



TUGAS AKHIR - KS184822

**OPTIMASI BIAYA DAN WAKTU PROYEK
PEMBANGUNAN JEMBATAN MENGGUNAKAN
*ARTIFICIAL BEE COLONY***

**MIRZA ROMI SETIAWAN
NRP 062116 4000 7001**

**Dosen Pembimbing
Dra. Wiwiek Setya Winahju, M.S.
Dr. Dra. Kartika Fithriasari, M.Si.**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020**



TUGAS AKHIR - KS184822

**OPTIMASI BIAYA DAN WAKTU PROYEK
PEMBANGUNAN JEMBRAN MENGGUNAKAN DAN
*ARTIFICIAL BEE COLONY***

**MIRZA ROMI SETIAWAN
NRP 062116 4000 7001**

**Dosen Pembimbing
Dra. Wiwiek Setya Winahju, M.S.
Dr. Dra. Kartika Fithriasari, M.Si.**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS SAINS DAN ANALITIKA DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020**



FINAL PROJECT - KS184822

BRIDGE CONSTRUCTION COST AND TIME OPTIMIZATION USING ARTIFICIAL BEE COLONY

**MIRZA ROMI SETIAWAN
NRP 062116 4000 7001**

Supervisors

Dra. Wiwiek Setya Winahju, M.S.

Dr. Dra. Kartika Fithriasari, M.Si.

**UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF SCIENCE AND DATA ANALYTICS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2020**

LEMBAR PENGESAHAN

**OPTIMASI BIAYA DAN WAKTU PROYEK
PEMBANGUNAN JEMBATAN MENGGUNAKAN DAN
ARTIFICIAL BEE COLONY**

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Statistika
pada
Program Studi Sarjana Departemen Statistika
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember

Oleh :
MIRZA ROMI SETIAWAN
NRP. 062116 4000 7001

Disetujui oleh Pembimbing:

Dra. Wiwiek Setya Winahju, M.S.
NIP. 19560424 198303 2 001

Dr. Dra. Kartika Fithriasari, M.Si.
NIP. 19691212 199303 2 002

(*Wiwiek Setya Winahju*)
(*Kartika Fithriasari*)



Mengetahui,
Kepala Departemen

Dr. Dra. Kartika Fithriasari, M.Si.
NIP. 19691212 199303 2 002

SURABAYA, 19 DESEMBER 2020

(Halaman ini sengaja dikosongkan)

OPTIMASI BIAYA DAN WAKTU PROYEK PEMBANGUNAN JEMBATAN MENGGUNAKAN *ARTIFICIAL BEE COLONY*

Nama Mahasiswa : Mirza Romi Setiawan
NRP : 062116 4000 7001
Departemen : Statistika
Dosen Pembimbing : Dra. Wiwiek Setya Winahju, M.S.
Dr. Dra. Kartika Fithriasari, M.Si.

Abstrak

Time-Cost Trade-Off (TCTO) adalah permasalahan penting pada manajemen proyek karena jika tidak diselesaikan dengan baik maka proyek yang dikerjakan akan memiliki biaya yang mahal dan waktu yang panjang. Oleh karena itu permasalahan optimasi waktu dan biaya ini sangat penting bagi suatu proyek agar proyek yang akan dikerjakan memiliki waktu yang relatif singkat dengan biaya yang tidak mahal. Terdapat banyak jenis metode untuk mengoptimasi waktu dan biaya pada sebuah proyek, salah tiganya yaitu metode konvensional, heuristic dan metaheuristic. Pada penelitian ini peneliti akan meneliti mengenai optimasi waktu dan biaya pada setiap item pekerjaan yang terdapat pada proyek pembangunan jembatan dengan meminimalkan waktu dan biaya dengan metode Artificial Bee Colony. Hasil pada penelitian ini akan dibandingkan dengan hasil dari penelitian sebelumnya dimana metode yang digunakan adalah Constraint Programming dan Genetic Algorithm. Hasil yang didapatkan adalah metode Genetic Algorithm menghasilkan nilai cost slope yang sama dengan lama waktu mendapatkan solusi lebih lambat daripada metode Artificial Bee Colony.

Kata Kunci: *Artificial Bee Colony, Biaya, Cost slope, Durasi, Penambahan Pekerja*

(Halaman ini sengaja dikosongkan)

BRIDGE CONSTRUCTION TIME AND COST OPTIMIZATION USING ARTIFICIAL BEE COLONY

Name : Mirza Romi Setiawan
Student Number : 062116 4000 7001
Department : Statistics
Supervisors : Dra. Wiwiek Setya Winahju, M.S.
Dr. Dra. Kartika Fithriasari, M.Si.

Abstract

Time-Cost Trade-Off (TCTO) is an important problem in project management because if it is not completed properly, the project will be expensive and time-consuming. Therefore, the problem of time and cost optimization is very important for a project so that the project to be done has a relatively short time and is not expensive. There are many types of methods to optimize time and cost on a project, one of them is conventional, heuristic, and metaheuristic methods. In this study, researchers will examine the optimization of time and costs for each activity contained in the bridge construction project by minimizing time and costs with the Artificial Bee Colony methods. The results of this study will be compared with the results of previous research where the methods used are Constraint Programming and Genetic Algorithm. The results obtained from both methods are that the Genetic Algorithm method produces the same cost slope value and the time to get the solution is slower.

Keywords: *Additional Workers, Artificial Bee Colony, Cost, Cost slope, Duration*

(Halaman ini sengaja dikosongkan)

KATA PENGANTAR

Puji syukur kehadiran Tuhan Yang Maha Esa, atas segala berkat dan karunia-Nya, sehingga penulis dapat menyelesaikan laporan Tugas Akhir yang berjudul **“Optimasi Biaya dan Waktu Proyek Pembangunan Jembatan Menggunakan *Artificial Bee Colony*”** dengan lancar.

Penulis menyadari bahwa Tugas Akhir ini dapat terselesaikan tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Dr. Dra. Kartika Fithriasari, M.Si. selaku Kepala Departemen Statistika dan dosen pembimbing Tugas Akhir yang senantiasa memberikan perhatian, bimbingan, dan pengarahan selama menyelesaikan Tugas Akhir ini.
2. Dr. Santi Wulan Purnami, S.Si, M.Si. selaku Sekretaris I Departemen Statistika FSAD ITS.
3. Dr. Sutikno, S. Si., M. Si. selaku dosen wali selama masa studi yang telah banyak memberikan saran dan arahan dalam proses belajar di Departemen Statistika.
4. Dra. Wiwiek Setya Winahju, M.S. dan Dr. Dra. Kartika Fithriasari, M.Si. selaku dosen pembimbing yang telah banyak memberi masukan kepada penulis dalam penyelesaian Tugas Akhir.
5. Prof. Drs. Nur Iriawan, M.Ikom., Ph.D. dan Dr. Irhamah S.Si., M.Si selaku dosen penguji yang selalu sabar dalam mengomentari serta memberikan masukan dan saran dalam penyelesaian Tugas Akhir penulis.
6. Orang tua dan keluarga yang selalu mendoakan dan memberikan motivasi sehingga penulis dapat menyelesaikan Tugas Akhir ini.
7. Zony Zulfadli, S.T., M.T. yang telah memberikan kesempatan dan kepercayaan kepada penulis untuk mengolah dan menganalisis data Rencana Anggaran Biaya proyek pembangunan jembatan di CV. Cremona Teknik Consultant.

8. Seluruh dosen dan *staff* pengajar Program Studi Sarjana Departemen Statistika FSAD ITS yang telah membantu penulis selama proses perkuliahan.
9. Seluruh pihak yang telah membantu dalam penyelesaian Tugas Akhir ini yang tidak bisa penulis sebutkan satu persatu.

Terlepas dari itu, penulis menyadari bahwa masih banyak kekurangan pada laporan ini baik dari segi penulisan maupun isi. Oleh karena itu, dengan terbuka kami menerima segala kritik dan saran dari pembaca sebagai bahan perbaikan bagi penulisan selanjutnya. Besar harapan penulis untuk mendapatkan kritik dan saran yang membangun sehingga Tugas Akhir ini dapat memberikan manfaat bagi semua.

Surabaya, Agustus 2020

Penulis

DAFTAR ISI

	Halaman
LEMBAR PENGESAHAN	iii
Abstrak.....	v
Abstract.....	vii
KATA PENGANTAR.....	ix
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
DAFTAR LAMPIRAN	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	5
1.2 Tujuan.....	5
1.4 Manfaat.....	6
1.5 Batasan Masalah.....	6
BAB II TINJAUAN PUSTAKA.....	7
2.1 Optimasi	7
2.2 <i>Artificial Bee Colony</i> (ABC)	11
2.3 Penjadwalan.....	22
2.4 Manajemen Proyek.....	23
2.5 Time Cost Trade-Off Problem.....	23
BAB III METODOLOGI PENELITIAN	25
3.1 Sumber Data	25
3.2 Variabel Penelitian dan Struktur	26
3.3 Langkah Analisis	27
BAB IV ANALISIS DAN PEMBAHASAN	31
4.1 Optimasi dengan <i>Artificial Bee Colony</i>	31
4.1.1 Inialisasi.....	31
4.1.2 <i>Employed Bee Phase</i>	32
4.1.3 <i>Onlooker Bee Phase</i>	33
4.1.4 <i>Scout Bee Phase</i>	35
4.1.5 Hasil Optimasi <i>Artificial Bee Colony</i>	36
4.2 Perbandingan Nilai <i>Cost Slope</i> dan Lama Waktu Untuk Mendapatkan Solusi	39

BAB V KESIMPULAN DAN SARAN	41
5.1 Kesimpulan.....	41
5.2 Saran	41
DAFTAR PUSTAKA	42
LAMPIRAN.....	47
BIODATA PENULIS.....	65

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Diagram Alir Algoritma ABC	16
Gambar 2.2 Diagram alir algoritma ABC	17
Gambar 2.3 Diagram alir algoritma ABC	18
Gambar 2.4 Ilustrasi nilai <i>fitness</i> yang sudah konvergen	22
Gambar 3.1 Diagram Alir Penelitian	30

(Halaman ini sengaja dikosongkan)

DAFTAR TABEL

	Halaman
Tabel 3.1 Item pekerjaan Pembangunan Proyek Jembatan	25
Tabel 3.2 Jenis pekerja yang mengerjakan proyek jembatan pada Kecamatan Samboja Kabupaten Kutai Kartanegara	25
Tabel 3.3 Variabel Penelitian	26
Tabel 3.4 Struktur Data	27
Tabel 4.1 Struktur Populasi	31
Tabel 4.2 Hasil dari fase <i>employed bee</i>	33
Tabel 4.3 Nilai probabilitas masing-masing <i>food source</i>	34
Tabel 4.4 Hasil dari fase <i>onlooker bee</i>	35
Tabel 4.5 Hasil dari fase <i>scout bee</i>	35
Tabel 4.6 Kombinasi Terbaik dengan <i>Artificial Bee Colony</i>	37
Tabel 4.7 Jumlah Pekerja Setelah Dilakukan Optimasi menggunakan <i>Artificial Bee Colony</i>	37
Tabel 4.8 Perbandingan <i>cost slope</i> dan lama waktu yang dibutuhkan metode ABC dan GA	39

(Halaman ini sengaja dikosongkan)

DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Perhitungan Nilai <i>Cost Slope</i>	47
Lampiran 2. Program <i>Python</i> untuk Menyusun Struktur Data.	48
Lampiran 3. Program <i>Python</i> untuk Artificial Bee Colony.....	51
Lampiran 4. Grafik Fitness Terbaik setiap food source pada 10 kali Percobaan Metode <i>Artificial Bee Colony</i>	60
Lampiran 5. Catatan Lama Waktu <i>Running</i> pada 10 kali Percobaan Metode <i>Artificial Bee Colony</i>	62
Lampiran 6. Surat Keterangan Pengambilan Data	63

(Halaman ini sengaja dikosongkan)

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pembangunan infrastruktur merupakan salah satu program yang digencarkan oleh presiden dalam kurun waktu 5 tahun terakhir. Pembangunan infrastruktur penting dilakukan karena akan meningkatkan konektivitas dan mendorong daya saing antardaerah di seluruh Indonesia. Basuki Hadimuljono mengatakan bahwa pembangunan infrastruktur yang telah dilakukan turut berkontribusi pada pertumbuhan ekonomi Indonesia serta peningkatan daya saing nasional (Hadimuljono, 2018). Keberhasilan pembangunan infrastruktur dapat ditinjau dari lama waktu pembangunan. Semakin lama waktu pembangunan infrastruktur dengan catatan melebihi waktu yang sudah direncanakan, maka dapat dipastikan bahwa terdapat permasalahan dalam pembangunan infrastruktur tersebut.

Berjalannya pembangunan infrastruktur dalam kurun waktu 5 tahun terakhir dipengaruhi oleh beberapa faktor. Faktor pertama adalah perencanaan pembangunan. Penelitian yang dilakukan oleh (Lestari, 2015) yang melakukan penelitian pada tahun 2015 tentang faktor-faktor yang mempengaruhi pembangunan infrastruktur (jalan dan jembatan) di kecamatan kampar kiri hulu kabupaten kampar tahun 2011-2013 mengungkapkan bahwa Perencanaan merupakan proses awal yang sangat penting. Karena jika pada tahap ini tidak berjalan dengan baik, maka pelaksanaannya juga tidak akan berjalan dengan baik. Faktor yang kedua adalah pendanaan proyek. Dikutip dari Komite Percepatan Penyediaan Infrastruktur Prioritas (KPPIP, 2019) menyatakan bahwa pada tahap penyiapan, terdapat masalah akibat lemahnya kualitas penyiapan proyek dan keterbatasan alokasi pendanaan. Oleh karena itu permasalahan mengenai penjadwalan merupakan permasalahan yang penting untuk diselesaikan agar pembangunan dapat berjalan dengan cepat sehingga dapat membantu perekonomian Indonesia.

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan yang secara keseluruhan harus dilakukan pada sumber daya yang terbatas, serta pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada. Penjadwalan pada umumnya dapat diterapkan pada banyak bidang seperti industri, akademik, komunikasi, perencanaan produksi, dan manajemen proyek. Penjadwalan dilakukan dengan tujuan agar dapat mengetahui waktu mulai dan selesai proyek, merencanakan kebutuhan proyek seperti material, peralatan, dan tenaga kerja.

