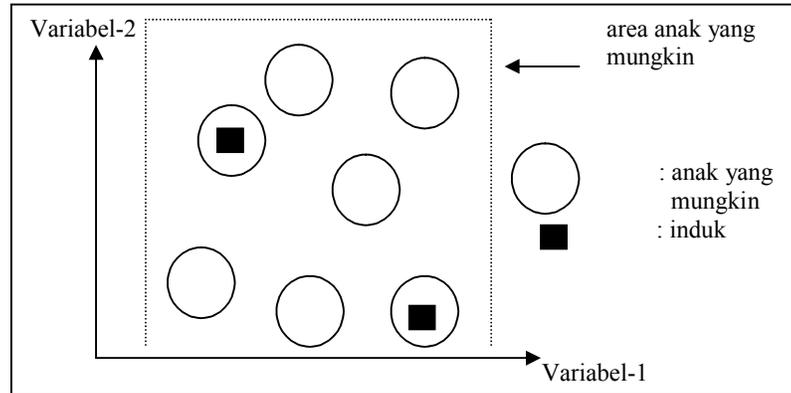
sampel 1 : 0,5 1,1 -0,1

sampel 2 : 0,1 0,8 0,5

Setelah rekombinasi, kromosom-kromosom baru yang terbentuk :

Anak 1 : 67,5 1,9 2,1

Anak 2 : 23,1 8,2 19,5



Gambar 2.6 Posisi anak yang mungkin pada rekombinasi menengah

Rekombinasi dapat digunakan untuk sembarang variabel (biner, real, atau simbol).

iii. Rekombinasi garis

Pada dasarnya rekombinasi garis ini hampir sama dengan rekombinasi menengah, hanya saja nilai alpha untuk semua variabel adalah sama. Misalkan ada 2 kromosom dengan 3 variabel:

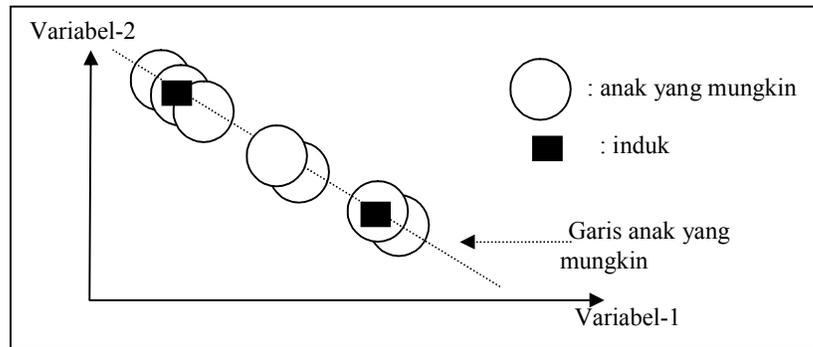
induk1 :	12	25	5
induk2 :	123	4	34

Untuk tiap-tiap variabel induk yang menyumbangkan variabelnya ke anak dipilih secara random dengan probabilitas yang sama

sample 1 :	0,5
sample 2 :	0,1

Setelah rekombinasi kromosom-kromosom baru yang terbentuk adalah:

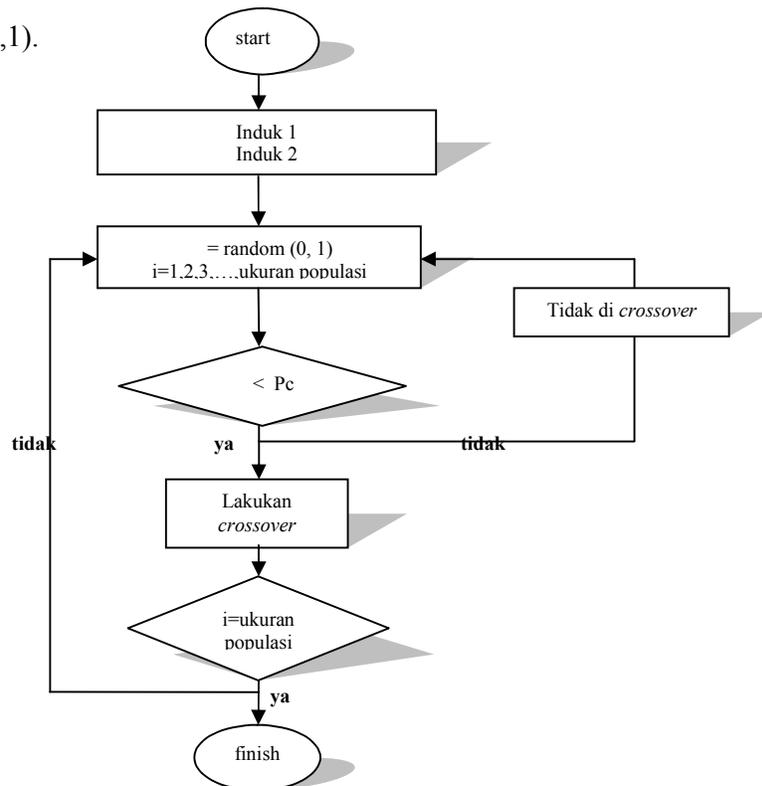
anak1:	67,5	14,5	19,5
anak2:	23,1	22,9	7,9



