SUYANTO, ST, MSC

EVOLUTIONARY COMPUTATION

KOMPUTASI BERBASIS "EVOLUSI" DAN "GENETIKA"

Evolutionary Computation (EC) sangat berguna untuk menghadapi masalah optimasi, penjadwalan, prediksi, learning dll yang sangat kompleks dan anda merasa kesulitan untuk menemukan langkah-langkah sistematis (algoritma) untuk menyelesaikannya. Dengan Evolutionary Algorithms (Eas) anda tidak perrlu langkah sistematis dan analisa matematis khusus untuk menyelesaikan masalah, karena semua proses bersifat probabilistik. Eas sangat mudah diimplementasikan, fleksible dan memiliki reusability yang tinggi. Begitu anda bisa merepresentasikan solusi ke dalam kromosom dan berhasil membangun fungsi fitness-nya, maka anda bisa menggunakan operator-operator evolusi yang hampir sama untuk berbagai masalah yang berbeda-beda.

Penerbit INFORMATIKA

Evolutionary Computation

Komputasi Berbasis "Evolusi" dan "Genetika"

Suyanto, S.T., M.Sc.

INFORMATIKA
BANDUNG

Kata Pengantar

Evolutionary Computation (EC) merupakan suatu abstraksi yang mengadopsi "evolusi" dan "genetika" yang disederhanakan. Dari abstraksi tersebut, lahirlah berbagai algoritma berbasis EC yang dikenal sebagai *Evolutionary Algorithms* (EAs). Sejak diperkenalkan pada tahun 1960-an, EAs terus diteliti dan dikembangkan hingga saat ini. Yang menarik pada EAs adalah "Hanya dengan menggunakan proses-proses yang sebagian besar dilakukan secara acak, EAs bisa menghasilkan solusi yang "bagus" (mungkin saja solusi terbaik) dengan kecepatan yang dapat diterima". Beberapa ahli memfokuskan penelitian pada bagaimana membangun analisis matematika yang bisa memberikan bukti sangat kuat bagaimana EAs bisa menghasilkan solusi yang "bagus" atau bahkan yang terbaik. Sebagian ahli dan praktisi lebih fokus pada bagaimana mengaplikasikan EAs untuk menyelesaikan beragam masalah sehari-hari yang semakin kompleks. Di masa depan, EAs menjadi teknik komputasi pamungkas yang diharapkan bisa menyelesaikan berbagai masalah komputasi yang sangat kompleks dengan ruang masalah yang sangat besar.

Pada buku ini, pembaca bisa menemukan bagaimana proses pengadopsian "evolusi" dan "genetika" sederhana ke dalam dunia komputer yang disebut EAs. Terdapat enam algoritma EAs yang dibahas dalam buku ini, yaitu: *Genetic Algorithm* (GA), *Evolution Strategies* (ES), *Evolutionary Programming* (EP), dan *Genetic Programming* (GP), *Differential Evolution*

(DE), dan *Grammatical Evolution* (GE). Pembahasan dilakukan secara detail dari ide-ide dasarnya, perbedaannya, bagaimana teknik implementasinya dan beragam studi kasus dalam pengaplikasiannya.

Buku ini disusun dalam 11 bab. Bab pertama membahas sejarah EC, kondisi EC saat ini dan masa depan. Bab 2 membahas apa itu EAs, mengapa, kapan, di mana, dan bagaimana menggunakan EAs. Bab 3 sampai Bab 8 membahas enam algoritma EAs. Suatu masalah penting dalam EAs, yaitu konvergensi prematur, dan bagaimana mencegahnya dibahas pada Bab 9. Untuk memahami EAs secara lebih mendalam dan sebagai latihan, Bab 10 menyediakan beragam ide mengenai strategi menggunakan EAs dan beberapa studi kasus. Sebagai penutup, Bab 11 mengulas kembali hal-hal penting dalam buku ini sebagai kesimpulan. Bagi pembaca yang belum mengenal EC, disarankan untuk terlebih dulu membaca Bab 1 sampai Bab 3 secara berurutan. Selanjutnya, Bab 4 sampai 8 bisa dibaca secara acak, tidak harus berurutan. Setelah itu, pembaca bisa melanjutkan ketiga bab terakhir. Pembaca yang sudah pernah mempelajari EC bisa membaca buku ini secara acak pada bab-bab yang dibutuhkan.

Penulis mengucapkan terima kasih yang sebesar-besarnya kepada sivitas akademika Institut Teknologi Telkom (dulu STT Telkom) khususnya dan kepada semua pihak yang telah mendukung penulisan buku ini. Penulis sangat mengharapkan koreksi, kritik, dan saran dari para pembaca yang bisa disampaikan melalui *email* suy@stttelkom.ac.id atau suyanto@student.chalmers.se atau melalui pos ke alamat: Departemen Teknik Informatika, Institut Teknologi Telkom, Jl. Telekomunikasi, Dayeuh Kolot, Bandung 40257. Semoga pembaca mendapatkan manfaat dari buku ini.

Bandung, 15 Jan 2008

Suyanto, S.T., M.Sc.
www.stttelkom.ai.id/staf/suy

Daftar Isi

KATA PENGANTAR iii
DAFTAR ISI v
DAFTAR GAMBAR ix
DAFTAR TABEL xxi

1. Pendahuluan 1
   1.1 "Evolusi" dan "Genetika" 1
   1.2 Posisi *Evolutionary Computation* 6
   1.3 Sejarah EC 6
   1.4 EC Saat Ini 8
   1.5 EC Masa Depan 9

2. Memahami *Evolutionary Algorithms* 11
   2.1 Apa Itu EAs? 11
   2.2 Mengapa Kita Membutuhkan EAs? 20
   2.3 Kapan EAs Digunakan? 24
   2.4 Dimana EAs Digunakan? 25
   2.5 Bagaimana Menggunakan EAs? 26

3. *Genetic Algorithms* 27
   3.1 Simple GA 27
   3.2 Studi Kasus: Masalah Optimasi 34
      3.2.1 Optimasi Fungsi 34
      3.2.2 Optimasi Pengepakan Barang 40

3.3 Representasi Individu 43
3.3.1 Representasi Biner 43
3.3.2 Representasi *Integer* 48
3.3.3 Representasi *Real* 50
3.3.4 Representasi Permutasi 51
3.4 Nilai *Fitness* 54
3.5 Seleksi Orang tua 57
3.5.1 *Fitness Proportionate Selection* (FPS) 57
3.5.2 *Rank-Based Selection* 65
3.5.2.1 Linear *Ranking* 65
3.5.2.2 Non-linear *Ranking* 65
3.5.3 *Tournament Selection* 67
3.6 Rekombinasi 68
3.6.1 Rekombinasi untuk Representasi Biner 69
3.6.2 Rekombinasi untuk Representasi *Integer* 71
3.6.3 Rekombinasi untuk Representasi *Real* 72
3.6.4 Rekombinasi untuk Representasi Permutasi 74
3.6.4.1 *Order crossover* 75
3.6.4.2 *Partially Mapped Crossover* 77
3.6.4.3 *Cycle Crossover* 79
3.6.4.4 *Edge Recombination* 81
3.6.5 Rekombinasi *Path Relinking* 83
3.6.6 Rekombinasi *Multi-parent* 84
3.7 Mutasi 86
3.7.1 Mutasi untuk Representasi Biner 90
3.7.2 Mutasi untuk Representasi *Integer* 91
3.7.3 Mutasi untuk Representasi *Real* 93
3.7.4 Mutasi untuk Representasi Permutasi 94
3.8 Seleksi *Survivor* 95
3.8.1 *Generational Model* 95
3.8.2 *Steady State Model* 96
3.8.3 Metode Seleksi *Survivor* 97
3.9 Pembuktian GA secara Matematis 99
3.10 Aplikasi GA 103
3.11 Studi Kasus 104

4. *Evolution Strategies* 111
4.1 ES pada Masa Awal 111
4.2 Representasi Individu 114
4.3 Seleksi Orang tua 115

4.4 Rekombinasi 115
4.5 Mutasi 115
4.6 Seleksi *Survivor* 116
4.7 *Self-Adaptation* 119
4.8 Proses Evolusi 118
4.9 Aplikasi ES 120

5. *Evolutionary Programming* 121
5.1 EP Tradisional 121
5.2 Representasi Individu 124
5.3 Seleksi Orang tua 124
5.4 Rekombinasi 124
5.5 Mutasi 125
5.6 Seleksi *Survivor* 125
5.7 Aplikasi EP 126

