



PENGEMBANGAN ALGORITMA GENE
EXPRESSION PROGRAMMING UNTUK
MASALAH OPTIMASI DINAMIS

SEMINAR BIDANG KAJIAN

SYALIS IBNIH MELATI ISTINI
99219033

PROGRAM DOKTOR TEKNOLOGI INFORMASI
UNIVERSITAS GUNADARMA
September, 202

Daftar Isi

| | |
|--|-----------|
| Daftar Isi | i |
| 1 Pendahuluan | 3 |
| 1.1 Latar Belakang | 3 |
| 1.2 Batasan Masalah | 4 |
| 1.3 Tujuan | 5 |
| 1.3 Kontribusi | 5 |
| 2 Tinjauan Pustaka | 6 |
| 2.1 Algoritma Genetik. | 6 |
| 2.2 Pemrograman Genetik. | 6 |
| 2.3 Gene Expression Programming. | 10 |
| 2.5 Ruang Kontinu dan Kombinatorial. | 11 |
| 2.6 Perbandingan Tinjauan..... | 12 |
| 3 Metodologi | 15 |
| 3.1 Pendekatan Penelitian | 15 |
| 3.2 Perancangan Awal Spesiasi GEP. | 17 |
| 3.3 Rencana Kerja | 18 |
| Daftar Pustaka | 19 |

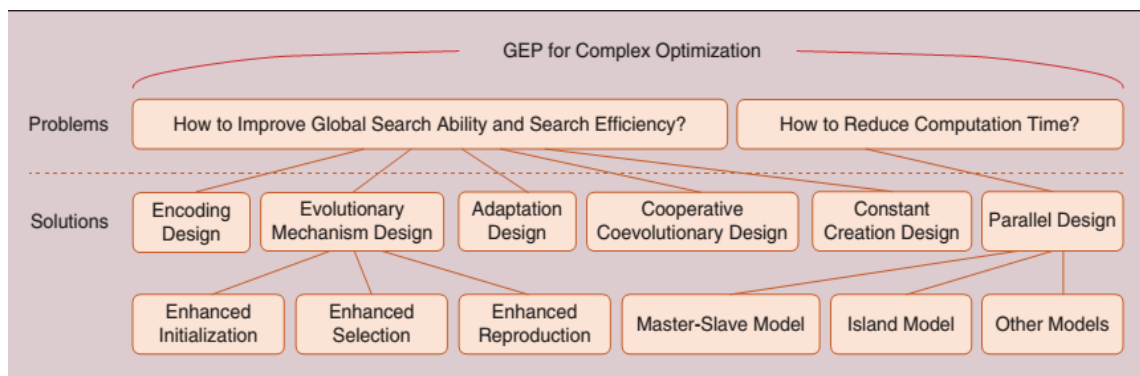
Bab 1

Pendahuluan

1.1 Latar Belakang

Manusia dan data kaitanya sangat erat tidak hanya pada kehidupan sehari-hari, akan tetapi dalam ruang lingkup industri yang beragam. Dalam praktiknya, industri yang sudah berkembang bahkan mapan, sebagian besar telah digantikan peran oleh *Machine Learning* atau *Artificial Intelligence*, Kecerdasan buatan umumnya dikaitkan dengan sebuah alat bantu untuk menemukan sebuah permasalahan dan menyelesaikan permasalahan yang kompleks pada berbagai permasalahan di bidang bisnis, korporasi, dan juga pemerintahan. Konsep utama dari kecerdasan buatan adalah menciptakan sebuah alat bantu atau mesin yang dapat berpikir seperti manusia (Goralski & Tan, 2020).

Untuk memenuhi kebutuhan industri sebuah *Artificial Intelligence* atau kecerdasan buatan harus memiliki kemampuan menganalisis, melakukan optimasi dan mengambil keputusan dengan kondisi yang berubah-ubah atau kompleks. Menurut Jinghui zhong, et al (2017) algoritma yang dapat menangani optimasi kompleks adalah algoritma *Gene Expression Programming*, algoritma ini merupakan pengembangan dari algoritma genetika dan penerapan *Evolutioner Algoritma* (EA) yang melakukan optimasi dengan menggunakan teori evolusi melalui program komputer.



Gambar 1.1 Kemampuan Algoritma GEP dalam Optimasi Kompleks

[Jinghui zhong, et al 2017]

Permasalahan optimisasi kompleks tentunya akan semakin rumit, terlebih dalam hal data yang dinamis. Penelitian mengenai DOP mempunyai perkembangan pesat khususnya dalam menggunakan algoritma evolusi, atau yang sering disebut *Evolutionary Dynamic Optimization* (EDO), Penelitian ini termasuk relatif muda karena sebagian besar penelitian telah dilakukan dalam 20 tahun terakhir, karena usianya yang relatif muda, bidang ini masih memiliki banyak area terbuka dengan pertanyaan penelitian terbuka, yang mungkin salah satu pertanyaan paling penting adalah tentang seberapa baik penelitian EDO dalam mencerminkan karakteristik umum DOP dan ragam nya jenis DOP yang belum dicakup oleh penelitian akademis saat ini.