Gambar 2.7 Posisi anak yang mungkin pada rekombinasi garis

b) Rekombinasi bernilai biner atau penyilangan (*Crossover*)

Crossover melibatkan dua induk untuk membentuk kromosom baru. Pindah silang menghasilkan titik baru dalam ruang pencarian untuk siap diuji. Proses *crossover* dilakukan pada setiap individu dengan probabilitas *crossover* (P_c) yang ditentukan secara acak dalam rentang (0,1).



Gambar 2.8 Diagram alir *crossover*

i. Penyilangan satu titik (*single-point Crossover*)

Pada penyilangan satu titik, posisi penyilangan k ($k=1,2,\dots,N-1$) dengan panjang kromosom (N) diseleksi secara random. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak. Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : 0 1 1 1 0 | 0 1 0 1 1 1 0

Induk 2 : 1 1 0 1 0 | 0 0 0 1 1 0 1

Posisi menyilang yang terpilih acak : misalkan setelah bit ke-5. Setelah dilakukan penyilangan, diperoleh kromosom-kromosom baru:

Anak 1 : 0 1 1 1 0 | 0 0 0 1 1 0 1

Anak 2 : 1 1 0 1 0 | 0 1 0 1 1 1 0

ii. Penyilangan dua titik (*two-point Crossover*)

Penyilangan ini menentukan dua titik secara acak sebagai batas untuk menukar 2 kromosom induk yang berada diantaranya untuk menghasilkan 2 individu yang baru. Misalkan ada 2 kromosom dengan panjang kromosom 10

Induk 1 : 110 | 000 | 1100

Induk 2 : 100 | 100 | 1011

Posisi menyilang yang terpilih acak : misalkan setelah bit ke-3 dan ke-6, maka setelah dilakukan penyilangan diperoleh kromosom baru :

Anak 1 : 110 | 100 | 1100

Anak 2 : 100 | 000 | 1011

iii. Penyilangan banyak titik (*multi-point Crossover*)

Pada penyilangan ini, jumlah titik posisi penyilangan ($k=1,2,\dots,N-1,i=1,2,\dots,m$) dengan panjang kromosom (N) diseleksi secara random dan tidak diperbolehkan ada posisi yang sama, serta diurutkan naik. Variabel-variabel ditukar antar kromosom pada titik tersebut untuk menghasilkan anak. Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : 011100101110

Induk 2 : 110100001101

Posisi penyilangan yang terpilih adalah setelah bit ke- 2, 6, dan 10.

Setelah penyilangan, diperoleh kromosom-kromosom baru :

anak 1 : 01 | 0100 | 1011 | 01

anak 2 : 11 | 1100 | 0011 | 10

iv. Penyilangan seragam (*uniform Crossover*)

Pada penyilangan seragam, setiap lokasi memiliki potensi sebagai tempat penyilangan. Sebuah *mask* penyilangan dibuat sepanjang panjang kromosoms secara random yang menunjukkan bit-bit dalam *mask* yang mana induk akan mensuplai anak dengan bit-bit yang ada. Induk mana yang akan menyumbangkan bit ke anak yang dipilih secara random dengan probabilitas yang sama. Misalkan ada 2 kromosom dengan panjang 12 :