6. *Genetic Programming* 127
6.1 Apa itu GP? 127
6.2 Proses Evolusi 128
6.3 Representasi Individu 129
6.4 Seleksi Orang tua 135
6.5 Rekombinasi 135
6.6 Mutasi 136
6.7 Seleksi Survivor 137
6.8 Inisialisasi 137
6.9 Aplikasi GP 138

7. *Differential Evolution* 139
7.1 Apa itu DE? 139
7.2 Masalah Optimasi 141
7.3 DE untuk Optimasi 142
7.3.1 Skema DE1 143
7.3.2 Skema DE2 147
7.4 Performansi DE 148
7.4.1 Minimum Global dan Kecepatan Konvergensi 148
7.4.2 Kemudahan Menentukan Nilai Parameter 152

8. *Grammatical Evolution* 155
8.1 Apa itu GE? 155
8.2 *Backus Naur Form* (BNF) 156
8.3 Representasi Individu 158
8.4 Operator Evolusi 162
8.4.1 *Duplicate* 162
8.4.2 *Prune* 164
8.5 Performansi GE 165

9. Konvergensi Prematur dan Pencegahannya 169
9.1 Konvergensi Prematur 169
9.2 *Island Model* EAs 171
9.3 Representasi Rumit (*Messy Encoding*) 173
9.4 *Adaptive* EAs 178

10. Studi Kasus 183
10.1 Strategi Menggunakan EAs 183
10.1.1 Kenali Masalah yang Anda Hadapi 184
10.1.1.1 Ruang Masalah 184
10.1.1.2 Analisis Matematika 186
10.1.1.3 Jenis Solusi 186
10.1.1.4 Batasan Waktu 187
10.1.1.5 Biner, *Real*, Automata atau Program? 187
10.1.1.6 Nilai Optimum dan Permutasi 187
10.1.1.7 Pengetahuan untuk masalah yang dihadapi 188
10.1.2 *Setting* Parameter EAs 188
10.1.2.1 Observasi Parameter 188
10.1.2.2 Pengontrolan Parameter secara Adaptif 190
10.2 *Graph Bisection* 190
10.3 Masalah 8-*queens* 193
10.4 TSP dengan Batasan 196
10.4.1 Inisialisasi Populasi 198
10.4.2 Rekombinasi 199
10.4.3 Mutasi 199
10.4.4 Fungsi *Fitness* 200
10.5 Penjadwalan Kuliah 201
10.5.1 Representasi Individu 202
10.5.2 Rekombinasi 204
10.5.3 Mutasi 206
10.5.4 Fungsi *Fitness* 207

10.6 Pemotongan Bahan 209
10.6.1 Representasi Individu 213
10.6.2 Rekombinasi 216
10.6.3 Mutasi 217
10.6.4 Fungsi *Fitness* 218
10.7 Peramalan untuk Data *Time Series* 219
10.7.1 Representasi Individu 220
10.7.2 Fungsi *Fitness* 222

11. Penutup 225

INDEKS 229
DAFTAR PUSTAKA 233

Daftar Gambar

Gambar 1-1	Satu sel tubuh manusia yang berisi 23 pasang kromosom	4
Gambar 1-2	*Crossover* selama proses *meiosis* pada sepasang kromosom.	4
Gambar 1-3	Proses pembuahan (*fertilisation*) pada sel manusia.....	5
Gambar 2-1	Skema umum EAs. Evolusi yang terjadi pada individu-individu dalam populasi digambarkan oleh siklus seleksi orangtua, rekombinasi dan mutasi, serta seleksi *survivor* (individu yang *survive* pada generasi berikutnya). Evolusi berhenti ketika Kondisi Terminasi sudah terpenuhi.	13
Gambar 2-2	*Pseudo-code* EAs dengan *Steady State Replacement.* Penggantian kromosom dilakukan setiap kali dihasilkan dua anak.	14
Gambar 2-3	*Pseudo-code* EAs dengan *Generational Replacement.* Penggantian kromosom dilakukan sekaligus	14
Gambar 2-4	Representasi kromosom menggunakan *binary encoding.* Kromosom tersebut mengkodekan dua variabel, x_1 dan x_2, masing-masing 10 *bit* (*binary digit*).....	15
Gambar 2-5	Gen, kromosom, *genome*, dan *genotype*.....	16
Gambar 2-6	Contoh konversi *phenotype* ke dalam *genotype*.....	16
Gambar 2-7	Pandangan Goldberg yang disampaikan pada tahun 1989. Performansi EAs lebih baik dibandingkan *random search,*	

tetapi lebih buruk daripada metode yang dirancang untuk masalah khusus. 23

Gambar 2-8 Pandangan Michalewicz yang disampaikan pada tahun 1996. Performansi EAs menjadi beragam bergantung pada *problem specific knowledge* yang ditambahkan padanya. 24

Gambar 3-1 *Pseudo-code* SGA. 29

Gambar 3-2 Representasi kromosom SGA menggunakan pengodean biner (*binary encoding*). Individu-individu yang berada di dalam *phenotype space* dikodekan ke dalam kromosom-kromosom biner di dalam *genotype space*. Begitu juga sebaliknya. 29

Gambar 3-3 Seleksi kromosom sebagai orangtua menggunakan algoritma *roulette-wheel*. Kromosom dengan *fitness* lebih besar menempati ruangan lebih besar dalam roda *roulette* sehingga memiliki peluang lebih besar untuk terpilih sebagai orang tua. 30

Gambar 3-4 Rekombinasi dengan satu titik pada posisi ke-2 (a) dan rekombinasi dengan dua titik, posisi ke-2 dan ke-5 (b). 32

Gambar 3-5 Rekombinasi seragam (*uniform*). Suatu pola, bernilai 1 atau 0, dibangkitkan secara acak. Gen-gen orang tua dipertukarkan ketika pola bernilai 1. 33

Gambar 3-6 Representasi kromosom untuk mengodekan x_1 dan x_2. masing-masing variabel dikodekan ke dalam 4 *bit*. 35

Gambar 3-7 Populasi pada generasi ke-1, di mana ukuran populasi sama dengan 4 (artinya, populasi berisi 4 kromosom). . 36

Gambar 3-8 Populasi dengan 4 kromosom (kiri) dan *mating pool* yang berisi kromosom yang terpilih sebagai orang tua (kanan). 37

Gambar 3-9 Populasi pada generasi ke-2. Pada SGA, ukuran populasi dibuat tetap sama dengan generasi sebelumnya (berisi 4 kromosom). Pada generasi ke-2 ini dihasilkan satu kromosom terbaik dengan nilai *fitness* sebesar 105, yaitu kromosom K1. 38

Gambar 3-10 Representasi kromosom untuk masalah pengepakan barang. K1 adalah kromosom yang valid karena memberikan total berat 17 kg, sedangkan K2 tidak *valid* karena memberikan total berat 21 kg (> 20 kg). 41

Gambar 3-11 Visualisasi fungsi Ackley untuk kasus 2 dimensi (a). Visualisasi pada area yang mendekati nilai minimum (b) [LAB07]..... 45
Gambar 3-12 Representasi biner untuk bilangan bulat 14, 15, dan 16. *Hamming distance* antara 14 dan 15 adalah 1. Sedangkan *Hamming distance* antara 15 dan 16 adalah 5..... 46
Gambar 3-13 Contoh representasi *gray coding* dan representasi biner dalam 3 bit untuk mengodekan bilangan bulat dari 0 sampai 7..... 47
Gambar 3-14 Pendekodean kromosom (dalam *gray coding*) menjadi individu yang berupa *integer*..... 47
Gambar 3-15 Masalah pengepakan gulungan kain ekspor yang direpresentasikan dengan kromosom *integer*. Terdapat *m* ukuran gulungan kain (G_1 sampai G_m) dan *n* ukuran kardus (K_1 sampai K_n). Pada kromosom tersebut, nilai-nilai gen pada posisi G_i dan K_j menyatakan jumlah gulungan berukuran G_i yang dimasukkan ke dalam kardus berukuran K_j..... 48
Gambar 3-16 Pendekodean kromosom *integer* menjadi individu bilangan real dalam interval [-1, 2]..... 49
Gambar 3-17 Pendekodean kromosom *real* menjadi individu bilangan *real* dalam interval [-1, 2]..... 51
Gambar 3-18 Contoh kromosom permutasi yang mengodekan masalah TSP dengan 6 lokasi yang diberi nomor 1 sampai 6. Kromosom tersebut menyatakan satu solusi yang mugkin berupa urutan kunjungan 1-5-2-6-3-4..... 52
Gambar 3-19 Contoh kromosom permutasi yang mengodekan masalah TSP dengan enam lokasi yang diberi nama A, B, C, D, E, dan F. Kromosom tersebut menyatakan satu solusi yang mugkin yang berupa urutan kunjungan A-E-B-F-C-D..... 53
Gambar 3-20 Algoritma *Roulette Wheel* memilih 4 kromosom sebagai orang tua menggunakan roda *roulette* 1 jarum. Karena memerlukan 4 kali pemutaran, ada kemungkinan kromosom paling fit (K4) tidak pernah terpilih. Padahal K4 memiliki nilai ekspektasi terpilih sama dengan 2..... 58
Gambar 3-21 Algoritma Baker's SUS memilih 4 kromosom sebagai orang tua menggunakan roda *roulette* dengan 4 jarum. Karena keempat jarum diletakkan secara merata dan

hanya memerlukan 1 kali pemutaran roda, maka dijamin bahwa kromosom/individu *i* akan terpilih sebanyak *floor*(*E*(*ni*)) atau *ceil*(*E*(*ni*)) 59