Manajemen proyek adalah proses merencanakan, mengorganisir, memimpin, dan mengendalikan sumber daya perusahaan untuk mencapai sasaran jangka pendek yang telah ditentukan (Rani, 2016). Manajemen proyek tumbuh karena dorongan mencari pendekatan pengelolaan yang sesuai dengan tuntutan dan sifat kegiatan proyek, suatu kegiatan yang dinamis dan berbeda dengan kegiatan operasional rutin. Manajemen proyek dapat membantu untuk mengontrol kesesuaian jadwal yang sudah direncanakan agar penyelesaian proyek tidak memakan waktu yang lama. Manajemen proyek juga dapat mengoptimalkan biaya dan waktu yang digunakan pada proyek tersebut.

Time Cost Trade Off (TCTO) adalah salah satu teknik yang dapat digunakan dalam manajemen proyek untuk menyelesaikan permasalahan optimasi biaya dan waktu. Tujuan utama dari perencanaan proyek adalah untuk mendapatkan biaya minimum dari proyek tersebut dengan durasi penyelesaian proyek yang sesuai (Zaid dkk, 2017). Penelitian mengenai *Time Cost Trade Off Problem* pernah dilakukan oleh (Priyo & Auliya, 2015) yang membahas tentang optimalisasi waktu dan biaya proyek pada pelaksanaan proyek Gedung Indonesia dengan metode penambahan jam kerja dan penambahan tenaga kerja. Setelah dilakukan pengolahan data, hasil yang didapat adalah penambahan jam kerja merupakan opsi terbaik karena dapat menghasilkan efisiensi waktu dan biaya paling tinggi dengan efisiensi waktu proyek sebanyak 24 hari (9,02%) dan efisiensi biaya proyek sebesar Rp. 43.019.556,39

(0,41%). Penelitian selanjutnya dilakukan (Izzah, 2017) yang membahas tentang analisa pertukaran waktu dan biaya menggunakan metode *time cost trade off* (TCTO) pada proyek pembangunan perumahan di PT. X. Tujuan penelitian tersebut adalah mendeskripsikan aktivitas jaringan kerja yang optimal menggunakan metode *Critical Path Method* (CPM), perencanaan kelayakan yang optimal dengan metode *Program Evaluation and Review Technique* (PERT), dan mendeskripsikan waktu dan biaya proyek yang efisien menggunakan metode *Time Cost Trade Off* (TCTO) pada proses pembangunan perumahan. Hasil dari penelitian tersebut adalah efisiensi waktu pengerjaan proyek dengan penambahan pekerja adalah 5,76% dengan selisih percepatan 32 hari kerja dan efisiensi biayanya adalah 0,156% dengan biaya optimal sebesar Rp. 6.753.245.793. Berdasarkan penelitian-penelitian yang telah disebutkan dapat disimpulkan bahwa permasalahan TCTO apabila dapat diselesaikan akan menyebabkan perencanaan pembangunan menjadi lebih cepat dan lebih murah.

Terdapat banyak jenis metode untuk menyelesaikan permasalahan TCTO, salah satunya yaitu metode konvensional, *heuristic* dan *metaheuristic*. Metode konvensional adalah algoritma yang menggunakan perhitungan matematis biasa. Metode *heuristic* adalah sub bidang dari kecerdasan buatan yang digunakan untuk melakukan pencarian dan penentuan jalur terpendek (Mutahkiroh dkk, 2007). Metode *metaheuristic* merupakan metode yang memiliki skema pencarian solusi yang terinspirasi dari prinsip-prinsip alamiah yang dikembangkan oleh makhluk hidup, antara lain: evolusi dan seleksi alam yang diadopsi oleh *Genetic Algorithm*, interaksi sosial dari sekawanan ikan maupun burung yang ditiru oleh *Particle Swarm Optimization* (PSO), maupun interaksi koloni semut untuk menemukan sumber makanan yang menginspirasi *Ant Colony Optimization* (ACO) (Prayogo dkk, 2017).

Peneliti ingin menggunakan algoritma *Artificial Bee Colony* (ABC) karena metode tersebut dapat menyelesaikan permasalahan TCTO dengan cepat dan menghasilkan nilai yang mendekati nilai

optimum atau bahkan menghasilkan nilai yang optimum. Algoritma *Artificial Bee Colony* (ABC) adalah suatu algoritma yang berdasarkan pada kecerdasan sekawanan serangga yang memiliki kecerdasan dan sangat populer serta efisien yang terinspirasi dari perilaku lebah madu ketika mencari makan. (Kocen & Akca, 2014).

Penelitian *Artificial Bee Colony* (ABC) pernah dilakukan oleh (Muthiah & Rajkumar, 2014) mengenai sebuah perbandingan antara ABC dengan GA untuk meminimumkan nilai *makespan* untuk permasalahan *Job Shop Scheduling* dimana terdapat 5 pekerjaan dan pada setiap pekerjaan memiliki 8 tugas untuk diselesaikan. Setelah dilakukan analisis, hasil yang didapatkan adalah algoritma ABC mencapai hasil yang lebih baik dibandingkan dengan GA. Penelitian selanjutnya dilakukan oleh (Karaboga & Akay, 2004) mengenai sebuah studi perbandingan dari *Artificial Bee Colony* dengan beberapa algoritma yang lain. Penelitian tersebut merupakan studi yang lengkap mengenai perbandingan algoritma ABC dengan algoritma-algoritma evolusi seperti *Genetic Algorithm* (GA), *Differential Evolution Algorithm* (DE), dan *Evolution Strategies* (ES) dan algoritma yang berbasis kawanan serangga seperti *Particle Swarm Optimization* (PSO) pada kasus optimasi kumpulan fungsi tes numerik yang besar. Setelah semua algoritma dijalankan didapatkan algoritma ABC sebagai algoritma yang terbaik dibandingkan dengan algoritma GA, PSO, DE, dan ES.

Pada penelitian ini peneliti akan meneliti mengenai waktu dan biaya yang optimum pada setiap item pekerjaan yang terdapat pada proyek pembangunan jembatan yang terletak pada Kecamatan Samboja Kabupaten Kutai Kartanegara dengan penambahan jumlah pekerja pada setiap item pekerjaan, karena semakin banyak jumlah pekerja yang mengerjakan proyek tersebut, maka akan semakin cepat selesai pula proyek tersebut. Data yang digunakan pada penelitian ini adalah data yang sama seperti yang digunakan oleh penelitian (Nafis, 2020). Item pekerjaan yang terdapat pada

proyek pembangunan jembatan ini memiliki jumlah sebanyak 15 item pekerjaan kritis. Item pekerjaan kritis yang dimaksud adalah item pekerjaan yang mempengaruhi proyek secara signifikan ketika terjadi perubahan perencanaan pada item pekerjaan tersebut yang didapatkan dengan menggunakan *Artificial Bee Colony* (ABC) dengan parameter *cost clope* dan kecepatan algoritma.

1.2 Rumusan Masalah

Permasalahan TCTO adalah permasalahan penting pada manajemen proyek karena jika tidak diselesaikan dengan baik maka proyek yang dikerjakan akan memiliki biaya yang mahal dan waktu yang panjang. Salah satu permasalahan pada permasalahan TCTO adalah kombinasi dari seluruh opsi yang banyak, sehingga dibutuhkan waktu yang cukup panjang untuk dapat menyelesaikannya secara manual. Metode *Artificial Bee Colony* dapat menentukan kombinasi antara persentase penambahan pekerja dengan item pekerjaan yang paling baik yang selanjutnya akan menghasilkan biaya dan waktu yang paling optimum dengan waktu yang relatif singkat. Metode *Artificial Bee Colony* akan digunakan untuk mengoptimasi biaya dan waktu pada proses pembangunan jembatan yang terletak pada Kecamatan Samboja Kabupaten Kutai Kartanegara sehingga menghasilkan biaya dan waktu yang paling minimum yang selanjutnya hasil dari algoritma tersebut dibandingkan dengan penelitian Nafis dengan nilai *cost slope* dan lama waktu mendapatkan solusi sebagai parameter.

1.3 Tujuan

Adapun tujuan yang ingin dicapai pada penelitian ini adalah sebagai berikut.

1. Mendapatkan hasil berupa persentase penambahan pekerja terbaik untuk seluruh item pekerjaan menggunakan *Artificial Bee Colony*.
2. Mendapatkan metode yang lebih efisien antara *Artificial Bee Colony* dan *Genetic Algorithm*.

1.4 Manfaat

Hasil dari penelitian ini diharapkan dapat memberikan gambaran mengenai optimasi peningkatan kinerja biaya dan waktu dari proyek bangunan sebuah jembatan, sehingga proyek sejenis di masa depan dapat direncanakan dan dikontrol dengan lebih baik serta memiliki durasi pengerjaan yang singkat dan biaya yang tidak mahal. Selain itu, penelitian ini juga dapat digunakan sebagai informasi atau wawasan penggunaan metode *Artificial Bee Colony* terkait optimasi biaya dan waktu dari proyek bangunan sebuah jembatan.

1.5 Batasan Masalah

Batasan masalah dalam penelitian ini adalah jembatan yang akan dibangun ini adalah jembatan yang terletak pada Kecamatan Samboja Kabupaten Kutai Kartanegara. Item pekerjaan yang digunakan pada penelitian ini hanyalah item pekerjaan yang bersifat kritis. Biaya yang digunakan pada penelitian ini hanya merupakan biaya langsung. Alokasi dana diasumsikan tidak memiliki kendala dengan berapapun biaya yang dibutuhkan. Pekerja yang mengerjakan proyek diasumsikan tidak mengalami kendala apapun sehingga proyek dapat dijalankan sesuai waktu yang direncanakan. Keterbatasan distribusi pekerja diasumsikan proporsional sesuai kebutuhan harian dari suatu item pekerjaan. Persentase penanaman pekerja dibatas dari 0% sampai 40%.

BAB II

TINJAUAN PUSTAKA

2.1 Optimasi

Optimasi adalah suatu aktivitas yang dilakukan untuk menghasilkan hasil yang terbaik dari seluruh opsi yang memungkinkan (Sugiyono, 2006). Tujuan dari optimasi adalah untuk meminimumkan usaha atau biaya dan memaksimalkan keuntungan yang diperoleh. Hal yang akan dioptimasi berupa suatu fungsi dengan variabel-variabel yang akan dihitung nilai optimumnya.

Fungsi tujuan pada penelitian ini adalah meminimumkan *cost slope* berdasarkan persentase penambahan pekerja pada setiap item pekerjaan yang ada. *cost slope* adalah perbandingan antara pertambahan biaya dan percepatan waktu penyelesaian proyek yang dihitung dari hasil pengurangan antara biaya *crashing* (*Crash Cost*) dengan biaya normal proyek (*Normal Cost*) kemudian dibagi dengan hasil pengurangan antara durasi normal dengan durasi percepatan (*Crash Duration*) (Muhammad & Indriyani, 2015). Rumus *Cost Slope* ditunjukkan seperti berikut.

$$CS = \frac{b_1 - b_0}{d_0 - d_1} \quad (2.1)$$

dimana :

- b_0 : Biaya awal (Rp)
- b_1 : Biaya setelah penambahan pekerja (Rp)
- CS : *Cost slope*
- d_0 : Durasi awal (Hari)
- d_1 : Durasi setelah penambahan pekerja (Hari)

Fungsi tujuan pada penelitian ini, adalah meminimumkan *cost slope*, dinotasikan sebagai $f(x_1, x_2, \dots, x_N)$, dimana fungsi tersebut adalah sebagai berikut.

$$f(x_1, x_2, \dots, x_{15}) = CS \quad (2.2)$$

- CS : *Cost slope* (Rp)
 x : Persentase penambahan pekerja pada item pekerjaan (%)

dengan N adalah jumlah item pekerjaan, x_i adalah persentase penambahan pekerja pada item pekerjaan ke- i ($i=1,2,\dots,N$), dan CS adalah nilai *Cost Slope* seperti pada Persamaan (2.1), yang perhitungannya dijabarkan oleh (Nafis, 2020) pada Lampiran 1. Adapun fungsi batasan yang harus dipenuhi pada penelitian ini yaitu,

$$d_0 - d_1 \geq 30 \quad (2.3)$$

$$0 \leq x_i \leq 0,4 \quad (2.4)$$

- d_0 : Durasi awal sebelum ditambah pekerja (hari)
 d_1 : Durasi akhir setelah ditambahkan pekerja (hari)
 x_i : Persentase penambahan pekerja pada item pekerjaan ke- i (Rp)

Nilai durasi awal (d_0) dari satu proyek tersebut yang sudah ditentukan pada data. Pada kasus ini urutan pekerjaan berbentuk kelompok urutan setiap item pekerjaan memiliki nomor urutan pekerjaan. Item pekerjaan pada kelompok urutan setelahnya hanya dapat dilaksanakan setelah semua item pekerjaan pada kelompok urutan sebelumnya diselesaikan. Sehingga program yang dibuat dalam penelitian ini akan disesuaikan untuk kasus serupa. Oleh karena itu, untuk menghitung durasi awal total adalah berdasarkan jumlah dari maksimal durasi pada masing-masing kelompok. Selisih durasi dari satu proyek sebelum dan sesudah ditambahkan pekerja sudah ditentukan oleh direktur CV. Cremonia Teknik Con-

sultant, yakni minimum 30 hari. Persentase penambahan pekerja juga sudah ditentukan oleh direktur, yakni antara 0% sampai 40%.

