



**ANALISIS OPTIMASI JARINGAN *FIBER TO THE HOME*
UNIVERSITAS JEMBER DENGAN METODE
*CAT SWARM OPTIMIZATION***

SKRIPSI

Oleh

Muhammad Rizza

NIM 161910201006

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER**

2020



**ANALISIS OPTIMASI JARINGAN *FIBER TO THE HOME*
UNIVERSITAS JEMBER DENGAN METODE
*CAT SWARM OPTIMIZATION***

SKRIPSI

Diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan mencapai gelar Sarjana Teknik

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2020**

PERSEMBAHAN

Dengan rasa syukur yang teramat dalam kepada Allah SWT yang mana telah memberikan rahmat, taufiq, dan hidayahNya sehingga saya dapat menyelesaikan tugas akhir ini dengan baik dan tepat waktu. Skripsi ini merupakan langkah awal saya untuk memulai lembaran pendidikan yang baru dan pencapaian berikutnya. Dengan penuh rasa terima kasih saya persembahkan skripsi ini kepada :

1. Kedua orang tua saya, Ayahanda Joko Purnomo dan Ibunda Dian Wahyu Widiyanti yang tidak ada hentinya untuk memberi dukungan dan doa serta restunya kepada saya sehingga saya dapat berada pada titik saat ini.
2. Dosen pembimbing utama Ibu Ir. Ike Fibriani, S.T, M.T. serta dosen pembimbing anggota Bapak Andrita Ceriana Eska, S.T, M.T.yang telah membimbing skripsi saya sehingga saya dapat menyelesaikannya dengan baik.
3. Bapak Ibu guru dan dosen yang senantiasa memberikan ilmu yang bermanfaat bagi saya.
4. Almamater jurusan teknik elektro, Fakultas Teknik, Universitas Jember.

MOTTO

“Tuhan tidak menuntut kita untuk sukses. Tuhan hanya menyuruh kita untuk berjuang tanpa henti”

(Emha Ainun Najib)

“Jika kamu bermimpi untuk menggapai suatu tujuan, berikan satu tingkat lebih tinggi. Seandainya kamu terjatuh, setidaknya kamu masih berada dimimpi mu sendiri”

(Muhammad Rizza)

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Muhammad Rizza

NIM : 161910201006

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Analisis Optimasi Jaringan *Fiber To The Home* Universitas jember Dengan *Metode Cat Swarm Optimization*” merupakan benar-benar hasil karya sendiri, kecuali beberapa kutipan yang telah saya sebutkan sumbernya, dan karya tulis ini belum pernah diajukan pada institusi manapun, dan bukan karya jiplakan. Saya bertanggung jawab atas keabsahan isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan yang telah saya buat dengan sebenarnya, tanpa adanya tekanan ataupun paksaan dari pihak manapun serta saya bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 14 Januari 2020

Yang menyatakan,

Muhammad Rizza

NIM 161910201006

SKRIPSI

ANALISIS OPTIMASI JARINGAN *FIBER TO THE HOME* UNIVERSITAS
JEMBER DENGAN METODE *CAT SWARM OPTIMIZATION*

Oleh :

Muhammad Rizza

NIM 161910201006

Pembimbing

Dosen Pembimbing Utama : Ir. Ike Fibriani, S.T., M.T.

Dosen Pembimbing Anggota : Andrita Ceriana Eska, S.T., M.T

PENGESAHAN

Skripsi berjudul “Analisis Optimasi Jaringan *Fiber To The Home* Universitas Jember Dengan *Metode Cat Swarm Optimization*” karya Muhammad Rizza telah diuji dan disahkan pada :

Hari : Selasa

Tanggal : 14 Januari 2020

Tempat : Fakultas Teknik Universitas Jember

Tim Penguji,

Ketua,



Ir. Ike Fibriani, S.T., M.T.
NIP 198002072015042001

Anggota I,



Andrita Ceriana Eska, S.T., M.T.
NRP 760014640

Anggota II,



Ir. Widyono Hadi, M.T.
NIP 196104141989021001

Anggota III,



Widya Cahyadi, S.T., M.T.
NIP 198511102014041001

Mengesahkan

Dekan,



Dr. Ir. Priwahju Hardianto, S.T., M.T.

NIP 197008261997021001

RINGKASAN

Analisis Optimasi Jaringan *Fiber To The Home* Universitas Jember Dengan Metode *Cat Swarm Optimization* ; Muhammad Rizza; 2019; 105 Halaman; Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Layanan telekomunikasi merupakan layanan yang sangat dibutuhkan pada saat ini, seperti halnya PT. Telkom Indonesia yang memberikan layanan telekomunikasi berupa jaringan internet dengan menggunakan teknologi terbaru yaitu kabel fiber optik dengan kecepatan tinggi, layanan PT. Telkom Indonesia yang dikemas dengan nama *Triple Play* yang mana dapat memberikan layanan berupa TV kabel, telepon rumah, dan akses internet menggunakan jaringan *Fiber To The Home* (FTTH).

Pada penelitian kali ini bertujuan untuk menganalisa jaringan *Fiber To The Home* (FTTH) yang ada pada Universitas Jember kampus tegal Boto dengan melakukan simulasi untuk mengoptimalkan jalur fiber optik yang digunakan sebagai media transmisi. Penelitian ini menggunakan metode algoritma *Cat Swarm Optimization* (CSO) guna melakukan optimasi pada jalur fiber optik yang difokuskan pada optimalisasi jalur ODP pada wilayah penelitian. Dengan menggunakan data ODP pelanggan pada Universitas Jember kampus Tegal Boto dimana terdapat 17 ODP yang tersebar pada wilayah penelitian didapatkan jarak antar ODP yang diperoleh dari pengukuran dengan menggunakan *google earth* dan akan dioptimasi dengan algoritma CSO serta melakukan perhitungan nilai redaman total yang dihasilkan dan nilai *power link budget* dari setiap pelanggan yang termasuk dalam wilayah penelitian.

Hasil dari penelitian ini adalah jarak total ODP pada wilayah penelitian yang telah dilakukan optimasi menggunakan metode algoritma *Cat Swarm Optimization* (CSO) sehingga didapatkan jalur ODP yang jaraknya lebih pendek dari pada jalur sebelum dilakukan optimasi. Seperti pada tabel 4.5 dimana perbandingan jarak total dari STO sampai dengan ONT Pelanggan mengalami penurunan dari sebelum

optimasi, pada sampel data pelanggan dengan jarak awal sebelum optimasi sebesar 13302,1 meter dan jarak total setelah dilakukan optimasi sebesar 11929,37 meter dimana perbandingan jarak sebelum dan sesudah optimasi sebesar 1372 meter lebih pendek. Jarak total pada jaringan fiber optik ini akan berpengaruh terhadap nilai redaman total sehingga dapat mempengaruhi nilai daya terima arah *downstream* dengan menggunakan perhitungan *power link budget*, dari sampel data penelitian yang didapat redaman total sebelum optimasi sebesar 21,45574 dan redaman total setelah optimasi adalah 20,97528 lebih kecil dari sebelumnya. Sedangkan untuk nilai daya terima arah *downstream* sebelum optimasi -19,39520 dan sesudah optimasi sebesar -18,91475 mengalami penguatan. Pada perhitungan *power link budget* data daya terima yang diperoleh dapat dijadikan penentuan kelayakan suatu layanan telekomunikasi khususnya jaringan internet.

PRAKATA

Puji syukur atas limpahan rahmat, taufik, dan hidayah yang telah diberikan oleh Allah SWT, sehingga dapat menyelesaikan skripsi yang berjudul “Analisis Optimasi Jaringan *Fiber To The Home* Universitas Jember Dengan *Metode Cat Swarm Optimization*” dengan baik dan tepat waktu. Skripsi ini disusun untuk memenuhi salah satu syarat untuk menyelesaikan pendidikan strata satu (S1) pada Program Studi Teknik Elektro Fakultas Teknik Universitas Jember.

Tersusunnya skripsi ini tidak lepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih yang sebesar-besarnya kepada :

1. Dr. Ir. Triwahju Hardianto, S.T., M.T., selaku Dekan Fakultas Teknik Universitas Jember.
2. Dr. Bambang Sri Kaloko, S.T., M.T., selaku ketua jurusan Teknik Elektro Universitas Jember.
3. Widjonarko, S.T., M.T., selaku ketua program studi S1 Teknik Elektro Fakultas Teknik Universitas Jember.
4. Ir. Widyono Hadi, M.T., selaku Dosen Pembimbing Akademik selama penulis menjadi mahasiswa program studi Teknik Elektro dan penguji 1 skripsi penulis.
5. Ir. Ike Fibriani, S.T., M.T., selaku Dosen Pembimbing Utama dan Andrita Ceriana Eska, S.T., M.T., selaku Dosen Pembimbing Anggota yang telah memberikan masukan dan saran untuk penulis.
6. Widya Cahyadi, S.T., M.T., selaku tim penguji 2 skripsi penulis.
7. Seluruh Dosen dan staf civitas akademik Fakultas Teknik Universitas Jember.
8. Bapak Fajar selaku Manager divisi SDI beserta staff yang telah mengizinkan dan membantu proses pengambilan data kepada penulis
9. Kedua orang tua Ayahanda Joko Purnomo dan Ibunda Dian Wahyu Widiyanti yang selalu memberikan dukungan dan semangat.
10. Keluarga kecil penulis yang senantiasa memotivasi dan memberikan dukungan agar dapat menyelesaikan skripsi dengan baik.

11. Teman-teman jurusan Teknik Elektro angkatan 2016 Fakultas Teknik Universitas Jember.
12. Seluruh asisten laboratorium telekomunikasi dan terapan Fakultas Teknik Universitas Jember.
13. Teman-teman kos Pak Didik jalan kalimantan 18 no.6-8, Jember yang selalu membantu dan memberikan dukungan untuk penulis.
14. Semua pihak yang telah membantu dan memberi ide ataupun dukungan dalam penyelesaian skripsi ini

Semoga skripsi ini dapat bermanfaat dan dijadikan referensi untuk mengerjakan tugas akhir, penulis juga menerima segala bentuk kritik dan saran untuk kesempurnaan skripsi ini.

Jember, 14 Januari 2020

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
PERSEMBAHAN	ii
MOTTO	iii
PERNYATAAN	iv
PENGESAHAN	vi
RINGKASAN	vii
PRAKATA	ix
DAFTAR ISI	xi
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
DAFTAR LAMPIRAN	xvii
BAB 1. PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	4
1.6 Sistematika Penulisan	5
BAB 2. TINJAUAN PUSTAKA	6
2.1 Tinjauan Pustaka	6

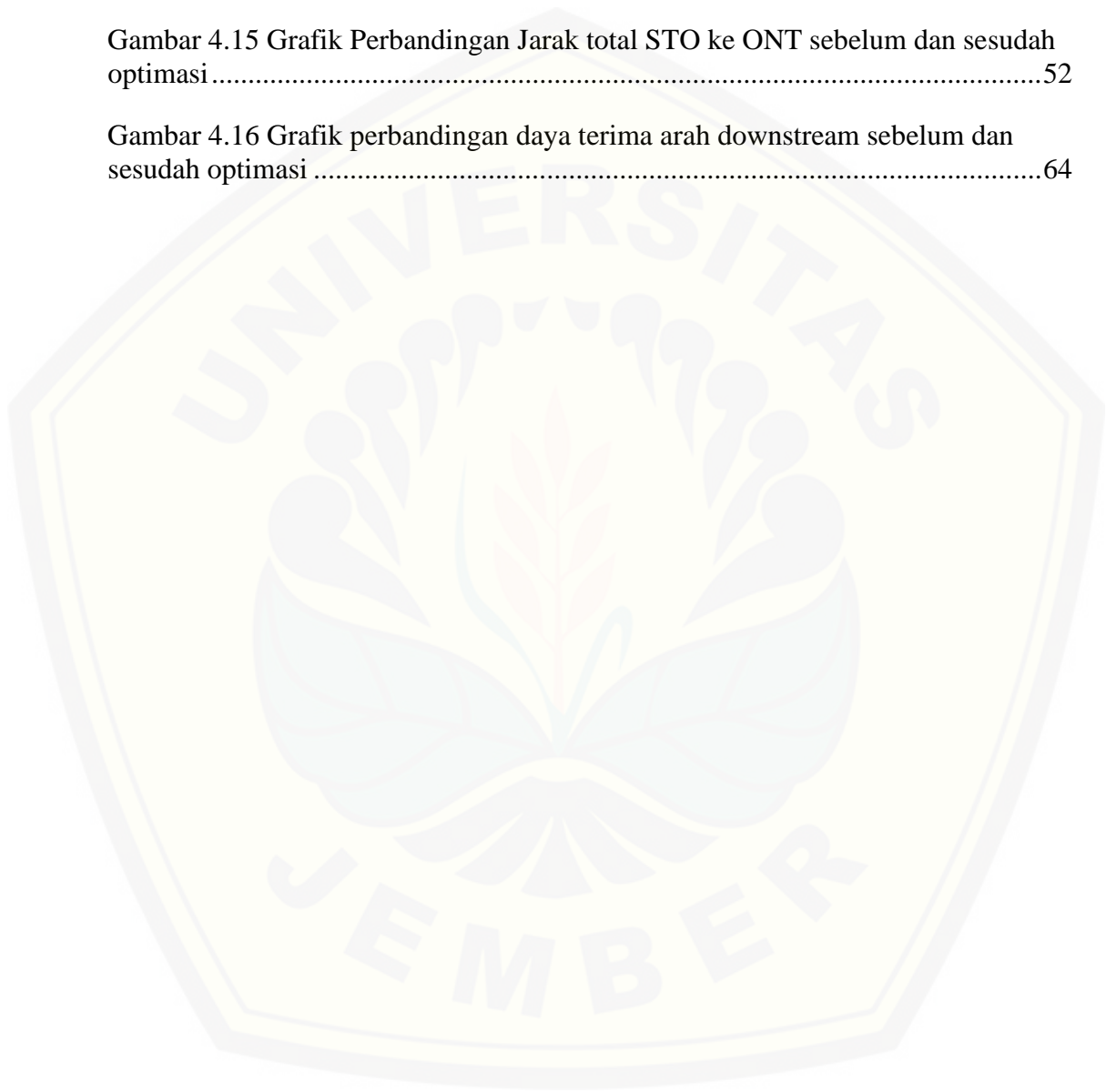
2.1.1 Hasil Penelitian yang Relevan.....	6
2.2 Serat Optik	9
2.3 Gigabyte Passive Optical Network (GPON)	10
2.4 Fiber To The Home (FTTH)	11
2.4.1 Network Management System (NMS)	11
2.4.2 Optikal Distribution Cabinet (ODC)	11
2.4.3 Optical Distribution Point (ODP)	12
2.4.4 Optical Network Termination (ONT).....	13
2.4.5 Optical Line Termination (OLT)	14
2.4.6 Passive Splitter	15
2.4.7 Splice	16
2.4.8 Redaman pada Fiber To The Home (FTTH)	16
2.5 Travelling Salesman Problem (TSP)	17
2.6 Cat Swarm Optimization (CSO)	17
2.6.1 Seeking Mode	18
2.6.2 Tracing Mode	19
2.7 Power Link Budget	20
2.8 Objek Penelitian	21
BAB 3. METODOLOGI PENELITIAN.....	23
3.1 Tempat Penelitian	23
3.2 Waktu Penelitian	23
3.3 Prosedur Penelitian	24
3.4 Diagram Alur Penelitian	26
3.5 Diagram Alur Metode CSO (Cat Swarm Optimization).....	27
3.6 Metode Analisa Data	30