Induk 1 : 011100101110

Induk 2 : 110100001101

Mask bit :

Sampel 1 : 100111001101

Sampel 2 : 011000110010

Setelah penyilangan, diperoleh kromosom-kromosom baru :

anak 1 : 010100001100

anak 2 : 111100101111

v. Penyilangan dengan permutasi (*permutation Crossover*)

Dengan teknik permutasi ini, kromosom-kromosom anak diperoleh dengan cara memilih sub-barisan suatu *tour* dari satu induk dengan tetap menjaga urutan dan posisi sejumlah kota yang mungkin terhadap induk yang lainnya. Sebagai contoh adalah :

Induk 1 : (1 2 3 | 4 5 6 7 | 8 9)

Induk 2 : (4 5 3 | 1 8 7 6 | 9 2)

Anak 1 : (x x x | 1 8 7 6 | x x)

Anak 2 : (x x x | 4 5 6 7 | x x)

Dari sini diperoleh hasil pemetaan :

1-4, 8-5, 7-6, 6-7.

Kemudian copy sisa gen di induk-1 ke anak-1 dengan menggunakan pemetaan yang sudah ada.

Anak 1 : (1-4 2 3 | 1 8 7 6 | 8-5 9)

Anak 2 : (4 2 3 | 1 8 7 6 | 5 9)

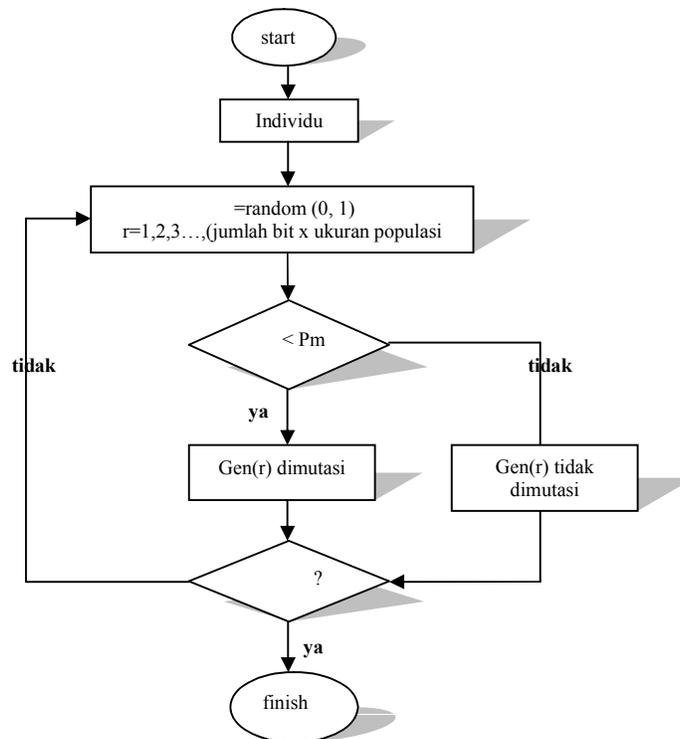
Lakukan hal yang sama untuk anak-2

Anak 1 : (4-1 5-8 3 | 4 5 6 7 | 9 2)

Anak 2 : (1 8 3 | 4 5 6 7 | 9 2)

2. Mutasi

Mutasi merupakan proses untuk mengubah nilai dari satu atau beberapa gen dalam suatu kromosom. Operasi *crossover* yang dilakukan pada kromosom dengan tujuan untuk memperoleh kromosom-kromosom baru sebagai kandidat solusi pada generasi mendatang dengan *fitness* yang lebih baik, dan lama-kelamaan menuju solusi optimum yang diinginkan. Akan tetapi, untuk mencapai hal ini, penekanan selektif juga memegang peranan yang penting. Jika dalam proses pemilihan kromosom-kromosom cenderung terus pada kromosom yang memiliki *fitness* yang tinggi saja, konvergensi prematur akan sangat mudah terjadi.



Gambar 2.9 Diagram alir proses mutasi

a) Mutasi dalam pengkodean Biner

Mutasi pada pengkodean biner merupakan operasi yang sangat sederhana. Proses yang dilakukan adalah menginversi nilai bit pada posisi tertentu yang dipilih secara acak (atau menggunakan skema tertentu) pada kromosom, yang disebut dengan *inverse bit*.