Gambar 3-22 Algoritma Baker's SUS untuk memilih 4 kromosom sebagai orang tua. Ada kemungkinan masing-masing kromosom terpilih sebanyak 1 kali karena memiliki nilai-nilai *fitness* yang hampir sama. 60

Gambar 3-23 Algoritma Baker's SUS untuk memilih 4 kromosom sebagai orang tua. Ada kemungkinan kromosom K4 terpilih sebanyak 4 kali karena nilai *fitness*-nya jauh lebih besar dibandingkan 3 kromosom lainnya. 61

Gambar 3-24 Penskalaan nilai *fitness* menggunakan *Linear Scaling* dengan C_{mult} = 2. *Fitness* awal maksimum diskalakan menjadi dua kali rata-rata *fitness* awal..... 62

Gambar 3-25 Penskalaan nilai *fitness* menggunakan *Linear Scaling* bisa menghasilkan nilai *fitness* negatif. 63

Gambar 3-26 Penskalaan menggunakan *Windowing*. Misalkan *window* = 5 dan *fitness* terendah selama 5 generasi terakhir adalah 1,97. *Windowing* menghasilkan nilai-nilai *fitness* baru yang lebih bervariasi. 64

Gambar 3-27 Grafik perbedaan antara *Linear Ranking* (LR) dan *Non-Linear Ranking* (NLR)..... 66

Gambar 3-28 Seleksi orangtua menggunakan *Tournament Selection* dengan ukuran turnamen sama dengan 5. Pertama, ambil secara acak 5 kromosom sebagai kontestan untuk mengikuti turnamen pemilihan orangtua. Selanjutnya, pilih satu kontestan sebagai pemenang (orang tua) menggunakan cara tertentu. 67

Gambar 3-29 Proses *crossover* pada kromosom makhluk hidup. Dua kromosom orangtua bersilangan pada satu titik (A) dan kemudian menghasilkan dua kromosom baru yang masing-masing mewarisi gen-gen orang tuanya (B). *Crossover* bisa juga terjadi pada lebih dari satu titik seperti diilustrasikan oleh gambar C, D dan E. 68

Gambar 3-30 Rekombinasi satu titik potong pada posisi 2. 69

Gambar 3-31 Rekombinasi dua titik potong (a) dan tiga titik potong (b). 70

Gambar 3-32 Rekombinasi seragam. Suatu pola dibangkitkan secara acak dengan cara seperti pelemparan koin. Jika pola bernilai 1 maka pewarisan gen dilakukan secara bersilangan. 71
Gambar 3-33 Rekombinasi untuk representasi *integer*: rekombinasi satu titik (a), rekombinasi dua titik (b) dan rekombinasi seragam (c)..... 71
Gambar 3-34 *Single arithmetic crossover* dengan $\alpha = 0{,}7$ 73
Gambar 3-35 *Simple arithmetic crossover* dengan $\alpha = 0{,}5$ 73
Gambar 3-36 *Whole arithmetic crossover* dengan $\alpha = 0{,}5$ 74
Gambar 3-37 Rekombinasi satu titik (a), rekombinasi dua titik (b) dan rekombinasi seragam (c) menghasilkan anak-anak yang tidak *valid*. Pada gambar (a) Anak 1, gen bernilai 1dan 5 muncul dua kali yang berarti lokasi nomor 1 dan 5 dikunjungi dua kali. Sebaliknya, gen bernilai 2 dan 3 tidak pernah muncul yang berarti lokasi nomor 2 dan 3 tidak pernah dikunjungi. 75
Gambar 3-38 Rekombinasi menggunakan metode *Order Crossover*. Kedua kromosom anak yang dihasilkan dijamin valid, tidak ada dua gen yang bernilai sama..... 76
Gambar 3-39 Algortima *Order Crossover*. 77
Gambar 3-40 Rekombinasi menggunakan metode *Partially Mapped Crossover*. Kedua kromosom yang dihasilkan, Anak 1 (a) maupun Anak 2 (b), dijamin valid. 78
Gambar 3-41 Algortima *Partially Mapped Crossover*..... 79
Gambar 3-42 Rekombinasi menggunakan metode *Cycle Crossover*. Kedua kromosom anak yang dihasilkan selalu *valid*..... 80
Gambar 3-43 Algortima *Cycle Crossover*. 81
Gambar 3-44 Rekombinasi menggunakan metode *Edge Crossover*. Anak yang dihasilkan selalu *valid*..... 82
Gambar 3-45 Algortima *Edge Recombination*. 83
Gambar 3-46 Pembangkitan *n* anak dari dua dua orang tua ($n = 4$). Keempat anak memiliki perbedaan secara berurutan sehingga mirip suatu jalur. 84
Gambar 3-47 Terdapat *n* anak yang memiliki nilai *fitness* berbeda. Anak 3 dan Anak *n*, yang memiliki *fitness* tertinggi, terpilih untuk dikopikan ke generasi berikutnya..... 84
Gambar 3-48 Rekombinasi *Multi-parent* dengan 3 orangtua berdasarkan frekuensi *allele*. Dua anak yang dihasilkan mewarisi gen-

gen dari ketiga orangtuanya. 85
Gambar 3-49 Rekombinasi *Multi-parent* dengan 3 orangtua berdasarkan segmentasi dan rekombinasi. Dua anak yang dihasilkan mewarisi gen-gen dari ketiga orang tuanya secara diagonal. 85
Gambar 3-50 Kiri: seekor lalat buah (*drosophila*) normal. Kanan: seekor lalat buah dengan kaki tumbuh di kepala; mutasi akibat radiasi [ADN07]. 87
Gambar 3-51 Efek mutasi yang mengenaskan pada tubuh manusia. Bocah laki-laki di samping adalah korban kecelakaan instalasi nuklir Chernobyl [ADN07]. 87
Gambar 3-52 Mutasi satu gen (pada posisi pertama) menghasilkan kromosom baru yang merupakan solusi terbaik (maksimum). 89
Gambar 3-53 Mutasi satu gen (pada posisi ke dua) menghasilkan kromosom baru yang lebih buruk daripada kromosom awal..... 89
Gambar 3-54 Contoh mutasi untuk representasi biner. 91
Gambar 3-55 Contoh mutasi untuk representasi *integer* menggunakan pembalikkan nilai integer. 91
Gambar 3-56 Contoh mutasi untuk representasi *integer* menggunakan pemilihan nilai secara acak. Gen pada posisi 3 yang bernilai 1 terpilih untuk dimutasi. Pembangkitan nilai gen baru secara acak menghasilkan nilai 5. Jadi, gen pada posisi 3 dimutasi dari 1 menjadi 5. 92
Gambar 3-57 Distribusi yang simetris pada 0 dan memiliki probabilitas lebih tinggi pada bilangan yang lebih kecil. Di sini maksimum perubahan nilai adalah 3 sehingga besarnya perubahan nilai berada dalam interval [-3, 3]. 92
Gambar 3-58 Contoh mutasi untuk representasi *integer* menggunakan metode Creep. Gen pada posisi 3 yang bernilai 1 terpilih untuk dimutasi. Pembangkitan bilangan kecil secara acak berdasarkan distribusi tertentu menghasilkan nilai 1. Jadi, gen pada posisi 3 dimutasi dari 1 menjadi 2 (yang diperoleh dari 1 + 1). 92
Gambar 3-59 Mutasi pertukaran (*Swap Mutation*). Pemilihan dua posisi gen secara acak menghasilkan posisi 2 dan 7. Sehingga gen bernilai 5 dipertukarkan dengan gen bernilai 8. 94
Gambar 3-60 Mutasi penyisipan (*Insert Mutation*). Pemilihan dua posisi