Biaya awal (b_0) dari setiap item pekerjaan dihitung dari nilai volume item pekerjaan, nilai OH (Orang Harian) awal pada setiap item pekerjaan dan jenis pekerja, serta biaya per hari untuk setiap jenis pekerja yang sudah ditentukan pada data.

$$b_{0_{i,j}} = V_i \times OH_{i,j} \times BPH_j \quad (2.5)$$

$$b_0 = \sum_{i=1}^N \sum_{j=1}^n b_{0_{i,j}} \quad (2.6)$$

dimana :

- b_0 : Biaya awal (Rp)
- $b_{0_{ij}}$: Biaya awal item pekerjaan ke- i pada jenis pekerja ke- j (Rp)
- BPH_j : Biaya per hari untuk jenis pekerja ke- j (Rp)
- OH_{ij} : Nilai OH (Orang Harian) awal pada item pekerjaan ke- i pada jenis pekerja ke- j
- V_i : Volume item pekerjaan ke- i
- n : Jumlah jenis pekerja (orang)

Jumlah pekerja untuk masing-masing item pekerjaan dan jenis pekerja, didapatkan dari nilai volume dan durasi awal pada item pekerjaan, serta nilai OH awal pada setiap item pekerjaan dan jenis pekerja. Perhitungan jumlah pekerja adalah sebagai berikut.

$$JP_{i,j} = \left\lceil \frac{V_i \times OH_{i,j}}{d_{0_i}} \right\rceil \quad (2.7)$$

dimana :

- d_{0_i} : Durasi awal item pekerjaan ke- i (hari)
- JP_{ij} : Jumlah pekerja pada item pekerjaan ke- i untuk jenis pekerja ke- j (orang)

OH_{ij} : Nilai OH (Orang Harian) awal pada item pekerjaan ke- i pada jenis pekerja ke- j

V_i : Volume item pekerjaan ke- i

Perhitungan biaya setiap item pekerjaan setelah penambahan pekerja akan melibatkan perhitungan koefisien OH yang baru, dengan persamaan sebagai berikut.

$$b_{1_{i,j}} = V_i \times OHB_{i,j} \times BPH_j \quad (2.8)$$

$$b_1 = \sum_{i=1}^N \sum_{j=1}^n b_{1_{i,j}} \quad (2.9)$$

$$OHB_{i,j} = \frac{OH_{i,j} \times [JP_{i,j} \times (1 + x_i)]}{JP_{i,j}} \quad (2.10)$$

dimana :

b_1 : Biaya item pekerjaan setelah penambahan pekerja (Rp)

$b_{1_{i,j}}$: Biaya item pekerjaan ke- i pada jenis pekerja ke- j setelah penambahan pekerja (Rp)

BPH_j : Biaya per hari untuk jenis pekerja ke- j (Rp)

JP_{ij} : Jumlah pekerja pada item pekerjaan ke- i untuk jenis pekerja ke- j (orang)

OH_{ij} : Nilai OH (Orang Harian) awal pada item pekerjaan ke- i pada jenis pekerja ke- j

OHB_{ij} : Nilai OH pada item pekerjaan ke- i pada jenis pekerja ke- j setelah penambahan pekerja

V_i : Volume item pekerjaan ke- i

n : Jumlah jenis pekerja (orang)

N : Jumlah item pekerjaan

x_i : Persentase penambahan pekerja pada item pekerjaan ke- i (%)

Adapun untuk menghitung durasi setelah penambahan pekerja adalah sebagai berikut.

$$d_{1i} = \frac{d_{0i} \times \sum_{j=1}^n JP_{i,j}}{\sum_{j=1}^n [JP_{i,j} \times (1 + x_i)]} \quad (2.11)$$

dimana :

- d_{0i} : Durasi awal item pekerjaan ke- i (hari)
- d_{1i} : Durasi item pekerjaan ke- i setelah penambahan pekerja (hari)
- JP_{ij} : Jumlah pekerja pada item pekerjaan ke- i untuk jenis pekerja ke- j (orang)
- n : Jumlah jenis pekerja (orang)
- x_i : Persentase penambahan pekerja pada item pekerjaan ke- i (%)

Kemudian perhitungan nilai d_1 yang akan dimasukkan dalam perhitungan nilai *cost slope*, sama dengan cara menghitung nilai d_0 yaitu jumlah dari maksimal durasi pada masing-masing kelompok urutan item pekerjaan, Sehingga setelah didapatkan nilai b_0 , b_1 , d_0 , dan d_1 , maka dapat dilanjutkan pada perhitungan nilai *cost slope*.

2.2 Artificial Bee Colony (ABC)

Artificial Bee Colony (ABC) Algorithm adalah salah satu algoritma berdasarkan suatu pendekatan *population-based meta-heuristic* yang diusulkan oleh Karaboga dan Basturk. Cara kerja algoritma ini seperti pada kawanan lebah dalam koloni yang bekerjasama untuk mencari sumber makanan. Pada model algoritma ini membagi lebah menjadi 3 kelompok, yaitu: *employed bee*, *onlookers bee*, dan *scouts bee*. *Employed bee* disini merupakan lebah yang berhubungan dengan sumber makanan tertentu, *onlooker bee* bertugas memilih sumber makanan setelah menyaksikan tarian lebah dalam sarang, dan *scout bee* bertugas mencari sumber ma-

kanan secara acak. *Onlooker bee* dan *scout bee* merupakan *unemployed bee*. Proses awal dimulai dengan *scout bee* menemukan lokasi dari semua sumber makanan, selanjutnya *employed bee* secara probabilitas memperoleh beberapa modifikasi pada posisi dalam memori untuk menargetkan sumber makanan baru dan menemukan jumlah nektar atau nilai *fitness* dari sumber baru. Kemudian, *scout bee* melakukan evaluasi informasi yang telah diambil dari semua *employed bee* dan memilih sumber makanan akhir dengan nilai probabilitas tertinggi terkait dengan jumlah nektar tersebut. Jika nilai *fitness* yang baru lebih besar dari sebelumnya, lebah tersebut akan melupakan informasi lama dan menghafal posisi baru. Hal ini disebut sebagai *greedy selection*. Kemudian *employed bee* dengan sumber makanan yang telah habis akan menjadi *scout bee* untuk kembali mencari sumber makanan selanjutnya sekali lagi. Adapau ilustrasi mengenai algoritma ABC ditunjukkan pada Gambar 2.1 sampai dengan Gambar 2.3.

Adapun langkah-langkah yang terdapat pada algoritma ABC ini adalah sebagai berikut.

1. Inisialisasi

- a. Dibangkitkan nilai awal populasi, *cost slope*, *fitness*, dan juga *array trial* yang berisikan nilai 0 sebagai *input* pada fase selanjutnya.
- b. Diingat pada *food source* berapa terdapat nilai *cost slope* terbaik beserta solusinya.
- c. Perhitungan nilai awal popoulasi menggunakan Persamaan (2.12).
- d. Kemudian perhitungan nilai *cost slope* menggunakan persamaan pada lampiran 1.
- e. Selanjutnya perhitungan nilai *fitness* dapat dilakukakan dengan menggunakan Persamaan (2.13).

$$X_{ij} = X_{j\min} + \text{rand}(0,1) \cdot (X_{j\max} - X_{j\min}) \quad (2.12)$$

dimana

- X_{ij} = Inisiasi kemungkinan solusi ke- i untuk item pekerjaan ke- j
 $X_{j\min}$ = Batas bawah dari seluruh kemungkinan solusi
 $X_{j\max}$ = Batas atas dari seluruh kemungkinan solusi
 $rand(0,1)$ = Nilai acak antara 0 sampai 1
 i = $1, 2, \dots, SN$, SN adalah jumlah kemungkinan solusi (sumber makanan)
 j = $1, 2, \dots, D$, D adalah jumlah item pekerjaan

$$fitness(x_i) = \left\{ \begin{array}{l} \frac{1}{(1 + f(x_i))}, f(x_i) \geq 0 \\ 1 + |f(x_i)|, f(x_i) < 0 \end{array} \right\} \quad (2.13)$$

dimana

$f(x_i)$ = Nilai fungsi objektif pada *food source* ke- i

2. Employed Bee Phase

- a. dilakukan penghitungan ulang untuk sebagian solusi pada seluruh *food source*.
- b. Apabila solusi yang baru memiliki nilai *fitness* yang lebih tinggi daripada sebelumnya, maka solusi yang baru akan diingat.
- c. Jika tidak maka solusi yang lama yang akan diingat dan *array trial* akan bertambah satu satuan.
- d. Perhitungan nilai soulsi yang baru dapat menggunakan Persamaan (2.14).

$$V_{ij} = X_{ij} + \phi_{ij} \cdot (X_{ij} - X_{kj}) \quad (2.14)$$

dimana

- V_{ij} = Nilai solusi baru solusi ke- i untuk item pekerjaan ke- j
 X_{ij} = Nilai solusi lama ke- i untuk item pekerjaan ke- j

- $i = 1, 2, \dots, SN$, SN adalah jumlah kemungkinan solusi (sumber makanan)
 $j = 1, 2, \dots, D$, D adalah jumlah item pekerjaan
 $k = 1, 2, \dots, SN$, SN adalah jumlah kemungkinan solusi (sumber makanan)
 ϕ_{ij} = Bilangan real acak antara -1 sampai 1

3. Onlooker Bee Phase

- Dihitung nilai probabilitas pada masing-masing *food source*.
- Pada fase ini tidak seluruh *food source* akan dilakukan perhitungan ulang, akan tetapi hanya pada *food source* tertentu saja.
- Penentuan *food source* tersebut ditentukan oleh nilai bilangan random antara 0 sampai 1 yang selanjutnya akan dibandingkan dengan nilai probabilitas.
- Apabila nilai bilangan random kurang dari nilai probabilitas maka pada *food source* tersebut akan dilakukan perhitungan solusi baru.
- Kemudian akan dilakukan perbandingan antara nilai *cost slope* baru dan yang terbaik yang diperoleh dengan nilai *cost slope* terbaik sebelumnya.
- Apabila nilai *cost slope* terbaru lebih kecil, maka nilai *cost slope* terbaru akan diingat, jika tidak maka nilai *cost slope* terbaik sebelumnya akan tetap diingat.
- Adapun perhitungan nilai probabilitas dapat dilakukan dengan menggunakan Persamaan (2.15).

$$P_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \quad (2.15)$$

dimana

$fitness_i$ = Nilai fitness solusi ke- i

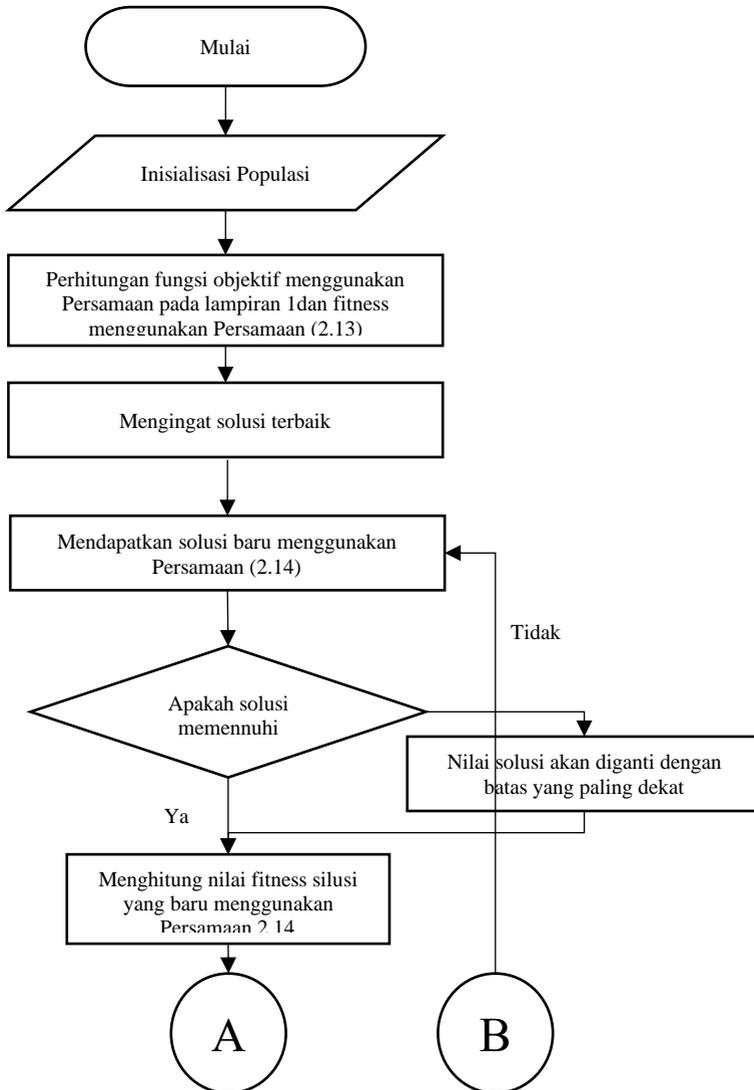
$$\sum_{i=1}^{SN} fitness_i = \text{Jumlah dari nilai fitness ke-}i \text{ sampai } SN$$

4. Scout Bee Phase

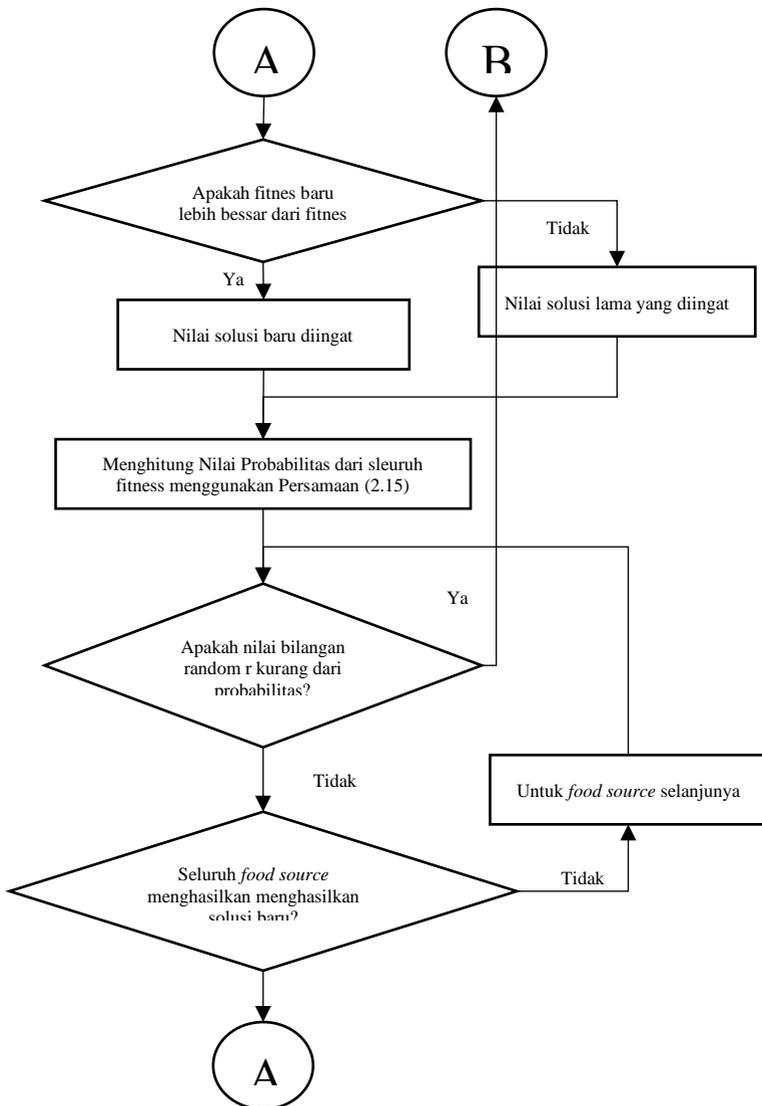
- a. Fase *scout bee* tidak selalu dilalui dalam satu *cycle*, akan tetapi hanya *food source* dengan nilai *trial* lebih dari nilai *limit* yang akan melalui fase *scout bee*.
- b. Adapun perhitungan pada fase *scout bee* dapat dilakukan dengan menggunakan Persamaan (2.12)