Tabel 2.6 Contoh mutasi pada pengkodean biner

Kromosom sebelum mutasi	1 0 0 1 0 1 1 1
Kromosom setelah mutasi	1 0 0 1 0 0 1 1

b) Mutasi dalam pengkodean Permutasi

Proses mutasi yang dilakukan dalam pengkodean biner dengan mengubah langsung bit-bit pada *kromosom* tidak dapat dilakukan pada pengkodean permutasi karena konsistensi urutan permutasi harus diperhatikan. Salah satu cara yang dapat dilakukan adalah dengan memilih dua posisi (*locus*) dari kromosom dan kemudian nilainya saling dipertukarkan.

Tabel 2.7 Contoh mutasi pada pengkodean permutasi

Kromosom sebelum mutasi	1 2 3 4 5 6 7 8 9
Kromosom setelah mutasi	1 2 7 4 5 6 3 8 9

c) Mutasi dalam pengkodean nilai

Pada pengkodean nilai hampir sama dengan yang dilakukan pada pengkodean biner, tetapi yang dilakukan bukan menginversikan nilai bit, serta penerapannya tergantung pada jenis nilai yang akan

digunakan. Sebagai contoh, untuk nilai riil proses mutasi dapat dilakukan seperti yang dilakukan pada pengkodean permutasi, dengan saling mempertukarkan nilai dua gen pada kromosom. Namun demikian, cara ini tidak menjamin adanya perbedaan pada populasi sehingga semua kromosom dapat dengan mudah mempunyai nilai yang sama, dan justru mempercepat terjadinya konvergensi prematur.

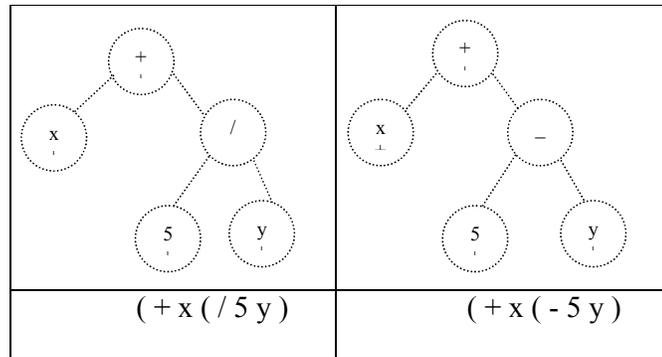
Cara lain yang lebih baik adalah dengan memilih sembarang posisi gen pada kromosom. Nilai yang ada tersebut kemudian ditambahkan atau dikurangkan dengan suatu nilai kecil tertentu yang diambil secara acak. Cara ini juga berlaku untuk pengkodean dengan bilangan bulat (cara mutasi lain yang relevan juga dapat digunakan).

Tabel 2.8 Contoh mutasi pada pengkodean nilai

Kromosom sebelum mutasi	1,43 1,09 4,51 9,11 6,94
Kromosom setelah mutasi	1,43 1,19 4,51 9,01 6,94

d) Mutasi dalam pengkodean pohon

Dalam metode ini dapat dilakukan dengan cara mengubah operator (+, -, *, /) atau nilai yang terkandung pada suatu vertex pohon yang dipilih, atau dengan memilih dua verteks pohon dan saling mempertukarkan operator atau nilainya (*Desiani dan Muhammad, 2006*).



Gambar 2.10 Contoh *genome* sebelum dan sesudah dimutasi

2.1.4.6 Paramater Kontrol

1. Kontrol parameter dalam Algoritma Genetika

Kontrol parameter dalam algoritma genetika diperlukan untuk mengendalikan operator-operator seleksi. Pemilihan parameter genetika menentukan penampilan kinerja algoritma genetika dalam memecahkan masalah. Ada dua parameter dasar dari algoritma genetika, yaitu *probabilitas crossover* (P_c), *probabilitas mutasi* (P_m) dan *ukuran populasi* (pop_size).

a) Probabilitas crossover (P_c)

Probabilitas *crossover* menyatakan seberapa sering proses *crossover* akan terjadi antara dua kromosom orang tua. Jika tidak terjadi *crossover*, satu orang tua dipilih secara random dengan probabilitas yang sama dan diduplikasi menjadi anak. Jika terjadi *crossover*, keturunan dibuat dari bagian-bagian kromosom orang tua. Jika probabilitas *crossover* 100% maka keseluruhan keturunan baru dibuat dengan *crossover*. Namun sebaliknya jika probabilitas *crossover* 0% maka seluruh generasi baru dibuat dari salinan kromosom-kromosom populasi lama yang belum tentu menghasilkan populasi

yang sama dengan populasi sebelumnya, karena adanya penekanan selektif.