3.7 Parameter Penelitian	30
3.8 Pengujian Hasil	31
3.9 Observasi Awal	31
BAB 4. HASIL DAN PEMBAHASAN	33
4.1 Universitas Jember kampus Tegal Boto	33
4.2 Optimasi Rute ODP dengan <i>Cat Swarm Optimization</i> (CSO)	34
4.2.1 Penentuan Koordinat pada <i>Google Earth</i>	34
4.2.2 Jarak Antar ODP pada <i>Google Earth</i>	36
4.2.3 Optimasi dengan Algoritma <i>Cat Swarm Optimization</i> (CSO) dengan software matlab r2015a	37
4.2.4 Pengujian Algoritma <i>Cat Swarm Optimization</i> (CSO)	42
4.3 Analisa Jarak	47
4.3.1 Sebelum Optimasi	48
4.3.2 Sesudah Optimasi	48
4.3.3 Perbandingan Jarak dan Redaman	48
4.4 Analisa Pengukuran	50
4.4.1 Hasil Perhitungan Jarak	51
4.4.2 Hasil Perhitungan Total pelanggan	61
BAB 5. PENUTUP	70
5.1 Kesimpulan	70
5.2 Saran	70
DAFTAR PUSTAKA	72
LAMPIRAN	73

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Struktur serat optik (Sumber : Samuel, 1988).....	9
Gambar 2.2 Arsitektur GPON (Faruqi & Panjaitan, 2014)	10
Gambar 2.3 ODC PT. Telkom Indonesia	11
Gambar 2.4 ODP PT. Telkom Indonesia	12
Gambar 2.5 Perangkat ONT dari PT. Telkom Indonesia.....	13
Gambar 2.6 Perangkat OLT dari PT. Telkom Indonesia	14
Gambar 2.7 Passive Splitter dengan Rasio 1 : 4	15
Gambar 2.8 Passive Splitter dengan Rasio 1 : 32	15
Gambar 3.1 Denah Lokasi Penelitian (Google Earth Pro, 2019)	22
Gambar 3.2. Diagram Alur Penelitian.....	25
Gambar 3.3 Diagram Alur Metode CSO (<i>Cat Swarm Optimization</i>)	28
Gambar 4.1 Peta Universitas Jember kampus Tegal Boto.....	33
Gambar 4.2 Tampilan GUI Algoritma CSO dengan Matlab	38
Gambar 4.3 Tampilan Keterangan Latitude dan Longtitude pada GUI Matlab	39
Gambar 4.4 Tampilan Keterangan Jarak antar ODP pada GUI Matlab.....	40
Gambar 4.5 Hasil Optimasi Jarak ODP	40
Gambar 4.6 Hasil Proses Pengukuran Jarak Optimasi	41
Gambar 4.7 Grafik pengaruh jumlah iterasi dengan nilai fitness.....	42
Gambar 4.8 Grafik pengaruh jumlah iterasi dengan waktu proses optimasi	43
Gambar 4.9 Gambar jalur ODP optimasi yang terpotong.....	44
Gambar 4.10 Gambar jalur ODP optimasi terbaik	44
Gambar 4.11 Grafik pengaruh jumlah populasi terhadap nilai fitness.....	45

Gambar 4.12 Grafik pengaruh jumlah populasi terhadap waktu optimasi.....	46
Gambar 4.13 Grafik perbandingan jarak total ODP sebelum dan sesudah optimasi	48
Gambar 4.14 Grafik perbandingan redaman total sebelum dan sesudah optimasi	49
Gambar 4.15 Grafik Perbandingan Jarak total STO ke ONT sebelum dan sesudah optimasi.....	52
Gambar 4.16 Grafik perbandingan daya terima arah downstream sebelum dan sesudah optimasi	64

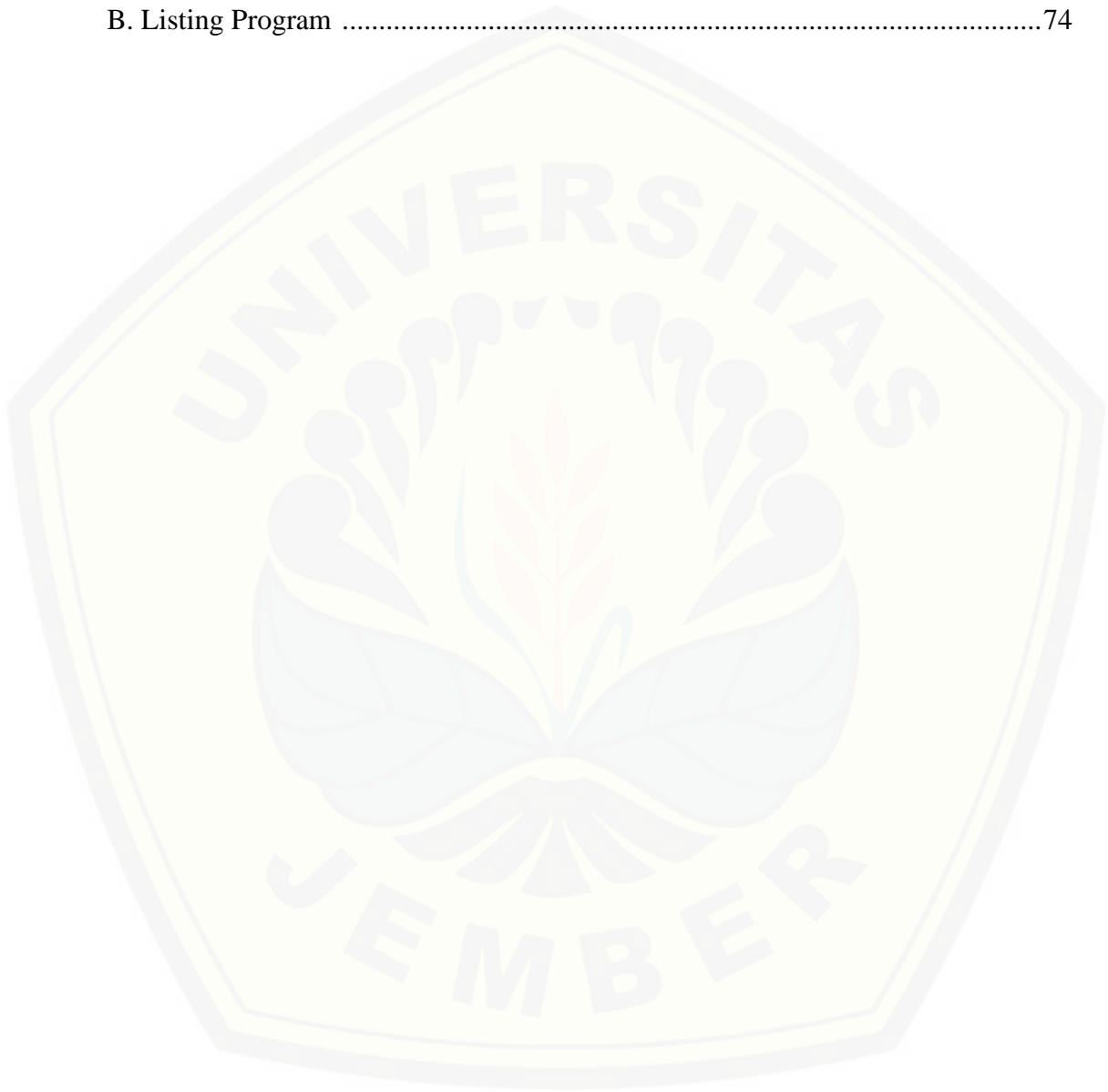


DAFTAR TABEL

	Halaman
Tabel 2.1 Matrik Hasil Penelitian yang Relevan	6
Tabel 2.2 Spesifikasi Perangkat ONT pada PT. Telkom Indonesia.....	13
Tabel 2.3 Spesifikasi Perangkat OLT dari PT. Telkom Indonesia	14
Tabel 2.4. Tabel Redaman <i>Passive Splitter</i>	15
Tabel 3.1 Rencana dan Jadwal Penelitian	23
Tabel 4.1 Nama ODP dan Titik Koordinat ODP pada <i>Google Earth</i>	34
Tabel 4.2 Jarak tiap ODP bagian 1	35
Tabel 4.3 Jarak tiap ODP bagian 2	36
Tabel 4.4 Jarak tiap ODP bagian 3	37
Tabel 4.5 Total Jarak kabel fiber optik dari STO sampai ONT pelanggan.....	51
Tabel 4.6 Spesifikasi kabel Fiber Optik	53
Tabel 4.7 Spesifikasi Sampel Pelanggan Bagian 1	54
Tabel 4.8 Spesifikasi Sampel Pelanggan Bagian 2	55
Tabel 4.9 Spesifikasi Sampel Pelanggan Bagian 3	56
Tabel 4.10 Spesifikasi Sampel Pelanggan Bagian 1	57
Tabel 4.11 Spesifikasi Sampel Pelanggan Bagian 2	58
Tabel 4.12 Spesifikasi Sampel Pelanggan Bagian 3	59
Tabel 4.13 Data pelanggan sebelum optimasi CSO	61
Tabel 4.14 Data pelanggan sesudah optimasi CSO	63
Tabel 4.15 Daya Terima (<i>downstream</i>) Sebelum dan Sesudah Optimasi	67

DAFTAR LAMPIRAN

	Halaman
A. Tabel Jarak Antar ODP Universitas Jember kampus Tegal Boto	73
B. Listing Program	74



BAB 1. PENDAHULUAN

1.1 Latar Belakang

Telekomunikasi merupakan kemajuan teknologi yang sangat diminati pada era modern ini dikarenakan telekomunikasi memiliki peranan penting untuk memungkinkan manusia bertukar informasi satu dengan yang lainnya secara cepat, tepat dan akurat. Begitupun dengan perkembangan telekomunikasi yang semakin maju dan berkembang pesat, meliputi komunikasi yang berupa penyiaran (*broadcast*), komunikasi seluler, dan komunikasi internet. Pada era sebelumnya komunikasi masih diartikan sebagai pertukaran informasi berupa penyedia informasi dan penerima informasi secara langsung sedangkan pada saat ini komunikasi internet sangat cepat berkembang dan sangat diminati di hampir seluruh dunia. Hal ini membuat penyedia layanan internet berlomba-lomba memberikan pelayanan internet terbaik untuk pelanggan dengan memberikan *services* yang dilakukan dengan teknologi terkini. Di Indonesia penyedia layanan internet terbesar yaitu PT. Telkom Indonesia yang memiliki pelanggan telepon tetap sebanyak 15 juta pelanggan dan pelanggan telepon seluler sebanyak 104 juta pelanggan memberikan layanan internet dengan berbagai macam pilihan sesuai dengan kebutuhan pelanggan. Dengan menggunakan teknologi Fiber Optik diharapkan PT. Telkom Indonesia dapat memberikan pelayanan maksimal dengan layanan terkini yaitu Indonesia Digital Home (IndiHome) ke seluruh wilayah Indonesia.

Kebutuhan layanan internet untuk mahasiswa sangat penting guna menunjang aktivitas akademik, khususnya pada Universitas Jember yang menerapkan pendidikan akademik dengan berbagai metode pembelajaran yang mengharuskan mahasiswa dan dosen menggunakan layanan internet. Penggunaan internet di Universitas Jember tidak hanya untuk bertukar informasi ataupun mencari referensi pembelajaran bagi mahasiswa, tetapi layanan internet juga digunakan untuk melakukan pembelajaran secara tidak tatap muka berupa pembelajaran dalam jaringan melalui E-Learning, begitu pula dengan teknologi absensi yang mengharuskan mahasiswa melakukan *scanning Quick Response*

Code (QR Code) pada saat mengikuti mata kuliah dengan aplikasi via *smartphone* yang mengharuskan terhubung dengan layanan internet dan jaringan yang telah ditentukan. Sehingga kebutuhan layanan internet di Universitas Jember khususnya ketika pada hari efektif perkuliahan sangat penting, penggunaan layanan internet pada jam tertentu yang digunakan banyak *user* secara bersamaan dapat mengakibatkan layanan internet tersebut mengalami pelemahan (*down*), hal ini dikarenakan *bandwidth* yang dibagi menjadi beberapa jalur penempatan tidak ditransmisikan secara optimal, maka perlu dilakukan pengoptimalan pada jalur transmisi sehingga dapat meningkatkan *downstream* jaringan dan dapat menekan redaman yang ada pada *Optical Distribution Point* (ODP).

Pada penelitian yang telah dilakukan oleh (Asri Rizky Dyah Vitaloka) dalam skripsi nya yang berjudul Penerapan *Cat Swarm Optimization Algorithm* (CSO) dan *Particle Swarm Optimization Algorithm* (PSO) Dalam Penyelesaian *Travelling Salesman Problem* (TSP). *Travelling Salesman Problem* (TSP) merupakan penyelesaian suatu masalah untuk mencari rute terpendek yang diadopsi dari kegiatan *salesman* pada sebuah perjalanan dengan berbagai kota yang dilalui, sehingga TSP ini dapat mencari solusi yang optimal untuk jalur optimasi dan menemukan secara pasti nilai minimum perjalanan seorang salesman.

Pada penelitian yang dilakukan oleh (I Wayan Radika Apriana, Ni Ketut Tari Tastrawati, Kartika Sari) yang berjudul Implementasi Algoritma *Cat Swarm Optimization* Dalam Menyelesaikan *Job Shop Scheduling Problem* (JSSP). Pada penelitian ini algoritma CSO digunakan untuk optimasi JSSP dengan menggunakan 3 parameter data, kasus yang digunakan adalah kasus penjadwalan 3 job – 2 mesin sebagai validasi program dan kasus nyata penjadwalan 5 job – 12 mesin, variabel yang digunakan dalam penelitian ini adalah waktu penyelesaian tercepat seluruh pekerjaan (*makespan*) dinotasikan dengan C_{max} . Dari penelitian yang telah dilakukan dapat disimpulkan bahwa algoritma CSO efektif dalam menyelesaikan kasus nyata JSSP untuk penjadwalan mesin di industri peralatan pertanian.

Dari beberapa jurnal rujukan yang telah diperoleh dapat memberikan ide dan gagasan kepada penulis untuk melakukan penelitian dengan judul Analisis Optimasi Jaringan *Fiber To The Home* Universitas Jember Dengan Metode *Cat Swarm Optimization* (CSO). Dengan menggunakan beberapa parameter data awal berupa jarak, redaman, transfer data *downstream* dan *upstream* pada beberapa fakultas yang akan diselesaikan menggunakan metode *Cat Swarm Optimization* (CSO) serta melakukan perhitungan *Power Link Budget* pada data yang telah dilakukan optimasi, Diharapkan setelah dilakukan optimasi nilai redaman dan nilai *Power Link Budget* lebih baik dibandingkan sebelum optimasi sehingga didapatkan hasil yang optimal khususnya nilai *downstream* pada beberapa fakultas.

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, berikut beberapa masalah yang akan diselesaikan dalam penelitian ini yaitu :

1. Bagaimana penyelesaian optimasi penentuan jalur transmisi terbaik menggunakan metode *Cat Swarm Optimization* (CSO) pada Universitas Jember ?
2. Bagaimana cara menekan nilai redaman pada jalur transmisi menggunakan *Cat Swarm Optimization* (CSO) pada Universitas Jember ?
3. Bagaimana meningkatkan nilai *downstream* dengan *Power Link Budget* pada Universitas Jember setelah dilakukan optimasi dengan algoritma *Cat Swarm Optimization* (CSO) ?

1.3 Batasan Masalah

Berdasarkan rumusan masalah diatas, supaya pembahasan tidak terlalu luas maka diperlukan suatu batasan masalah yaitu :

1. Data pengukuran dilakukan oleh PT. Telkom Akses Witel Jember pada divisi SDI.
2. Pengukuran parameter data dilakukan pada jaringan FTTH (*Fiber To The Home*) PT. Telkom Indonesia di Universitas Jember Kampus Tegal Boto.

