

PENGOPTIMALAN PENEMPATAN FASILITAS KESEHATAN
MENGUNAKAN ALGORITMA MEMETIKA
(Studi Kasus : Fasilitas Kesehatan Bagi Penderita HIV/AIDS
di Sumatera Utara)

Skripsi
Disusun untuk melengkapi syarat-syarat
guna memperoleh gelar Sarjana Sains



FIRDHA DEFITA SARI
3125111193

PROGRAM STUDI MATEMATIKA
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS NEGERI JAKARTA
2017

LEMBAR PERSETUJUAN HASIL SIDANG SKRIPSI

Pengoptimalan Penempatan Fasilitas Kesehatan Menggunakan Algoritma Memetika (Studi Kasus Fasilitas Kesehatan bagi Penderita HIV/AIDS di Sumatera Utara)

Nama : Firdha Defita Sari

No. Registrasi : 3125111193

	Nama	Tanda Tangan	Tanggal
Penanggung Jawab			
Dekan	: Prof. Dr. Suyono, M.Si. NIP. 19671218 199303 1 005
Wakil Penanggung Jawab			
Wakil Dekan I	: Dr. Muktiningsih, M.Si. NIP. 19640511 198903 2 001
Ketua	: Dr. Makmuri, M.Si. NIP. 19640715 198903 1 006
Sekretaris	: Vera Maya Santi, M.Si. NIP. 19790531 200501 2 006
Penguji	: Ratna Widyati, S.Si., M.Kom. NIP. 19750925 200212 2 002
Pembimbing I	: Med Irzal, M.Kom. NIP. 19770615 200312 1 001
Pembimbing II	: Drs. Mulyono, M.Kom. NIP. 19660517 199403 1 003

Dinyatakan lulus ujian skripsi tanggal: 14 Agustus 2017

ABSTRACT

FIRDHA DEFITA SARI, 3125111193. The Optimization of Public Health care Facility Location Problem with Memetic Algorithm Method (A Case Study : Health care Facility Location for HIV/AIDS Sufferers in Sumatera Utara). Thesis. Faculty of Mathematics and Science Jakarta State University. 2017.

Memetic Algorithms (MAs) is a class of global search heuristic based evolutionary algorithms to solve optimization problems. This algorithms combines genetic algorithm and local search method. Facility location problem is solved by this algorithms. This thesis will discuss about memetic algorithms solving healthcare facility location problem. To solve the problem, start from forming the problem into facility location based model, Maximum Covering Location Problem (MCLP), then use memetic algorithms to solve the problem. The results of these problems are locations of the healthcare facilities by considering the distance factor and the number of facilities which will built.

Keyword : Memetic Algorithm, Facility Location Problem, Maximum Covering Location Problem, Local Search, Greedy Adding Algorithm.

ABSTRAK

FIRDHA DEFITA SARI, 3125111193. Pengoptimalan Penempatan Fasilitas Kesehatan Menggunakan Algoritma Memetika (Studi Kasus : Fasilitas Kesehatan bagi Penderita HIV/AIDS di Sumatera Utara). Skripsi. Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta. 2017.

Algoritma Memetika (MAs) adalah bagian dari pencarian heuristik berbasis algoritma evolusi untuk memecahkan suatu masalah optimasi. Algoritma ini mengkombinasikan algoritma genetik dan pencarian lokal untuk mencari solusi permasalahan. Permasalahan penempatan fasilitas kesehatan salah satu yang dapat diselesaikan dengan algoritma memetika. Pada skripsi ini, akan membahas algoritma memetika dalam penyelesaian permasalahan penempatan fasilitas kesehatan. Langkah awal pencarian solusi adalah dengan membentuk permasalahan kedalam model dasar penempatan fasilitas, *maximum covering location problem* (MCLP), yang selanjutnya akan dicari solusi menggunakan algoritma memetika. Hasil dari pencarian solusi ini berupa lokasi yang terpilih yang akan dibangun fasilitas kesehatan dengan mempertimbangkan faktor jarak jangkauan fasilitas kesehatan dan jumlah fasilitas yang akan dibangun.

Kata kunci : Algoritma Memetika, Penempatan Fasilitas, Maximum Covering Location Problem, Pencarian Lokal, Algoritma Greedy Adding.

PERSEMBAHANKU...

Skrípsi ini Ku persembahkan untuk Ibuku tercinta.

Kerja kerasku menyusun skrípsi ini, tak sepadan
dengan kerja kerasmu untukku selama hidupmu

Terima Kasih atas segalanya...

Semoga Ibu disayang Allah selalu...

Allahummagh-fir Lii Wa Liwaalidayya Warhamhumaa
Kamaa Rabbayaanii Shagiíraa

Anakmu,

KATA PENGANTAR

Puji syukur kepada Allah SWT atas pengetahuan dan kemampuan yang diberikan-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul "Pen-goptimalan Penempatan Fasilitas Kesehatan Menggunakan Algoritma Memeti-ka (Studi Kasus : Fasilitas Kesehatan bagi Penderita HIV/AIDS di Sumatera Utara)" yang merupakan salah satu syarat dalam memperoleh gelar Sarjana Sains Jurusan Matematika Universitas Negeri Jakarta.

Skripsi ini berhasil diselesaikan tidak terlepas dari adanya bantuan dari berbagai pihak. Oleh karena itu, dalam kesempatan ini penulis ingin menyam-paikan terima kasih terutama kepada:

1. Bapak Med Irzal, M.Kom dan Bapak Drs. Mulyono, M.Kom selaku Dosen Pembimbing I dan II, yang telah meluangkan waktunya dalam memberikan bimbingan, saran, nasehat serta arahan sehingga skripsi ini dapat menjadi lebih baik dan terarah.
2. Ibu Lukita Ambarwati, M.Si., selaku Koordinator Program Studi Mate-matika FMIPA UNJ yang telah membantu penulis dalam perkuliahan dan penyusunan skripsi.
3. Bapak Drs. Mulyono, M.Si., selaku Pembimbing Akademik atas segala bimbingan dan kerja sama Bapak selama perkuliahan, dan seluruh Ba-pak/Ibu dosen atas ilmu dan nasehat yang telah diberikan, serta karyawan/karyawati FMIPA UNJ yang telah memberikan informasi yang penulis bu-tuhkan dalam menyelesaikan skripsi.

4. Kedua orangtua penulis Bapak dan Ibu yang selalu mendukung serta mendoakan penulis selama perkuliahan sampai penyusunan skripsi ini.
5. Kakak penulis, Destiara Syahputri, yang telah memberi semangat, mendoakan penulis. Serta kepada keluarga besar, terutama sepupu penulis Annisa Rahmiany, tante penulis Dian Budiwati, Yama, Titi Handayani, dan Om penulis Muchlis Tabrani dan Johny Nanlohy yang selalu mendukung, mendoakan dan memberikan nasehat kepada penulis selama penyusunan skripsi ini.
6. Teman-teman terdekat penulis, CTT, Afifah, Anti, Danti, Gia, Debi dan Lina. Serta Nida dan Hamas selalu mendukung dan menghibur penulis di kala menemukan kendala dalam penyusunan skripsi ini. Dan juga kepada keluarga matematika murni 2011.
7. Terakhir kepada Cho Kyuhyun dan kawan-kawan yang selalu menghibur penulis selama penyusunan skripsi ini.

Penulis menyadari bahwa skripsi ini masih jauh dari sempurna. Masukan dan kritik yang diberikan akan sangat berarti. Semoga skripsi ini dapat bermanfaat bagi pembaca sekalian.

Jakarta, Agustus 2017

Firdha Defita Sari

DAFTAR ISI

ABSTRACT	i
ABSTRAK	ii
KATA PENGANTAR	iv
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	3
1.3 Pembatasan Masalah	4
1.4 Tujuan Penulisan	5
1.5 Manfaat Penulisan	5
1.6 Metode Penelitian	5
II LANDASAN TEORI	7
2.1 Penempatan Fasilitas (<i>Facility Location</i>)	7
2.1.1 Model <i>Set Covering</i>	8
2.1.2 Model <i>Maximum Covering</i>	9
2.1.3 Model P-Median	13
2.2 Kecerdasan Buatan (<i>Artificial intelegent</i>)	15

2.2.1	Algoritma Genetika	16
2.2.2	Algoritma Pencarian Lokal	23
2.2.3	Algoritma <i>Greedy Adding</i>	25
2.2.4	Algoritma Memetika	25
2.3	Contoh Kasus	30
2.3.1	Representasi Individu	32
2.3.2	Inisialisasi Populasi	32
2.3.3	Evaluasi Individu	33
2.3.4	Seleksi Orangtua	35
2.3.5	<i>Crossover</i>	37
2.3.6	Mutasi	38
2.3.7	<i>Alternate Procedure</i>	39
2.3.8	Interchange Procedure	41
2.3.9	Penggantian populasi	43
III PEMBAHASAN		44
3.1	Model Optimasi	45
3.2	Pencarian Solusi dengan Algoritma Memetika	50
3.2.1	Pendefinisian Kromosom	50
3.2.2	Inisialisasi Individu	52
3.2.3	Evaluasi Individu	53
3.2.4	Seleksi <i>Roulette</i>	54
3.2.5	Pindah Silang (<i>Crossover</i>)	55
3.2.6	Mutasi	57
3.2.7	<i>Alternate Procedure</i>	58
3.2.8	<i>Interchange Procedure</i>	60

3.2.9 Penggantian Populasi Baru	61
3.3 Hasil Komputasi	62
IV PENUTUP	67
4.1 Kesimpulan	67
4.2 Saran	67
DAFTAR PUSTAKA	69
LAMPIRAN I PROSES PENCARIAN LOKAL	71
LAMPIRAN II PENGHITUNGAN <i>ALTERNATE PROCEDURE</i>	73
LAMPIRAN III Program Menggunakan Matlab 7.8.0 (R2009)	75
LAMPIRAN IV DATA INPUT	99

DAFTAR TABEL

2.1	Contoh Pembagian Luas pada Mesin Roulette	19
2.2	Data Jumlah Penderita HIV/AIDS di Provinsi Sumatera Utara .	30
2.3	Jarak Kandidat Lokasi ke Titik Permintaan	30
2.4	Pemenuhan Titik Permintaan	31
2.5	Inisialiasasi Individu	32
2.6	Total jarak fasilitas-fasilitas terpilih ke titik permintaan	40
2.7	Penyesuaian Total jarak fasilitas terpilih ke titik permintaan . . .	40
2.8	Penyesuaian Total jarak fasilitas terpilih ke titik permintaan . . .	41
3.1	Lokasi dalam jangkauan S	48
3.2	Contoh Hasil Variabel Keputusan	50
3.3	Representasi Kromosom	51
3.4	Kromosom yang Dimutasi	58
3.5	Nilai Objektif dari Proses <i>Alternate</i>	59
3.6	Nilai Objektif dari Proses <i>Interchange</i>	61
3.7	Hasil Percobaan	63
3.8	Persentase Pasien Terlayani	65
3.9	Hasil Komputasi	65
4.1	Total jarak 1 fasilitas individu 2	71
4.2	Total jarak 2 fasilitas individu 2	71
4.3	Total jarak 3 fasilitas individu 2	72
4.4	Data Penderita HIV/AIDS	100
4.5	Data Jarak antar Permintaan dan Calon Kandidat Lokasi 1	101

4.6	Data Jarak antar Permintaan dan Calon Kandidat Lokasi 2 . . .	102
4.7	Data Jarak antar Permintaan dan Calon Kandidat Lokasi 3 . . .	103
4.8	Data Jarak antar Permintaan dan Calon Kandidat Lokasi 4 . . .	104

DAFTAR GAMBAR

2.1	Sistem menggunakan kecerdasan buatan	15
2.2	Flowchart Algoritma Genetika	17
2.3	Mesin Roulette Pada Tahap Seleksi	19
2.4	Contoh <i>One Point Crossover</i>	21
2.5	Contoh <i>Two Point Crossover</i>	21
2.6	Contoh <i>N-Point Crossover</i>	22
2.7	Contoh <i>Uniform Crossover</i>	22
2.8	Contoh Mutasi Biner	23
2.9	Flowchart Algoritma Pencarian Lokal	24
2.10	Flowchart Algoritma Memetika	29
3.1	Titik Calon Lokasi	45
3.2	Pergerakan Nilai <i>Fitness</i>	64
3.3	Grafik Persentase Pasien Terlayani	64
3.4	Daerah Jangkauan Setiap Fasilitas Kesehatan	66

BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Permasalahan penempatan fasilitas (*location facility problem*) adalah proses pengidentifikasian lokasi geografis terbaik dari suatu fasilitas produksi atau jasa. Menurut Daskin, penempatan lokasi untuk riset operasi diselesaikan dengan pemodelan, pengembangan algoritma, dan teori-teori kompleks. Permasalahan penempatan lokasi akan merancang bagaimana menempatkan suatu fasilitas baik fasilitas umum ataupun pribadi agar optimal sesuai dengan faktor yang diperhatikan. Salah satu metode untuk menyelesaikan permasalahan tersebut dengan membuat model matematika permasalahan penempatan kemudian dicari solusi dari model tersebut menggunakan metode tertentu.

Hal terpenting dalam permasalahan penempatan fasilitas kesehatan adalah mendapatkan hasil solusi yang paling optimal sehingga tujuan dari penempatan fasilitas tersebut tercapai. Salah satu hal yang mempengaruhi hasil solusi adalah pemilihan metode untuk menyelesaikan permasalahan tersebut. Banyak teknik solusi yang dapat digunakan untuk menyelesaikan permasalahan penempatan fasilitas yaitu dengan pendekatan heuristik, contohnya algoritma *greedy* dan algoritma genetika, dan dengan program linier metode simplex. Salah satu metode yang dapat menghasilkan solusi yang sangat baik adalah dengan menggunakan algoritma memetika karena dapat menghasilkan solusi yang optimal

dibandingkan metode lainnya.

Algoritma memetika (MAs) (Moscato, 1989) adalah pendekatan algoritma berbasis populasi untuk mengoptimalkan suatu permasalahan dengan cara mensinergikan semua kemungkinan informasi dari permasalahan yang dituju. Algoritma ini merupakan suatu metode dalam ilmu Kecerdasan Buatan (*Artificial Intelligence*) berupa pencarian heuristik yang berdasarkan mekanisme seleksi alam dan proses genetik. Algoritma memetika adalah penggabungan antara algoritma genetika dan pencarian lokal. Pada prinsipnya, algoritma pencarian lokal diharapkan dapat memberikan solusi lain yang lebih baik disekitar optimum lokal yang dihasilkan oleh algoritma genetika. Algoritma memetika menggunakan fungsi random sehingga menyebabkan algoritma ini menjadi algoritma berbasis komputer.

Dalam menyelesaikan permasalahan penempatan fasilitas, agar dapat diselesaikan dengan algoritma memetika sebelumnya permasalahan akan dibentuk menjadi sebuah model matematika atau dinamakan dengan model penempatan fasilitas. Terdapat 3 model dasar penempatan fasilitas yaitu model *set covering*, model *maxium covering problem*, dan model *p-median*. Banyak ilmuwan yang mengembangkan model baru dari model dasar penempatan fasilitas, salah satunya adalah model *maximum covering location problem* (MCLP) yang dikembangkan oleh Love dan Morris. Model ini digunakan untuk mendapatkan jumlah permintaan yang terlayani dengan batasan hanya ada beberapa fasilitas kesehatan yang akan ditempatkan. Model inilah yang nantinya akan digunakan dalam skripsi ini.

Skripsi ini akan menerapkan permasalahan penempatan lokasi fasilitas pada bidang kesehatan dalam kasus penyakit HIV/AIDS. AIDS (*Acquired Immune Deficiency Syndrome*) adalah suatu kondisi dimana sistem kekebalan tubuh melemah akibat terinfeksi virus HIV sehingga daya tahan tubuh melemah hingga

tubuh mudah terserang penyakit yang kemungkinan tidak mempengaruhi orang dengan sistem kekebalan yang sehat. Sekitar 232.323 orang menderita HIV dan 86.780 orang menderita AIDS di Indonesia (BP2IP,2016). Jumlah kematian akibat penyakit ini terus menapai 5.500 jiwa. Menurut data Badan Pusat Statistik 2016, Sumatera Utara sendiri mempunyai jumlah penderita HIV/AIDS terbanyak ke-6 di Indonesia namun memiliki peningkatan yang sangat besar. Jumlah ini terus meningkat dari waktu ke waktu sehingga pemerintah membuat strategi penanggulangan AIDS Nasional. Salah satu programnya adalah memberikan layanan pengobatan dan perawatan dengan cara memberikan fasilitas kesehatan yang memadai bagi para penderitanya Tujuan penempatan fasilitas kesehatan ini adalah mendapatkan lokasi fasilitas kesehatan (faskes) sehingga penderita HIV/AIDS mendapatkan pelayanan yang baik dalam jangkauan jarak tertentu. Fasilitas kesehatan yang dimaksud dapat berupa rumah sakit, puskesmas (tetap/keliling), apotik maupun pusat penanggulangan kesehatan lainnya. Metode penyelesaian permasalahanan ini dengan menggunakan algoritma memetika sebagai pencari solusi dari model dibentuk.

Penelitian tentang penempatan fasilitas kesehatan pertama kali dilakukan oleh A.A. Aly dan J.A. White pada tahun 1978 tentang penempatan pelayanan gawat darurat (*emergency service*). Kemudian, penelitian terus berkembang dalam menentukan lokasi penempatan fasilitas menggunakan model-model baru yang sesuai dengan kondisi yang ada.

1.2 Perumusan Masalah

Perumusan masalah yang akan dikaji adalah sebagai berikut:

1. Bagaimana langkah-langkah algoritma memetika untuk menentukan lokasi

fasilitas kesehatan bagi penderita HIV/AIDS di Sumatera Utara agar dapat memfasilitasi penderita dalam jarak 17 KM ?

2. Bagaimana solusi optimal penempatan fasilitas sehingga penderita HIV/AIDS di Sumatera Utara dapat terfasilitasi dalam jarak 17 KM ?

1.3 Pembatasan Masalah

Pembatasan masalah dalam penulisan ini adalah:

1. Data penderita HIV/AIDS yang digunakan per Kecamatan dari 6 Kota di Sumatera Utara
2. Jangkauan jarak pelayanan tiap fasilitas kesehatan sama.
3. Tidak ada pertimbangan faktor biaya yang dikeluarkan.
4. Titik permintaan dan titik penawaran sama.
5. Matriks jarak merupakan matriks simetri.
6. Jarak yang diambil merupakan jarak terdekat yang didapatkan dari aplikasi *google maps*.
7. Algoritma Pencarian lokal yang digunakan adalah algoritma *greedy adding*.
8. Jarak maksimal pelayanan fasilitas kesehatan sejauh 17 KM.
9. Pada proses penghitungan dengan algoritma memetika, ukuran populasi yang digunakan sebanyak 15, 20, 25 individu dan penghitungan digenerasikan sebanyak 50, 100, dan 150 kali.

