

**PENGARUH RASIO BANYAKNYA *EMPLOYED BEE* DAN
ONLOOKER BEE TERHADAP PERFORMA ALGORITMA
ARTIFICIAL BEE COLONY PADA PENYELESAIAN
*TRAVELING SALESMAN PROBLEM***

SKRIPSI

**OLEH
YUNIAR RIZQI
NIM. 220601110021**



**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026**

**PENGARUH RASIO BANYAKNYA *EMPLOYED BEE* DAN
ONLOOKER BEE TERHADAP PERFORMA ALGORITMA
ARTIFICIAL BEE COLONY PADA PENYELESAIAN
*TRAVELING SALESMAN PROBLEM***

SKRIPSI

**Diajukan Kepada
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang
untuk Memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Matematika (S.Mat)**

**Oleh
Yuniar Rizqi
NIM. 220601110021**

**PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM MALANG
2026**

**PENGARUH RASIO BANYAKNYA *EMPLOYED BEE* DAN
ONLOOKER BEE TERHADAP PERFORMA ALGORITMA
ARTIFICIAL BEE COLONY I ADA PENYELESAIAN
*TRAVELING SALESMAN PROBLEM***

SKRIPSI

Oleh
Yuniar Rizqi
NIM. 220601110021

Telah Disetujui Untuk Diuji

Malang, 30 April 2026

Dosen Pembimbing I

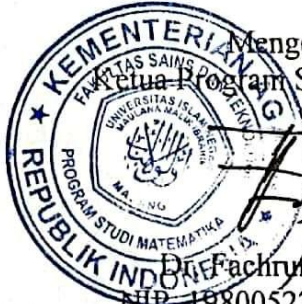

Mohammad Nafie Jauhari, M.Si
NIPPPK. 19870218 202321 1 018

Dosen Pembimbing II


Abdul Aziz, M.Si
NIP. 19760318 200604 1 002

Mengetahui,
Ketua Program Studi Matematika

Diefachnur Rozi, M.Si
NIP. 19800527 200801 1 012



**PENGARUH RASIO BANYAKNYA *EMPLOYED BEE* DAN
ONLOOKER BEE TERHADAP PERFORMA ALGORITMA
ARTIFICIAL BEE COLONY PADA PENYELESAIAN
*TRAVELING SALESMAN PROBLEM***

SKRIPSI

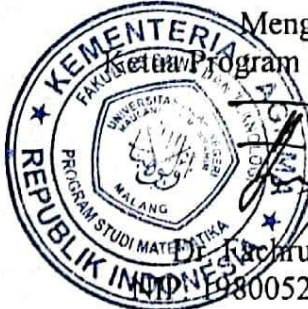
Oleh
Yuniar Rizqi
NIM. 220601110021

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima sebagai Salah Satu Persyaratan
untuk Memperoleh Gelar Sarjana Matematika (S.Mat)

Tanggal 5 Juni 2026

Ketua Penguji : Hisyam Fahmi, M.Kom
Anggota Penguji 1 : Muhammad Khudzaifah, M.Si
Anggota Penguji 2 : Mohammad Nafie Jauhari, M.Si
Anggota Penguji 3 : Abdul Aziz, M.Si

.....
.....
.....
.....

Mengetahui,
Ketua Program Studi Matematika

Dr. Fachrur Rozi, M.Si
NIM. 2206027 200801 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini

Nama : Yuniar Rizqi
NIM : 220601110021
Program Studi : Matematika
Fakultas : Sains dan Teknologi
Judul Skripsi : Pengaruh Rasio Banyaknya *Employed Bee* dan *Onlooker bee* terhadap Performa Algoritma *Artificial Bee Colony* pada Penyelesaian *Traveling Salesman Problem*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini merupakan hasil karya sendiri, bukan pengambilan tulisan atau pemikiran orang lain yang saya akui sebagai pemikiran saya, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka di halaman terakhir. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi yang berlaku atas perbuatan tersebut.

Malang, 5 Juni 2026



Yuniar Rizqi
NIM. 220601110021

MOTO

“Hidup itu sudah tentu naik-turun. Maka nikmati saja prosesnya, pungut hikmahnya. Saat naik rendah hatilah. Saat turun, tabahlah. Sungguh bersama kesulitan pasti ada kemudahan”

(Habib Husein Ja'far Al Hadar)

“Selalu ada seleksi, pejuang yang sampai pada kesuksesan hanyalah yang sabar. Dan kita diberi pilihan, menjadi manusia yang mudah rapuh oleh tantangan atau justru menghebat seiring hebatnya rintangan. Percayalah, badai selalu menyisakan pohon-pohon terkuat.”

“Akan tiba saatnya nanti, air matamu akan jatuh bukan karena masalah, tetapi karena Do'amu telah Allah kabulkan”

(KH. Maimun Zubair)

PERSEMBAHAN

Bismillahirrahmanirrahim. Dengan memanjatkan puji syukur kepada Allah SWT, perjalanan dalam menyusun skripsi yang melalui proses yang cukup panjang dari mulai perencanaan sampai penulisan skripsi ini akhirnya telah selesai. Skripsi ini penulis persembahkan untuk:

1. Orang tua tercinta, Bapak Suparji dan Ibu Sri Purwaningsih. Terima kasih untuk doa yang tidak pernah terputus, segala bentuk dukungannya, dan keyakinan bahwa saya pasti bisa menempuh pendidikan ini.
2. Mas Arif sekeluarga, Mas Zainul sekeluarga, dan Mas Rudin yang selalu membantu dan mengarahkan penulis untuk proses ini, tempat bercerita, dan tempat bergembira di tengah banyaknya revisi.
3. Para dosen Pembimbing, atas bimbingannya kepada penulis dan kepercayaannya sehingga skripsi ini terselesaikan dengan baik.
4. Mustaqim, Fatul, Badi', Sasa, dan Lu'lu, teman-teman bidang Teori Graf yang berjuang bersama-sama dari awal sampai akhir. Terima kasih atas diskusi, dukungan, dan kebersamaannya.
5. Teman-teman yang selalu membantu dan memotivasi dalam setiap kesulitan yang penulis hadapi yang penulis tidak dapat menyebutkan satu per satu
6. Diri saya sendiri, yang telah berani berproses mengambil keputusan yang baik, yang mau berjuang dan berusaha penuh dalam menghadapi segala tantangan selama menempuh jenjang pendidikan ini.

Semoga setiap doa, dukungan, dan kebaikan yang kalian berikan dibalas oleh Allah dengan balasan yang terbaik dan kebaikan yang banyak.

KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Allah SWT atas segala limpahan rahmat, taufik, dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi ini dengan judul “Pengaruh Rasio Banyaknya *Employed Bee* dan *Onlooker Bee* terhadap Performa Algoritma *Artificial Bee Colony* pada Penyelesaian *Traveling Salesman Problem*” sebagai bagian dari persiapan dalam rangkaian kegiatan akademik untuk memperoleh gelar Sarjana pada Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang.

Dalam proses penyusunan skripsi ini, penulis banyak menerima masukan berupa bimbingan, dukungan, serta bantuan dari berbagai pihak. Oleh karena itu, penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim, yang telah memberikan kesempatan untuk mencari ilmu di universitas ini.
2. Dr. Agus Mulyono, M.Kes, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim, atas kebijakannya yang telah memfasilitasi kegiatan akademik ini.
3. Dr. Fachrur Rozi, M.Si, selaku Ketua Program Studi Matematika, Universitas Islam Negeri Maulana Malik Ibrahim, atas bimbingan, dukungan, dan motivasinya selama proses belajar di program studi ini.
4. Mohammad Nafie Jauhari, M.Si, selaku Dosen Pembimbing I, atas bimbingan, saran, dan dukungannya dalam penyusunan skripsi ini.
5. Abdul Aziz, M.Si, selaku Dosen Pembimbing II, atas bimbingan, saran, dan dukungannya dalam penyusunan skripsi ini.
6. Hisyam Fahmi, M.Kom, selaku Penguji Utama dalam serangkaian ujian skripsi atas saran dan bimbingannya.
7. Muhammad Khudzaifah, M.Si, selaku Anggota Penguji 1 serangkaian ujian skripsi atas saran dan bimbingannya.

8. Seluruh dosen Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim, atas ilmu dan bimbingan yang telah diberikan selama proses belajar.
9. Orang tua dan seluruh keluarga, atas doa dan dukungan yang tidak ada hentinya.
10. Kakak tingkat yang telah membantu dan mendukung untuk menyelesaikan skripsi ini.
11. Seluruh mahasiswa angkatan 22, atas kebersamaan, dukungan, dan semangatnya selama proses belajar.

Penulis menyadari bahwa skripsi ini masih banyak kekurangan. Oleh karena itu, penulis bersedia menerima saran dan masukan dari segala pihak untuk memperbaiki setiap kekurangan yang ada. Semoga skripsi ini dapat memberikan manfaat kepada pembaca khususnya dalam bidang ilmu pengetahuan.

Malang, 5 Juni 2026

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGANTAR.....	ii
HALAMAN PERSETUJUAN.....	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTO	vi
PERSEMBAHAN.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	x
DAFTAR TABEL.....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR SIMBOL	xiv
DAFTAR LAMPIRAN	xv
ABSTRAK	xvi
ABSTRACT	xvii
مستخلص البحث.....	xviii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	5
1.3 Tujuan Penelitian.....	5
1.4 Manfaat Penelitian.....	5
1.5 Batasan Masalah	6
1.6 Definisi Istilah	7
BAB II KAJIAN TEORI	9
2.1 Teori Pendukung.....	9
2.1.1 Teori Graf.....	9
2.1.2 <i>Traveling Salesman Problem</i>	12
2.1.3 <i>Algoritma Artificial Bee Colony</i>	14
2.1.4 <i>Uji Analysis of Variance</i>	26
2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits.....	28
2.3 Kajian Topik Dengan Teori Pendukung.....	30
BAB III METODE PENELITIAN	32
3.1 Jenis Penelitian	32
3.2 Data dan Sumber Data.....	32
3.3 Tahapan Penelitian	32
BAB IV HASIL DAN PEMBAHASAN.....	37
4.1 Persiapan Data	37
4.2 Inisialisasi Parameter.....	37
4.3 Menetapkan Lima Variasi Rasio <i>Employed Bee</i> dan <i>Onlooker Bee</i>	40
4.4 Implementasi Algoritma dan Eksperimen	41
4.4.1 Hasil dari Graf Pertama.....	50
4.4.2 Hasil dari Graf Kedua	51
4.4.3 Hasil dari Graf Ketiga	53
4.5 Analisis Data Statistik dan Penarikan Kesimpulan	55

4.5.1 Uji Normalitas Data	55
4.5.2 Uji Homogenitas	57
4.5.3 Uji <i>One Way</i> ANOVA.....	59
4.6 Pembahasan	64
4.7 Kajian Penelitian dalam Perspektif Islam.....	65
BAB V PENUTUP	68
5.1 Kesimpulan.....	68
5.2 Saran untuk Penelitian Lanjutan.....	69
DAFTAR PUSTAKA	70
LAMPIRAN.....	72
RIWAYAT HIDUP	81

DAFTAR TABEL

Tabel 4.1	Hasil Fase Inisialisasi.....	42
Tabel 4.2	Perhitungan <i>Swap Operator</i>	42
Tabel 4.3	Individu Terpilih	43
Tabel 4.4	Perhitungan <i>Swap Sequence</i>	44
Tabel 4.5	Perhitungan Probabilitas Kumulatif.....	45
Tabel 4.6	<i>Roulette Wheel Selection</i>	46
Tabel 4.7	Perhitungan <i>Insert Operator</i>	46
Tabel 4.8	Individu Terpilih	47
Tabel 4.9	Perhitungan <i>Insert Sequence</i>	48
Tabel 4.10	Evaluasi Solusi Akhir.....	49

DAFTAR GAMBAR

Gambar 1.2	Graf <i>A</i>	9
Gambar 2.2	Graf <i>B</i>	10
Gambar 2.3	Graf <i>C</i>	11
Gambar 2.4	Graf Komplit <i>K7</i>	12
Gambar 2.5	Ilustrasi TSP	13
Gambar 2.6	<i>Swap Operator SO(0,3)</i>	17
Gambar 2.7	<i>Insert Operator IO(1,4)</i>	21
Gambar 3.1	<i>Flowchart</i> Penelitian	36
Gambar 4.1	Grafik Data Hasil dari Graf Pertama	50
Gambar 4.2	Visualisasi Konvergensi Rasio 0,1 dari Graf 50 Simpul	51
Gambar 4.3	Grafik Data Hasil dari Graf Kedua	52
Gambar 4.4	Visualisasi Konvergensi Rasio 0,1 dari Graf 100 Simpul	53
Gambar 4.5	Grafik Data Hasil dari Graf Ketiga	53
Gambar 4.6	Visualisasi Konvergensi Rasio 0,1 dari Graf 200 Simpul	54
Gambar 4.7	Hasil Uji Normalitas Data dari Graf Pertama	56
Gambar 4.8	Hasil Uji Normalitas Data dari Graf Kedua	56
Gambar 4.9	Hasil Uji Normalitas Data dari Graf Ketiga	57
Gambar 4.10	Hasil Uji Homogenitas Data dari Graf Pertama	58
Gambar 4.11	Hasil Uji Homogenitas Data dari Graf Kedua	58
Gambar 4.12	Hasil Uji Homogenitas Data dari Graf Ketiga	59
Gambar 4.13	Hasil Uji ANOVA Graf Pertama	60
Gambar 4.14	Hasil Uji <i>Post-Hoc</i> Graf Pertama	60
Gambar 4.15	Hasil Uji ANOVA Graf Kedua	61
Gambar 4.16	Hasil Uji <i>Post-Hoc</i> Graf Kedua	62
Gambar 4.17	Hasil Uji ANOVA Graf Ketiga	62
Gambar 4.18	Hasil Uji <i>Post-Hoc</i> Graf Ketiga	63

DAFTAR SIMBOL

G	: Graf
$V(G)$	: Himpunan simpul (<i>vertices</i>) dari G
$E(G)$	: Himpunan sisi (<i>edge</i>) dari G
$\{u, v\}$	: Sisi yang menghubungkan simpul u dan v
$w(u, v)$	: Bobot pada sisi antara simpul u dan v
K_n	: Graf komplet dengan n simpul
$(v_0, v_1, \dots, v_k)$	: Barisan simpul-simpul secara berurutan
$d_{x_i x_j}$	: Jarak atau bobot antara kota i ke kota j
NP	: Ukuran populasi atau banyaknya lebah
SN	: Banyaknya sumber makanan (<i>food sources</i>)
NP_e	: Banyaknya <i>employed bee</i>
NP_o	: Banyaknya <i>onlooker bee</i>
r_c	: Rasio banyaknya <i>employed bee</i> dan <i>onlooker bee</i>
x_{ij}	: Simpul dari solusi ke- i pada urutan ke- j
fit_i	: Nilai <i>fitness</i> suatu solusi ke- i
p	: <i>P Value</i> , yaitu nilai probabilitas yang menyatakan signifikansi

DAFTAR LAMPIRAN

Lampiran 1 Visualisasi Graf Komplit 50, 100, dan 200 Simpul.....	72
Lampiran 2 <i>Source Code Python</i> Implementasi Algoritma ABC.....	73
Lampiran 3 Data Hasil Eksperimen dari Graf 50 Simpul.....	73
Lampiran 4 Data Hasil Eksperimen dari Graf 100 Simpul.....	75
Lampiran 5 Data Hasil Eksperimen dari Graf 200 Simpul.....	78

ABSTRAK

Rizqi, Yuniar. 2026. **Pengaruh Rasio Banyaknya *Employed Bee* dan *Onlooker Bee* terhadap Performa Algoritma *Artificial Bee Colony* pada Penyelesaian *Traveling Salesman Problem***. Skripsi. Program Studi Matematika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Mohammad Nafie Jauhari, M.Si. (II) Abdul Aziz, M.Si.

Kata kunci: *Artificial Bee Colony*, *Employed Bee*, *Onlooker bee*, Optimasi, Algoritma Metaheuristik, *Traveling Salesman Problem*.

Algoritma *Artificial Bee Colony* (ABC) merupakan metode optimasi metaheuristik yang terinspirasi dari perilaku pencarian makanan oleh koloni lebah madu, yang dapat diadaptasi untuk menyelesaikan *Traveling Salesman Problem* (TSP). Dalam algoritma ABC, dinamika eksplorasi dan eksploitasi mempengaruhi performa algoritma dalam menemukan solusi optimal. Secara standar, algoritma ABC menetapkan rasio banyaknya *employed bee* dan *onlooker bee* sebesar 1:1, namun konfigurasi ini dinilai kurang memiliki mekanisme efektif untuk mengontrol tingkat eksplorasi dan eksploitasi. Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma ABC dalam menemukan jarak rute terpendek pada TSP, serta mengidentifikasi rasio yang paling optimal. Metode penelitian ini menggunakan pendekatan kuantitatif eksperimental. Pengujian dilakukan dengan menerapkan lima variasi rasio (1:9, 3:7, 1:1, 7:3, dan 9:1) pada tiga model graf komplit berbobot acak yang masing-masing berukuran 50, 100, dan 200 simpul yang mempresentasikan masalah TSP. Untuk memastikan validitas dan kestabilan data, algoritma dijalankan sebanyak 20 kali pengulangan pada setiap variasi rasio. Analisis data statistik dilakukan menggunakan uji ANOVA. Hasil pengujian memberikan nilai signifikansi 0,000, yang membuktikan bahwa variasi rasio tersebut memberikan pengaruh yang signifikan secara statistik terhadap panjang rute terpendek yang dihasilkan. Analisis lebih lanjut melalui uji *Post-Hoc* menunjukkan bahwa rasio 1:9 menghasilkan performa paling optimal pada skenario eksperimen penelitian ini. Temuan ini menunjukkan bahwa peningkatan proporsi *onlooker bee* secara efektif memperkuat intensitas mekanisme eksploitasi, sehingga algoritma menjadi lebih terarah dalam menyempurnakan kualitas solusi terbaik.

ABSTRACT

Rizqi, Yuniar. 2026. **The Effect of Ratio the Number of Employed Bees and Onlooker Bees on the Performance of the Artificial Bee Colony Algorithm in Solving the Traveling Salesman Problem.** Thesis. Mathematics Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim, Malang. Advisors: (I) Mohammad Nafie Jauhari, M.Si. (II) Abdul Aziz, M.Si.

Keywords: Artificial Bee Colony, Employed Bee, Onlooker Bee, Optimization, Metaheuristic Algorithm, Traveling Salesman Problem.

The Artificial Bee Colony (ABC) algorithm is a metaheuristic optimization method inspired by the foraging behavior of honeybee colonies, which can be adapted to solve the Traveling Salesman Problem (TSP). In the ABC algorithm, the dynamics of exploration and exploitation influence the algorithm's performance in finding optimal solutions. By default, the ABC algorithm sets the ratio of employed bees to onlooker bees at 1:1; however, this configuration is considered to lack an effective mechanism for controlling the levels of exploration and exploitation. Therefore, this study aims to evaluate the effect of varying the ratio of employed bees and onlooker bees on the ABC algorithm's performance in finding the shortest route distance in the TSP, as well as to identify the optimal ratio. This study employs a quantitative experimental approach. Testing was conducted by applying five ratio variations (1:9, 3:7, 1:1, 7:3, and 9:1) to three randomly weighted complete graph models, each with 50, 100, and 200 nodes, representing the TSP. To ensure the validity and stability of the data, the algorithm was run 20 times for each ratio variation. Statistical data analysis was performed using an ANOVA test. The results provide a significance value of 0,000, demonstrating that variations in the ratio had a statistically significant effect on the length of the shortest route generated. Further analysis via a Post-Hoc test showed that the 1:9 ratio produced the optimal performance in the experimental scenarios of this study. This finding shows that increasing the proportion of onlooker bees effectively strengthens the intensity of the exploitation mechanism, thereby making the algorithm more focused on refining the quality of the best solution.

مستخلص البحث

رزقي، يونيار. ٢٠٢٦. تأثير نسبة عدد النحلات العاملة والنحلات المراقبة على أداء خوارزمية مستعمرة النحل الاصطناعية في حل مشكلة البائع المتجول. البحث العلمي. قسم الرياضيات، كلية العلوم والتكنولوجيا، جامعة مولانا مالك إبراهيم الإسلامية الحكومية في مالانج. المشرفان (١) محمد نافع جوهرى، الماجستير في العلوم (٢) عبد العزيز، الماجستير في العلوم.

الكلمات الأساسية: مستعمرة النحل الاصطناعية، النحلة العاملة، النحلة المتفرجة، التحسين، خوارزمية الميتاهيورستيك، مشكلة البائع المتجول.

تُعد خوارزمية مستعمرة النحل الاصطناعية (ABC) طريقة تحسين ميتاهيورستيك مستوحاة من سلوك البحث عن الغذاء لدى طوائف نحل العسل، والتي يمكن تكييفها لحل مشكلة البائع المتجول (TSP). في خوارزمية ABC ، تؤثر ديناميكيات الاستكشاف ($exploration$) والاستغلال ($exploitation$) على أداء الخوارزمية في إيجاد الحلول المثلى. افتراضياً، تحدد خوارزمية ABC نسبة النحل العامل ($employed bees$) إلى النحل المراقب ($onlooker bees$) بمقدار ١:١؛ ومع ذلك، يُعد هذا التكوين مفتقراً إلى آلية فعالة للتحكم في مستويات الاستكشاف والاستغلال. لذلك، هدفت هذه الدراسة إلى تقييم تأثير تباين نسبة النحل العامل إلى النحل المراقب على أداء خوارزمية ABC في إيجاد أقصر مسافة للمسار في مشكلة TSP ، بالإضافة إلى تحديد النسبة المثلى. يستخدم هذا البحث المنهج الكمي التجريبي، حيث أُجريت الاختبارات بتطبيق خمسة تباينات في النسب (١:٩، ٣:٧، ١:١، ٧:٣، ٩:١) على ثلاثة نماذج لمخططات كاملة ($complete graphs$) عشوائية الأوزان، تتكون من ٥٠ و ١٠٠ و ٢٠٠ عقدة، والتي تمثل مشكلة TSP . ولضمان صحة البيانات واستقرارها، تم تشغيل الخوارزمية ٢٠ مرة لكل تباين في النسبة. أُجري تحليل البيانات الإحصائية باستخدام اختبار تحليل التباين ($ANOVA$)، وذلك بعد التأكد من استيفاء البيانات لافتراضات التوزيع الطبيعي عبر اختبار شايبرو-ويلك ($Shapiro-Wilk$) وتجانس التباين عبر اختبار ليفين ($Levene$). وقد أسفرت النتائج عن قيمة دلالة إحصائية تبلغ ٠,٠٠٠، مما يثبت أن تباين النسب له تأثير ذو دلالة إحصائية على طول أقصر مسار ناتج. كما أظهر تحليل إضافي من خلال اختبار البعدي ($Post-Hoc$) أن النسبة ١:٩ حققت الأداء الأمثل في السيناريوهات التجريبية لهذه الدراسة. ويشير هذا الاكتشاف إلى أن زيادة نسبة النحل المراقب تعزز بشكل فعال من كثافة آلية الاستغلال، مما يجعل الخوارزمية أكثر تركيزاً على تحسين جودة الحل الأفضل.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Optimasi merupakan proses pencarian solusi terbaik berdasarkan suatu fungsi objektif dalam ruang pencarian suatu permasalahan (Karaboga, 2014). Untuk menyelesaikan permasalahan optimasi di dunia nyata khususnya masalah yang bersifat *NP-Hard* di mana kompleksitas komputasi meningkat secara eksponensial terhadap dimensi masalah sehingga menjadikan metode eksak sangat sulit untuk memecahkannya, penerapan algoritma metaheuristik sering dilakukan. Algoritma metaheuristik merupakan metode optimisasi tingkat tinggi yang meniru fenomena alam atau perilaku sosial dalam menemukan suatu solusi (Talbi, 2009). Meskipun algoritma metaheuristik telah menunjukkan performa yang baik untuk masalah nonlinier dan berskala besar, peningkatan efisiensi dan kecepatan konvergensi tetap perlu dilakukan agar hasilnya lebih stabil (Talbi, 2009). Hal ini karena sebagian besar algoritma metaheuristik bersifat stokastik dan sangat bergantung pada konfigurasi parameter tertentu yang memengaruhi hasil pencarian solusi.