3. Pengambilan data awal berupa pengukuran jarak, koordinat ODP, data pelanggan, nilai redaman, transfer data (*Downstream*).
4. Optimasi jalur transmisi dilakukan dengan algoritma *Cat Swarm Optimization* (CSO) pada total jalur ODP.
5. Parameter pencapaian nilai *downstream* merupakan arah daya terima sebelum dan setelah optimasi.
6. Optimasi penentuan jalur transmisi terbaik menggunakan algoritma *Cat Swarm Optimization* (CSO) yang diselesaikan dengan MATLAB R2015a.

1.4 Tujuan Penelitian

Penelitian pada tugas akhir ini diharapkan mampu memberi manfaat sebagai berikut :

1. Dapat mengetahui jalur transmisi ODP terbaik menggunakan metode *Cat Swarm Optimization* (CSO) pada Universitas Jember kampus Tegal Boto.
2. Dapat mengetahui nilai redaman paling optimal pada jalur transmisi setelah dilakukan optimasi menggunakan metode *Cat Swarm Optimization* (CSO) pada Universitas Jember.
3. Dapat mengetahui nilai optimal *Power Link Budget* setelah dilakukan optimasi menggunakan metode *Cat Swarm Optimization* (CSO) pada Universitas Jember kampus Tegal Boto.

1.5 Manfaat Penelitian

Penelitian ini diharapkan mampu memberikan manfaat seperti tujuan diatas sebagai berikut:

1. Hasil optimasi jalur transmisi ODP dapat meningkatkan layanan internet guna memenuhi kebutuhan internet di Universitas Jember kampus Tegal Boto.
2. Optimasi ini dapat digunakan sebagai acuan untuk memperbaiki nilai redaman pada jalur transmisi sehingga dapat mengoptimalkan nilai *Power Link Budget*.
3. Hasil optimasi dapat meningkatkan nilai *downstream* sehingga dapat digunakan acuan untuk peningkatan pelayanan.

1.6 Sistematika Penulisan

Penyusunan skripsi kali ini memiliki sistematika penulisan sebagai berikut:

BAB 1. PENDAHULUAN

Merupakan latar belakang, rumusan masalah, batasan masalah, tujuan penulisan, manfaat penulisan, dan sistematika penulisan.

BAB 2. TINJAUAN PUSTAKA

Merupakan penjelasan dan teori yang relevan terhadap penelitian yang dilakukan.

BAB 3. METODOLOGI PENELITIAN

Penjelasan tentang metode yang digunakan untuk pengambilan data penelitian.

BAB 4. HASIL DAN PEMBAHASAN

Merupakan hasil dan analisa dari penelitian.

BAB 5. PENUTUP

Merupakan kesimpulan dan saran hasil penelitian.

BAB 2. TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

Pada tinjauan pustaka berikut ini, dibuat sebagai acuan penulis untuk melakukan penelitian terkait tugas akhir. Selain itu tinjauan pustaka dapat menambah wawasan dan dapat mengembangkan ilmu pengetahuan sehingga bidang penelitian ini dapat dikembangkan lebih lanjut.

2.1.1 Hasil Penelitian yang Relevan

Pada hasil penelitian yang relevan berisikan tinjauan pustaka yang telah ada sebelumnya sehingga dapat dijadikan bahan acuan dan perbandingan bagi penulis.

Tabel 2.1 Matrik Hasil Penelitian yang Relevan

No.	Masalah	Solusi	Judul Jurnal	Nama (Tahun)
1.	Menentukan urutan terbaik operasi mesin-mesin dengan waktu penyelesaian seluruh pekerjaan (<i>makespan</i>) yang minimum.	Menentukan solusi acak awal dan parameter data dan digunakan algortima CSO untuk mendapatkan nilai <i>fitness</i> terbaik.	Implementasi algortima <i>cat swarm optimization</i> dalam menyelesaikan <i>Job shop scheduling problem</i> (JSSP).	I Wayan Radika Apriana, Ni Ketut Tari Tastrawati, Kartika Sari (2016).
2.	Mengukur kelayakan redaman kabel serat optik pada setiap <i>Optical Termination Box</i> (OTB) antar STO dengan menggunakan OTDR dan melakukan optimasi redaman.	Data hasil pengukuran dilakukan perhitungan dengan <i>Power Link Budget</i> sehingga nilai redaman sesuai dengan standarisasi PT. Telkom.	Analisis Pengukuran Redaman Kabel Serat Optik Antara Sto Pemangkat – Sto Tebas Menggunakan Otdr Exfo Ftb-200	Indra Lesmana, Dasril, Dedy Suryadi (2018).
3.	Perancangan FTTH dengan teknologi GPON pada wilayah atau jalur baru.	Melakukan perancangan awal jaringan FTTH dan uji kelayakan dengan <i>Power Link Budget</i> .	Implementasi Jaringan Fiber To The Home (Ftth) Dengan Teknologi Gigabit Passive	Sunarsan Sitohang, Sabbram Agus Setiawan (2018).

			Optical Network (Gpon).	
4.	Optimasi jaringan FTTH guna mendukung JARLOKAF.	Menganalisa jalur transmisi dan merancang jalur terbaik serta melakukan perhitungan <i>Power Link Budget</i> .	Analisis Jaringan FttH (Fiber To The Home) Berteknologi Gpon (Gigabit Passive Optical Network)	Brilian Dermawan, Imam Santoso, and Teguh Prakoso (2016).
5.	Pencarian rute jaringan <i>Fiber To The Home</i> paling baik dan optimasi redaman.	Menggunakan optimasi dengan <i>Particle Swarm Optimization (PSO)</i> dan perhitungan <i>Power Link Budget</i> .	Optimasi rute jaringan <i>Fiber To The Home (FTTH)</i> perumahan bernady land Jember menggunakan metode <i>Particle Swarm Optimization (PSO)</i> .	Evhi Nur Imamah (2018).
6.	Penyelesaian masalah <i>Travelling Salesman Problem (TSP)</i> menggunakan 2 metode yang berbeda.	Melakukan penyelesaian dengan menggunakan metode <i>Particle Swarm Optimization Algorithm (PSO)</i> dan <i>Cat Swarm Optimization Algorithm (CSO)</i> .	Penerapan <i>Cat Swarm Optimization Algorithm (CSO)</i> dan <i>Particle Swarm Optimization Algorithm (PSO)</i> dalam penyelesaian <i>Travelling Salesman Problem (TSP)</i> .	Asri Rizky Dyah Vitaloka (2016).
7.	Penyelesaian <i>Travelling Salesman Problem (TSP)</i> .	Dengan menentukan parameter berupa operasi yang di gunakan pada algoritma CSO dan modus pencarian (<i>Seeking Mode</i>)	<i>Discrete Cat Swarm Optimization</i> to Resolve the <i>Traveling Salesman Problem</i>	Abdelhamid Bouzidi, Mohammed Essaid Riffi (2013)

Dari berbagai sumber hasil penelitian diatas pada tabel 1, penelitian yang dilakukan oleh I Wayan Radika Apriana, Ni Tetut Tari Tastrawati dan Kartika Sari (2016) dapat disimpulkan bahwa *Cat Swarm Optimization Algorithm* (CSO) memberikan solusi untuk menentukan waktu minimum yang digunakan untuk penyelesaian seluruh pekerjaan dan Menentukan solusi acak awal dan parameter data menggunakan algoritma CSO untuk mendapatkan nilai *fitness* terbaik. Selanjutnya penelitian Indra Lesmana, Dasril dan Dedy Suryadi (2018) melakukan pengukuran redaman kabel serat optik pada setiap OTB untuk mengetahui kelayakan menggunakan OTDR dan dilakukan perhitungan *Power Link Budget* sehingga dapat mengetahui kelayakan redaman sesuai dengan standarisasi yang ada. Selanjutnya penelitian dari Sunarsan Sitohang dan Sabbram Agus Setiawan (2018) yaitu membuat rancangan FTTH dengan teknologi GPON pada jalur transmisi baru yang di analisis menggunakan *Power Link Budget* untuk uji kelayakan. Selanjutnya penelitian dari Brilian Dermawan, Imam Santoso, dan Teguh Prakoso (2016) melakukan optimasi jaringan FTTH yang ada agar mendapatkan nilai *throughput* yang memadai untuk mendukung JARLOKAF dengan menganalisa jalur transmisi dan melakukan uji kelayakan dengan perhitungan *Power Link Budget*. Lalu pada penelitian yang dilakukan Evhi Nur Imamah (2018) melakukan optimasi rute terbaik jaringan FTTH dengan metode *Particle Swarm Optimization* (PSO) dan perhitungan *Power Link Budget* untuk uji kelayakan. Selanjutnya pada penelitian oleh Asri Rizky Dyah Vitaloka (2016) melakukan analisa penyelesaian masalah *Travelling Salesman Problem* (TSP) menggunakan metode *Particle Swarm Optimization Algorithm* (PSO) dan *Cat Swarm Optimization Algorithm* (CSO) didapatkan kesimpulan bahwa CSO lebih optimal untuk menyelesaikan masalah optimasi. Kemudian penelitian internasional oleh Abdelhamid Bouzidi dan Mohammed Essaid Riffi (2013) tentang penyelesaian *Travelling Salesman Problem* (TSP) dengan menentukan parameter berupa operasi yang di gunakan pada algoritma CSO dan modus pencarian (*Seeking Mode*).

Berdasarkan penjelasan beberapa jurnal dan penelitian diatas dapat disimpulkan beberapa poin untuk meningkatkan *downstream* pada jaringan internet khususnya *Fiber To The Home*. Maka dari itu, penulis akan melakukan penelitian

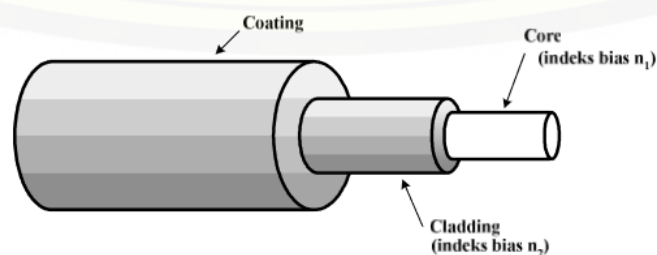
tentang Analisis Optimasi Jaringan *Fiber To The Home* Universitas Jember Dengan Metode *Cat Swarm Optimization* (CSO).

2.2 Serat Optik

Serat optik adalah media saluran transmisi yang terbuat dari kaca yang sangat tipis dan memiliki diameter core sekitar $2\ \mu\text{m}$ sampai dengan $50\ \mu\text{m}$ yang digunakan untuk mentransmisikan sinyal cahaya dari suatu tempat ke tempat lain. Serat optik dapat mengirimkan sinyal dengan mengubah kedalaman bentuk cahaya dengan kecepatan $3 \times 10^8\ \text{m/s}$. Berdasarkan mode transmisi yang digunakan serat optik terdiri atas *Multimode Step Index*, *Multimode Graded Index*, dan *Singlemode Step Index*.

Keuntungan serat optik adalah lebih murah, bentuknya lebih ramping, kapasitas transmisi yang lebih besar, sedikit sinyal yang hilang, data diubah menjadi sinyal cahaya sehingga lebih cepat, tenaga yang dibutuhkan sedikit, dan tidak mudah terbakar. Kelemahan serat optik antara lain biaya yang mahal untuk peralatannya, memerlukan konversi data listrik ke cahaya dan sebaliknya yang rumit, memerlukan peralatan khusus dalam prosedur pemakaian dan pemasangannya, serta untuk perbaikan yang kompleks membutuhkan tenaga yang ahli di bidang ini.

Serat optik yang sering digunakan untuk media transmisi yaitu ITU-T G.652.D dan G.657.A yang memiliki fungsi masing-masing untuk kabel feeder dan kabel distribusi. Adapun rugi-rugi yang terdapat pada kabel optik tersebut, dengan panjang gelombang 1310 nm yaitu sebesar $\leq 0.35\ \text{dB}$ setiap kilo meter dan untuk panjang gelombang 1490 nm sebesar $\leq 0.28\ \text{db}$ setiap kilo meter.

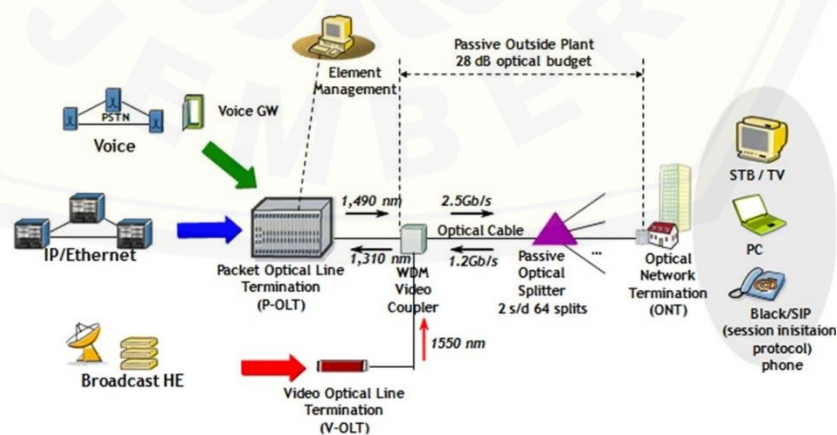


Gambar 2.1. Struktur serat optik (Sumber : Samuel, 1988)

Pada serat optik memiliki 2 bagian utama yang sangat penting, yaitu bagian *core* dan *cladding* yang terbuat dari bahan silika kaca yang berkualitas tinggi dan bebas air. *Core* memiliki indeks bias yang lebih besar daripada *cladding* ($n_1 > n_2$) sampai dengan batas kritis, sehingga dapat terjadi pembiasan dalam total refleksi sehingga cahaya yang ditransmisikan tetap merambat didalam *core* hingga ke ujung serat. *Coating* (*jacket*) berfungsi sebagai pelindung *core* dan *cladding* dari tekanan fisik luar, terbuat dari bahan plastik atau karet yang sangat tebal dan berkualitas (Zanger, 1991, Thomas, 1995, Samuel, 1988).

2.3 Gigabyte Passive Optical Network (GPON)

GPON adalah teknologi jaringan optik berbasis Passive Optical Network (PON) yang distandarisi oleh ITU-T. G.984. GPON mampu memberikan layanan dengan kecepatan 2,4.Gbps Downstream dan 1,2 untuk Upstream. Jarak antara OLT (Optical Line Terminal) dengan ONT (Optical Network Terminal) mencapai 20 km. Ciri khas dari teknologi ini dibanding teknologi optik lainnya semacam SDH (Synchronous Digital Hierarchy) adalah teknik distribusi traffic nya dilakukan secara pasif. Dari sentral hingga ke arah subscriber akan didistribusikan menggunakan pasif splitter (1:2, 1:4, 1:8, 1:16, 1:32, 1:64, 1:128). Teknologi ini mampu memberikan jaringan yang *cost-effective*, *flexible pada provisioning voice* maupun *data service* yang *reliable* berbasis pada *optical access network* (Pratama, Sukadarmika, & Sudiarta, 2017).



Gambar 2.2. Arsitektur GPON (Faruqi & Panjaitan, 2014)

2.4 Fiber To The Home (FTTH)

Fiber To The Home (FTTH) merupakan suatu proses untuk mentransmisikan sinyal optik dari penyedia data ke pengguna dengan menggunakan media transmisi berupa fiber optik. FTTH merupakan teknologi yang dapat digunakan untuk menghemat biaya operasi dan dapat memberikan pelayanan services yang baik kepada pelanggan penyedia jaringan data. Untuk mentransmisikan sinyal optik dengan teknologi FTTH diperlukan perangkat jaringan sebagai berikut.