Hasil penelitian yang sudah ada dan pernah dilakukan oleh praktisi algoritma genetika menunjukkan bahwa sebaiknya angka probabilitas crossover cukup tinggi, sekitar 80% sampai dengan 95% untuk memberikan hasil yang baik. Untuk beberapa permasalahan juga sebaiknya probabilitas crossover sekitar 60% bisa memberikan hasil yang baik.

b) Probabilitas mutasi (P_m)

Probabilitas mutasi menyatakan seberapa sering bagian-bagian kromosom akan dimutasikan. Jika tidak ada mutasi, keturunan diambil atau disalin langsung setelah *crossover* tanpa perubahan. Jika mutasi dilakukan, maka bagian-bagian tertentu kromosom diubah. Jika probabilitas mutasinya 100%, keseluruhan kromosom diubah. Jika probabilitas mutasi 0%, maka tidak ada yang diubah. Probabilitas mutasi dalam ruang algoritma genetik seharusnya diberi nilai yang kecil. Umumnya, probabilitas mutasi diset untuk mendapatkan rata-rata satu mutasi per kromosom, yaitu $\text{angka}/\text{allele} = 1/(\text{panjang kromosom})$. Dari hasil penelitian yang sudah pernah dilakukan, menunjukkan bahwa angka probabilitas terbaik adalah antara 0,5% sampai 1%. Hal ini karena tujuan mutasi ini sendiri adalah menjaga perbedaan kromosom dalam populasi, untuk menghindari terjadinya konvergensi prematur.

c) Ukuran populasi (pop_size)

Parameter lain yang juga ikut menentukan efisiensi kerja algoritma genetika adalah ukuran populasi, yaitu banyaknya kromosom dalam satu populasi. Jika terlalu sedikit kromosom dalam populasi, maka algoritma genetika mempunyai kemungkinan yang sedikit untuk melakukan *crossover* dan hanya sebagian kecil dari ruang pencarian yang dieksplorasi. Akan tetapi, jika terlalu banyak jumlah kromosom juga bisa membuat algoritma genetika cenderung menjadi lambat dalam menemukan solusi. Ukuran populasi yang sering digunakan oleh peneliti yang sudah ada adalah antara 20 sampai 30, tetapi terkadang ukuran 50 sampai 100 juga dilaporkan baik. Beberapa penelitian juga menunjukkan bahwa ukuran populasi yang baik tergantung dari pengkodean, yaitu ukuran dari barisan yang dikodekan.

2. Rekomendasi Untuk nilai parameter kontrol

Nilai parameter ini juga ditentukan berdasarkan permasalahan yang ada. Ada beberapa rekomendasi yang biasa digunakan, antara lain :
(Kusumadewi, 2003).

- i. Untuk permasalahan yang memiliki kawasan solusi cukup besar, De Jong merekomendasikan untuk nilai parameter kontrol :

$$(pop_size; pc; pm) = (50; 0,6; 0,001).$$

- ii. Bila rata-rata fitness setiap generasi digunakan sebagai indikator, maka Grefenstette merekomendasikan untuk nilai parameter kontrol :

$$(pop_size; pc; pm) = (30; 0,95; 0,01)$$

- iii. Bila fitness dari individu terbaik dipantau pada setiap generasi, maka usulannya adalah:

$$(pop_size; pc; pm) = (80; 0,45; 0,01)$$

Ukuran populasi sebaiknya tidak lebih kecil dari 30, untuk sembarang jenis permasalahan.