1.4 Tujuan Penulisan

Penulisan ini bertujuan untuk :

1. Mengetahui apa algoritma memetika dapat digunakan dalam permasalahan optimasi penempatan fasilitas kesehatan bagi penderita HIV/AIDS di Sumatera Utara.
2. Mengetahui bagaimana algoritma memetika dalam menyelesaikan permasalahan optimasi penempatan fasilitas kesehatan bagi penderita HIV/AIDS di Sumatera Utara.
3. Mendapatkan jumlah dan lokasi penempatan fasilitas kesehatan yang dapat memfasilitasi penderita HIV/AIDS di Sumatera Utara dalam jarak pelayanan 17 KM.

1.5 Manfaat Penulisan

Manfaat yang diharapkan dari skripsi ini adalah memperoleh penempatan fasilitas kesehatan yang optimal dengan tujuan menurunkan jumlah penderita HIV/AIDS yang tidak mendapatkan pelayanan dengan baik (pasien dapat menjangkau fasilitas kesehatan dalam jarak 17 km).

1.6 Metode Penelitian

Skripsi ini merupakan rekayasa produk dalam bidang matematika optimasi berdasarkan jurnal-jurnal yang berkaitan. Jurnal yang menjadi referensi utama pada skripsi ini adalah jurnal J.F Alegre, Ada Alvarez, dan Silvia C dan

J.A. Pacheco (2010) dan buku Location of Health Care facilities karangan M.S Daskin.

BAB II

LANDASAN TEORI

2.1 Penempatan Fasilitas (*Facility Location*)

Penempatan fasilitas (*Facility Location*) merupakan suatu proses pengidentifikasian lokasi geografis terbaik suatu fasilitas tertentu. Permasalahan penempatan fasilitas dapat berupa menentukan pelayanan gawat darurat (*emergency service*), letak halte kendaraan umum, sekolah, ataupun rumah sakit.

Pemodelan penempatan secara matematis diyakini dapat mengidentifikasi pola lokasi yang optimal berdasarkan pada beberapa tujuan yang realistis dapat diidentifikasi dengan parameter yang diukur. Sebagai contoh, pada analisis area fasilitas pribadi, seperti mencari lokasi gudang penyimpanan yang bertujuan meminimalkan biaya produksi dan distribusi. Karena faktor biaya dimasukkan kedalam fungsi objektif yang dapat diestimasi, lalu model dapat digambarkan kedalam lokasi sebenarnya yang ingin diselesaikan. Tetapi, pada analisis fasilitas umum lebih sulit dalam hal menentukan tolak ukur dari tujuan umum agar menghasilkan solusi yang baik secara umum. Ada dua tolak ukur yang banyak diperhatikan dalam model penempatan yaitu, total jarak atau waktu tempuh ke fasilitas dan jarak atau waktu yang harus ditempuh seorang konsumen terjauh dari fasilitas tersebut (jarak maksimal pelayanan).

Menurut Mark S. Daskin dan Layota K. Dean ada tiga model utama penempatan suatu fasilitas, yaitu model *set covering*, model *maximal covering*,

dan model *p-median*. Ketiga model tersebut menjadi model dasar hampir semua model penempatan fasilitas yang digunakan. Model-model tersebut termasuk kedalam model penempatan fasilitas diskrit. Model diskrit menunjukkan terdapat batasan-batasan permintaan pada suatu titik (*node*) yang sekaligus dijadikan sebagai titik alternatif lokasi.

2.1.1 Model *Set Covering*

Model *Set Covering* merupakan model untuk menemukan jumlah minimal dari fasilitas yang akan ditempatkan namun dapat melayani semua titik permintaan, sehingga permintaan paling sedikit terlayani oleh satu fasilitas. Model ini juga dapat digunakan untuk mendapatkan biaya yang minimal namun semua permintaan dapat terlayani. Formulasi untuk meminimalkan total biaya semua fasilitas yang terpilih adalah :

$$\text{minimal} \quad \sum_{j \in J} f_j X_j \quad (2.1)$$

Dengan fungsi kendala,

$$\sum_{j \in J} a_{ij} X_j \geq 1 \quad \forall i \in I \quad (2.2)$$

$$X_j \in \{0, 1\} \quad \forall j \in J \quad (2.3)$$

Dan variabel keputusan :

$$a_{ij} = \begin{cases} 1 & \text{jika permintaan di titik } i \text{ dapat terlayani oleh fasilitas} \\ & \text{di lokasi kandidat } j \\ 0 & \text{jika tidak} \end{cases}$$

$$X_j = \begin{cases} 1 & \text{jika menempatkan fasilitas di lokasi kandidat } j \\ 0 & \text{jika tidak} \end{cases}$$

Dimana, I = titik permintaan, J = kandidat lokasi fasilitas, f_j = biaya penempatan fasilitas di lokasi j , a_{ij} = permintaan i dapat terlayani oleh fasilitas di lokasi j dan X_j = penempatan fasilitas di lokasi j .

Fungsi kendala (2.2) menetapkan tiap titik permintaan harus terlayani dengan setidaknya satu fasilitas yang terpilih. Persamaan (2.3) merepresentasikan total banyaknya fasilitas yang terpilih yang bisa melayani permintaan di titik i .

2.1.2 Model *Maximum Covering*

Model penempatan *maximum covering* merupakan model untuk memaksimalkan jumlah permintaan yang terlayani dengan jarak pelayanan fasilitas yang sudah ditentukan atau memiliki limit waktu pelayanan tertentu. Model ini dikembangkan oleh Churuch dan Reville menunjukkan jumlah maksimal permintaan yang terlayani dengan batasan hanya ada P fasilitas yang akan ditempatkan. Rumus matematika dari permasalahan ini dapat dibentuk menjadi :

$$\text{maksimal} \quad \sum_{i \in I} h_i r_i \quad (2.4)$$

Fungsi kendalanya :

$$\sum_{j \in N_i} X_j \geq r_i \quad \forall i \in I \quad (2.5)$$

$$\sum_{j \in J} X_j = P \quad (2.6)$$

$$X_j \in \{0, 1\} \quad \forall j \in J \quad (2.7)$$

$$r_i \in \{0, 1\} \quad \forall i \in I \quad (2.8)$$

Dan variabel keputusan :

$$r_i = \begin{cases} 1 & \text{jika permintaan di titik } i \text{ dapat terlayani dalam jangkauan } S \\ 0 & \text{jika tidak} \end{cases}$$

dimana,

h_i = permintaan di titik i ;

P = jumlah fasilitas yang akan dibangun;

S = Jarak maksimal pelayanan;

h_i = jumlah pasien yang terlayani di titik permintaan i ;

$N_i = \{j \in J | d_{ij} \leq S\}$;

d_{ij} = jarak terpendek dari titik i ke titik j ;

N_i merupakan himpunan dari fasilitas yang dapat melayani pasien di titik i . Titik permintaan dapat terlayani saat jarak fasilitas terdekat ke titik tersebut kurang dari atau sama dengan S . Jika kebalikannya, maka titik permintaan di-

anggap tidak dapat terlayani.

Fungsi objektif diatas untuk memaksimalkan jumlah pasien yang dapat terlayani dalam jarak pelayanan tertentu. Kendala (2.5) menjelaskan y_i sama dengan 1 jika saat satu atau lebih fasilitas akan dibangun di lokasi dalam jarak N_i (yaitu, satu atau lebih fasilitas yang berada dalam jarak S dari titik permintaan i). Jumlah fasilitas yang dibangun dibatasi sebesar P pada kendala (2.6). Solusi yang didapat dari permasalahan ini tidak hanya pada banyaknya populasi yang dapat terlayani tetapi banyaknya P fasilitas yang dibangun untuk melayani pasien secara maksimum.

Persamaan yang ekuivalen dengan MCLP dapat ditemukan dengan substitusikan $\mathbf{1} - \bar{r}_i = r_i$ dimana

$$\bar{r}_i = \begin{cases} 1 & \text{jika permintaan di titik } i \text{ tidak dapat terlayani oleh fasilitas} \\ & \text{dalam jarak } S \\ 0 & \text{jika tidak} \end{cases}$$

Persamaan (2.4) dapat ditulis sebagai berikut :

$$\sum_{j \in N_i} X_j \geq 1 - \bar{r}_i \quad \forall i \in I$$

Persamaan ini ekuivalen dengan

$$\sum_{j \in N_i} X_j + \bar{r}_i \geq 1 \quad \forall i \in I$$

Pertidaksamaan ini mengakibatkan

$$\sum_{j \in N_i} \geq 1 \quad \text{dan} \quad \bar{r}_i = 1$$

yaitu, ada satu atau lebih fasilitas yang dibangun dalam jarak S dari titik permintaan i , atau titik permintaan i tidak terlayani dan $\bar{r}_i = 1$. Setelah variable disubstitusikan, fungsi objektif dapat ditulis sebagai :

$$\text{maksimal} \quad \sum_{i \in I} h_i + \sum_{i \in I} -h_i \bar{r}_i$$

Perlu dicatat bahwa penjumlahan pertama bernilai konstan. Karena memaksimalkan dari persamaan negatif ekuivalen dengan meminimalkan dari persamaan positifnya, maka fungsi objektifnya dapat disimpulkan menjadi

$$\text{minimal} \quad \sum_{i \in I} h_i \bar{r}_i$$

Fungsi objektif ini bertujuan untuk meminimalkan jumlah pasien yang tidak terlayani dalam jarak maksimal pelayanan. Sehingga permasalahan lengkap ditulis dengan substitusikan variabel sebagai berikut :

$$\text{minimal} \quad \sum_{i \in I} h_i \bar{r}_i \quad (2.9)$$

Fungsi kendalanya :

$$\sum_{j \in N_i} X_j + \bar{r}_i \geq 1 \quad \forall i \in I \quad (2.10)$$

$$\sum_{j \in J} X_j = P \quad (2.11)$$

$$X_j \in \{0, 1\} \quad \forall j \in J \quad (2.12)$$

$$\bar{r}_i \in \{0, 1\} \quad \forall i \in I \quad (2.13)$$

Fungsi objektif ini untuk meminimalkan pasien yang tidak terlayani jika ada P fasilitas yang akan dibangun.

Persamaan diatas yang berikutnya akan digunakan untuk mencari solusi dari permasalahan penempatan fasilitas kesehatan pada skripsi ini. Persamaan tersebut akan diselesaikan dengan teknik heuristik algoritma memetika.

2.1.3 Model P-Median

Model p-median bertujuan untuk meminimalkan rata-rata jarak permintaan menuju lokasi fasilitas. Pada kasus penempatan fasilitas kesehatan, jarak rata-rata tersebut adalah jarak rata-rata pasien menuju fasilitas kesehatan. Fungsi untuk model p-median dibutuhkan keterangan :

d_{ij} = jarak antara titik permintaan i ke titik kandidat lokasi j .

Dengan variabel keputusan

$$Y_{ij} = \begin{cases} 1 & \text{jika permintaan di titik } i \text{ dapat terlayani oleh fasilitas} \\ & \text{di lokasi kandidat } j \\ 0 & \text{jika tidak} \end{cases}$$

dan fungsi tujuan dari model p-median adalah :

$$\text{minimal} \quad \sum_{j \in J} \sum_{i \in I} h_i d_{ij} Y_{ij} \quad (2.14)$$

dengan fungsi kendala :

$$\sum_{j \in J} Y_{ij} = 1 \quad \forall i \in I \quad (2.15)$$

$$Y_{ij} - X_j \leq 0 \quad \forall i \in I; \forall j \in J \quad (2.16)$$

$$\sum_{j \in J} X_j = P \quad (2.17)$$

$$X_j \in \{0, 1\} \quad \forall j \in J \quad (2.18)$$

$$Y_{ij} \in \{0, 1\} \quad \forall i \in I; \forall j \in J \quad (2.19)$$

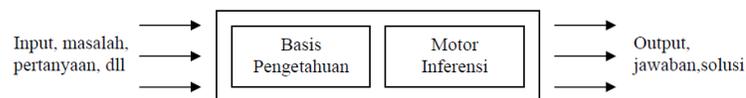
Fungsi objektif (2.14) untuk meminimalkan total jarak permintaan. Ini sama dengan meminimalkan rata-rata jarak permintaan karena banyaknya permintaan konstan. Kendala (2.15) menyatakan tiap titik permintaan harus dilayani dengan satu fasilitas yang ditempatkan. Kendala (2.16) menetapkan bahwa tiap titik permintaan hanya bisa dilayani oleh satu fasilitas yang ada.

Banyak variasi dari algoritma heuristik yang disarankan untuk permasalahan p-median. Dua algoritma terbaik yaitu *neighborhood search algorithm* dan *the exchange algorithm*. Algoritma yang terbaru yang disarankan untuk permasalahan ini adalah algoritma genetik, pencarian tabu, dan *variable neighborhood search algorithm*.

2.2 Kecerdasan Buatan (*Artificial intelegent*)

Kecerdasan Buatan (*Artificial Intelegent*) adalah sebuah studi tentang bagaimana komputer mengerjakan sesuatu yang dapat dikerjakan manusia (Rich, 1991). Tidak ada kesepakatan mengenai definisi dari Kecerdasan Buatan, tetapi para ahli sepakat bahwa Kecerdasan Buatan berhubungan dengan 2 (dua) ide dasar, yaitu menyangkut studi proses berfikir manusia, dan berhubungan dengan mempresentasikan proses tersebut melalui mesin.

Penerapan Kecerdasan Buatan dapat diterapkan di berbagai bidang sebagai contoh dalam sistem pakar (*Expert system*), pemrosesan bahasa alami, sistem pemahaman suara, dan sistem sensor dan robotika. Pada skripsi ini, penerapan kecerdasan dalam menyelesaikan suatu permasalahan penempatan fasilitas. Metodologi yang digunakan untuk menyelesaikannya adalah dengan algoritma genetika dan algoritma pencarian lokal. Sistem yang digunakan untuk menyelesaikan permasalahan optimasi ini akan memberikan output berupa solusi dari masalah berdasarkan kumpulan pengetahuan yang ada.



Gambar 2.1: Sistem menggunakan kecerdasan buatan

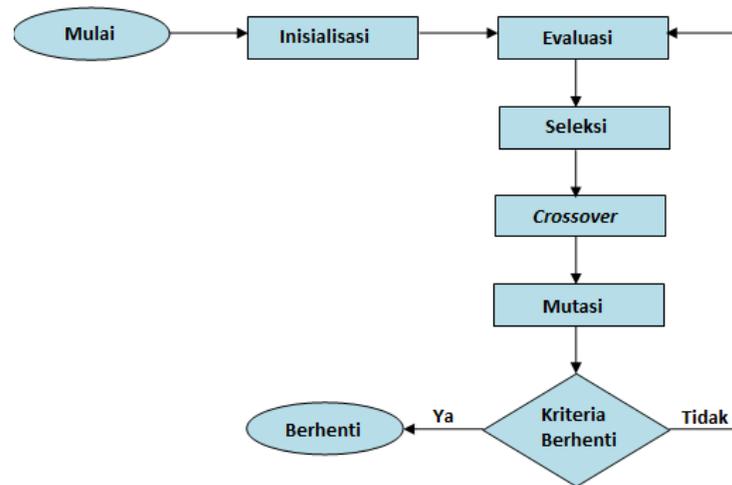
Pada gambar, input yang diberikan pada sistem yang menggunakan kecerdasan buatan adalah masalah. Sistem juga harus dilengkapi dengan pengetahuan-pengetahuan lain yang mendukung proses pada basis pengetahuan. Pada kasus ini, yang menjadi input adalah jumlah penderita HIV/AIDS dan jarak antar calon lokasi fasilitas nanti. Kemudian sistem harus memiliki motor inferensi sehingga mampu menarik kesimpulan berdasarkan pengetahuan. Pada kasus ini, yang

menjadi inferensi adalah algoritma memetika yang akan digunakan sebagai pencari solusi dari permasalahan yang ada. Selanjutnya, akan menghasilkan output berupa solusi dari masalah sebagai hasil dari inferensi.

2.2.1 Algoritma Genetika

Algoritma genetika diperkenalkan pertama kali oleh John Holland merupakan algoritma dengan metode adaptasi yang biasa digunakan untuk memecahkan permasalahan pencarian nilai dalam masalah optimasi. Algoritma genetika ini berbasiskan pada proses genetik makhluk hidup dan teori evolusi Darwin. Algoritma ini bekerja dengan sebuah populasi yang terdiri dari individu-individu yang mewakili solusi dari sebuah permasalahan. Individu dikaitkan dengan nilai *fitness* yang digunakan sebagai tolak ukur solusi terbaik dari permasalahan. Individu-individu tersebut akan melakukan reproduksi dengan proses *crossover* dengan individu lain untuk menghasilkan keturunan atau individu baru. Individu-individu baru tersebut akan membawa gen induknya maka akan muncul beberapa keturunan yang memiliki gen bagus. Gen-gen itulah yang nanti akan menjadi solusi bagi permasalahan yang ada.

Sebelum menjalankan algoritma genetika, mula-mula akan dilakukan pengkodean dari permasalahan yang akan dicari solusinya. Pengkodean dilakukan di kromosom, dalam hal ini kode tersebut disebut gen. Pengkodean bisa direpresentasikan dalam bentuk bit, bilangan riil, daftar aturan atau lainnya. Berikut *flowchart* algoritma genetika.



Gambar 2.2: Flowchart Algoritma Genetika

Inisialisasi Populasi

Pada tahap ini akan dilakukan pembentukan populasi. Pembentukan populasi ini bergantung dari permasalahan yang akan dipecahkan. Populasi awal ini berisi individu-individu yang memiliki gen yang berisi nilai yang diberikan secara acak (*random*).