2.4.1 Network Management System (NMS)

Merupakan perangkat lunak yang berfungsi untuk mengontrol dan konfigurasi perangkat yang bekerja pada teknologi GPON. NMS bekerja untuk mengkonfigurasi perangkat OLT yang berfungsi sebagai titik akhir serat optik dan ONT yang berfungsi sebagai penyedia interface antara jaringan serat optik kepada pelanggan.

2.4.2 Optikal Distribution Cabinet (ODC)

ODC merupakan perangkat pasif yang tidak dapat dipindah letakan yang biasanya berada diluar STO tetapi juga terdapat ODC yang didalam ruang (*Indoor*), ODC berfungsi sebagai titik terminasi ujung kabel *feeder* dan pangkal distribusi, ODC ini berfungsi membagi jaringan serat optik yang berasal dari kabel dengan kapasitas besar (*feeder*) menjadi bagian-bagian kecil yang digunakan untuk distribusi data ke pengguna ataupun pelanggan, pada ODC terdapat bagian pembagi yang disebut splitter, splitter merupakan peralatan yang berbentuk terminal yang berfungsi untuk membagi satu jalur menjadi beberapa jalur data.



Gambar 2.3. ODC PT. Telkom Indonesia

2.4.3 *Optical Distribution Point (ODP)*

ODP merupakan perangkat pasif yang berada pada jaringan luar ruang (*outdoor*), biasanya ODP terletak menempel pada tiang kabel telepon atau menggantung diantara kabel serat optik yang dibagi. Berfungsi sebagai tempat instalasi sambungan jaringan serat optik dengan *single mode* untuk membagi satu kabel serat optik dengan kapasitas besar (*feeder*) untuk keperluan pelanggan atau sebagai titik terminasi kabel drop optik dengan splitter. Penggunaan kabel pada ODP tergantung dengan kapasitas sambungan yang digunakan, biasanya memiliki kapasitas lebih kecil berkisar 6 sampai dengan 48 core tergantung jenis dan kabel yang digunakan.



Gambar 2.4. ODP PT. Telkom Indonesia

2.4.4 Optical Network Termination (ONT)

Optical Network Termination (ONT) atau ONU merupakan perangkat yang menyediakan *interface* antara jaringan serat optik dengan pelanggan, alat ini berfungsi untuk mengubah sinyal data yang berbentuk cahaya ke sinyal elektrik. Pada data ITU-T G.984 ONT memiliki *upstream* sebesar 1,2 Gbps dan *downstream* sebesar 2,4 Gbps. Pada tabel 2 merupakan spesifikasi dari perangkat ONT (Dermawan, Santoso, & Prakoso, 2016).

Tabel 2.2. Spesifikasi Perangkat ONT pada PT. Telkom Indonesia

Parameter	Spesifikasi	Unit
<i>Downstream Rate</i>	2,4	Gbps
<i>Upstream Rate</i>	1,2	Gbps
<i>Spectrum Width</i>	1	nm
<i>Downlink Wavelength</i>	1490	nm
<i>Uplink Wavelength</i>	1310	nm
<i>Video Wavelength</i>	1550	nm
<i>Optical Rise Time</i>	200	ps

Pada gambar 2.5 merupakan foto dari perangkat ONT yang dimiliki PT. Telkom Indonesia.



Gambar 2.5. Perangkat ONT dari PT. Telkom Indonesia

2.4.5 *Optical Line Termination (OLT)*

OLT merupakan perangkat pada jaringan serat optik yang berfungsi sebagai jaringan terakhir atau sering disebut (*end-point*) dari layanan jaringan fiber optik. OLT juga berfungsi untuk mengubah sinyal elektrik menjadi sebuah sinyal optik, perangkat ini terhubung langsung dengan metro *Ethernet* yang ada di setiap STO. OLT yang digunakan pada PT. Telkom Indonesia dengan jenis ITU-T G.859. Tabel 3 merupakan spesifikasi dari perangkat OLT (Dermawan, Santoso, & Prakoso, 2016).

Tabel 2.3. Spesifikasi Perangkat OLT dari PT. Telkom Indonesia

Parameter	Spesifikasi	Unit
<i>Optical Transmit Power</i>	5	dBm
<i>Downlink Wavelength</i>	1490	nm
<i>Uplink Wavelength</i>	1310	nm
<i>Video Wavelength</i>	1550	nm
<i>Spectrum Width</i>	1	nm
<i>Downstream Rate</i>	2,4	Gbps
<i>Upstream Rate</i>	1,2	Gbps
<i>Optical Rise Time</i>	160	ps

Pada gambar 2.6 merupakan foto perangkat OLT yang dimiliki PT. Telkom Indonesia, terlihat beberapa sambungan kabel optik yang terhubung dengan modul pada OLT.



Gambar 2.6. Perangkat OLT dari PT. Telkom Indonesia

2.4.6 *Passive Splitter*

Passive Splitter merupakan port atau perangkat yang berfungsi untuk membagi satu buah serat optik menjadi beberapa bagian, *Passive Splitter* juga dapat berfungsi untuk membagi sinyal optik. *Passive Splitter* sendiri memiliki berbagai jenis yang disesuaikan dengan kebutuhan pembagian fiber optik, dari jenis tersebut

memiliki nilai redaman yang berbeda beda sesuai dengan kapasitas setiap jenisnya. Seperti pada perangkat ODC biasanya menggunakan *Passive Splitter* dengan kapasitas 1 : 4 sedangkan untuk perangkat ODP menggunakan *Passive Splitter* dengan kapasitas 1 : 8. Berikut ini merupakan tabel parameter redaman maksimal pada *Passive Splitter* .(Zainal Arifin, 2016).

Tabel 2.4. Tabel Redaman *Passive Splitter*

Rasio	Redaman
1 : 2	2,8 – 4,0 dB
1 : 4	5,8 – 7,5 dB
1 : 8	8,8 – 11,0 dB
1 : 16	10,7 – 14,4 dB
1 : 32	14,6 – 18,0 dB



Gambar 2.7. *Passive Splitter* dengan Rasio 1 : 4



Gambar 2.8. *Passive Splitter* dengan Rasio 1 : 32

2.4.7 *Splice*

Pada saat melakukan penyambungan kabel fiber optik untuk media transmisi layanan internet biasanya dilakukan beberapa sambungan atau *splice* pada setiap ujung fiber optik ataupun ketika menggunakan *splitter* juga terdapat sambungan atau *splice*. Sambungan ini memiliki redaman yang harus diperhatikan agar cahaya masih bisa ditransmisikan pada fiber optik. Pada penyambungan menggunakan *fusion splicer* harus memiliki redaman maksimal sebesar 0,1 dB. Jika redaman melebihi batas standar tersebut maka dapat mengurangi atau tidak dapat mentransmisikan cahaya sehingga dilakukan penyambungan ulang.

2.4.8 Redaman pada *Fiber To The Home (FTTH)*

Pada perancangan jaringan FTTH, besar kecil nya redaman pada jalur transmisi sangat berpengaruh agar tidak terjadi gangguan dari media transmisi

ketika fiber optik digunakan. Terdapat batas redaman untuk perancangan FTTH agar tidak mengganggu kecepatan transmisi. Redaman minimum yang digunakan sebesar 13 dB dan untuk nilai redaman maksimum sebesar 25 dB. Jika nilai redaman lebih dari batas maksimum dan kurang dari batas minimum maka dapat diindikasikan bahwa terdapat masalah pada jalur transmisi fiber optik. Berikut merupakan cara menghitung redaman pada fiber optik. (Idham, Tadarus, & Wildan, 2016).

$$\text{Redaman G652D} = \text{Jarak (km)} \times 0,35 \text{ dB} \quad (2.1)$$

$$\text{Redaman Splice} = 0,1 \text{ dB} \times n \text{ Splice} \quad (2.2)$$

$$\text{Redaman Splitter 1:4} = \text{maksimum } 7,25 \text{ dB} \quad (2.3)$$

$$\text{Redaman Splitter 1:8} = \text{maksimum } 11,0 \text{ dB} \quad (2.4)$$

$$\begin{aligned} \text{Redaman Total} &= \text{Redaman Kabel OLT sampai ODC} + \text{Redaman Kabel} \\ &\quad \text{ODC sampai ODP} + \text{Redaman Kabel ODP sampai} \\ &\quad \text{ONT} + \text{Redaman Splitter ODC} + \text{Redaman Splitter} \\ &\quad \text{ODP} + \text{Redaman Splice} \end{aligned} \quad (2.5)$$

2.5 Travelling Salesman Problem (TSP)

Travelling Salesman Problem (TSP) merupakan salah satu cara menyelesaikan masalah untuk mendapatkan jalur terpendek untuk mencapai suatu tujuan, TSP dikembangkan dari permasalahan salesman yang terus berjalan dari titik satu ke titik yang lain guna mendapatkan jalur terbaik dan efisien sampai dengan kembali ke titik semula dengan menggunakan biaya seminimum mungkin. TSP sendiri berfungsi untuk meminimalisir biaya dan mendapatkan jalur terpendek dan efisien. (Fatmawati, Prihandono, & Noviani, 2015).

2.6 Cat Swarm Optimization (CSO)

Cat Swarm Optimization (CSO) merupakan algoritma yang dikembangkan oleh Shu-Chan Chu dan Pei-Wei Tsai pada tahun 2006, algoritma ini didapatkan dari pengamatan perilaku kucing dengan kawanannya yang waspada dan selalu mengintai keadaan sekelilingnya, dari sekumpulan kucing tersebut dan perilakunya

digunakan sebagai penyelesaian masalah optimasi. Algoritma *Cat Swarm Optimization* (CSO) pada penerapannya dibagi menjadi dua sub mode yang dibagi berdasarkan perilaku kucing yaitu *seeking mode* dan *tracking mode*. (Dhanasaputra dan Santosa, 2010).

Algoritma *Cat Swarm Optimization* (CSO) dapat digunakan untuk optimasi suatu permasalahan seperti halnya optimasi jarak, algoritma CSO dapat dituliskan seperti pada persamaan berikut :

$$V'_{k,d} = V_{k,d} + r_{k,d} \times [(X_{best,d} - X_{k,d})] - r_{2k,d} [(x_{k,d} + v'_{k,d})] \quad (2.6)$$

Algoritma CSO akan bekerja dengan melakukan update posisi populasi kucing dimana nilai k dan d akan berkelanjutan, serta akan didapatkan nilai posisi terbaik dari kandidat yang terpilih dan akan dibangkitkan sesuai dengan nilai konstanta pada rentan tertentu.

2.6.1 *Seeking Mode*

Seeking Mode pada algoritma *Cat Swarm Optimization* (CSO) digunakan untuk memodelkan keadaan dimana ketika kucing dalam keadaan istirahat, kucing saat keadaan istirahat selalu melihat sekelilingnya dan mencari posisi selanjutnya untuk bergerak dan istirahat pada posisi yang lain. Pada *Seeking Mode* terdapat 4 faktor terpenting yaitu, *seeking memory pool* (SMP) merupakan faktor yang berfungsi untuk menentukan memori pencarian setiap kucing yang diindikasikan sebagai titik-titik yang telah dituju oleh kucing, selanjutnya kucing akan memilih titik yang pernah dituju dengan beberapa aturan yang ditentukan. *Seeking range of the selected dimension* (SRD) merupakan faktor yang berfungsi menyatakan perpindahan dalam dimensi atau tujuan yang terpilih, jika setiap dimensi dinyatakan telah berpindah, maka nilai selisih dari kedua poin tersebut tidak menjauhi rentan yang diberikan, rentan yang dimaksud merupakan batasan yang didefinisikan oleh SRD. Selanjutnya faktor *Self-Position Considering* (SPC) yang merujuk pada pada variable benar atau salah, pada faktor ini akan membuat keputusan kepada pergerakan kucing untuk melakukan pergerakan pada titik titik yang pernah dituju kucing. (Dhanasaputra dan Santosa, 2009).

Berikut ini merupakan langkah-langkah *seeking mode* yang dilalui untuk membuat keputusan dengan CSO dideskripsikan sebagai berikut :

- a. Membangkitkan j tiruan dari posisi awal kucing, dimana $j = \text{SMP}$. Jika nilai SPC benar maka nilai $j = (\text{SMP}-1)$, kemudian jadikan posisi benar sebagai kandidat.

$$\text{Posisi awal} = \frac{i}{n \text{ Populasi}} \quad (2.7)$$

- b. Untuk setiap tiruan disesuaikan dengan CDC, CDC merupakan dimensi yang akan berubah, tambah atau kurangi nilai SRD persen dari nilai hasil secara acak untuk menggantikan nilai sebelumnya.

$$\text{Jumlah modifikasi} = \text{CDC} * n \quad (2.8)$$

- c. Hitung nilai yang cocok (FS) untuk titik kandidat.
- d. Jika FS berbeda maka hitung kemungkinan terpilih setiap kandidat dengan persamaan (2.9).
- e. Pilih secara acak titik bergerak dari setiap titik kandidat, dan pindahkan posisi awal ke posisi kandidat.

$$\text{Probabilitas Terpilih}_i = \frac{|fitness_i - fitness_{max}|}{fitness_{max} - fitness_{min}} \quad (2.9)$$

2.6.2 Tracking Mode

Tracking Mode merupakan langkah untuk menggambarkan keadaan kucing ketika sedang mengikuti jejak targetnya. Ketika kucing tersebut memasuki *tracking mode* maka akan bergerak dengan dimensi yang telah ditentukan.

Ada 3 tahap untuk menentukan *tracking mode* pada suatu data sebagai berikut:

- a. Memperbarui nilai kecepatan untuk setiap dimensi yang digunakan ($V_{k,d}$) dengan persamaan (2.10).
- b. Analisa nilai dalam keadaan maksimum atau tidak, jika melebihi nilai yang ditentukan maka sesuaikan dengan batas maksimal.
- c. Memperbarui posisi kucing dengan persamaan (2.11).

$$V'_{k,d} = V_{k,d} + r \times c \times (X_{best,d} - X_{k,d}), \quad (2.10)$$

Dimana, $k=1,2,3, \dots, m$ dan $d=1,2,3, \dots, n$

Pada persamaan (2.10) $X_{best,d}$ merupakan posisi dimana kucing memiliki keadaan paling cocok, pada $X_{k,d}$ merupakan posisi awal, c adalah nilai konstanta dan r adalah nilai yang diperoleh pada rentan $[0,1]$.

$$x'_{k,d} = x_{k,d} + v'_{k,d} \quad (2.11)$$

Dari dua model rasio yang didapat harus dilakukan rasio campuran atau *mixture ratio* (MR) untuk menggabungkan dari kedua mode diatas. Dari penjelasan sebelumnya diketahui bahwa perubahan kucing dengan perlahan ketika sedang istirahat merupakan *seeking mode*. Dan perilaku kucing ketika mengikuti mangsanya secara cepat adalah *tracking mode*.

Pada dasarnya proses algoritma CSO dapat dilakukan dengan 6 tahap berikut :

- a. Membangkitkan jumlah kucing sebanyak n dalam proses mencari titik terbaik.
- b. Lakukan penyebaran kucing pada calon solusi secara acak dan tentukan rentan kecepatan maksimum, ambil sampel kucing untuk dilakukan proses *tracking mode* dan sisa sampel dimasukkan ke *seeking mode*.
- c. Analisa posisi kucing dalam fungsi kecocokan, simpan kucing terbaik ($x_{best,d}$) yang merupakan solusi terbaik.
- d. Pindahkan kucing sesuai dengan mode yang telah ditentukan
- e. Lakukan secara berulang langkah (b) sampai mendapat beberapa sampel yang diinginkan.
- f. Perhatikan *terminating condition* atau kondisi akhir jika hasil memuaskan ulangi langkah (c) sampai dengan (e).