Penelitian terbaru dalam menggunakan EDO untuk Masalah Optimalisasi Dinamis (DOP), yang secara umum dapat dibagi menjadi dua kategori: (1) menemukan / melacak optimasi dari waktu ke waktu (DOP hanya dalam ruang kontinu) [S.Jiang dan S,yang,2017], dan (2) mengadaptasi solusi saat ini terhadap perubahan (DOP hanya dalam ruang kombinatorial) [S,yang,2017]. Dengan mengacu pada penelitian sebelumnya, pada penelitian ini bertujuan mengembangkan dan merancang algoritma GEP dalam menyelesaikan permasalahan DOP, baik DOP kontinu dan kombinasi, hal ini tentunya membutuhkan sebuah konsep baru untuk algoritma GEP dalam menangani masalah DOP dalam multiruang (ruang kontinu dan ruang kombinatorial), oleh karena itu penelitian ini membuat kajian pemikiran mengenai *Spesiasi Design-GEP*

1.2 Batasan Masalah

Penelitian ini berfokus pada perancangan spesiasi design-GEP yaitu pengembangan metode algoritma Gene expression Programming dalam mengatasi optimasi kompleks (dalam ruang kontinu dan kombinatorial). Batasan masalah pada penelitian ini adalah :

1. Bagaimana merancang spesiasi design-GEP untuk mengatasi masalah optimasi kompleks (DOP) dalam ruang kontinu dan kombinatorial
2. Bagaimana menguji keunggulan spesiasi design-GEP dengan algoritma sebelumnya
3. Bagaimana mengimplementasi spesiasi design-GEP pada ruang kontinu dan ruang kombinatorial.

1.3 Tujuan

Tujuan umum penelitian ini adalah mengembangkan metode algoritma Gene Expression Programming dalam mengatasi optimasi kompleks yang dapat diimplementasikan pada ruang kontinu dan ruang kombinatorial.

Secara khusus penelitian ini bertujuan untuk.

1. Merancang *spesiasi design-GEP* untuk mengatasi masalah optimasi kompleks (DOP) dalam ruang kontinu dan kombinatorial
2. Menguji keunggulan *spesiasi design-GEP* dengan algoritma sebelumnya
3. Mengimplementasi *spesiasi design-GEP* pada ruang kontinu dan ruang kombinatorial

1.4 Kontribusi

Sesuai dengan latar belakang dan tujuan penelitian, keutamaan dalam pelaksanaan penelitian ini adalah dapat menghasilkan rancangan algoritma terbaru untuk menghasilkan optimasi pada ruang kontinu dan ruang kombinatorial.

Perancangan ini diharapkan menjadi alternative salah satu algoritma dari kecerdasan buatan atau *artificial intelegence* yang dapat digunakan industri khususnya dalam hal optimasi waktu dan biaya.

Bab 2

Tinjauan Pustaka

2.1 Algoritma Genetik

Algoritma genetik adalah model algoritmik yang dikembangkan untuk model simulasi sistem evolusi genetik, dengan sifat individu dinyatakan dengan menggunakan *genotype* (komposisi genetik individu yang diwarisi oleh induknya). Operator algoritma genetik terdiri dari seleksi dan rekombinasi. Seleksi digunakan untuk memodelkan ketahanan hidup dari individu paling *fit*, sedangkan rekombinasi digunakan untuk memodelkan reproduksi.

Algoritma genetik mempunyai implementasi sebagai berikut :

- *Chromosome* disajikan secara biner
- Seleksi proporsional dengan tujuan untuk memilih induk untuk rekombinasi
- *Crossover* satu titik (memilih satu titik secara acak) untuk menghasilkan *offspring*.
- Mutasi seragam

2.2 Pemrograman Genetik

Pemrograman genetik (*Genetic Programming*, GP) merupakan salah satu kelas dari algoritma evolusioner berbasis pada algoritma genetik (*Genetic Algorithm*, GA) yang berkonsentrasi pada *evolusi genotype*. GP merupakan metode potensial untuk menulis program komputer secara otomatis. Perbedaan GP dan GA terletak pada skema penyajian *chromosom* yang digunakan, GA menggunakan penyajian *string* (vector), sedangkan GP menggunakan penyajian *tree* (pohon). Pada awalnya GP dikembangkan oleh Koza untuk mengevolusikan program komputer. Pada setiap generasi, setiap program yang berevolusi (individu) dieksekusi untuk mengukur kinerjanya di dalam domain permasalahan. Hasil yang diperoleh dari program komputer yang berevolusi kemudian digunakan untuk kuantifikasi *fitness* tersebut.

2.3 Gene Expression Programming

Gene Expression Programming (GEP) ditemukan dan dikembangkan oleh Ferreira (1999) yang merupakan perkembangan dari GA dan GP. GEP menggunakan

mekanisme dari GP, tetapi entitas yang dihasilkan GEP (*expression trees*) merupakan ekspresi dari *chromosome* tersebut. Pokok utama dari GEP adalah *chromosome* dan *the expression trees* (ETs), informasi terakhir yang terdapat pada ekspresi tersebut dikodekan dalam bentuk *chromosome*. Proses informasi yang dikodekan disebut *translation*, dan *translation* tersebut menunjukkan suatu kode atau suatu set aturan dari suatu permasalahan yang ingin dicari solusi optimal. Kode tersebut berupa suatu kode genetik yang dihubungkan satu ke satu (one to one) antara *chromosome* dan terdapat fungsi-fungsi dan terminal yang dijadikan parameter permasalahan.