Evaluasi

Pada tahap ini akan dilakukan evaluasi terhadap individu-individu yang telah terbentuk. Evaluasi dilakukan menggunakan fungsi *fitness*. Fungsi *fitness* merupakan fungsi untuk mendapatkan nilai *fitness*. Nilai *fitness* adalah nilai yang menyatakan baik tidaknya suatu individu dalam populasi. Dalam algoritma genetika, nilai *fitness* dijadikan acuan untuk mencapai nilai optimal suatu permasalahan. Jika semakin besar nilai *fitness* suatu individu maka semakin baik individu tersebut. Fungsi *fitness* yang digunakan untuk mendapatkan nilai *fitness* berbeda pada tiap permasalahan. Jika fungsi tujuannya adalah memaksimalkan

fungsi objektif, maka fungsi *fitness* yang digunakan adalah fungsi objektifnya. Namun, jika tujuannya adalah meminimalkan, maka fungsi *fitness* yang digunakan adalah invers dari fungsi objektifnya. Berikut fungsi objektif dengan tujuan meminimalkan :

$$F(x) = \frac{1}{f(x) + \varepsilon} \quad (2.20)$$

dimana :

$F(x)$ = fungsi *fitness*;

$f(x)$ = fungsi objektif;

ε = bilangan yang sangat kecil;

Seleksi

Pada tahap ini akan dilakukan seleksi untuk memilih individu-individu mana yang akan dipilih untuk proses selanjutnya (*crossover* dan mutasi). Seleksi digunakan untuk mendapatkan calon induk yang baik sehingga dapat menghasilkan keturunan yang baik juga. Individu dengan nilai *fitness* tinggi maka kemungkinan terpilihnya semakin besar. Ada beberapa metode yang dapat digunakan dalam proses seleksi, yaitu:

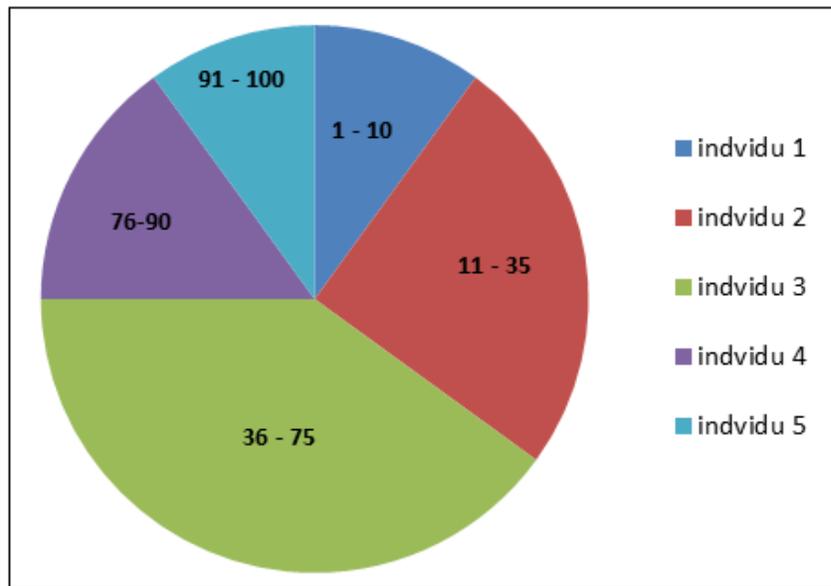
1. Mesin Roulette.

Metode mesin roulette (*stochastic sampling with replacement*) mengadaptasi dari permainan roulette, dimana setiap individu akan mendapatkan tempat pada mesin roulette dengan luas sebanding dengan besarnya nilai *fitness*-nya. Semakin tinggi nilai *fitness* yang dimiliki maka semakin luas bagian yang didapatkan dalam mesin roulette tersebut. Pertama dihitung probabilitas dari individu tersebut, setelah itu dari probabilitas tersebut akan

dihitung nilai yang didapatkan dari angka 1 sampai 100 dari tiap individu. Kemudian akan dibangkitkan angka secara acak dari 1 sampai 100. Dari angka acak yang dihasilkan, ditentukan individu - individu mana yang terpilih dalam proses seleksi. Setelah itu akan dibangkitkan angka secara acak antara 1 sampai 100 sebanyak 5 kali. Misal yang terpilih adalah 32, 85, 64, 18 dan 45 maka individu yang terpilih untuk proses *crossover* dan mutasi adalah individu 2, individu 4, dan individu 3.

Tabel 2.1: Contoh Pembagian Luas pada Mesin Roulette

Individu	Probabilitas <i>fitness</i>	Angka yang didapatkan
1	10%	1 - 10
2	25%	11 - 35
3	40%	36 - 75
4	15%	76 - 90
5	10%	91 - 100



Gambar 2.3: Mesin Roulette Pada Tahap Seleksi

2. Turnamen.

Metode penyeleksian dengan turnamen dengan cara akan menetapkan suatu nilai *tour* untuk individu-individu yang akan dipilih secara acak dari suatu populasi. Individu-individu yang memiliki nilai *fitness* lebih besar akan dipilih untuk dilakukan *crossover* dan mutasi. Parameter yang digunakan adalah ukuran *tour* yang bernilai antara 2 sampai jumlah individu dalam populasi tersebut(n).

3. Seleksi ranking (*rank-based fitness assignment*).

Menentukan induk dengan cara mengurutkan menurut nilai objektifnya. Nilai *fitness* setiap individu hanya tergantung pada posisi individu tersebut dalam urutan, dan tidak dipengaruhi oleh nilai objektifnya. Individu dengan nilai objektif tinggi akan memiliki nilai *fitness* tertinggi dan sebaliknya, sesuai dengan fungsi tujuan.

4. *Truncation selection*.

Merupakan metode seleksi yang digunakan oleh populasi dengan jumlah besar. Metode ini memilih individu dengan cara mengurutkan individu-individu berdasarkan nilai *fitness*-nya, hanya individu dengan *fitness* terbaik yang dapat diseleksi sebagai induk. Parameter yang digunakan adalah suatu nilai ambang *trunc* yang mengindikasikan ukuran populasi yang akan diseleksi sebagai induk berkisar 50% - 100%. Individu-individu yang ada dibawah nilai abang tidak akan terpilih sebagai induk.

Crossover

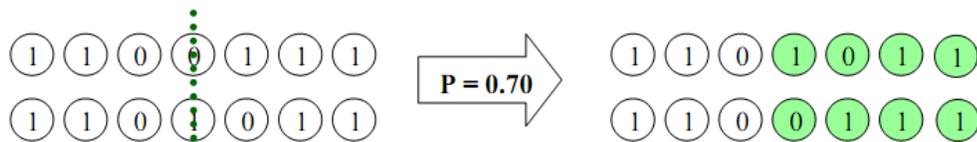
Crossover adalah langkah pada algoritma genetika untuk membentuk individu baru. Pada proses ini individu-individu yang telah terpilih pada proses

seleksi akan dilakukan *crossover* hingga terbentuk individu baru yang diharapkan akan mempunyai nilai *fitness* yang lebih besar. Jika *crossover* tidak dilakukan, maka nilai dari induknya akan diturunkan kepada keturunannya.

Pemilihan individu yang akan *dicrossing* dilakukan secara acak dengan memenuhi probabilitas *crossover*. Menurut Zbiniew Michalewics (1996) probabilitas *crossover* yang baik berada pada range 0,60 sampai 1.

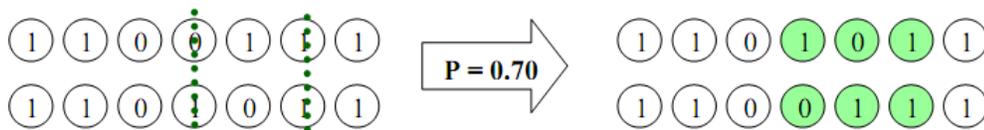
Ada beberapa jenis cara untuk melakukan *crossover*, yaitu :

1. *One point crossover* : *Crossover* ini biasanya dilakukan jika gen dari individu merupakan bilangan biner. Pada proses ini, posisi *crossover* dipilih secara random.



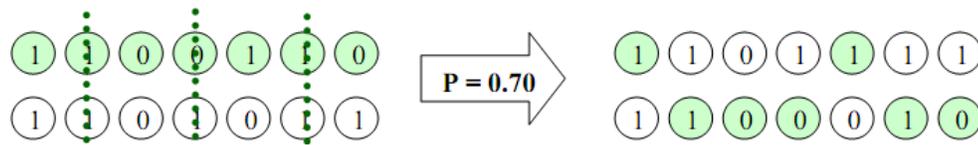
Gambar 2.4: Contoh *One Point Crossover*

2. *Two point crossover* : *Crossover* ini dilakukan dengan dua titik sebagai penanda letak yang akan disilangkan.

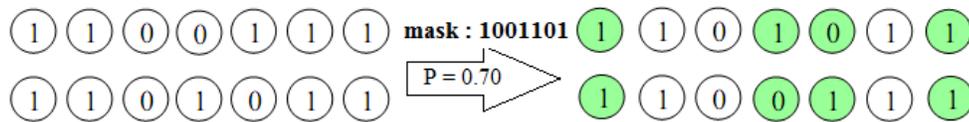


Gambar 2.5: Contoh *Two Point Crossover*

3. *N-point crossover* : Proses *crossover* ini dilakukan secara acak dan tidak boleh ada posisi penyilangan yang sama, serta diurutkan naik.

Gambar 2.6: Contoh *N-Point Crossover*

4. *Uniform crossover* : Pindah silang yang dilakukan dengan membentuk mask bit yang berfungsi sebagai pembanding dalam pembentukan individu baru (*offspring*). Nilai pada mask bit mempengaruhi proses pembentukan *offspring*, jika nilai mask bit 1 maka gen *offspring* didapatkan dengan menukar gen pada induk 1 dan induk 2. Jika nilai mask bit 0, maka nilai gen *offspring* langsung diturunkan dari masing-masing individu.

Gambar 2.7: Contoh *Uniform Crossover*

Mutasi

Mutasi gen adalah proses penggantian gen dengan nilai inversinya, dalam mutasi biner gen 0 menjadi 1 dan 1 menjadi 0. Dalam algoritma genetika, mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi. Tujuan dari mutasi ini adalah untuk mencegah terjadinya seluruh individu dalam satu populasi terjebak dalam optimal lokal.

Proses mutasi dilakukan secara acak pada posisi gen tertentu pada individu-individu yang dipilih untuk dimutasikan. Banyak individu yang mengalami mu-

0 0 1 1 1 0 0 1
0 0 1 1 1 1 0 1

Gambar 2.8: Contoh Mutasi Biner

tasi ditentukan oleh besarnya probabilitas mutasi (P_m). Biasanya peluang mutasi berada antara 0,001 sampai 0,01.

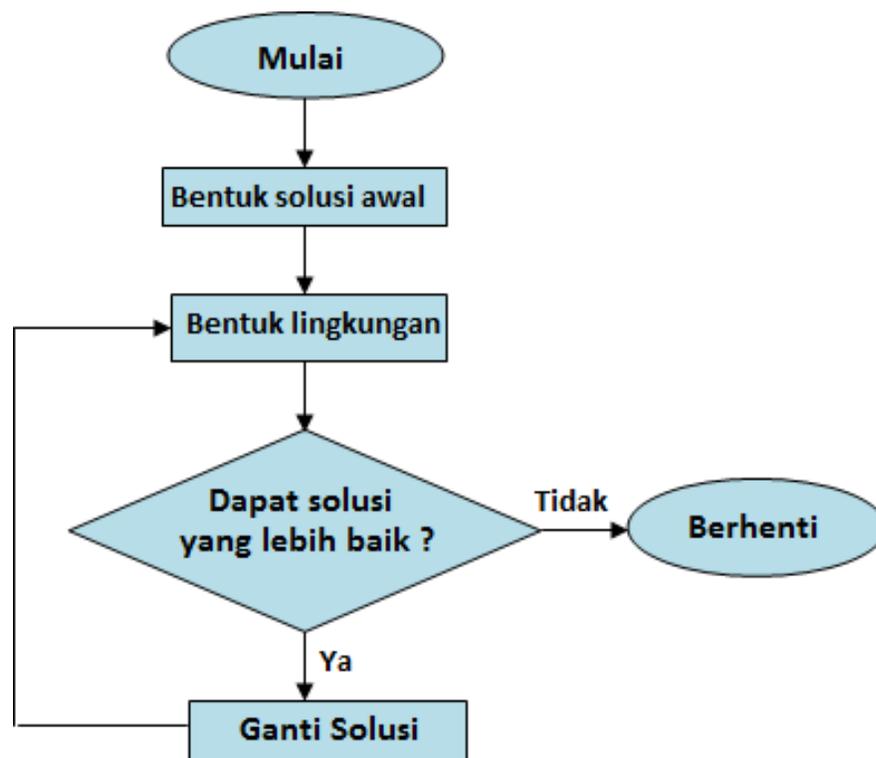
2.2.2 Algoritma Pencarian Lokal

Dalam ilmu komputer, pencarian lokal adalah suatu metode heuristik untuk menyelesaikan permasalahan optimasi. Pencarian lokal dapat digunakan dalam permasalahan yang dapat dirumuskan (formulasikan) untuk memaksimalkan kriteria solusi diantara beberapa calon solusi. Algoritma pencarian lokal bergerak dari solusi ke solusi dalam sebuah ruang kandidat solusi (*the search space*) dengan menerapkan perubahan lokal, sampai solusi dianggap optimal atau dalam batas waktu tertentu.

Langkah awal algoritma pencarian lokal adalah membentuk sebuah solusi awal secara acak, kemudian dibentuk sebuah lingkungan baru dari solusi awal tersebut dengan harapan mendapatkan solusi yang lebih baik. Pembentukan lingkungan baru dari solusi sebelumnya terus dilakukan sampai menemukan solusi yang lebih baik dari lingkungan tersebut. Kemudian proses akan berhenti dapat ditentukan oleh batas waktu tertentu atau dengan mendapatkan solusi terbaik yang ditandai jika solusi pada lingkungan berikutnya tidak mendapatkan hasil yang lebih baik. Algoritma pencarian lokal merupakan algoritma taksiran atau

dugaan atau *incomplete algorithms* karena pencarian solusi mungkin akan berhenti walaupun solusi terbaik yang ditemukan oleh algoritma bukan merupakan solusi terbaik. Hal tersebut dapat terjadi jika proses pemberhentian algoritma harus dilakukan walaupun masih bisa mendapatkan kemungkinan solusi terbaik.

Algoritma pencarian lokal diaplikasikan secara luas untuk berbagai masalah komputasi, seperti dalam kecerdasan buatan, matematika, teknik riset operasi, dan bioinformatika. Contoh algoritma pencarian lokal adalah *hill-climbing*, *gradient methods*, dan *simulated annealing*. Teknik algoritma pencarian lokal yang digunakan dalam skripsi ini adalah dengan metode *greedy adding* yang akan dibahas pada bagian selanjutnya. Berikut *flowchart* algoritma pencarian lokal.



Gambar 2.9: Flowchart Algoritma Pencarian Lokal

2.2.3 Algoritma *Greedy Adding*

Metode heuristik pertama yang digunakan dalam menyelesaikan permasalahan penempatan fasilitas adalah algoritma *greedy adding* atau algoritma *myopic* yang dikembangkan oleh Daskin untuk pencarian solusi penempatan fasilitas. Langkah pertama algoritma *greedy adding* dalam mencari solusi adalah dengan mencari lokasi fasilitas pertama yang akan ditempatkan yang dapat melayani hampir semua pasien. Kemudian, akan dipilih lokasi fasilitas kedua yang akan ditempatkan dimana lokasi tersebut dapat melayani paling banyak pasien yang belum terlayani oleh fasilitas dilokasi pertama. Kemudian, lokasi ketiga akan dipilih, dimana lokasi tersebut dapat melayani paling banyak pasien yang tidak terlayani oleh fasilitas dilokasi pertama dan kedua. Proses ini terus berlangsung sampai tujuan terpenuhi.

Pada model *set covering*, langkah tersebut terus dilakukan sampai semua pasien terlayani. Pada model *maximal covering* dan model *p-median* proses tersebut dilakukan secara berulang sampai P fasilitas ditempatkan. Ketika algoritma *Greedy Adding* mencari solusi untuk P -fasilitas, algoritma tersebut juga memberikan solusi optimal untuk 1 sampai P fasilitas yang ditempatkan. Namun, hasil optimum global yang diberikan algoritma ini belum tentu merupakan solusi terbaik.

2.2.4 Algoritma Memetika

Memetika adalah suatu ilmu pengetahuan yang mempelajari proses replikasi, penyebaran, dan evolusi dari pola informasi (meme). Pola informasi tersebut dapat direplikasi melalui komunikasi antar individu, tanpa ada keterkaitan dengan replikasi dari gen masing-masing individu. Pola tersebut juga dapat

disebarkan ke individu-individu lainnya. Algoritma memetika terinspirasi dari teori evolusi dan terinspirasi oleh gagasan Richard Dawkin. Algoritma memetika adalah bagian dari pencarian heuristik yang menggunakan pendekatan algoritma genetika dan algoritma pencarian lokal. Pada prinsipnya, algoritma memetika diharapkan mampu memberikan solusi yang lebih optimal di sekitar optimum lokal yang diberikan oleh algoritma genetika. Algoritma memetika menggunakan fungsi random sehingga algoritma ini menjadi suatu algoritma berbasis komputer.

Proses replikasi meme pada algoritma memetika disebut juga proses *self-reproduction* atau *memethic life-cycle*. Setelah melakukan proses tersebut dilanjutkan proses penyebaran meme. Penyebaran terjadi melalui sebuah populasi. Sehingga, meme sama halnya seperti suatu gen.

Algoritma memetika merupakan pencarian heuristik yang menggunakan manfaat dari operator-operator evolusi dalam menentukan bagian yang baik dari ruang pencarian, seperti operator pada pencarian lokal yang dengan cepat menemukan solusi yang baik dalam lingkup kecil daerah pencarian. Metode ini terbukti sukses dalam menyelesaikan berbagai permasalahan dan terutama dalam permasalahan optimasi penempatan fasilitas.

Langkah-langkah yang dilakukan dalam algoritma memetika sama dengan algoritma genetika dan ditambah dengan proses dari pencarian lokal. Dalam algoritma memetika, Pencarian lokal diterapkan terhadap seluruh individu di populasi tersebut dengan tujuan untuk meningkatkan nilai *fitness* dari setiap individu. Nilai *fitness* merupakan tolak ukur dari baik tidaknya suatu individu dan solusi.

Dalam beberapa tahun terakhir, banyak peneliti memiliki ketertarikan dengan metode ini sebagai solusi jika memiliki beberapa keterbatasan dalam suatu pendekatan. Hal ini dikarenakan dibandingkan dengan algoritma genetika,

memetika lebih cepat dalam memberikan solusi. Berdasarkan penelitian, dari metode algoritma greedy, algoritma genetika dan algoritma memetika yang paling efektif dalam memberikan hasil optimal adalah algoritma memetika (Alegre et al, 2010). Algoritma memetika memiliki kelebihan dibandingkan algoritma lainnya, yaitu sebagai berikut :

1. Permasalahan kompleks mungkin dapat berkurang sedikit dan sub - permasalahan yang berbeda dapat diselesaikan dengan metode berbeda.
2. Penggunaan algoritma pencarian lokal dapat digunakan untuk melakukan dalam proses sebelum atau sesudah dalam sebuah solusi.
3. Waktu komputasi algoritma memetika lebih cepat namun hasil solusi yang didapatkan lebih baik.