2.7 Power Link Budget

Power Link Budget merupakan langkah perhitungan untuk mengetahui batas redaman total yang masih dapat di toleransi, pada perhitungan ini untuk

memastikan nilai daya yang diterima oleh penerima memiliki nilai lebih besar atau sama dengan batas minimum. Perhitungan *Power Link Budget* didasari oleh standarisasi ITU-T G.984 dan juga standarisasi yang digunakan oleh PT. Telkom Indonesia yaitu nilai redaman total tidak boleh melebihi -28 dBm dengan jarak maksimum 20 kilometer. Serta nilai margin pada daya transmisi harus memiliki nilai lebih dari 0 agar memenuhi standarisasi atau uji kelayakan. Margin daya adalah daya yang masih tersisa dari power transmit setelah dikurangi dengan *loss* selama proses transmisi, pengurangan nilai *safety margin* dan pengurangan nilai sensitifitas *receiver* (Umar Fachreza, 2016).

Dapat dilakukan perhitungan dengan persamaan berikut :

$$P_{rx} = P_{tx} - (af + ac + as + M) \quad (2.12)$$

Berdasarkan :

P_{rx} = Daya sinyal diterima (dBm)

P_{tx} = Daya dari sumber cahaya (dBm)

af = Redaman serat optik ((panjang kabel (km) x *loss* kabel (0,35)).

ac = Redaman pada konektor (jumlah konektor \times *loss* konektor (0,5))

as = Redaman pada *splicer* (jumlah *splice* \times *loss splice* (0,1))

M = Nilai Kompensasi fiber optik

Untuk spesifikasi level terima pada perangkat PT. Telkom Indonesia terletak pada batas level terima -10 dBm sampai dengan -30 dBm (Melyana Dwi Haryani, 2017).

2.8 Objek Penelitian

Penelitian ini dilakukan untuk mendapatkan jalur transmisi yang paling optimal sehingga dapat menekan nilai redaman yang terjadi pada jalur transmisi serta meningkatkan nilai *downstream* pada jaringan. Penelitian ini dilakukan di wilayah Universitas Jember meliputi beberapa fakultas dan gedung-gedung yang

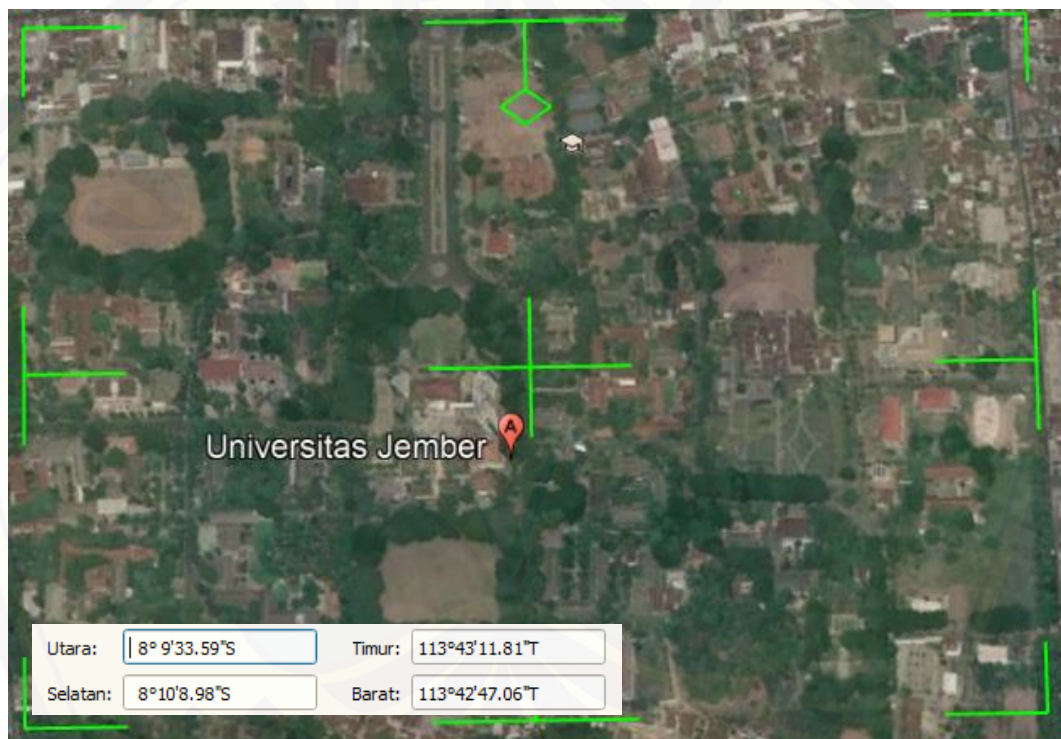
terinstalasi jaringan fiber optik yang merupakan cakupan wilayah kerja PT. Telkom Indonesia Witel Jember khususnya STO Jember 2. Pada wilayah instalasi FTTH Universitas Jember telah terdapat beberapa *Optical Distribution Point* (ODP) dan *Optical Distribution Cabinet* (ODC) yang dapat dijadikan titik untuk mengambil sampel data.

Penelitian dilakukan dengan pengambil data yang dimulai dari STO Jember 2 yang beralamatkan di Jalan Letjend Suprpto No. 70 , Kebonsari, Jember. Lokasi penelitian di Universitas Jember ini dipilih karena diperlukannya optimasi jaringan FTTH untuk meningkatkan nilai *throughput* pada wilayah Universitas Jember dikarenakan penggunaan layanan internet yang semakin tinggi sehingga perlu adanya peningkatan layanan guna mendukung kegiatan akademis di Universitas Jember.

BAB 3. METODOLOGI PENELITIAN

3.1 Tempat Penelitian

Penelitian ini dilaksanakan di wilayah Universitas Jember kampus Tegal Boto meliputi beberapa fakultas, lembaga, dan UPT yang terinstalasi jaringan fiber optik yang merupakan cangkupan wilayah kerja PT. Telkom Indonesia Witel Jember khususnya STO Jember 2. Penelitian ini direncanakan akan dilakukan pada bulan September 2019 sampai dengan selesai.



Gambar 3.1. Denah Lokasi Penelitian (*Google Earth Pro*, 2019)

3.2 Waktu Penelitian

Waktu penelitian dan pembuatan laporan ini dimulai pada bulan September 2019 – Desember 2019.

3.3 Prosedur Penelitian

Dalam metodologi penelitian yang dilakukan, terdapat beberapa langkah pelaksanaan penelitian sebagai berikut :

1. Studi Literatur

Studi Literatur merupakan tahap awal penelitian dengan melakukan *research* dan mencari literatur dari penelitian sebelumnya yang relevan melalui buku, internet, maupun jurnal nasional dan internasional untuk mengetahui dasar dari penelitian yang akan dilakukan. Pada tahap ini diharapkan dapat memberikan komponen penelitian sehingga meminimalisir kesalahan saat penelitian.

2. Pengambilan Data

Tahapan kedua dari penelitian ini yaitu melakukan pengambilan data awal yang digunakan untuk parameter penelitian, melakukan klasifikasi data yang diperlukan dan mengambil komponen data yang akan digunakan dalam optimasi. Pengambilan data juga disesuaikan dengan metode *Cat Swarm Optimization* (CSO) sehingga dapat mempermudah tahap selanjutnya.

3. Pengolahan Data

Tahap ketiga dalam penelitian ini yaitu melakukan pengolahan data yang didapat dari pengukuran data awal yang dikembangkan dalam metode yang digunakan sehingga dapat menghasilkan data optimasi yang diinginkan, pada tahap ini data yang telah diolah diubah menjadi sebuah laporan hasil penelitian.

4. Analisis Data

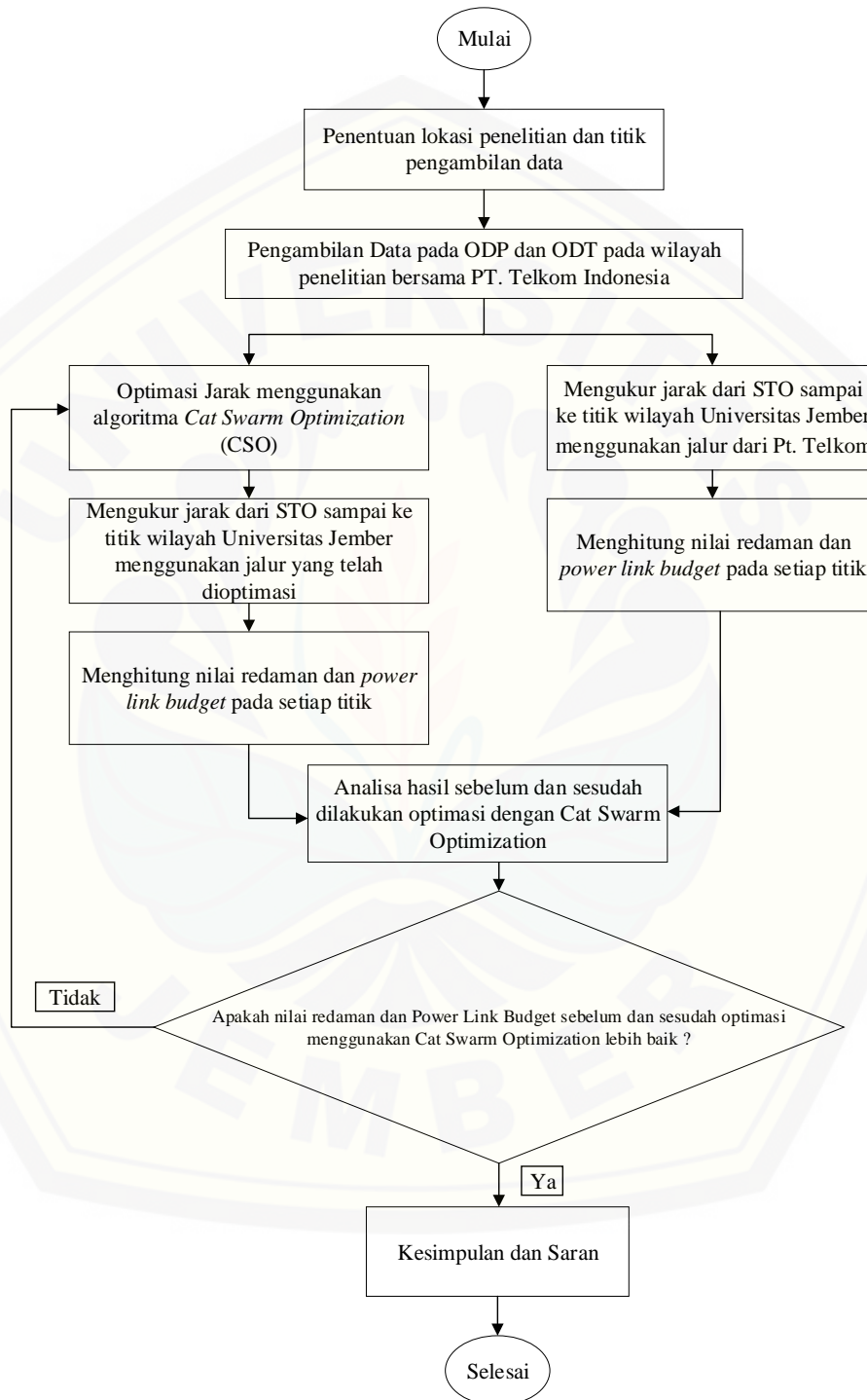
Pada tahap ini, data yang telah diambil dan diolah menggunakan metode yang diusulkan akan dianalisis dan dibandingkan, sehingga akan diketahui bahwa *Cat Swarm Optimization* (CSO) dapat melakukan optimasi pada data dan memiliki nilai yang lebih tinggi dari data yang belum dilakukan optimasi.

5. Pembuatan laporan

Pada tahap ini melakukan pembuatan laporan dari hasil penelitian yang dilakukan serta mengambil kesimpulan berdasarkan hasil analisis data dan perhitungan menggunakan metode yang sesuai. Pembuatan laporan juga memperhatikan saran – saran untuk pertimbangan untuk pengembangan penelitian. Diperlukan pengambilan kesimpulan dan saran dari analisa data penelitian yang didapat.