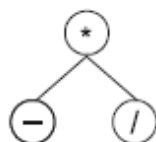
Didalam struktur GEP *genes* untuk membuat suatu *genes* atau *reading frames* (ORFs) lebih mudah dipahami, karena dalam biologi ORF merupakan terdapat urutan-urutan kode untuk membuat suatu *genes* dengan dimulai dengan awal mulai posisi *gene*, lalu dibentuknya informasi genetik. Pada GEP struktur ini dapat disebut K-expression (Karva Expression) yaitu jenis bahasa pemrograman untuk GEP yang dibuat oleh Ferreira (2001). Misalkan terdapat suatu K-expression seperti berikut :

$$\begin{array}{l} 0123456789 \\ *- / Qb + b + aaab \end{array} \quad (2.1)$$

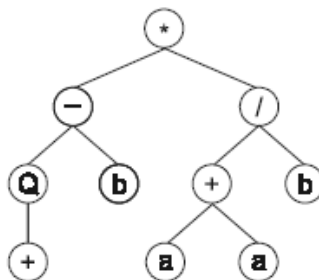
Ekspresi diatas pada *Expression Tree* (ETs) dibaca dari kiri ke kanan, lalu dimulai dari atas ke bawah, dimulai dari *root position* pada posisi 0 :



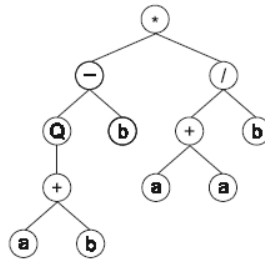
Tanda fungsi “perkalian” di atas mempunyai dua argument (*node*) pada posisi 1 dan 2:



Tanda fungsi “pengurangan” dan “pembagian” masing-masing mempunyai dua argument lagi, pada symbol posisi 3,4,5,6 :



Pada struktur baru diatas, terdapat 3 *nodes*, terdiri atas satu fungsi (+) dan dua variable “a”, tersisa dua elemen pada ORF yaitu pada posisi 10 dan 11 :

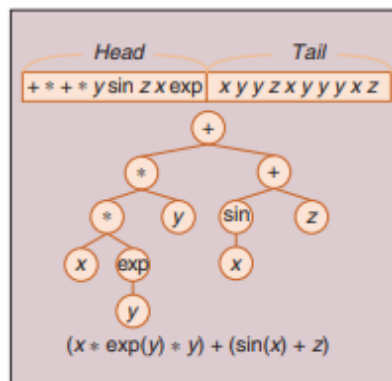


Dengan struktur baru diatas telah terbentuk ETs pada ORF yang telah didefinisi pada (2.1). GEP *genes* tersusun atas *head* dan *tail*. *Head* menandakan simbol yang mempresentasikan baik fungsi ataupun terminal, sedangkan *tail* hanya menandakan simbol yang mempresentasikan terminal. Untuk menyelesaikan suatu masalah, panjang dari *head* (h) telah ditentukan sejak awal, dimana panjang dari *tail* (t) ditentukan dari rumus berikut, dengan (n) sebagai argumen :

$$t=h(n-1)+1 \quad (2.2)$$

misalkan pada *gene* terdapat aturan fungsi $F = \{Q, *, /, -, +\}$ dan aturan terminal $T = \{a, b\}$. Dalam kasus ini, jumlah argument $n=2$, dan jika kita misalkan suatu nilai *head* = 15 dan *tail* = 16, maka panjang dari *gene* tersebut adalah $15+16=31$.

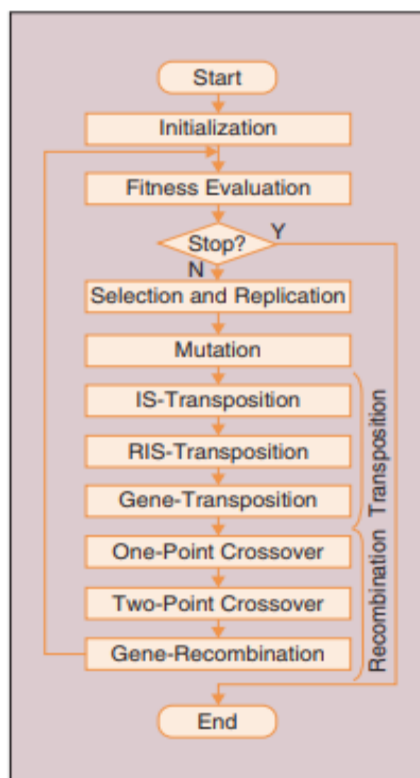
Gambar. 2.1 mengilustrasikan prosedur GEP tradisional, yang terdiri dari operasi berikut. Inisialisasi: Langkah pertama adalah menghasilkan satu set kromosom acak untuk membentuk populasi awal. Untuk setiap kromosom pada populasi awal, setiap elemen string dengan panjang tetap secara acak ditugaskan berdasarkan jenis elemen. Unsur yang termasuk dalam bagian *head* adalah fungsi dan terminal yang ditugaskan, sementara elemen milik bagian *tail* adalah terminal yang ditugaskan.