2.2 Model Regresi Statistik

Menurut Hardle (1989) suatu garis regresi menggambarkan sebuah hubungan antara variabel penjelas X dan variabel respon Y . jika ada T data observasi $\{(x_i, y_i)\}$ maka hubungan fungsi regresi secara umum dapat digambarkan dengan model sebagai berikut :

$$y_i = f(x_i) + \epsilon_i; i=1,2,3,\dots,T,$$

Dengan fungsi f adalah fungsi regresi yang tidak diketahui dan akan diestimasi, serta ϵ_i adalah error. Model regresi itu sendiri dibagi menjadi dua jenis, yaitu model regresi parametrik dan non-parametrik. Untuk model regresi parametrik dibagi lagi menjadi dua jenis yaitu model regresi linier dan non-linier.

2.2.1 Model regresi linier dan non-linier

Secara umum di dalam statistika ada dua macam regresi yaitu linier dan non-linier. Linier atau tidaknya suatu fungsi regresi dapat dijabarkan dalam dua cara sebagai berikut (Gujarati, 2003) :

- a. Linier dalam variabel

Yang pertama dan mungkin arti yang lebih alami dari linieritas adalah dugaan bersyarat dari Y adalah fungsi linier dari X_i , seperti bentuk

$$E(Y_i | X_i) = \beta_0 + \beta_1 X_i. \text{ Secara geometri, garis regresi dalam kasus ini}$$

adalah garis lurus. Dalam penggambaran ini, fungsi regresi seperti $(I) = +$ bukanlah sebuah fungsi linier karena variabel X muncul dengan kekuatan atau dengan pangkat 2.

b. Linier dalam parameter

Penggambaran kedua tentang linieritas adalah dugaan bersyarat dari Y adalah fungsi linier dari parameter, boleh linier ataupun tidak dalam variabel X. Dalam penjabaran ini $(I) = +$ adalah model regresi linier (dalam parameter). Untuk mengetahuinya, misal di ambil nilai X=3. Sehingga, $E(Y|X=3) = + 9$, yang benar-benar linier dalam dan. Beberapa contoh model regresi linier yang linier di dalam parameter antara lain regresi *quadratic*, *exponential*, dan *cubic*..

Selanjutnya untuk model $E(Y|X_i) = +$. Missal diambil X=3 kemudian didapat $E(Y|X_i) = + 3$, yang tidak linier dalam parameter. Model yang terakhir ini adalah model regresi non linier dalam parameter.

Dari dua penjabaran tentang linieritas diatas, secara skema bisa dilihat pada tabel tabel 2.9 berikut ini :

Tabel 2.9 Model Regresi

model linier dalam parameter?	model linier dalam variabel?	
	ya	tidak
ya	LRM	LRM
tidak	NLRM	NLRM

Keterangan: LRM (*Linier Regression model*)

NLRM (*Non-Linier Regression model*)

Namun demikian, dalam melihat model diperlukan ketelitian karena beberapa model bisa saja terlihat non-linier dalam parameter namun pada dasarnya adalah linier karena dengan transformasi yang tepat model dapat dijadikan model regresi linier dalam parameter. Namun jika ada model yang tidak dapat dilinierkan dalam parameter disebut model regresi non-linier dasar, dan salah satu bentuk fungsi regresi non-linier adalah fungsi produksi *cobb-douglas*.

2.2.2 Model Statistik Non-linear

Pada umumnya realitas situasi perekonomian dapat dilakukan pendekatan secara linier atau ditransformasikan dalam bentuk linier. Namun demikian, banyak juga model non-linier yang tidak bisa ditangani oleh model linier, oleh karena itu diperlukan model non-linier dalam pemecahannya. Tidak berbeda dengan model linier, estimasi model non-linier didasarkan pada minimisasi atau maksimisasi fungsi objektif. Berdasarkan teori, terdapat dua jenis fungsi objektif, yaitu *the sum squared errors* dan *the likelihood function*. Namun demikian, berbeda dengan model linier, memecahkan parameter model non-linier merupakan tugas yang sulit. Hal ini disebabkan banyaknya pertimbangan yang harus dilibatkan dalam proses menentukan titik optimum secara statis (*static optimization*). Pertimbangan-pertimbangan tersebut antara lain perlu atau tidaknya pembatas observasi (*constraint*) yang akan mendefinisikan letak titik optimum dan *sufficient conditions* untuk lokal atau global minimum. Penaksiran terhadap parameter model non-linear akan menghasilkan nilai yang berbeda untuk penaksir yang sama karena error random-nya mempunyai power function. Oleh karena itu, berbeda dengan *least square rule* pada model linear, penaksir atau estimator pada *least square rule* yang diterapkan pada model non-linear ditentukan dengan

melakukan suatu prosedur atau algoritma yang dapat menjamin bahwa penaksir tersebut secara nyata memenuhi kriteria dari fungsi tujuan, yaitu memberikan *the sum of squares error* pada titik yang paling minimum atau memberikan titik maksimum pada *likelihood function*.