5. Seleksi Solusi Terbaik

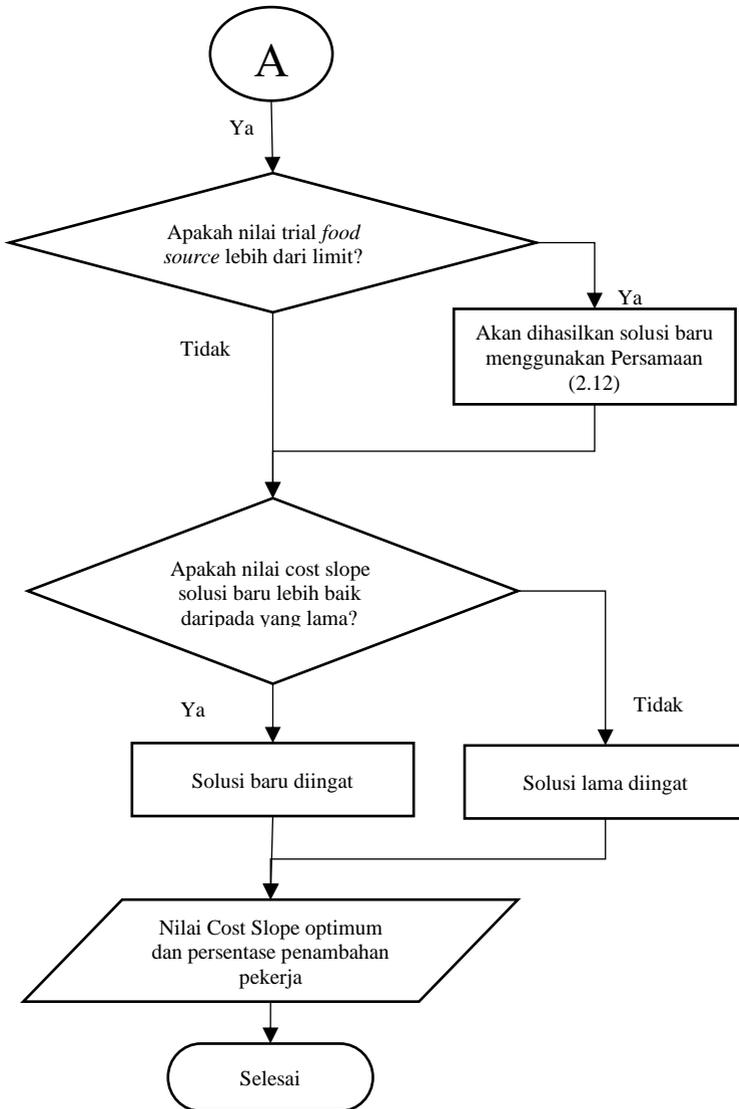
- a. Membandingkan apakah nilai *cost slope* terbaru yang dihasilkan memiliki nilai *fitness* yang lebih besar daripada nilai *cost slope* terbaik sebelumnya.
- b. Jika iya maka akan nilai *cost slope* terbaru akan diingat.
- c. Jika tidak, maka akan dilupakan dan tetap mengingat solusi beserta nilai *cost slope* yang lama.



Gambar 2.1 Diagram Alir Algoritma ABC



Gambar 2.2 Diagram alir algoritma ABC



Gambar 2.3 Diagram alir algoritma ABC

2.3.1 Inisialisasi Populasi

Pada tahap inisiasi, sebuah kumpulan posisi dari *food source* secara acak dipilih oleh lebah beserta jumlah nektar dari *food source*. *Food source* yang di maksud merupakan seluruh solusi yang memungkinkan pada permasalahan optimasi. Jumlah nektar yang yang dimaksud berhubungan dengan kualitas dari nilai *fitness* yang dihasilkan sehingga berhubungan juga dengan kebaikan dari solusi yang dihasilkan. Kemudian lebah-lebah tersebut akan datang ke sarang untuk membagikan informasi mengenai nektar dari setiap *food source*.

Pada algoritma ABC untuk mendapatkan posisi *food source* neserta jumlah nektar (nilai solusi), maka dapat dilakukan perhitungan menggunakan rumus seperti pada Persamaan (2.12). Pada penelitian ini, peneliti menggunakan jumlah *food source* sebesar 10. Selain menentukan *food source*, pada tahap ini ditentukan juga batas atas dan batas bawah dari solusi yang akan dihasilkan dimana batas atas yang akan digunakan adalah 0,4 dan batas bawah yang akan digunakan adalah 0. Kemudian pada tahap inisiasi ditentukan juga nilai *limit* yang akan digunakan untuk mendapatkan solusi baru pada tahap *scout bee*.

Setelah mendapatkan nilai pada seluruh *food source*, maka tahap selanjutnya akan dihitung nilai fungsi objektif dari masing-masing *food source* menggunakan persamaan yang terdapat pada lampiran. Setelah mendapatkan nilai fungsi objektif dari masing-masing *food source*, maka selanjutnya dapat dihitung nilai *fitness* dari masing-masing *food source* menggunakan Persamaan (2.13).

Setelah didapatkan nilai *fitness*, maka selanjutnya nilai *fitness* tersebut akan diingat oleh lebah untuk selanjutnya dibandingkan dengan nilai *fitness* baru yang dihasilkan pada tahap selanjutnya.

2.3.2 Employed Bee Phase

Pada tahap *employed bee*, setelah lebah memberikan informasi mengenai jumlah nektar pada masing-masing *food source* maka masing-masing *employed bee* akan pergi ke masing-masing *food source* yang telah dikunjunginya pada tahap sebelumnya karena

food source tersebut masih ada pada ingatan dari *employed bee*. Kemudian *employed bee* melakukan modifikasi dari posisi *food source* sebelumnya pada ingatannya berdasarkan informasi yang telah ada sebelumnya untuk menemukan *food source* yang baru dan melakukan pengujian pada nektar *food source* yang baru. Jika jumlah nektar dari *food source* yang baru lebih besar daripada jumlah nektar pada *food source* yang lama, maka *employed bee* akan mengingat posisi *food source* yang baru dan melupakan posisi *food source* yang lama. Jika jumlah nektar dari *food source* yang baru lebih kecil dari pada jumlah nektar pada *food source* yang lama, maka *employed bee* akan tetap mengingat posisi *food source* yang lama. Pada algoritma ABC rumus untuk menghitung jumlah nektar yang baru maka digunakan rumus seperti pada Persamaan (2.14).

Setelah mendapatkan solusi yang baru maka akan dibandingkan dengan batas atas dan batas bawah dari seluruh kemungkinan solusi. Apabila nilai dari solusi baru kurang dari batas bawah, maka nilai solusi baru tersebut akan diganti dengan batas bawah. Apabila nilai solusi baru lebih dari batas atas, maka nilai baru tersebut akan digantikan dengan batas atas. Kemudian setelah seluruh sumber makanan menghasilkan nilai solusi yang baru, maka tahap selanjutnya yaitu *greedy selection*.

Pada tahap *greedy selection*, nilai yang akan dibandingkan adalah nilai *fitness* dari dari seluruh *food source* dengan solusi yang baru dan seluruh *food source* dengan solusi yang lama. Apabila nilai *fitness* dari dari *food source* yang baru lebih dari nilai *fitness* dari *food source* yang lama, maka nilai solusi dari *food source* yang lama akan dilupakan dan digantikan dengan nilai *food source* yang baru. Apabila nilai *fitness* dari solusi yang lama lebih besar daripada nilai *fitness* dari solusi yang lama, maka nilai solusi dari *food source* yang lama akan tetap diingat oleh *employed bee* kemudian nilai *trial* akan ditambah satu satuan.

2.3.3 *Onlooker Bee Phase*

Pada tahap selanjutnya *onlooker bee* memilih *food source* berdasarkan dari informasi jumlah nektar yang diberikan oleh *employed bee* pada sarang. Apabila jumlah nektar semakin bertambah, maka probabilitas dimana *food source* tersebut terpilih oleh *onlooker bee* juga bertambah. *Employed bee* merekrut *onlooker bee* untuk melakukan modifikasi *food source* seperti yang dilakukan *employed bee*. Setelah sampai ke *food source* *onlooker bee* akan *onlooker bee* akan memilih *food source* dengan probabilitas yang tinggi untuk dilakukan modifikasi agar didapatkan solusi yang baru. Pada algoritma ABC, perhitungan probabilitas dari masing-masing *food source* dilakukan dengan Persamaan (2.15).

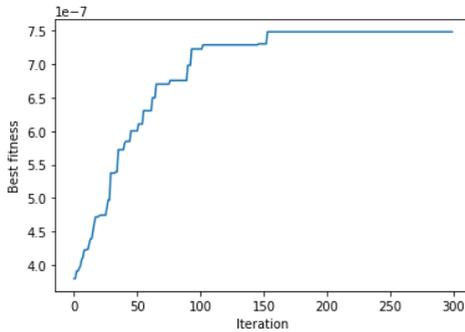
Setelah dilakukan perhitungan nilai probabilitas, maka tahap selanjutnya akan dilakukan pemilihan *food source* oleh *onlooker bee* berdasarkan informasi yang diperoleh dari *employed bee*. Pada tahap ini akan dibangkitkan nilai random antara 0 sampai dengan 1, apabila nilai random tersebut kurang dari nilai probabilitas dari *food source* tertentu, maka *food source* tersebut akan dilakukan perhitungan solusi baru. Apabila nilai random lebih dari probabilitas, maka pada *food source* tersebut tidak dilakukan perhitungan nilai solusi yang baru. Selanjutnya nilai *fitness* terbaik dari seluruh *food source* yang baru akan dibandingkan dengan nilai terbaik dari *food source* yang lama pada tahap inisiasi. Apabila nilai *fitness* dari *food source* yang baru lebih besar daripada nilai *fitness* dari *food source* yang lama, maka seluruh *food source* pada posisi itu akan diganti dengan solusi pada *food source* terbaru. Apabila nilai *fitness* terbaik dari *food source* yang baru lebih kecil dari nilai *fitness* terbaik dari *food source* yang lama maka solusi pada *food source* tersebut akan tidak diubah dan nilai *trial* akan bertambah satu satuan.

2.3.4 *Scout Bee Phase*

Tahap *scout bee* tidak selalu ada pada satu siklus. Tahap ini akan terjadi apabila nilai *trial* dari masing-masing *food source* lebih

dari limit yang ditentukan. Pada tahap ini akan dihitung nilai solusi yang baru dari food source yang memiliki trial lebih dari limit yang ditentukan menggunakan Persamaan (2.12).

X_{\min}^j dan X_{\max}^j adalah batas bawah dan batas atas dari seluruh solusi yang memungkinkan. Setelah fase *scout bee* selesai, maka satu siklus telah terpenuhi dan diulang dari fase *employed bee* untuk siklus selanjutnya. Siklus akan berhenti ketika nilai fitness sudah konvergen. Adapaun ilustrasi dimana fitness sudah konvergen ditunjukkan pada Gambar 2.2 dimana nilai fitness tidak mengalami perubahan setelah perubahan terakhir.



Gambar 2.4 Ilustrasi nilai *fitness* yang sudah konvergen

2.3 Penjadwalan

Penjadwalan adalah aktivitas perencanaan untuk menentukan kapan dan dimana setiap operasi sebagai bagian dari pekerjaan secara keseluruhan harus dilakukan pada sumber daya yang terbatas, serta pengalokasian sumber daya pada suatu waktu tertentu dengan memperhatikan kapasitas sumber daya yang ada (Prasetya, 2017). Penjadwalan memiliki banyak manfaat baik bagi *client* maupun kontraktor. Manfaat penjadwalan bagi *client* adalah mengetahui waktu mulai dan selesai dari proyek, merencanakan aliran dana, mengevaluasi efek perubahan terhadap waktu penyelesaian dan biaya proyek. Sedangkan manfaat penjadwalan bagi kontraktor adalah memprediksi kapan suatu kegiatan yang spesifik dimulai

dan di akhiri, merencanakan kebutuhan material, peralatan, dan tenaga kerja, dan mengevaluasi efek perubahan terhadap waktu penyelesaian dan biaya proyek.

2.4 Manajemen Proyek

Manajemen proyek adalah proses merencanakan, mengorganisir, memimpin, dan mengendalikan sumber daya perusahaan untuk mencapai sasaran jangka pendek yang telah ditentukan (Rani, 2016). Manajemen proyek tumbuh karena dorongan mencari pendekatan pengelolaan yang sesuai dengan tuntutan dan sifat kegiatan proyek, suatu kegiatan yang dinamis dan berbeda dengan kegiatan operasional rutin.

2.5 *Time-Cost Trade-Off Problem*

Time-Cost Trade-Off (TCTO) Problem adalah salah satu teknik yang dapat digunakan dalam manajemen proyek untuk menyelesaikan permasalahan optimasi biaya dan waktu. Tujuan utama dari perencanaan proyek adalah untuk mendapatkan biaya minimum dari proyek tersebut dengan durasi penyelesaian proyek yang se-suai (Zaid dkk,2017).

(Halaman ini sengaja dikosongkan)

BAB III

METODOLOGI PENELITIAN

3.1 Sumber Data

Sumber data yang digunakan pada penelitian ini adalah data sekunder yaitu Rancangan Anggaran Biaya (RAB) proyek pembangunan jembatan yang terletak pada Kecamatan Samboja Kabupaten Kurai Kartanegara yang diperoleh dari perusahaan CV. Cremona Teknik Consultant yang bergerak dibidang konsultasi proyek bangunan. Data Rancangan Penjadwalan Bangunan dan Rancangan Anggaran Bangunan terdiri dari 15 item pekerjaan seperti ditunjukkan pada Tabel 3.1.