Gambar 2.1 Model Tradisional GEP (Jinghui zhong, et all 2017)

Operasi transposisi bertujuan untuk mengganti beberapa elemen kromosom dengan sebuah segmen elemen berurutan dalam kromosom yang sama. Ini berisi tiga sub-langkah, yang dilakukan dengan probabilitas masing-masing.

- IS-transposisi: Urutan Penyisipan
(IS) adalah segmen elemen yang berurutan dalam kromosom. Dalam subtep ini, IS dipilih secara acak. Lalu membuat disisipkan di a posisi acak di bagian kepala (*head*) gen.
- RIS-transposisi: Penyisipan Root
Sequence (RIS) adalah segmen elemen berurutan yang dimulai dengan fungsi fungsi dari kepala gen. Dalam subtep ini, kromosom dan gen menjadi dimodifikasi, posisi awal RIS dan panjangnya, dipilih secara acak. Setelah RIS dipilih, Salinan RIS dibuat dan dimasukkan ke dalam akar gen yang dipilih, seperti yang dilakukan pada langkah IS-transposisi
- Transposisi gen (*gene transposition*)
Dalam transposisi gen, kromosom harus dimodifikasi dipilih secara acak. Lalu, acak gen (tetapi bukan gen pertama) dari kromosom yang dipilih dipilih dan dialihkan ke awal
- Rekombinasi:
Tujuan dari operasi rekombinasi adalah untuk bertukar informasi gen dari dua kromosom induk untuk menghasilkan dua keturunan. Ini Operasi juga mengandung tiga sub-langkah. Ini adalah satu titik rekombinasi, rekombinasi dua poin, dan gen rekombinasi.

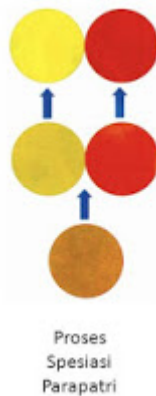


Gambar 2.2 Tahapan Proses Tradisional GEP (Jinghui zhong, et all 2017)

2.4 Spesiasi

Spesiasi adalah suatu proses pembentukan jenis baru. Spesiasi terjadi bila aliran gen antara populasi yang pada mulanya ada secara efektif telah mereda dan disebabkan oleh mekanisme isolasi (Hale et al., 1995). Jenis baru dapat terbentuk dalam kurun waktu sejarah yang panjang maupun pendek tergantung model spesiasi mana yang dilaluinya. Spesiasi merupakan respon makhluk hidup terhadap kondisi lingkungannya berupa adaptasi sehingga kelompok ini dapat bertahan hidup dan tidak punah. Spesiasi merupakan sebuah proses evolusi munculnya spesies baru. Terdapat empat jenis spesiasi alami, tergantung pada sejauh mana populasi yang berspesiasi terisolasi secara geografis dari satu populasi ke yang lainnya. Empat jenis spesiasi alami tersebut adalah: spesiasi alopatrik, spesiasi peripatrik, spesiasi parapatrik, dan spesiasi simpatrik. Spesiasi juga dapat dilakukan secara buatan, melalui domestikasi ataupun eksperimen laboratorium.

Proses pembentukan spesies atau spesiasi dapat ditinjau dari adanya isolasi geografi (penghalang/barier/sawar pada aspek geografis) dan isolasi reproduksi (penghalang pada segi reproduktif). Salah satunya adalah Spesiasi parapatri yang merupakan proses spesiasi yang terjadi di daerah yang bersebelahan dengan daerah dari suatu spesies yang paling dekat hubungan kekerabatannya. Spesiasi parapatrik adalah proses evolusi populasi yang secara geografis bersebelahan menjadi spesies yang baru. Dalam ilmu biogeografi, parapatri adalah hubungan antar organisme yang persebarannya tidak sama tetapi bersebelahan; mereka kadang-kadang bertemu di suatu zona yang sempit. Parapatri merupakan persebaran geografis yang berbeda dengan simpatri (wilayah yang sama) dan alopatri (wilayah yang sepenuhnya berbeda). Adanya perbedaan menyebabkan spesies tersebut sebenarnya terbagi dalam populasi yang berbeda-beda dengan persyaratan yang berbeda-beda. Dengan adanya seleksi alam tiap populasi akan memiliki komposisi genetik yang berbeda-beda, tetapi mungkin kurang diperhatikan dari segi morfologi. Dengan berjalannya waktu, terbentuklah suatu populasi yang tetap bersebelahan, tetapi kemampuan interfertilitasnya secara gradual menurun, berbanding lurus dengan jarak antara dua populasi. Oleh karena itu, pada suatu keadaan akan ada dua populasi yang sudah tidak mampu berinteraksi interfertil, dan harus dianggap sebagai spesies tersendiri.



Gambar 2.3 Proses Spesiasi Parapatric

2.5 Ruang Kontinu dan Kombinatorial

1. Ruang Kombinatorial

Teori Kombinatorial merupakan salah satu pokok bahasan Matematika Diskrit yang telah banyak dikembangkan dan diaplikasikan dalam berbagai bidang. Kombinatorial juga merupakan cabang matematika untuk menghitung jumlah penyusunan objek-objek tanpa harus mengenumerasi semua kemungkinan susunannya. Dalam perkembangan Matematika, dapat dilihat bahwa kajian kombinatorial sangat menarik bagi sebagian orang. Salah satu contoh permasalahan yang dapat diselesaikan dengan kombinatorial adalah menghitung banyaknya kombinasi karakter password.