gen secara acak menghasilkan posisi 2 dan 7. Kemudian gen bernilai 8 disisipkan setelah gen bernilai 5..... 94
Gambar 3-61 Mutasi pengacakan (*Scramble Mutation*). Pemilihan segmen kromosom menghasilkan {5, 2, 7, 6}. Kemudian pengacakan gen dalam segmen menghasilkan {2, 6, 5, 7}. 94
Gambar 3-62 Mutasi pembalikan (*Inversion Mutation*). Pemilihan segmen kromosom menghasilkan {5, 2, 7, 6}. Kemudian pembalikan posisi gen dalam segmen tersebut menghasilkan {6, 7, 2, 5}..... 95
Gambar 3-63 Populasi dengan *Generational model*. *N* kromosom baru menggantikan *N* kromosom lama pada generasi ..... sebelumnya..... 96
Gambar 3-64 Populasi dengan *Steady State model*. Hanya sejumlah *M* kromosom baru yang menggantikan *M* kromosom lama. 97
Gambar 3-65 Tiga *schema* (000, 01x, dan 1xxx) yang diilustrasikan dengan ruang tiga dimensi. 102
Gambar 3-66 Populasi dengan 50 individu pada generasi ke-10. Semua individu memiliki *fitness* yang jauh dari *fitness* tertinggi (100)..... 106
Gambar 3-67 Populasi dengan 50 individu pada generasi ke-10. Terdapat individu dengan *fitness* yang mendekati *fitness* tertinggi (100)..... 109
Gambar 3-68 Populasi dengan 50 individu pada generasi ke-100. Terdapat semakin banyak individu dengan *fitness* yang mendekati *fitness* tertinggi, yaitu 100..... 110
Gambar 4-1 *Pseudo-code* ES. 112
Gambar 4-2 Distribusi Normal atau Gaussian. 113
Gambar 4-3 Representasi kromosom yang lengkap untuk ES. 114
Gambar 4-4 Masalah optimasi fungsi yang berubah-ubah pada setiap periode sejumlah generasi tertentu. Fungsi dengan pada periode (kiri) dan pada periode berikutnya (kanan)..... 119
Gambar 5-1 Suatu contoh FSM dengan 3 *state*, 2 *input*, dan 3 *output*. 122
Gambar 5-2 FSM dengan 4 *state*, 2 *input*, dan 2 *output*..... 123
Gambar 6-1 *Flowchart* proses evolusi pada GA (kiri) dan pada GP (kanan)..... 129
Gambar 6-2 Suatu contoh *tree* yang digunakan untuk merepresentasikan model IF (IPK > 3,00 AND Psikologi > 83) THEN Diterima ELSE Tidak..... 130

Gambar 6-3 Pohon yang merepresentasikan suatu formula aritmetika. 131
Gambar 6-4 Pohon yang merepresentasikan suatu formula logika. 131
Gambar 6-5 Pohon yang merepresentasikan suatu program..... 132
Gambar 6-6 Pemetaan kromosom (representasi *integer*) ke dalam program menggunakan JB *mapping*. 133
Gambar 6-7 Contoh operasi Rekombinasi. Kedua anak memiliki ukuran yang berbeda. Anak 1 lebih besar dibandingkan kedua orang tuanya. 136
Gambar 6-8 Contoh operasi mutasi. Suatu *subtree* dipilih secara acak, misalkan seperti yang diberi tanda abu-abu (kiri). Kemudian *subtree* tersebut diganti dengan *subtree* baru yang dibangkitkan secara acak (kanan). 136
Gambar 6-9 Inisialisasi menggunakan *Full method* dengan $D_{max} = 4$. Setiap cabang selalu memiliki kedalaman 4. 137
Gambar 6-10 Inisialisasi menggunakan *Grow method* dengan $D_{max} = 4$. Suatu cabang tidak selalu memiliki kedalaman 4. 138
Gambar 7-1 Proses *differential mutation* yang menghasilkan vektor baru v pada skema DE1 untuk kasus fungsi dua dimensi berbentuk *contour*. 144
Gambar 7-2 Proses *crossover* untuk $D = 7$, $n = 3$, dan $L = 2$..... 146
Gambar 7-3 Proses *crossover* untuk $D = 7$, $n = 5$, dan $L = 3$. Karena $n+L-1$ (7) lebih besar dari j maksimum (6), maka indeks j kembali sama dengan 0..... 146
Gambar 7-4 Proses *differential mutation* yang menghasilkan vektor baru v pada skema DE2 untuk kasus fungsi dua dimensi berbentuk *contour*. 147
Gambar 8-1 Sistem *Grammatical Evolution*..... 158
Gambar 8-2 Perbandingan *Grammatical Evolution* dengan Biologi di alam (dunia nyata)..... 159
Gambar 8-3 Suatu contoh individu GE dengan 8 gen integer yang merupakan hasil transkripsi dari kromosom biner. 159
Gambar 8-4 Kromosom GE dengan 8 gen yang tidak *valid* sehingga akan menghasilkan program yang tidak lengkap. 162
Gambar 8-5 Kromosom 16 gen yang dihasilkan dari pengaplikasian operator *duplicate*..... 163
Gambar 8-6 Rekombinasi yang tidak berguna akibat banyaknya gen yang tidak digunakan dalam translasi. Kedua anak yang dihasilkan jika ditranslasi akan menghasilkan program atau

fungsi yang sama dengan kedua orang tuanya..... 165
Gambar 8-7 Frekuensi kumulatif GE menemukan solusi untuk masalah *Symbolic Regression*. 166
Gambar 8-8 Frekuensi kumulatif GE menemukan solusi untuk masalah *Trigonometric Identities*. 167
Gambar 8-9 Frekuensi kumulatif GE menemukan solusi untuk masalah *Symbolic Integration*..... 168
Gambar 9-1 Konvergensi prematur. Individu-individu tersebar dengan nilai *fitness* rendah, tetapi individu A memiliki nilai *fitness* paling tinggi dibanding yang lain (kiri). Setelah beberapa generasi, populasi konvergen pada suatu solusi optimum lokal, dekat dengan individu A. 170
Gambar 9-2 EAs berbasis *island model* dengan 4 sub populasi. Setiap sub populasi ber-evolusi dan kemungkinan bisa konvergen pada optimum lokal maupun optimum global..... 171
Gambar 9-3 Pertukaran individu antar subpopulasi yang menggunakan interaksi melingkar. 173
Gambar 9-4 Representasi individu menggunakan *messy encoding* (atas) dan hasil penerjemahannya ke dalam *binary encoding* (bawah). 174
Gambar 9-5 Rekombinasi bisa menghasilkan dua kromosom baru yang tidak valid. Pada Anak 1, tiga posisi gen (2, 3 dan 5) muncul dua kali dengan nilai gen yang berbeda. Sebaliknya posisi gen 4, 6, dan 7 tidak pernah muncul sama sekali..... 175
Gambar 9-6 Kromosom *messy encoding* dengan panjang dua kali semula. Kedua masalah akibat rekombinasi bisa diselesaikan. 176
Gambar 9-7 Proses mutasi pada representasi *messy encoding* sama dengan yang ada pada representasi biner maupun *integer*. 177
Gambar 9-8 Mutasi gen pada *messy encoding*. Gen ke-1, yang berada di posisi 14 bisa mengalami mutasi..... 178
Gambar 9-9 Interaksi antara EAs dengan *Fuzzy Government*..... 179
Gambar 10-1 Suatu *graph* dengan 6 *node*. 191
Gambar 10-2 Tiga contoh pembagian *graph* menjadi dua sub-*graph* yang menghasilkan perbedaan jumlah busur yang terpotong. Cara P1 memotong 3 busur, P2 memotong 5 busur, dan P3 memotong 4 busur..... 191