Algoritma memetika sudah terbukti sukses dalam menyelesaikan permasalahan optimasi kombinasi, optimasi dari fungsi tak stasioner, optimasi multi-objektif dan bioinformatika. Algoritma memetika memiliki nama lain seperti algoritma genetika *hybrid*, algoritma evolusi Lamarkian, atau algoritma evolusi Baldwinian.

Skripsi ini akan menggunakan algoritma memetika yang telah dikembangkan oleh Mladenovic. Pengembangan metode terletak pada saat pencarian lokal yang berdasarkan pada dua prosedur yaitu prosedur Alternate dan Interchange. Kedua prosedur ini bertujuan untuk memenuhi fungsi kendala yang ada apa permasalahan model. Pada prosedur alternate akan dilakukan proses penggantian gen pada tiap individu agar keterbatasan-keterbatasan yang ada dalam permasalahan dapat terpenuhi, yaitu sampai ada P fasilitas yang akan dibangun sesuai dengan fungsi kendala. Pada proses ini menggunakan algoritma *greedy*

adding atau algoritma myopic seperti yang telah dijelaskan sebelumnya. Langkah pada proses alternate yang diterapkan pada setiap individu adalah sebagai berikut :

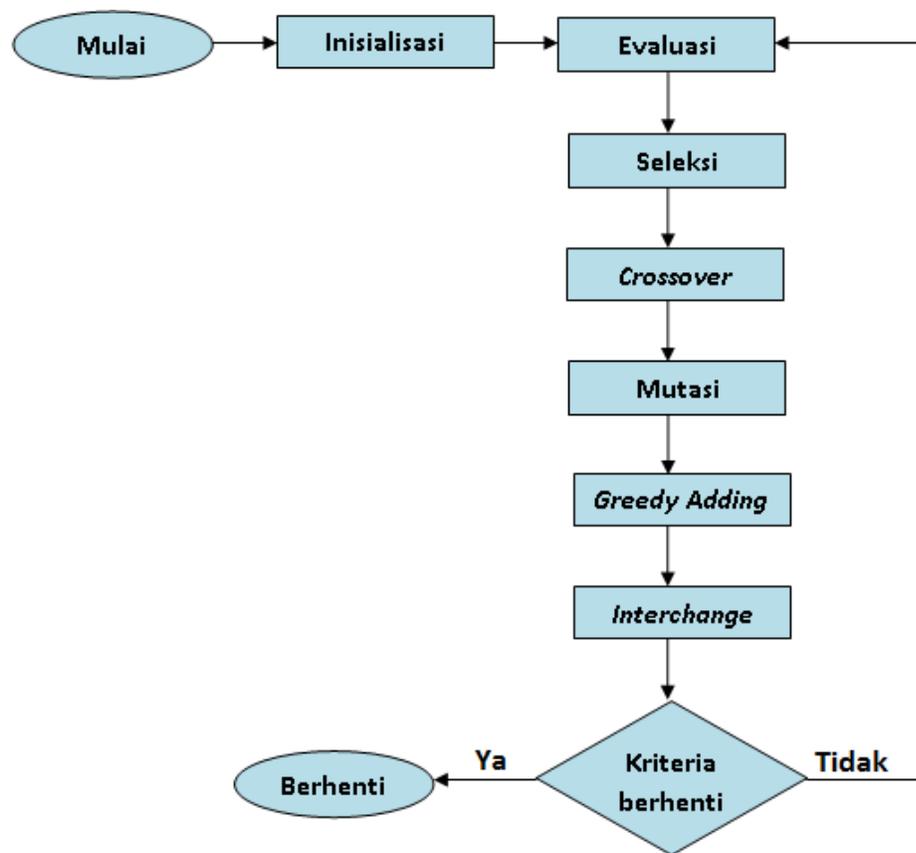
- Inisialisasikan x_j himpunan lokasi terpilih j untuk dibangun fasilitas kesehatan dan l titik fasilitas yang tidak terpilih.
- Hitung $k = \sum x_j$ yaitu jumlah fasilitas yang dibangun.
- Jika $k > P$, maka selesaikan permasalahan k dengan mencari nilai $m = \min\{\sum w_j\}$ yaitu nilai minimal dari total jarak tiap fasilitas w_j .
- Tentukan titik j^* yang memiliki nilai m . Tambahkan j^* pada x_k .
- Jika nilai $k = P$ proses berhenti. Jika belum lakukan kembali langkah 3 untuk $k = k + 1$.
- Jika $k < P$, maka hitung nilai $m = \min\{\sum w_l\}$ yaitu nilai minimal dari total jarak tiap titik lokasi fasilitas tidak terpilih.
- Tentukan titik l^* yang memiliki nilai m . Tambahkan l^* pada x_k .
- Jika $k = P$, proses berhenti. Jika belum lakukan kembali langkah 6 untuk $k = k + 1$.

Prosedur interchange merupakan proses penukaran gen dalam individu dengan tujuan agar solusi tidak terjebak dalam minimum lokal dalam mencari nilai *fitness* terbaik. Berikut langkah dari prosedur interchange :

1. Untuk setiap $k \in J - S$ dan $m \in S$. S himpunan titik lokasi fasilitas terpilih. Hitung nilai dari fungsi objektif v_{km} jika fasilitas pindah dari m ke k . V_{km} merupakan nilai fungsi objektif jika fasilitas dipindah dari titik m ke k .

2. Hitung $v_{k^*m^*} = \min\{v_{km} | k \in J - S\}$ dan $m \in S$. $v_{k^*m^*}$ merupakan nilai fungsi objektif baru setelah penggantian titik k dan m .
3. Jika $v_{k^*m^*} < Z$ maka, $S = S - m^*$, $S \cup k^*$ dan $Z = v_{k^*m^*}$.
4. Lakukan sampai tidak ada penggantian.

Berikut *flowchart* algoritma memetika yang digunakan pada skripsi ini :



Gambar 2.10: Flowchart Algoritma Memetika

2.3 Contoh Kasus

Pemerintah merencanakan pembangunan fasilitas kesehatan berupa puskesmas keliling yang dapat melayani penderita HIV/AIDS di Provinsi Sumatera Utara. Namun, pemerintah menginginkan anggaran yang minimum dengan membangun fasilitas paling sedikit yang dapat melayani semua pasien. Data tabel 2.2 adalah data banyaknya penderita HIV/AIDS pada suatu lokasi di daerah Sumatera Utara dan kandidat lokasi proyek tersebut.

Tabel 2.2: Data Jumlah Penderita HIV/AIDS di Provinsi Sumatera Utara

Indeks	Kota/Kabupaten	Jumlah Penderita HIV/AIDS
1	Medan	95
2	Asahan	25
3	Labuhan Batu	70
4	Deli Serdang	82
5	Tanjung Balai	100
6	Tebing Tinggi	65

Lokasi kandidat yang ideal menurut penderita untuk menempatkan puskesmas keliling berada pada tidak lebih dari 14 km dari lokasi pasien. Berikut data jarak kandidat lokasi ke titik permintaan.

Tabel 2.3: Jarak Kandidat Lokasi ke Titik Permintaan

lokasi d_{ij}	1	2	3	4	5	6
1	0	19.8	14.7	20.7	11	16
2	19.8	0	9	17.6	13.2	22
3	14.7	9	0	19.5	17.2	18.2
4	20.7	17.6	19.5	0	24.2	14.1
5	11	13.2	17.2	24.2	0	22.9
6	16	22	18.2	14.1	22.9	0

Dilihat dari tabel (2.3) bahwa permintaan di titik 1 dapat dipenuhi oleh kandidat lokasi 1 dan 5 permintaan di titik 2 dapat dipenuhi oleh lokasi dititik

2, 3 dan 5, permintaan di titik 3 dapat dipenuhi oleh kandidat lokasi 2 dan 3. Kesimpulan pemenuhan titik permintaan dapat dilihat pada tabel berikut.

Tabel 2.4: Pemenuhan Titik Permintaan

Lokasi Permintaan	Kandidat yang dapat memenuhi	Jarak
1	1	0
	5	11
2	2	0
	3	9
	5	13.2
3	2	9
	3	0
4	4	0
5	1	11
	2	13.2
	5	0
6	6	0

Misalkan j titik kandidat lokasi dan i lokasi permintaan, dan X_j adalah penempatan fasilitas di lokasi j dan h_i adalah jumlah penderita di setiap lokasi. Maka fungsi tujuan untuk meminimalkan jumlah pasien yang tidak terlayani menggunakan persamaan (2.9) yaitu :

$$\text{minimal} \quad \sum_{i \in I} h_i \bar{r}_i \quad i = 1, 2, \dots, 6 \quad (2.21)$$

$$\sum_{i \in I} h_i \bar{r}_i = h_1 \bar{r}_1 + h_2 \bar{r}_2 + \dots + h_6 \bar{r}_6 \quad (2.22)$$

$$\sum_{i \in I} h_i \bar{r}_i = 95 \bar{r}_1 + 25 \bar{r}_2 + \dots + 65 \bar{r}_6 \quad (2.23)$$

Dengan variabel keputusannya,

$$X_j = \begin{cases} 1 & \text{jika menempatkan fasilitas di lokasi kandidat } j \\ 0 & \text{jika tidak} \end{cases}$$

Persamaan (2.21) merupakan fungsi objektif yang akan digunakan sebagai penghitungan untuk langkah selanjutnya. Setelah mendapatkan model tujuan permasalahan, maka selanjutnya akan dicari solusi dari permasalahan tersebut dengan menggunakan algoritma memetika. Langkah-langkah pencarian solusi dengan algoritma memetika ada pada diagram alir pada gambar 2.10.

2.3.1 Representasi Individu

Pada tahap ini akan direpresentasikan gen-gen pada kromosom yang akan digunakan dalam proses algoritma memetika. Pada kasus ini kromosom yang akan digunakan adalah $X_1, X_2, X_3, X_4, X_5, X_6$ yang mewakili kandidat calon lokasi fasilitas kesehatan berupa puskesmas keliling. Sistem pengkodean yang digunakan adalah bilangan biner (0 atau 1) sesuai pembatasan pada variabel keputusan.

2.3.2 Inisialisasi Populasi

Pada tahap ini akan dilakukan pembangkitan populasi awal. Misalkan akan dipilih individu sebanyak 5 individu yang masing-masing terdiri dari 6 kromosom. Kemudian akan dibangkitkan populasi dengan nilai gen-gennya adalah bilangan biner.

Tabel 2.5: Inisialiasasi Individu

Individu 1	0	0	1	0	0	1
Individu 2	0	1	0	1	0	1
Individu 3	1	0	0	1	1	0
Individu 4	0	1	0	1	0	0
Individu 5	1	1	1	0	1	0

2.3.3 Evaluasi Individu

Tahap selanjutnya adalah mengevaluasi tiap-tiap individu dengan cara mencari nilai *fitness* dari masing-masing individu yang didapatkan dari fungsi *fitness*. Fungsi tujuan pada kasus ini adalah mencari nilai minimum, sehingga fungsi *fitness* yang digunakan adalah invers dari fungsi tujuannya, yaitu :

$$F_x = \frac{1}{Z + \varepsilon} = \frac{1}{95r_1 + 25r_2 + 70r_3 + 82r_4 + 100r_5 + 65r_6 + \varepsilon}$$

Dengan ε adalah nilai yang sangat kecil untuk menghindari pembagian dengan 0. Pada skripsi ini berdasarkan jurnal acuan, nilai ε yang digunakan adalah 1. Sebelum melakukan penggantian, dihitung terlebih dahulu nilai fungsi objektif (Z) dari individunya sesuai dengan persamaan (2.17):

- Individu 1 : titik fasilitas terpilih (m) adalah 3 dan 6.

i	1	2	3	4	5	6
$d_{ij}min$	14,7	9	0	14,1	17,2	0
r_i	1	0	0	1	1	0

$$Z_1 = 95(1) + 25(0) + 70(0) + 82(1) + 100(1) + 65(0) = 277$$

- Individu 2 : titik fasilitas terpilih (m) adalah 2, 4 dan 6.

i	1	2	3	4	5	6
$d_{ij}min$	16	0	9	0	13,2	0
r_i	1	0	0	0	0	0

$$Z_2 = 95(1) + 25(0) + 70(0) + 82(0) + 100(0) + 65(0) = 95$$

- Individu 3 : titik fasilitas terpilih (m) adalah 1, 4 dan 5.

$$Z_3 = 95(0) + 25(0) + 70(1) + 82(0) + 100(0) + 65(1) = 135$$

i	1	2	3	4	5	6
$d_{ij}min$	0	13,2	14,7	0	0	14,1
r_i	0	0	1	0	0	1

- Individu 4 : titik fasilitas terpilih (m) adalah 2 dan 4.

i	1	2	3	4	5	6
$d_{ij}min$	19,8	0	9	0	13,2	14,1
r_i	1	0	0	0	0	1

$$Z_4 = 95(1) + 25(0) + 70(0) + 82(0) + 100(0) + 65(1) = 160$$

- Individu 5 : titik fasilitas terpilih (m) adalah 1, 2, 3 dan 5.

i	1	2	3	4	5	6
$d_{ij}min$	0	0	0	14,1	0	16
r_i	0	0	0	1	0	1

$$Z_5 = 95(0) + 25(0) + 70(0) + 82(1) + 100(0) + 65(1) = 147$$

Maka nilai *fitness* dari masing-masing individu diatas adalah sebagai berikut :

1. Nilai *fitness* individu 1 adalah :

$$F_1 = \frac{1}{Z_1 + \varepsilon} = \frac{1}{277 + 1} = 0,36 \times 10^{-2}$$

2. Nilai *fitness* individu 2 adalah :

$$F_2 = \frac{1}{Z_2 + \varepsilon} = \frac{1}{95 + 1} = 1,04 \times 10^{-2}$$

3. Nilai *fitness* individu 3 adalah :

$$F_3 = \frac{1}{Z_3 + \varepsilon} = \frac{1}{135 + 1} = 0,73 \times 10^{-2}$$

4. Nilai *fitness* individu 4 adalah :

$$F_4 = \frac{1}{Z_4 + \varepsilon} = \frac{1}{160 + 1} = 0,62 \times 10^{-2}$$

5. Nilai *fitness* individu 5 adalah :

$$F_5 = \frac{1}{Z_5 + \varepsilon} = \frac{1}{147 + 1} = 0,68 \times 10^{-2}$$

Berdasarkan evaluasi nilai *fitness* diatas, maka individu terbaik adalah individu 2 dan 3. Setelah melakukan evaluasi individu, selanjutnya akan dilakukan seleksi untuk pemilihan orang tua untuk dilakukan *crossover* dan mutasi.

2.3.4 Seleksi Orangtua

Pada tahap ini, untuk memilih individu yang akan dilakukan dilakukan *crossover* dan mutasi adalah menggunakan metode mesin *roulette*. Langkahnya adalah sebagai berikut :

1. Menghitung total nilai *fitness* dari populasi :

$$\text{Total Fitness} = \sum_{k=1}^5 F_n = F_1 + F_2 + F_3 + F_4 + F_5 = 3,43 \times 10^{-2}$$

2. Selanjutnya, menghitung nilai probabilitas seleksi dari masing-masing individu :

$$P_n = \frac{F_n}{Totalfitness} \quad n = 1, 2, 3, 4, 5.$$

$$P_1 = \frac{0,36 \times 10^{-2}}{3,43 \times 10^{-2}} = 0,105$$

$$P_2 = \frac{1,04 \times 10^{-2}}{3,43 \times 10^{-2}} = 0,303$$

$$P_3 = \frac{0,73 \times 10^{-2}}{3,43 \times 10^{-2}} = 0,213$$

$$P_4 = \frac{0,62 \times 10^{-2}}{3,43 \times 10^{-2}} = 0,181$$

$$P_5 = \frac{0,68 \times 10^{-2}}{3,43 \times 10^{-2}} = 0,198$$

3. Setelah itu, menghitung probabilitas kumulatif yang menjadi nilai bagian individu dalam roda *roulette* :

- $q_1 = p_1 = 0,105$
- $q_2 = p_1 + p_2 = 0,408$
- $q_3 = p_1 + p_2 + p_3 = 0,621$
- $q_4 = p_1 + p_2 + p_3 + p_4 = 0,802$
- $q_5 = p_1 + p_2 + p_3 + p_4 + p_5 = 1$

4. Lalu bangkitkan nilai acak R dari range 0 sampai 1, jika nilai $R < q_1$ maka dipilih individu 1 sebagai orangtua, lalu pilih individu ke- k sebagai induk dengan syarat $q_{k-1} < R < q_k$. Pembangkitan dilakukan sebanyak jumlah populasi, yaitu 5 kali. Misalkan yang terpilih : 0,13 ; 0,63 ; 0,81 ; 0,90 ; 0,91. Maka individu yang terpilih adalah :

- Individu 1* = individu 2 = 0 1 0 1 0 1

- Individu 2* = individu 4 = 0 1 0 1 0 0
- Individu 3* = individu 5 = 1 1 1 0 1 0
- Individu 4* = individu 5 = 1 1 1 0 1 0
- Individu 5* = individu 5 = 1 1 1 0 1 0

2.3.5 *Crossover*

Selanjutnya dilakukan *crossover* untuk individu yang terpilih. *Crossover* dilakukan dengan metode *crossover* satu titik (*one point crossover*) dengan probabilitas *crossover* (p_c) sebesar 65%. Prosesnya sebagai berikut :

1. Bangkitkan bilangan acak R sebanyak jumlah populasi. Misalkan :

- $R_1 = 0,09$
- $R_2 = 0,28$
- $R_3 = 0,54$
- $R_4 = 0,95$
- $R_5 = 0,96$

Artinya, individu ke- k akan dipilih sebagai induk jika $R_k < p_c$. Nilai R_1, R_2 dan R_3 kurang dari 0,65 oleh karena itu yang akan dilakukan *crossover* adalah individu 1,2 dan 3

2. Selanjutnya, mendapatkan individu anak dengan cara mengambil gen dari orang tua pertama dan kedua. Titik potong diambil secara acak. Berikut hasil *crossover* tersebut :

- *offspring* 1 = Individu 3 \times Individu 1 = [1 1 1 0 1 0] \times [0 1 0 1 0 1]
= 1 1 0 1 0 1

- *offspring* 2 = Individu 1 \times Individu 2 = [0 1 0 1 0 1] \times [0 1 0 1 0 0]
= 0 1 0 1 0 0
- *offspring* 3 = Individu 2 \times Individu 1 = [0 1 0 1 0 0] \times [0 1 0 1 0 1]
= 0 1 0 1 0 1
- *offspring* 4 = Individu 3 \times Individu 1 = [1 1 1 0 1 0] \times [0 1 0 1 0 1]
= 1 1 1 1 0 1
- *offspring* 5 = Individu 3 \times Individu 1 = [1 1 1 0 1 0] \times [0 1 0 1 0 1]
= 1 1 1 0 1 1

Dengan demikian populasi sudah mengalami *crossover*.