Dalam pandangan Islam, konsep optimasi tercermin dalam prinsip keteraturan dan efisiensi ciptaan Allah SWT sebagaimana difirmankan dalam Al-Qur'an surat Al-Anbiya ayat 16 yang artinya:

"Kami tidak menciptakan langit dan bumi dan apa yang ada di antara keduanya dengan main-main." (QS. Al-Anbiya [21]: 16).

Ayat ini menjelaskan bahwa setiap ciptaan Allah memiliki tujuan dan keteraturan yang jelas, tanpa ada proses yang sia-sia. Nilai ini sejalan dengan prinsip dasar optimasi dalam ilmu pengetahuan, yaitu upaya untuk memperoleh hasil paling efisien dan bermakna melalui proses yang terarah dan terukur (Shihab, 2002).

Salah satu kelompok dari algoritma metaheuristik adalah kecerdasan kawanan (*Swarm Intelligence*), yang berkaitan dengan perancangan algoritma optimisasi cerdas berdasarkan perilaku kolektif makhluk sosial seperti burung, ikan, dan lebah (Talbi, 2009). Di antara algoritma yang termasuk dalam kelompok *Swarm Intelligence* adalah *Artificial Bee Colony* (ABC). Algoritma ABC pertama kali diperkenalkan oleh Karaboga pada tahun 2005 untuk menyelesaikan masalah optimisasi numerik yang terinspirasi pada perilaku lebah madu dalam mencari sumber makanan yang berkualitas (Karaboga, 2005). Dalam algoritma ini, koloni lebah dibagi menjadi tiga kelompok: *employed bee*, *onlooker bee*, dan *scout bee*. Pada fase *employed bee*, algoritma melakukan proses eksplorasi untuk mencari solusi baru di sekitar sumber makanan yang telah ditemukan. Pada fase *onlooker bee*, algoritma melakukan proses eksploitasi dengan memilih sumber makanan terbaik berdasarkan probabilitas fungsi *fitness*, dan *scout bee* bertugas menemukan sumber makanan atau solusi baru ketika solusi lama tidak lagi produktif (Karaboga & Basturk, 2007). Dalam algoritma ABC dasar, rasio banyaknya *employed bee* dan *onlooker bee* adalah 1:1 (Karaboga, 2005).

ABC telah diterapkan dalam berbagai bidang dunia nyata, seperti peramalan keuangan, penjadwalan produksi, pengelompokan data (*clustering*), segmentasi citra, dan optimisasi sistem tenaga listrik. Pada perkembangannya, algoritma ABC telah diadaptasi untuk menyelesaikan permasalahan optimisasi diskrit, salah satunya adalah *Traveling Salesman Problem* (TSP), yang merupakan permasalahan yang bersifat *NP-Hard* (Lin & Kernighan, 1973). TSP merupakan pencarian rute terpendek yang memungkinkan seorang sales mengunjungi seluruh kota tepat satu kali dan kembali ke kota asal. Karena struktur ruang solusi TSP yang kompleks dan

banyaknya kemungkinan jalur yang sangat besar, metode eksak menjadi tidak efisien (Khan & Maiti, 2019). Dalam hal ini, algoritma metaheuristik seperti ABC menjadi alternatif yang efektif karena dapat memberikan solusi mendekati optimal dengan waktu komputasi yang relatif singkat (Shi dkk., 2007).

Namun demikian, seperti algoritma metaheuristik berbasis populasi lainnya, ABC masih memiliki keterbatasan bawaan (Banharnsakun dkk., 2011). Salah satu kelemahan utamanya adalah kecepatan konvergensi yang relatif lebih lambat dibandingkan algoritma metaheuristik lain seperti *Differential Evolution* (DE) dan *Particle Swarm Optimization* (PSO) (Qin dkk., 2015). Selain itu, ABC juga cenderung mengalami konvergensi prematur ketika menangani masalah kompleks (Xiang & An, 2013). Untuk mengatasi keterbatasan tersebut, banyak penelitian mengusulkan berbagai varian dari algoritma ABC. Secara umum, modifikasi terhadap algoritma ABC dasar ini dikelompokkan menjadi tiga pendekatan yaitu pengaturan parameter konfigurasi (*parameter tuning*), hibridisasi dengan operator evolusioner lain untuk meningkatkan performa, dan perancangan strategi pembelajaran baru (Qin dkk., 2015).

Diketahui bahwa eksplorasi dan eksploitasi adalah dua faktor paling penting yang mempengaruhi performa algoritma optimisasi berbasis populasi (Karaboga & Basturk, 2007). Algoritma optimisasi berbasis solusi yang baik harus mampu untuk menyeimbangkan kedua tujuan yang saling bertentangan ini dengan tepat (Cheng dkk., 2014). Hasil pengamatan menunjukkan bahwa ABC dasar memiliki kemampuan eksplorasi yang baik tetapi eksploitasi yang relatif lemah, yang dapat menghambat algoritma untuk mencapai optimum global (Zhu & Kwong, 2010).

Allah SWT menjelaskan tentang perilaku lebah madu dalam mencari makanan yang menjadi dasar dari pengembangan algoritma *Artificial Bee Colony* (ABC) dalam Surat An-Nahl ayat 69 yang artinya:

“Kemudian makanlah dari tiap-tiap (macam) buah-buahan dan tempuhlah jalan Tuhanmu yang telah dimudahkan (bagimu). Dari perut lebah itu keluar minuman (madu) yang bermacam-macam warnanya, di dalamnya terdapat obat yang menyembuhkan bagi manusia. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kebesaran Allah) bagi orang-orang yang berpikir.” (QS. An-Nahl [16]: 68–69).

Ayat ini menjelaskan bahwa perilaku lebah bukan hanya hasil insting biologis, tetapi merupakan hasil dari wahyu dan bimbingan langsung dari Allah SWT, yang mengatur setiap aspek kehidupannya secara sistematis dan efisien. Dalam konteks sains modern, pola ini dapat dipahami sebagai cerminan prinsip optimasi alami, yaitu kemampuan makhluk hidup dalam menemukan solusi paling efektif untuk bertahan dan memproduksi (Shihab, 2002).

Algoritma ABC memiliki sedikit parameter sehingga mudah untuk disesuaikan, namun kurang memiliki cara yang efektif untuk mengontrol kemampuan eksplorasi dan eksploitasi. Oleh karena itu, penelitian ini mengusulkan penyesuaian rasio banyaknya *employed bee* dan *onlooker bee* sebagai parameter baru yang dapat memengaruhi proses pencarian solusi. Fokus penelitian ini adalah untuk mengetahui pengaruh rasio tersebut terhadap performa algoritma ABC dalam penyelesaian *Traveling Salesman Problem* (TSP). Melalui pengujian beberapa rasio, penelitian ini diharapkan dapat mengidentifikasi konfigurasi rasio yang memberikan kualitas solusi dan konvergensi terbaik.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka rumusan masalah dalam penelitian ini dapat dirumuskan sebagai berikut:

1. Bagaimana pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma ABC dalam menyelesaikan TSP?
2. Apakah terdapat rasio tertentu dari banyaknya *employed bee* dan *onlooker bee* yang menghasilkan performa optimal dalam hal kualitas solusi?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini adalah:

1. Mengevaluasi pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma ABC pada penyelesaian TSP.
2. Menentukan rasio banyaknya *employed bee* dan *onlooker bee* yang paling optimal dalam meningkatkan kualitas solusi pada algoritma ABC.

1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah:

1. Bagi Penulis:

Penelitian ini memberikan pengalaman langsung dalam optimasi algoritma metaheuristik dengan mengevaluasi pengaruh rasio banyaknya *employed* dan *onlooker bee* pada performa algoritma ABC dan melatih kemampuan analitis.

2. Bagi Pembaca:

Penelitian ini dapat menjadi referensi dan sumber pembelajaran bagi pembaca atau peneliti yang tertarik dalam bidang optimasi algoritma metaheuristik, khususnya yang berhubungan dengan *swarm intelligence* dan algoritma ABC.

3. Bagi Instansi:

Penelitian ini dapat menyumbang penemuan terkait optimasi di lingkungan akademik, terutama dalam pengembangan kajian algoritma metaheuristik serta mendukung peningkatan kualitas penelitian di bidang ilmu komputer dan matematika terapan.

1.5 Batasan Masalah

Agar penelitian ini memiliki arah yang jelas dan fokus pada tujuan yang ingin dicapai, maka ruang lingkup penelitian dibatasi sebagai berikut:

1. Parameter yang divariasikan dalam penelitian ini adalah rasio banyaknya *employed bee* dan *onlooker bee*, sedangkan parameter lain seperti ukuran koloni (*colony size*), batas percobaan (*limit*), dan banyaknya iterasi maksimum (*cycles*) dibuat konstan pada setiap percobaan.
2. *Traveling Salesman Problem* (TSP) pada penelitian ini dimodelkan sebagai graf komplit yang dibangkitkan secara acak menggunakan *Python*, bukan graf nyata hasil observasi lapangan. Pemodelan ini didasarkan pada formulasi matematis TSP, di mana permasalahan didefinisikan sebagai pencarian siklus Hamilton minimum pada graf komplit berbobot. Penggunaan graf komplit menjamin bahwa setiap permutasi simpul membentuk solusi yang *feasible*, sehingga ruang solusi TSP dapat direpresentasikan secara utuh sebagai himpunan semua siklus Hamilton pada graf tersebut.
3. *Neighborhood Operator* yang digunakan dalam penelitian ini adalah *swap operator*, *swap sequence*, *insert operator*, dan *insert sequence*. Keempat *operator* tersebut digunakan untuk menjalankan algoritma ABC dalam menyelesaikan TSP.

4. Pengujian algoritma pada penelitian ini dibatasi sebanyak 50 iterasi sebagai acuan evaluasi yang seragam pada variasi rasio yang diuji. Dalam penelitian ini, performa algoritma dinilai berdasarkan kualitas solusi terbaik yang diperoleh dalam batas iterasi tersebut, tanpa mengklaim jaminan tercapainya solusi optimal global.
5. Metrik performa algoritma ABC dalam penelitian ini dibatasi hanya pada panjang rute terpendek yang dihasilkan. Parameter lain seperti kecepatan konvergensi tidak diteliti, sehingga evaluasi kinerja algoritma difokuskan pada kualitas solusi yang diperoleh.

1.6 Definisi Istilah

Beberapa istilah penting didefinisikan sebagai berikut:

1. *Artificial Bee Colony* (ABC): Algoritma optimasi metaheuristik berbasis perilaku lebah madu dalam mencari sumber makanan, yang dikembangkan oleh Dervis Karaboga (2005). Algoritma ini terdiri dari tiga jenis lebah utama yaitu *employed bee*, *onlooker bee*, dan *scout bee*, yang bekerja secara kolaboratif untuk menemukan solusi optimal dalam suatu ruang pencarian.
2. *Employed bee*: Lebah pekerja yang berperan dalam mengeksplorasi sumber makanan yang telah ditemukan sebelumnya dan melakukan perbaikan solusi lokal melalui metode *swap operator* dan *swap sequence*. Pada penelitian ini, banyaknya *employed bee* dikendalikan berdasarkan rasio koloni yang ditentukan.
3. *Onlooker bee*: Lebah pengamat yang berperan mengeksploitasi sumber makanan dengan memilihnya berdasarkan nilai *fitness* dari informasi yang dibawa *employed bee*, kemudian memperbaikinya menggunakan operasi *insert*

operator dan *insert sequence*. Banyaknya *onlooker bee* juga ditentukan oleh rasio koloni yang digunakan dalam eksperimen.

4. *Scout Bee*: Lebah pengintai yang menggantikan solusi yang stagnan (tidak mengalami perbaikan dalam batas *limit* tertentu) dengan solusi baru yang dihasilkan secara acak. Tahapan ini menjaga keberagaman populasi dan mencegah konvergensi dini.
5. Nilai *Trial*: Nilai yang diberikan untuk menghitung banyaknya percobaan ketika suatu solusi gagal mengalami perbaikan.
6. Fungsi *fitness*: Fungsi evaluasi yang digunakan untuk menilai kualitas suatu solusi.
7. Rasio Banyaknya *Employed* dan *Onlooker bee*: Proporsi pembagian koloni lebah antara *employed bee* dan *onlooker bee*. Rasio ini memengaruhi dinamika antara proses eksploitasi dan eksplorasi dalam pencarian solusi optimal.
8. *Traveling Salesman Problem* (TSP): permasalahan dalam mencari rute terpendek yang dilalui seorang sales dengan mengunjungi seluruh kota di suatu daerah, tepat satu kali di tiap kota dan kembali ke kota awal.
9. Konvergensi Algoritma: Keadaan di mana nilai *fitness* tidak mengalami peningkatan setelah beberapa iterasi, yang menandakan algoritma mulai mendekati solusi optimal.

BAB II

KAJIAN TEORI

2.1 Teori Pendukung

2.1.1 Teori Graf

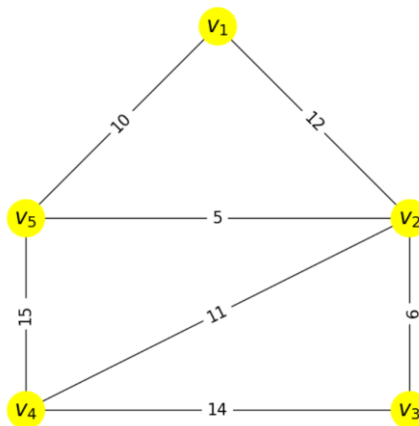
Definisi 2.1.1.1.

Graf adalah pasangan himpunan (V, E) ditulis dengan notasi $G = (V, E)$ di mana V adalah himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan E adalah himpunan sisi atau *edge* yang menghubungkan sepasang simpul. Setiap sisi $\{u, v\}$ di G dilambangkan dengan uv atau vu . Jika $e = uv$ adalah sebuah sisi di G , maka e dikatakan sisi yang menghubungkan u dan v (Chartrand dkk., 2015).

Setelah struktur dasar graf dirumuskan sebagai pasangan (V, E) , perluasan alami dari konsep ini adalah mempertimbangkan adanya fungsi yang memetakan setiap sisi ke suatu nilai numerik. Hal inilah yang menjadi dasar definisi graf berbobot.

Definisi 2.1.1.2.

Graf berbobot adalah graf yang setiap sisinya diberi suatu bobot (Chartrand dkk., 2015).



Gambar 1.2 Graf **A**

Pada Gambar 1.2 Graf *A* merupakan contoh dari graf berbobot. Selain bobot pada sisi, karakteristik lain yang berpengaruh terhadap struktur graf adalah arah pada setiap sisi. Berdasarkan arah pada sisi graf, graf dibedakan menjadi dua jenis yaitu graf berarah dan tidak berarah (Chartrand dkk., 2015).

1. Graf Berarah

Graf berarah adalah graf yang memiliki arah pada setiap sisinya. Pada graf berarah, $(u, v) \in E(G)$ dan $(v, u) \in E(G)$ menyatakan dua arah yang berbeda, sehingga $(u, v) \in E(G) \neq (v, u) \in E(G)$.

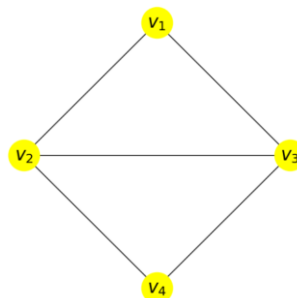
2. Graf Tidak Berarah

Graf tidak berarah adalah graf yang tidak memiliki arah pada setiap sisinya. Pada graf tidak berarah, $\{u, v\} \in E(G)$ dan $\{v, u\} \in E(G)$ tidak diberikan arah, sehingga $\{u, v\} \in E(G) = \{v, u\} \in E(G)$.

Dengan mengetahui apakah suatu sisi memiliki arah atau tidak, hubungan antar simpul dalam graf dapat didefinisikan secara lebih rinci melalui konsep ketetanggaan, yang menjelaskan keterhubungan langsung antar dua simpul.

Definisi 2.1.1.3.

Dua buah simpul pada graf tidak berarah G dikatakan bertetangga bila keduanya dihubungkan oleh sebuah sisi. Dengan kata lain, simpul u bertetangga dengan simpul v jika $\{u, v\} \in E(G)$ adalah sebuah sisi pada graf G (Chartrand dkk., 2015).



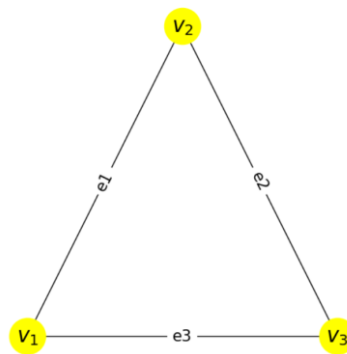
Gambar 2.2 Graf **B**

Pada Gambar 2.2 Graf B , simpul v_1 bertetangga dengan simpul v_2 dan v_3 , tetapi simpul v_1 tidak bertetangga dengan simpul v_4 .

Konsep ketetanggaan tersebut selanjutnya memungkinkan penyusunan urutan simpul yang saling terhubung, yang secara formal dinyatakan sebagai lintasan dalam graf.

Definisi 2.1.1.4.

Lintasan adalah barisan dari simpul $(v_0, v_1, \dots, v_k)$ di mana setiap dua simpul berurutan dihubungkan oleh sebuah sisi dan tidak ada simpul maupun sisi yang berulang (Chartrand dkk., 2015).



Gambar 2.3 Graf C

Contoh lintasan yang terdapat pada Gambar 2.3, di mana Graf C memuat lintasan atau barisan simpul (v_1, v_2, v_3) .

Lintasan yang telah didefinisikan sebelumnya memiliki bentuk khusus ketika simpul awal dan simpul akhir adalah simpul yang sama. Keadaan ini melahirkan konsep siklus sebagai objek penting dalam teori graf.

Definisi 2.1.1.5.

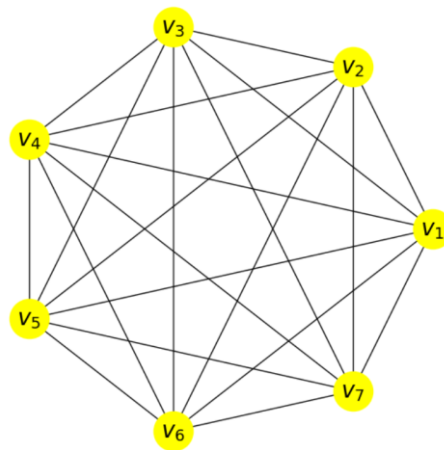
Siklus (*cycle*) adalah lintasan yang berakhir di simpul yang pertama, sehingga semua simpul pada lintasan tersebut berbeda kecuali simpul pertama yang muncul kembali sebagai simpul terakhir untuk menutup lintasan (Chartrand dkk., 2015).

Perhatikan Gambar 2.2 Graf B , barisan simpul $(v_1, v_2, v_3, v_4, v_1)$ adalah siklus karena membentuk lintasan yang berakhir di simpul yang pertama yaitu v_1 .

Keberadaan siklus dalam suatu graf sangat ditentukan oleh struktur keterhubungan antar simpul. Pada graf dengan tingkat keterhubungan maksimum, yaitu graf komplit, siklus-siklus tersebut muncul secara sistematis sebagai konsekuensi dari setiap pasangan simpul yang saling terhubung.

Definisi 2.1.1.6.

Graf komplit adalah graf tak berarah dengan setiap pasangan simpul yang berbeda di graf tersebut terhubung oleh satu buah sisi. Graf komplit yang terdiri atas n buah simpul dinyatakan dengan lambang K_n (Chartrand dkk., 2015).



Gambar 2.4 Graf Komplit K_7

Gambar 2.4 merupakan graf komplit K_7 , karena setiap simpul terhubung dengan semua simpul lainnya. Graf komplit menjamin keterhubungan antar setiap pasangan simpul, struktur ini menjadi model representasi yang sesuai bagi permasalahan optimasi rute seperti *Traveling Salesman Problem*.

2.1.2 *Traveling Salesman Problem*

Traveling Salesman Problem (TSP) merupakan permasalahan optimisasi kombinatorial yang bertujuan mencari rute terpendek bagi seorang penjual

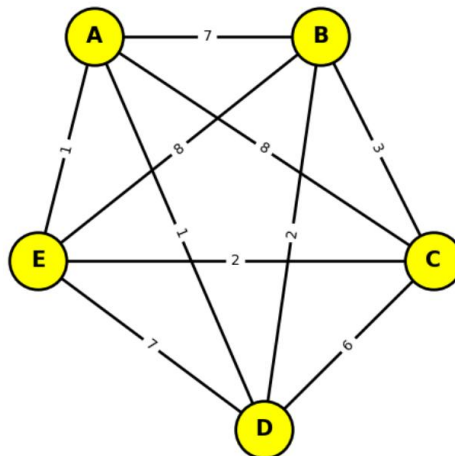
(*salesman*) untuk mengunjungi seluruh kota tepat satu kali dan kembali ke kota asal dengan total jarak atau biaya minimum. Dengan kata lain, TSP didefinisikan sebagai pencarian siklus Hamilton dengan total bobot terendah. Siklus Hamilton yaitu siklus yang melewati setiap simpul tepat satu kali dan kembali ke simpul awal (Khan & Maiti, 2019). Secara matematis, TSP didefinisikan sebagai berikut:

Misalkan $(x_0, x_1, \dots, x_{n-1}, x_0)$ adalah suatu rute dari seorang sales, di mana $x_j \in V$ dan setiap elemen x_j dalam rute adalah simpul atau kota yang berbeda kecuali kota pertama yang diulang di akhir untuk menutup rute. Maka, masalah di atas dapat disederhanakan menjadi bentuk berikut:

Diberikan sebuah rute $(x_0, x_1, \dots, x_{n-1}, x_0)$, maka untuk meminimalkan

$$Z = \sum_{j=0}^{n-2} d_{x_j x_{j+1}} + d_{x_{n-1} x_0} \quad (2.1)$$

di mana $d_{x_j x_{j+1}}$ adalah jarak dari kota x_j ke kota x_{j+1} , sedangkan $d_{x_{n-1} x_0}$ adalah jarak dari kota terakhir kembali ke kota awal.