Gambar 10-3 Suatu kromosom untuk masalah *graph bisection* yang berupa representasi permutasi (bawah) dan hasil dekodenya (atas). 192
Gambar 10-4 Masalah 8 *queens*. Kemungkinan langkah dari sebuah *queen* (kiri). Suatu kemungkinan posisi 8 *queens* di atas papan catur (kanan). 194
Gambar 10-5 Suatu kromosom untuk masalah 8-*queens* yang berupa representasi permutasi (bawah) dan hasil dekodenya (atas)..... 195
Gambar 10-6 Tiga contoh kromosom yang mungkin bisa dibangkitkan menggunakan aturan di atas dengan nilai probabilitas P_k yang berbeda: (a) $P_k = 0$, (b) $P_k = 1$, dan (c) P_k antara 0 sampai 1. 199
Gambar 10-7 Proses rekombinasi menggunakan *order crossover* dengan batasan bahwa peletakkan kota harus mengikuti aturan seperti pada sub sub bab 10.4.1. 199
Gambar 10-8 Dua contoh proses *swap mutation*. (a) Suatu gen (posisi P1) yang berupa kota di benua SA dipertukarkan dengan gen lainnya yang berupa kota di benua SA juga (posisi P2). (b) Contoh lainnya untuk kota di benua NA 200
Gambar 10-9 Suatu representasi kromosom untuk masalah penjadwalan kuliah dengan menggunakan 36 gen. Setiap gen menyatakan ruangan..... 202
Gambar 10-10 Konversi kromosom dengan 36 gen ke dalam kromosom menggunakan 400 gen. Setiap gen menyatakan nomor urut pertemuan kuliah..... 203
Gambar 10-11 Rekombinasi 1-*point crossover* pada kromosom 3 dimensi (dengan 36 gen). Kedua Anak yang dihasilkan tidak *valid* karena beberapa pertemuan kuliah dijadwalkan lebih dari satu kali..... 204
Gambar 10-12 Rekombinasi *order crossover* pada kromosom 2 dimensi (dengan 400 gen). Kedua Anak yang dihasilkan selalu *valid*. 205
Gambar 10-13 Mutasi pada kromosom 3 dimensi..... 206
Gambar 10-14 *Swap mutation* pada kromosom 2 dimensi (dengan 400 gen). 206
Gambar 10-15 Mutasi pada kromosom 2 dimensi (dengan 400 gen) yang dilakukan dengan mengubah suatu gen terpilih dengan

suatu bilangan acak antara 1 sampai jumlah pertemuan kuliah yang belum pernah muncul pada kromosom tersebut. Misal, gen ke-4 (bernilai 65) diubah dengan 539 (yang belum pernah muncul). 207

Gambar 10-16 (a) Pola pemotongan stok 2 dimensi seluas 165 cm^2 yang gagal memenuhi order (order ke-2 kurang satu bagian) dan menghasilkan sisa bahan seluas (5 x 1) + (1 x 5) + (4 x 5) = 30 cm^2 atau 30/165 = 18,18% dari stok yang ada. (b) Pola pemotongan yang berhasil memenuhi order dan sisa bahan hanya seluas 1 x 6 cm^2 atau 6/165 = 3,363%. 212

Gambar 10-17 Suatu representasi kromosom untuk masalah pemotongan bahan dimana nomor order dan orientasinya diletakkan ke dalam dua gen yang berdekatan (berpasangan)..... 214

Gambar 10-18 Suatu representasi kromosom untuk masalah pemotongan bahan dimana semua nomor order diletakkan ke dalam gen-gen awal yang berwarna abu-abu sedangkan orientasinya secara berurutan diletakkan pada gen-gen bagian belakang yang berwarna putih..... 215

Gambar 10-19 Contoh *order crossover* dengan batasan bahwa dua titik rekombinasi yang dibangkitkan harus pada posisi genap agar dapat mempertahankan setiap pasangan gen ganjil dan genap yang menyatakan nomor dan orientasi order. Pewarisan gen dilakukan untuk setiap pasangan gen tersebut..... 216

Gambar 10-20 *Order crossover* dengan batasan bahwa dua titik rekombinasi yang dibangkitkan harus pada posisi di setengah bagian kromosom yang pertama (pada kasus di atas, antara 1 sampai 7)..... 217

Gambar 10-21 *Swap mutation* untuk representasi kromosom yang pertama pada masalah pemotongan bahan..... 218

Daftar Tabel

Tabel 3-1 Rangkuman spesifikasi teknis *Simple GA* (SGA)..... 28
Tabel 3-2 Proses seleksi orang tua pada generasi ke-1. 36
Tabel 3-3 Proses rekombinasi pasangan orang tua pada generasi ke-1. 37
Tabel 3-4 Daftar barang untuk dimasukkan ke koper. Barang mana saja yang harus dimasukkan koper berkapasitas 20 kg sehingga diperoleh total nilai yang maksimum? 41
Tabel 3-5 Contoh penggunaan *linear fitness ranking* untuk menskalakan nilai-nilai *fitness* ke dalam f_{LR} menggunakan formula pada persamaan 3.21..... 56
Tabel 3-6 Contoh proses perankingan menggunakan *Linear Ranking* (LR) dan *Non-Linear Ranking* (NLR)..... 66
Tabel 3-7 Populasi pada generasi ke-1 105
Tabel 3-8 Populasi pada generasi ke-10..... 107
Tabel 4-1 Rangkuman spesifikasi teknis ES. 112
Tabel 4-2 Empat metode rekombinasi pada ES. 115
Tabel 5-1 Rangkuman spesifikasi teknis EP. 123
Tabel 6-1 Rangkuman spesifikasi teknis EP. 128
Tabel 6-2 Data histori penerimaan pegawai. 129
Tabel 6-3 Operasi bahasa pemrograman PL dan kode *integer*-nya. 133
Tabel 8-1 *Production rules* dan jumlah pilihan yang tersedia pada masing-masing aturan tersebut. 157
Tabel 8-2 Proses translasi kromosom {40, 83, 4, 62, 237, 107, 247, 179} menjadi ekspresi program atau fungsi $X - \cos(X)$..... 161
Tabel 8-3 Proses translasi kromosom {24, 120, 61, 11, 16, ... 203, 101, 83} menjadi ekspresi program atau fungsi yang .tidak lengkap..... 162
Tabel 8-4 Proses translasi kromosom {24, 120, 61, 11, 16, 203, 101, 83} menjadi ekspresi program atau fungsi yang tidak lengkap..... 163

Tabel 9-1 Himpunan aturan *fuzzy* untuk pengontrolan probabilitas *crossover* (p_c) yang diusulkan oleh Xu & Koleganya [TETO1] 180
Tabel 9-2 Himpunan aturan *fuzzy* untuk pengontrolan probabilitas mutasi (p_m) yang diusulkan oleh Xu & Koleganya [TETO1]. 181
Tabel 9-3 Himpunan aturan *fuzzy* untuk pengontrolan *exploitation-oriented crossover rate* yang diusulkan oleh Herrera & Lozano [TETO1]. 181
Tabel 9-4 Himpunan aturan *fuzzy* untuk pengontrolan *selective pressure* yang diusulkan oleh Herrera & Lozano [TETO1].. 182
Tabel 10-1 Daftar barang untuk dimasukkan ke koper. Barang mana saja yang harus dimasukkan koper berkapasitas 20 kg sehingga diperoleh total nilai yang maksimum? 185
Tabel 10-2 Observasi untuk menemukan ukuran populasi dan probabilitas mutasi yang paling optimal. Dalam observasi ini, jumlah maksimum individu yang dievaluasi adalah 600.000. 189
Tabel 10-3 Daftar 60 kota di tiga benua berbeda dengan posisi *Latitude* dan *Longitude*-nya. 196
Tabel 10-4 Basis Data Pertemuan Kuliah. 202
Tabel 10-5 Batasan-batasan dan nilai prioritas untuk masalah penjadwalan kuliah. 208
Tabel 10-6 *Order list* (daftar permintaan). 211
Tabel 10-7 Suatu contoh *order list* dengan tiga ukuran. 211

1. Pendahuluan

Evolutionary Computation (EC) sudah sejak lama dibahas dalam berbagai seminar dan sudah berhasil diimplementasikan dalam berbagai bidang. Apa yang melatarbelakangi munculnya EC, di mana posisi EC dalam ilmu pengetahuan, seperti apakah EC dulu, saat ini dan masa depan? Bab 1 ini mengupas semua hal tersebut sebagai pengantar dalam memahami EC.

1.1 "Evolusi" dan "Genetika"

Pada judul buku ini, kata "Evolusi" dan "Genetika" sengaja diberi tanda petik untuk menunjukkan bahwa kedua istilah tersebut memiliki pengertian yang berbeda-beda di kalangan para ahli dan masih menjadi perdebatan panjang. Beberapa tahun terakhir ini, kedua teori tersebut memang sedang mendapatkan sorotan dan kritikan tajam dari para ilmuwan. Misalnya, pada tahun 2005, BBC Indonesia menyatakan bahwa negara bagian Kansas di Amerika telah menyetujui standar pelajaran sains untuk sekolah-sekolah umum yang meragukan teori evolusi. Standar baru tersebut telah memasukkan beberapa tantangan khusus antara lain pernyataan bahwa terdapat kelemahan bukti dan penjelasan alamiah mengenai kode genetik dan menganggap bahwa catatan mengenai fosil tidak konsisten dengan teori evolusioner [BBC05].