Tabel 3.1 Item pekerjaan Pembangunan Proyek Jembatan

Kode	Item Pekerjaan	Urutan
A	Penyediaan Baja Struktur Grade 345	1
B	Penyediaan Tiang Pancang Baja 508x12 mm	1
C	Penyediaan Tiang Pancang Baja 408x10 mm	1
D	Penyediaan Tiang Pancang Beton Pratekan Pracetak ukuran 200x200 mm	1
E	Penyediaan Struktur Jembatan Rangka Baja Standar 60 m	2
F	Pengangkutan Bahan Jembatan yang disediakan Pengguna Jasa	2
G	Pemancangan Tiang Pancang Beton Pratekan Pracetak 200x200 mm	3
H	Pembuatan Beton, fc'15 Mpa	4
I	Pembuatan Beton, fc'10 Mpa	4
J	Pembuatan Beton struktur, fc'20 MPa	5
K	Pembuatan Beton struktur, fc'20 MPa yang dilaksanakan di air	5
L	Pembuatan Beton struktur, fc'30 MPa	6
M	Pembuatan Beton struktur, fc'25 Mpa	6

Tabel 3.2 Item pekerjaan Pembangunan Proyek Jembatan (Lanjutan)

Kode	Item Pekerjaan	Urutan
N	Pemasangan Jembatan Rangka Baja Standar Panjang 60 M	7
O	Pemasangan Baja Struktur	8

Adapun jenis pekerja yang mengerjakan proyek ini adalah pekerja, tukang, dan mandor.

3.2 Variabel Penelitian dan Struktur Data

Adapun variabel yang akan digunakan pada penelitian ini ditampilkan pada Tabel 3.2 sebagai berikut.

Tabel 3.2 Variabel Penelitian

Variabel	Nama Variabel	Skala Data
BPH	Biaya per hari untuk setiap jenis pekerja	Rasio
V	Volume item pekerjaan	Rasio
<i>da</i>	Durasi awal setiap item pekerjaan	Rasio
R	Urutan pekerjaan	Ordinal
<i>OH</i>	Nilai <i>OH</i> (Orang Harian) awal setiap item pekerjaan, pada setiap jenis pekerja	Rasio
X	Persentase penambahan pekerja setiap item pekerjaan	Rasio
CS	<i>Cost slope</i>	Rasio

Biaya per hari untuk setiap jenis pekerja, volume item pekerjaan, durasi awal setiap item pekerjaan, urutan pekerjaan dan nilai orang harian sudah ada pada data. Penelitian ini akan menghasilkan kombinasi persentase pekerjaan yang optimum, yang dievaluasi menggunakan nilai *cost slope*. Struktur data secara umum yang digunakan dalam penelitian ini terdapat pada Tabel 3.3.

Tabel 3.3 Struktur Data

Item Pekerjaan ke-	V	d_0	R	OH	x	CS
1	V_1	d_{01}	R_1	OH_1	x_1	CS_1
2	V_2	d_{02}	R_2	OH_2	x_2	CS_2
3	V_3	d_{03}	R_3	OH_3	x_3	CS_3
4	V_4	d_{04}	R_4	OH_4	x_4	CS_4
5	V_5	d_{05}	R_5	OH_5	x_5	CS_5
6	V_6	d_{06}	R_6	OH_6	x_6	CS_6
7	V_7	d_{07}	R_7	OH_7	x_7	CS_7
8	V_8	d_{08}	R_8	OH_8	x_8	CS_8
9	V_9	d_{09}	R_9	OH_9	x_9	CS_9
10	V_{10}	d_{010}	R_{10}	OH_{10}	x_{10}	CS_{10}

3.3 Langkah Analisis

Langkah-langkah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Melakukan studi literatur dan referensi terkait metode dan topik penelitian
2. Menghitung nilai durasi dan biaya pekerjaan. Adapun tahap-tahap dalam menghitung nilai durasi dan biaya pekerjaan adalah sebagai berikut.
 - a. Nilai durasi pekerjaan sudah ditentukan.
 - b. Biaya pekerjaan dapat dihitung dengan perkalian dari volume, orang harian, dan biaya pekerja setiap harinya.
3. Melakukan optimasi menggunakan algoritma ABC dengan bantuan *software python* yang didapatkan dari Nafis dan modifikasi dari internet. Adapun beberapa tahapan algoritma ABC adalah sebagai berikut.
 - a. Membangkitkan populasi awal yang terdiri dari 10 *food source* (Sharma & Pant, 2013) dengan jumlah solusi

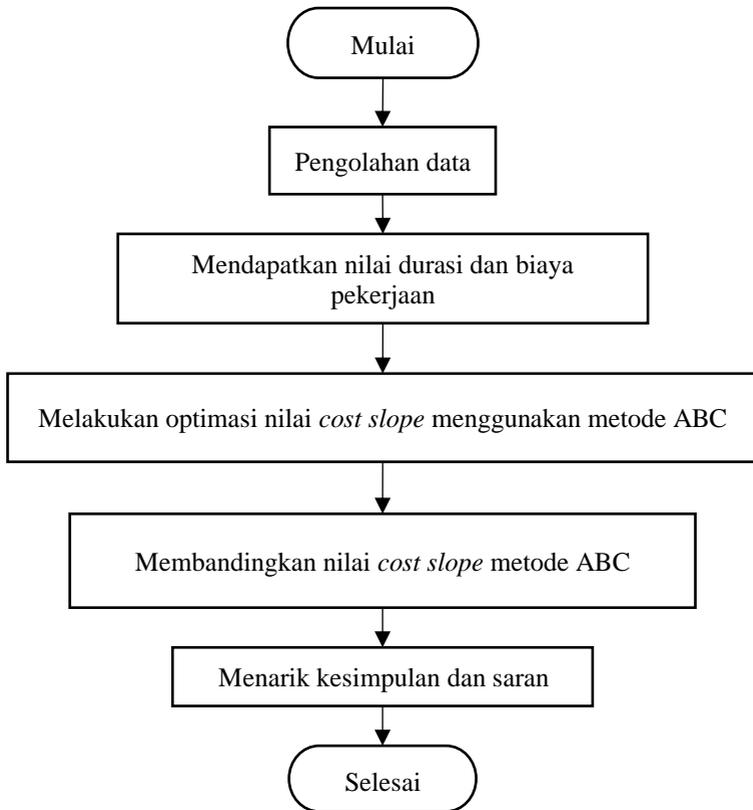
untuk masing-masing *food source* adalah sebanyak jumlah item pekerjaan yaitu 15 dengan menggunakan Persamaan (2.12), menentukan ite-rasi sebanyak 300, nilai limit sebesar 150, menentukan batas bawah solusi yaitu 0%, dan batas solusi atas yaitu 40%.

- b. Menghitung nilai fungsi objektif dalam penelitian ini adalah *cost slope* menggunakan persamaan pada lampiran 1 dan nilai *fitness* dari masing–masing *food source* yang selanjutnya akan diingat untuk nilai *cost slope* yang paling kecil.
- c. Tahap *employed bee* dimana pada tahap ini seluruh *food source* akan dihasilkan solusi baru yang lebih baik dari solusi sebelumnya apabila nilai *fitness* dari solusi yang baru lebih besar daripada nilai *fitness* solusi yang lama. Jika tidak, maka solusi yang lama akan tetap dipertahankan.
- d. Tahap *Onlooker bee* dimana pada tahap ini akan dihitung nilai probabilitas *fitness* dari masing-masing *food source* yang selanjutnya akan dihasilkan solusi baru apabila nilai probabilitas lebih besar daripada bilangan random 0 sampai 1. Kemudian nilai *cost slope* yang sudah diingat pada tahap inisiasi akan dibandingkan dengan nilai *cost slope food source* dengan nilai solusi yang baru. Apabila nilai *cost slope* terbaik *food source* yang baru lebih kecil daripada nilai *cost slope* food source dengan solusi yang lama, maka yang akan diingat adalah *cost slope* terbaik dari solusi yang baru jika tidak maka yang diingat adalah nilai *cost slope* yang lama.
- e. Tahap selanjutnya adalah tahap *scout bee* dimana dalam satu *cycle* tahap ini tidak harus dilalui. Tahap ini akan terjadi apabila nilai *trial* dari masing-masing *food source* lebih dari *limit* yang ditentukan.
- f. Setelah nilai *fitness* terbaik dari seluruh *food source* telah konvergen, maka tahapan terakhir adalah membandingkan nilai *cost slope* yang lama dengan yang baru. Apabila

nilai *cost slope* yang baru kurang dari nilai *cost slope* yang lama maka nilai *cost slope* yang baru akan diingat dan menjadi solusi yang paling optimum, jika tidak maka nilai *cost slope* yang lama yang akan diingat dan menjadi nilai *cost slope* yang optimum.

4. Melakukan perbandingan hasil antara algoritma ABC dan GA.
5. Menarik kesimpulan dan saran.

Adapun langkah-langkah di atas dapat digambarkan dengan diagram alir yang disajikan pada Gambar 3.1



Gambar 3.1 Diagram Alir Penelitian

BAB IV

ANALISIS DAN PEMBAHASAN

Pada bab ini akan dibahas tentang bagaimana cara untuk menambah jumlah pekerja dengan tujuan mempercepat waktu proyek dengan mempertimbangkan waktu serta biaya, menggunakan algoritma *Artificial Bee Colony*. Kemudian akan dilakukan perbandingan dari kedua metode tersebut menggunakan nilai *cost slope*.

4.1 Optimasi dengan *Artificial Bee Colony*

Artificial Bee Colony (ABC) dapat digunakan untuk melakukan pendekatan dalam mendapatkan persentase penambahan pekerja yang terbaik pada setiap item pekerjaan. Untuk melakukan optimasi, algoritma ABC memiliki beberapa tahapan.

4.1.1 Inisialisasi Populasi

Tahap ini adalah tahap dimana ditentukan jumlah beserta nilai dari *food source* untuk membentuk sebuah populasi. Jumlah *food source* yang digunakan pada penelitian ini adalah 10 *food source* dengan jumlah limit sebesar 300. Nilai populasi pada tahap inisiasi didapatkan dari bilangan random antara 0% sampai 40%. Adapun struktur awal populasi yang akan terbentuk ditunjukkan oleh Tabel 4.1.

Tabel 4.1 Struktur Populasi

<i>Food Source</i>	Item 1	Item 2	...	Item 14	Item 15	<i>trial</i>	<i>Cost Slope</i>	<i>Fitness</i>
1	15	22	...	38	15	0	2.586.686	3,87E-07
2	7	11	...	14	28	0	2.910.814	3,44E-07
3	7	2	...	8	38	0	inf	0,00E+00
4	9	38	...	37	20	0	2.623.616	3,81E-07
5	6	24	...	14	3	0	2.520.344	3,97E-07
6	13	20	...	4	20	0	3.212.072	3,11E-07
7	5	34	...	38	31	0	2.665.388	3,75E-07
8	38	6	...	33	13	0	3.045.222	3,28E-07

Tabel 4.1 Struktur Populasi (Lanjutan)

<i>Food Source</i>	<i>Item 1</i>	<i>Item 2</i>	...	<i>Item 14</i>	<i>Item 15</i>	<i>trial</i>	<i>Cost Slope</i>	<i>Fitness</i>
9	26	5	...	15	32	0	3.254.934	3,07E-07
10	33	33	...	40	9	0	2.967.562	3,37E-07

Selanjutnya dilakukan perhitungan fungsi objektif dari masing-masing *food source* menggunakan persamaan seperti pada lampiran. Setelah didapatkan nilai fungsi objektif selanjutnya dihitung nilai *fitness* menggunakan Persamaan (2.13). Setelah dilakukan perhitungan menggunakan untuk mendapatkan nilai *fitness* maka dapat ditemukan nilai fungsi objektif paling kecil dari seluruh *food source*. Nilai fungsi objektif beserta letaknya pada *food source* ke berapa akan diingat untuk dibandingkan dengan nilai fungsi objektif yang akan didapatkan pada fase selanjutnya. Adapun nilai fungsi objektif terbaik pada ilustrasi ini adalah Rp. 2.385.362 yang terletak pada *food source* ke 8. Selanjutnya didefinisikan *array* untuk nilai *trial* yaitu nilai yang akan bertambah satu satuan pada *food source* tertentu apabila *food source* tersebut gagal menghasilkan solusi baru dan menjadi 0 lagi ketika *food source* tersebut dapat menghasilkan solusi baru. Definisi awal dari nilai *trial* adalah 0.

4.1.2 *Employed Bee Phase*

Fase selanjutnya adalah fase *employed bee* dimana pada fase ini seluruh *food source* akan dihitung ulang untuk mendapatkan solusi yang lebih baik dengan menggunakan Persamaan (2.14). Adapun penghitungan menggunakan Persamaan (2.14) adalah sebagai berikut.

$$\begin{aligned}
 V_{5,1} &= X_{5,1} + \phi_{5,1} \cdot (X_{5,1} - X_{7,1}) \\
 &= 0,06 + (-0,56)(0,06 - 0,05) \\
 &= 0,05435
 \end{aligned}$$

Hasil dari fase *employed bee* ditampilkan pada Tabel 4.2.

Tabel 4.2 Hasil dari fase *employed bee*

<i>Food Source</i>	<i>Item 1</i>	<i>Item 2</i>	...	<i>Item 14</i>	<i>Item 15</i>	<i>trial</i>	<i>Cost Slope</i>	<i>Fitness</i>
1	15	22	...	38	15	0	2.513.731	3,98E-07
2	7	11	...	14	28	1	2.910.814	3,44E-07
3	7	2	...	8	38	1	inf	0,00E+00
4	9	38	...	37	20	1	2.623.616	3,81E-07
5	5,43	24	...	14	3	0	2.512.950	3,98E-07
6	13	20	...	4	20	0	3.153.898	3,17E-07
7	5	34	...	38	31	1	2.665.388	3,75E-07
8	38	6	...	33	13	1	3.045.222	3,28E-07
9	26	5	...	15	32	0	3.202.999	3,12E-07
10	33	33	...	40	9	1	2.967.562	3,37E-07

Berdasarkan Tabel 4.2 dapat dilihat bahwa *food source* 7, 9, 11, 12 dan 13 berhasil menghasilkan solusi baru karena nilai *trial* dari *food source* 7, 9, 11, 12 dan 13 tidak berubah atau bernilai 0, sehingga nilai *cost slope* dari *food source* 7, 9, 11, 12 dan 13 lebih kecil dari nilai *cost slope* sebelumnya.