Gambar 2.5 Ilustrasi TSP

Sebagai contoh, Gambar 2.5 menunjukkan representasi graf berbobot untuk TSP dengan lima simpul atau kota, yaitu A, B, C, D, dan E. Setiap sisi memiliki

bobot yang menunjukkan jarak antar kota. Contoh rute yang dihasilkan dari penyelesaian TSP di atas adalah rute (A, E, C, B, D, A) . Permasalahan ini menjadi penting karena efisiensi rute berpengaruh langsung terhadap penghematan biaya dan waktu operasional (Lazuardy dkk., 2018).

2.1.3 Algoritma *Artificial Bee Colony*

Algoritma *Artificial Bee Colony* (ABC) adalah algoritma metaheuristik berbasis populasi yang dikembangkan oleh Karaboga pada tahun 2005 berdasarkan perilaku pencarian makanan oleh koloni lebah madu untuk menyelesaikan masalah optimasi numerik. Pada algoritma ini, solusi potensial atas permasalahan dipandang sebagai posisi sumber makanan dan kualitas dari sumber makanan digambarkan oleh nilai fungsi *fitness* dari suatu solusi tersebut. Koloni lebah dalam algoritma ini terdiri dari tiga kelompok yaitu *employed bee*, *onlooker bee*, dan *scout bee*. Setengah dari koloni tersebut terdiri dari *employed bee* dan setengahnya lagi terdiri dari *onlooker bee*. Banyaknya *employed bee* sama dengan banyaknya sumber makanan. *Employed bee* yang sumber makanannya telah ditinggalkan akan menjadi *scout bee* (Karaboga & Basturk, 2008).

Proses pencarian solusi terdiri dari empat fase algoritma pada setiap iterasi. Fase pertama yaitu inisialisasi solusi awal secara acak. Lalu pada fase *employed bee*, setiap lebah berupaya memperbarui solusi yang diwakilinya dengan membuat solusi baru berdasarkan aturan tertentu. Lalu pada fase *onlooker bee*, lebah memilih solusi berdasarkan nilai *fitness* dan berupaya memperbaikinya. Setiap solusi dilengkapi dengan nilai *trial* yang bernilai awal nol, yang akan bertambah ketika solusi tersebut tidak mengalami perbaikan baik pada fase *employed bee* maupun *onlooker bee*. Jika nilai *trial* tersebut melebihi batas tertentu yang disebut *limit*,

maka pada fase *scout bee* solusi tersebut dianggap stagnan dan diganti dengan solusi baru yang dibangkitkan secara acak, kemudian nilai *trial*-nya disetel kembali menjadi nol (Karaboga, 2005).

Untuk memahami algoritma ABC secara komprehensif, pembahasan pada sub bab ini disusun ke dalam beberapa bagian, meliputi algoritma ABC untuk penyelesaian *Traveling Salesman Problem* (TSP), parameter-parameter penting yang memengaruhi kinerjanya, peran *employed bee* dan *onlooker bee* dalam mengatur dinamika eksplorasi dan eksploitasi, serta evaluasi performa algoritma ABC.

1. *Artificial Bee Colony* untuk Penyelesaian *Traveling Salesman Problem*

Algoritma ABC pada dasarnya digunakan untuk menyelesaikan masalah optimasi numerik. Oleh karena itu, untuk menerapkan algoritma ABC terhadap masalah diskrit, salah satunya yaitu *Traveling Salesman Problem* (TSP), maka diperkenalkanlah konsep metode *swap operator*, *swap sequence*, *insert operator*, *insert sequence* yang merupakan *neighborhood operator* (Amri dkk., 2012). Konsep tersebut digunakan oleh *employed bee* dan *onlooker bee* untuk mencoba meningkatkan solusinya masing-masing. Dalam algoritma ABC, setiap sumber makanan mewakili solusi potensial dari masalah, sehingga dalam kasus TSP urutan simpul diambil sebagai solusi. Berikut adalah tahapan algoritma ABC yang sudah diadaptasikan untuk menyelesaikan TSP.

a. Fase Inisialisasi

Fase inisialisasi dilakukan untuk memperoleh solusi awal yang dibangkitkan secara acak sesuai dengan parameter yang telah ditentukan dalam TSP. Jadi, suatu solusi X_i dari himpunan solusi awal dibuat secara acak di ruang

solusi dengan:

$$X_i = (x_{i0}, x_{i1}, \dots, x_{in-1}), i = 1, 2, \dots, f_s \quad (2.2)$$

di mana, $x_{ij} \in V, j = 0, 1, 2, \dots, n - 1$, dan $V = \{0, 1, 2, \dots, n - 1\}$ adalah himpunan simpul dari graf G , $x_{ij} \neq x_{ik}$ untuk $j \neq k$, dan f_s adalah banyaknya solusi dalam himpunan solusi awal. Karena solusi direpresentasikan dalam bentuk permutasi, maka *neighborhood operator* bekerja secara langsung pada posisi elemen dalam permutasi tersebut. Setelah operator diterapkan, solusi baru tetap merupakan permutasi dan secara otomatis menjadi rute TSP dengan menghubungkan simpul terakhir x_{n-1} kembali ke simpul awal x_0 pada proses evaluasi *fitness*. Setelah solusi awal diperoleh, selanjutnya adalah menghitung nilai *fitness* dari solusi ke- i menggunakan rumus berikut:

$$fit_i = \frac{1}{1 + f(X_i)} \quad (2.3)$$

di mana, $f(X_i)$ adalah panjang total rute dari solusi ke- i . Pada fase ini setiap solusi memiliki nilai *trial* (penghitung banyaknya percobaan ketika suatu solusi gagal mengalami perbaikan) yaitu nol.

b. Fase *Employed Bee*

Fase *employed bee* bertujuan untuk memperbarui solusi awal yang dihasilkan pada fase inisialisasi. Dalam fase ini, *employed bee* mencari solusi baru dengan mengeksplorasi solusi tetangga dari solusi awal yang dihasilkan pada fase inisialisasi menggunakan *neighborhood operator* yang berupa *swap operator* dan *swap sequence* (Amri dkk, 2012). Penggunaan *swap operator* pada fase ini didasarkan pada kemampuannya menghasilkan perubahan struktur rute yang lebih beragam melalui pertukaran dua simpul sehingga dapat

memperluas ruang pencarian solusi yang mendukung proses eksplorasi.

Proses *swap operator* dan *swap sequence* dijabarkan sebagai berikut:

i. *Swap Operator*

Diberikan sebuah solusi awal TSP yang direpresentasikan sebagai suatu permutasi simpul $X_i = (x_{i0}, x_{i1}, \dots, x_{in-1})$ di mana $x_{ij} \in V$, dengan $V = \{0, 1, 2, \dots, n-1\}$ adalah himpunan simpul dan $x_{ij} \neq x_{ik}$ untuk $j \neq k$. *Swap Operator*, $SO(j, k)$ di X_i , didefinisikan sebagai pertukaran posisi simpul x_{ij} dan x_{ik} pada urutan solusi X_i .

Definisi 2.1.3.1.

Misal solusi awal TSP $X_i = (x_{i0}, x_{i1}, \dots, x_{ij}, \dots, x_{ik}, \dots, x_{in-1})$. $X_i \diamond SO(j, k)$ didefinisikan sebagai berikut:

$$X_i \diamond SO(j, k) := (x_{i0}, x_{i1}, \dots, x_{ik}, \dots, x_{ij}, \dots, x_{in-1}).$$

Dengan kata lain operator $SO(j, k)$ menukar posisi x_{ij} dan x_{ik} di X_i .

Hasil penerapan operator ini menghasilkan solusi baru:

$$Y_i = X_i \diamond SO(j, k) \quad (2.4)$$

Contoh:

Misalkan $X_1 = (7, 8, 2, 5, 3)$ dan diberikan *swap operator* $SO(0, 3)$.

Maka hasil proses *swap operator* adalah sebagai berikut:

$$\begin{aligned} Y_1 &= X_1 \diamond SO(0, 3) \\ &= (7, 8, 2, 5, 3) \diamond SO(0, 3) \\ &= (5, 8, 2, 7, 3) \end{aligned}$$



Gambar 2.6 *Swap Operator* **$SO(0, 3)$**

Pada Gambar 2.6, terlihat bahwa operator $SO(0,3)$ bekerja dengan cara menukar elemen indeks nol dan indeks ketiga dari X_1 . Setelah kandidat solusi baru Y_i dihasilkan dari *swap operator*, dilakukan *greedy selection* pertama di mana nilai *fitness* dari Y_i dibandingkan dengan solusi awal X_i . Solusi yang memiliki nilai *fitness* lebih besar akan terpilih menjadi basis solusi untuk tahap eksplorasi selanjutnya. *Greedy selection* yang pertama ini digunakan untuk memastikan bahwa solusi terbaik sudah didapatkan di sekitar solusi tetangga, sebelum solusi tersebut diperbaiki lebih jauh menggunakan operator *sequence* agar solusi tidak terjebak pada optimal lokal.

ii. *Swap Sequence*

Swap sequence (SS) adalah kumpulan dari beberapa *swap operator* yang diterapkan secara berurutan sebanyak *number of sequence* (Nse). Misalkan $SS = (SO_1, SO_2, \dots, SO_N)$ di mana $SO_1, SO_2, \dots, SO_N$ adalah *swap operator* sebanyak N kali. Proses untuk memperoleh solusi baru dengan menggunakan metode SS dijabarkan sebagai berikut.

$$\begin{aligned}
 Y_i &= X_i \diamond SS \\
 &= X_i \diamond (SO_1, SO_2, \dots, SO_N) \\
 &= (\dots((X_i \diamond SO_1) \diamond SO_2) \dots \diamond SO_N)
 \end{aligned} \tag{2.5}$$

di mana Y_i adalah solusi baru dari hasil *swap sequence*, X_i adalah solusi awal.

Setelah solusi baru Y_i dibangkitkan, proses *greedy selection* kedua dilakukan dengan membandingkan nilai *fitness* solusi baru Y_i dengan nilai *fitness* solusi lama X_i . Jika nilai *fitness* Y_i lebih baik atau lebih besar

dari pada X_i , maka X_i akan digantikan oleh Y_i dan nilai *trial*-nya diatur kembali menjadi nol. Namun, jika solusi Y_i tidak lebih baik, maka X_i tetap dipertahankan dan nilai *trial*-nya ditambah satu.

c. Fase *Onlooker bee*

Fase *onlooker bee* bertujuan untuk menyeleksi dan mengeksploitasi solusi yang dihasilkan pada fase *employed bee*. Setiap *onlooker bee* mengingat solusi dari salah satu *employed bee* berdasarkan nilai *fitness*-nya untuk memperbarui solusi. *Onlooker bee* memilih setiap solusi dari *employed bee* kemudian menghitung nilai probabilitasnya dengan rumus berikut:

$$p_i = \frac{fit_i}{\sum_{j=1}^{SN} fit_j} \quad (2.6)$$

di mana fit_i adalah nilai *fitness* dari solusi dari *employed bee* ke- i , $\sum_{j=1}^{SN} fit_j$ adalah jumlah semua nilai *fitness* dari setiap solusi dan p_i merupakan probabilitas solusi ke- i . Setelah menentukan nilai probabilitas, setiap *onlooker bee* melakukan proses pemilihan solusi menggunakan *roulette wheel selection*. Pada tahap ini, pertama-tama dihitung nilai probabilitas kumulatif menggunakan rumus berikut:

$$cp_i = \sum_{j=1}^i p_j \quad (2.7)$$

Di mana cp_i adalah nilai probabilitas kumulatif solusi ke- i . Selanjutnya dihasilkan sebuah bilangan acak r pada interval (0,1). Apabila nilai $r < cp_1$, maka solusi pertama akan dipilih. Jika tidak, maka solusi ke- i akan dipilih apabila memenuhi kondisi $cp_{i-1} \leq r < cp_i$.

Dalam fase ini, dilakukan eksploitasi solusi terpilih dengan menerapkan *neighborhood operator* yaitu *insert operator* dan *insert sequence* (Amri dkk,

2012). Penggunaan *insert operator* pada fase ini didasarkan pada kemampuannya menghasilkan perubahan rute yang lebih terarah melalui pemindahan satu simpul ke posisi lain sehingga mendukung proses eksploitasi dan penyempurnaan solusi di wilayah solusi yang menjanjikan. Hal ini mendukung tujuan fase *onlooker bee*, yaitu mengarahkan algoritma menuju solusi yang lebih optimal (Lazuardy dkk., 2018). Berikut adalah penjelasan dari prosesnya:

i. *Insert Operator*

Diberikan sebuah solusi TSP yang direpresentasikan sebagai suatu permutasi simpul $X_i = (x_{i0}, x_{i1}, \dots, x_{in-1})$ di mana $x_{ij} \in V$, dengan $V = \{0, 1, 2, \dots, n-1\}$ adalah himpunan simpul dan $x_{ij} \neq x_{ik}$ untuk $j \neq k$. *Insert Operator*, $IO(j, k)$ di X_i didefinisikan sebagai pemindahan simpul pada posisi x_{ij} menuju posisi x_{ik} .

Definisi 2.1.3.2.

Diberikan $X_i = (x_{i0}, x_{i1}, \dots, x_{ij}, x_{i(j+1)}, \dots, x_{ik}, x_{i(k+1)}, \dots, x_{in-1})$. Untuk kasus $j < k$, $X_i \triangleleft IO(j, k)$ didefinisikan sebagai berikut:

$$X_i \triangleleft IO(j, k) := (x_{i0}, x_{i1}, \dots, x_{i(j+1)}, x_{i(j+2)}, \dots, x_{ik}, x_{ij}, x_{i(k+1)}, \dots, x_{in-1})$$

Dengan kata lain operator $IO(j, k)$ memindahkan elemen pada posisi j ke posisi k di X_i dan menggeser elemen lain yang berada di antara posisi tersebut ke kiri. Sebaliknya jika kasus $j > k$, maka elemen di antaranya bergeser ke kanan.

Hasil penerapan operator ini menghasilkan solusi baru:

$$Y_i = X_i \triangleleft IO(j, k) \quad (2.8)$$

Contoh:

Misalkan $X_1 = (9,7,4,2,5)$ dan diberikan *insert operator* $IO(1,4)$. Maka hasil proses *insert operator* adalah sebagai berikut:

$$\begin{aligned} Y_1 &= X_1 \blacktriangleleft IO(1,4) \\ &= (9,7,4,2,5) \blacktriangleleft IO(1,4) \\ &= (9,4,2,5,7) \end{aligned}$$



Gambar 2.7 *Insert Operator* **$IO(1,4)$**

Pada Gambar 2.7, terlihat bahwa operator $IO(1,4)$ bekerja dengan cara memindahkan elemen kedua ke posisi elemen kelima, lalu memindahkan menggeser elemen ketiga, keempat, dan kelima ke sebelah kiri untuk mengisi posisi yang kosong. Setelah kandidat solusi baru Y_i dihasilkan dari *insert operator*, dilakukan *greedy selection* pertama di mana nilai *fitness* dari Y_i dibandingkan dengan solusi awal X_i . Solusi yang memiliki nilai *fitness* lebih besar akan terpilih menjadi basis solusi untuk tahap eksploitasi selanjutnya.

ii. *Insert Sequence*

Insert Sequence (IS) adalah kumpulan dari *insert operator* yang diterapkan secara berurutan sebanyak *number of sequence (Nse)*. Misalkan $IS = (IO_1, IO_2, \dots, IO_N)$ di mana $IO_1, IO_2, \dots, IO_N$ adalah *insert operator* sebanyak N kali. Proses untuk memperoleh solusi baru dengan menggunakan metode *IS* dijabarkan sebagai berikut:

$$\begin{aligned}
Y_i &= X_i \blacktriangleleft IS \\
&= X_i \blacktriangleleft (IO_1, IO_2, \dots, IO_N) \\
&= (\dots ((X_i \blacktriangleleft IO_1) \blacktriangleleft IO_2) \dots \blacktriangleleft IO_N)
\end{aligned} \tag{2.9}$$

di mana Y_i adalah solusi baru hasil dari *insert sequence* dan X_i adalah solusi awal.

Setelah solusi baru Y_i dibangkitkan, solusi tersebut dibandingkan dengan solusi lama (solusi yang dihasilkan pada fase *employed bee*) berdasarkan nilai *fitness*-nya. Jika nilai *fitness* Y_i lebih baik dari solusi lama, maka solusi lama akan digantikan oleh Y_i dan nilai *trial*-nya diatur kembali menjadi nol. Namun, jika solusi Y_i tidak lebih baik, maka solusi lama tetap dipertahankan dan nilai *trial*-nya ditambah satu.

d. Fase *Scout Bee*

Pada fase terakhir ini, nilai *trial* dari seluruh solusi dievaluasi untuk mengidentifikasi solusi yang mengalami stagnasi. Semua nilai *trial* dari setiap solusi akan dibandingkan dengan parameter batas (*limit*). Apabila *trial* suatu solusi telah melampaui nilai *limit*, maka solusi tersebut akan ditinggalkan. Dalam hal ini, *employed bee* yang bertanggung jawab terhadap solusi tersebut akan diubah menjadi *scout bee* dan selanjutnya menghasilkan solusi baru secara acak menggunakan Persamaan 2.2. Fase ini adalah mekanisme penting dalam algoritma ABC untuk mencegah konvergensi prematur dan menjaga daya jelajah algoritma.

2. Parameter Algoritma *Artificial Bee Colony*

Algoritma ABC dapat bekerja dengan bergantung pada beberapa nilai parameter yang harus ditetapkan di awal proses (Amanah dkk., 2022). Parameter-

parameter tersebut adalah sebagai berikut:

- a. *Colony size*, yaitu banyaknya keseluruhan lebah yang digunakan dalam sistem.
 - b. *Problem size*, yaitu parameter yang merujuk pada ukuran atau dimensi permasalahan yang akan diselesaikan. Misalnya, pada permasalahan TSP, *problem size* dinyatakan sebagai banyaknya kota atau simpul yang harus dikunjungi. Semakin besar *problem size*, semakin tinggi kompleksitas pencarian solusi.
 - c. *Limit*, yaitu parameter yang membatasi jumlah percobaan (*trial*) gagal yang terjadi berturut-turut pada suatu solusi. Apabila suatu solusi tidak mengalami peningkatan kualitas solusi selama melebihi nilai *limit*, maka solusi tersebut akan ditinggalkan dan akan diganti dengan solusi baru secara acak oleh *scout bee*.
 - d. Iterasi maksimum, yaitu jumlah maksimum iterasi kerja yang diizinkan untuk koloni lebah dalam proses pencarian solusi. Ketika jumlah iterasi telah mencapai nilai maksimum tersebut, maka eksekusi algoritma akan dihentikan, dan solusi terbaik yang ditemukan sampai iterasi tersebut ditetapkan sebagai output dari algoritma.
3. Peran *Employed Bee* dan *Onlooker bee* dalam Mengatur Dinamika Eksplorasi dan Eksploitasi

Eksplorasi dan eksploitasi merupakan dua faktor penting yang mempengaruhi performa algoritma *Artificial Bee Colony* (ABC) (Karaboga, 2005). Eksplorasi adalah kemampuan suatu algoritma untuk menelusuri wilayah-wilayah yang belum diketahui dalam ruang pencarian agar dapat menemukan solusi yang menjanjikan. Di samping itu, eksploitasi adalah

kemampuan suatu algoritma untuk memusatkan pencarian di sekitar wilayah solusi yang telah dipilih untuk dilakukan penyempurnaan solusi lebih lanjut (Qin dkk., 2015).

Pada algoritma ABC, terdapat tiga jenis fase lebah yaitu *employed bee*, *onlooker bee*, dan *scout bee*. Pada fase *employed bee*, algoritma melakukan eksplorasi dengan menghasilkan solusi tetangga secara acak dari solusi awal yang dihasilkan pada fase inisialisasi untuk memperbarui solusi. Sementara itu, pada fase *onlooker bee*, algoritma melakukan eksploitasi yang ditandai dengan proses pemilihan solusi berdasarkan nilai *fitness*-nya, sehingga solusi yang lebih baik memiliki probabilitas yang lebih besar untuk dipilih dan diperbarui lebih lanjut. Secara standar, algoritma ABC mengasumsikan setengah dari jumlah koloni adalah *employed bee* dan setengahnya lagi adalah *onlooker bee*. Banyaknya solusi atau sumber makanan (*SN*) adalah setengah dari ukuran populasi (*NP*), sehingga banyaknya *employed bee* (NP_e) sama dengan banyaknya sumber makanan (*SN*) dan setiap sumber makanan (*SN*) diwakili oleh satu *employed bee* (Karaboga & Basturk, 2007).

Untuk menerapkan variasi rasio banyaknya *employed bee* dan *onlooker bee* di luar rasio dasar 1:1, banyaknya sumber makanan (*SN*) tetap mengikuti definisi standar algoritma ABC yaitu $SN = \frac{1}{2} \times NP$, sedangkan banyaknya *employed bee* (NP_e) dan *onlooker bee* (NP_o) ditentukan melalui rasio $r_c = NP_e / NP_o$ melalui perhitungan $NP_e = \text{round}(r_c \times NP)$ dan $NP_o = NP - NP_e$. Variasi rasio tersebut berpengaruh langsung terhadap mekanisme penempatan *employed bee* terhadap sumber makanan. Jika $NP_e > SN$, maka sebanyak *SN* dari *employed bee* pertama ditempatkan satu per satu pada setiap sumber

makanan (SN) dan *employed bee* yang tersisa ditempatkan secara acak pada sumber makanan tersebut. Sebaliknya jika $NP_e < SN$, maka setiap *employed bee* memilih sumber makanan secara acak.

Dalam algoritma *Artificial Bee Colony* (ABC), pembagian banyaknya *employed bee* dan *onlooker bee* memegang peranan penting dalam mengatur dinamika antara proses eksplorasi dan eksploitasi ruang pencarian solusi (Qin dkk., 2015). Proporsi antara dua jenis lebah ini menentukan kemampuan algoritma dalam menemukan solusi global tanpa terjebak pada lokal optimum. Oleh karena itu, proporsi rasio banyaknya *employed bee* dan *onlooker bee* perlu dianalisis secara kuantitatif untuk menentukan konfigurasi optimal pada domain masalah diskrit yaitu TSP.

4. Performa Algoritma *Artificial Bee Colony*

Evaluasi performa algoritma bertujuan untuk menilai efektivitas dan efisiensi dalam menemukan solusi optimal terhadap suatu permasalahan. Menurut Karaboga dan Basturk (2007), performa algoritma *Artificial Bee Colony* (ABC) dinilai berdasarkan kualitas solusi dan kecepatan konvergensi. Berikut adalah beberapa metrik evaluasi untuk menguji performa algoritma ABC pada penyelesaian TSP:

- a. Nilai *Fitness*, yaitu untuk mengukur kualitas solusi. Nilai *fitness* yang lebih besar menunjukkan jalur yang lebih optimal.
- b. Panjang Rute Terbaik, yaitu total bobot jalur minimum yang dihasilkan algoritma, semakin kecil nilainya, maka akan semakin efisien rute yang ditempuh oleh seorang sales.

- c. Kecepatan konvergensi, yaitu banyaknya iterasi pencarian solusi yang dibutuhkan algoritma hingga tercapainya solusi terbaik.

Dengan demikian, metrik-metrik tersebut menggambarkan performa algoritma ABC dalam mencapai solusi terbaik dengan proses konvergensi yang efisien.