2.2.3 Fungsi Produksi *Cobb-Douglas*

2.2.3.1 Definisi dan gambaran umum

Beberapa istilah umum yang digunakan dalam ekonomi produksi memiliki arti yang berbeda dari penggunaan umum, sehingga untuk menghindarkan terjadinya pengertian yang membingungkan perlu dipahami beberapa definisi sebagai berikut (*Beattie and Robert Taylor, 1994*) :

Definisi 1 : *Produksi*

Yaitu proses kombinasi dan koordinasi material-material dan kekuatan-kekuatan (input, faktor, sumber daya atau jasa-jasa produksi) dalam pembuatan suatu barang atau jasa (output atau produk). Kata *input* dan *output* hanya memiliki pengertian dalam hubungannya dengan proses produksi tertentu.

Definisi 2 : *Fungsi Produksi*

Yaitu sebuah deskripsi matematis atau kuantitatif dari berbagai macam kemungkinan-kemungkinan produksi teknis yang dihadapi oleh suatu perusahaan. Fungsi produksi memberikan output maksimum dalam pengertian fisik tiap-tiap tingkat input dalam pengertian fisik. Dalam spesifikasi multiproduksi sangat penting membedakan antara faktor-faktor variabel dan tetap.

Definisi 3 : *Fungsi Cobb-Douglas*

Fungsi *Cobb-Douglas* adalah suatu fungsi atau persamaan yang melibatkan dua atau lebih variabel, di mana variabel yang satu disebut variabel

independen, yang menjelaskan atau dengan simbol x sedangkan variabel dependen atau variabel yang dijelaskan dengan simbol y .

Definisi 4 : *Faktor-faktor Variabel*

Yaitu faktor-faktor produksi yang dapat berkurang selama suatu periode tertentu. *Faktor-faktor tetap* adalah faktor-faktor yang tidak akan berubah selama periode produksi. Maka dari itu, perbedaan antara faktor-faktor variabel dan tetap sangat tergantung pada lamanya periode produksi.

Definisi 5 : *Jangka Panjang (long run)*

Yaitu periode produksi manakala semua faktor-faktor dianggap n variabel, sedangkan satu hingga $(n-1)$ adalah variabel dalam *jangka pendek (short run)*. Berbagai peristiwa atau *event* ekonomi penting, khususnya berkenaan dengan pendekatan matematis untuk optimasi, kita tertarik pada “syarat yang diperlukan” (*necessary condition*) dan “syarat yang mencukupkan” (*sufficient condition*) untuk peristiwa tertentu.

Definisi 6 : *Syarat yang diperlukan*

Yaitu suatu keadaan yang menjamin terjadinya peristiwa yang dimaksud. Sebagai contoh dengan pendekatan kalkulus untuk optimasi, jika $y=f(x)$ syarat urutan pertama (*first order condition/FOC* adalah ($f'(x) = 0$)), dan syarat urutan kedua (*second order condition/SOC*) adalah ($f''(x) < 0$) masing-masing merupakan “syarat yang diperlukan untuk keadaan maksimum lokal bagi fungsi tak berkendala, $y=f(x)$. Secara bersama-sama keduanya merupakan “syarat yang mencukupkan”, yaitu jika ($dy/dx \neq 0$) atau ($f'(x) > 0$), solusi maksimumnya tidak terjadi, dan kalau ($f'(x) = 0$) dan ($f''(x) < 0$) pada x kritis (nilai x ,

dimana $\frac{\partial Q}{\partial L} = 0$), maksimum terjadi. Tapi jika $\frac{\partial Q}{\partial L} = 0$ dan $\frac{\partial Q}{\partial K} > 0$, minimum terjadi.