Cara paling sederhana untuk menyelesaikan persoalan sejenis adalah dengan mengenumerasi semua kemungkinan jawabannya. Mengenumerasi berarti mencacah atau menghitung satu per satu setiap kemungkinan jawaban. Akan tetapi enumerasi masih mungkin dilakukan jika jumlah objek sedikit, sedangkan untuk persoalan di atas, cara enumerasi jelas tidak efisien.

$$P(n, r) = n(n-1)(n-2) \dots (n-(r-1))$$

$$= \frac{n!}{(n-r)!}$$

Gambar 2.4 Contoh Perhitungan Ruang Kombiantorial

2. Ruang Kontinu

Distribusi peluang untuk variabel acak kontinu tidak dapat disajikan dalam bentuk tabel, tetapi dinyatakan dalam sebuah fungsi yang disebut fungsi densitas. Fungsi tersebut dinyatakan sedemikian sehingga luas daerah di bawah kurva, di atas sumbu x » 1. Distribusi peluang kontinu adalah peubah acak yang dapat memperoleh semua nilai pada skala kontinu.

Ruang sampel kontinu adalah bila ruang sampel mengandung titik sampel yang tak terhingga. Salah satu teori ruang kontin yang sering digunakan adalah Distribusi Normal (Gaussian). Distribusi Normal (Gaussian) mungkin merupakan distribusi probabilitas yang paling penting baik dalam teori maupun aplikasi statistik. Distribusi ini paling banyak digunakan sebagai model bagi data riil di berbagai bidang yang meliputi antara lain karakteristik fisik makhluk hidup (berat, tinggi badan manusia, hewan, dll). Terdapat empat alasan mengapa distribusi normal menjadi distribusi yang paling penting:

- a. Distribusi normal terjadi secara alamiah.
- b. Beberapa variabel acak yang tidak terdistribusi secara normal dapat dengan mudah ditransformasi menjadi suatu distribusi variabel acak yang normal.
- c. Banyak hasil dan teknik analisis yang berguna dalam pekerjaan statistik hanya bisa berfungsi dengan benar jika model distribusinya merupakan distribusi normal.
- d. Ada beberapa variabel acak yang tidak menunjukkan distribusi normal pada populasinya, namun distribusi dari rata-rata sampel yang diambil secara random dari populasi tersebut ternyata menunjukkan distribusi normal. banyaknya

$$f_N(x; \mu_x, \sigma_x) = P(X \leq x) = \int_{-\infty}^x f_N(t; \mu_x, \sigma_x) dt = \int_{-\infty}^x \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\frac{(t-\mu_x)^2}{2\sigma_x^2}} dt$$

Gambar 2.5 Contoh Ruang Kontinu

2.6 Perbandingan Tinjauan

Berbagai metode sudah digunakan para peneliti sebelumnya dalam hal optimasi dengan menggunakan teori evolusi dan turunannya. Berikut beberapa jurnal sebagai acuan dalam penelitian ini

1. S. Yang, “*Evolutionary computation for dynamic optimization problems*,” in Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ser. GECCO Companion '15. New York,

NY, USA: ACM, 2015, pp. 629–649. Jurnal publikasi ini fokus mengenai kompilasi tentang perkembangan mutakhir dari komputasi evolusioner untuk masalah optimisasi dinamis. Motivasi untuk buku ini muncul dari kenyataan bahwa banyak masalah optimisasi dunia nyata dan sistem rekayasa tunduk pada lingkungan yang dinamis, di mana perubahan terjadi seiring waktu

2. S. Jiang and S. Yang, “A steadystate and generational evolutionary algorithm for dynamic multiobjective optimization,” *IEEE Trans. Evolut. Comput.*, vol. 21, no. 1, pp. 65–82, Feb. 2017. Jurnal publikasi ini fokus mengenai kajian algoritma evolusioner steadystate dan generasi, yang menggabungkan kemampuan pelacakan yang cepat dari algoritma steady-state dan pelestarian keanekaragaman yang baik dari hasil algoritma berupa generasi, untuk menangani optimisasi multi dinamis.
3. Herimanto., Zarlis, Muhammad,. Efendi, Syahril. “*Improving Performance of The Genetic Algorithm on NP-Complete Problem*”. *Jurnal Nasional Informatika Dan Teknologi Jaringan- Vol. 6 No. 1 (2021)*. Jurnal ini mengangkat Masalah Lengkap Polinomial Non-Deterministik dalam strategi algoritma untuk mengisi teka-teki sudoku yang kosong, rumus tertentu tidak berlaku, tetapi mengisi sudoku adalah masalah keputusan. Sehingga dibutuhkan algoritma dan strategi khusus untuk menyelesaikannya. Salah satu metode yang digunakan adalah algoritma genetika. Namun karena banyaknya proses dan data yang digunakan dalam implementasi algoritma genetika, maka hasil yang diperoleh seringkali tidak optimal. Jurnal ini akan memperkenalkan strategi khusus dalam mengimplementasikan algoritma genetika pada permasalahan NP-Complete, yaitu dengan mengoptimalkan algoritma genetika dalam proses pembentukan populasi.
4. Andoyo, Fachrizal., Arifudin, Riza. “*Optimization of Classification Accuracy Using K-Means and Genetic Algorithm by Integrating C4.5 Algorithm for Diagnosis Breast Cancer Disease*”. *Journal of Advances in Information Systems and Technology* 3(1) April 2021, 1-8. Jurnal ini mengambil tema data mining. menggunakan 32 dataset dengan 569 sampel Wisconsin Diagnostic Breast Cancer yang diambil dari UCI Machine Learning Repository. Data ini memiliki tipe data yang kontinu dan berdimensi tinggi. Penelitian ini bertujuan untuk meningkatkan akurasi C4.5 dengan kombinasi K-Means dan Algoritma Genetika. Hasilnya