2.1.4 Uji *Analysis of Variance*

Analysis of Variance (ANOVA) merupakan metode analisis statistik yang digunakan untuk menguji signifikansi perbedaan rata-rata dari tiga atau lebih kelompok data yang berbeda. Prinsip dasar pengujian ini adalah membandingkan variasi yang terjadi di antara kelompok perlakuan dengan variasi yang ada di dalam kelompok itu sendiri. Dalam konteks penelitian ini, uji ANOVA diterapkan untuk membuktikan secara objektif apakah terdapat perbedaan pada variabel bebas benar-benar menghasilkan pengaruh yang berbeda secara nyata terhadap variabel terikat (Al Fatih dkk., 2026).

Berikut adalah langkah-langkah untuk menggunakan uji ANOVA:

1. Penentuan Hipotesis dan Variabel

Langkah pertama yaitu menentukan hipotesis dan variabel-variabel yang akan diuji serta memilih model ANOVA berdasarkan jumlah variabel bebas dan variabel terikatnya. Dalam Penelitian ini dipilih model *One-Way* ANOVA atau ANOVA Satu Arah karena terdapat satu variabel bebas kategorik yaitu rasio dan terdapat satu variabel terikat yaitu panjang rute terpendek.

2. Uji Asumsi

Langkah ini berfungsi sebagai penentu kelayakan penggunaan uji ANOVA.

Berikut adalah uji asumsi-asumsi yang harus terpenuhi:

a. Uji Asumsi Normalitas

Uji asumsi ini penting, karena uji signifikansi ANOVA didasarkan pada distribusi F yang valid hanya jika data variabel terikat berdistribusi normal. Kriteria uji Normalitas adalah jika nilai *Sig. (P Value)* $< 0,05$ maka bisa disimpulkan bahwa data tidak berdistribusi secara normal. Di samping itu, jika nilai *Sig. (P Value)* $> 0,05$ maka bisa disimpulkan bahwa data berdistribusi normal.

b. Uji Asumsi Homogenitas

Uji asumsi Homogenitas dilakukan untuk mengetahui apakah varians data pada setiap kelompok yang dibandingkan memiliki tingkat keragaman yang sama. Kriteria uji asumsi Homogenitas ini adalah jika nilai *Sig. (P Value)* $< 0,05$ maka bisa disimpulkan bahwa data tidak homogen. Di samping itu, jika nilai *Sig. (P Value)* $> 0,05$ maka bisa disimpulkan bahwa data homogen.

3. Uji Signifikansi ANOVA

Pada langkah ini, dilakukan analisis utama dengan model *One Way* ANOVA yang telah ditentukan pada uji asumsi homogenitas. Kriteria uji signifikansi ANOVA adalah jika nilai *Sig. (P Value)* $< 0,05$ maka bisa disimpulkan bahwa ada pengaruh secara signifikan. Di samping itu, jika nilai *Sig. (P Value)* $> 0,05$ maka bisa disimpulkan bahwa tidak ada pengaruh secara signifikan.

4. Uji Lanjut

Uji lanjut hanya dilakukan jika hasil uji signifikansi ANOVA adalah signifikan. Uji lanjut ini dilakukan untuk menentukan variabel terikat mana yang perbedaanya signifikan. Jenis uji lanjut dalam *One Way* ANOVA ditentukan berdasarkan hasil uji asumsi Homogenitas. Apabila uji asumsi

Homogenitas terpenuhi, maka dilakukan uji lanjut *Bonferroni*. Di samping itu, apabila uji asumsi Homogenitas tidak terpenuhi, maka akan dilakukan uji lanjut *Games-Howell*.

2.2 Kajian Integrasi Topik dengan Al-Quran/Hadits

Konsep optimasi tercermin dalam prinsip keteraturan dan efisiensi ciptaan Allah SWT sebagaimana Allah berfirman dalam Al-Qur'an Surat Al-Anbiya ayat 16 yang artinya:

“Kami tidak menciptakan langit dan bumi dan apa yang ada di antara keduanya dengan main-main.” (QS. Al-Anbiya [21]: 16).

Menurut Quraish Shihab (2002) dalam *Tafsir Al-Mishbah*, ayat di atas menegaskan bahwa Allah SWT tidak menciptakan langit, bumi, dan seluruh isinya dengan main-main, melainkan dengan tujuan dan keteraturan yang pasti. Ayat ini menggambarkan bahwa setiap ciptaan Allah memiliki makna, keseimbangan, dan efisiensi dalam fungsinya, tanpa ada unsur kesia-siaan. Nilai tersebut sejalan dengan prinsip dasar dalam ilmu optimasi, di mana setiap proses diarahkan untuk memperoleh hasil paling efisien dan bermakna melalui langkah-langkah yang terukur dan sistematis. Dalam konteks penelitian ini, konsep optimasi pada algoritma *Artificial Bee Colony* (ABC) mencerminkan upaya manusia dalam meneladani *sunnatullah*, yakni pola keteraturan dan keseimbangan yang melekat pada ciptaan Allah, yang menjadi dasar bagi seluruh bentuk efisiensi dan keoptimalan dalam sistem ilmiah maupun alamiah.

Fenomena perilaku lebah madu yang menjadi dasar dari pengembangan *Artificial Bee Colony* (ABC) memiliki keterkaitan kuat dengan Firman Allah SWT dalam Surat An-Nahl ayat 69 yang artinya:

“Kemudian makanlah dari tiap-tiap (macam) buah-buahan dan tempuhlah jalan Tuhanmu yang telah dimudahkan (bagimu). Dari perut lebah itu keluar minuman (madu) yang bermacam-macam warnanya, di dalamnya terdapat obat yang menyembuhkan bagi manusia. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda (kebesaran Allah) bagi orang-orang yang berpikir.” (QS. An-Nahl [16]: 68–69)

Ayat ini menjelaskan bahwa perilaku lebah bukan hanya hasil insting biologis, tetapi merupakan hasil dari wahyu dan bimbingan langsung dari Allah SWT, yang mengatur setiap aspek kehidupannya secara sistematis dan efisien. Dalam konteks sains modern, pola ini dapat dipahami sebagai cerminan prinsip optimasi alami, yaitu kemampuan makhluk hidup dalam menemukan solusi paling efektif untuk bertahan dan memproduksi (Shihab, 2002).

Menurut Tafsir Al-Misbah (Quraish Shihab, 2002), frasa *“tempuhlah jalan Tuhanmu yang telah dimudahkan bagimu”* menggambarkan bagaimana lebah mengikuti jalur tertentu yang telah ditetapkan secara teratur dan efisien untuk mencapai sumber makanan dan kembali ke sarangnya tanpa tersesat. Hal ini menunjukkan bahwa dalam pencarian sumber daya, lebah selalu menempuh rute optimal yang meminimalkan energi dan waktu. Prinsip yang sama diadopsi dalam algoritma *Artificial Bee Colony* untuk mencari rute terpendek dalam ruang pencarian solusi.

Lebah juga diajarkan untuk “makan dari setiap buah-buahan” dan menghasilkan “minuman (madu) yang bermacam-macam warnanya” yang menandakan proses diversifikasi sumber dan hasil, yang paralel dengan proses eksplorasi solusi dalam algoritma ABC. Seperti lebah yang mencoba berbagai bunga untuk menemukan nektar terbaik, algoritma ABC juga menjelajahi berbagai kandidat solusi hingga menemukan *fitness* maksimum, yang dalam konteks penelitian ini diinterpretasikan sebagai rute terpendek dengan efisiensi tertinggi.

2.3 Kajian Topik Dengan Teori Pendukung

Penelitian ini secara khusus mengkaji bagaimana rasio banyaknya *employed bee* dan *onlooker bee* memengaruhi performa algoritma *Artificial Bee Colony* (ABC) dalam menyelesaikan masalah TSP. Dalam konteks permasalahan optimasi rute, setiap jalur dapat direpresentasikan sebagai kandidat solusi. Sementara proses pencarian rute terpendek merepresentasikan perilaku koloni lebah dalam mencari sumber makanan dengan jarak dan kualitas terbaik melalui dua mekanisme utama yaitu eksplorasi dan eksploitasi.

Proporsi dua mekanisme tersebut dikendalikan melalui rasio banyaknya *employed bee* dan *onlooker bee*. Secara teoritis, *employed bee* berperan dalam eksplorasi, yaitu memperluas ruang pencarian dengan membentuk variasi jalur baru berdasarkan solusi yang ada, sementara *onlooker bee* bertugas untuk eksploitasi, yaitu memperdalam pencarian di sekitar solusi dengan nilai *fitness* tertinggi. Jika proporsi *employed bee* terlalu besar, proses pencarian cenderung menghasilkan banyak variasi solusi tanpa peningkatan kualitas yang signifikan. Sebaliknya, jika *onlooker bee* terlalu dominan, algoritma cenderung terfokus pada solusi tertentu sehingga meningkatkan risiko konvergensi prematur pada optimum lokal. Oleh karena itu, variasi rasio keduanya menjadi titik kritis dalam menentukan stabilitas hasil dan kemampuan algoritma menemukan solusi global optimum.

Kajian teori sebelumnya menunjukkan bahwa performa algoritma *Artificial Bee Colony* (ABC) sangat dipengaruhi oleh mekanisme pengaturan parameter yang mengatur dinamika eksplorasi dan eksploitasi. Qin dkk. (2015) menyatakan bahwa tanpa pengaturan parameter tersebut dengan tepat, dinamika pencarian solusi oleh algoritma ABC menjadi tidak stabil dan rentan terjebak pada solusi lokal.

Sementara itu, Zhu & Kwong (2010) menguatkan hal ini dengan menunjukkan bahwa algoritma ABC memiliki kecenderungan eksploitasi yang lemah, yang menyebabkan konvergensi prematur dan menetap pada solusi lokal. Hal ini menunjukkan bahwa proporsi *employed bee* dan *onlooker bee* memiliki peran penting dalam mengendalikan mengatur dinamika eksplorasi dan eksploitasi.

Dengan demikian, penelitian ini difokuskan untuk mengetahui pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap kualitas solusi yang dihasilkan algoritma ABC dalam penyelesaian TSP, khususnya panjang rute terpendek. Penelitian ini juga diarahkan untuk mengidentifikasi rasio yang memberikan performa paling optimal yang mampu mengatur dinamika eksplorasi dan eksploitasi secara efektif. Penelitian ini diharapkan menghasilkan bukti empiris yang dapat memperkuat dasar pengaturan rasio lebah dalam ABC dan menunjang pengembangan strategi parameter yang lebih stabil dan adaptif.

BAB III

METODE PENELITIAN

3.1 Jenis Penelitian

Penelitian ini merupakan penelitian kuantitatif eksperimental berbasis simulasi algoritmik, yang bertujuan untuk melihat pengaruh rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma *Artificial Bee Colony* (ABC) dalam menyelesaikan *Traveling Salesman Problem* (TSP). Pendekatan kuantitatif digunakan karena seluruh hasil penelitian berupa data numerik yaitu panjang rute dan iterasi konvergensi (Andriani dkk., 2020). Sementara itu, pendekatan eksperimental dilakukan dengan cara memanipulasi parameter rasio dan mengamati dampaknya terhadap performa algoritma.

3.2 Data dan Sumber Data

Data pada penelitian ini berupa graf komplit berbobot yang dibangkitkan menggunakan bahasa pemrograman *Python* dengan bantuan pustaka *NetworkX*. Graf tersebut digunakan sebagai representasi dari permasalahan TSP, di mana setiap simpul menyatakan kota dan setiap sisi menyatakan jarak antar kota. Data hasil penelitian ini diperoleh dari implementasi algoritma ABC yang telah dimodifikasi melalui penerapan variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap tiga graf tersebut menghasilkan data hasil eksperimen yaitu panjang rute terpendek.

3.3 Tahapan Penelitian

Penelitian ini dilakukan melalui beberapa tahapan yaitu sebagai berikut:

1. Persiapan Data

Pada tahap ini dilakukan penentuan data yang akan diolah, data berupa graf komplit tidak berarah dengan bobot yang ditentukan secara acak sebagai representasi TSP. Data tersebut didapatkan dengan menggunakan bantuan pustaka *NetworkX* pada Python. Data yang akan digunakan sebanyak tiga graf komplit acak untuk mendapatkan hasil yang konsisten dengan banyaknya simpul masing-masing adalah 50, 100, dan 200.

2. Inisialisasi Parameter

Pada tahap ini ditetapkan nilai konstan untuk parameter dasar algoritma ABC yaitu *colony size*, *limit*, dan iterasi maksimum dengan nilai sebagai berikut:

- a. *Colony size* atau *Number of Population (NP)*. Semakin banyak jumlah *NP*, maka rute yang akan dibentuk juga semakin banyak dan probabilitas didapatkan solusi terbaik juga semakin tinggi, maka dari pada penelitian ini dipilih nilai *NP* sebanyak 80.
- b. *Limit*. Nilai ini mengatur batas percobaan oleh suatu solusi untuk diperbaiki. Oleh karena itu pada penelitian ini dipilih nilai *limit* sebesar 20.
- c. Iterasi maksimum. Semakin banyak iterasi, semakin mudah untuk menentukan tingkat konvergensi. Oleh karena itu, pada penelitian ini dipilih nilai iterasi maksimum sebesar 50.

3. Penetapan Lima Variasi Rasio Banyaknya *Employed bee* dan *Onlooker bee*.

Pada tahap ini, ditentukan lima variasi rasio antara banyaknya *employed bee* dan *onlooker bee* ($NP_e : NP_o$), yaitu 1:9, 3:7, 1:1, 7:3, dan 9:1, sebagai kelompok dari variabel bebas penelitian untuk mengamati pengaruhnya pada performa algoritma ABC.

4. Eksperimen dan Implementasi Algoritma *Artificial Bee Colony* untuk Penyelesaian TSP.

Proses eksperimen dilakukan pada tiga graf komplit yang sudah dibuat. Pada setiap graf, algoritma ABC dijalankan untuk setiap rasio dengan solusi awal yang dihasilkan pada fase inisialisasi adalah sama, dan setiap rasio dijalankan sebanyak dua puluh kali untuk menjamin validitas statistik. Algoritma ABC diimplementasikan melalui proses yang iteratif sebagai berikut:

- a. Fase Inisialisasi, yaitu solusi awal dibangkitkan secara acak. Setiap solusi adalah rute perjalanan dengan nilai *trial* nol. Selanjutnya nilai *fitness* dihitung untuk setiap solusi awal yang terbentuk.
- b. Fase *Employed bee*, yaitu setiap *employed bee* mencari solusi tetangga baru dari solusi awal yang ia pilih secara acak menggunakan *neighborhood operator* berupa *swap operator* dan *swap sequence*. Selanjutnya dilakukan mekanisme *greedy selection* yaitu jika solusi kandidat lebih baik, maka solusi lama diganti dan *trial* direset menjadi nol jika tidak, solusi lama dipertahankan dan *trial* ditambah satu.
- c. Fase *Onlooker bee*, yaitu *onlooker bee* memilih solusi yang dihasilkan oleh *employed bee* berdasarkan probabilitas yang proporsional dengan *fitness* solusi. Solusi terpilih diperbaiki dengan *neighborhood operator* berupa *insert operator* dan *insert sequence* untuk mencari solusi kandidat yang lebih baik, diikuti mekanisme *greedy selection*.
- d. Fase *Scout Bee*, yaitu solusi yang telah melampaui batas *limit* pada *trial* akan ditinggalkan, dan *employed bee* yang bertanggung jawab diubah menjadi *scout bee* yang akan menghasilkan solusi baru secara acak untuk

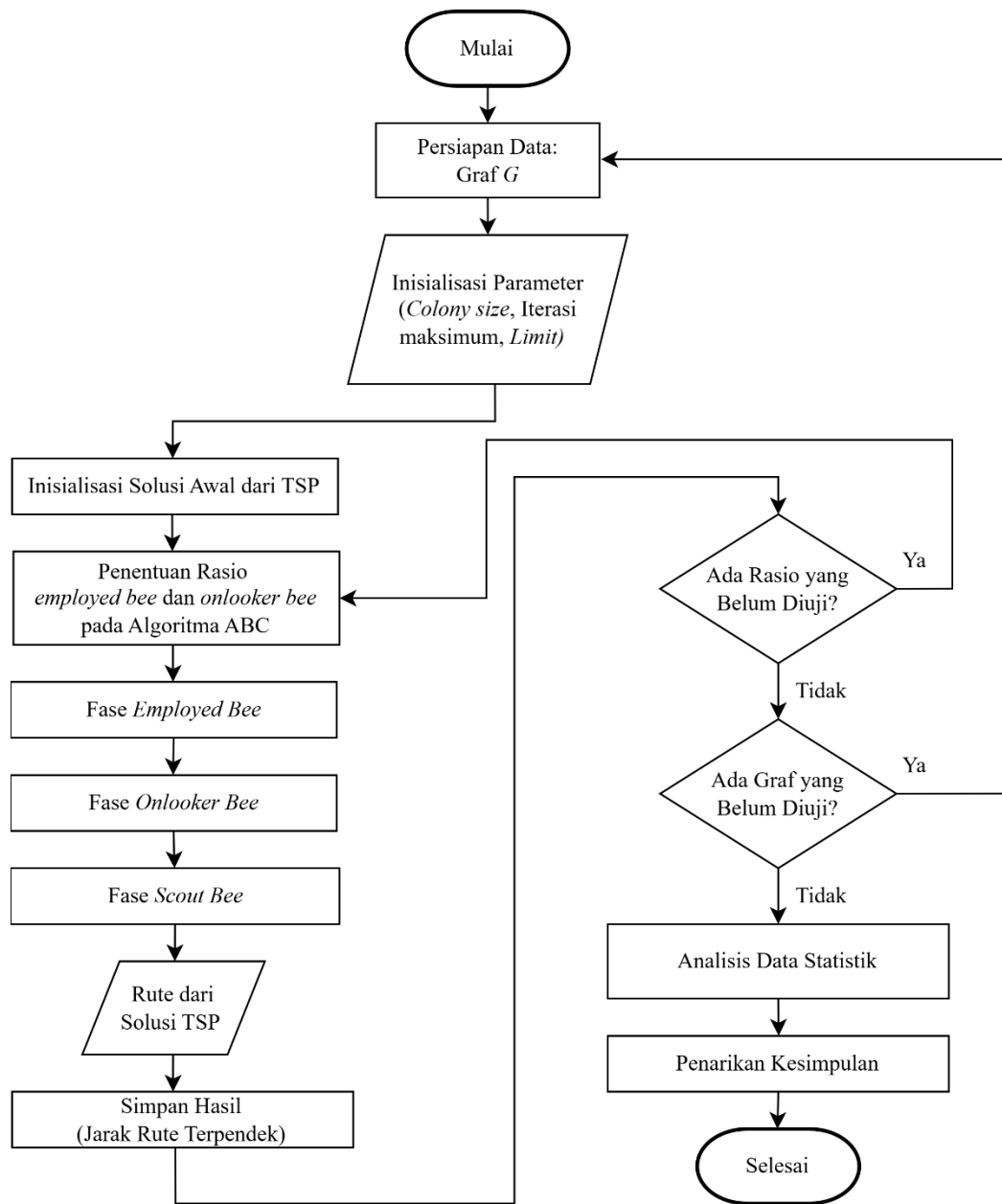
mempertahankan keragaman populasi. Seluruh iterasi diulang dengan proses yang sama hingga iterasi maksimum tercapai.

Data hasil eksperimen yang dikumpulkan adalah panjang rute terpendek pada setiap rasio dari beberapa graf yang diuji.

5. Analisis Data Statistik dan Penarikan Kesimpulan

Data yang diperoleh dari hasil eksperimen dianalisis secara statistik untuk menguji pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma. Tahap awal dilakukan uji normalitas untuk memastikan data berdistribusi normal. Selanjutnya, dilakukan uji homogenitas untuk menguji kesamaan varians antar kelompok data. Apabila kedua asumsi terpenuhi, maka dilakukan uji ANOVA satu arah untuk mengetahui apakah terdapat perbedaan yang signifikan antar variasi rasio. Jika hasil uji ANOVA menunjukkan perbedaan yang signifikan, maka dilanjutkan dengan uji lanjut (*post-hoc*) untuk mengidentifikasi pasangan rasio yang memiliki perbedaan signifikan. Berdasarkan hasil analisis tersebut, kemudian ditarik kesimpulan mengenai pengaruh variasi rasio terhadap performa algoritma. Rasio yang direkomendasikan sebagai rasio optimal adalah rasio yang menghasilkan panjang rute terpendek.

Seluruh tahapan penelitian di atas disusun secara sistematis dan divisualisasikan melalui sebuah *flowchart*. *Flowchart* tersebut memberikan gambaran secara umum mengenai proses penelitian yang dilakukan mulai dari persiapan data sampai penarikan kesimpulan. *Flowchart* penelitian dapat dilihat pada Gambar 3.1.

Gambar 3.1 *Flowchart* Penelitian

BAB IV

HASIL DAN PEMBAHASAN

4.1 Persiapan Data

Penelitian ini menggunakan data berupa graf komplit berbobot sebagai representasi dari *Traveling Salesman Problem* yang dihasilkan menggunakan bahasa pemrograman *Python* dengan pustaka *NetworkX*. Untuk mendapatkan hasil yang konsisten pada penelitian ini, maka dibangun sebanyak tiga graf dengan banyaknya simpul masing-masing adalah 50, 100, dan 200. Graf tersebut dibangkitkan dengan bobot sisi secara acak dalam rentang $[1, 500]$.

Mekanisme penelitian kali ini didesain untuk membuat dan menggunakan satu graf sekaligus untuk menjalankan eksperimen penelitian. Graf akan diberikan suatu parameter yaitu *seed* agar data graf akan tetap sama selama dijalankan, hal ini memungkinkan data graf dapat digunakan kembali dengan mudah tanpa kehilangan struktur awalnya. Dengan pendekatan ini, proses selanjutnya dapat langsung memanfaatkan graf yang telah dibuat tanpa perlu membangunnya dari awal.

4.2 Inisialisasi Parameter

Tahap inisialisasi parameter bertujuan untuk menetapkan nilai-nilai parameter dasar algoritma *Artificial Bee Colony* (ABC) yang digunakan secara konstan pada seluruh proses eksperimen. Penetapan parameter ini dilakukan untuk memastikan bahwa perbedaan performa algoritma yang diamati benar-benar disebabkan oleh variasi rasio banyaknya *employed bee* dan *onlooker bee*, bukan oleh perubahan parameter lain di luar fokus penelitian.

Berdasarkan rancangan eksperimen yang telah dijelaskan pada Bab III Metode Penelitian, parameter-parameter yang diinisialisasi pada penelitian ini adalah sebagai berikut:

1. Ukuran masalah (*problem size*) ditetapkan berdasarkan banyaknya simpul pada graf komplit yang merepresentasikan jumlah kota pada *Traveling Salesman Problem* (TSP). Terdapat tiga graf komplit yang diuji pada penelitian ini dengan banyak simpul yang beragam yaitu graf komplit dengan 50 simpul, 100 simpul, dan 200 simpul untuk meningkatkan kompleksitasnya.
2. Ukuran populasi atau *colony size* (*NP*) ditetapkan sebesar 80. Nilai ini menyatakan total jumlah lebah dalam satu koloni algoritma ABC, yang terdiri atas *employed bee* dan *onlooker bee*. Menurut Lazuardy dkk. (2018), pemilihan nilai $NP = 80$ dilakukan untuk memberikan keseimbangan antara keragaman solusi dan efisiensi komputasi. Ukuran populasi yang terlalu kecil berpotensi menyebabkan algoritma kehilangan keberagaman solusi, sedangkan ukuran populasi yang terlalu besar dapat meningkatkan waktu komputasi secara signifikan.
3. Banyaknya sumber makanan (*food sources* atau *SN*) ditentukan mengikuti definisi standar algoritma ABC, yaitu setengah dari ukuran populasi, sehingga diperoleh $SN = \frac{NP}{2} = 40$. Setiap sumber makanan merepresentasikan satu solusi kandidat TSP berupa satu rute yang mengunjungi seluruh simpul tepat satu kali.
4. Batas percobaan (*limit*) ditetapkan sebesar 20. Parameter ini menyatakan jumlah maksimum percobaan berturut-turut suatu solusi yang tidak

mengalami perbaikan kualitas. Apabila nilai *trial* suatu solusi telah mencapai nilai limit, maka solusi tersebut dianggap stagnan dan akan digantikan oleh solusi baru yang dibangkitkan secara acak melalui fase *scout bee*. Penetapan nilai limit sebesar 20 bertujuan untuk mencegah solusi terjebak pada optimum lokal.