4.1.3 *Onlooker Bee Phase*

Fase selanjutnya adalah *onlooker bee* dimana pada fase ini tidak seluruh *food source* akan dilakukan penghitungan ulang, yakni hanya *food source* tertentu yang akan dilakukan penghitungan ulang untuk mendapatkan solusi terbaru. Pemilihan *food source* tertentu didasarkan kepada nilai probabilitas dari masing-masing *food source* menggunakan Persamaan (2.15) dan nilai bilangan random dari 0 sampai 1. Apabila nilai bilangan random kurang dari nilai probabilitas, maka *food source* tersebut yang akan dipilih oleh *onlooker bee*. Adapun matriks probabilitas dari setiap *food source* disajikan pada Tabel 4.3.

$$P_i = \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i}$$

$$P_0 = \frac{3,98 \times e^{-7}}{3,19 \times e^{-6}}$$

$$= 0,12469$$

Setelah didapatkan nilai probabilitas dari masing-masing *food source*, maka selanjutnya akan dilakukan seleksi dari *food source* dengan bilangan random 0 sampai 1. perhitungan *food source* terbaru adalah sebagai berikut, yaitu pada *food source* pertama dengan nilai bilangan random adalah 0,00852091 dan nilai probabilitas adalah 0,065034.

$$V_{9,3} = X_{9,3} + \phi_{9,3} \cdot (X_{9,3} - X_{6,3})$$

$$= 0,13 + (0,84)(0,13 - 0,19)$$

$$= 0,079$$

Tabel 4.3 Nilai probabilitas masing-masing *food source*

<i>Food Source</i>	Probabilitas
1	0,124696
2	0,107686
3	0
4	0,119474
5	0,124735
6	0,099386
7	0,117601
8	0,102933
9	0,097862
10	0,105627

Adapun hasil perhitungan pada fase *onlooker bee* disajikan pada Tabel 4.4.

Tabel 4.4 Hasil dari fase *onlooker bee*

<i>Food Source</i>	<i>Item 1</i>	<i>Item 2</i>	...	<i>Item 14</i>	<i>Item 15</i>	<i>trial</i>	<i>Cost Slope</i>	<i>Fitness</i>
1	15	22	...	38	15	2	2.513.731	3.98E-07
2	7	11	...	14	28	0	2.910.814	3.44E-07
3	7	2	...	8	38	3	inf	0.00E+00
4	9	38	...	37	20	1	2.623.616	3.81E-07
5	5.43	24	...	14	3	0	2.512.950	3.98E-07
6	13	20	...	4	20	1	3.153.898	3.17E-07
7	5	34	...	38	31	1	2.651.073	3.77E-07
8	38	6	...	33	13	3	3.045.222	3.28E-07
9	26	5	...	15	32	0	3.163.447	3.16E-07
10	33	33	...	40	9	2	2.967.562	3.37E-07

Berdasarkan Tabel 4.4 dapat diketahui bahwa *food source* 2, 5, 9, dan 11 mengalami perhitungan ulang karena nilai *trial* dari *food source* 2, 5, 9, dan 11 adalah 0 serta terdapat perubahan pada nilai *cost slope* yang menjadi lebih kecil dan nilai *fitness* yang menjadi lebih besar.

4.1.4 Scout Bee Phase

Apabila pada suatu *food source* tertentu tidak menghasilkan solusi terbaik setelah 300 kali percobaan, maka selanjutnya akan dilanjutkan ke fase *scout bee* dimana pada fase ini seluruh nilai pada satu *food source* tersebut akan dihitung ulang untuk mendapatkan solusi terbaru menggunakan Persamaan (2.16). Adapun ilustrasi dari hasil fase *scout bee* disajikan pada Tabel 4.5.

Tabel 4.5 Hasil dari fase *scout bee*

<i>Food Source</i>	<i>Item 1</i>	<i>Item 2</i>	...	<i>Item 14</i>	<i>Item 15</i>	<i>trial</i>	<i>Cost Slope</i>	<i>Fitness</i>
1	2,4	0	...	22,1	0	141	1.323.733	7,55E-07
2	2,2	0	...	14	0	50	1.559.880	6,41E-07
3	0	2	...	8	0	117	1.609.060	6,21E-07

Tabel 4.5 Hasil dari fase scout bee (Lanjutan)

<i>Food Source</i>	<i>Item 1</i>	<i>Item 2</i>	...	<i>Item 14</i>	<i>Item 15</i>	<i>trial</i>	<i>Cost Slope</i>	<i>Fitness</i>
4	2,5	3,3	...	31,6	0	134	1.468.711	6,81E-07
5	0,2	15,1	...	35,1	10,4	0	2.711.068	3,69E-07
6	0	3,7	...	4	0	70	1.496.051	6,68E-07
7	0	2,0	...	17,7	0	70	1.389.178	7,20E-07
8	0	3,8	...	33,2	0	75	1.499.781	6,67E-07
9	0	0	...	15	1,1	13	1.384.158	7,22E-07
10	2,1	3,0	...	32,5	0,9	95	1.701.160	5,88E-07

Pada Tabel 4.5 dapat dilihat bahwa terdapat perubahan nilai baik pada nilai *food source*, *trial*, *cost slope*, maupun *fitness* pada *food source* ke 9. Hal ini terjadi karena nilai *trial* pada *food source* ke-9 lebih dari limit yang telah ditentukan, yakni 302.

$$\begin{aligned}
 X_{5,1} &= X_{\min}^1 + \text{rand}(0,1)(X_{\max}^1 - X_{\min}^1) \\
 &= 0 + 0,0065(0,4 - 0) \\
 &= 0,00263
 \end{aligned}$$

Setelah sampai pada fase *scout bee*, selanjutnya seluruh tahapan diulang sampai pada iterasi tertentu atau sampai mendapatkan nilai *fitness* yang konvergen. Setelah didapatkan nilai *fitness* yang konvergen, maka selanjutnya akan dilakukan perbandingan nilai fungsi objektif dari iterasi terakhir dan juga nilai fungsi objektif yang terbaik sebelumnya. Solusi dari nilai fungsi objektif yang paling kecil akan dipilih sebagai solusi terbaik.

4.1.5 Hasil Optimasi *Artificial Bee Colony*

Optimasi menggunakan *ABC* menunjukkan hasil bahwa penambahan pekerja hanya dilakukan pada kegiatan Penyediaan Struktur Jembatan Rangka Baja Standar 60 m, Pengangkutan Bahan Jembatan yang disediakan Pengguna Jasa, Pemasangan Tiang Pancang Beton Pratekan Pracetak 200x200 mm, dan Pemasangan Jembatan Rangka Baja Standar Panjang 60 M.

Tabel 4.6 Kombinasi Terbaik dengan *Artificial Bee Colony*

Item Pekerjaan	Persentase Penambahan Pekerja (%)
Penyediaan Struktur Jembatan Rangka Baja Standar 60 m	15
Pengangkutan Bahan Jembatan yang disediakan Pengguna Jasa	25
Pemancangan Tiang Pancang Beton Pratekan Pracetak 200x200 mm	25
Pemasangan Jembatan Rangka Baja Standar Panjang 60 M	13,498

Kombinasi pada Tabel 4.6 merupakan kombinasi terbaik dari berbagai kemungkinan kombinasi yang dihasilkan dari seluruh *food source* pada algoritma ABC dengan nilai *cost slope* sebesar Rp1.274.097,96. Pada lampiran, ditunjukkan nilai *fitness* terbaik dari seluruh *food source* dalam 300 iterasi dengan 10 kali percobaan. Nilai *fitness* hampir selalu mengalami kenaikan dari iterasi ke iterasi. Konvergensi nilai *fitness* juga bervariasi dari 10 kali percobaan yang dilakukan mulai dari 150 iterasi hingga 250 iterasi untuk mencapai nilai *fitness* yang konvergen.

Optimasi yang dilakukan dengan algoritma ABC sudah memenuhi *constraint* dari penelitian ini dimana selisih durasi sebelum dan sesudah dioptimasi sebesar 30 hari dan rentang persentase penambahan pekerja diantara 0% sampai 40%. Kerugian total yang dihasilkan akibat penambahan pekerja adalah sebesar Rp 38.222.939,00. Nilai *cost slope* dari metode optimasi ini adalah sebesar Rp 1.274.097,96, yang memiliki arti bahwa kerugian per hari percepatan pembangunan jembatan dengan metode ABC adalah Rp 1.274.097,96.

Tabel 4.7 Jumlah Pekerja Setelah Dilakukan Optimasi menggunakan *Artificial Bee Colony*

Item Pekerjaan	Peke rja	Tuk ang	Man dor
Penyediaan Baja Struktur Grade 345	36	7	4
Penyediaan Tiang Pancang Baja 508x12 mm	21	7	2
Penyediaan Tiang Pancang Baja 408x10 mm	42	14	4
Penyediaan Tiang Pancang Beton Pratekan Pracetak ukuran 200x200 mm	87	29	8
Penyediaan Struktur Jembatan Rangka Baja Standar 60 m	5	2	2
Pengangkutan Bahan Jembatan yang disediakan Pengguna Jasa	7	2	2
Pemancangan Tiang Pancang Beton Pratekan Pracetak 200x200 mm	4	4	3
Pembuatan Beton, fc'15 Mpa	43	64	7
Pembuatan Beton, fc'10 Mpa	28	56	7
Pembuatan Beton struktur, fc'20 MPa yang dilaksanakan di air	7	11	1
Pembuatan Beton struktur, fc'30 MPa	11	27	2
Pembuatan Beton struktur, fc'25 Mpa	19	29	3
Pemasangan Jembatan Rangka Baja Standar Panjang 60 M	4	2	2
Pemasangan Baja Struktur	70	21	4

Berdasarkan percobaan sebanyak 10 kali pada algoritma ABC, menunjukkan bahwa rata-rata lama waktu untuk *running* program adalah 90 detik. Dimana catatan lama waktu pada masing-masing percobaan telah dilampirkan pada lampiran. Selanjutnya akan dilakukan perbandingan dari kedua metode yang di-gunakan pada

penelitian ini ditinjau dari nilai *cost slope* dan lama waktu untuk menghasilkan kombinasi paling baik.

4.2 Perbandingan Nilai *Cost Slope* dan Lama Waktu Untuk Mendapatkan Solusi

Tabel 4.8 Perbandingan *cost slope* dan lama waktu yang dibutuhkan metode ABC dan GA

Metode	<i>Cost Slope</i>	Lama Waktu Mendapatkan Solusi (detik)
ABC	Rp 1.274.097,96	90
GA	Rp 1.274.097,96	117

Tabel 4.8 menyajikan perbandingan nilai *cost slope* yang didapatkan menggunakan metode ABC dan GA beserta lama waktu yang dibutuhkan untuk mendapatkan solusi yang optimum. Nilai *cost slope* yang optimum dari kedua metode adalah Rp1.274.097,96, yang berarti bahwa kedua metode dapat menghasilkan nilai *cost slope* yang optimum. Apabila ditinjau dari lama waktu untuk menghasilkan solusi, metode ABC lebih cepat dibandingkan dengan metode GA, sehingga dapat disimpulkan bahwa metode ABC lebih efisien.

(Halaman ini sengaja dikosongkan)

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut.

1. Optimasi menggunakan ABC menunjukkan hasil bahwa persentase penambahan pekerja dilakukan pada kegiatan penyediaan struktur jembatan rangka baja standar 60 meter yaitu sebesar 15%, pengangkutan bahan jembatan yang disediakan pengguna jasa sebesar 25%, pemancangan tiang pancang beton pratekan pracetak ukuran 200 mm x 200 mm sebesar 25%, dan pemasangan jembatan rangka baja standar dengan panjang 60 m sebesar 13,498%.
2. Nilai cost slope yang dihasilkan oleh metode ABC adalah sebesar Rp1.274.097,96 dengan waktu rata-rata dari 10 kali percobaan sebesar 90 detik. Apabila dibandingkan dengan hasil dari penelitian yang dilakukan oleh Moch. Abdillah Nafis pada tahun 2020 dengan menggunakan metode *Genetic Algorithm* (GA), nilai cost slope yang dihasilkan oleh metode ABC dan GA adalah sama yaitu Rp1.274.097,96 dengan waktu 117 detik. Dalam hal waktu menjalankan program, metode ABC lebih cepat yaitu membutuhkan rata-rata 90 detik.

5.2 Saran

Berdasarkan pertimbangan hasil penelitian ini, maka terdapat beberapa saran yang direkomendasikan sebagai berikut:

1. Pada penelitian selanjutnya diharapkan untuk memperhatikan *computational cost* agar algoritma yang dijalankan lebih cepat dan lebih efisien.
2. Penelitian ini dapat dikembangkan tidak hanya optimasi yang dilakukan terhadap jumlah pekerja. Jumlah *shift* maupun sumberdaya lainnya dapat dijadikan sebagai variabel yang akan meminimumkan nilai *cost slope*.

(Halaman ini sengaja dikosongkan)

DAFTAR PUSTAKA

- Cavalcante, V., Cardonha, C., & Herrmann, R. (2013). A Resource Constrained Project Scheduling Problem with Bounded Multitasking. *6th IFAC Conference on management and control of production and logistics*.
- Dorigo, M., Birattari, M., & Thomaz, S. (2006). Ant Colony Optimization. *IEEE Computational Intelligence Magazine*.
- Hadimuljono, B. (2018, Oktober 24). *berita PUPR*. Diambil kembali dari [pu.go.id: https://www.pu.go.id/berita/view/16302/4-tahun-kabinet-kerja-infrastruktur-dukung-pertumbuhan-ekonomi-dan-peningkatan-daya-saing-nasional](https://www.pu.go.id/berita/view/16302/4-tahun-kabinet-kerja-infrastruktur-dukung-pertumbuhan-ekonomi-dan-peningkatan-daya-saing-nasional)
- Hong, W., Tangling, L., & Dan, L. (2010). Efficient Genetic Algorithm for Resource-Constrained Project Scheduling Problem. *Trans. Tianjin Univ.*, 376-382.
- Izzah, N. (2017). Analisis Pertukaran Waktu dan Biaya Menggunakan Metode Time Cost Trade Off (TCTO) pada Proyek Pembangunan Perumahan di PT.X. *Jurnal Ilmiah Rekayasa*, 51-58.
- Karaboga, D., & Akay, B. (2004). A Comparative Study of Artificial Bee Colony Algorithm. *Applied Mathematics and Computation*. 108-132.
- Karaboga, D., & Basturk, B. (2007). A Powerful And Efficient Algorithm For Numerical Function Optimization: Artificial Bee Algorithm (Abc) Algorithm. *J Glob Optim*, 459-471.