Akurasi C4.5 adalah 91,228%. Sedangkan akurasi C4.5 setelah dioptimasi menggunakan K-Means dan Genetic Algorithm adalah 94,824%, Dengan demikian, penerapan K-Means dan Algoritma Genetika pada Algoritma C4.5 dapat meningkatkan akurasi diagnosis kanker payudara sebesar 3,596%.

5. Trung Thanh Nguyen, “*Continuous dynamic optimisation using evolutionary algorithms*”.2011. Pada tesis ini membahas Evolutionary dynamic optimization (EDO), atau studi penerapan algoritma evolusioner untuk masalah optimisasi dinamis (DOP) .Tesis ini membantu mengkarakterisasi DOP dengan lebih baik dengan menyediakan kerangka kerja definisi baru, dua set masalah benchmark baru (untuk kelas DOP kontinu tertentu) dan beberapa baru set ukuran kinerja (untuk kelas DOP kontinu tertentu).
6. Wang, Huanhuan., Liu, Shuming,. et-all. “*Gene Expression Programming Algorithms for Optimization of Water Distribution Network*”. Procedia Engineering 37 (2012) 359 – 264. Jurnal ini mencari solusi untuk sistem distribusi air (WDS) agar mendapat desain biaya yang optimal, menggunakan metode evolusi seperti algoritma genetika (GA), ant colony optimization algorithm (ACO) dan simulated annealing (SA), maka melalui algoritma ekspresi gen (GEP) diadopsi untuk menyelesaikan masalah tersebut. Hasilnya algoritma tersebut menjadi alternative lanjutan untuk desain WDS dengan mempertimbangkan efisiensi komputasi