Bahkan di Inggris sendiri, tanah kelahiran bapak evolusi Charles Darwin, para ilmuwan dan intelektual banyak yang menentang Darwinisme. Di sekolah-sekolah di Inggris, fakta penciptaan maupun teori *Intelligent Design* (Perancangan Cerdas), sebagai pembanding teori evolusi, sudah mulai diajarkan [HID07]. Para ilmuwan pendukung *Intelligent Design* beralasan bahwa teori evolusi tidak mampu menjelaskan secara penuh asal usul dan perkembangan makhluk hidup di bumi. Menurut mereka, kehidupan di bumi, dan alam semesta secara umum, memperlihatkan sedemikian banyak keteraturan, kegunaan, dan perancangan sehingga sudah pasti harus ada "Sang Perancang" di balik kemunculannya. Bagian terpenting dari teori *Intelligent Design* adalah "desain". Jagat raya dan makhluk hidup pastilah didesain dengan cara atau proses apa pun dan tidak mungkin ada dengan sendirinya tanpa disengaja [HID07].

Sebuah studi terbaru yang dilakukan oleh tiga ilmuwan Inggris (Susannah Thorpe, Robin Crompton, dan Roger Holder) menyatakan bahwa "Berjalan Tegak dengan dua kaki" merupakan ciri dasar kelompok monyet besar dan tidak melewati tahapan "Berjalan menggunakan Empat Kaki" terlebih dulu seperti yang dinyatakan oleh teori evolusi [BBC07].

Bagaimanapun, kedua istilah tersebut, "Evolusi" dan "Genetika", akan sering kita gunakan dalam suatu teknik komputasi yang disebut *Evolutionary Computation* (EC). Tetapi, tentu saja dalam bentuk yang sederhana. Misalkan, teori evolusi yang menyebutkan bahwa jerapah yang berleher panjang akan bertahan hidup ketika dedaunan yang tersisa hanya terletak di tempat yang tinggi ataupun gigi pemangsa yang ber-evolusi menjadi lebih tajam ketika kulit mangsanya bertambah keras, masih dikategorikan sederhana dan tidak ada pertentangan. Tetapi, teori evolusi yang memungkinkan suatu spesies berubah menjadi spesies lain (misalkan, kera yang mengalami evolusi menjadi manusia) bukanlah teori evolusi yang sederhana dan masih banyak pertentangan di kalangan ilmuwan dan intelektual. Demikian juga dengan teori genetika. Teori yang menyatakan bahwa perubahan yang terjadi pada gen-gen suatu spesies bisa mengakibatkan spesies tersebut ber-evolusi menjadi spesies baru bukanlah teori genetika yang sederhana dan teori ini dianggap masih banyak kelemahannya. Sebagian ilmuwan masih menganggap bahwa mutasi gen hanya akan mengakibatkan perubahan pada struktur tubuh tertentu pada suatu spesies dan tidak memungkinkan spesies tersebut berubah menjadi spesies lain.

Evolusi di dunia nyata

Bagaimana proses evolusi yang terjadi di dunia nyata? Setiap lingkungan (wilayah) memiliki sumber daya terbatas yang hanya bisa digunakan oleh individu-individu dalam jumlah terbatas. Setiap makhluk hidup memiliki

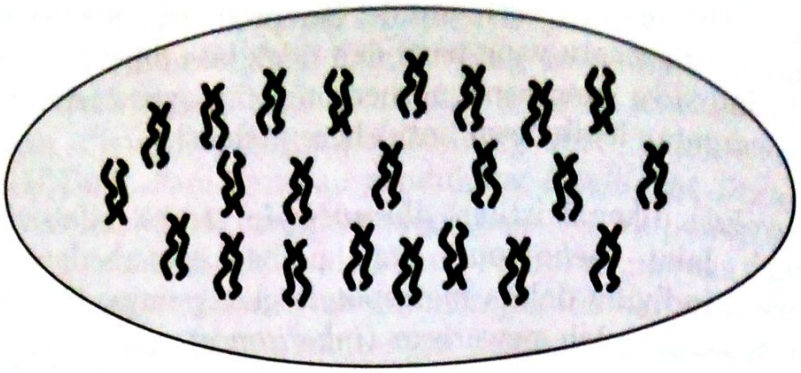
naluri dasar (*basic instinct*) untuk bereproduksi. Sehingga, **seleksi alamiah** menjadi sesuatu yang pasti dan tidak bisa dihindari. Oleh karena itu, individu-individu yang sanggup berkompetisi secara lebih efektif akan memiliki kesempatan lebih besar untuk bereproduksi.

Di dalam evolusi dikenal istilah ***phenotypic traits*** (sifat-sifat fenotip), yaitu tingkah laku (*behaviour*) atau perbedaan-perbedaan fisik yang mempengaruhi individu dalam merespon lingkungannya. *Phenotypic traits* ditentukan sebagian oleh pewarisan (*inheritance*) dan sebagian lagi oleh faktor-faktor selama perkembangan individu. *Phenotypic traits* bersifat unik untuk setiap individu di mana hal ini sebagian dihasilkan dari perubahan-perubahan yang bersifat acak. Jika *phenotypic traits* memiliki kesempatan bereproduksi yang tinggi dan dapat diwariskan, maka *phenotypic traits* tersebut akan cenderung bertambah banyak pada generasi-generasi yang akan datang.

Dari uraian di atas, kita bisa mengatakan bahwa individu-individu yang unik dengan berbagai variasinya adalah **unit-unit seleksi**. Sedangkan suatu populasi yang terdiri dari individu-individu, dalam satu spesies tertentu, adalah **unit evolusi**.

Genetika di alam

Di dalam kehidupan nyata, informasi-informasi yang membangun organisme dikodekan ke dalam molekul-molekul DNA (*Dioxyribo Nucleat Acid*). Kumpulan kode DNA (yang disebut *genotype*) tersebut menentukan gen-gen pada *phenotype*. Sifat-sifat *phenotype* atau *phenotypic traits* adalah suatu pemetaan yang kompleks. Satu gen dapat mempengaruhi banyak *traits* (*pleiotropy*). Sebaliknya, banyak gen bisa mempengaruhi suatu *trait* (*polygeny*). Perubahan kecil pada *genotype* mengarah pada perubahan kecil dalam organisme tersebut, misalnya tinggi badan, bentuk dan warna rambut, dan sebagainya. Sekumpulan gen yang dikodekan ke dalam untaian DNA disebut kromosom. Sebagian besar sel terdiri dari dua kopi kromosom (*diploidy*). Materi genetik yang lengkap dalam suatu individu disebut *genome*. Dalam suatu spesies, sebagian besar materi genetik adalah sama. Sebagai contoh, DNA manusia terorganissasi ke dalam kromosom. Satu sel pada tubuh manusia terdiri dari 23 pasang kromosom yang secara bersama-sama mendefinisikan atribut-atribut fisik seperti terlihat pada gambar berikut ini.

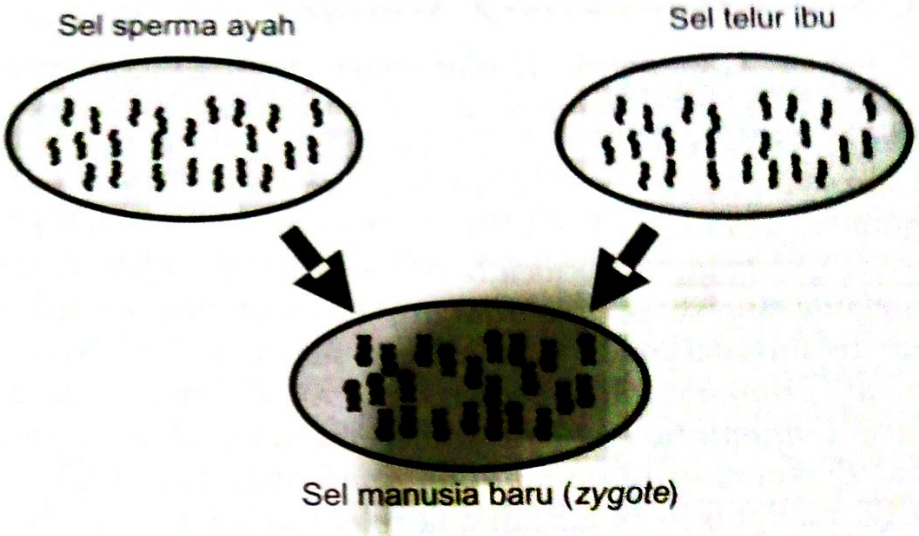


Gambar 1-1 Satu sel tubuh manusia yang berisi 23 pasang kromosom.