2.3.6 Mutasi

Menentukan jumlah kromosom yang akan dilakukan proses mutasi dengan parameter probabilitas mutasi (p_{mut}). Dibangkitkan nilai acak R antara 0 dan 1, jika nilai R lebih kecil dari p_{mut} maka dilakukan proses mutasi. Kemudian menggantikan gen pada individu tersebut dengan lawan dari nilai tersebut. Dalam kasus ini, jika nilai 0 diganti dengan 1, dan sebaliknya. Prosesnya sebagai berikut :

1. Hitung panjang total gen dalam satu populasi. Dalam kasus ini ada 30 gen.
2. Pilih gen yang akan dimutasikan dengan membangkitkan bilangan acak dari 1 sampai 30. Jika R lebih kecil dari p_{mut} , maka pilih posisi tersebut sebagai gen yang akan dimutasikan. Misal $p_{mut} = 10\%$ maka jumlah gen yang akan dialami mutasi adalah 10% dari 30 individu yaitu 3 individu.
3. Bangkitkan bilangan acak R , misal terpilih posisi 2, 20 dan 26, berarti gen

ke-1 dari individu 2, gen ke-4 dari individu 5 dan gen ke-5 dari individu 1 akan diganti dengan lawannya. Hasil populasi baru dari mutasi adalah :

- *offspring* 1 = 110100
- *offspring* 2 = 110100
- *offspring* 3 = 010101
- *offspring* 4 = 111101
- *offspring* 5 = 111111

2.3.7 *Alternate Procedure*

Tahap selanjutnya adalah memperbaiki tiap individu dengan pencarian lokal dengan algoritma *greedy adding*. Pada tahap ini akan ditentukan 3 fasilitas terdekat untuk tiap individu dari hasil mutasi. Langkahnya sebagai berikut :

1. gen dari individu 4 : 1 1 1 1 0 1 berarti bahwa penempatan fasilitas kesehatan di titik 1,2, 3, 4 dan 6. Dari kelima lokasi terpilih tersebut akan dipilih 3 lokasi terdekat dari seluruh titik permintaan, lokasi pertama yang akan dipilih berdasarkan total jarak terkecil dari semua jarak antar tiap fasilitas ke penderita. Pada tabel 2.6 terdapat kolom total jarak tiap fasilitas terpilih ke semua titik permintaan. Dan, terpilih fasilitas dan terdekat titik 3 sebagai lokasi pertama yang dipilih untuk penempatan lokasi fasilitas dari individu 1.
2. Selanjutnya, akan dipilih lokasi kedua untuk individu 1. Sebelum melakukan seperti cara pertama, perlu diubah lokasi tiap fasilitas dengan nilai d_{ij} dengan nilai minimum antara nilai jarak tiap titik permintaan ke tiap fasilitas

Tabel 2.6: Total jarak fasilitas-fasilitas terpilih ke titik permintaan

	1	2	3	4	6
1	0	19,8	14,7	20,7	16
2	19,8	0	9	17,6	22
3	14,7	9	0	19,5	18,2
4	20,7	17,6	19,5	0	14,1
5	11	13,2	17,2	24,2	22,9
6	16	22	18,2	14,1	0
Σ	82,2	81,6	78,6	96,1	93,2

terpilih dan ke fasilitas 3 seperti yang tersaji pada tabel 2.7. Terdapat jumlah minimum yang sama antara fasilitas 4 dan 6, maka fasilitas dipilih secara bebas. Pada kasus ini dipilih fasilitas dilokasi 4.

Tabel 2.7: Penyesuaian Total jarak fasilitas terpilih ke titik permintaan

	1	2	3	4	6
1	0	14,7	14,7	14,7	14,7
2	9	0	9	9	9
3	0	0	0	0	0
4	19,5	17,6	19,5	0	14,1
5	11	13,2	17,2	17,2	17,2
6	16	18,2	18,2	14,1	0
Σ	55,5	63,7	78,6	55	55

- Selanjutnya dilakukan pemilihan lokasi ke 3 dengan cara yang sama dengan langkah kedua. Dari tabel 2.8 terlihat fasilitas dititik 1 terpilih karena memiliki total jarak terkecil.
- Sudah terpilih 3 fasilitas terdekat yaitu 1,3 dan 4. Maka langkah selesai dan individu 1 berubah menjadi 1 0 1 1 0 0.

Individu 1,2 dan 3 karena sudah memiliki 3 fasilitas yang ditempatkan, maka gen-gen tersebut sudah menjadi hasil dari *greedy adding*. Hasil dari proses

Tabel 2.8: Penyesuaian Total jarak fasilitas terpilih ke titik permintaan

	1	2	3	4	6
1	0	14,7	14,7	14,7	14,7
2	9	0	9	9	9
3	0	0	0	0	0
4	0	0	0	0	0
5	11	13,2	17,2	17,2	17,2
6	14,1	14,1	14,1	14,1	0
Σ	34,1	43,8	55	55	40,9

pencarian lokal untuk seluruh individu dalam populasi adalah sebagai berikut.

Penghitungan lihat pada lampiran 1 :

- Individu 1 : 1 1 0 1 0 0
- Individu 2 : 1 1 0 1 0 0
- Individu 3 : 0 1 0 1 0 1
- Individu 4 : 1 0 1 1 0 0
- Individu 5 : 1 0 1 1 0 0

2.3.8 Interchange Procedure

Setelah melakukan proses mutasi selanjutnya menerapkan pencarian lokal untuk semua anak pada populasi. Proses *interchange* untuk memperbaiki solusi. Pada tahap ini akan dilakukan pergantian tiap gen dalam individu dengan semua kemungkinan lokasi fasilitas yang ada. Jika nilai fungsi objektif lebih baik maka dilakukan proses penggantian tersebut. Langkah-langkah dalam prosedur ini adalah sebagai berikut :

1. Setiap gen di individu 4 kita ubah satu per satu dengan fasilitas yang ada. Fasilitas terpilih oleh individu 4 (m) adalah 1,3 dan 4. Fasilitas yang belum

terpilih (k) adalah 2,5 dan 6. Sebelum melakukan penggantian, dihitung terlebih dahulu nilai fungsi objektifnya.

i	1	2	3	4	5	6
$d_{ij}min$	0	9	0	0	11	14,1
r_i	0	0	0	0	0	1

$$Z_1 = 95(0) + 25(0) + 70(0) + 82(0) + 100(0) + 65(1) = 65$$

2. Selanjutnya diubah setiap gen pada individu dengan fasilitas yang belum terpilih. Misalkan, pertama ganti $m = 1$ pada individu dengan $k = 2$, sehingga fasilitas terpilih menjadi 2,3 dan 4. lalu hitung nilai v_{km} yaitu nilai fungsi objektif baru, jika $v_{km} < Z_1$ fasilitas terpilih pindah dari m ke k .

i	1	2	3	4	5	6
$d_{ij}min$	14,7	0	0	0	13,2	14,1
r_i	1	0	0	0	0	1

$$v_{21} = 95(1) + 25(0) + 70(0) + 82(0) + 100(0) + 65(1) = 160$$

Karena, nilai v_{21} tidak lebih baik dari Z , maka tidak perlu diubah gen pada individu.

3. Lakukan penggantian tersebut untuk semua kemungkinan k , sampai tidak ada perubahan pada nilai Z . Untuk individu 4 nilai Z akhir yang didapatkan adalah 65, dengan penempatan dititik 1, 3 dan 4, atau dikodekan dengan 1 0 1 1 0 0. Semua penghitungan untuk individu 1 dapat dilihat pada lampiran 2.

Hasil dari langkah ini untuk semua individu adalah sebagai berikut :

- *Offspring* 1 : 1 1 0 1 0 0 dengan $f_1 = 0,015$

- *Offspring* 2 : 1 1 0 1 0 0 dengan $f_2 = 0,015$
- *Offspring* 3 : 1 1 0 1 0 0 dengan $f_3 = 0,015$
- *Offspring* 4 : 1 0 1 1 0 0 dengan $f_4 = 0,015$
- *Offspring* 5 : 1 0 1 1 0 0 dengan $f_5 = 0,015$

2.3.9 Penggantian populasi

Setelah mendapatkan anak(*offspring*) serta nilai fungsi objektifnya, maka solusi tersebut digantikan untuk dilakukan iterasi kedua. Proses iterasi kedua dimulai dari seleksi orang tua dan seterusnya. Proses terus dilakukan sampai batasan iterasi tertentu. Jika nilai Z pada iterasi berikutnya tidak menemukan hasil *fitness* yang lebih baik dari iterasi pertama, yaitu sebesar 0,015, maka solusi tetap Z sebesar 65, jika lebih besar maka berubah mengikuti hasil yang baru.

Hasil optimum dari kasus ini adalah memiliki nilai *fitness* sebesar 0,015 dengan kesimpulan bahwa jumlah minimal penderita yang tidak terlayani sebanyak 65 orang dan lokasi terpilih untuk menempatkan puskesmas keliling dititik 1,3 dan 4 atau dititik 3,4 dan 5.

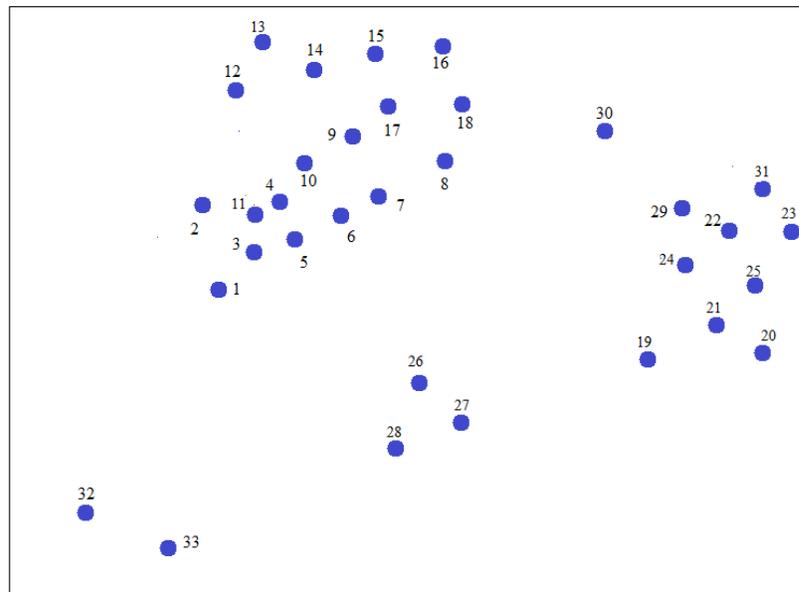
BAB III

PEMBAHASAN

Pada bab ini akan mengaplikasikan algoritma memetika untuk menyelesaikan permasalahan penempatan lokasi yang optimal bagi penderita HIV/AIDS, yang selanjutnya mengambil studi kasus optimasi penempatan fasilitas kesehatan di Provinsi Sumatera Utara. Hasil yang didapatkan berupa lokasi dan jumlah fasilitas kesehatan yang harus dibangun yang dapat melayani semua pasien. Fasilitas yang dimaksud adalah puskesmas keliling beroda 4, hal ini dimaksudkan agar model berlaku jika terdapat perubahan jumlah penderita.

Data yang digunakan adalah penderita HIV/AIDS di 33 Kecamatan di Sumatera Utara dan jarak antar kecamatan dari 6 Kota/Kabupaten di Sumatera Utara tersebut (Deli Serdang, Asahan, Labuhanbatu Selatan, Labuhan Batu Dalam, Batubara Dalam, dan Gunung Sitoli). Calon lokasi puskesmas keliling yang digunakan sama dengan daerah pasien, sehingga jumlah lokasi pasien sama dengan jumlah calon lokasi fasilitas kesehatannya. Menurut BPS Sumatera Utara tahun 2013, Jumlah puskesmas keliling di setiap Kabupaten/Kota berbeda-beda tetapi ada Kota/Kabupaten yang tidak memiliki puskesmas keliling seperti di Kabupaten Batu Bara. Sehingga dalam kasus ini diambil jumlah puskesmas keliling yang akan ditempatkan berjumlah 7, yaitu jumlah puskesmas keliling terbanyak yang ada yaitu di Kabupaten Asahan, untuk melayani semua Kecamatan. Hasil yang diharapkan adalah akan mendapatkan lokasi yang optimal untuk penempatan fasilitas tersebut yang dapat melayani dalam jarak jangkauan 17 KM.

Langkah awal dalam melakukan optimasi adalah mengambil data permintaan dan kandidat lokasi fasilitas kesehatan, kemudian membuat model matematika sesuai dengan tujuan. Setelah membuat model dan mengatur parameter dari model, algoritma matematika akan memproses model dan menghasilkan *output* berupa lokasi yang terpilih akan dibangun fasilitas kesehatan.



Gambar 3.1: Titik Calon Lokasi

3.1 Model Optimasi

Pengoptimalan kasus ini bertujuan untuk mendapatkan lokasi yang dapat melayani penderita HIV/AIDS dengan menggunakan model *maximum covering* pada persamaan (2.9). dengan tujuan untuk memberikan jumlah minimal penderita yang tidak terlayani oleh fasilitas kesehatan dengan batasan jumlah fasilitas kesehatan yang akan dibangun sebanyak P fasilitas di Kecamatan terpilih dan

jarak maksimal pelayanan (d_{ij}) dari lokasi fasilitas kesehatan ke tiap-tiap pasien.

Asumsi permasalahan penempatan lokasi fasilitas kesehatan ini adalah sebagai berikut :

1. Titik permintaan dan titik penawaran sama yaitu sebanyak 33 titik dari 6 Kota/Kabupaten.
2. Jarak antar titik simetris, artinya jarak dari titik A ke B sama dengan jarak dari titik B ke A.
3. Jarak antar titik adalah jarak terdekat yang didapatkan dari *google maps*. Aplikasi *google maps* adalah aplikasi yang dapat menentukan jarak tempuh terdekat antara dua titik pada peta.
4. Batas jarak pelayanan maksimum (S) adalah 17 km.
5. Pada proses penghitungan, ukuran populasi yang digunakan sebanyak 15, 20, 25 individu dan penghitungan digenerasikan sebanyak 50,100, dan 150 kali.

Selanjutnya adalah menentukan jarak dari tiap calon lokasi fasilitas ke semua penderita yang merupakan jarak antar Kecamatan. Data jarak tersebut (d_{ij}) ada pada lampiran III. Jarak antar titik yang sama selalu nol. Pada skripsi ini diasumsikan kandidat lokasi puskesmas keliling yang akan dibangun sama dengan lokasi pasien, yaitu ada 33 kandidat lokasi fasilitas kesehatan.

Selanjutnya akan dibentuk model optimasi dari kasus ini yang merujuk pada persamaan (2.9) dengan n sebanyak 33. Sesuai data yang diperoleh, berikut

formulasi fungsi tujuannya :

$$\begin{aligned} \text{minimal } Z &= \sum_{i \in I} h_i \bar{r}_i \\ &= 20r_1 + 1r_2 + 16r_3 + \dots + 5r_{131} + 3r_{32} + 5r_{33} \end{aligned} \quad (3.1)$$

dimana, I adalah himpunan titik permintaan (1,2,3.....33) dan $J =$ himpunan titik lokasi fasilitas (1,2,3,....33). Dan h_i merupakan jumlah pasien pada titik i

Pada kasus ini, memiliki kendala permasalahan yang merujuk pada persamaan (2.10) dan (2.11). Jarak antar kecamatan pada tabel (3.9) merepresentasikan d_{ij} . Fungsi kendalanya sebagai berikut :

1. Fungsi kendala pertama sebagai berikut :

$$\sum_{j \in N_i} X_j + \bar{r}_i \geq 1 \quad \forall i \in I$$

Persamaan diatas menjelaskan bahwa r_i bernilai 1 hanya jika ada satu atau lebih faskes dibangun dititik yang berada pada himpunan N_i . Variabel N_i adalah himpunan dari j yang berada dalam jangkauan jarak S untuk setiap titik permintaan i . Dimana, S adalah jarak batas pelayanan atau jangkauan tiap fasilitas kesehatan. Artinya, jika jarak antara lokasi penderita i ke fasilitas j lebih dari S maka, penderita i dianggap tidak bisa dilayani oleh fasilitas kesehatan dititik j , disini $S = 17$ Km. Sehingga, dari tabel jarak kandidat dan lokasi pada lampiran, N_i adalah :

Tabel 3.1: Lokasi dalam jangkauan S

N_i	j	N_i	j
N_1	1,2,5	N_{18}	8,9,15,16,17,18
N_2	1,2,3,4,5	N_{19}	19,20,21,23
N_3	2,3,4,5,10,11,12	N_{20}	19,20,21,24,25
N_4	2,3,4,5,6,10,11,12	N_{21}	19,20,21,23,24,25
N_5	1,2,3,4,5,6,10,11	N_{22}	22,23,24,25,31
N_6	4,5,6,7,10,11	N_{23}	19,21,22,23,24,25
N_7	6,7,8	N_{24}	20,21,22,23,24,25
N_8	6,7,9,17,18	N_{25}	20,21,22,23,24,25
N_9	8,9,10,11,14,15,17,18	N_{26}	26
N_{10}	3,4,5,9,10,11	N_{27}	27
N_{11}	3,4,5,6,9,10,11	N_{28}	28
N_{12}	3,4,12,13	N_{29}	29
N_{13}	12,13	N_{30}	30
N_{14}	9,14,15,16,17	N_{31}	22,31
N_{15}	9,14,15,16,17,18	N_{32}	32
N_{16}	14,15,16,17,18	N_{33}	33
N_{17}	8,9,14,15,16,17,18		

Sehingga, fungsi kendala pertama dapat ditulis sebagai berikut:

$$\sum_{j \in N_1} X_j + \bar{r}_1 = X_1 + \bar{r}_1 + X_2 + \bar{r}_1 + X_5 + \bar{r}_1 \geq 1$$

$$\sum_{j \in N_2} X_j + \bar{r}_2 = X_1 + \bar{r}_2 + X_2 + \bar{r}_2 + X_3 + \bar{r}_2 + X_4 + \bar{r}_1 + X_5 + \bar{r}_1 \geq 1$$

$$\begin{aligned} \sum_{j \in N_3} X_j + \bar{r}_3 &= X_2 + \bar{r}_3 + X_3 + \bar{r}_3 + X_4 + \bar{r}_3 + X_5 + \bar{r}_3 + X_{10} + \bar{r}_3 \\ &\quad + X_{11} + \bar{r}_3 + X_{12} + \bar{r}_3 \geq 1 \end{aligned}$$

$$\begin{aligned} \sum_{j \in N_4} X_j + \bar{r}_4 &= X_2 + \bar{r}_4 + X_3 + \bar{r}_4 + X_4 + \bar{r}_4 + X_5 + \bar{r}_4 + X_{10} + \bar{r}_4 \\ &\quad + X_{11} + \bar{r}_4 + X_{12} + \bar{r}_4 \geq 1 \end{aligned}$$

$$\begin{aligned}
 & \cdot \\
 & \cdot \\
 & \cdot \\
 & \sum_{j \in N_{31}} X_j + \bar{r}_{31} = X_{22} + \bar{r}_{31} + X_{31} \geq 1 \\
 & \sum_{j \in N_{32}} X_j + \bar{r}_{32} = X_{32} + \bar{r}_{32} + X_{32} \geq 1 \\
 & \sum_{j \in N_{33}} X_j + \bar{r}_{33} = X_{33} + \bar{r}_{18} \geq 1
 \end{aligned}$$

2. Fungsi kendala kedua adalah :

$$\sum_{j \in J} X_j = P \quad (3.2)$$

P merupakan banyaknya fasilitas kesehatan yang akan dibangun. Calon lokasi faskes (J) sejumlah 33. Banyaknya calon lokasi disimbolkan dengan J , sedangkan j merupakan nomor calon lokasi fasilitas kesehatan, untuk $j = 1, 2, 3, \dots, 33$. Sehingga persamaan fungsi kendala dapat ditulis dengan :

$$\sum_{j \in J} X_j = x_1 + x_2 + x_3 + \dots + x_{33} = P \quad (3.3)$$

Banyaknya P berubah dalam *range* 1 sampai 10.