5. Banyaknya iterasi (*cycles*) ditetapkan sebesar 50. Parameter ini menyatakan batas maksimum siklus kerja algoritma ABC dalam satu kali eksekusi. Nilai ini dipilih dengan mempertimbangkan kebutuhan untuk mengamati proses konvergensi algoritma tanpa menyebabkan waktu komputasi yang berlebihan. Kriteria berhenti yang dicatat dalam penelitian ini adalah iterasi ketika solusi terbaik pertama kali ditemukan dan pada saat ini sistem mulai menunjukkan perilaku stagnasi, di mana koloni lebah tidak lagi menemukan solusi yang lebih baik.
6. Banyaknya pengulangan eksperimen (*run*) ditetapkan sebanyak 20 kali untuk setiap variasi rasio. Pengulangan ini dilakukan untuk mengakomodasi sifat stokastik algoritma ABC, sehingga data hasil eksperimen yang diperoleh lebih stabil dan layak dianalisis secara statistik.

Dengan ditetapkannya seluruh parameter di atas secara konstan, penelitian ini memastikan bahwa variasi performa algoritma ABC yang diamati pada bab-bab selanjutnya semata-mata dipengaruhi oleh perbedaan rasio banyaknya *employed bee* dan *onlooker bee*. Pendekatan ini memperkuat validitas analisis dan kesimpulan yang dihasilkan dalam penelitian ini.

4.3 Menetapkan Lima Variasi Rasio *Employed Bee* dan *Onlooker Bee*

Penetapan rasio ini menjadi fokus penelitian karena pembagian peran antara *employed bee* dan *onlooker bee* secara langsung memengaruhi performa algoritma ABC dalam mencari solusi terbaik. Pada algoritma ABC standar, rasio banyaknya *employed bee* dan *onlooker bee* ditetapkan sama besar yaitu 1:1. Namun, pada penelitian ini pengaturan rasio tersebut dimodifikasi untuk mengamati pengaruh proporsi *employed bee* dan *onlooker bee* terhadap dinamika eksplorasi dan eksploitasi dalam proses pencarian solusi. Rasio dalam penelitian ini dinyatakan sebagai rasio banyaknya *employed bee* terhadap populasi koloni yang dinotasikan sebagai parameter rasio r_c untuk memudahkan proses perhitungan. Berdasarkan definisi tersebut, variasi rasio 1:9, 3:7, 1:1, 7:3, dan 9:1 masing-masing direpresentasikan sebagai $r_c = 0,1, 0,3, 0,5, 0,7$, dan $0,9$.

Dengan ukuran populasi sebanyak 80, banyaknya *employed bee* (NP_e) dan *onlooker bee* (NP_o) untuk setiap rasio ditentukan menggunakan rumus berikut:

$$NP_e = \lfloor (r_c \times NP) \rfloor, NP_o = NP - NP_e.$$

Sehingga diperoleh konfigurasi banyaknya lebah untuk setiap variasi rasio sebagai berikut:

- a. $r_c = 0,1$, artinya terdapat 8 *employed bee* dan 72 *onlooker bee*
- b. $r_c = 0,3$, artinya terdapat 24 *employed bee* dan 56 *onlooker bee*
- c. $r_c = 0,5$, artinya terdapat 40 *employed bee* dan 40 *onlooker bee*
- d. $r_c = 0,7$, artinya terdapat 56 *employed bee* dan 24 *onlooker bee*
- e. $r_c = 0,9$, artinya terdapat 72 *employed bee* dan 8 *onlooker bee*

Dengan demikian, pada subbab selanjutnya dari lima rasio ini diharapkan dapat dievaluasi tentang pengaruhnya terhadap performa algoritma ABC terkait dinamika

eksplorasi dan eksploitasi serta mampu mengidentifikasi rasio yang memberikan kualitas solusi terbaik dalam penyelesaian TSP.

4.4 Implementasi Algoritma dan Eksperimen

Subbab ini menjelaskan secara sistematis bagaimana proses kerja algoritma *Artificial Bee Colony* (ABC). Implementasi algoritma dalam penelitian ini menggunakan parameter-parameter yang telah ditetapkan dan tiga graf komplit dengan bobot sisi acak untuk menjaga konsistensi eksperimen yaitu graf dengan 50 simpul, 100 simpul, dan 200 simpul dengan label simpul $0, 1, 2, 3, \dots, n$ di mana n adalah banyaknya simpul. Namun, untuk memperjelas mengenai mekanisme kerja algoritma ABC, pada subbab ini akan ditunjukkan contoh ilustratif satu iterasi pertama algoritma ABC menggunakan graf 10 simpul, nilai limit sebesar 1, banyaknya lebah (NP) sebesar 50, dan rasio yang dipilih untuk dijadikan contoh adalah $r_c = 0,5$ atau rasio 1: 1.

Dalam pencarian rute terpendek pada *Traveling Salesman Problem* (TSP), prosesnya dilakukan melalui beberapa tahapan sebagai berikut:

1. Fase Inisialisasi

Pada fase ini solusi awal dibangkitkan secara acak sebanyak $SN = \frac{NP}{2}$ yaitu 25. Setiap solusi merepresentasikan suatu rute perjalanan yang mengunjungi seluruh kota tepat satu kali dan kembali ke kota asal. Setiap solusi diberikan nilai *trial* sebesar nol karena belum mengalami proses perbaikan maupun kegagalan dalam menghasilkan solusi yang lebih baik. Selanjutnya, nilai *fitness* dihitung untuk setiap solusi awal yang terbentuk berdasarkan total jarak rute yang diperoleh. Tabel 4.1 menunjukkan solusi-solusi awal yang dihasilkan beserta hasil perhitungan nilai *fitness*-nya pada fase inisialisasi.

Tabel 4.1 Hasil Fase Inisialisasi

Solusi	Rute	Jarak	<i>Fitness</i>
1	[3, 9, 4, 7, 5, 1, 2, 0, 6, 8]	450	0,002217
2	[1, 2, 7, 4, 5, 3, 0, 9, 8, 6]	536	0,001862
3	[7, 2, 9, 1, 0, 8, 3, 5, 4, 6]	421	0,002370
4	[1, 0, 6, 7, 4, 8, 3, 9, 5, 2]	457	0,002183
5	[3, 1, 5, 4, 7, 9, 8, 0, 6, 2]	509	0,001961
6	[5, 7, 3, 8, 4, 9, 2, 6, 1, 0]	420	0,002375
7	[8, 5, 1, 0, 6, 3, 2, 9, 7, 4]	484	0,002062
8	[7, 5, 6, 1, 8, 2, 0, 4, 3, 9]	536	0,001862
...	...	...	...
25	[0, 4, 1, 2, 5, 7, 8, 3, 6, 9]	544	0,001835

2. Fase *Employed Bee*

Pada fase ini, setiap *employed bee* memilih secara acak satu solusi awal dan memperbaikinya menggunakan *swap operator* dan *swap sequence*.

a. Perhitungan *Swap Operator*

Pada tahap ini, setiap *employed bee* menghasilkan solusi kandidat baru dengan menggunakan *swap operator* terhadap solusi awal yang telah dipilih dari fase inisialisasi. Hasil perhitungan *swap operator* tersebut ditunjukkan pada Tabel 4.2, yang memuat informasi mengenai *fitness* awal, operasi *swap*, kandidat solusi baru berupa rute baru hasil pertukaran, nilai *fitness* dari kandidat solusi baru, serta nilai *trial* yang menunjukkan bahwa apakah ada perbaikan solusi atau tidak.

Tabel 4.2 Perhitungan *Swap Operator*

<i>Bee</i>	<i>Fitness</i> Awal	Operasi	Rute Baru	<i>Fitness</i> Baru	<i>Trial</i>
1	0,002217	<i>SO</i> (5,2)	[3, 9, 1, 7, 5, 4, 2, 0, 6, 8]	0,002433	0
2	0,001862	<i>SO</i> (1,3)	[1, 4, 7, 2, 5, 3, 0, 9, 8, 6]	0,001497	1
3	0,002370	<i>SO</i> (7,5)	[7, 2, 9, 1, 0, 5, 3, 8, 4, 6]	0,002463	0
4	0,002183	<i>SO</i> (4,2)	[1, 0, 4, 7, 6, 8, 3, 9, 5, 2]	0,001976	1
...	...	...	...	...	...
25	0,001835	<i>SO</i> (0,3)	[2, 4, 1, 0, 5, 7, 8, 3, 6, 9]	0,002584	0

Aturan penentuan nilai *trial* tersebut didasarkan pada mekanisme *greedy selection* yaitu jika nilai *fitness* solusi baru memiliki *fitness* yang lebih besar, maka solusi tersebut akan terpilih untuk tahap selanjutnya dan nilai *trial* pada solusi baru tersebut direset menjadi nol. Jika nilai *fitness* solusi baru tidak lebih besar, maka solusi lama yang akan dipilih untuk tahap selanjutnya dan nilai *trial* ditambah satu. Tabel 4.3 menunjukkan individu yang terpilih untuk dilanjutkan pada tahap perhitungan *swap sequence*.

Tabel 4.3 Individu Terpilih

<i>Bee</i>	Solusi Terpilih	<i>Fitness</i>	<i>Trial</i>
1	[3, 9, 1, 7, 5, 4, 2, 0, 6, 8]	0,002433	0
2	[1, 2, 7, 4, 5, 3, 0, 9, 8, 6]	0,001862	1
3	[7, 2, 9, 1, 0, 5, 3, 8, 4, 6]	0,002463	0
4	[1, 0, 6, 7, 4, 8, 3, 9, 5, 2]	0,002183	1
5	[3, 1, 5, 4, 7, 9, 8, 0, 6, 2]	0,001961	1
6	[5, 7, 3, 4, 8, 9, 2, 6, 1, 0]	0,002597	0
7	[8, 5, 1, 0, 6, 3, 2, 9, 7, 4]	0,002062	1
8	[7, 5, 9, 1, 8, 2, 0, 4, 3, 6]	0,001996	0
...	...	...	...
25	[2, 4, 1, 0, 5, 7, 8, 3, 6, 9]	0,002584	0

b. Perhitungan *Swap Sequence*

Pada tahap ini, setiap individu yang terpilih akan digunakan sebagai solusi awal untuk diperbarui menggunakan *swap sequence* sebanyak *number of sequence* (*Nse*) yaitu $\left\lfloor \frac{n}{2} \right\rfloor = \left\lfloor \frac{10}{2} \right\rfloor = 5$, artinya dilakukan *swap* sebanyak lima kali secara berurutan. Tabel 4.4 menunjukkan proses pada tahap *swap sequence* sampai diperoleh hasil perhitungan *swap sequence*. Setiap operasi *swap* menghasilkan solusi baru yang kemudian dievaluasi berdasarkan nilai jarak tempuh dan nilai *fitness*-nya.

Tabel 4.4 Perhitungan *Swap Sequence*

<i>Bee</i>	Operasi	Rute	Jarak	<i>Fitness</i>	<i>Trial</i>
1	<i>SO</i> (7,1)	[3, 0, 1, 7, 5, 4, 2, 9, 6, 8]	407	0,002451	0
	<i>SO</i> (3,0)	[7, 0, 1, 3, 5, 4, 2, 9, 6, 8]	379	0,002632	
	<i>SO</i> (0,1)	[0, 7, 1, 3, 5, 4, 2, 9, 6, 8]	414	0,002410	
	<i>SO</i> (5,3)	[0, 7, 1, 4, 5, 3, 2, 9, 6, 8]	458	0,002179	
	<i>SO</i> (6,0)	[2, 7, 1, 4, 5, 3, 0, 9, 6, 8]	532	0,001876	
2	<i>SO</i> (4,9)	[1, 2, 7, 4, 6, 3, 0, 9, 8, 5]	539	0,001852	0
	<i>SO</i> (5,3)	[1, 2, 7, 3, 6, 4, 0, 9, 8, 5]	497	0,002008	
	<i>SO</i> (6,4)	[1, 2, 7, 3, 0, 4, 6, 9, 8, 5]	507	0,001969	
	<i>SO</i> (7,9)	[1, 2, 7, 3, 0, 4, 6, 5, 8, 9]	574	0,001739	
	<i>SO</i> (6,2)	[1, 2, 6, 3, 0, 4, 7, 5, 8, 9]	542	0,001842	
3	<i>SO</i> (8,3)	[7, 2, 9, 4, 0, 5, 3, 8, 1, 6]	518	0,001927	1
	<i>SO</i> (8,0)	[1, 2, 9, 4, 0, 5, 3, 8, 7, 6]	417	0,002392	
	<i>SO</i> (1,6)	[1, 3, 9, 4, 0, 5, 2, 8, 7, 6]	511	0,001953	
	<i>SO</i> (2,1)	[1, 9, 3, 4, 0, 5, 2, 8, 7, 6]	514	0,001942	
	<i>SO</i> (4,0)	[0, 9, 3, 4, 1, 5, 2, 8, 7, 6]	554	0,001802	
...	...	...	...	...	...
	...	...	...	...	
	...	...	...	...	
	...	...	...	...	
	...	...	...	...	
25	<i>SO</i> (3,7)	[2, 4, 1, 3, 5, 7, 8, 0, 6, 9]	512	0,001949	1
	<i>SO</i> (8,3)	[2, 4, 1, 6, 5, 7, 8, 0, 3, 9]	648	0,001541	
	<i>SO</i> (3,6)	[2, 4, 1, 8, 5, 7, 6, 0, 3, 9]	574	0,001739	
	<i>SO</i> (2,1)	[2, 1, 4, 8, 5, 7, 6, 0, 3, 9]	537	0,001859	
	<i>SO</i> (0,1)	[1, 2, 4, 8, 5, 7, 6, 0, 3, 9]	513	0,001946	

Hasil dari *swap sequence* kemudian dipilih satu solusi baru yang terbaik (*fitness* terbesar) di setiap *bee*. Selanjutnya solusi terbaik dari *swap sequence* tersebut akan dibandingkan dengan solusi awal pada tahap ini menggunakan *greedy selection*. Hasil dari *greedy selection* ini untuk menghitung *trial* setiap *employed bee* dan memilih hasil akhir solusi terbaik yang diperbarui oleh *employed bee* yang selanjutnya akan dipilih dan diperbarui lebih lanjut pada fase *onlooker bee*.

3. Fase *Onlooker bee*

Pada fase ini, setiap solusi akhir yang dihasilkan oleh fase *employed bee* akan menjadi solusi awal bagi fase *onlooker* dengan memilih solusi berdasarkan kualitasnya. Pemilihan solusi ini dilakukan secara probabilistik, di mana solusi dengan nilai *fitness* yang lebih besar memiliki peluang yang lebih besar untuk dipilih, mekanisme ini dikenal sebagai *roulette wheel selection*. Setelah suatu solusi terpilih, *onlooker bee* akan memperbarui solusi tersebut menggunakan *insert operator* dan *insert sequence*.

a. Perhitungan Probabilitas Kumulatif

Pada tahap ini, setiap solusi akhir yang diperoleh pada fase *employed bee* akan dihitung nilai probabilitasnya dengan Persamaan 2.6 dan nilai probabilitas kumulatifnya dengan menggunakan Persamaan 2.7. Nilai probabilitas kumulatif ini digunakan sebagai interval pemilihan solusi oleh *onlooker bee* dalam mekanisme *roulette wheel selection*. Tabel 4.5 menunjukkan proses perhitungan probabilitas kumulatif yang dimulai dari perhitungan probabilitas atas nilai *fitness* yang dihasilkan oleh *bee*.

Tabel 4.5 Perhitungan Probabilitas Kumulatif

<i>Bee</i>	<i>Fitness</i>	Probabilitas	Probabilitas Kumulatif
1	0,002631579	0,043206889	0,043206889
2	0,002008032	0,032969112	0,076176001
3	0,002463054	0,040439945	0,116615947
...	...	...	...
25	0,002583979	0,042425369	1

b. *Roulette Wheel Selection*

Pada tahap ini dilakukan pemilihan solusi oleh *onlooker bee* menggunakan nilai acak r yang dibangkitkan dalam interval $[0, 1]$, Nilai r kemudian

dibandingkan dengan rentang probabilitas kumulatif yang telah dihitung sebelumnya untuk menentukan solusi dari *employed bee* yang terpilih. Tabel 4.6 menunjukkan hasil pemilihan solusi menggunakan *roulette wheel selection*.

Tabel 4.6 *Roulette Wheel Selection*

<i>Bee</i>	Nilai <i>r</i>	<i>Employed Bee</i>	<i>Fitness</i>
1	0,33529367	9	0,002222222
2	0,840200636	22	0,002717391
3	0,201187756	6	0,002597403
...	...	...	...
25	0,834257188	22	0,002717391

c. Perhitungan *Insert Operator*

Pada tahap ini, setiap *onlooker bee* menghasilkan solusi kandidat baru dengan menggunakan *insert operator* terhadap solusi yang telah dipilih dengan *roulette wheel selection*. Hasil perhitungan *insert operator* tersebut menghasilkan kandidat solusi baru yang ditunjukkan pada Tabel 4.7 yaitu sebagai berikut.

Tabel 4.7 Perhitungan *Insert Operator*

<i>Bee</i>	<i>Fitness Awal</i>	Operasi	Rute Baru	<i>Fitness Baru</i>	<i>Trial</i>
1	0,002222	<i>IO</i> (8,4)	[5, 6, 0, 7, 2, 1, 4, 9, 8, 3]	0,002101	3
2	0,002717	<i>IO</i> (5,9)	[7, 1, 2, 6, 8, 4, 3, 0, 5, 9]	0,002315	1
3	0,002597	<i>IO</i> (5,2)	[5, 7, 9, 3, 4, 8, 2, 6, 1, 0]	0,002525	2
...	...	...	...	...	...
25	0,002717	<i>IO</i> (9,3)	[7, 1, 2, 5, 6, 8, 9, 4, 3, 0]	0,002198	1

Aturan penentuan nilai *trial* tersebut didasarkan pada mekanisme *greedy selection*. Tabel 4.8 menunjukkan individu yang terpilih berdasarkan mekanisme *greedy selection* untuk dilanjutkan pada tahap perhitungan *insert sequence*.

Tabel 4.8 Individu Terpilih

<i>Bee</i>	Solusi Terpilih	<i>Fitness</i>	<i>Trial</i>
1	[5, 6, 0, 7, 1, 4, 9, 8, 2, 3]	0,002222	3
2	[7, 1, 2, 6, 8, 9, 4, 3, 0, 5]	0,002717	1
3	[5, 7, 3, 4, 8, 9, 2, 6, 1, 0]	0,002597	2
...	...	...	...
25	[7, 1, 2, 6, 8, 9, 4, 3, 0, 5]	0,002717	1

d. Perhitungan *Insert Sequence*

Pada tahap ini, setiap individu yang terpilih akan digunakan sebagai solusi awal untuk diperbarui menggunakan *insert sequence* sebanyak *Number of Sequence* (Nse) yaitu $\left\lfloor \frac{n}{2} \right\rfloor = \left\lfloor \frac{10}{2} \right\rfloor = 5$, artinya dilakukan *insert* sebanyak lima kali secara berurutan. Dari hasil *insert sequence* ini kemudian dipilih satu solusi baru yang terbaik di setiap *bee*. Setiap solusi yang dihasilkan dari rangkaian operasi *insert* kemudian dievaluasi berdasarkan nilai jarak dan nilai *fitness*-nya. Dari seluruh solusi yang terbentuk, dipilih satu solusi terbaik pada masing-masing *bee*, yaitu solusi yang memiliki nilai *fitness* terbesar atau jarak rute paling kecil. Solusi terbaik hasil *insert sequence* tersebut selanjutnya dibandingkan dengan solusi awal menggunakan mekanisme *greedy selection*. Melalui mekanisme ini, solusi baru akan menggantikan solusi awal apabila memiliki kualitas yang lebih baik, sedangkan solusi awal dipertahankan apabila solusi baru tidak memberikan perbaikan. Hasil dari *greedy selection* ini untuk menghitung *trial* setiap *onlooker bee* dan memilih hasil akhir solusi terbaik yang diperbarui oleh *onlooker bee* yang selanjutnya akan dievaluasi pada fase *scout bee*. Tabel 4.9 menunjukkan proses pada tahap *insert sequence* sampai diperoleh hasil perhitungannya.

Tabel 4.9 Perhitungan *Insert Sequence*

<i>Bee</i>	<i>Operasi</i>	<i>Tour</i>	<i>Jarak</i>	<i>Fitness</i>	<i>Trial</i>
1	<i>IO</i> (6,3)	[5, 6, 0, 9, 7, 1, 4, 8, 2, 3]	540	0,001848	0
	<i>IO</i> (1,6)	[5, 0, 9, 7, 1, 4, 6, 8, 2, 3]	431	0,002315	
	<i>IO</i> (2,1)	[5, 9, 0, 7, 1, 4, 6, 8, 2, 3]	516	0,001934	
	<i>IO</i> (7,9)	[5, 9, 0, 7, 1, 4, 6, 2, 3, 8]	575	0,001736	
	<i>IO</i> (0,8)	[9, 0, 7, 1, 4, 6, 2, 3, 5, 8]	476	0,002096	
2	<i>IO</i> (8,7)	[7, 1, 2, 6, 8, 9, 4, 0, 3, 5]	431	0,002315	2
	<i>IO</i> (9,6)	[7, 1, 2, 6, 8, 9, 5, 4, 0, 3]	450	0,002217	
	<i>IO</i> (4,1)	[7, 8, 1, 2, 6, 9, 5, 4, 0, 3]	505	0,001976	
	<i>IO</i> (6,9)	[7, 8, 1, 2, 6, 9, 4, 0, 3, 5]	486	0,002053	
	<i>IO</i> (3,1)	[7, 2, 8, 1, 6, 9, 4, 0, 3, 5]	584	0,001709	
3	<i>IO</i> (1,0)	[7, 5, 3, 4, 8, 9, 2, 6, 1, 0]	405	0,002463	0
	<i>IO</i> (0,3)	[5, 3, 4, 7, 8, 9, 2, 6, 1, 0]	377	0,002646	
	<i>IO</i> (4,7)	[5, 3, 4, 7, 9, 2, 6, 8, 1, 0]	392	0,002545	
	<i>IO</i> (1,6)	[5, 4, 7, 9, 2, 6, 3, 8, 1, 0]	363	0,002747	
	<i>IO</i> (9,8)	[5, 4, 7, 9, 2, 6, 3, 8, 0, 1]	428	0,002331	
...	...	...	...	...	...
25	<i>IO</i> (0,5)	[1, 2, 6, 8, 9, 7, 4, 3, 0, 5]	436	0,002288	2
	<i>IO</i> (7,6)	[1, 2, 6, 8, 9, 7, 3, 4, 0, 5]	394	0,002532	
	<i>IO</i> (2,3)	[1, 2, 8, 6, 9, 7, 3, 4, 0, 5]	396	0,002519	
	<i>IO</i> (7,2)	[1, 2, 4, 8, 6, 9, 7, 3, 0, 5]	421	0,002370	
	<i>IO</i> (5,0)	[9, 1, 2, 4, 8, 6, 7, 3, 0, 5]	480	0,002079	

4. Fase *Scout Bee*

Pada fase terakhir ini, nilai *trial* dari seluruh solusi akhir dievaluasi untuk mengidentifikasi solusi yang mengalami stagnasi. Semua nilai *trial* dari setiap solusi dibandingkan dengan parameter *limit*. Apabila *trial* suatu solusi telah melampaui nilai *limit*, maka solusi tersebut akan ditinggalkan. Dalam hal ini, *employed bee* yang bertanggung jawab terhadap solusi tersebut akan diubah menjadi *scout bee* dan selanjutnya menghasilkan solusi baru secara acak dan nilai *trial* direset menjadi nol. Fase ini adalah mekanisme penting dalam algoritma ABC untuk mencegah konvergensi prematur dan menjaga daya

jelajah algoritma. Tabel 4.10 menunjukkan proses evaluasi solusi akhir oleh *scout bee*.