3.4 Diagram Alur Penelitian



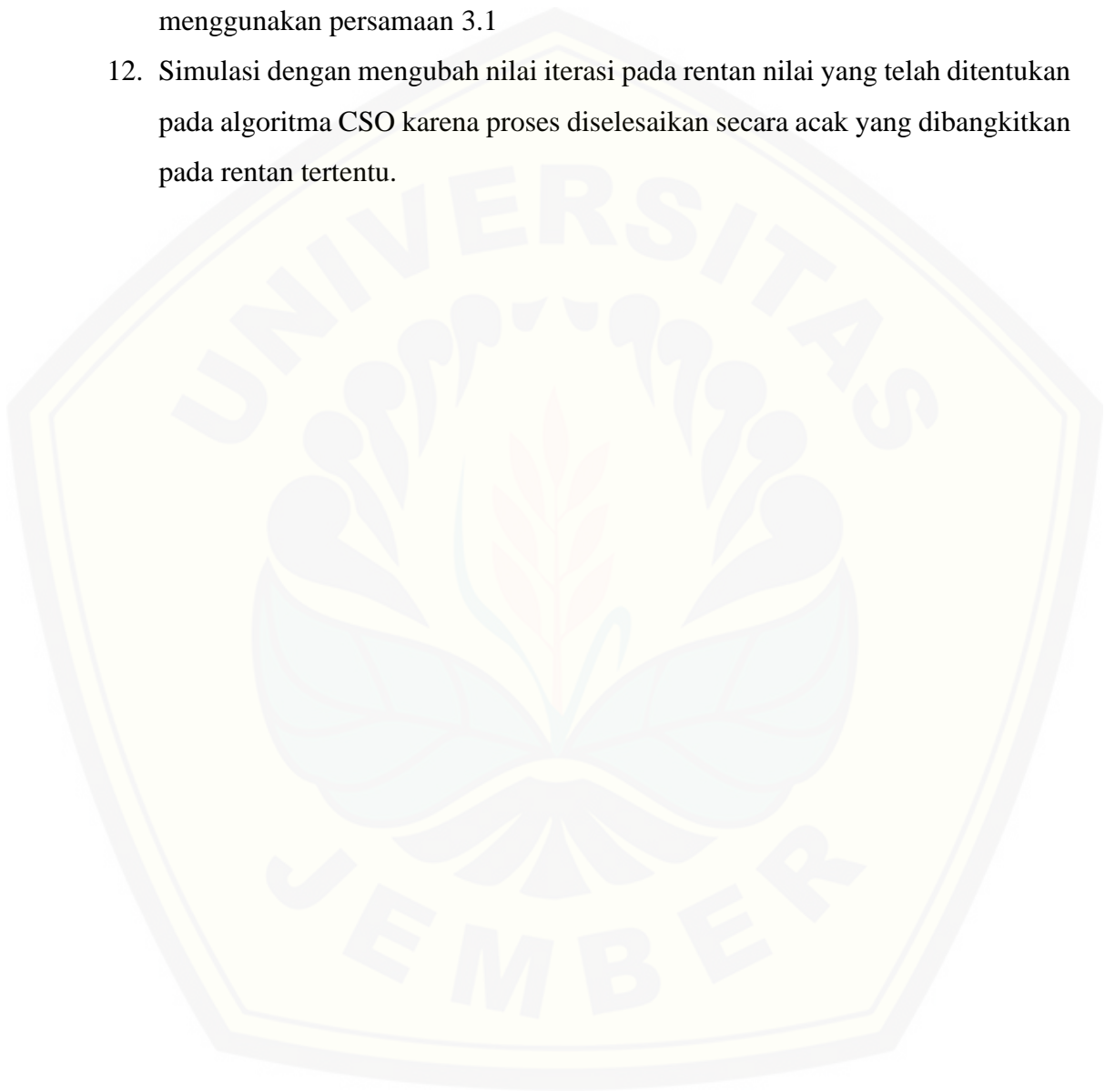
Gambar 3.2. Diagram Alur Penelitian

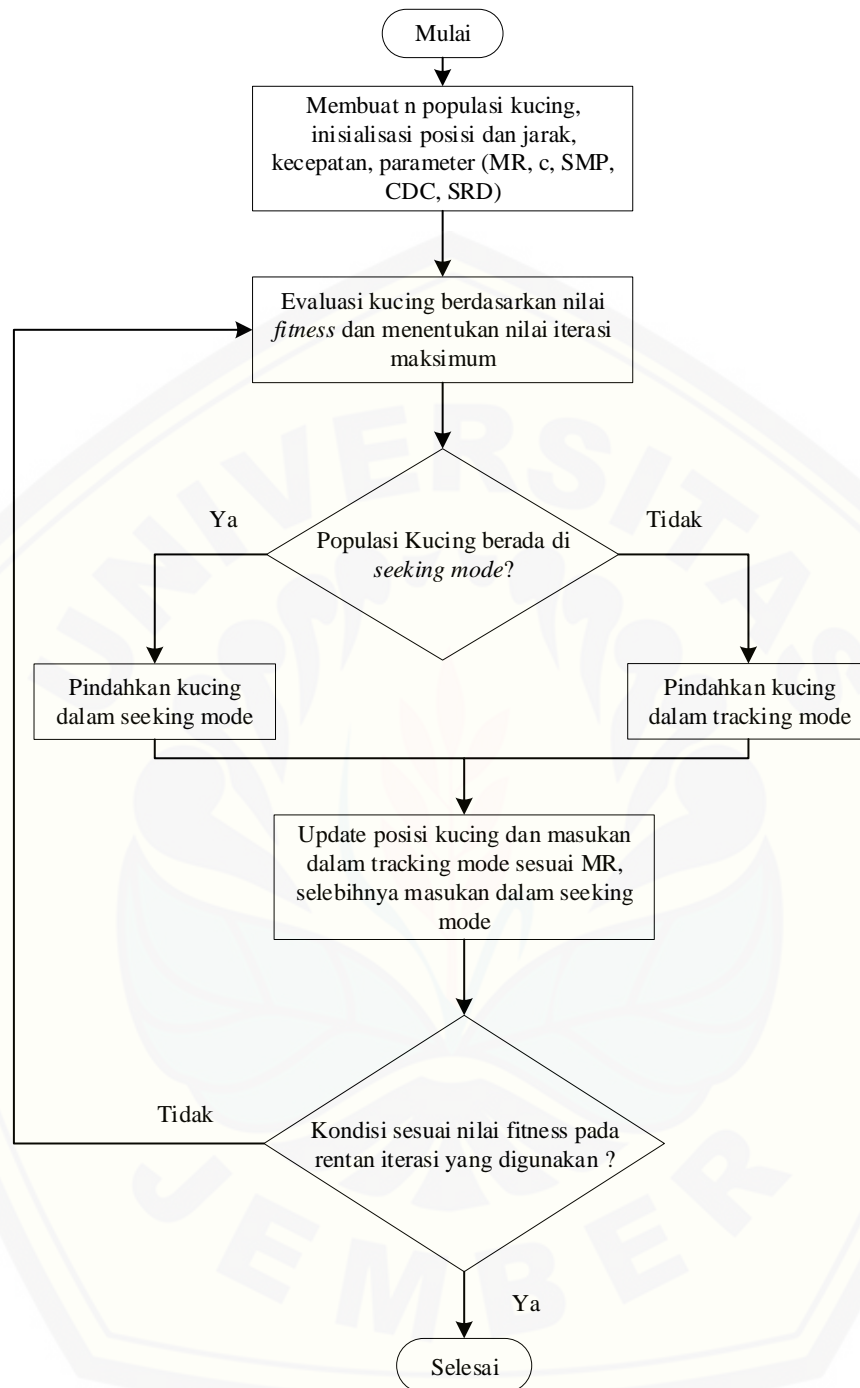
3.5 Diagram Alur Metode CSO (*Cat Swarm Optimization*)

Pada Gambar 3.1 Diagram Alur Penelitian CSO (*Cat Swarm Optimization*) adapun langkah-langkah yang harus dilakukan untuk mencari jalur transmisi ODP terpendek dengan menggunakan metode CSO (*Cat Swarm Optimization*). Dengan mengadopsi masalah *Travelling Salesman Problem* (TSP) maka akan dapat diselesaikan dengan menggunakan metode CSO dengan langkah-langkah sebagai berikut :

1. Menentukan parameter awal berupa rentan iterasi yang akan dijalankan, input data berupa data kolektif pada format .xlsx yang mana *sheet* pertama merupakan koordinat letak ODP dan *sheet* kedua merupakan total jarak ODP.
2. Memasukan parameter - parameter yang digunakan untuk program CSO seperti n populasi kucing, jumlah iterasi, Mixture Ratio (MR), c, SMP, CDC, dan SRD.
3. Inisialisasi posisi awal kucing, kecepatan yang digunakan untuk simulasi dan poin - poin yang memungkinkan kucing berada di titik tersebut dengan menggunakan persamaan 2.7.
4. Menentukan dimensi simulasi, menggunakan nilai sumbu *X* dan sumbu *Y* sebagai jalur *seeking* dan *tracking* kucing dengan persamaan 2.8.
5. Evaluasi nilai fungsi dari setiap kucing ($f(C_1)$) dimana menggunakan persamaan 2.10.
6. Tentukan P_{best} dan G_{best} awal.
7. Melakukan *seeking mode* atau pencarian, jika kucing berada pada *Tracking mode*, pindahkan pada *seeking mode* dan perlu diperhatikan untuk nilai MR harus relatif kecil agar kucing berada pada *seeking mode*.
8. *Update* posisi kucing yang berada pada *seeking mode* atau *tracking mode* sampai mencapai kondisi terbaik sementara.
9. Evaluasi nilai fungsi setiap kucing, dan tentukan kondisi terbaik = G_{best} dan dapat dilakukan penentuan kondisi terbaik dengan persamaan 2.9.

10. Jika iterasi telah maksimal maka berhenti, dan jika belum lakukan langkah ke dua. Proses iterasi maksimal akan menghasilkan jarak terpendek dari metode CSO (*Cat Swarm Optimization*).
11. Menghitung nilai fitness yang didapat tiap individu pada iterasi tertentu dengan menggunakan persamaan 3.1
12. Simulasi dengan mengubah nilai iterasi pada rentan nilai yang telah ditentukan pada algoritma CSO karena proses diselesaikan secara acak yang dibangkitkan pada rentan tertentu.



Gambar 3.3 Diagram Alur Metode CSO (*Cat Swarm Optimization*)

3.6 Metode Analisa Data

Pada penelitian ini mengacu pada jaringan *Fiber To The Home* yang mana pada perancangannya dibutuhkan perhitungan nilai redaman yang berfungsi untuk meminimalkan gangguan transmisi yang diakibatkan dari jalur transmisi yang digunakan. Ada batas maksimum dan minimum yang harus dipenuhi pada saat membangun jaringan FTTH agar tidak terjadi gangguan yang fatal. Redaman minimum sebesar 13 dB dan untuk redaman maksimum sebesar 25 dB, jika redaman tidak memenuhi nilai minimum dan melebihi angka maksimum maka dapat dipastikan terdapat gangguan pada jalur transmisinya.

1. Melakukan perhitungan redaman pada kabel fiber optik dimulai dari OLT yang berada pada STO Telkom Jember 2 sampai dengan ONT pelanggan dengan ketentuan redaman setiap kilometer nya 0,35 db, dapat dilakukan perhitungan dengan persamaan 2.1.
2. Melakukan perhitungan sambungan (*splice*) fiber optik dari data yang didapatkan pada data pelanggan dengan ketentuan 0,1 dB setiap satu sambungan (*splice*) dengan menggunakan persamaan 2.2.
3. Perhitungan redaman akibat splitter dibagi menjadi dua jenis, splitter dengan rasio 1 : 4 memiliki nilai maksimal redaman 7,25 dB dan untuk rasio 1 : 8 memiliki nilai maksimal redaman 11,0 dB, redaman splitter dapat dihitung sesuai konektor yang digunakan dengan persamaan 2.11.
4. Sehingga didapatkan nilai redaman total dari panjang fiber optik beserta parameter yang mempengaruhi dengan menggunakan persamaan 2.5.

3.7 Parameter Penelitian

Power link budget merupakan cara yang dilakukan untuk mengetahui batas nilai redaman total yang sesuai ketentuan, perhitungan ini dilakukan berdasarkan standar dari ITU-T G.984 serta ketentuan dan aturan yang diterapkan oleh PT. Telkom Indonesia yaitu dengan jarak tidak melebihi 20 kilo meter maka nilai redaman total tidak boleh lebih dari 28 dB. Nilai P_{rx} atau daya sinyal yang diterima dapat dijadikan pedoman bahwa nilai tersebut juga akan berpengaruh terhadap nilai *downstream* setiap jaringan. Adapun langkah – langkah untuk menentukan

parameter penelitian *Power Link Budget* dapat dihitung dengan persamaan sebagai berikut :

1. Daya sinyal yang diterima dalam dBm merupakan parameter hasil akhir dari proses analisa data yang dilakukan, nilai daya terima atau P_{rx} dapat dihitung dengan *power link budget* dari persamaan 2.12.
2. Nilai daya kirim atau P_{tx} merupakan nilai daya yang dipancarkan atau dikirimkan oleh penyedia jasa layanan dalam dBm.
3. Redaman kabel serat optik atau *af* didapatkan dengan menghitung keseluruhan jarak dalam kilometer dikalikan *loss* kabel (0,35). Seperti persamaan 2.1.
4. Redaman pada konektor atau sering disebut splitter (*ac*) dapat dilakukan perhitungan dengan mengkalikan jumlah konektor dengan *loss* konektor (0,5).
5. Redaman pada splicer dapat dihitung dengan menggunakan persamaan 2.2.
6. Dari perhitungan *power link budget* ini diberikan ketetapan nilai kompensasi redaman pada serat optik (M) sebesar 2 dBm sesuai dengan standarisasi yang berlaku.

3.8 Pengujian Hasil

Pada tahap pengujian hasil ditentukan dengan melakukan simulasi percobaan sebanyak jumlah iterasi yang digunakan. Pengujian hasil berfungsi untuk memastikan nilai yang didapat dari optimasi benar optimal dan memiliki nilai *fitness* yang paling tinggi dari hasil percobaan, nilai *fitness* optimal akan berbanding lurus dengan rute terpendek yang didapatkan.

Berikut merupakan fungsi rumus dari nilai *fitness* :

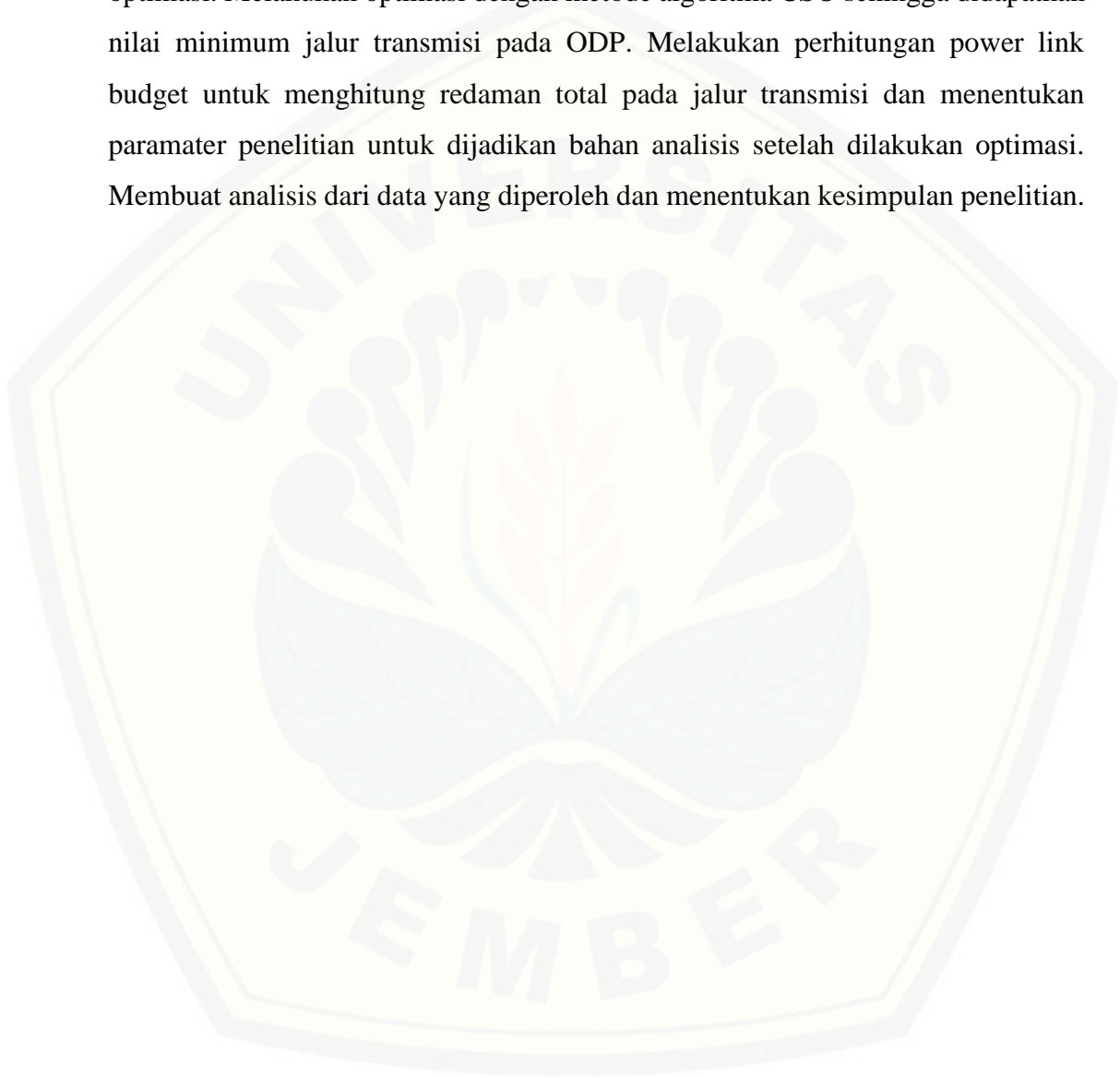
$$fitness = \frac{1}{f_x} \quad (3.1)$$

f_x merupakan nilai dari total jarak antar *Optical Distribution Point* (ODP).

3.9 Observasi Awal

Melakukan pengumpulan data berupa denah dan letak ODP dan ODC yang berada pada wilayah Universitas Jember kampus Tegal Boto yang diperoleh dari

PT. Telkom Indonesia Akses Witel Jember, melakukan pengukuran jarak dengan menggunakan *Google Earth Pro* untuk menentukan seberapa panjang jalur transmisi. Selanjutnya, menentukan parameter yang digunakan untuk metode CSO yaitu jarak ODP dan titik-titik lokasi yang akan dijadikan nilai awal sebelum optimasi. Melakukan optimasi dengan metode algoritma CSO sehingga didapatkan nilai minimum jalur transmisi pada ODP. Melakukan perhitungan power link budget untuk menghitung redaman total pada jalur transmisi dan menentukan parameter penelitian untuk dijadikan bahan analisis setelah dilakukan optimasi. Membuat analisis dari data yang diperoleh dan menentukan kesimpulan penelitian.