- Kocen, H., & Akca, M. (2014). Cybernetics and Systems: An International Journal. An Improved Artificial Bee Colony Algorithm with Local Search For Traveling Salesman Problem. 635-649.
- KPPIP. (2019). *Perkembangan Pembangunan Infrastruktur Di Indonesia*. Diambil kembali dari <https://kppip.go.id/tentang-kppip/perkembangan-pembangunan-infrastruktur-di-indonesia/>
- kuwado, F., & Wedhaswari, I. (2018). Diambil kembali dari [kompas.com: https://nasional.kompas.com/read/2018/10/20/14144381/4-tahun-jokowi-jk-dan-catatan-pembangunan-infrastruktur?page=all](https://nasional.kompas.com/read/2018/10/20/14144381/4-tahun-jokowi-jk-dan-catatan-pembangunan-infrastruktur?page=all).
- Lestari, I. (2015). Faktor-Faktor Yang Mempengaruhi Pembangunan Infrastruktur (Jalan dan Jembatan) di Kecamatan Kampar Kiri Hulu Kabupaten Kampar Tahun 2011-2013. *Jom FISIP Volume 3 No. 1*.
- Muhammad, A. A., & Indriyani, R. (2015). Analisa Time Cost Trade Off pada Proyek Pasar Sentral Gadang Malang. *JURNAL TEKNIK ITS Vol. 4, No. 1, (2015) ISSN 2337-3539 (2301-9271 Printed)*.
- Mutakhirah, I., Saptono, F., Hasanah, N., & Wiryadinata, R. (2007). Pemanfaatan Metode Heuristic dalam Pencarian Jalur Terpendek dengan Algoritma Semut dan Algoritma Genetika. *Seminar Nasional Aplikasi Teknologi Informasi 2007*.
- Muthiah, A., & Rajkumar, R. (2014). A Comparison of Artificial Bee Colony algorithm and Genetic Algorithm to minimize

- the makespan for Job Shop Scheduling. *Procedia Engoneering* , 1745-1754.
- Nafis, M. A. (2020). *Optimasi Biaya Dan Waktu Proyek Pembangunan Jembatan Menggunakan Constraint Programming Dan Genetic Algorithm*. Surabaya: Institut Teknologi Sepuluh Nopember.
- Prasetya, N. E. (2017). *Penjadwalan Fleksibel Flowshop Dengan Menggunakan Algoritma Long Processing Time-Ln Untuk Minimasi Biaya Energi (Studi Kasus PT. Sinaraya Nugraha Ahmadaris Medika)*. Malang: Universitas Muhammadiyah Malang.
- Prayogo, D., Gosno, R., Evander, R., & Limanto, S. (2017). Implementasi Metode Metaheuristic Symbiotic Organisms Search dalam Penentuan Tata Letak Fasilitas Proyek Konstruksi Berdasarkan Jarak Tempuh Pekerja. *Jurnal Teknik Industri*, 19 No 2, 103-114.
- Priyo, M., & Auliya, M. (2015). Aplikasi Metode Time Cost Trade Off Pada Proyek Konstruksi: Studi Kasus Proyek Pembangunan Gedung Indonesia . *JURNAL ILMIAH SEMESTA TEKNIKA* , 18 No 1, 30-43.
- Rani, H. A. (2016). *Manajemen Proyek Konstruksi*. Yogyakarta: Deepublish.
- Sugiyono, A. (2006). Perkembangan Dan Aplikasi Teknik Optimasi Untuk Perencanaan Energi. *Prosiding Semiloka Teknologi Simulasi dan Komputasi serta Aplikasi*, BPPT.
- Yustika, A. E. (2018, Oktober 20). *4 Tahun Jokowi-JK dan Catatan Pembangunan Infrastruktur*. Diambil kembali dari <https://nasional.kompas.com/read/2018/10/20/14144381/>

4-tahun-jokowi-jk-dan-catatan-pembangunan-
infrastruktur?page=all

Zaid, M., Korany, R., Etman, E., & Taher, S. (2017). Optimization Of Project Time-Cost Trade-Off With Controlled Cash Flows. *International Conference on Advances in Structural and Geotechnical Engineering*.

LAMPIRAN

Lampiran 1. Perhitungan Nilai *Cost Slope*

$$CS = \frac{\sum_{i=1}^N \sum_{j=1}^n V_i \times OH_{i,j} \left[\frac{\left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \times (1+x_i) \right]}{\left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right]} - 1 \right] \times BPH_j}{\sum_{i=1}^N \max [d_{0,i}, d_{0,2}, \dots, d_{0,8}] - \sum_{i=1}^N \max \left[\frac{\left(\frac{d_{0,i} \times \sum_{j=1}^n \left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right]}{\sum_{j=1}^n \left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right] \times (1+x_i)} \right)_1, \left(\frac{d_{0,i} \times \sum_{j=1}^n \left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right]}{\sum_{j=1}^n \left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right] \times (1+x_i)} \right)_2, \dots, \left(\frac{d_{0,i} \times \sum_{j=1}^n \left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right]}{\sum_{j=1}^n \left[\frac{V_i \times OH_{i,j}}{d_{0,i}} \right] \times (1+x_i)} \right)_8 \right]}$$

dimana :

BPH_j : Biaya per hari setiap jenis pekerja

CS : *Cost slope*

da_i : Durasi awal setiap item pekerjaan

n : Jumlah jenis pekerja

N : Jumlah item pekerjaan

OH_{ij} : Koefisien orang harian setiap item pekerjaan dan jenis pekerja

V_i : Volume setiap item pekerjaan setiap item pekerjaan

x_i : Persentase penambahan pekerja setiap item pekerjaan

Lampiran 2. Program *Python* untuk Menyusun Struktur Data

```

import numpy as np
import pandas as pd
import math
import random
import matplotlib.pyplot as plt
import statistics
import itertools
import time
from tqdm import tqdm

data =
pd.read_csv("E:/DOWNLOAD/DOCUMENT/TA_MIRZA_1.csv",
sep=";")
data.head()

#Definisi BPH, Volume, Durasi Awal, Orang Harian dan Biaya Awal
bph = data.BPH.T.dropna(axis=0)
vol = data.VOL.dropna(axis=0)
da = data.DURASI_AWAL.dropna(axis=0)
da_total = sum(da)
urutan = data.Urutan.dropna(axis=0)
jumlah_urutan = max(urutan)
oh = data.drop(data.columns[0:5], axis=1).dropna(axis=0)
oh = np.array(oh)
jumlah_item = pd.DataFrame(da).shape[0]
jenis_pekerja = pd.DataFrame(bph).shape[0]
a = pd.concat([da,urutan], axis=1)
b = a.groupby(urutan)
c = np.zeros(jumlah_urutan+1)
for i in range(1,jumlah_urutan+1):
    c[i] = c[i-1] + max(b.get_group(i-1).DURASI_AWAL)
total_durasi_awal = c

```

Lampiran 2. Program *Python* untuk Menyusun Struktur Data (Lanjutan)

```

start = np.zeros(len(data))
finish = np.zeros(len(data))
for i in range(len(data)):
    start[i] = total_durasi_awal[urutan[i]]
    finish[i] = da[i] + total_durasi_awal[urutan[i]]
start = pd.DataFrame(start)
finish = pd.DataFrame(finish)
start.rename(columns={0:'start'}, inplace=True)
finish.rename(columns={0:'finish'}, inplace=True)
data_jadwal = pd.concat([data["kegiatan"],start,finish,urutan,da],
axis=1)

fig,ax=plt.subplots(figsize=(6,3))
labels=[]
for i, kegiatan in enumerate(data_jadwal.groupby("kegiatan")):
    labels.append(kegiatan[0])
    for r in kegiatan[1].groupby("Urutan"):
        data = r[1][["start", "DURASI_AWAL"]]
        ax.broken_barh(data.values, (i-0.4,0.8) )
ax.set_yticks(range(len(labels)))
ax.set_yticklabels(labels)
ax.set_xlabel("hari")
plt.tight_layout()
plt.show()

```

Lampiran 2. Program *Python* untuk Menyusun Struktur Data
(Lanjutan)

```
pekerja_awal=np.zeros((jumlah_item,jenis_pekerja))
pekerja_awal_peritem=np.zeros(jumlah_item)
ba=np.zeros((jumlah_item,jenis_pekerja))
biaya_awal=np.zeros(jumlah_item)
for i in range(jumlah_item):
    for j in range(jenis_pekerja):
        # Change the required number of zeros to ones
        pekerja_awal[i,j] = math.ceil(oh[i,j]*vol[i]/da[i])
        pekerja_awal_peritem[i] = sum(pekerja_awal[i,:])
        ba[i,j] = bph[j]*vol[i]*oh[i,j]
        biaya_awal[i] = sum(ba[i,:])
a = pd.concat([da,urutan], axis=1)
b = a.groupby(urutan)
c = 0
for i in range(jumlah_urutan+1):
    c = c + max(b.get_group(i).DURASI_AWAL)
total_durasi_awal = c
```

Lampiran 3. Program Python untuk Artificial Bee Colony

```

import numpy as np
import pandas as pd
import math
import random
import matplotlib.pyplot as plt
%matplotlib inline
import statistics
import itertools
import time
from tqdm import tqdm

data = pd.read_csv("E:/MIRZA/KULIAH/SEMESTER 8/TUGAS
AKHIR/DATA/TA_MIRZA_1.csv", sep=";")
data.head()

#####
#####
#Definisi BPH, Volume, Durasi Awal, Orang Harian dan Biaya Awal
bph = data.BPH.T.dropna(axis=0)
vol = data.VOL.dropna(axis=0)
da = data.DURASI_AWAL.dropna(axis=0)
da_total = sum(da)
urutan = data.Urutan.dropna(axis=0)
jumlah_urutan = max(urutan)
oh = data.drop(data.columns[0:5], axis=1).dropna(axis=0)
oh = np.array(oh)
jumlah_item = pd.DataFrame(da).shape[0]
jenis_pekerja = pd.DataFrame(bph).shape[0]

pekerja_awal=np.zeros((jumlah_item,jenis_pekerja))
pekerja_awal_peritem=np.zeros(jumlah_item)
ba=np.zeros((jumlah_item,jenis_pekerja))
biaya_awal=np.zeros(jumlah_item)

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

for i in range(jumlah_item):
    for j in range(jenis_pekerja):

        pekerja_awal[i,j] = math.ceil(oh[i,j]*vol[i]/da[i])
        pekerja_awal_peritem[i] = sum(pekerja_awal[i,:])
        ba[i,j] = bph[j]*vol[i]*oh[i,j]
        biaya_awal[i] = sum(ba[i,:])

a = pd.concat([da,urutan], axis=1)
b = a.groupby(urutan)
c = 0
for i in range(jumlah_urutan+1):
    c = c + max(b.get_group(i).DURASI_AWAL)
total_durasi_awal = c
total_durasi_awal

def create_starting_population(food_source):
    population = np.zeros((food_source, jumlah_item))

    for i in range(food_source):
        for j in range(jumlah_item):
            population[i,j] = round(random.uniform(0,40/100),2)

    return population.astype(float)

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

def calculate_obj(population):
    food_source = len(population)
    penambahan_pekerja =
np.zeros((food_source,jumlah_item,jenis_pekerja))
    penambahan_pekerja_peritem =
np.zeros((food_source,jumlah_item))
    pekerja_akhir =
np.zeros((food_source,jumlah_item,jenis_pekerja))
    ohb = np.zeros((food_source,jumlah_item,jenis_pekerja))
    bf = np.zeros((food_source,jumlah_item,jenis_pekerja))
    biaya_akhir = np.zeros((food_source,jumlah_item))
    df = np.zeros((food_source,jumlah_item))
    cs = np.zeros(food_source)

    for i in range(food_source):
        for j in range(jumlah_item):
            for k in range(jenis_pekerja):

penambahan_pekerja[i,j,k]=math.ceil(population[i,j]*pekerja_awal[j,
k])

pekerja_akhir[i,j,k]=penambahan_pekerja[i,j,k]+pekerja_awal[j,k]
    ohb[i,j,k]=oh[j,k]*pekerja_akhir[i,j,k]/pekerja_awal[j,k]
    bf[i,j,k]=bph[k]*vol[j]*ohb[i,j,k]
    biaya_akhir[i,j]=sum(bf[i,j,:])

penambahan_pekerja_peritem[i,j]=sum(penambahan_pekerja[i,j,:])

df[i,j]=math.ceil(da[j]*pekerja_awal_peritem[j]/(pekerja_awal_perite
m[j]+penambahan_pekerja_peritem[i,j]))
    a = pd.DataFrame([df[i]]).transpose()
    a.rename(columns={0:'DURASI_AKHIR'}, inplace=True)
    b = a.groupby(urutan)
    c = 0
    for l in range(jumlah_urutan+1):

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

c = c + max(b.get_group(1).DURASI_AKHIR)
total_durasi_akhir = c
cs[i] = sum(biaya_akhir[i]-biaya_awal)/(total_durasi_awal-
total_durasi_akhir)
if total_durasi_awal-total_durasi_akhir < 30: cs[i] = np.inf

return cs.astype(float)

def FitFun(f):
    if f.all()>=0:
        fit = 1/(1+f)
    else:
        fit = 1 + abs(f)
    return fit

def init(food_source, limit):
    lb = 0
    ub = 0.4
    lb_v = np.zeros(food_source)+lb
    ub_v = np.zeros(food_source)+ub
    f = np.zeros(food_source)
    fit = np.zeros(food_source)
    trial = np.zeros(food_source)
    P = create_starting_population(food_source)
    f = calculate_obj(P)
    fit = FitFun(f)
    return P, f, fit, trial, lb, ub, food_source, limit

def bestsolution(P, f):
    PP = pd.DataFrame(P)
    bestobj = min(f)
    indrow_ = np.where(f == np.amin(f))
    bestsol = PP.iloc[indrow_[0][0], :]

    return bestobj, np.array(bestsol).reshape(1,15), indrow_

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

Def GenNewSol(P, f, fit, trial, lb, ub, food_source):
    for a in range(food_source):
        A = np.zeros((food_source, jumlah_item))
        p = random.randint(0, jumlah_item-1)
        r = random.randint(0, jumlah_item-1)

        while a==p:
            p = random.randint(0, jumlah_item-1)
            if a != p:
                break

        Phi = -1 + (1-(-1))*np.random.rand(1,1)
        XNew = P[a][r] + Phi*(P[a][r]-P[p][r])

        XNew = min(XNew,ub)
        XNew = max(XNew,lb)

        for b in range(food_source):
            for c in range(jumlah_item):
                A[b][c] = P[b][c]

        A[a][r] = XNew

        ObjNewSol = calculate_obj(A)[a]
        FitNewSol = FitFun(ObjNewSol)

        if FitNewSol < fit[a]:
            P[a][r] = A[a][r]
            f[a] = ObjNewSol
            fit[a] = FitNewSol
            trial[a] = 0
        else:
            trial[a] = trial[a]+1

    return P, f, fit, trial

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

Def GenNewSol_n(n,P, f, fit, trial, lb, ub, food_source):
    A = np.zeros((food_source,jumlah_item))
    p = random.randint(0, jumlah_item-1)
    r = random.randint(0, jumlah_item-1)

    while n==p:
        p = random.randint(0, jumlah_item-1)

    Phi = -1 + (1-(-1))*np.random.rand(1,1)
    XNew = P[n][r] + Phi*(P[n][r]-P[p][r])

    XNew = min(XNew,ub)
    XNew = max(XNew,lb)

    for b in range(food_source):
        for c in range(jumlah_item):
            A[b][c] = P[b][c]

    A[n][r] = XNew

    ObjNewSol = calculate_obj(A)[n]
    FitNewSol = FitFun(ObjNewSol)

    if FitNewSol>fit[n]:
        P[n][r] = A[n][r]
        f[n] = ObjNewSol
        fit[n] = FitNewSol
        trial[n] = 0
    else:
        trial[n] = trial[n]+1

    return P, f, fit, trial

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

def onlooker(P, f, fit, trial, bestobj, bestsol, food_source):
    probability = fit/sum(fit)
    k=0
    n=0

    while k < food_source:
        if probability[k]==0:
            k=k+1
        else:
            if np.random.rand(1)<probability[k]:
                P, f, fit, trial = GenNewSol_n(n,P, f, fit, trial, lb, ub,
                food_source)
                k=k+1
            else:
                n = n+1
                n = np.mod(n,food_source)

    ObjNewSol_1 = np.append(f,bestobj)
    bestobj_2 = min(ObjNewSol_1)
    indrow = np.where(ObjNewSol_1 == np.amin(ObjNewSol_1))
    XNew_2 = np.append(P,bestsol,0)
    bestsol_2 = XNew_2[indrow[0][0]]

    return P, f, fit, bestobj_2, bestsol_2.reshape(1,15), trial, indrow
def scoutbee(P, f, fit, trial, limit):
    value = max(trial)
    indmax = []
    for i in range(len(trial)):
        if value<=trial[i]:
            indmax = np.append(indmax, i)

    if len(indmax) != 1:
        a = random.randint(0,len(indmax)-1)
        if trial[int(indmax[int(a)])] > limit:
            trial[int(indmax[int(a)])] = 0