Variabel keputusan untuk permasalahan ini adalah sebagai berikut :

$$X_j = \begin{cases} 1 & \text{jika ada fasilitas kesehatan yang ditempatkan dititik } j \\ 0 & \text{jika tidak} \end{cases}$$

$$\bar{r}_i = \begin{cases} 1 & \text{jika penderita di } i \text{ tidak terlayani oleh faskes dalam jarak } S \\ 0 & \text{jika tidak} \end{cases}$$

Variabel keputusan ini merujuk pada persamaan (2.12). Selanjutnya, penghitungan permasalahan optimasi diatas menggunakan metode algoritma memetika menggunakan batuan *software* Matlab.

Tabel 3.2: Contoh Hasil Variabel Keputusan

i	1	2	3	4	5	6	7	8	9	28	29	30	31	32	33
\bar{r}_i	1	0	0	1	0	1	0	1	1	0	1	0	0	0	0

3.2 Pencarian Solusi dengan Algoritma Memetika

Pencarian solusi pada permasalahan ini menggunakan algoritma memetika yang sesuai dengan *flowchart* pada gambar 2.10. Hasil yang ditampilkan untuk mewakili hasil setiap langkah berikut ini untuk $P = 5$.

3.2.1 Pendefinisian Kromosom

Pada kasus ini, kromosom merepresentasikan lokasi yang akan dibangun fasilitas (j). Sehingga, jumlah gen yang digunakan dalam tiap individu ada 33 gen sesuai dengan banyaknya j .

Tabel 3.3: Representasi Kromosom

Kromosom	lokasi J	Kecamatan
Kromosom 1	1	Sibolangit
Kromosom 2	2	Kutalimbaru
Kromosom 3	3	Pancur Batu
Kromosom 4	4	Namo Rambe
Kromosom 5	5	Biru-biru
Kromosom 6	6	Tanjung Muda Hilir
Kromosom 7	7	Bangun Purba
Kromosom 8	8	Galang
Kromosom 9	9	Tanjung Morawa
Kromosom 10	10	Patumbak
Kromosom 11	11	Deli Tua
Kromosom 12	12	Sunggal
Kromosom 13	13	Hamparan Perak
Kromosom 14	14	Percut Sei Tuan
Kromosom 15	15	Batang Kuis
Kromosom 16	16	Pantai Labu
Kromosom 17	17	Lubuk Pakam
Kromosom 18	18	Pagar Merbau
Kromosom 19	19	Simpang Empat
Kromosom 20	20	Air Batu
Kromosom 21	21	Sei Dadap
Kromosom 22	22	Meranti
Kromosom 23	23	Air Joman
Kromosom 24	24	Kisaran Barat
Kromosom 25	25	Kisaran Timur
Kromosom 26	26	Rantau Selatan
Kromosom 27	27	Kota Pinang
Kromosom 28	28	Kampung Rakyat
Kromosom 29	29	Limapuluh
Kromosom 30	30	Medang Deras
Kromosom 31	31	Tanjung Tiram
Kromosom 32	32	Gunung Sitoli Alo'a
Kromosom 33	33	Gunung Sitoli Idanoi

3.2.2 Inisialisasi Individu

Individu dalam kasus ini merupakan mewakili solusi lokasi mana yang terpilih untuk dibangun fasilitas. Pada kasus ini, banyaknya individu yang dibangun dalam satu populasi sebanyak 15 individu yang tiap individu berisi 33 kromosom. Nilai dari kromosom berupa bilangan biner (0,1). Berikut populasi awal yang dibangkitkan menggunakan Matlab:

- Individu 1 : 0 1 0 1 1 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1
- Individu 2 : 0 1 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0 1 0 0 1 0 1 1
- Individu 3 : 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0
- Individu 4 : 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 0 0 1 1
- Individu 5 : 1 0 1 1 1 1 0 1 0 1 1 0 0 0 0 1 1 0 0 0 1 1 1 0 0 1 1 1 0 0 1 0 1
- Individu 6 : 1 1 0 1 0 0 1 0 1 0 1 1 0 0 0 1 0 0 1 0 0 0 1 1 0 0 1 1 1 0 0 1 1
- Individu 7 : 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0 0 0 0 0 0 1 1
- Individu 8 : 1 0 1 0 0 0 0 1 0 1 0 0 1 0 1 1 0 0 0 1 0 1 0 0 0 1 0 0 0 1 0 1 0
- Individu 9 : 0 1 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1 0 0 0 1 0 1 0
- Individu 10 : 1 1 0 1 1 0 0 0 0 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 1 0 1 1 1 1
- Individu 11 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 12 : 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 1 0 0 1
- Individu 13 : 0 0 1 1 0 0 0 0 0 0 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 0 1 1 1 1 0 1 0
- Individu 14 : 1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

- Individu 15 : 0 0 1 0 1 0 0 1 0 1 0 1 0 1 0 1 0 0 0 0 1 0 1 1 0 0 1 0 1 1 1 1 1

3.2.3 Evaluasi Individu

Setelah membangkitkan populasi awal secara acak, selanjutnya adalah menghitung nilai *fitness* dari tiap-tiap individu dalam populasi. Penghitungan nilai *fitness* menggunakan fungsi *fitness* yang merujuk pada persamaan (2.20). Berikut nilai objektif Z *fitness* yang didapatkan menggunakan bantuan Matlab :

- Individu 1: $Z_1 = 21$. Fitness = 0,0455
- Individu 2: $Z_2 = 41$. Fitness = 0,0238
- Individu 3: $Z_3 = 38$. Fitness = 0.0256
- Individu 4: $Z_4 = 21$. Fitness = 0,0455
- Individu 5: $Z_5 = 19$. Fitness = 0,0500
- Individu 6: $Z_6 = 30$. Fitness = 0,0323
- Individu 7: $Z_7 = 38$. Fitness = 0,0256
- Individu 8: $Z_8 = 18$. Fitness = 0,526
- Individu 9: $Z_9 = 18$. Fitness = 0,0526
- Individu 10: $Z_{10} = 94$. Fitness = 0,0105
- Individu 11: $Z_{11} = 14$. Fitness = 0,0667
- Individu 12: $Z_{12} = 40$. Fitness = 0,0244
- Individu 13: $Z_{13} = 58$. Fitness = 0.0169

- Individu 14: $Z_{14} = 49$. Fitness = 0.0200
- Individu 15: $Z_{15} = 25$. Fitness = 0.0358

3.2.4 Seleksi *Roulette*

Setelah mendapatkan nilai *fitness*, selanjutnya akan dilakukan pemilihan orang tua atau induk yang akan dicrossover dan dimutasi.

Pemilihan dilakukan dengan metode roda *roulette*, yang sesuai dengan langkah yang sudah dijelaskan pada bab sebelumnya. Pada proses perhitungannya, menggunakan bantuan *software* Matlab. Berikut individu hasil pada langkah seleksi :

- Individu 1 = individu 1 : 0 1 0 1 1 1 0 1 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1 1
0 0 0 1 0 0 1
- Individu 2 = individu 2 : 0 1 0 1 1 0 0 1 1 0 0 0 0 1 0 1 1 1 0 0 1 0 0 0 1 0
1 0 0 1 0 1 1
- Individu 3 = individu 3 : 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 0 0
0 1 0 1 0 1 0
- Individu 4 = individu 4 : 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1
0 1 1 0 0 1 1
- Individu 5 = individu 4 : 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1
0 1 1 0 0 1 1
- Individu 6 = individu 4 : 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 0 1 0 0 1
0 1 1 0 0 1 1

- Individu 7 = individu 7 : 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0
0 0 0 0 0 1 1
- Individu 8 = individu 7 : 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 1 0 0 1 1 0 1 0
0 0 0 0 0 1 1
- Individu 9 = individu 9 : 0 1 0 0 0 0 0 1 1 0 1 0 1 0 1 0 0 0 0 1 0 1 1 1 1 1
0 0 0 1 0 1 0
- Individu 10 = individu 11 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0
1 1 0 0 0 0 1 1
- Individu 11 = individu 11 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0
1 1 0 0 0 0 1 1
- Individu 12 = individu 11 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0
1 1 0 0 0 0 1 1
- Individu 13 = individu 11 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0
1 1 0 0 0 0 1 1
- Individu 14 = individu 12 : 0 1 1 0 1 1 1 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0
0 1 1 1 1 0 0 1
- Individu 15 = individu 14 : 1 0 1 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0
1 0 0 0 0 0 0 0

3.2.5 Pindah Silang (*Crossover*)

Setelah terpilih induk-induk dari proses sebelumnya, selanjutnya individu-individu tersebut dilakukan pindah silang *crossover* dengan metode *crossover*

satu titik. *Crossover* satu titik adalah mendapatkan kromosom-kromosom anak dengan cara menggabungkan dari kromosom 2 induk yang dipotong dari satu titik. Titik potong ditentukan secara acak. Jika nilai gen yang dimutasi bernilai 1 maka diganti menjadi 0, dan sebaliknya.

Crossover dilakukan sebanyak individu yaitu 15 kali dengan probabilitas populasi sebanyak 0.60. Berikut hasil dari *crossover* :

- Individu 1 : 1 0 1 1 0 1 1 1 1 0 1 0 1 0 1 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1
- Individu 2 : 0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0
- Individu 3 : 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0 1 0 1 0
- Individu 4 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 5 : 0 1 0 1 1 1 0 1 1 0 1 0 1 0 1 0 1 1 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1
- Individu 6 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 7 : 1 0 1 0 1 0 1 1 0 0 1 0 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0
- Individu 8 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 9 : 1 0 1 1 0 1 0 0 0 0 1 0 0 1 0 1 0 1 0 1 1 1 0 1 0 0 1 0 1 1 0 0 1 1
- Individu 10 : 1 0 1 1 0 1 1 0 0 0 1 1 1 0 1 1 0 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1
- Individu 11 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 12 : 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 0 0 0 1 0 0 0 1 1
- Individu 13 : 1 0 1 0 1 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 14 : 0 1 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 15 : 1 0 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0

3.2.6 Mutasi

Selanjutnya akan dilakukan proses mutasi dengan cara mutasi satu titik. Banyaknya kromosom yang akan dimutasi adalah 10% dari jumlah seluruh kromosom.

Berikut hasil mutasi yang diperoleh dengan bantuan Matlab.

- Individu 1 : 0 0 1 1 0 1 1 1 1 0 0 1 1 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1
- Individu 2 : 0 1 0 1 1 0 0 1 1 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 0 1 1 1 0
- Individu 3 : 0 1 0 1 1 0 0 1 1 0 0 1 0 0 1 0 1 0 0 1 1 0 0 1 1 0 0 0 0 1 0 0 0
- Individu 4 : 1 0 1 0 0 0 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 1 0 0 1 1 0 0 0 0 1 1
- Individu 5 : 0 1 0 1 1 1 0 1 1 0 1 0 1 0 0 0 0 1 1 0 1 0 0 0 1 0 1 0 0 1 0 1 1
- Individu 6 : 1 0 1 0 0 0 1 1 0 0 1 0 1 1 0 1 0 0 0 1 0 1 0 0 0 1 1 1 0 0 0 1 1
- Individu 7 : 1 0 1 0 1 0 1 1 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0
- Individu 8 : 1 0 1 0 1 1 1 1 0 0 0 1 1 1 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 9 : 0 0 1 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1 1 1 1 0 1 0 0 1 0 1 1 1 0 1 1
- Individu 10 : 1 0 1 1 0 1 1 0 0 0 1 1 1 0 1 0 1 1 1 0 0 0 0 0 1 1 0 0 0 1 0 0 1
- Individu 11 : 1 0 1 0 1 0 1 1 0 1 1 0 1 1 0 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 0 1 1
- Individu 12 : 1 1 1 1 1 0 1 0 0 1 1 1 0 0 1 1 1 0 1 0 1 0 1 1 0 0 0 1 1 0 0 1 1
- Individu 13 : 0 0 1 0 1 1 1 1 0 0 1 0 1 1 0 0 0 0 0 1 0 1 0 0 0 1 1 0 0 0 0 0 1
- Individu 14 : 1 1 0 1 0 1 0 0 1 1 0 1 0 0 0 0 0 1 0 1 0 1 0 0 1 1 1 0 1 0 0 1 1
- Individu 15 : 1 0 1 0 1 0 1 1 0 0 1 1 0 1 1 1 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 0 0

Gen-gen yang dimutasi untuk hasil diatas tersaji pada tabel berikut :

Tabel 3.4: Kromosom yang Dimutasi

individu	Kromosom yang dimutasi
1	1,11,12,16,17,29
2	15,18,22,31
3	16,19,23,25,28,32
4	5,23
5	15,16
6	5,16,28
7	10,26,27
8	6,11,12,18
9	1,6,19,30
10	16,17
11	10,18
12	20,29
13	1,6,32
14	1,25,29
15	25,30

3.2.7 *Alternate Procedure*

Langkah selanjutnya adalah dengan *alternate procedure* menggunakan algoritma *greedy adding*. Pada hasil ini akan mencari solusi kendala kedua. Jika j yang terpilih (gen bernilai 1 pada individu) lebih dari P maka dari j yang sudah terpilih akan dipilih hanya P titik. Jika j terpilih kurang dari P , maka akan ditambahkan j sampai memenuhi syarat P lokasi terpilih.

Berikut hasil dari *alternate* prosedur untuk semua individu untuk $P = 5$:

- Individu 1 : 0 0 0 1 0 0 0 1 0 1 1 0 0 0 0 0 1 0
- Individu 2 : 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0
- Individu 3 : 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0
- Individu 4 : 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0

- Individu 5 : 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1 0
- Individu 7 : 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0
- Individu 8 : 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0 1 0
- Individu 9 : 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 1 0
- Individu 10 : 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1
- Individu 11 : 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 1 0
- Individu 12 : 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 1 0
- Individu 13 : 0 0 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 1
- Individu 14 : 0 0 0 0 0 1 0 0 1 0 1 0 1 0 0 0 0 1 0
- Individu 15 : 0 0 0 0 0 0 0 1 0 0 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0

Dengan nilai objektifnya ada pada tabel 3.5 berikut ini.

Tabel 3.5: Nilai Objektif dari Proses *Alternate*

Individu	Nilai Objektif	Nilai <i>fitness</i>	Individu	Nilai Objektif	Nilai <i>fitness</i>
1	87	0,0114	9	76	0,0130
2	102	0,0097	10	67	0,0147
3	76	0,0130	11	90	0,0110
4	113	0,0088	12	84	0,0118
5	103	0,0096	13	88	0,0112
6	116	0,0085	14	66	0,0149
7	84	0,0118	15	48	0,0204
8	114	0,0087			

- Individu 15 : 0 0 0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0

Dengan nilai objektif (Z) tiap individu tersaji pada tabel 3.6 yang merupakan hasil *fitness* dari proses *interchange*. Berdasarkan tabel 3.6 berubah menjadi lebih baik dari nilai *fitness* dari proses sebelumnya. Ini berarti bahwa proses *interchange* efektif dapat memperbaiki nilai tiap individu.

Tabel 3.6: Nilai Objektif dari Proses *Interchange*

Individu	Nilai Objektif	Nilai <i>Fitness</i>
1	61	0,0161
2	76	0,0130
3	61	0,0161
4	87	0,0114
5	77	0,0128
6	90	0,0110
7	63	0,0156
8	88	0,0112
9	76	0,0130
10	67	0,0147
11	90	0,0110
12	84	0,0118
13	88	0,0112
14	66	0,0149
15	48	0,0204

3.2.9 Penggantian Populasi Baru

Setelah mendapatkan hasil dari langkah *interchange* individu, maka didapatkan individu-individu baru yang merupakan populasi baru yang akan dilakukan iterasi selanjutnya. Iterasi dilakukan sampai memenuhi syarat pemberhentian pencarian solusi.

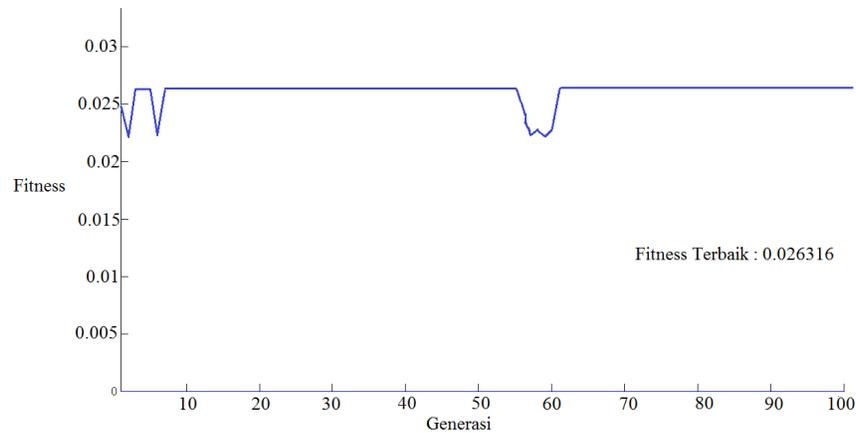
3.3 Hasil Komputasi

Berikut hasil untuk semua batasan P . Algoritma memetika merupakan algoritma bersifat *random generator*, sehingga setiap melakukan proses seleksi maka akan menghasilkan solusi yang tidak selalu sama. Oleh karena itu, diperlukan beberapa kali percobaan dalam mengaplikasikan algoritma memetika menggunakan *software* Matlab agar didapatkan solusi yang optimal, dengan cara mencoba beberapa nilai ukuran populasi dan jumlah generasi yang berbeda. Tabel 3.7 menunjukkan hasil dari seluruh penghitungan untuk P sebesar 5. Pada tabel dapat disimpulkan bahwa nilai *fitness* terbaik adalah 0,0263.