BAB 5. PENUTUP

5.1 Kesimpulan

Dari penelitian yang telah dilakukan didapatkan kesimpulan bahwa :

1. Jarak total ODP setelah dilakukan optimasi menggunakan *Cat Swarm Optimization* (CSO) memiliki nilai jarak yang lebih pendek dari jarak total sebelum optimasi. Dari tabel 4.5 total Jarak kabel fiber optik dari STO sampai ONT pelanggan sebelum dan sesudah optimasi memiliki perbedaan nilai jarak lebih kecil sebesar 1372,73 meter atau 1,37273 kilometer dibandingkan jarak sebelum optimasi.
2. Nilai redaman dipengaruhi oleh jarak total ODP sebelum dan sesudah dilakukan optimasi dengan metode algoritma CSO. Dapat ditinjau dari tabel 4.13 dan tabel 4.14 dimana pada redaman total setelah optimasi jarak ODP memiliki nilai yang lebih kecil sebesar 0,48045 dBm dari pada sebelum optimasi jarak ODP.
3. Nilai *power link budget* pada daya terima arah *downstream* memiliki nilai yang lebih baik setelah dilakukan optimasi menggunakan *Cat Swarm Optimization* (CSO). Ditinjau dari tabel 4.13 dan 4.14 nilai daya terima sesudah optimasi jalur ODP lebih besar sebesar 0,48046 dari pada sebelum dilakukan optimasi jarak ODP. Nilai *power link budget* merupakan parameter untuk menentukan kelayakan layanan telekomunikasi.

5.2 Saran

Untuk penelitian selanjutnya, diharapkan dapat mempertimbangkan saran berikut :

1. Dapat dilakukan simulasi menggunakan *software optisystem* dari nilai sebelum dan sesudah optimasi dengan *Cat Swarm Optimization* (CSO) untuk membandingkan *Signal to Noise Rasio* (SNR)

2. Mengganti lokasi penelitian dengan lokasi yang lebih kompleks untuk jaringan *Fiber To The Home* (FTTH) seperti halnya lingkungan perkotaan atau *Fiber To The Building* (FTTB).



DAFTAR PUSTAKA

- Apriana, I.W.R., Ni Ketut, T.T., Kartika, Sari. (2016). Implementasi algoritma *Cat Swarm Optimization* dalam menyelesaikan *Job Shop Scheduling Problem (JSSP)*. E Jurnal Matematika Vol. 5, 90-97.
- Bouzidi, Abdelhamid. Mohammed. E.R., (2013). Discrete *Cat Swarm Optimization* to Resolve the *Traveling Salesman Problem*. International Journal of Advanced Research in Computer Science and Software Engineering, 2277.
- Dermawan, Brilian, Imam, S., Teguh. S. (2016). Analisis Jaringan FTTH (Fiber To The Home) Berteknologi GPON (Gigabit Passive Optical Network). Transmisi 18 (1), 2407-6422.
- Faruqi, I., & Panjaitan, S.P. (2014). Studi Perancangan Jaringan Akses *Fiber To The Home (FTTH)* dengan Menggunakan *Teknologi Gigabit Passive Optical Network (GPON)* di Perumahan Cbd Polonia Medan. SINGUDA ENSIKOM, 25-59.
- Imamah, Nur Evhi. (2018). Optimasi rute jaringan *Fiber To The Home (FTTH)* perumahan bernady land Jember menggunakan metode *Particle Swarm Optimization (PSO)*. Digital Repository Universitas Jember.
- Lesmana, Indra. Dasril, Dedy, S. (2018). Analisis Pengukuran Redaman Kabel Serat Optik Antara STO Pemangkat – STO Tebas Menggunakan OTDR Exfo Ftb-200.
- Pratama, I. P., Sukadarmika, G., & Sudiarta, P.K. (2017). Perancangan Jaringan *Fiber To The Home (FTTH)* Menggunakan teknologi *Gigabyte Passive Network (GPON)* pada Mall Park23 Tuban. Teknologi Elektro,60-65.
- Sitohang, Sunarsan., Sabbram A.S. (2018). Implementasi Jaringan Fiber To The Home (Ftth) Dengan Teknologi Gigabit Passive Optical Network (Gpon). Jurnal SIMETRIS Vol. 9, 2549-3108.
- Vitaloka, Asri Rizky Dyah. (2016). Penerapan *Cat Swarm Optimization Algorithm (CSO)* dan *Particle Swarm Optimization Algorithm (PSO)* dalam penyelesaian *Travelling Salesman Problem (TSP)*. Digital Repository Universitas Jember.

LAMPIRAN A

Tabel Jarak Antar ODP Universitas Jember kampus Tegal Boto

ODP	19	18	17	15	14	13	12	11	10	9	8	7	6	4	3	2	1	0
19	0	23,34	255,62	426,17	600,75	792,2	711,02	79,04	281,38	310,48	466,74	592,08	294,3	671,1	633	534,9	740,87	59,5
18	23,34	0	279,57	450,12	624,7	816,15	734,97	55,7	257,95	287,22	443,3	568,49	318,25	695,05	656,95	558,85	764,82	83,45
17	255,62	279,57	0	346,6	482,72	675,7	631,99	337,04	563,44	569,57	728,12	849,02	430,92	807,72	769,62	671,52	877,49	196,12
15	426,17	450,12	346,6	0	169,19	359,57	315,46	507,59	733,99	740,12	898,67	1019,57	601,47	978,27	940,17	842,07	1048,04	366,67
14	600,75	624,7	482,72	169,19	0	330,53	342,35	682,17	908,57	914,7	1073,25	1194,15	776,05	1152,85	1114,75	1016,65	1222,62	541,25
13	792,2	816,15	675,7	359,57	330,53	0	562,75	873,61	1100,01	1106,14	1264,69	1385,59	967,49	1344,29	1306,19	1208,09	1414,06	732,69
12	711,02	734,97	631,99	315,46	342,35	562,75	0	792,44	1018,84	1024,97	1183,52	1304,42	886,32	1263,12	1225,02	1126,92	1332,89	651,52
11	79,04	55,7	337,04	507,59	682,17	873,61	792,44	0	227,63	232,19	387,3	509,5	375,72	752,52	714,42	616,32	822,29	140,92
10	281,38	257,95	563,44	733,99	908,57	1100,0	1018,8	227,63	0	211,59	366,86	489,9	602,12	978,92	940,82	842,72	1048,69	367,32
9	310,48	287,22	569,57	740,12	914,7	1106,1	1024,9	232,19	211,59	0	214,11	338,94	608,25	985,05	946,95	848,85	1054,82	373,45
8	466,74	443,3	728,12	898,67	1073,2	1264,6	1183,5	387,3	366,86	214,11	0	345,29	766,8	1143,6	1105,5	1007,4	1213,37	532
7	592,08	568,49	849,02	1019,5	1194,1	1385,5	1304,4	509,5	489,9	338,94	345,29	0	887,67	1264,47	1226,37	1128,24	1334,24	652,87
6	294,3	318,25	430,92	601,47	776,05	967,49	886,32	375,72	602,12	608,25	766,8	887,67	0	525,51	429,63	328,4	541,64	234,8
4	671,1	695,05	807,72	978,27	1152,8	1344,2	1263,1	752,52	978,92	985,05	1143,6	1264,47	525,51	0	297,36	310,49	523,73	611,55
3	633	656,95	769,62	940,17	1114,7	1306,1	1225,0	714,42	940,82	946,95	1105,5	1226,37	429,63	297,36	0	213,97	427,21	573,5
2	534,9	558,85	671,52	842,07	1016,6	1208,0	1126,9	616,32	842,72	848,85	1007,4	1128,24	328,4	310,49	213,97	0	213,24	475,37
1	740,87	764,82	877,49	1048,0	1222,6	1414,0	1332,8	822,29	1048,69	1054,82	1213,37	1334,24	541,64	523,73	427,21	213,24	0	681,37
0	59,5	83,45	196,12	366,67	541,25	732,69	651,52	140,92	367,32	373,45	532	652,87	234,8	611,55	573,5	475,37	681,37	0

LAMPIRAN B

Listing Program

```
% PROGRAM UNTUK MENGHITUNG JARAK DARI DATA MASUKAN CalculateDist.m

function Jarak = CalculateDist(Urutan, Data_Dist)
Node = length(Urutan);
Jarak = 0;
for i = 1:Node-1
    Jarak = Jarak + Data_Dist(Urutan(i),Urutan(i+1));
end
Jarak = Jarak + Data_Dist(Urutan(end),Urutan(1));

% PROGRAM UNTUK PLOT RUTE DARI TITIK KOORDNIAT Plot_Route.m

function Plot_Route(Route, Data_Node, Data_Dist)
Data_Kode = Data_Node(:,1);
Data_Coor = Data_Node(:,2:3);

% Koordinat (x,y)
Coor = [Data_Coor(:,2) Data_Coor(:,1)];
% Longitude (x), Latitude (y)
Node = length(Route);

RuteMap = zeros(Node);
for i = 1:Node-1
    RuteMap(Route(i),Route(i+1)) = Data_Dist(Route(i),Route(i+1));
end
RuteMap(Route(end),Route(1)) = Data_Dist(Route(end),Route(1));
gplot(RuteMap,Coor,'-r');
% gplot -> graph plot : menggambarkan graf

% Setting tampilan axes
axis image
% Setting ukuran sebenarnya
inlat = max(Coor(:,2))-min(Coor(:,2));
inlon = max(Coor(:,1))-min(Coor(:,1));
for i = 1:Node

line(Coor(i,1),Coor(i,2),'Marker','s','MarkerEdgeColor','k','MarkerFaceColor','g','MarkerSize',5); % Menambahkan titik
    text(Coor(i,1),Coor(i,2),sprintf(['N'
num2str(Data_Kode(i))]),'FontSize',8,'Color','b'); %
Menambahkan nama titik
end
line(Coor(Route(1),1),Coor(Route(1),2),'Marker','s','MarkerEdgeColor','k','MarkerSize',15); % Menambahkan tanda titik awal
% Tambahkan jarak antar titik
for i = 1:Node-1
    xn = (Coor(Route(i),1)+Coor(Route(i+1),1))/2;
    yn = (Coor(Route(i),2)+Coor(Route(i+1),2))/2;
```



```
text(xn,yn,num2str(Data_Dist(Route(i),Route(i+1))), 'FontSize',8, 'Fontweight', 'bold');
end
xn = (Coor(Route(end),1)+Coor(Route(1),1))/2;
yn = (Coor(Route(end),2)+Coor(Route(1),2))/2;
text(xn,yn,num2str(Data_Dist(Route(end),Route(1))), 'FontSize',8, 'Fontweight', 'bold');

xlim([min(Coor(:,1))-inlon/10 max(Coor(:,1))+inlon/8])
% Setting batas sumbu x
ylim([min(Coor(:,2))-inlat/10 max(Coor(:,2))+inlat/10])
% Setting batas sumbu y
set(gca, 'FontSize',8, 'Fontweight', 'bold');
xlabel('Longitude'); ylabel('Latitude');

% PROGRAM UNTUK ALGORITMA CSO ProgramCSO.m

function varargout = ProgramCSO(varargin)
% PROGRAMCSO MATLAB code for ProgramCSO.fig
% PROGRAMCSO, by itself, creates a new PROGRAMCSO or raises
the existing
% singleton*.
%
% H = PROGRAMCSO returns the handle to a new PROGRAMCSO or
the handle to
% the existing singleton*.
%
% PROGRAMCSO('CALLBACK',hObject,eventData,handles,...) calls
the local
% function named CALLBACK in PROGRAMCSO.M with the given
input arguments.
%
% PROGRAMCSO('Property','Value',...) creates a new PROGRAMCSO
or raises the
% existing singleton*. Starting from the left, property
value pairs are
% applied to the GUI before ProgramCSO_OpeningFcn gets
called. An
% unrecognized property name or invalid value makes property
application
% stop. All inputs are passed to ProgramCSO_OpeningFcn via
varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
% instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help ProgramCSO

% Last Modified by GUIDE v2.5 14-Nov-2019 15:17:09

% Begin initialization code - DO NOT EDIT
```

```
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @ProgramCSO_OpeningFcn, ...
                  'gui_OutputFcn',  @ProgramCSO_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before ProgramCSO is made visible.
function ProgramCSO_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to ProgramCSO (see VARARGIN)

% Choose default command line output for ProgramCSO
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);
clc;
% Bersihkan Command Window
movegui(gcf, 'center');
% Pindah GUI ke tengah layar
set(handles.pushbutton4, 'string', 'Jalur');
set(gcf, 'Name', 'Program CSO');
% Setting header name
set(handles.uipanel1, 'visible', 'on');
% Menampilkan panel data
set(handles.uipanel2, 'visible', 'off');
% Menyembunyikan panel jalur
set(handles.uitable1, 'data', [], 'UserData', [], 'rowname', 0:10);
% Reset tabel koordinat
set(handles.uitable2, 'data', [], 'UserData', [], 'rowname', 0:10, 'column
name', 0:10); % Reset tabel jarak
set(handles.edit1, 'string', '');
% Reset parameter
set(handles.edit2, 'string', '');
set(handles.edit3, 'string', '');
set(handles.edit4, 'string', '');
set(handles.edit5, 'string', '');
```

```
set(handles.edit6,'string','');
set(handles.edit7,'string','');
cla(handles.axes1,'reset');
% Reset axes 1 (grafik kekonvergenan)
set(handles.axes1,'XLim',[0 1000],'YLim',[0
500],'FontSize',8,'Fontweight','bold');
xlabel('Iterasi'); ylabel('Total Jarak');
cla(handles.axes2,'reset');
% Reset axes 2 (gambar jalur)
set(handles.axes2,'XTick',[],'YTick',[]);
set(handles.listbox1,'string',' ','value',1,'UserData',[]);
% Reset listbox
% UIWAIT makes ProgramCSO wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = ProgramCSO_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject     handle to listbox1 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns
listbox1 contents as cell array
%           contents{get(hObject,'Value')} returns selected item from
listbox1

% --- Executes during object creation, after setting all
properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject     handle to listbox1 (see GCBO)
% eventdata   reserved - to be defined in a future version of
MATLAB
% handles     empty - handles not created until after all
CreateFcns called

% Hint: listbox controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
% Bersihkan Command Window
[File, Path] = uigetfile({'*.xlsx;*.xls','Excel Spreadsheet
(*.xlsx, *xls)'}, 'Open File'); % Buat window open file
if File ~= 0
    [Cust, txt] = xlsread(fullfile(Path,File),1);
% Baca excel file sheet 1 (koordinat)

set(handles.uitable1, 'data', Cust(:,2:3), 'UserData', Cust, 'rowname',
Cust(:,1)); % Tampilkan data koordinat
    [Dist, txt] = xlsread(fullfile(Path,File),2);
% Baca excel file sheet 2 (jarak)

set(handles.uitable2, 'data', Dist(2:end,2:end), 'UserData', Dist(2:en
d,2:end), ...

'rowname', Dist(2:end,1), 'columnname', Dist(1,2:end), 'columnwidth', '
auto'); % Tampilkan data jarak
    set(gcf, 'Name', ['Program CSO - ' fullfile(Path,File)]);
% Setting header name
end

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
% Bersihkan Command Window
movegui(gcf, 'center');
% Pindah GUI ke tengah layar
set(handles.pushbutton4, 'string', 'Jalur');
set(gcf, 'Name', 'Program CSO');
% Setting header name
set(handles.uipanel1, 'visible', 'on');
% Menampilkan panel data
set(handles.uipanel2, 'visible', 'off');
% Menyembunyikan panel jalur
set(handles.uitable1, 'data', [], 'UserData', [], 'rowname', 0:10);
% Reset tabel koordinat
set(handles.uitable2, 'data', [], 'UserData', [], 'rowname', 0:10, 'colum
nname', 0:10); % Reset tabel jarak
```

```
set(handles.edit1,'string','');
% Reset paramater
set(handles.edit2,'string','');
set(handles.edit3,'string','');
set(handles.edit4,'string','');
set(handles.edit5,'string','');
set(handles.edit6,'string','');
set(handles.edit7,'string','');
cla(handles.axes1,'reset');
% Reset axes 1 (grafik kekonvergenan)
set(handles.axes1,'XLim',[0 1000],'YLim',[0
500],'FontSize',8,'Fontweight','bold');
xlabel('Iterasi'); ylabel('Total Jarak');
cla(handles.axes2,'reset');
% Reset axes 2 (gambar jalur)
set(handles.axes2,'XTick',[],'YTick',[]);
set(handles.listbox1,'string',' ','value',1,'UserData',[]);
% Reset listbox

function edit1_Callback(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of
edit1 as a double

% --- Executes during object creation, after setting all
properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit1 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
```



```
% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of
edit3 as a double

% --- Executes during object creation, after setting all
properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
```

```
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of
edit5 as a double

% --- Executes during object creation, after setting all
properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit7_Callback(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit7 as text
%         str2double(get(hObject,'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit7 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
```

```
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
% Bersihkan Command Window
set(handles.pushbutton4,'string','Jalur');
set(handles.uipanel1,'visible','on');
% Menampilkan panel data
set(handles.uipanel2,'visible','off');
% Menyembunyikan panel jalur
cla(handles.axes1,'reset');
% Reset axes 1 (grafik kekonvergenan)
set(handles.axes1,'XLim',[0 1000],'YLim',[0
500],'FontSize',8,'Fontweight','bold');
xlabel('Iterasi'); ylabel('Total Jarak');
cla(handles.axes2,'reset');
% Reset axes 2 (gambar jalur)
set(handles.axes2,'XTick',[],'YTick',[]);
set(handles.listbox1,'string',' ','value',1,'UserData',[]);
% Reset listbox
pause(0.0001);
% Menghentikan proses sejenak untuk menampilkan hasil reset

% --- Data ---
Data_Node = get(handles.uitable1,'UserData');
% Ambil data titik dari tabel 1
if ~isempty(Data_Node)
    Data_Kode = Data_Node(:,1);
    Data_Coor = Data_Node(:,2:3);
end
Data_Dist = get(handles.uitable2,'UserData');
% Ambil data jarak dari tabel 2

% --- Parameter ---
Npop = str2num(get(handles.edit1,'string'));
% get(handles.edit,'string') -> untuk mengambil string (text)
Maxiter = str2num(get(handles.edit2,'string'));
% str2num() -> convert string/text ke number
MR = str2num(get(handles.edit3,'string'));
c = str2num(get(handles.edit4,'string'));
SMP = str2num(get(handles.edit5,'string'));
CDC = str2num(get(handles.edit6,'string'));
SRD = str2num(get(handles.edit7,'string'));

% --- CSO ---
```

```
if isempty(Data_Node) || isempty(Data_Dist) || isempty(Npop) ||
isempty(Maxiter) || ...
    isempty(MR) || isempty(c) || isempty(SMP) || isempty(CDC)
|| isempty(SRD)
    error('Data atau Parameter kosong');
else
    tic;
% Memulai perhitungan waktu komputasi
    axes(handles.axes1);
    Node = length(Data_Kode);
% Banyaknya titik

    % Inisialisasi posisi awal
    for i = 1:Npop
        X(i,:) = rand(1,Node);
% rand(m,n) -> random matriks m baris n kolom dengan nilai [0, 1]
        [Xurut, Y(i,:)] = sort(X(i,:));
% [A, idx] = sort(X) -> pengurutan X, A : hasil urut, idx :
indeks; indeks sebagai urutan
        Jarak(i) = CalculateDist(Y(i,:), Data_Dist);
% Panggil function CalculateDist -> Hitung total jarak
    end

    % Inisialisasi kecepatan awal
    V = zeros(Npop,Node);
% zeros(m,n) -> membuat matriks nol, m baris n kolom

    % Posisi terbaik
    best = find(Jarak == min(Jarak));
% Kandidat solusi terbaik memiliki total jarak terkecil
    CgX = X(best(1),:);
    CgY = Y(best(1),:);
    JarakCg = Jarak(best(1));

    % Best-so-Far
    bsf(1) = JarakCg;
    nonim = 0;

% Iterasi
    t = 0;
    while t < Maxiter
        % Update flag & SPC
        flag = rand(Npop,1);
% flag : vektor Npop baris bernilai [0, 1]
        SPC = zeros(Npop,1);
% SPC : vektor Npop baris bernilai 0
        SPC(Jarak == min(Jarak)) = 1;
% Mengubah nilai SPC pada posisi terbaik menjadi 1

        % Pembagian seeking & tracing
        numtrac = ceil(MR*Npop);
        numseek = Npop - numtrac;
        [flags, idx] = sort(flag);
        Cseek = idx(1:numseek);
        Ctrac = idx(numseek+1:Npop);
```



```

        % Seeking mode
        numd = ceil(CDC*Node);
% numd : banyaknya dimensi yang dimutasi
        for i = 1:numseek
            %Bangkitkan copy
            if SPC(Cseek(i)) == 0
% Bukan posisi terbaik
                Copy = repmat(X(Cseek(i),:),SMP,1);
% Copy awal = posisi kucing; repmat(A,m,n) -> mengulang matriks A
sebanyak m baris n kolom
                for j = 1:SMP
                    rnd = rand(1,Node);
                    [rnd2, idk] = sort(rnd);
                    pst = idk(1:numd);
% posisi dimensi yang akan dimutasi
                    Copy(j,pst) = Copy(j,pst) + SRD*((-
1).^ (1:numd)).*rand(1,numd).*Copy(j,pst); % Mutasi
                end
            else
% Posisi terbaik
                Copy = repmat(X(Cseek(i),:),SMP,1);
% Copy awal = posisi kucing; repmat(A,m,n) -> mengulang matriks A
sebanyak m baris n kolom
                for j = 1:(SMP-1)
                    rnd = rand(1,Node);
                    [rnd2, idk] = sort(rnd);
                    pst = idk(1:numd);
% Posisi dimensi yang akan dimutasi
                    Copy(j,pst) = Copy(j,pst) + SRD*((-
1).^ (1:numd)).*rand(1,numd).*Copy(j,pst); % Mutasi
                end
            end
            %Ubah ke urutan & hitung jarak
            for j = 1:SMP
                [Copyurut, CopyY(j,:)] = sort(Copy(j,:));
                JarakCopy(j) = CalculateDist(CopyY(j,:),
Data_Dist);
            end
            %Pilih copy terbaik
            if max(JarakCopy) ~= min(JarakCopy)
% Jika total jarak dari Copy berbeda
                prob = abs(max(JarakCopy)-JarakCopy) /
(max(JarakCopy)-min(JarakCopy)); % Hitung probabilitas relatif
                for j = 1:SMP
                    probter(j) = sum(prob(1:j)) / sum(prob);
% Hitung probabilitas kumulatif
                end
                rd = rand;
                for j = 1:SMP
                    if rd <= probter(j)
                        ter = j;
% Indeks Copy terpilih
                    break;
                end
            end
        end
    end
end