```

Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

        P[int(indmax[int(a)])[:]] = lb + (ub-
lb)*np.random.rand(1,jumlah_item)
        f[int(a)] = calculate_obj(P)[int(a)]
        fit[int(a)] = FitFun(f)[int(indmax[int(a))]]
    else:
        if trial[int(indmax[0])] > limit:
            trial[int(indmax[0])] = 0
            P[int(indmax[0])[:]] = lb + (ub-
lb)*np.random.rand(1,jumlah_item)
            f[int(indmax[0])] = calculate_obj(P)[int(indmax[0])]
            fit[int(indmax[0])] = FitFun(f)[int(indmax[0])]

    return P, f, fit, trial, indmax, value

def bestsolution_1(P, f, bestobj, bestsol):
    ObjNewSol_1 = np.append(f,bestobj)
    bestobj_2 = min(ObjNewSol_1)
    indrow = np.where(ObjNewSol_1 == np.amin(ObjNewSol_1))
    XNew_2 = np.append(P, bestsol,0)
    bestsol_2 = XNew_2[indrow[0][0]]
    return bestsol_2, bestobj_2, indrow[0][0], P, f

```

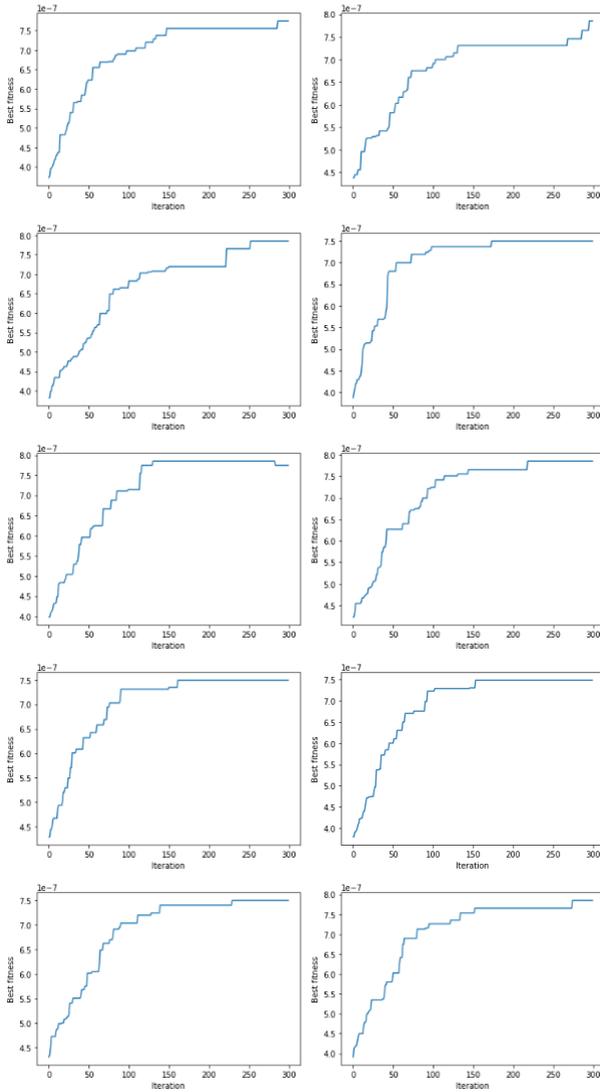
Lampiran 3. Program *Python* untuk *Artificial Bee Colony* (Lanjutan)

```

def ABC_optim_2(num_trial, iterations, food_source, limit):
    Time = []
    fit_1 = np.zeros((num_trial, iterations))
    bestfit_1 = np.zeros((num_trial, iterations))
    bestsol_1 = np.zeros((num_trial, jumlah_item))
    bestobj_1 = np.zeros(num_trial)
    for r in tqdm(range(num_trial)):
        start_time = time.time()
        P, f, fit, trial, lb, ub, food_source, limit = init(food_source, limit)
        bestobj, bestsol, indrow_ = bestsolution(P, f)
        for t in tqdm(range(iterations)):
            P, f, fit, trial = GenNewSol(P, f, fit, trial, lb, ub, food_source)
            P, f, fit, bestobj, bestsol, trial, indrow = onlooker(P, f, fit, trial,
bestobj, bestsol, food_source)
            P, f, fit, trial, indmax, value = scoutbee(P, f, fit, trial, limit)
            fit_1[r][t] = max(fit)
            bestfit_1[r][t] = FitFun(bestobj)
            bestsol, bestobj, indrow, P, f = bestsolution_1(P, f, bestobj,
bestsol)
            bestfit_1[r][iterations-1] = FitFun(bestobj)
        for e in range(jumlah_item):
            bestsol_1[r][e] = bestsol[e]
        bestobj_1[r] = bestobj
        end_time = time.time()
        total_time = end_time - start_time
        Time.append(total_time)
    return P, f, fit, trial, bestsol, bestobj, bestfit_1, bestsol_1,
bestobj_1, Time
P, f, fit, bestsol, bestobj, bestfit_1, bestsol_1, bestobj_1, Time =
ABC_optim_2(num_trial = 1, iterations = 300, food_source = 20,
limit = 300)
plt.plot(bestfit_1[0])
plt.xlabel('Iteration')
plt.ylabel('Best fitness')
plt.show()

```

Lampiran 4. Grafik *Fitness* Terbaik setiap *food source* pada 10 kali Percobaan Metode *Artificial Bee Colony*



Lampiran 5. Perhitungan pada algoritma ABCPerhitungan manual tahap *employed bee*

$$\begin{aligned}
 V_{10,12} &= X_{10,12} + \phi_{10,12} \cdot (X_{10,12} - X_{10,2}) \\
 &= 0,27 + 0,36(0,27 - 0,16) \\
 &= 0,3096
 \end{aligned}$$

Perhitungan nilai probabilitas tahap *onlooker bee*

$$\begin{aligned}
 P_i &= \frac{fitness_i}{\sum_{i=1}^{SN} fitness_i} \\
 P_0 &= \frac{3,209 \times e^{-7}}{5,457 \times e^{-6}} \\
 &= 0,058805
 \end{aligned}$$

Perhitungan manual tahap *onlooker bee*

$$\begin{aligned}
 V_{3,4} &= X_{3,4} + \phi_{3,4} \cdot (X_{3,4} - X_{7,4}) \\
 &= 0,08 + (-0,34)(0,08 - 0,39) \\
 &= 0,1854
 \end{aligned}$$

Perhitungan manual tahap *scout bee*

$$\begin{aligned}
 X_{10,3} &= X_{\min}^1 + rand(0,1)(X_{\max}^1 - X_{\min}^1) \\
 &= 0 + 0,0834(0,4 - 0) \\
 &= 0,03336
 \end{aligned}$$

Lampiran 6. Catatan Nilai *Cost Slope* dan Lama Waktu *Running* pada 10 kali Percobaan Metode *Artificial Bee Colony*

Percobaan Ke-	<i>Cost Slope</i> (Rp)	Lama Waktu (detik)
1	1.291.161	67,99771
2	1.274.097	89,17483
3	1.274.097	70,43817
4	1.334.840	99,10782
5	1.274.097	93,82643
6	1.274.097	95,4135
7	1.334.840	97,15066
8	1.336.942	97,48919
9	1.334.840	101,0408
10	1.274.097	93,72523

Lampiran 7. Surat Keterangan Pengambilan Data


CV. CREMONA TEKNIK CONSULTANT
PLA. JEREMBAH 3. A. BLD. BILATE. 801/010
 Alamat : Jl. Samelitan No.5 RT.34 Telp. 0541-7744482 / 08127969255 Fax. 0541-736493 Samarinda - Kalimantan Timur

Nomor : 2/CTC-01/V/2020
 Perihal : Izin memperoleh data untuk Tugas Akhir

Samarinda, 5 Mei 2020

Yth,
 Kepala Departemen Statistika
 Institut Teknologi Sepuluh Nopember Fakultas Sains Dan Analitika Data
 Dr. Dra. Kartika Fithriyani, M.Si.

Dengan Hormat,

Yang bertanda tangan dibawah ini :

Nama : ZONY YULFADLLS.T.,M.T
 Jabatan : Direktur

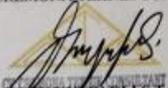
Menerangkan bahwa,

Nama : MIRZA ROMI SETIAWAN
 NRP : 06211640007001
 Program Studi : Sarjana (S1)
 Judul Tugas Akhir : Optimasi Biaya dan Waktu Proyek Bangunan X Menggunakan
 Algoritma Ant Colony Optimization, Artificial Bee Colony dan
 Linear Programming

Telah kami setuju untuk melaksanakan proses pengambilan data Rencana Anggaran Biaya.

Demikian surat ini kami sampaikan untuk dapat dipergunakan seperlunya.

DIREKTUR
 CV. CREMONA TEKNIK CONSULTANT


 ZONY YULFADLLS.T., M.T.

(Halaman ini sengaja dikosongkan)

BIODATA PENULIS



Penulis dengan nama lengkap Mirza Romi Setiawan merupakan anak pertama dari pasangan Makromi dan Siti Munawaroh. Penulis lahir di Sidoarjo pada tanggal 02 Juni 1999. Pendidikan formal penulis ditempuh di SDN Banjarebendo, MTs Unggulan Amanatul Ummah, dan MA Unggulan Amanatul Ummah. Selanjutnya, penulis melanjutkan pendidikan formal pada Program Studi Sarjana Departemen Statistika

ITS. Penulis diterima di ITS pada tahun 2016 melalui jalur PBSB (Program Beasiswa Santri Berprestasi). Selama perkuliahan, penulis aktif dalam organisasi dan kompetisi. Penulis aktif di *Community of Santri Scholars of Ministry of Religious Affairs ITS* (CSSMoRA ITS) sebagai Staf Departemen Dalam Negeri pada tahun 2017-2018 dan Kepala Departemen Dalam Negeri CSSMoRA ITS periode 2018-2019. Penulis juga pernah terlibat sebagai Sie Akomodasi dan Transportasi Pekan Raya Statistika (PRS) pada tahun 2017 dan menjadi Koordinator Sie Akomodasi dan Transportasi pada tahun 2018. Selain aktif dalam organisasi, penulis juga mengikuti berbagai perlombaan di bidang statistika yang berskala nasional. Capaian prestasi penulis yaitu tercatat beberapa kali sebagai semi-finalis pada kompetisi-kompetisi tersebut. Penulis juga pernah mendapatkan kesempatan untuk menjadi surveyor. Segala kritik dan saran akan diterima oleh penulis untuk perbaikan kedepannya. Jika ingin berdiskusi mengenai Tugas Akhir ini, maka dapat mengirimkan email ke mirzaromi12@gmail.com atau melalui nomor 087751655141.