Bab 3

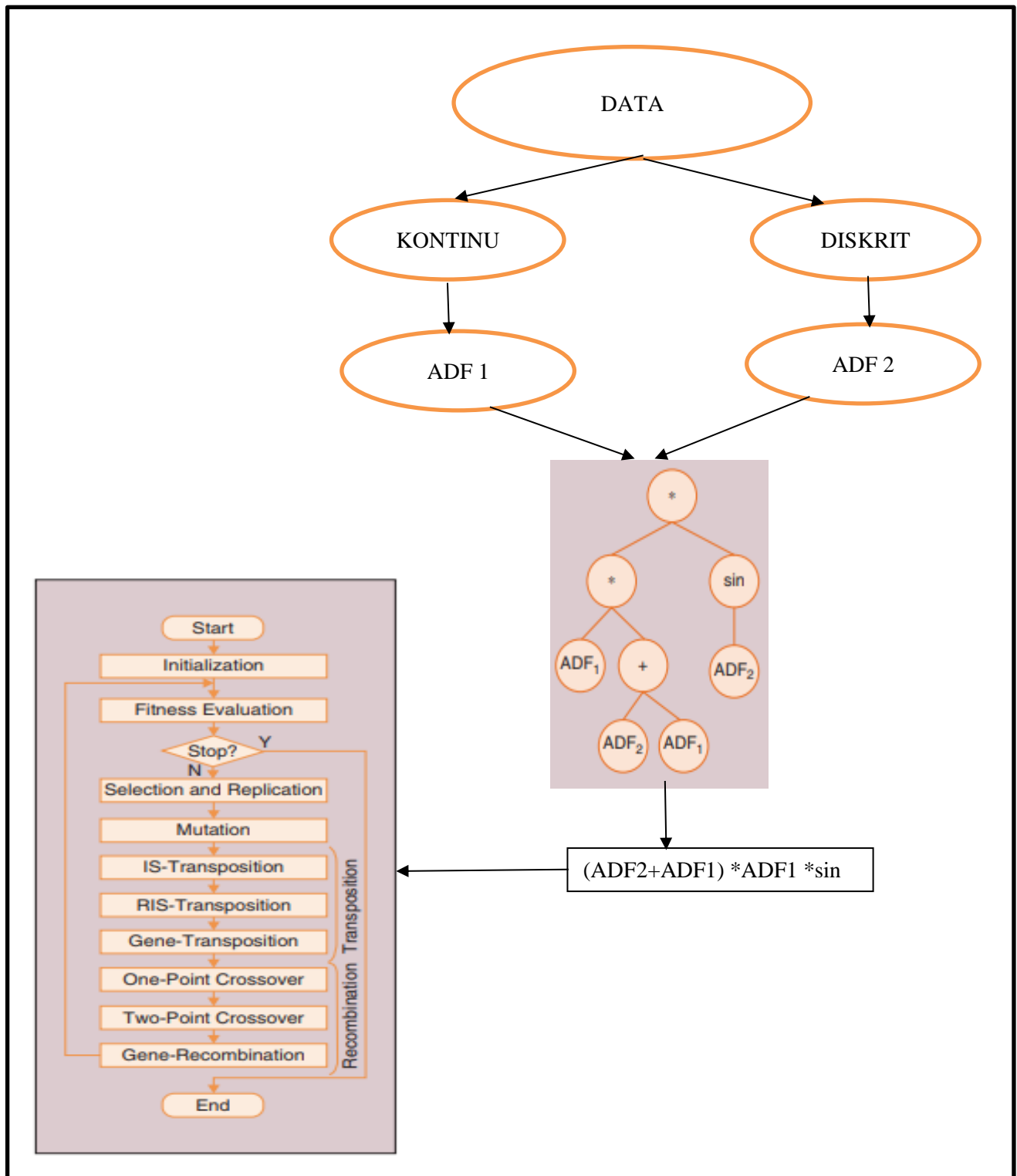
Metodologi

Metodelogi yang digunakan adalah sebagai berikut:

1. Studi literature. Pada tahap ini dilakukan studi terhadap beberapa artikel dan buku yang menguraikan mengenai optimasi kompleks pada data yang dinamis, sehingga memunculkan satu konsep yang memungkinkan dapat mengembangkan teori dari gene expression Programming, yaitu desain spesiasi (Spesiasi Design-GEP)
2. Merancang algoritma Spesiasi Design-GEP untuk mengatasi masalah optimasi kompleks (DOP) dalam ruang kontinu dan kombinatorial
3. menguji keunggulan spesiasi design-GEP dengan algoritma sebelumnya
4. Bagaimana mengimplementasi spesiasi design-GEP pada ruang kontinu dan ruang kombinatorial

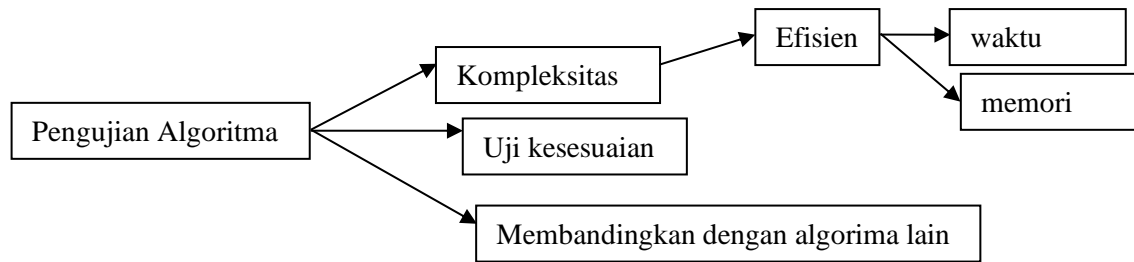
3.1 Pendekatan Penelitian

Pendekatan penelitian spesiasi dilakukan dengan melakukan perubahan model spesiasi biologi terhadap model komputasi komputer, dengan menggabungkan teori algoritma GEP dan ADF (*Automatically Defined Function*) (Koza). Berawal dari data berupa masukkan berisi data kombinatorial atau kontinu, lalu algoritma membandingkan dan mendefinisikan data tersebut ke dalam dua bagian ADF 1 (untuk kombinatorial) dan ADF 2 (untuk kontinu), serta bagian program utama. Setiap ADF diberi ekspresi *tail* dan *head* sehingga memiliki ciri tertentu. Ciri tersebut dilakukan K-expression tree, yang hasilnya dapat dijadikan masukkan algoritma GEP untuk mendapatkan nilai *fitness evaluation* sebagai nilai optimasi, sehingga diharapkan dapat mengatasi nilai data yang dinamis atau permasalahan DOP *dynamic optimization Problem*.



Gambar 3.1 Pendekatan Penelitian

Untuk mengetahui algoritma memiliki fungsi yang baik maka dilakukan pengujian



Gambar 3.2 Pengujian Algoritma

3.2 Perancangan Awal Spesiasi GEP

Perancangan awal pada penelitian ini adalah dengan mengimplementasi kedalam studi kasus. Dalam penelitian ini perancangan awal implementasi spesiasi GEP dalam optimasi pencarian jalur tercepat, didalamnya dimasukan beberapa parameter seperti Lokasi, Jarak, Durasi Waktu, Jenis Kendaraan, Masalah (Semua kendaraan digunakan, Kendaraan harus kembali ke depot, Jam mulai bekerja, Batas waktu perjalanan, Batas jarak minimum), Tampilan Visual, Solusi. Program utama terdapat 6 solusi yang menentukan fungsi atau tidaknya spesiasi GEP dalam menghasilkan optimasi.