```

```
else
% Jika total jarak semua Copy sama
    ter = ceil(rand*SMP);
% Copy dipilih acak
end
%Update posisi
X(Cseek(i),:) = Copy(ter,:);
Y(Cseek(i),:) = CopyY(ter,:);
Jarak(Cseek(i)) = JarakCopy(ter);
end

% Tracing mode
for i = 1:numtrac
    V(Ctrac(i),:) = V(Ctrac(i),:) + c*rand(1,Node).*(CgX-
X(Ctrac(i),:));
    X(Ctrac(i),:) = X(Ctrac(i),:) + V(Ctrac(i),:);
    [Xurut, Y(Ctrac(i),:)] = sort(X(Ctrac(i),:));
    Jarak(Ctrac(i)) = CalculateDist(Y(Ctrac(i),:),
Data_Dist);
end

% Update Best Position
if min(Jarak) < JarakCg
    best = find(Jarak == min(Jarak));
    CgX = X(best(1),:);
    CgY = Y(best(1),:);
    JarakCg = Jarak(best(1));
end

% Update Best-so-Far
bsf(t+2) = JarakCg;
if bsf(t+2) ~= bsf(t+1)
    nonim = t+1;
end

% Plot
plot(0:t+1,bsf,'r','LineWidth',2);
% Plot hasil terbaik setiap iterasi

line(nonim,bsf(nonim+1),'Marker','s','MarkerEdgeColor','k','Marker
FaceColor','y','MarkerSize',5); % Menambahkan tanda untuk non
improvement iteration
set(handles.axes1,'YLim',[min(bsf)-0.1*max(bsf)
1.1*max(bsf)], 'FontSize',8, 'FontWeight', 'bold'); % Setting batas
sumbu y & font
xlabel('Iterasi'); ylabel('Total Jarak');
pause(0.0001);

% next iterasi
t = t + 1;
end

% Hasil
Urutan = '';
for i = 1:Node
```

```

        Urutan = [Urutan num2str(Data_Kode(CgY(i))) '-'];
    end
    Urutan = [Urutan num2str(Data_Kode(CgY(1)))];
    Hasil = {'Urutan Jalur:'; Urutan; ['Total Jarak = '
num2str(JarakCg)];...
        ['Iterasi Non-Improvement = ' num2str(nonim)];...
        ['Waktu Komputasi = ' num2str(toc) ' detik']];
% toc -> hasil akhir perhitungan waktu komputasi
    set(handles.listbox1,'string',Hasil,'value',1,'UserData',CgY);
% Menampilkan hasil di listbox1
end

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
StrPB = get(handles.pushbutton4,'string');
if strcmp(StrPB,'Jalur')
    Route = get(handles.listbox1,'UserData');
    if ~isempty(Route)
        Data_Node = get(handles.uitable1,'UserData');
        Data_Dist = get(handles.uitable2,'UserData');
        axes(handles.axes2);
        Plot_Route(Route, Data_Node, Data_Dist);
    else
        cla(handles.axes2,'reset');
        set(handles.axes2,'XTick',[],'YTick',[]);
    end
    set(handles.uipanel1,'visible','off');
% Menyembunyikan panel data
    set(handles.uipanel2,'visible','on');
% Menampilkan panel jalur
    set(handles.pushbutton4,'string','Data');
elseif strcmp(StrPB,'Data')
    set(handles.uipanel1,'visible','on');
% Menampilkan panel data
    set(handles.uipanel2,'visible','off');
% Menyembunyikan panel jalur
    set(handles.pushbutton4,'string','Jalur');
end

% -----
-----
function mnfile_Callback(hObject, eventdata, handles)
% hObject    handle to mnfile (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```
% -----  
----  
function mnedit_Callback(hObject, eventdata, handles)  
% hObject    handle to mnedit (see GCBO)  
% eventdata  reserved - to be defined in a future version of  
MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
  
% -----  
----  
function mnreset1_Callback(hObject, eventdata, handles)  
% hObject    handle to mnreset1 (see GCBO)  
% eventdata  reserved - to be defined in a future version of  
MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
clc;  
% Bersihkan Command Window\  
set(handles.edit1,'string','');  
% Reset paramater  
set(handles.edit2,'string','');  
set(handles.edit3,'string','');  
set(handles.edit4,'string','');  
set(handles.edit5,'string','');  
set(handles.edit6,'string','');  
set(handles.edit7,'string','');  
  
% -----  
----  
function mnreset2_Callback(hObject, eventdata, handles)  
% hObject    handle to mnreset2 (see GCBO)  
% eventdata  reserved - to be defined in a future version of  
MATLAB  
% handles    structure with handles and user data (see GUIDATA)  
clc;  
% Bersihkan Command Window  
movegui(gcf,'center');  
% Pindah GUI ke tengah layar  
set(handles.pushbutton4,'string','Jalur');  
set(handles.uipanel1,'visible','on');  
% Menampilkan panel data  
set(handles.uipanel2,'visible','off');  
% Menyembunyikan panel jalur  
cla(handles.axes1,'reset');  
% Reset axes 1 (grafik kekonvergenan)  
set(handles.axes1,'XLim',[0 1000],'YLim',[0  
500],'FontSize',8,'Fontweight','bold');  
xlabel('Iterasi'); ylabel('Total Jarak');  
cla(handles.axes2,'reset');  
% Reset axes 2 (gambar jalur)  
set(handles.axes2,'XTick',[],'YTick',[]);  
set(handles.listbox1,'string',' ','value',1,'UserData',[]);  
% Reset listbox  
  
% -----  
----
```

```
function mnnew_Callback(hObject, eventdata, handles)
% hObject    handle to mnnew (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
% Bersihkan Command Window
movegui(gcf, 'center');
% Pindah GUI ke tengah layar
set(handles.pushbutton4, 'string', 'Jalur');
set(gcf, 'Name', 'Program CSO');
% Setting header name
set(handles.uipanel1, 'visible', 'on');
% Menampilkan panel data
set(handles.uipanel2, 'visible', 'off');
% Menyembunyikan panel jalur
set(handles.uitable1, 'data', [], 'UserData', [], 'rowname', 0:10);
% Reset tabel koordinat
set(handles.uitable2, 'data', [], 'UserData', [], 'rowname', 0:10, 'column
name', 0:10); % Reset tabel jarak
set(handles.edit1, 'string', '');
% Reset paramater
set(handles.edit2, 'string', '');
set(handles.edit3, 'string', '');
set(handles.edit4, 'string', '');
set(handles.edit5, 'string', '');
set(handles.edit6, 'string', '');
set(handles.edit7, 'string', '');
cla(handles.axes1, 'reset');
% Reset axes 1 (grafik kekonvergenan)
set(handles.axes1, 'XLim', [0 1000], 'YLim', [0
500], 'FontSize', 8, 'Fontweight', 'bold');
xlabel('Iterasi'); ylabel('Total Jarak');
cla(handles.axes2, 'reset');
% Reset axes 2 (gambar jalur)
set(handles.axes2, 'XTick', [], 'YTick', []);
set(handles.listbox1, 'string', ' ', 'value', 1, 'UserData', []);
% Reset listbox

% -----
----
function mnopen_Callback(hObject, eventdata, handles)
% hObject    handle to mnopen (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
clc;
% Bersihkan Command Window
[File, Path] = uigetfile({'*.xlsx;*.xls', 'Excel Spreadsheet
(*.xlsx, *.xls)'}, 'Open File'); % Buat window open file
if File ~= 0
    [Cust, txt] = xlsread(fullfile(Path, File), 1);
% Baca excel file sheet 1 (koordinat)

set(handles.uitable1, 'data', Cust(:, 2:3), 'UserData', Cust, 'rowname',
Cust(:, 1)); % Tampilkan data koordinat
```



```
[Dist, txt] = xlsread(fullfile(Path,File),2);
% Baca excel file sheet 2 (jarak)

set(handles.uitable2, 'data', Dist(2:end,2:end), 'UserData', Dist(2:en
d,2:end), ...

'rowname', Dist(2:end,1), 'columnname', Dist(1,2:end), 'columnwidth', '
auto'); % Tampilkan data jarak
    set(gcf, 'Name', ['Program CSO - ' fullfile(Path,File)]);
% Setting header name
end

% -----
----
function mnexit_Callback(hObject, eventdata, handles)
% hObject    handle to mnexit (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
close;
```