Gametes (misalnya sel sperma atau sel telur) berisi 23 kromosom tunggal (bukan 23 pasang). Sel yang hanya berisi satu kopi kromosom tunggal disebut *Haploid*. *Gametes* dibentuk oleh suatu proses pemisahan sel secara khusus yang disebut *meiosis*. Selama *meiosis*, pasangan-pasangan kromosom menjalankan operasi yang disebut *crossover*. Pasangan-pasangan kromosom melakukan *alignment* dan menduplikasi. Pasangan-pasangan kromosom bagian dalam terhubung pada *centromere* dan bagian-bagiannya saling bertukaran (*swap*). Hasilnya adalah satu kopi kromosom *maternal/paternal* dan dua kombinasi yang benar-benar baru. Setelah proses *crossover*, satu dari setiap pasang kromosom membentuk *gamete*. Perhatikan ilustrasi pada gambar berikut.

Gambar 1-2 *Crossover* selama proses *meiosis* pada sepasang kromosom.

Selanjutnya, bagaimana terjadinya proses pembuahan atau *fertilisation* pada sel manusia? Perhatikan gambar di bawah ini.



Gambar 1-3 Proses pembuahan (*fertilisation*) pada sel manusia.

Setelah proses *fertilisation*, *zygote* membagi dirinya dengan sangat cepat sehingga menghasilkan sangat banyak sel yang semuanya memiliki komposisi genetik yang sama. Meskipun semua sel memiliki gen-gen yang sama, tetapi mereka akan memiliki tingkah laku yang berbeda. Proses pembentukan tingah laku yang berbeda ini disebut *ontogenesis*. Semua proses tersebut menggunakan dan dikontrol oleh mekanisme pendekodean gen-gen dalam DNA.

Di dunia nyata dikenal istilah **hereditas**, yaitu sebuah ide yang menyatakan bahwa sifat-sifat individu dapat dikodekan dengan cara tertentu sehingga sifat-sifat tersebut dapat diturunkan kepada generasi berikutnya. Bagaimana informasi yang dapat diturunkan ini disimpan ke dalam suatu individu? Jawabnya adalah bahwa setiap individu dari suatu *species* membawa sebuah ***genome*** yang berisi beberapa **kromosom** dalam bentuk molekul-molekul DNA. Setiap kromosom berisi sejumlah ***gen***, di mana unit-unit hereditas dan pengodean informasi diperlukan untuk membangun dan menjaga suatu individu. Setiap gen dibangun dari suatu deretan ***bases***. Terdapat empat bases dalam kromosom yang dinyatakan sebagai **A**, **C**, **G**, dan **T**. Jadi, informasi disimpan dalam suatu pola *digital* menggunakan keempat simbol tersebut. Selama perkembangan dan juga selama kehidupan suatu individu, DNA dibaca dengan suatu enzim yang dsebut RNA *polymerase*. Proses ini dikenal sebagai *transcription* yang menghasilkan *messenger* RNA (mRNA). Selanjutnya, protein dibentuk dalam suatu proses yang disebut *translation* menggunakan mRNA sebagai sebuah *template*. Masing-masing gen bisa memiliki beberapa *setting*. Sebagai contoh adalah gen yang mengodekan warna mata manusia. Manusia bisa memiliki warna mata berbeda-beda: coklat, hitam, biru, hijau, dan sebagainya.

1.2 Posisi *Evolutionary Computation*

Terdapat istilah lain untuk *Evolutionary Computation* yang banyak digunakan di buku teks maupun makalah ilmiah, yaitu ***Evolutionary Computing***. Di buku ini, kita menggunakan istilah *Evolutionary Computation*. Alasan utamanya adalah karena istilah tersebut sudah menjadi standar dan digunakan oleh *Institute of Electrical and Electronics Engineers* (IEEE) maupun *Association for Computing Machinery* (ACM). IEEE menggunakan istilah *Evolutionary Computation* dalam penerbitan jurnal ilmiah internasional yang diberi nama *IEEE Transaction on Evolutionary Computation*. Sedangkan ACM menggunakan istilah *Evolutionary Computation* dalam konferensi internasional yang disebut *Genetic And Evolutionary Computation Conference* (GECCO). Seperti kita ketahui, IEEE maupun ACM adalah dua referensi yang paling terkenal dan paling banyak digunakan dalam *advanced technologies* di bidang elektro, elektronika maupun ilmu komputer.

Julie Leung, Keith Kern, Jeremy Dawson, mendefinisikan EC sebagai: "*Evolutionary Computation is an abstraction from the theory of biological evolution that is used to create optimization procedures or methodologies, usually implemented on computers, that are used to solve problems*" [JUL07]. Jadi, EC merupakan suatu abstraksi yang memerlukan rekayasa tertentu untuk menghasilkan prosedur, metodologi atau algoritma optimasi. Algortima yang berbasis abstraksi EC ini sering disebut sebagai *Evolutionary Algorithms* (EAs). Pada bab berikutnya kita akan membahas lebih detail mengenai EAs.

Kemunculan EC terinspirasi oleh ilmu-ilmu pengetahuan tentang kehidupan (*life sciences*) dan biologi. Meski demikian, posisi EC berada dalam kelompok ilmu komputer karena EC merupakan teknik komputasi berbasis komputer yang digunakan untuk menyelesaikan berbagai masalah komputasi yang sangat kompleks. Dalam perkembangannya, EC mengadopsi beragam teori dari berbagai bidang ilmu, seperti Fisika, Statistika, Probabilitas, Biogeografi, dan sebagainya. Dalam bukunya, AE Eiben dan JE Smith menyatakan bahwa EC merupakan bagian dari ilmu komputer dan bukan bagian dari *life sciences* maupun ilmu biologi [EIB03].

1.3 Sejarah EC

Pada dasarnya, EC sudah muncul sejak lama ketika Turing mengusulkan teknik pencarian yang disebut *genetical or evolutionary search* pada tahun 1948. Kemudian, pada tahun 1962 Bremermann menyampaikan teknik

optimasi melalui evolusi dan rekombinasi (*optimization through evolution and recombination*).

Tidak lama kemudian, beberapa ilmuwan melakukan penelitian secara terpisah dan menghasilkan beberapa teknik komputasi yang sedikit berbeda. Pada tahun 1964, Rechenberg mengusulkan teknik komputasi yang disebut ***Evolution Strategies* (ES)**. Pada waktu yang hampir bersamaan, yakni 1965, L. Fogel, Owens dan Walsh mengusulkan ***Evolutionary Programming* (EP)**. Sepuluh tahun berikutnya, 1975, John Holland menyampaikan gagasan tentang ***Genetic Algorithms* (GA)**. Pada tahun 1992, Koza mengusulkan idenya tentang ***Genetic Programming* (GP)**. Selanjutnya, para ahli komputasi berusaha mengembangkan berbagai ide untuk memperbaiki performansi keempat teknik tersebut. Pada tahun 1995, Kenneth Price dan Rainer Storn mengusulkan ***Differential Evolution* (DE)** yang sangat mirip dengan ES. Dalam berbagai referensi dinyatakan bahwa DE memiliki performansi jauh lebih baik dibandingkan ES. Pada tahun 1998, Conor Ryan, Michael O'Neill, dan JJ Collins mengusulkan ***Grammatical Evolution* (GE)** yang merupakan pengembangan dari GP (yang fokus pada bahasa LISP). Dengan merepresentasikan individu sebagai *Backus Naur Form* (BNF), GE bisa digunakan untuk meng-"evolusi" program yang bebas bahasa.

Teknik-teknik komputasi atau algoritma tersebut di atas memiliki pijakan yang sama, yakni dua buah teori: **"evolusi"** dan **"genetika"**. Tetapi, kedua teori tersebut mengalami **penyederhanaan** dalam implementasinya ke dalam EC [EIB03]. Pengadopsian kedua teori tersebut dilakukan secara sangat sederhana mulai dari representasi individu, proses evolusi yang terjadi, rekombinasi (*crossover*) dan mutasi, sampai pada proses penggantian populasi. Semuanya direpresentasikan dengan cara yang sangat sederhana ke dalam dunia komputer. Sehingga, bisa dikatakan bahwa sesungguhnya EC tidak berpijak secara kuat pada kedua "teori" tersebut.