| 1 | Lokasi Awal | Lokasi Tujuan | Jarak Tempuh | Durasi Waktu | Metoda: | Bing Maps |
|----|-------------|---------------|--------------|--------------|---------|-----------|
| 2 | ciawi | ciawi | 0.00 | 0:00 | | |
| 3 | ciawi | depok | 40.48 | 1:07 | | |
| 4 | ciawi | bogor | 10.73 | 0:21 | | |
| 5 | ciawi | jakarta | 65.72 | 1:55 | | |
| 6 | ciawi | tangerang | 68.33 | 1:58 | | |
| 7 | ciawi | bekasi | 70.52 | 1:53 | | |
| 8 | ciawi | cikarang | 55.09 | 1:02 | | |
| 9 | depok | ciawi | 40.24 | 1:08 | | |
| 10 | depok | depok | 0.00 | 0:00 | | |
| 11 | depok | bogor | 26.63 | 0:53 | | |
| 12 | depok | jakarta | 28.60 | 0:54 | | |
| 13 | depok | tangerang | 46.94 | 1:16 | | |
| 14 | depok | bekasi | 37.42 | 1:10 | | |
| 15 | depok | cikarang | 91.70 | 2:02 | | |
| 16 | bogor | ciawi | 10.56 | 0:19 | | |
| 17 | bogor | depok | 26.84 | 0:52 | | |
| 18 | bogor | bogor | 0.00 | 0:00 | | |
| 19 | bogor | jakarta | 57.40 | 1:41 | | |
| 20 | bogor | tangerang | 64.68 | 1:40 | | |
| 21 | bogor | bekasi | 62.20 | 1:40 | | |
| 22 | bogor | cikarang | 64.39 | 1:17 | | |
| 23 | jakarta | ciawi | 65.93 | 1:57 | | |

Gambar 3.3 Simulasi Penentuan Lokasi, Jarak dan Waktu

Pada gambar 3.3 penentuan lokasi adalah pemulaian spesiasi gep dalam membentuk ADF disini data mulai beragam diantaranya data kontinu seperti jarak tempuh dan lokasi, dan data kombinatorial seperti durasi, untuk lokasi menggunakan bingmaps untuk terlihat keluaran

dalam bentuk visualisasi, kemudian akan dimasukan parameter lainnya yang mengharuskan spesiasi gep menangani beberapa indicator dengan jenis data yang berbeda agar menampilkan solusi optimal dengan beberapa kriteria dan informasi yang beragam.

3.3 Rencana Kerja

| Kegiatan | Bulan Ke- | | | | | | | | | | | | | | | | | |
|--|-----------|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
| Studi Pustaka | | | | | | | | | | | | | | | | | | |
| Pemilihan Topik | | | | | | | | | | | | | | | | | | |
| Penyusunan Proposal | | | | | | | | | | | | | | | | | | |
| Mencari dan mempelajari literatur | | | | | | | | | | | | | | | | | | |
| Mendesain algoritma spesiasi dalam pemrograman | | | | | | | | | | | | | | | | | | |
| Implementasi data algoritma | | | | | | | | | | | | | | | | | | |
| Pengujian algoritma | | | | | | | | | | | | | | | | | | |
| Publikasi Ilmiah/Seminar | | | | | | | | | | | | | | | | | | |
| Pengembangan atas saran seminar | | | | | | | | | | | | | | | | | | |
| Penyusunan Disertasi | | | | | | | | | | | | | | | | | | |
| Sidang Disertasi | | | | | | | | | | | | | | | | | | |

Bibliografi

- Andoyo, Fachrizal., Arifudin, Riza. “*Optimization of Classification Accuracy Using K-Means and Genetic Algorithm by Integrating C4.5 Algorithm for Diagnosis Breast Cancer Disease*”. Journal of Advances in Information Systems and Technology 3(1) April 2021, 1-8.
- C. Ferreira “*Gene expression programming: A new adaptive algorithm for solving problems,*” Complex Syst., vol. 13, no. 2, pp. 87–129, 2001
- Goralski, M. A., & Tan, T. K. (2020). Artificial intelligence and sustainable development. The International Journal of Management Education, 18(1), 100330.
- Hale, W.G., J.P. Margham, and V.A. Saunders. 1995. *Collins Dictionary of Biology*. Harper Collins Publishers. Glasgow G4 0NB.
- Herimanto., Zarlis, Muhammad., Efendi, Syahril. “*Improving Performance of The Genetic Algorithm on NP-Complete Problem*”. Jurnal Nasional Informatika Dan Teknologi Jaringan- Vol. 6 No. 1 (2021).
- Jinghui Zhong, Liang Feng, Yew-soon Ong (2017). *Gene Expression Programming : A Survey*. IEEE Computational Intelligence Magazine, august 2017
- [Koza] J. R. Koza and J. Noyes, Genetic Programming II Videotape: The Next Generation. Cambridge, MA: MIT Press, 1994
- S. Jiang and S. Yang, “*A steady-state and generational evolutionary algorithm for dynamic multiobjective optimization,*” IEEE Trans. Evolut. Comput., vol. 21, no. 1, pp. 65–82, Feb. 2017
- S. Yang, “*Evolutionary computation for dynamic optimization problems,*” in Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, ser. GECCO Companion '15. New York, NY, USA: ACM, 2015, pp. 629–649
- Trung Thanh Nguyen, “*Continuous dynamic optimisation using evolutionary algorithms*”.2011.
- Wang, Huanhuan., Liu, Shuming., et-all. “*Gene Expression Programming Algorithms for Optimization of Water Distribution Network*”. Procedia Engineering 37 (2012) 359 – 264.