Tabel 4.10 Evaluasi Solusi Akhir

<i>Bee</i>	<i>Trial Awal</i>	<i>Trial Akhir</i>	Status	Rute	Jarak	<i>Fitness</i>
1	0	0	Tetap	[5, 0, 9, 7, 1, 4, 6, 8, 2, 3]	431	0,002315
2	2	0	Baru	[1, 7, 2, 9, 5, 0, 8, 6, 4, 3]	498	0,002004
3	0	0	Tetap	[5, 4, 7, 9, 2, 6, 3, 8, 1, 0]	363	0,002747
4	1	1	Tetap	[5, 7, 3, 8, 9, 4, 2, 6, 1, 0]	360	0,002770
5	2	0	Baru	[7, 9, 4, 2, 6, 3, 0, 1, 8, 5]	447	0,002232
6	0	0	Tetap	[1, 2, 4, 3, 6, 7, 8, 9, 0, 5]	362	0,002755
7	2	0	Baru	[7, 0, 4, 3, 6, 8, 9, 2, 5, 1]	462	0,002160
8	2	0	Baru	[4, 1, 8, 9, 7, 6, 5, 0, 3, 2]	524	0,001905
9	2	0	Baru	[5, 7, 3, 0, 9, 8, 2, 4, 6, 1]	520	0,001919
10	0	0	Tetap	[9, 7, 1, 2, 0, 5, 8, 4, 3, 6]	283	0,003521
11	0	0	Tetap	[8, 7, 1, 2, 3, 5, 0, 4, 6, 9]	316	0,003155
12	4	0	Baru	[6, 1, 7, 2, 0, 8, 5, 9, 4, 3]	497	0,002008
13	1	1	Tetap	[9, 4, 3, 8, 2, 6, 7, 1, 0, 5]	343	0,002907
14	2	0	Baru	[0, 9, 3, 6, 1, 5, 4, 8, 7, 2]	526	0,001898
15	0	0	Tetap	[1, 2, 7, 3, 4, 0, 5, 8, 9, 6]	452	0,002208
16	0	0	Tetap	[9, 3, 7, 0, 1, 2, 8, 5, 4, 6]	293	0,003401
17	0	0	Tetap	[2, 1, 0, 5, 7, 8, 3, 6, 4, 9]	288	0,003460
18	0	0	Tetap	[4, 9, 8, 3, 6, 2, 0, 1, 7, 5]	266	0,003745
19	2	0	Baru	[7, 8, 0, 5, 2, 4, 1, 3, 6, 9]	465	0,002146
20	0	0	Tetap	[3, 5, 6, 1, 7, 9, 8, 2, 0, 4]	418	0,002387
21	0	0	Tetap	[3, 5, 1, 0, 2, 6, 4, 8, 9, 7]	302	0,003300
22	2	0	Baru	[7, 6, 2, 4, 5, 1, 8, 0, 3, 9]	502	0,001988
23	3	0	Baru	[5, 9, 4, 6, 1, 2, 0, 3, 8, 7]	469	0,002128
24	0	0	Tetap	[5, 6, 0, 7, 1, 9, 8, 2, 4, 3]	408	0,002445
25	2	0	Baru	[7, 9, 6, 8, 5, 4, 3, 2, 1, 0]	360	0,002770

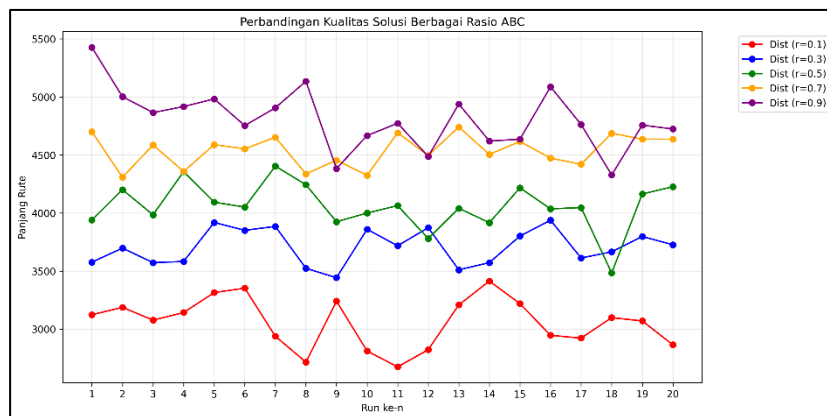
Pada iterasi selanjutnya, solusi akhir yang diperoleh dari iterasi sebelumnya digunakan sebagai solusi awal untuk melanjutkan proses pencarian solusi. Algoritma kemudian kembali menjalankan fase *employed bee*, fase *onlooker bee*, dan fase *scout bee* dengan prosedur yang sama seperti iterasi pertama. Proses iteratif ini dilakukan secara berulang untuk memperbaiki kualitas solusi

hingga mencapai jumlah iterasi maksimum yang telah ditetapkan. Pada iterasi terakhir, setelah fase *scout bee* menggantikan solusi yang tidak mengalami perbaikan dengan solusi yang baru, seluruh solusi dievaluasi kembali menggunakan nilai *fitness*. Solusi dengan nilai *fitness* terbesar atau rute paling pendek dipilih sebagai solusi akhir hasil optimasi algoritma ABC.

Eksperimen pada penelitian ini menggunakan tiga graf komplit berbobot acak, yaitu graf dengan 50 simpul, 100 simpul, dan 200 simpul. Setiap graf diproses menggunakan konfigurasi parameter yang telah ditetapkan di Bab III. Berikut adalah hasil dari perhitungan rute terpendek untuk setiap graf menggunakan algoritma *Artificial Bee Colony* untuk setiap rasio r_c dengan bahasa pemrograman Python.

4.4.1 Hasil dari Graf Pertama

Data hasil perhitungan rute terpendek yang dihasilkan oleh algoritma ABC untuk graf dengan 50 simpul divisualisasikan sebagai berikut.

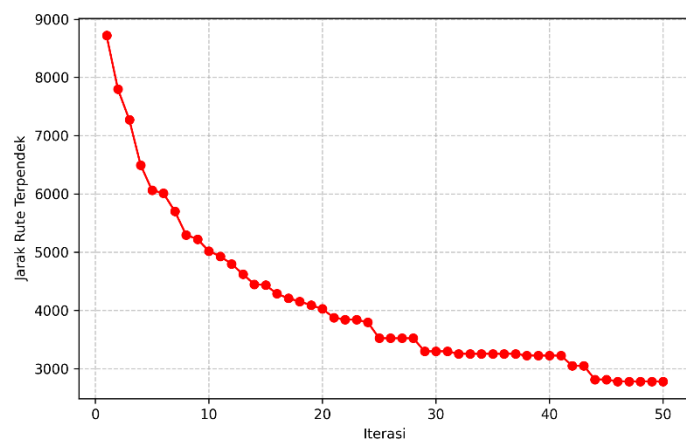


Gambar 4.1 Grafik Data Hasil dari Graf Pertama

Gambar 4.1 menyajikan visualisasi data dari hasil pengujian algoritma *Artificial Bee Colony* (ABC) pada graf 50 simpul dengan lima variasi rasio banyaknya *employed bee* dan *onlooker bee* yang berbeda. Visualisasi tersebut

merupakan ringkasan dari data hasil eksperimen yang disajikan secara lengkap pada Lampiran 3, yang memuat hasil 20 kali percobaan untuk setiap variasi rasio beserta nilai fitness dan iterasi berhenti yang diperoleh. Berdasarkan data tersebut, terlihat bahwa penggunaan rasio 0,1 cenderung menghasilkan rata-rata jarak rute yang lebih pendek dibandingkan variasi rasio lainnya.

Untuk memberikan gambaran mengenai proses pencarian solusi yang dilakukan algoritma ABC, visualisasi konvergensi untuk rasio terbaik, ditunjukkan pada Gambar 4.2.



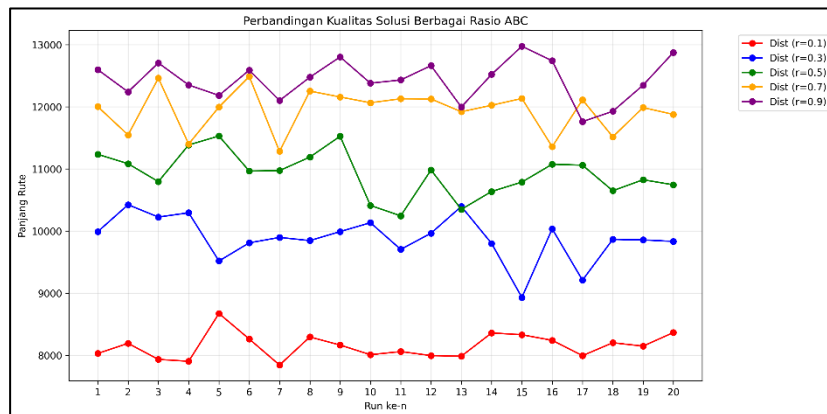
Gambar 4.2 Visualisasi Konvergensi Rasio 0,1 dari Graf 50 Simpul

Gambar 4.2 menunjukkan bahwa algoritma ABC dengan rasio 0,1 atau 1:9 mengalami penurunan jarak rute yang konsisten hingga mencapai kondisi konvergen. Dominasi jumlah *onlooker bee* pada rasio ini menyebabkan proses eksploitasi terhadap sumber makanan yang memiliki nilai *fitness* tinggi menjadi lebih intensif dibandingkan eksplorasi. Akibatnya, algoritma lebih efektif dalam menyempurnakan solusi yang menjanjikan sehingga menghasilkan kualitas solusi terbaik pada graf 50 simpul.

4.4.2 Hasil dari Graf Kedua

Data hasil perhitungan rute terpendek yang dihasilkan oleh algoritma ABC

untuk graf dengan 100 simpul divisualisasikan sebagai berikut.

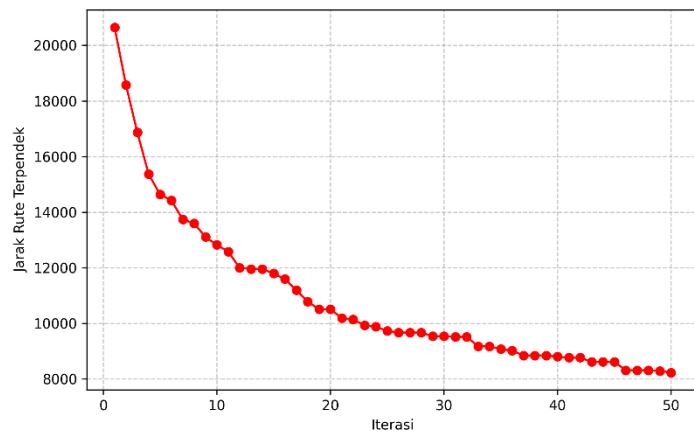


Gambar 4.3 Grafik Data Hasil dari Graf Kedua

Gambar 4.3 menyajikan visualisasi data dari hasil pengujian algoritma *Artificial Bee Colony* (ABC) pada graf 100 simpul dengan lima variasi rasio banyaknya *employed bee* dan *onlooker bee* yang berbeda. Visualisasi tersebut merupakan ringkasan dari data hasil eksperimen yang disajikan secara lengkap pada Lampiran 4, yang memuat hasil 20 kali percobaan untuk setiap variasi rasio beserta nilai fitness dan iterasi berhenti yang diperoleh. Berdasarkan data tersebut, terlihat bahwa penggunaan rasio 0,1 cenderung menghasilkan rata-rata jarak rute yang lebih pendek dibandingkan variasi rasio lainnya. Hal ini menunjukkan bahwa pada skala graf sedang, konfigurasi dengan banyaknya *onlooker bee* yang lebih dominan memberikan hasil pencarian solusi yang lebih optimal.

Gambar 4.4 menunjukkan perkembangan jarak rute terbaik yang diperoleh algoritma ABC pada setiap iterasi untuk graf 100 simpul dengan rasio *employed bee* dan *onlooker bee* sebesar 1:9. Terlihat bahwa jarak rute mengalami penurunan yang cukup signifikan pada iterasi awal, kemudian laju perbaikannya semakin melambat hingga kurva cenderung mendatar pada iterasi akhir. Kondisi tersebut mengindikasikan bahwa algoritma telah mencapai konvergensi dan

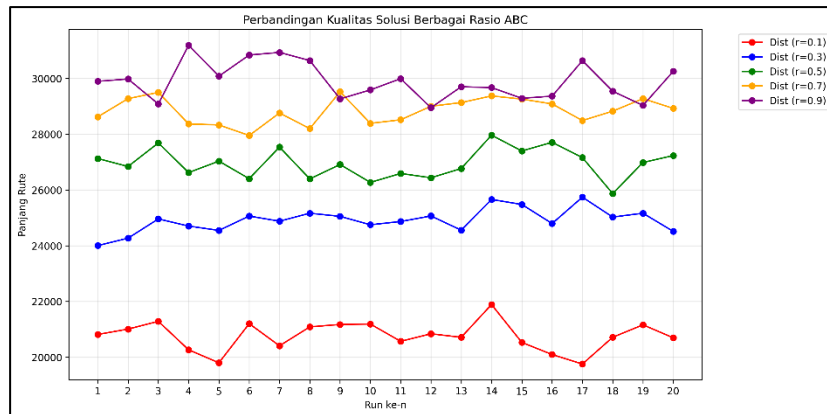
memperoleh solusi yang stabil setelah melalui proses eksploitasi yang dilakukan secara intensif oleh *onlooker bee*.



Gambar 4.4 Visualisasi Konvergensi Rasio 0,1 dari Graf 100 Simpul

4.4.3 Hasil dari Graf Ketiga

Data hasil perhitungan rute terpendek yang dihasilkan oleh algoritma ABC untuk graf dengan 200 simpul divisualisasikan sebagai berikut.

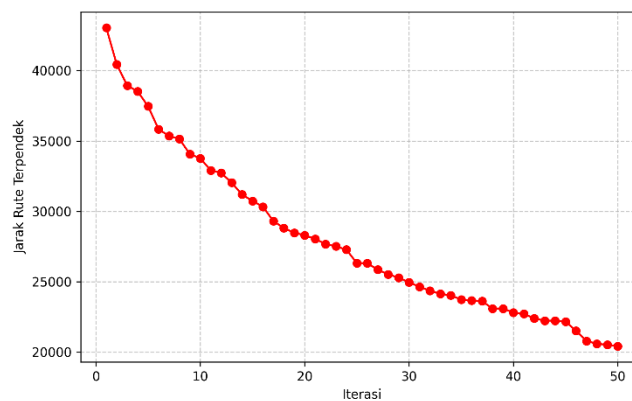


Gambar 4.5 Grafik Data Hasil dari Graf Ketiga

Gambar 4.5 menyajikan visualisasi data dari hasil pengujian algoritma *Artificial Bee Colony* (ABC) pada graf 200 simpul dengan lima variasi rasio banyaknya *employed bee* dan *onlooker bee* yang berbeda. Visualisasi tersebut merupakan ringkasan dari data hasil eksperimen yang disajikan secara lengkap pada Lampiran 5, yang memuat hasil 20 kali percobaan untuk setiap variasi rasio

beserta nilai fitness dan iterasi berhenti yang diperoleh. Berdasarkan data tersebut, terlihat bahwa penggunaan rasio 0,1 cenderung menghasilkan rata-rata jarak rute yang lebih pendek dibandingkan variasi rasio lainnya. Hal ini menunjukkan bahwa pada skala graf besar, konfigurasi dengan banyaknya *onlooker bee* yang lebih dominan memberikan hasil pencarian solusi yang lebih optimal.

Untuk memberikan gambaran mengenai proses pencarian solusi yang dilakukan algoritma ABC, visualisasi konvergensi untuk rasio terbaik, ditunjukkan pada Gambar 4.6.



Gambar 4.6 Visualisasi Konvergensi Rasio 0,1 dari Graf 200 Simpul

Gambar 4.6 menunjukkan perkembangan jarak rute terbaik yang diperoleh algoritma ABC pada setiap iterasi untuk graf 200 simpul dengan rasio *employed bee* dan *onlooker bee* sebesar 1:9. Terlihat bahwa jarak rute mengalami penurunan yang cukup signifikan pada iterasi awal, kemudian terus membaik secara bertahap hingga iterasi akhir dengan laju perbaikan yang semakin kecil. Pola tersebut menunjukkan bahwa dominasi *onlooker bee* meningkatkan intensitas eksploitasi terhadap solusi berkualitas sehingga proses pencarian tetap terarah dan mampu menghasilkan solusi yang semakin baik meskipun kompleksitas graf lebih besar.

4.5 Analisis Data Statistik dan Penarikan Kesimpulan

Teknik analisis data yang digunakan untuk mengevaluasi pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma *Artificial Bee Colony* (ABC) pada penyelesaian TSP adalah *One Way Analysis of Variance* (ANOVA) menggunakan program SPSS. Analisis ini bertujuan untuk menguji apakah terdapat perbedaan secara signifikan antara penggunaan lima variasi rasio banyaknya *employed bee* dan *onlooker bee* (r_c) yaitu 0,1, 0,3, 0,5, 0,7, dan 0,9 terhadap hasil rute terpendek yang didapat. ANOVA memiliki beberapa uji asumsi yang harus terpenuhi sebagai penentu kelayakan penggunaannya sebelum uji signifikansi dapat dilakukan. Uji asumsi tersebut meliputi uji normalitas untuk memastikan bahwa data berdistribusi secara normal dan uji homogenitas untuk mengetahui apakah varians data pada setiap kelompok rasio yang dibandingkan memiliki tingkat keragaman yang sama.

4.5.1 Uji Normalitas Data

Pengujian normalitas data pada penelitian ini menggunakan metode *Shapiro-Wilk*. Metode ini dipilih karena memiliki tingkat sensitivitas dan akurasi yang tinggi untuk ukuran sampel kecil (kurang dari 50 data/sampel per kelompok) hingga menengah. Hipotesis dari uji normalitas ini yaitu:

$$H_0 : \text{Data berdistribusi normal}$$

$$H_1 : \text{Data tidak berdistribusi normal}$$

Dengan kriteria pengujiannya adalah:

Jika nilai signifikansi ($Sig.$) $> 0,05$, maka H_0 diterima dan H_1 ditolak.

Jika nilai signifikansi ($Sig.$) $< 0,05$, maka H_0 ditolak dan H_1 diterima.

1. Uji Normalitas Data dari Graf Pertama

Berikut adalah output uji normalitas menggunakan program SPSS:

Tests of Normality						
Rasio	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Jarak Rasio 1:9	.125	20	.200 [*]	.970	20	.751
Rasio 3:7	.142	20	.200 [*]	.942	20	.258
Rasio 1:1	.144	20	.200 [*]	.937	20	.211
Rasio 7:3	.136	20	.200 [*]	.937	20	.210
Rasio 9:1	.103	20	.200 [*]	.979	20	.923

Gambar 4.7 Hasil Uji Normalitas Data dari Graf Pertama

Berdasarkan hasil program SPSS pada Gambar 4.7, diperoleh nilai signifikansi untuk setiap variasi rasio yaitu rasio 1: 9 sebesar 0,751, rasio 3: 7 sebesar 0,258, rasio 1: 1 sebesar 0,211, rasio 7: 3 sebesar 0,210, dan rasio 9: 1 sebesar 0,923. Secara matematis, seluruh nilai signifikansi tersebut lebih besar dari 0,05 ($p > 0,05$). Dengan demikian, H_0 diterima, sehingga dapat disimpulkan bahwa data jarak rute terpendek pada graf 50 simpul untuk kelima variasi rasio berdistribusi normal.

2. Uji Normalitas Data dari Graf Kedua

Berikut adalah output uji normalitas menggunakan program SPSS:

Tests of Normality						
Rasio	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Statistic	df	Sig.	Statistic	df	Sig.
Jarak Rasio 1:9	.124	20	.200 [*]	.948	20	.339
Rasio 3:7	.206	20	.026	.916	20	.081
Rasio 1:1	.098	20	.200 [*]	.973	20	.826
Rasio 7:3	.202	20	.032	.910	20	.063
Rasio 9:1	.097	20	.200 [*]	.981	20	.947

Gambar 4.8 Hasil Uji Normalitas Data dari Graf Kedua

Berdasarkan hasil program SPSS pada Gambar 4.8, diperoleh nilai signifikansi terkecil terdapat pada rasio 7: 3 sebesar 0,063 dan tertinggi pada rasio 9:1 sebesar 0,947. Walaupun terdapat fluktuasi nilai signifikansi dibandingkan graf sebelumnya, setiap nilai signifikansi tetap lebih besar dari 0,05 ($p > 0,05$). Dengan demikian, H_0 diterima, sehingga dapat disimpulkan

bahwa data jarak rute terpendek pada graf 100 simpul untuk kelima variasi berdistribusi normal.

3. Uji Normalitas Data dari Graf Ketiga

Berikut adalah output uji normalitas menggunakan program SPSS:

Tests of Normality							
Rasio		Kolmogorov-Smirnov ^a			Shapiro-Wilk		
		Statistic	df	Sig.	Statistic	df	Sig.
Jarak	Rasio 1:9	.109	20	.200 [*]	.966	20	.673
	Rasio 3:7	.129	20	.200 [*]	.977	20	.888
	Rasio 1:1	.075	20	.200 [*]	.986	20	.988
	Rasio 7:3	.117	20	.200 [*]	.956	20	.475
	Rasio 9:1	.115	20	.200 [*]	.948	20	.343

Gambar 4.9 Hasil Uji Normalitas Data dari Graf Ketiga

Hasil uji normalitas untuk graf dengan kompleksitas tertinggi (200 simpul) ditunjukkan pada Gambar 4.9. Hasil uji diperoleh bahwa nilai signifikansi yang cukup tinggi, khususnya pada rasio 1:1 dengan $p = 0,988$. Karena seluruh nilai signifikansi pada setiap rasio lebih besar dari 0,05 ($p > 0,05$), maka H_0 diterima, sehingga dapat disimpulkan bahwa data jarak rute terpendek pada graf 200 simpul untuk kelima variasi berdistribusi normal.