Ada beberapa kelebihan *fungsi cobb-douglas* bila dibandingkan dengan fungsi-fungsi produksi lainnya. Kelebihan-kelebihan tersebut antara lain :

- a) Bentuknya sederhana dan penyelesaian fungsi *Cobb-Douglas* relatif lebih mudah dibandingkan dengan fungsi produksi yang lain,
- b) Hasil yang diperoleh konsisten dengan teori, yaitu apabila penggunaan input terus ditambah, maka produk marginal dari input tersebut akan menurun,
- c) Hasil pendugaan garis melalui fungsi *Cobb-Douglas* akan memberikan koefisien regresi yang sekaligus juga menunjukkan besaran elastisitas produksi dari inputnya.

2.2.3.2 Bentuk *fungsi cobb-douglas*

Berikut merupakan bentuk fungsi produksi *cobb-douglas* yang terkenal. Anggap y =output, L =input tenaga kerja, K = input modal, u =error dan kemudian fungsi ini akan ditulis dalam tiga cara berbeda (*Gujarati, 2003*):

$$Q = A \cdot L^\alpha \cdot K^\beta \cdot u \dots\dots\dots (2.2.3.2.1)$$

Atau

$$\ln Q = \ln A + \alpha \ln L + \beta \ln K + \ln u \dots\dots\dots (2.2.3.2.1.1)$$

Ini dalam format fungsi *cobb-douglas* adalah linier dasar. Sekarang perhatikan versi fungsi *cobb-douglas* berikut ini:

$$Q = A \cdot L^\alpha \cdot K^\beta \cdot u \dots\dots\dots (2.2.3.2.2)$$

Atau

$$\ln Q = \ln A + \alpha \ln L + \beta \ln K + \ln u \dots\dots\dots(2.2.3.2.2.1)$$

Model ini juga linier dalam parameter.

Namun sekarang perhatikan versi fungsi *cobb-douglas* berikut ini:

$$= \dots + \dots \dots \dots (2.2.3.2.3)$$

Seperti yang bisa dilihat pada versi *cobb-douglas* persamaan (2.2.3.2.1.1) dan (2.2.3.2.2) adalah model regresi linier dasar dalam parameter. Namun demikian pada persamaan (2.2.3.2.3) tidak ada cara untuk mentransformasikannya agar model yang telah ditransformasikan dapat dibuat linier dalam parameter. Sehingga persamaan (2.2.3.2.3) adalah sebuah model regresi non-linier.

Penggunaan dari *fungsi produksi Cobb-Douglas* sangat dikenal oleh para ahli ekonomi yang menggunakan metode analisis kuantitatif dan banyaknya manfaat. Sebagai contoh, β dan β mengukur elastisitas tenaga dan modal terhadap output. Jumlah dari $(\beta + \beta)$ memberikan informasi mengenai *returns to scale*, yaitu besarnya reaksi output terhadap perubahan input secara proporsional. Kalau $\beta + \beta = 1$, maka akan ada *returns to scale* yang konstan, artinya kalau input menjadi dua kali, maka secara proporsional output akan berkurang dari dua kali. Akhirnya kalau $\beta + \beta > 1$, akan terjadi kenaikan *returns to scale*, artinya jika input menjadi dua kali, maka secara proporsional output akan menjadi lebih dari dua kali (Supranto, 2005).

Selain itu, menurut Soekartawi (1990), *return to scale* adalah hukum perluasan produksi yang terjadi karena keadaan teknologi (skala usaha) yang dipakai, ini terdiri dari 3 keadaan yaitu :

- a) *Increasing Return to scale*, apabila $(\beta + \beta) > 1$, artinya yaitu bila semua faktor produksi ditambah dengan kelipatan tertentu maka jumlah akan meningkat dengan kelipatan yang lebih besar

- b) *Constant Return to Scale* apabila $(\beta + \beta) = 1$, artinya yaitu jika jumlah produk meningkat sesuai dengan kelipatan peningkatan penggunaan input
- c) *Decreasing Return to Scale*, apabila $(\beta + \beta) < 1$, artinya jika kelipatan tambahan produk yang dihasilkan lebih kecil dari pada kelipatan kenaikan faktor produksi.