Sejak tahun 1985, EC mulai mendapatkan banyak perhatian. Berbagai konferensi dan *workshop* yang membahas EC mulai banyak diselenggarakan. Konferensi internasional EC yang pertama kali adalah *International Conference on Genetic Algorithms* yang diadakan di Pittsburgh Amerika pada bulan July 1985 [JOH85]. selanjutnya, pada bulan Oktober tahun 1990 diadakan *workshop* pertama tentang *Parallel Problem Solving from Nature* (PPSN) di Dortmund, Jerman [HAN90]. Berbagai teori dan aplikasi didiskusikan pada *workshop* tersebut. Pada tahun 1993, jurnal ilmiah EC diterbitkan oleh *The Massachusetts Institute of Technology* (MIT) Press. Pada tahun 1997, sebuah jaringan yang didanai oleh Komisi Eropa (*European Commission*) mendirikan *The European Network of Excellence in Evolutionary Computing* (EvoNet) untuk mengkoordinasi dan

memfasilitasi kerja sama antar-institusi-institusi akademik yang melakukan riset di bidang EC dan juga untuk mentransfer ilmu pengetahuan dan mengaplikasikan EC ke dalam dunia industri. EvoNet dikoordinasi oleh Napier University, UK yang menerbitkan publikasi tiga bulanan (*quarterly newsletter*) yang disebut *EvoNews* [ALA07]. Saat ini, EvoNet telah memiliki anggota lebih dari 150 institusi [EIB03].

1.4 EC Saat Ini

Pada awal abad 21, semakin banyak konferensi yang membahas EC. Terdapat tiga konferensi dan *workshop* internasional berskala besar yaitu: *Genetic and Evolutionary Computation Conference* (GECCO), *Foundations of Genetic Algorithms* (FOGA) dan *Congress on Evolutionary Computation* (CEC). Kedua konferensi GECCO dan CEC mencakup semua aspek tentang EC (teori, hubungannya dengan bidang ilmu lainnya, aplikasi dalam dunia nyata, dan sebagainya), sedangkan FOGA fokus pada aspek teori EC [HEI07]. Masih banyak konferensi dan *workshop* lainnya yang jumlahnya puluhan, tetapi tidak fokus pada EC. Artinya, EC hanya merupakan salah satu topik yang dibahas pada konferensi dan *workshop* tersebut. Pada tahun 2003, diperkirakan 750 sampai 1000 makalah yang dipublikasikan dalam konferensi, *workshop*, maupun jurnal ilmiah.

Hingga saat ini, sangat banyak produk atau aplikasi yang telah berhasil dibangun menggunakan EC. Selain itu, banyak sekali perusahaan konsultan maupun *Research* & *Development* (R&D) *Center* bidang EC yang telah didirikan di seluruh dunia. Aplikasi-aplikasi EC di antaranya adalah:

Aplikasi EC untuk optimasi

- Sistem penjadwalan kuliah di universitas, rumah sakit, *call center*, dan sebagainya. Sistem penjadwalan ini bisa ditujukan untuk meminimalkan jumlah bentrok, melakukan efisiensi waktu, memaksimalkan utilitas ruangan, dan sebagainya.
- Sistem pengepakan hasil produksi. Misalkan di sebuah pabrik tekstil, beberapa gulungan kain dengan beragam diameter dan ketinggian harus dimasukkan ke dalam kardus-kardus yang memiliki beragam ukuran. Sistem pengepakan dibuat untuk meminimalkan jumlah kardus yang dibutuhkan dan di setiap kardus diharapkan tidak terlalu banyak ruang kosong yang mengakibatkan gesekan antargulungan kain.
- Sistem pemotongan bahan. Misalkan suatu produk sepatu dibuat dari beragam bentuk dan ukuran bahan kulit. EC digunakan untuk melakukan pemotongan lembaran kulit besar menjadi potongan-potongan kecil dengan beragam ukuran sehingga meminimalkan jumlah bahan yang terbuang.

- *Optimised satellite designs*, suatu sistem perancangan satelit yang digunakan NASA untuk memaksimlakan *vibration isolation*.

Aplikasi EC untuk pemodelan

- *Loan applicant creditibility*. Misalkan *British bank* yang membangun sistem untuk membuat model kredibilitas yang ber-evolusi untuk memprediksi tingkah laku nasabah-nasabah barunya dalam melakukan pembayaran utang.
- Sistem prediksi penggunaan *bandwidth* jaringan telepon atau *internet*. Dengan menggunakan data-data pada masa lalu atau *historical data*, EC dapat digunakan untuk membangun model yang bisa memprediksi tingkah laku pelanggan menggunakan teknik *learning* (pembelajaran). Tentu saja data-data yang digunakan untuk proses *learning* harus representatif (menggambarkan kondisi berbagai macam kasus penggunaan *bandwidth* jaringan).

Aplikasi EC untuk simulasi

EC bisa digunakan untuk mensimulasikan perdagangan, ekonomi, kompetisi, dan sebagainya untuk melakukan kalibrasi model. Dengan menggunakan model tersebut, kita bisa melakukan optimasi strategi dan pengambilan kebijakan. Salah satu contohnya adalah *Evolving Artificial Societies*.

1.5 EC Masa Depan

Banyak tantangan yang harus dihadapi EC di masa depan. Permasalahan dunia nyata semakin kompleks dan besar. Beberapa di antaranya adalah permasalahan pada bidang keuangan dan ekonomi (*finiance and economics*), robotika, bioinformatika, dan masalah optimasi dengan banyak tujuan (*multi-objective optimization*).

Pada bidang keuangan dan ekonomi, EC diharapkan bisa digunakan untuk menemukan aturan-aturan permalan (*forecasting rules*), membangun strategi-strategi tawar menawar (*bargaining strategies*), pemodelan ekonomi (*economic models*), maupun mencari portofolio yang memiliki pengembalian maksimum (*maximum return*) dan risiko paling kecil (*minimum risk*) [EDW07]. Penggunaan *artificial markets* berbasis EC juga membantu kita memahami konsep-konsep dasar bidang ekonomi, seperti rasionalitas dan hipotesis pasar yang efisien. Di masa depan, ukuran dan kompleksitas data-data keuangan dan ekonomi global bisa dipastikan akan sangat besar. Jika perkembangan teknologi komputer, prosesor dan memori, tidak cukup cepat dalam mengimbangi besar dan kompleksitasnya

permasalahan, maka EC akan menjadi satu-satunya teknik komputasi masa depan untuk bidang ini.

Pada bidang robotika, EC diharapkan bisa berperan banyak pada *RoboCup*, yaitu suatu proyek internasional yang mendorong lahirnya berbagai bidang riset seperti robotika dan *artificial intelligence*. Salah satu yang menarik adalah proyek *RoboCup soccer*, yakni membangun tim robot sepak bola juara. Tujuan akhir dari *RoboCup soccer* adalah mengalahkan tim (manusia) juara piala dunia sepak bola pada tahun 2050. Lingkungan *RoboCup soccer* memberikan banyak masalah yang sangat menantang, seperti pemrosesan informasi secara waktu nyata (*real-time information processing*), penanganan data berderau (*noisy data handling*), kerjasama antarrobot dalam bermain bola, dan strategi untuk memenangkan permainan. EC diharapkan bisa digunakan untuk membangun sistem strategi tim yang berevolusi (*evolving soccer team strategies*). Suatu strategi tim dikodekan sebagai untaian bilangan bulat (*integer string*) yang merepresentasikan sekumpulan aturan aksi dari sepuluh pemain [TOM07]. Di sini EC digunakan untuk mencari sekumpulan aturan aksi yang optimal. Kalau kita bayangkan permainan sepak bola, jumlah strategi yang mungkin digunakan oleh masing-masing tim tentu saja sangat banyak atau bahkan tak terhingga. Selain itu, suatu tim bisa mengubah strategi setiap saat sehingga masalah ini bersifat *real-time*. Masalah dengan karakteristik tersebut tentu saja sangat sesuai untuk EC.

2. Memahami *Evolutionary Algorithms*

Untuk memahami sesuatu secara detail, seringkali kita bertanya menggunakan lima kata tanya: apa, mengapa, kapan, di mana, dan bagaimana. Bab 2 ini membahas bagaimana memahami *Evolutionary Algorithms* (EAs) melalui lima pertanyaan tersebut.

2.1 Apa itu EAs?

Bagi yang baru mengenal istilah *Evolutionary Algorithms* (EAs), mungkin akan membayangkan bahwa EAs adalah semacam teknik komputasi atau algoritma yang bisa ber-evolusi, dengan melahirkan anak-anak yang berupa algoritma-algoritma baru yang lebih sesuai dengan lingkungan yang dihadapi. Imajinasi ini pada awalnya mungkin terkesan berlebihan. Tetapi, hal ini bisa benar dan bahkan sudah bisa diimplementasikan. Sebagian algoritma yang termasuk ke dalam EAs memang ada yang berfokus pada pencarian solusi secara langsung, tetapi ada suatu algoritma EAs yang fokusnya adalah bagaimana menemukan suatu program yang tepat untuk menyelesaikan masalah. Program ini bisa ber-evolusi dan adaptif terhadap perubahan permasalahan.