Tabel 3.7: Hasil Percobaan

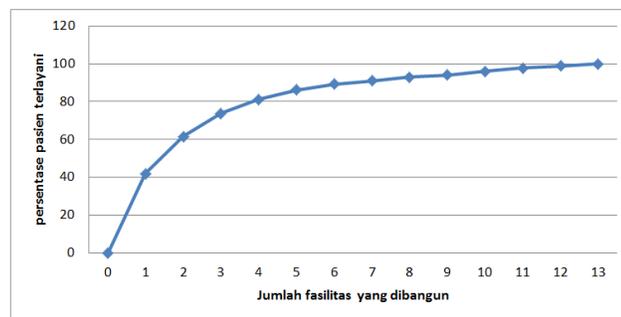
Ukuran populasi	Jumlah generasi	Nilai <i>Fitness</i>	Pasien tak terlayani	lokasi terpilih
15	50	0,0263	37	5,12,17,21,36
15	50	0,0263	37	5,12,21,17,36
15	50	0,0263	37	5,12,21,17,36
15	100	0,0222	44	1,4,17,25,26
15	100	0,0263	37	5,12,17,25,26
15	100	0,0263	37	5,12,17,25,26
15	150	0,0263	37	5,12,17,25,26
15	150	0,0263	37	5,12,17,25,26
15	150	0,0263	37	5,12,17,25,26
15	150	0,0263	37	5,12,17,25,26
20	50	0,0263	37	5,12,17,25,26
20	50	0,0263	37	5,12,17,21,26
20	50	0,0263	37	5,12,17,25,26
20	100	0,0263	37	5,12,17,25,26
20	100	0,0263	37	5,12,17,21,26
20	100	0,0250	39	5,9,12,21,26
20	150	0,0263	37	5,12,17,21,26
20	150	0,0263	37	5,12,17,25,26
20	150	0,0263	37	5,12,17,25,26
25	50	0,0263	37	5,12,17,25,26
25	50	0,0263	37	5,12,17,21,26
25	50	0,0263	37	5,12,17,21,26
25	100	0,0263	37	5,12,17,25,26
25	100	0,0263	37	5,12,17,25,26
25	100	0,0263	37	5,12,17,21,26
25	150	0,0263	37	5,12,17,21,26
25	150	0,0263	37	5,12,17,21,26
25	150	0,0263	37	5,12,17,21,26

Gambar 3.2 menunjukkan pergerakan nilai *fitness* sepanjang iterasi. Gambar menunjukkan nilai *fitness* berubah dan cenderung menunjukkan perubahan hasil yang lebih baik pada setiap iterasi penghitungan.



Gambar 3.2: Pergerakan Nilai *Fitness*

Selanjutnya, akan ditunjukkan grafik solusi optimal maksimum *covering* yang menunjukkan bertambahnya jumlah pasien yang terlayani seiring dengan pertambahan jumlah fasilitas yang dibangun. Solusi yang paling optimal adalah dengan membangun 13 fasilitas kesehatan yang dapat melayani semua pasien yang ada.



Gambar 3.3: Grafik Persentase Pasien Terlayani

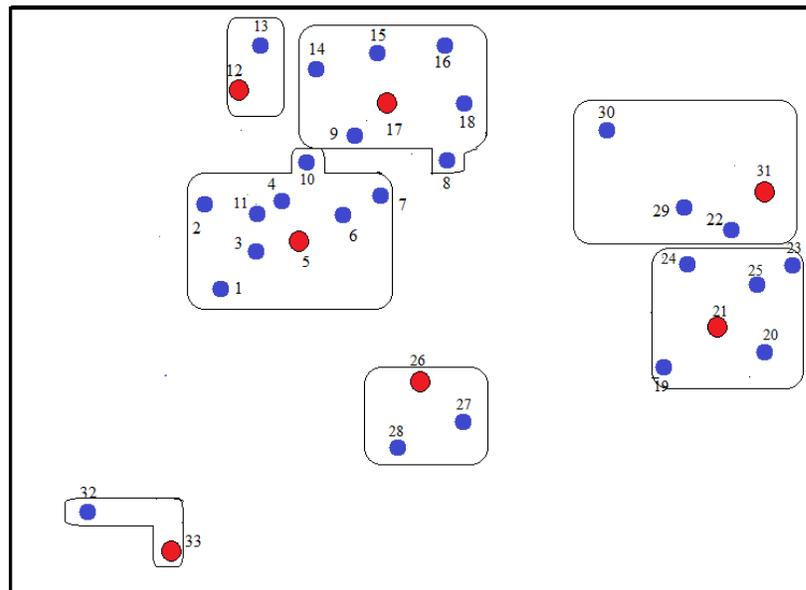
Tabel 3.8: Persentase Pasien Terlayani

Fasilitas dibangun	Persentase Pasien terlayani	Nilai <i>Fitness</i>
1	41,58	0,0063
2	61,48	0,0095
3	73,70	0,0139
4	81,11	0,0192
5	86,30	0,0263
6	89,26	0,0333
7	91,11	0,0400
8	92,96	0,0500
9	94,07	0,0588
10	96,29	0,0909
11	97,78	0,0143
12	98,89	0,2500
13	100	1

Tabel 3.9: Hasil Komputasi

Fasilitas yang ditempatkan	Nilai <i>Fitness</i>	Pasien tak terlayani	lokasi terpilih
1	0,0063	157	9
2	0,0095	104	5,17
3	0,0139	71	5,17,21
4	0,0192	51	5,17,21,26
5	0,0263	37	5,12,17,25,26
6	0,0333	29	5,12,17,21,22,26
7	0,0400	24	5,12,17,21,26,31,33
8	0,0500	19	5,12,17,21,22,26,28,33
9	0,0588	16	5,7,12,15,19,22,26,28,30
10	0,0909	10	5,8,17,21,26,27,28,30,31,33

Tabel 3.9 menunjukkan hasil output lokasi terbaik untuk semua P . Jika solusi sudah efektif jika penempatan dapat melayani ada 90% dari total pasien dapat terlayani, maka berdasarkan tabel 3.8 banyaknya puskesmas keliling yang dibangun sebanyak 7 dengan persentase terlayani sebesar 91,11% dengan lokasi terpilihnya yaitu di Kecamatan Biru-biru, Sunggal, Lubuk Pakam, Sei Dadap, Rantau Selatan, Tanjung Tiram, dan Gunung Sitoli Idanoi. Tetapi, solusi agar semua pasien yang terlayani ditempatkan sebanyak 13 puskesmas keliling di Kecamatan Biru-biru, Galang, Sunggal, Lubuk Pakam, Sei Dadap, Meranti, Rantau Selatan, Kota Pinang, Kampung Rakyat, Limapuluh, Medang Deras, Gunung Sitoli Alo'oa dan Gunung Sitoli Idanoi. Berikut Gambar daerah jangkauan dari 7 lokasi terpilih. Dimana, titik berwarna merah menunjukkan lokasi yang akan dibangun fasilitas kesehatan dan titik biru yang ada dalam garis batas menunjukkan titik-titik permintaan yang dilayani oleh lokasi tersebut.



Gambar 3.4: Daerah Jangkauan Setiap Fasilitas Kesehatan

BAB IV

PENUTUP

4.1 Kesimpulan

1. Pencarian solusi untuk permasalahan penempatan fasilitas dapat dilakukan dengan menggunakan algoritma memetika dengan penambahan langkah pencarian lokal dapat digunakan sebagai pemroses sesudah solusi terbentuk.
2. Berdasarkan proses penghitungan dengan tujuan meminimalkan jumlah pasien tak terlayani, didapatkan output solusi untuk tingkat pelayanan sebesar 90% maka 7 lokasi yang dipilih untuk ditempatkan fasilitas kesehatan adalah lokasi di Kecamatan Biru-biru, Galang, Sunggal, Lubuk Pakam, Sei Dadap, Kisaran Timur, dan Rantau Selatan.
3. Berdasarkan proses penghitungan dengan tujuan agar semua pasien dapat terlayani, maka banyaknya fasilitas kesehatan yang harus ditempatkan sebanyak 13 pada lokasi Biru-biru, Galang, Sunggal, Lubuk Pakam, Sei Dadap, Meranti, Rantau Selatan, Kota Pinang, Kampung Rakyat, Limapuluh, Medang Deras, Gunung Sitoli Alo'oa dan Gunung Sitoli Idanoi.

4.2 Saran

1. Hasil solusi yang dihasilkan oleh algoritma memetika dapat berbeda untuk setiap percobaan yang dilakukan. Sehingga perlu dibandingkan dengan metode lain agar dapat diketahui hasil optimalnya.
2. Perlu dibandingkan mencari solusi permasalahan ini dengan metode lain untuk memperkuat keefektifan algoritma memetika.
3. Penambahan input lainnya pada permasalahan sehingga solusi yang didapatkan lebih spesifik dan optimal.
4. Metode memetika dapat digunakan untuk menyelesaikan berbagai macam persoalan dalam kehidupan sehari-hari seperti *locating facility* halte bus, peramalan, Traveling Salesman Problem (TSP), dan penjadwalan, dan lainnya. Oleh karena itu, diharapkan metode ini dapat juga diterapkan dalam bidang lain, selain bidang pengoptimalan permasalahan penempatan fasilitas (*locating facility problem*) pada bidang kesehatan.

DAFTAR PUSTAKA

- Alegre, Jesus F., Ada, Alvarez., Silvia, C dan Joaquin, A. (2010), Use of Memetic Algorithms to Solve A Stochastic Location Model: Health Resources for Diabetics in Some Provinces of Castilla-León, *XIII Jornadas de ASEPUMA*.
- Arogundade, O.T., Akinwale A.T., Adekoya A.F., dan Awe A.G., Model for Fire and Emergency Service Facility Location Selection: A Case Study in Nigeria, *Journal of Theoretical and Applied Information Technology, 2005-2009*. University of Agriculture: Nigeria.
- Church, Richard dan Charles ReVell (1974), The Maximal Covering Location Problem, *Papers of The Regional Science Assosiation, 32, 101-118*. The Johns Hopkins University.
- Daskin, M. S. dan L.K. Dean (2004), *Location of Health Care Facilities*, Chapter 3 in the Handbook of OR/MS in Health Care: A Handbook of Methods and Applications. Evanston: Northwestern University.
- Dzator, M. dan J. Dzator (2015), "An Efficient Modified Greedy Algorithm for the P-Median Problem". *21st International Congress on Modelling and Simulation*. Gold Coast, Australia.
- Farahani, R.Z., Hekmatfar, Masound (2009), *Facility Location: Concepts, Models, Algorithms and Case Studies*, Physica-Verlag Heidelberg.
- Krasnogor, N., Alberto A. dan Joaquin, P. (2001), *Memetic Algorithms*. New York: Springer-Verlag.

Love, R.F., Morris, J.G dan Wesolowsky, G.O,(1988), *Facilities Location: Models and Methods*, North Holland.

Neri, F. (2010), *Handbook of Memetic Algorithms*,Edisi ke-2. Verlag Berlin Heidelberg: Springer-Verlag.

LAMPIRAN I PROSES PENCARIAN LOKAL

Proses Pencarian Lokal Algoritma *Greedy Adding*

- Individu 5

Tabel 4.1: Total jarak 1 fasilitas individu 2

	1	2	3	4	5	6
1	0	19,8	14,7	20,7	11	16
2	19,8	0	9	17,6	13,2	22
3	14,7	9	0	19,5	17,2	18,2
4	20,7	17,6	19,5	0	24,2	14,1
5	11	13,2	17,2	24,2	0	22,9
6	16	22	18,2	14,1	22,9	0
Σ	82,2	81,6	78,6	96,1	88,5	93,1

Titik 3 terpilih karena memiliki jarak minimum. Selanjutnya, dipilih untuk titik kedua.

Tabel 4.2: Total jarak 2 fasilitas individu 2

	1	2	4	5	6
1	0	14,7	14,7	11	16
2	9	0	9	9	9
3	0	0	0	0	0
4	19,5	17,6	0	19,5	14,1
5	11	13,2	17,2	0	17,2
6	16	18,2	14,1	18,2	0
Σ	55,5	63,7	55	57,7	56,3

Maka, terpilih titik 4 sebagai lokasi terpilih yang kedua. Selanjutnya untuk memilih lokasi ketiga.

Tabel 4.3: Total jarak 3 fasilitas individu 2

	1	2	5	6
1	0	14,7	11	14,7
2	9	0	9	9
3	0	0	0	0
4	0	0	0	0
5	11	13,2	0	17,2
6	14,1	14,1	14,1	0
Σ	34,1	42	34,1	40,9

Titik 1 dipilih. Sehingga, 3 lokasi fasilitas terpilih untuk individu 5 adalah 1,3 dan 4. Dikodekan dengan 101100.

LAMPIRAN II PENGHITUNGAN *ALTERNATE PROCEDURE*

- Untuk $k = 5$, dan $m = 1$ sehingga fasilitas terpilihnya 3,4 dan 5:

i	1	2	3	4	5	6
$d_{ij}min$	11	9	0	0	0	14,1
r_i	0	0	0	0	0	1

$$v_{51} = 95(0) + 25(0) + 70(0) + 82(0) + 100(0) + 65(1) = 65$$

- Untuk $k = 6$, dan $m = 1$ sehingga fasilitas terpilihnya 3,4 dan 6:

i	1	2	3	4	5	6
$d_{ij}min$	14,7	9	0	0	17,2	0
r_i	1	0	0	1	0	

$$v_{61} = 95(1) + 25(0) + 70(0) + 82(0) + 100(1) + 65(0) = 195$$

- Untuk $k = 2$, dan $m = 3$ sehingga fasilitas terpilihnya 1,2 dan 4:

i	1	2	3	4	5	6
$d_{ij}min$	0	0	9	0	11	14,1
r_i	0	0	0	0	0	1

$$v_{23} = 95(0) + 25(0) + 70(0) + 82(1) + 100(0) + 65(1) = 65$$

- Untuk $k = 5$, dan $m = 3$ sehingga fasilitas terpilihnya 1,5 dan 4:

i	1	2	3	4	5	6
$d_{ij}min$	0	13,2	14,7	0	0	14,1
r_i	0	0	1	0	0	1

$$v_{53} = 95(0) + 25(0) + 70(1) + 82(0) + 100(0) + 65(1) = 135$$

i	1	2	3	4	5	6
$d_{ij}min$	0	17,2	14,7	0	11	0
r_i	0	1	1	0	0	0

- Untuk $k = 6$, dan $m = 3$ sehingga fasilitas terpilihnya 1,4 dan 6:

$$v_{63} = 95(0) + 25(1) + 70(1) + 82(0) + 100(0) + 65(0) = 95$$

- Untuk $k = 2$, dan $m = 4$ sehingga fasilitas terpilihnya 1,2 dan 3:

i	1	2	3	4	5	6
$d_{ij}min$	0	0	0	17,6	11	16
r_i	0	0	0	1	0	1

$$v_{24} = 95(0) + 25(0) + 70(10) + 82(1) + 100(0) + 65(1) = 147$$

- Untuk $k = 5$, dan $m = 4$ sehingga fasilitas terpilihnya 1,3 dan 5:

i	1	2	3	4	5	6
$d_{ij}min$	0	9	0	19,5	0	16
r_i	0	0	0	1	0	1

$$v_{54} = 95(0) + 25(0) + 70(0) + 82(1) + 100(0) + 65(1) = 147$$

- Untuk $k = 6$, dan $m = 4$ sehingga fasilitas terpilihnya 1,3 dan 6:

i	1	2	3	4	5	6
$d_{ij}min$	0	9	0	14,1	11	0
r_i	0	0	0	1	0	0

$$v_{64} = 95(0) + 25(0) + 70(0) + 82(1) + 100(0) + 65(0) = 82$$

LAMPIRAN III Program Menggunakan Matlab 7.8.0 (R2009)

```
%Program Utama Memetika
clear;
clc;

%input
Pop=input ('Jumlah individu=');
P=input('Jumlah Fasilitas Kesehatan yang akan dibangun=');
g=input ('Jumlah Generasi=');

%open data
data = fopen('pasien2.txt');
C = textscan(data, '%f %f', 'delimiter', ',', '');
fclose(data);
h=[C{2}]; %data jumlah pasien
dij=[0 10.68 17.03 20.85 12.27 20.01 29.65 40.29 36.65 27.03
24.57 32.44 49.23 43.01 45.88 52.44 45.23 46.45 141.86 129.57
129.40 122.15 133.91 124.42 126.62 200.83 234.09 231.56 103.63
89.85 191.49 251.11 261.55;
10.68 0 9.86 16.23 15.35 23.92 33.52 42.54 35.10 24.42 21.40
23.39 39.89 37.85 41.91 48.47 43.33 45.90 149.02 137.08 136.57
```

128.21 140.36 131.23 133.32 210.49 243.65 240.58 108.64 93.29
183.76 258.84 269.48;
17.03 9.86 0 6.94 13.37 20.14 28.55 35.73 26.23 15.50 12.40
15.60 32.38 27.99 32.14 38.67 34.13 37.25 145.23 133.74 132.76
123.36 135.83 127.12 129.08 210.28 243.17 239.12 102.94 86.45
171.67 267.65 278.32;
20.85 16.23 6.94 0 12.59 16.50 23.43 29.38 19.29 8.57 5.47
15.75 31.45 22.29 25.18 32.40 27.25 30.32 139.89 128.66 127.43
117.53 130.15 121.66 123.55 206.87 239.56 234.93 96.78 79.89
164.61 271.92 282.23;
12.27 15.35 13.37 12.59 0 8.64 18.40 28.35 24.69 15.95 14.17
27.77 43.96 32.82 34.79 41.23 33.24 34.21 133.77 121.93 121.31
112.86 125.01 115.92 117.99 197.07 230.03 226.30 93.42 78.60
176.89 261.03 270.94;
20.01 23.92 20.14 16.50 8.64 0 9.79 20.32 20.22 15.02 14.94
32.17 47.07 31.50 31.91 37.87 28.31 27.94 125.50 113.82 113.04
104.32
116.54 107.55 109.59 190.49 223.27 219.0 84.78 70.07 105.94
264.17 273.52;
29.65 33.52 28.55 23.43 18.40 9.79 0 11.47 17.78 18.34 19.84
38.04 51.05 31.59 30.06 34.99 24.02 21.79 116.69 105.30 104.32
94.98 107.35 98.58 100.55 184.17 216.65 211.63 75.12 60.29
139.32 269.15 277.89;
40.29 42.54 35.73 29.38 28.35 20.32 11.47 0 14.86 21.76 24.34
41.35 51.30 28.71 25.06 28.27 16.89 12.50 111.63 101.02 99.42