4.5.2 Uji Homogenitas

Uji homogenitas dilakukan untuk memverifikasi apakah kelompok data yang berasal dari berbagai variasi rasio lebah memiliki tingkat sebaran yang setara. Asumsi homogenitas merupakan syarat mutlak dalam analisis *One Way* ANOVA agar perbedaan yang muncul pada variabel terikat murni disebabkan oleh perlakuan rasio, bukan akibat ketidakkonsistenan varians antar kelompok data. Metode yang digunakan adalah Uji *Levene* dan hipotesis yaitu:

H_0 : Varians data pada setiap kelompok rasio adalah sama (homogen)

H_1 : Varians data pada setiap kelompok rasio adalah tidak sama (non-homogen)

Dengan kriteria pengujiannya sebagai berikut.

Jika nilai signifikansi (*Sig.*) $> 0,05$, maka H_0 diterima dan H_1 ditolak.

Jika nilai signifikansi (*Sig.*) $< 0,05$, maka H_0 ditolak dan H_1 diterima.

1. Uji Homogenitas Data dari Graf Pertama

Berikut adalah output uji homogenitas menggunakan program SPSS:

Test of Homogeneity of Variances			
Jarak			
Levene Statistic	df1	df2	Sig.
1.674	4	95	.162

Gambar 4.10 Hasil Uji Homogenitas Data dari Graf Pertama

Berdasarkan hasil uji *Levene* pada Gambar 4.10 diperoleh nilai signifikansi yaitu sebesar 0,162. Karena nilai $p = 0,162 > 0,05$, maka H_0 diterima.

Dengan demikian tidak terdapat perbedaan varians yang signifikan di antara kelima kelompok rasio pada graf 50 simpul, sehingga data dinyatakan homogen.

2. Uji Homogenitas Data dari Graf Kedua

Berikut adalah output uji homogenitas menggunakan program SPSS:

Test of Homogeneity of Variances			
Jarak			
Levene Statistic	df1	df2	Sig.
1.218	4	95	.309

Gambar 4.11 Hasil Uji Homogenitas Data dari Graf Kedua

Hasil uji homogenitas untuk graf 100 simpul disajikan pada Gambar 4.11, hasil uji menunjukkan nilai signifikansi sebesar 0,309. Karena nilai $p = 0,309 > 0,05$, maka H_0 diterima. Dengan demikian asumsi homogenitas untuk data pada graf 100 simpul telah terpenuhi.

3. Uji Homogenitas Data dari Graf Ketiga

Gambar 4.12 adalah *output* uji homogenitas menggunakan program SPSS.

Test of Homogeneity of Variances			
Jarak			
Levene Statistic	df1	df2	Sig.
1.438	4	95	.228

Gambar 4.12 Hasil Uji Homogenitas Data dari Graf Ketiga

Hasil uji homogenitas untuk graf 200 simpul disajikan pada Gambar 4.12, hasil uji menunjukkan nilai signifikansi sebesar 0,228. Karena nilai $p = 0,228 > 0,05$, maka H_0 diterima. Dengan demikian asumsi homogenitas untuk data pada graf 200 simpul juga telah terpenuhi.

4.5.3 Uji *One Way* ANOVA

Setelah seluruh uji asumsi normalitas dan uji homogenitas dinyatakan terpenuhi, maka analisis dilanjutkan dengan pengujian menggunakan metode *One Way Analysis of Variance* (ANOVA). Uji ini bertujuan untuk menganalisis pengaruh variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap panjang rute terpendek yang dihasilkan oleh algoritma *Artificial Bee Colony*. Hipotesis dalam pengujian ini yaitu:

H_0 : Tidak terdapat perbedaan signifikan dalam penggunaan variasi rasio.

H_1 : Ada perbedaan signifikan dalam penggunaan variasi rasio.

Dengan kriteria pengujiannya adalah:

Jika nilai signifikansi (*Sig.*) $> 0,05$, maka H_0 diterima dan H_1 ditolak.

Jika nilai signifikansi (*Sig.*) $< 0,05$, maka H_0 ditolak dan H_1 diterima.

Apabila hasil uji ANOVA pada penelitian ini menunjukkan adanya perbedaan yang signifikan, maka analisis dilanjutkan dengan uji lanjut (*Post-Hoc*

test) untuk mengidentifikasi pasangan kelompok rasio yang memiliki perbedaan secara spesifik.

1. Uji ANOVA Data dari Graf Pertama

ANOVA					
Jarak					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	38271809.84	4	9567952.460	242.893	.000
Within Groups	3742203.600	95	39391.617		
Total	42014013.44	99			

Gambar 4.13 Hasil Uji ANOVA Graf Pertama

Berdasarkan hasil uji ANOVA pada Gambar 4.13, diperoleh nilai signifikansi (*Sig.*) sebesar 0,000 yang lebih kecil dari taraf signifikansi 0,05. Dengan demikian, H_0 ditolak dan dapat disimpulkan bahwa terdapat perbedaan rata-rata yang signifikan antar variasi rasio *employed bee bee* dan *onlooker bee* terhadap panjang rute terpendek yang dihasilkan. Karena hasil uji ANOVA menunjukkan adanya perbedaan signifikan, maka akan dilakukan uji *Post Hoc* dilakukan untuk mengidentifikasi rasio mana yang terbaik.

(I) Rasio	(J) Rasio	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Rasio 1:9	Rasio 3:7	-648.90000*	62.76274	.000	-829.2847	-468.5153
	Rasio 1:1	-1001.15000*	62.76274	.000	-1181.5347	-820.7653
	Rasio 7:3	-1480.60000*	62.76274	.000	-1660.9847	-1300.2153
	Rasio 9:1	-1749.80000*	62.76274	.000	-1930.1847	-1569.4153
Rasio 3:7	Rasio 1:9	648.90000*	62.76274	.000	468.5153	829.2847
	Rasio 1:1	-352.25000*	62.76274	.000	-532.6347	-171.8653
	Rasio 7:3	-831.70000*	62.76274	.000	-1012.0847	-651.3153
	Rasio 9:1	-1100.90000*	62.76274	.000	-1281.2847	-920.5153
Rasio 1:1	Rasio 1:9	1001.15000*	62.76274	.000	820.7653	1181.5347
	Rasio 3:7	352.25000*	62.76274	.000	171.8653	532.6347
	Rasio 7:3	-479.45000*	62.76274	.000	-659.8347	-299.0653
	Rasio 9:1	-748.65000*	62.76274	.000	-929.0347	-568.2653
Rasio 7:3	Rasio 1:9	1480.60000*	62.76274	.000	1300.2153	1660.9847
	Rasio 3:7	831.70000*	62.76274	.000	651.3153	1012.0847
	Rasio 1:1	479.45000*	62.76274	.000	299.0653	659.8347
	Rasio 9:1	-269.20000*	62.76274	.000	-449.5847	-88.8153
Rasio 9:1	Rasio 1:9	1749.80000*	62.76274	.000	1569.4153	1930.1847
	Rasio 3:7	1100.90000*	62.76274	.000	920.5153	1281.2847
	Rasio 1:1	748.65000*	62.76274	.000	568.2653	929.0347
	Rasio 7:3	269.20000*	62.76274	.000	88.8153	449.5847

Gambar 4.14 Hasil Uji *Post-Hoc* Graf Pertama

Berdasarkan hasil uji lanjut Post-Hoc pada Gambar 4.14, seluruh pasangan rasio menunjukkan nilai signifikansi sebesar 0,000 yang lebih kecil dari 0,05. Dengan demikian dapat disimpulkan bahwa setiap pasangan variasi rasio memiliki perbedaan yang signifikan satu sama lain dalam menghasilkan panjang rute terpendek. Berdasarkan nilai *mean difference* pada hasil uji *Post-Hoc*, terlihat bahwa rasio 1:9 memiliki nilai selisih rata-rata yang bernilai negatif terhadap seluruh rasio pembanding, yaitu rasio 3:7, 1:1, 7:3, dan 9:1. Nilai negatif ini menunjukkan bahwa rata-rata panjang rute yang dihasilkan oleh rasio 1:9 lebih kecil dibandingkan dengan rasio lainnya. Sebaliknya, rasio dengan nilai *mean difference* positif menunjukkan rata-rata yang lebih besar, sehingga menghasilkan solusi yang kurang optimal.

2. Uji *One Way* ANOVA Data dari Graf Kedua

ANOVA					
Jarak					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	235608900.7	4	58902225.18	555.391	.000
Within Groups	10075258.70	95	106055.355		
Total	245684159.4	99			

Gambar 4.15 Hasil Uji ANOVA Graf Kedua

Berdasarkan hasil uji ANOVA pada Gambar 4.15, diperoleh nilai signifikansi (*Sig.*) sebesar 0,000 yang lebih kecil dari taraf signifikansi 0,05. Dengan demikian, H_0 ditolak dan dapat disimpulkan bahwa terdapat perbedaan rata-rata yang signifikan antar variasi rasio *employed bee* dan *onlooker bee* terhadap panjang rute terpendek yang dihasilkan. Karena hasil uji ANOVA menunjukkan adanya perbedaan signifikan, maka akan dilakukan uji *Post Hoc* dilakukan untuk mengidentifikasi rasio mana yang terbaik.

(I) Rasio	(J) Rasio	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Rasio 1:9	Rasio 3:7	-1738.65000 [*]	102.98318	.000	-2034.6312	-1442.6688
	Rasio 1:1	-2775.85000 [*]	102.98318	.000	-3071.8312	-2479.8688
	Rasio 7:3	-3796.10000 [*]	102.98318	.000	-4092.0812	-3500.1188
	Rasio 9:1	-4287.95000 [*]	102.98318	.000	-4583.9312	-3991.9688
Rasio 3:7	Rasio 1:9	1738.65000 [*]	102.98318	.000	1442.6688	2034.6312
	Rasio 1:1	-1037.20000 [*]	102.98318	.000	-1333.1812	-741.2188
	Rasio 7:3	-2057.45000 [*]	102.98318	.000	-2353.4312	-1761.4688
	Rasio 9:1	-2549.30000 [*]	102.98318	.000	-2845.2812	-2253.3188
Rasio 1:1	Rasio 1:9	2775.85000 [*]	102.98318	.000	2479.8688	3071.8312
	Rasio 3:7	1037.20000 [*]	102.98318	.000	741.2188	1333.1812
	Rasio 7:3	-1020.25000 [*]	102.98318	.000	-1316.2312	-724.2688
	Rasio 9:1	-1512.10000 [*]	102.98318	.000	-1808.0812	-1216.1188
Rasio 7:3	Rasio 1:9	3796.10000 [*]	102.98318	.000	3500.1188	4092.0812
	Rasio 3:7	2057.45000 [*]	102.98318	.000	1761.4688	2353.4312
	Rasio 1:1	1020.25000 [*]	102.98318	.000	724.2688	1316.2312
	Rasio 9:1	-491.85000 [*]	102.98318	.000	-787.8312	-195.8688
Rasio 9:1	Rasio 1:9	4287.95000 [*]	102.98318	.000	3991.9688	4583.9312
	Rasio 3:7	2549.30000 [*]	102.98318	.000	2253.3188	2845.2812
	Rasio 1:1	1512.10000 [*]	102.98318	.000	1216.1188	1808.0812
	Rasio 7:3	491.85000 [*]	102.98318	.000	195.8688	787.8312

Gambar 4.16 Hasil Uji *Post-Hoc* Graf Kedua

Berdasarkan hasil uji lanjut *Post-Hoc* pada Gambar 4.16, seluruh pasangan rasio menunjukkan nilai signifikansi sebesar 0,000 yang lebih kecil dari 0,05. Dengan demikian dapat disimpulkan bahwa setiap pasangan variasi rasio memiliki perbedaan yang signifikan satu sama lain dalam menghasilkan panjang rute terpendek. Berdasarkan nilai *mean difference* pada hasil uji *Post-Hoc*, terlihat bahwa rasio 1:9 memiliki nilai selisih rata-rata yang bernilai negatif terhadap seluruh rasio pembandingan, yaitu rasio 3:7, 1:1, 7:3, dan 9:1. Nilai negatif ini menunjukkan bahwa rata-rata panjang rute yang dihasilkan oleh rasio 1:9 lebih kecil dibandingkan dengan rasio lainnya. Sebaliknya, rasio dengan nilai *mean difference* positif menunjukkan rata-rata yang lebih besar, sehingga menghasilkan solusi yang kurang optimal.

3. Uji *One Way* ANOVA Data dari Graf Ketiga

ANOVA					
Jarak					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	1047969069	4	261992267.2	918.309	.000
Within Groups	27103375.20	95	285298.686		
Total	1075072444	99			

Gambar 4.17 Hasil Uji ANOVA Graf Ketiga

Berdasarkan hasil uji ANOVA pada Gambar 4.17, diperoleh nilai signifikansi (*Sig.*) sebesar 0,000 yang lebih kecil dari taraf signifikansi 0,05. Dengan demikian, H_0 ditolak dan dapat disimpulkan bahwa terdapat perbedaan rata-rata yang signifikan antar variasi rasio *employed bee* dan *onlooker bee* terhadap panjang rute terpendek yang dihasilkan. Karena hasil uji ANOVA menunjukkan adanya perbedaan signifikan, maka akan dilakukan uji *Post Hoc* dilakukan untuk mengidentifikasi rasio mana yang terbaik.

(I) Rasio	(J) Rasio	Mean Difference (I-J)	Std. Error	Sig.	95% Confidence Interval	
					Lower Bound	Upper Bound
Rasio 1:9	Rasio 3:7	-4152.20000*	168.90787	.000	-4637.6536	-3666.7464
	Rasio 1:1	-6186.00000*	168.90787	.000	-6671.4536	-5700.5464
	Rasio 7:3	-8078.70000*	168.90787	.000	-8564.1536	-7593.2464
	Rasio 9:1	-9134.85000*	168.90787	.000	-9620.3036	-8649.3964
Rasio 3:7	Rasio 1:9	4152.20000*	168.90787	.000	3666.7464	4637.6536
	Rasio 1:1	-2033.80000*	168.90787	.000	-2519.2536	-1548.3464
	Rasio 7:3	-3926.50000*	168.90787	.000	-4411.9536	-3441.0464
	Rasio 9:1	-4982.65000*	168.90787	.000	-5468.1036	-4497.1964
Rasio 1:1	Rasio 1:9	6186.00000*	168.90787	.000	5700.5464	6671.4536
	Rasio 3:7	2033.80000*	168.90787	.000	1548.3464	2519.2536
	Rasio 7:3	-1892.70000*	168.90787	.000	-2378.1536	-1407.2464
	Rasio 9:1	-2948.85000*	168.90787	.000	-3434.3036	-2463.3964
Rasio 7:3	Rasio 1:9	8078.70000*	168.90787	.000	7593.2464	8564.1536
	Rasio 3:7	3926.50000*	168.90787	.000	3441.0464	4411.9536
	Rasio 1:1	1892.70000*	168.90787	.000	1407.2464	2378.1536
	Rasio 9:1	-1056.15000*	168.90787	.000	-1541.6036	-570.6964
Rasio 9:1	Rasio 1:9	9134.85000*	168.90787	.000	8649.3964	9620.3036
	Rasio 3:7	4982.65000*	168.90787	.000	4497.1964	5468.1036
	Rasio 1:1	2948.85000*	168.90787	.000	2463.3964	3434.3036
	Rasio 7:3	1056.15000*	168.90787	.000	570.6964	1541.6036

Gambar 4.18 Hasil Uji *Post-Hoc* Graf Ketiga

Berdasarkan hasil uji lanjut *Post-Hoc* pada Gambar 4.18, seluruh pasangan rasio menunjukkan nilai signifikansi sebesar 0,000 yang lebih kecil dari 0,05. Dengan demikian dapat disimpulkan bahwa setiap pasangan variasi rasio memiliki perbedaan yang signifikan satu sama lain dalam menghasilkan panjang rute terpendek. Berdasarkan nilai *mean difference* pada hasil uji *Post-Hoc*, terlihat bahwa rasio 1:9 memiliki nilai selisih rata-rata yang bernilai negatif terhadap seluruh rasio pembandingan, yaitu rasio 3:7, 1:1, 7:3, dan 9:1. Nilai negatif ini menunjukkan bahwa rata-rata panjang rute yang

dihasilkan oleh rasio 1:9 lebih kecil dibandingkan dengan rasio lainnya. Sebaliknya, rasio dengan nilai *mean difference* positif menunjukkan rata-rata yang lebih besar, sehingga menghasilkan solusi yang kurang optimal.

4.6 Pembahasan

Berdasarkan hasil analisis statistik menggunakan uji *One Way Analysis of Variance* (ANOVA) pada tiga graf komplit berbobot acak dengan banyaknya simpul adalah 50, 100, dan 200, diperoleh nilai signifikansi (*Sig.*) $< 0,05$ pada setiap pengujian. Hal ini menunjukkan bahwa variasi rasio antara *employed bee* dan *onlooker bee* memberikan pengaruh yang signifikan terhadap performa algoritma *Artificial Bee Colony* dalam menyelesaikan permasalahan *Traveling Salesman Problem* (TSP). Secara inferensial, hasil ini menegaskan bahwa perbedaan komposisi populasi dalam algoritma berkontribusi terhadap perbedaan rata-rata kualitas solusi yang dihasilkan, sehingga hipotesis nol (H_0) yang menyatakan tidak adanya perbedaan rata-rata antar kelompok ditolak.

Hasil uji lanjut (*Post Hoc test*) menunjukkan bahwa setiap pasangan variasi rasio memiliki perbedaan yang signifikan (*Sig.* $< 0,05$), sehingga masing-masing konfigurasi rasio memberikan dampak performa yang berbeda secara statistik. Penentuan rasio optimal dilakukan dengan mempertimbangkan nilai rata-rata panjang rute terpendek serta arah *mean difference*, yang menunjukkan perbandingan relatif antar kelompok rasio. Berdasarkan hasil tersebut, rasio 1:9 secara konsisten menghasilkan nilai rata-rata panjang rute paling rendah dibandingkan rasio pembanding lainnya.

Secara teoritis, dominasi banyaknya *onlooker bee* pada rasio tersebut memperkuat mekanisme eksploitasi berbasis probabilistik, yaitu proses pemilihan

dan penyempurnaan solusi terbaik berdasarkan nilai *fitness*. Peningkatan intensitas eksploitasi ini memungkinkan algoritma lebih fokus pada wilayah solusi yang menjanjikan, sehingga meningkatkan peluang untuk memperoleh solusi yang lebih optimal. Dengan demikian, hasil penelitian ini tidak hanya membuktikan adanya pengaruh rasio terhadap performa algoritma, tetapi juga menunjukkan bahwa rasio tertentu, yaitu 1:9, merupakan konfigurasi yang paling efektif dalam meningkatkan kualitas solusi pada algoritma ABC untuk kasus TSP yang diteliti.

4.7 Kajian Penelitian dalam Perspektif Islam

Penelitian mengenai pengaruh rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma *Artificial Bee Colony* (ABC) dalam penyelesaian *Traveling Salesman Problem* (TSP) memiliki keterkaitan erat dengan konsep kesempurnaan penciptaan dan petunjuk Allah SWT yang dijelaskan dalam Al-Qur'an. Salah satu ayat yang sangat relevan dengan perilaku lebah yang menjadi dasar algoritma dalam penelitian ini adalah Surat An-Nahl ayat 68-69 yang menunjukkan bahwa perilaku lebah bukan sekadar hasil insting biologis, melainkan cerminan dari prinsip optimasi alami yang diatur secara sistematis oleh Allah SWT agar makhluk tersebut dapat menemukan solusi paling efektif. Frasa "*tempuhlah jalan Tuhanmu yang telah dimudahkan bagimu*" sangat relevan dengan tujuan penyelesaian TSP, di mana lebah mengajarkan prinsip pencarian rute terpendek dan paling optimal untuk meminimalkan energi serta waktu.

Prinsip dasar algoritma ABC yang digunakan dalam penelitian ini meniru kolaborasi dan pembagian tugas yang teratur dalam koloni lebah madu. Allah menciptakan *employed bee* yang bertugas mengeksplorasi sumber makanan baru dan *onlooker bee* yang bertugas mengeksploitasi sumber makanan terbaik

berdasarkan kualitasnya (*fitness*). Pengaturan rasio antara kedua jenis lebah tersebut memengaruhi dinamika eksplorasi dan eksploitasi dalam pencarian solusi yang optimal. Pengaturan ini mencerminkan prinsip keteraturan tanpa kesia-siaan dalam ciptaan Allah, sebagaimana ditegaskan dalam firman-Nya dalam QS. Al-Anbiya [21]: 16. Ayat tersebut menjelaskan bahwa setiap ciptaan Allah memiliki makna, fungsi, dan efisiensi yang terukur. Dalam konteks komputasi, penyesuaian rasio lebah dilakukan agar proses algoritma berjalan secara efisien dan tidak membuang waktu komputasi, meneladani *sunnatullah* berupa pola keteraturan di alam.

Dalam algoritma ABC, solusi atau rute terus dievaluasi dan diperbarui melalui proses iterasi. Rute yang lebih baik akan menggantikan rute yang lama melalui metode *greedy selection*, sementara rute yang stagnan akan ditinggalkan oleh *scout bee* untuk mencari kemungkinan rute baru. Proses perbaikan solusi secara terus-menerus ini sangat sejalan dengan prinsip ajaran Islam untuk selalu memperbaiki diri dari waktu ke waktu. Konsep ini selaras dengan filosofi iterasi dalam optimasi, di mana sebuah sistem harus selalu mengalami peningkatan *fitness* dari waktu ke waktu untuk menghasilkan rute yang lebih pendek dan efisien. Sama seperti lebah pengintai (*scout bee*) yang mencegah koloni terjebak pada solusi yang buruk atau stagnan, manusia juga diperintahkan untuk terus belajar, mengevaluasi diri, dan tidak berhenti pada pencapaian yang stagnan.

Dengan demikian, penelitian ini tidak hanya berkontribusi dalam bidang matematika terapan dan optimasi, tetapi juga membuktikan bahwa ilmu pengetahuan modern dapat digali dari kaidah-kaidah yang telah Allah tetapkan di alam semesta. Algoritma ABC merupakan bukti nyata betapa luar biasa dan

sempurnanya sistem yang Allah rancang pada makhluk sekecil lebah. Selain itu, algoritma ABC juga menjadi salah satu bentuk implementasi dari upaya manusia dalam meneladani sistem alami yang telah Allah ciptakan secara sempurna, sehingga menghasilkan solusi yang optimal dan bermanfaat dalam kehidupan.

BAB V

PENUTUP

5.1 Kesimpulan

Penelitian ini berfokus pada variasi rasio banyaknya *employed bee* dan *onlooker bee* pada algoritma *Artificial Bee Colony* (ABC) dalam menyelesaikan masalah *Traveling Salesman Problem* (TSP). Berdasarkan rumusan masalah dan hasil serta pembahasan, maka diperoleh kesimpulan sebagai berikut:

1. Eksperimen menggunakan lima variasi rasio, yaitu 1: 9, 3: 7, 1: 1, 7: 3, dan 9: 1.

Pengujian dilakukan secara konsisten pada tiga graf komplit dengan bobot sisi acak yang memiliki jumlah simpul masing-masing 50, 100, dan 200 simpul. Setiap kombinasi graf dan rasio diuji sebanyak 20 kali pengulangan (*run*) untuk menghindari bias stokastik, sehingga diperoleh data rute terpendek yang stabil. Berdasarkan hasil analisis statistik pada ketiga graf, diperoleh nilai signifikansi (*Sig.*) sebesar 0,000, yang mana lebih kecil dari nilai 0,05. Hal ini secara statistik membuktikan bahwa terdapat pengaruh yang signifikan dari variasi rasio banyaknya *employed bee* dan *onlooker bee* terhadap performa algoritma ABC dalam menemukan jarak rute terpendek.

2. Hasil uji lanjut *Post-Hoc* menunjukkan bahwa rasio 1: 9 memiliki selisih rata-rata panjang rute (*mean difference*) yang bernilai negatif terhadap seluruh rasio pembandingan lainnya (3: 7, 1: 1, 7: 3, dan 9: 1). Dengan demikian, disimpulkan bahwa rasio 1: 9 merupakan rasio yang menghasilkan performa paling optimal pada skenario eksperimen penelitian ini karena secara konsisten menghasilkan jarak rute yang lebih pendek. Dominasi jumlah *onlooker bee* pada rasio

tersebut terbukti memperkuat mekanisme eksploitasi, sehingga mampu meningkatkan kemampuan algoritma dalam menemukan solusi terbaik.

5.2 Saran untuk Penelitian Lanjutan

Penelitian selanjutnya disarankan untuk mengujicobakan variasi rasio *employed bee* dan *onlooker bee* pada permasalahan optimasi kombinatorial lain seperti *Vehicle Routing Problem* (VRP), *Job Shop Scheduling*, dan optimasi multi-objektif guna menguji konsistensi pengaruh rasio tersebut pada berbagai domain masalah. Selain itu, penelitian lanjutan juga dapat mengembangkan pendekatan adaptif terhadap rasio *employed bee* dan *onlooker bee*, di mana rasio tersebut tidak bersifat tetap, melainkan berubah secara dinamis mengikuti kondisi pencarian solusi pada setiap iterasi. Penelitian selanjutnya juga dapat mengkombinasikan algoritma ABC dengan strategi *neighborhood operator* yang lebih beragam atau algoritma heuristik lain untuk lebih meningkatkan performanya.

DAFTAR PUSTAKA

- Al Fatih, F., Lubis, N. P., & Nisah, K. (2026). Konsep Homogenitas dan Normalitas dalam Statistik serta Teknik Pengujiannya. *Education Journal*, 1, 805–817.
- Amanah, S. N., Noviani, E., & Yudhi. (2022). Algoritma *Artificial Bee Colony* (ABC) dalam Menyelesaikan *Traveling Salesman Problem* (TSP) Studi Kasus: Data Pelanggan Agen Surat Kabar Di Kota Singkawang. *Buletin Ilmiah Math, Stat, Dan Terapannya (Bimaster)*, 11, 611–620.
- Amri, F., Nababan, E. B., & Syahputra, M. F. (2012). Artificial Bee Colony Algorithm untuk Menyelesaikan Travelling Salesman Problem. *Jurnal Dunia Teknologi Informasi*, 1, 8–13.
- Andriani, H., Utami, E. F., Hardani, & Ustiawaty, J. (2020). *Metode Penelitian: Kualitatif dan Kuantitatif*. CV. Pustaka Ilmu.
- Banharnsakun, A., Achalakul, T., & Sirinaovakul, B. (2011). The best-so-far selection in Artificial Bee Colony algorithm. *Applied Soft Computing*, 11(2), 2888–2901. <https://doi.org/10.1016/j.asoc.2010.11.025>
- Chartrand, G., Lesniak, L., & Zhang, P. (2015). *Graphs & Digraphs* (0 ed.). Chapman and Hall/CRC. <https://doi.org/10.1201/b19731>
- Cheng, S., Shi, Y., Qin, Q., Zhang, Q., & Bai, R. (2014). Population Diversity Maintenance In Brain Storm Optimization Algorithm. *Journal of Artificial Intelligence and Soft Computing Research*, 4(2), 83–97. <https://doi.org/10.1515/jaiscr-2015-0001>
- Karaboga, D. (2005). An Idea Based On Honey Bee Swarm For Numerical Optimization. *Computer Engineering Department Kayseri/Türkiye*.
- Karaboga, D. (2014). A quick artificial bee colony (qABC) algorithm and its performance on optimization problems. *Applied Soft Computing*.
- Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39(3), 459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, 8(1), 687–697. <https://doi.org/10.1016/j.asoc.2007.05.007>
- Kemenag. (2019). *Al-Qur'an dan Terjemahannya Edisi Penyempurnaan 2019, Juz 11-20*. Lajnah Pentashihan Mushaf Al-Qur'an.
- Khan, I., & Maiti, M. K. (2019). A swap sequence based Artificial Bee Colony algorithm for Traveling Salesman Problem. *Swarm and Evolutionary Computation*, 44, 428–438. <https://doi.org/10.1016/j.swevo.2018.05.006>
- Lazuardy, M. A. N., Cholissodin, I., & Furqon, M. T. (2018). Optimasi Rute Multiple Travelling Salesman Problem Pada Distribusi Es Batu Dengan

Algoritme Artificial Bee Colony (ABC). *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, Vol. 2, No. 12, 6261–6268.

- Lin, S., & Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling-Salesman Problem. *Operations Research*, 21(2), 498–516. <https://doi.org/10.1287/opre.21.2.498>
- Qin, Q., Cheng, S., Zhang, Q., Li, L., & Shi, Y. (2015). Artificial Bee Colony Algorithm with Time-Varying Strategy. *Discrete Dynamics in Nature and Society*, 2015, 1–17. <https://doi.org/10.1155/2015/674595>
- Shi, X. H., Liang, Y. C., Lee, H. P., Lu, C., & Wang, Q. X. (2007). Particle swarm optimization-based algorithms for TSP and generalized TSP. *Information Processing Letters*, 103(5), 169–176. <https://doi.org/10.1016/j.ipl.2007.03.010>
- Shihab, Q. (2002). *Tafsir Al-Misbah*. Lentera Hati.
- Talbi, E. (2009). *Metaheuristics: From Design to Implementation* (1st ed.). Wiley. <https://doi.org/10.1002/9780470496916>
- Xiang, W., & An, M. (2013). An efficient and robust artificial bee colony algorithm for numerical optimization. *Computers & Operations Research*, 40(5), 1256–1265. <https://doi.org/10.1016/j.cor.2012.12.006>
- Zhu, G., & Kwong, S. (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization. *Applied Mathematics and Computation*, 217(7), 3166–3173. <https://doi.org/10.1016/j.amc.2010.08.049>

LAMPIRAN

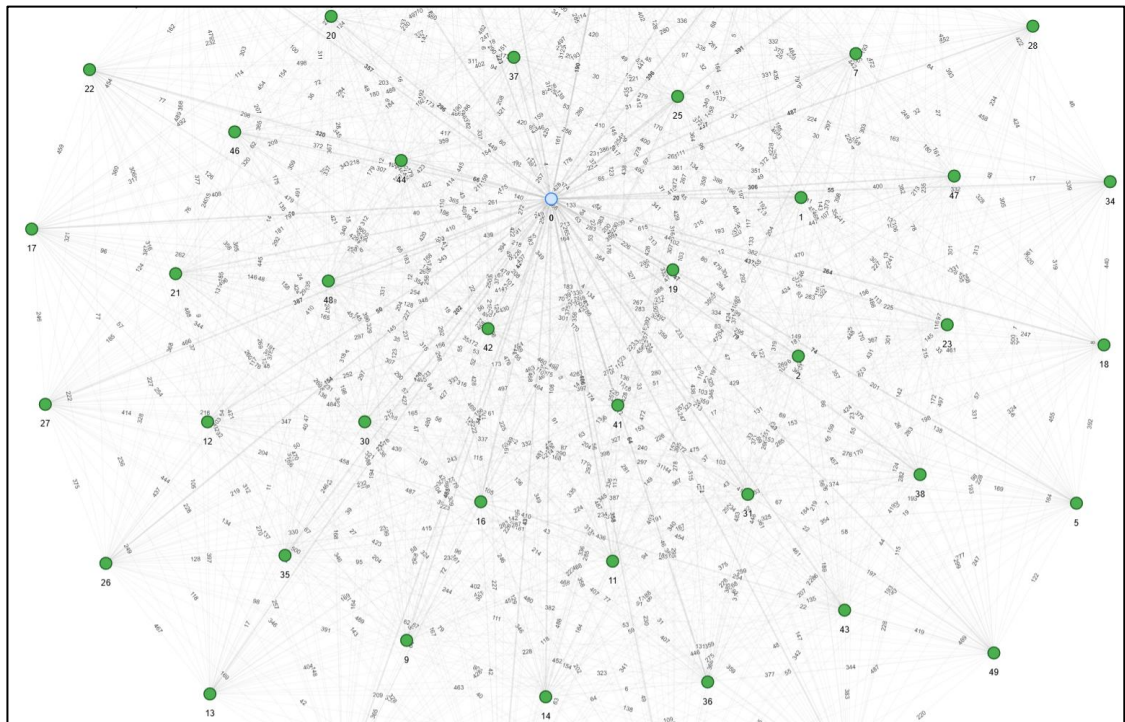
Lampiran 1 Visualisasi Graf Komplit 50 Simpul, 100 Simpul, dan 200 Simpul

Visualisasi Graf Komplit yang digunakan dalam penelitian ini secara lengkap dapat diakses melalui repositori GitHub berikut.



URL:

<https://rizqingk12.github.io/Optimization-ABC-Algorithm-For-TSP/>



Gambar 1. Cuplikan Visualisasi Graf Komplit 50 Simpul

Lampiran 2 *Source Code Python* Implementasi Algoritma ABC

Kode program lengkap penelitian dapat diakses melalui repositori GitHub berikut.



URL:

<https://github.com/rizqingk12/Optimization-ABC-Algorithm-For-TSP>

Lampiran 3 Data Hasil Eksperimen dari Graf 50 Simpul

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
1	0,1	3123	46
2	0,1	3187	44
3	0,1	3077	49
4	0,1	3142	47
5	0,1	3314	50
6	0,1	3352	45
7	0,1	2938	49
8	0,1	2715	49
9	0,1	3240	50
10	0,1	2811	44
11	0,1	2674	50
12	0,1	2822	45
13	0,1	3208	37
14	0,1	3413	50
15	0,1	3219	47
16	0,1	2946	50
17	0,1	2922	47
18	0,1	3098	40
19	0,1	3070	40
20	0,1	2864	50
1	0,3	3576	50
2	0,3	3697	48
3	0,3	3572	42
4	0,3	3582	44
5	0,3	3918	44

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
6	0,3	3850	49
7	0,3	3884	46
8	0,3	3523	50
9	0,3	3442	45
10	0,3	3859	48
11	0,3	3717	36
12	0,3	3873	50
13	0,3	3510	50
14	0,3	3572	47
15	0,3	3801	49
16	0,3	3937	48
17	0,3	3612	47
18	0,3	3665	46
19	0,3	3797	47
20	0,3	3726	41
1	0,5	3940	47
2	0,5	4200	50
3	0,5	3984	46
4	0,5	4355	46
5	0,5	4093	49
6	0,5	4050	50
7	0,5	4403	47
8	0,5	4243	42
9	0,5	3924	49
10	0,5	3999	48
11	0,5	4063	41
12	0,5	3779	47
13	0,5	4040	49
14	0,5	3916	40
15	0,5	4216	45
16	0,5	4035	48
17	0,5	4046	49
18	0,5	3483	44
19	0,5	4163	46
20	0,5	4226	34
1	0,7	4700	49
2	0,7	4307	46
3	0,7	4585	48
4	0,7	4357	45
5	0,7	4588	38
6	0,7	4552	45
7	0,7	4652	47
8	0,7	4336	46

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
9	0,7	4454	47
10	0,7	4324	38
11	0,7	4690	48
12	0,7	4495	48
13	0,7	4740	36
14	0,7	4504	50
15	0,7	4615	50
16	0,7	4472	46
17	0,7	4419	42
18	0,7	4686	50
19	0,7	4636	31
20	0,7	4635	47
1	0,9	5426	48
2	0,9	5001	50
3	0,9	4864	47
4	0,9	4916	47
5	0,9	4982	50
6	0,9	4752	45
7	0,9	4905	48
8	0,9	5134	47
9	0,9	4381	44
10	0,9	4666	49
11	0,9	4772	38
12	0,9	4486	46
13	0,9	4938	47
14	0,9	4619	50
15	0,9	4635	45
16	0,9	5086	45
17	0,9	4762	49
18	0,9	4327	42
19	0,9	4756	49
20	0,9	4723	50

Lampiran 4 Data Hasil Eksperimen dari Graf 100 Simpul

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
1	0,1	8027	47
2	0,1	8189	50
3	0,1	7931	50
4	0,1	7900	50
5	0,1	8671	50
6	0,1	8259	50
7	0,1	7840	50

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
8	0,1	8292	42
9	0,1	8162	49
10	0,1	8006	49
11	0,1	8057	49
12	0,1	7992	50
13	0,1	7982	50
14	0,1	8357	48
15	0,1	8328	50
16	0,1	8238	49
17	0,1	7991	50
18	0,1	8199	48
19	0,1	8145	50
20	0,1	8363	50
1	0,3	9989	50
2	0,3	10423	50
3	0,3	10224	49
4	0,3	10293	49
5	0,3	9516	47
6	0,3	9809	49
7	0,3	9896	49
8	0,3	9845	50
9	0,3	9990	49
10	0,3	10133	49
11	0,3	9703	50
12	0,3	9964	49
13	0,3	10395	49
14	0,3	9799	47
15	0,3	8929	50
16	0,3	10034	50
17	0,3	9207	47
18	0,3	9865	49
19	0,3	9856	47
20	0,3	9832	50
1	0,5	11235	49
2	0,5	11084	50
3	0,5	10795	50
4	0,5	11386	49
5	0,5	11530	49
6	0,5	10966	49
7	0,5	10973	49
8	0,5	11190	48
9	0,5	11526	50
10	0,5	10411	49

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
11	0,5	10244	50
12	0,5	10982	50
13	0,5	10348	47
14	0,5	10636	47
15	0,5	10788	49
16	0,5	11075	50
17	0,5	11060	50
18	0,5	10648	50
19	0,5	10825	43
20	0,5	10744	50
1	0,7	12003	46
2	0,7	11547	49
3	0,7	12462	48
4	0,7	11401	47
5	0,7	11997	49
6	0,7	12488	48
7	0,7	11282	45
8	0,7	12254	44
9	0,7	12159	50
10	0,7	12066	50
11	0,7	12130	49
12	0,7	12127	49
13	0,7	11921	49
14	0,7	12025	49
15	0,7	12135	48
16	0,7	11357	45
17	0,7	12114	39
18	0,7	11517	50
19	0,7	11988	50
20	0,7	11878	50
1	0,9	12599	49
2	0,9	12241	46
3	0,9	12707	50
4	0,9	12352	47
5	0,9	12184	48
6	0,9	12591	48
7	0,9	12102	46
8	0,9	12477	50
9	0,9	12804	50
10	0,9	12381	46
11	0,9	12433	44
12	0,9	12664	49
13	0,9	11997	50

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
14	0,9	12523	49
15	0,9	12976	50
16	0,9	12742	49
17	0,9	11762	49
18	0,9	11928	50
19	0,9	12349	40
20	0,9	12876	50

Lampiran 5 Data Hasil Eksperimen dari Graf 200 Simpul

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
1	0,1	20816	50
2	0,1	21009	50
3	0,1	21288	50
4	0,1	20266	50
5	0,1	19800	50
6	0,1	21204	50
7	0,1	20410	49
8	0,1	21087	50
9	0,1	21176	49
10	0,1	21190	50
11	0,1	20570	50
12	0,1	20841	50
13	0,1	20715	50
14	0,1	21890	47
15	0,1	20531	50
16	0,1	20097	50
17	0,1	19756	50
18	0,1	20716	50
19	0,1	21162	50
20	0,1	20697	50
1	0,3	24004	49
2	0,3	24270	49
3	0,3	24960	48
4	0,3	24704	49
5	0,3	24548	50
6	0,3	25063	46
7	0,3	24879	49
8	0,3	25165	50
9	0,3	25052	50
10	0,3	24751	50
11	0,3	24867	46
12	0,3	25067	50

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
13	0,3	24556	50
14	0,3	25658	50
15	0,3	25480	49
16	0,3	24795	50
17	0,3	25741	49
18	0,3	25027	49
19	0,3	25165	48
20	0,3	24513	50
1	0,5	27127	47
2	0,5	26841	49
3	0,5	27686	50
4	0,5	26623	49
5	0,5	27033	50
6	0,5	26403	44
7	0,5	27537	49
8	0,5	26396	49
9	0,5	26912	47
10	0,5	26270	46
11	0,5	26591	50
12	0,5	26437	49
13	0,5	26768	49
14	0,5	27964	49
15	0,5	27398	50
16	0,5	27708	48
17	0,5	27161	50
18	0,5	25869	50
19	0,5	26984	49
20	0,5	27233	48
1	0,7	28621	48
2	0,7	29274	47
3	0,7	29499	50
4	0,7	28368	49
5	0,7	28331	49
6	0,7	27953	50
7	0,7	28760	49
8	0,7	28200	50
9	0,7	29522	48
10	0,7	28388	48
11	0,7	28516	50
12	0,7	29001	46
13	0,7	29133	48
14	0,7	29372	50
15	0,7	29258	48

Percobaan ke-	Rasio	Jarak Rute Terpendek	Berhenti pada Iterasi ke-
16	0,7	29085	47
17	0,7	28489	50
18	0,7	28825	50
19	0,7	29274	50
20	0,7	28926	50
1	0,9	29896	50
2	0,9	29979	50
3	0,9	29082	50
4	0,9	31189	50
5	0,9	30080	50
6	0,9	30839	47
7	0,9	30933	48
8	0,9	30641	48
9	0,9	29265	50
10	0,9	29588	49
11	0,9	29991	50
12	0,9	28949	48
13	0,9	29702	49
14	0,9	29671	50
15	0,9	29287	50
16	0,9	29363	50
17	0,9	30637	48
18	0,9	29541	49
19	0,9	29031	46
20	0,9	30254	50

RIWAYAT HIDUP



Yuniar Rizqi lahir di Kabupaten Nganjuk pada tanggal 12 Juni 2004. Penulis merupakan anak keempat dari empat bersaudara dari pasangan suami istri Bapak Suparji dan Ibu Sri Purwaningsih. Pendidikan formal penulis dimulai di TK Al-Hidayah Termas dan diselesaikan pada tahun 2010. Setelah itu, penulis melanjutkan pendidikan di MIN 5 Nganjuk dan lulus pada tahun 2016. Pendidikan jenjang menengah pertama ditempuh di MTsN 4 Nganjuk dan lulus pada tahun 2019. Selanjutnya, penulis melanjutkan pendidikan ke jenjang menengah atas di SMA Negeri 1 Kertosono dan lulus pada pertengahan tahun 2022. Setelah itu, penulis melanjutkan pendidikan ke jenjang perguruan tinggi di Universitas Islam Negeri Maulana Malik Ibrahim Malang dengan memilih Program Studi S1 Matematika di Fakultas Sains dan Teknologi. Selama menjalani masa perkuliahan, penulis aktif dalam berbagai kegiatan, baik akademik maupun non-akademik. Penulis aktif mengikuti pelatihan dan seminar ilmiah. Selain itu, penulis juga aktif mengikuti perlombaan non-akademik yaitu bulutangkis dari tingkat fakultas sampai universitas. Penulis merupakan *awardee* Beasiswa Bank Indonesia periode tahun 2025 yang berperan aktif dalam mengikuti program dari komunitas Generasi Baru Indonesia (GenBI) Malang. Penulis juga mendapatkan pelatihan *capacity building* dari Bank Indonesia di antaranya yaitu sertifikasi BNSP *Social Media Marketing*, *leadership*, manajemen keuangan, dan lain sebagainya. Penulis juga memiliki pengalaman Praktik Kerja Lapangan (PKL) di PERUMDA Air Minum Tugu Tirta Kota Malang tepatnya di bagian *Non-Revenue Water* (NRW) di mana penulis bekerja untuk mengatur jadwal pengecekan Aset PDAM yang tersebar di seluruh wilayah Kota Malang. Selain pendidikan formal, penulis juga mengikuti pendidikan non-formal yaitu kursus bahasa Inggris di Kampung Inggris *World English* (WE) Kediri pada tahun 2023 dan lulus mendapatkan nilai yang baik.



KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI
Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341)558933

BUKTI KONSULTASI SKRIPSI

Nama : Yuniar Rizqi
NIM : 220601110021
Fakultas / Jurusan : Sains dan Teknologi / Matematika
Judul Skripsi : Pengaruh Rasio Banyaknya *Employed Bee* dan *Onlooker Bee* terhadap Performa Algoritma *Artificial Bee Colony* pada Penyelesaian *Traveling Salesman Problem*
Pembimbing I : Mohammad Nafie Jauhari, M.Si.
Pembimbing II : Abdul Aziz, M.Si.

No	Tanggal	Hal	Tanda Tangan
1.	9 September 2025	Konsultasi Bab I, II, dan III	1.
2.	16 September 2025	Konsultasi Bab I, II, dan III	2.
3.	30 September 2025	Konsultasi Bab I, II, dan III	3.
4.	13 Oktober 2025	Konsultasi Kajian Agama Bab I dan II	4.
5.	14 Oktober 2025	Konsultasi Bab I, II, dan III	5.
6.	17 November 2025	Konsultasi Kajian Agama Bab I dan II	6.
7.	21 November 2025	Konsultasi Bab I, II, dan III	7.
8.	11 Desember 2025	ACC Bab I, II, dan III	8.
9.	12 Desember 2025	ACC Kajian Agama Bab I dan II	9.
10.	14 Januari 2026	ACC Seminar Proposal	10.
11.	21 Januari 2026	Konsultasi Revisi Seminar Proposal	11.
12.	18 Februari 2026	Konsultasi Bab IV dan V	12.
13.	10 Maret 2026	Konsultasi Bab IV dan V	13.
14.	1 April 2026	Konsultasi Bab IV dan V	14.
15.	2 April 2026	Konsultasi Kajian Agama Bab IV	15.



**KEMENTERIAN AGAMA RI
UNIVERSITAS ISLAM NEGERI
MAULANA MALIK IBRAHIM MALANG
FAKULTAS SAINS DAN TEKNOLOGI**

Jl. Gajayana No.50 Dinoyo Malang Telp. / Fax. (0341) 58933

16.	16 April 2026	ACC Kajian Agama Bab IV	16.
17.	16 April 2026	ACC Bab IV dan V	17.
18.	30 April 2026	Seminar Hasil	18.
19.	4 Mei 2026	Konsultasi Revisi Seminar Hasil	19.
20.	18 Mei 2026	Konsultasi Revisi Seminar Hasil Kajian Agama Bab IV	20.
21.	5 Juni 2026	ACC Sidang Skripsi	21.

Malang, 5 Juni 2026

Yang bertanda,

Ketua Program Studi Matematika



Dr. Achnur Rozi, M.Si.

NIP. 19800527 200801 1 012