88.48 101.28 93.20 94.96 183.58 215.52 209.23 67.42 50.82 122.02
279.64 288.05;

36.65 35.10 26.23 19.29 24.69 20.22 17.78 14.86 0 10.75 13.84
27.66 36.45 14.29 12.31 17.75 8.57 11.08 125.88 115.56 113.55
102.11 115.09 107.32 109.03 198.39 230.37 224.05 80.35 62.07
140.83 284.40 293.72;

27.03 24.42 15.50 8.57 15.95 15.02 18.34 21.76 10.75 0 3.10
20.03 32.87 17.02 18.93 25.45 18.92 21.75 133.16 122.27 120.74
110.24 123.01 114.80 116.60 202.47 234.89 229.55 89.09 71.70
156.83 276.97 286.81;

24.57 21.40 12.40 5.47 14.17 14.94 19.84 24.41 13.84 3.10 0
18.20 32.18 18.71 21.32 27.91 21.93 24.85 135.52 124.50 123.08
112.82 125.53 117.21 119.05 203.96 236.49 231.41 91.82 74.63
159.13 275.06 285.07;

32.44 23.39 15.60 15.75 27.77 32.17 38.04 41.35 27.66 20.03
18.20 0 16.80 21.02 26.89 32.44 32.83 37.86 152.98 142.23 140.58
129.63 142.53 134.55 136.29 222.13 254.68 249.55 108.0 89.55
181.04 281.77 292.84;

49.23 39.89 32.38 31.45 43.96 47.07 51.05 51.30 36.45 32.87
32.18 16.80 0 23.83 29.63 32.29 38.06 44.00 161.59 151.67 149.37
137.00 150.18 143.02 144.53 234.62 266.74 260.48 114.50 94.54
187.28 297.59 308.96;

43.01 37.85 27.99 22.29 32.82 31.50 31.59 28.71 14.29 17.02
18.71 21.02 23.83 0 6.32 11.49 14.24 20.19 137.79 128.01 125.60
113.18 126.36 119.22 120.71 212.20 243.99 237.17 90.74 71.06

151.54 293.77 303.75;
45.88 41.91 32.14 25.18 34.79 31.91 30.06 25.06 12.31 18.93
21.32 26.89 29.63 6.32 0 6.60 8.92 14.92 132.28 122.73 120.15
107.44 120.56 113.7 115.14 207.82 239.4 232.18 84.87 64.99 143.46
295.66 305.27;
52.44 48.47 38.67 32.40 41.23 37.87 34.99 28.27 17.75 25.45
27.91 32.44 32.29 11.49 6.60 0 11.42 16.41 131.45 122.42 119.49
106.08 119.36 93.20 114.20 208.95 240.18 232.31 83.21 62.69
148.53 301.90 311.38;
45.23 43.33 34.13 27.25 33.24 28.31 24.02 16.89 8.57 18.92
21.93 32.83 38.06 14.24 8.92 11.42 0 6.02 123.58 113.90 111.41
98.95 112.12 112.90 106.48 198.94 230.49 223.28 76.57 57.17
132.27 292.36 301.46;
46.45 45.90 37.25 30.32 34.21 27.94 21.79 12.50 11.08 21.75
24.85 37.86 44.00 20.19 14.92 16.41 6.02 0 117.60 107.88 105.42
93.08 106.22 99.03 100.52 193.06 224.55 217.27 70.81 51.74
136.09 290.93 299.66;
141.86 149.02 145.23 139.89 133.77 125.50 116.69 111.63 125.88
133.16 135.52 152.98 161.59 137.79 132.28 131.45 123.58 117.60
0 13.66 12.47 26.89 14.77 18.58 17.33 90.91 116.28 103.54
49.73 71.26 32.42 301.57 302.32;
129.57 137.08 133.74 128.66 121.93 113.82 105.30 101.02 115.56
122.27 124.50 142.23 151.67 128.01 122.73 122.42 113.90 107.88
13.66 0 7.01 24.17 18.21 12.76 13.83 92.62 120.59 110.26
43.91 64.78 32.06 289.97 291.26;

129.40 136.57 132.76 127.43 121.31 113.04 104.32 99.42 113.55
120.74 123.08 140.58 149.37 125.60 120.15 119.49 111.41 105.42
12.47 7.01 0 17.78 11.46 6.98 7.17 98.37 125.54 114.19 39.00
60.31 25.32 293.91 287.45;
122.15 128.21 123.36 117.53 112.86 104.32 94.98 88.48 102.11
110.24 112.82 129.63 137.00 113.18 107.44 106.08 98.95 93.08
26.89 24.17 17.78 0 13.31 11.45 10.63 116.09 142.76 130.44
23.17 44.68 8.53 305.05 307.62;
133.91 140.36 135.83 130.15 125.01 116.54 107.35 101.28 115.09
123.01 125.53 142.53 150.18 126.36 120.56 119.36 112.12 106.22
14.77 18.21 11.46 13.31 0 10.96 8.20 105.57 131.01 117.83
36.46 57.93 17.69 307.12 308.79;
124.42 131.23 127.12 121.66 115.92 107.55 98.58 93.20 107.32
114.80 117.21 134.55 143.02 119.22 113.7 93.20 112.90 99.03
18.58 12.76 6.98 11.45 10.96 0 2.78 105.14 132.49 121.13
32.02 53.36 19.62 297.19 299.20;
126.62 133.32 129.08 123.55 117.99 109.59 100.55 94.96 109.03
116.60 119.05 136.29 144.53 120.71 115.14 114.20 106.48 100.52
17.33 13.83 7.17 10.63 8.20 2.78 0 105.4 132.36 120.52
32.54 54.03 18.24 299.82 301.77;
200.83 210.49 210.28 206.87 197.07 190.49 184.17 183.58 198.39
202.47 203.96 222.13 234.62 212.20 207.82 208.95 198.94 193.06
90.91 92.62 98.37 116.09 105.57 105.14 105.47 0 33.59 41.08
136.37 156.61 122.91 270.59 265.20;
234.09 243.65 243.17 239.56 230.03 223.27 216.65 215.52 230.37

234.89 236.49 254.68 266.74 243.99 239.4 240.18 230.49 224.55
116.28 120.59 125.54 142.76 131.01 132.49 132.36 33.59 0
24.50 164.42 185.36 148.69 291.78 284.47;
231.56 240.58 239.12 234.93 226.30 219.00 211.63 209.23 224.05
229.55 231.41 249.55 260.48 237.17 232.18 232.31 223.28 217.27
103.54 110.26 114.19 130.44 117.83 121.13 120.52 41.08 24.50 0
153.06 174.48 135.37 310.11 303.83;
103.63 108.64 102.94 96.78 93.42 84.78 75.12 67.42 80.35 89.09
91.82 108.0 114.50 90.74 84.87 83.21 76.57 70.81 49.73 43.91
39.00 23.17 36.46 32.02 32.54 136.37 164.42 153.06 0 21.55
21.47 304.50 308.63;
89.85 93.29 86.45 79.89 78.60 70.07 60.29 50.82 62.07 71.70
74.63 89.55 94.54 71.06 64.99 62.69 57.17 51.74 71.26 64.78
60.31 44.68 57.93 53.36 54.03 156.61 185.36 174.48 21.55 0
42.14 307.39 312.91;
191.49 183.76 171.67 164.61 176.89 105.94 139.32 122.02 140.83
156.83 159.13 181.04 187.28 151.54 143.46 148.53 132.27 136.09
32.42 32.06 25.32 8.53 17.69 19.62 18.24 122.91 148.69 135.37
21.47 42.14 0 312.87 315.65;
251.11 258.84 267.65 271.92 261.03 264.17 269.15 279.64 284.40
276.97 275.06 281.77 297.59 293.77 295.66 301.90 292.36 290.93
301.57 289.97 293.91 305.05 307.12 297.19 299.82 270.59 291.78
310.11 304.50 307.39 312.87 0 20.86;
261.55 269.48 278.32 282.23 270.94 273.52 277.89 288.05 293.72
286.81 285.07 292.84 308.96 303.75 305.27 311.38 301.46 299.66

```
302.32 291.26 287.45 307.62 308.79 299.20 301.77 265.20 284.47
303.83 308.63 312.91 315.65 20.86 0];

m= size(dij,2); %Jumlah Kandidat lokasi;

N=17; %jarak maksimal pelayanan

%mencari nilai xij dan yij
xij=zeros(size(dij)); %0=jarak <N 1=jarak>N
yij=zeros(size(dij)); %1=jarak>N 1=jarak<N
ndata=numel(dij);
i=1;
while i<=ndata
    if dij(i)>N
        xij(i)=1;
    yij(i)=0;
    else
        xij(i)=0;
    yij(i)=1;
    end
    i=i+1;
end

%insialisasi grafis
pjh=30;
```

```
fth=1/pjh;
bg=fth;
hfig=figure;
hold on
set(hfig, 'position', [50,50,600,400]);
set(hfig, 'DoubleBuffer', 'on');
axis([1 g 0 bg]);
hbestplot1= plot(1:g,zeros(1,g));
hbestplot2= plot(1:g,zeros(1,g));
htext1= text(0.6*g, 0.25*bg, sprintf('Fitness Terbaik:
%7.6f', 0.0));

xlabel('Generasi');
ylabel('Fitness');
hold off
drawnow;

%Inisialisasi Populasi
G=floor(rand(Pop,m)*2); %1 dibangun 0 tidak
display(G);

%evaluasi
[Z] = fo(G,m,yij,h,Pop);
display(Z);
```

```
gen=1;
while gen<=g

%seleksi Roulette
[newpop] = seleksi(G,Z,Pop);

%crossover
offspring = crossover(newpop,Pop);

%mutasi
hmutasi= mutasi(offspring);

%alternate
[indv,f] = alternate(hmutasi,dij,yij,m,P,h);

%interchange
[z_interchange,h_interchange,finter]= interchange(indv,yij,h,f);

%Penggantian populasi untuk generasi selajutnya
G=h_interchange;
Z=z_interchange;
fmax=max(finter);
Zmin=min(Z);
individu=find(Z==Zmin);
solusi=h_interchange(individu(1),:)
```

```
lokasi_terpilih=find(solusi==1);
f_akhir=fmax;

plotvektor1=get(hbestplot1, 'YData');
plotvektor1(gen)=f_akhir;
set(hbestplot1,'YData',plotvektor1);
set(htext1, 'String', sprintf('Fitness terbaik
: %7.6f', f_akhir));
drawnow

gen=gen+1;
end

display(lokasi_terpilih);display(Zmin);display(solusi);
display(f_akhir);
```

```
%Program seleksi roulette
function [newpop] = seleksi(G,Z,Pop)

%fitness
i=1;
while i<=Pop
fitness=1./(Z+1);
```

```
i=i+1;
end
display=fitness;

%Parameter populasi
a = size(fitness,1); %jumlah individu
b = size(fitness,2); %jumlah kandidat lokasi

%probabilitas gen
totalFit = sum(fitness(:,b));
prob=fitness(:,b) / totalFit;
prob=cumsum(prob);

rnums=sort(rand(a,1)); %bangkitkan bilangan random

%Pilih individu G sebagai pop baru
fitIn=1;newIn=1;
while newIn<=a
    if(rnums(newIn)<prob(fitIn))
        newpop(newIn,:) = G(fitIn,:);
        newIn = newIn+1;
    else
        fitIn = fitIn + 1;
    end
end
end
```

```
end
```

```
%Program crossover

function offspring = crossover(newpop,Pop)

rxover=(rand(Pop,1));
pc=0.60;

%memilih ortu yang akan dicrossover
r=1;t=1;
while t<=Pop
if rxover(t)<pc
ortu(r,:)=newpop(t,:);
r=r+1;
end
t=t+1;
end

%menentukan individu untuk disilangkan
sz=size(ortu,1);
n=size(ortu,2);
t=1;
```

```
while t<=Pop
a=randi(n); b=randi(sz);
%if sz==1
%offspring(t,:)=[ortu(b,1:a) ortu(b,a+1:n)];
if b==sz
offspring(t,:)=[ortu(b,1:a) ortu(1,a+1:n)];
else offspring(t,:)=[ortu(b,1:a) ortu(b+1,a+1:n)];
end
t=t+1;
end

end
```

```
%Program mutasi
function hmutasi=mutasi(offspring)
pm=0.1;
k=numel(offspring);
nmut=pm*k;

for i=1:nmut
r=randi(k);
if offspring(r)==0
offspring(r)=1;
end
end
```

```
i=i+1;
    else
        offspring(r)=0;
        i=i+1;
    end
    hmutasi=offspring;
end
```

```
%Program proses alternate
function [indv,f] = alternate(hmutasi,dij,yij,m,P,h)

numvar=numel(hmutasi);
a=hmutasi;
i=1;
while i<=numvar
    if a(i)==0
        a(i)=1;
    else
        a(i)=0;
    end
    i=i+1;
end
```

```

Pop=size(hmutasi,1);
indv=zeros(size(hmutasi)); %individu baru sbg solusi
ind=1; %individu ke- yang dihitung
while ind<=Pop

    %memilih p faskes untuk j=p
    [indv,sj] = tigafaskes(ind,P,hmutasi,indv);

    %memilih p faskes untuk j<p
    [hm] = ktigafaskes(a,ind,P,sj,dij);

    %memilih p faskes untuk j>p
    [hm] = ltigafaskes(hmutasi,ind,P,dij,hm);

    %individu baru
    [indv] = s(hm,m,sj,P,ind,indv);

    %menghitung nilai fo baru
    [f] = fobaru(indv,m,yij,h,Pop);

ind=ind+1;
end
end

function [indv,sj] = tigafaskes(ind,P,hmutasi,indv)

```

```
sj=numel(find(hmutasi(ind,:)));
if sj==P
    indv(ind,:)=hmutasi(ind,:);
else
    indv(ind,:)=indv(ind,:);
end
end

function [hm] = ktigafaskes(a,ind,P,sj,dij)
hm=a(ind,:);
br=size(hm,1);
kl=size(hm,2);
if sj < P
    krng=P-sj;
    j=1;

    for j=1:kl
k(j,:)=a(ind,:);
    end

sdijs=sum(dij);

i=1; %individu yang perlu ditambahkan
while i<= krng
```

```
w=find(hm==1);
ak=size(w,2);

for l=1:ak
    sk(:,l)=sdij(w(l));
end
mn=min(sk);
wj=find(sk== mn);
jtam=w(wj);
hm(jtam)=0;

%perubahan dij
dijb=dij;
k=1;l=1;
while k<=kl %baris
    while l<=kl %kolom
        if dijb(k,l)> dijb(k,jtam);
            dijb(k,l)=dijb(k,jtam);
        end
        l=l+1;
    end
    k=k+1;
end
sdij=sum(dijb);
i=i+1;
```

```
end
hm=hm;
else
hm=zeros(br,kl);
end

end

function [hm] = ltigafaskes(hmutasi,ind,P,dij,hm)

sj=numel(find(hmutasi(ind,:)));
m=size(hmutasi,2);
sdij=sum(dij);

if sj > P
hm=ones(size(hmutasi(ind,:)));
g=hmutasi(ind,:);
dijb=dij;
i=0; %individu yang perlu dipilih
while i<P
    w=find(g==1);
    a=w;
    ak=numel(w);
    l=1;
while l<=ak
```

```
    wk(l)=sdi(a(l));  
    l=l+1;  
end  
  
wks=sort(wk);  
mn=wks(1);  
jter=find(wk== mn);  
e=1;  
if jter > numel(w)  
    mn=wks(1+e);  
    jter=find(wk== mn);  
end  
  
jtam=w(jter(1));  
jtam=jtam(1);  
hm(:,jtam)=0;  
g(:,jtam)=0;  
  
%perubahan dij  
k=1;  
while k<=m %baris  
    for l=1:m %kolom  
        if dijb(k,l)> dijb(k,jtam);  
            dijb(k,l)=dijb(k,jtam);  
        end  
    end  
end
```

```
        end
        k=k+1;
    end
    sdiij=sum(dijb);
    i=i+1;
    end
    hm=hm;
else
    hm=hm;
end
end
```

```
%Program interchange
function [z_interchange,h_interchange,finter]= interchange
    (indv,yij,h,f)

Pop=size(indv,1);
ind=1;
while ind<=Pop
    z=f(ind,:);
    ic=indv(ind,:);
    ich=indv(ind,:);
    hinter=ic;
```

```
zinter=z;
j_tpilih=find(ic==1);
j_tdkpilih=find(ic==0);
jk=numel(j_tpilih);
jj=numel(j_tdkpilih);
k=1;
while k<=jk
    j=1;
    while j<=jj
        ich(j_tpilih(k))=0;
        ich(j_tdkpilih(j))=1;
        sol(j,:)=ich;
        ich=ic;
        j=j+1;
    end

    [zb] =zbaru(sol,yij,h);

    %menghitung nilai z baru
    zbmin=min(zb);
    solmin=find(zb== zbmin);
    inter=sol(solmin,:);
    if zbmin < z
        z=zbmin;
        hinter=inter(1,:);
```

```
        zinter=z;
    end
    k=k+1;
end

h_interchange(ind,:)=hinter;
z_interchange(ind,:)=zinter;

ind=ind+1;
end

%menghitung nilai fitness baru
i=1;
while i<=Pop
finter=1./(z_interchange+1);
i=i+1;
end
end

%menghitung z baru
function [zb] = zbaru(sol,yij,h)

nsol=size(sol,1);
msol=size(sol,2);
```

```
i=1; %individu ke-i
while i<=nsol

for j=1:msol
mi(j,:)=sol(i,:);
end

%menghitung nilai yij
b=mi.*yij;

%menghitung kemungkinan nilai
for j=1:msol
    sumb(j)=sum(b(j,:));

    if sumb(j)<1
        sumb(j)=1;
    else
        sumb(j)=0;
    end

end

end

%menghitung nilai objektif (Z)
zb(i,:)=sumb*h;
%f(i,:)=1./(Z+1);
```

```
i=i+1;  
end  
end
```

LAMPIRAN IV DATA INPUT

Tabel 4.4: Data Penderita HIV/AIDS

Indeks	Kecamatan	Jumlah Penderita
1	Sibolangit	20
2	Kutalimbaru	1
3	Pancur Batu	16
4	Namo Rambe	4
5	Biru-biru	4
6	Tanjung Muda Hilir	6
7	Bangun Purba	3
8	Galang	10
9	Tanjung Morawa	34
10	Patumbak	9
11	Deli Tua	9
12	Sunggal	7
13	Hampanan Perak	7
14	Percut Sei Tuan	23
15	Batang Kuis	4
16	Pantai Labu	2
17	Lubuk Pakam	21
18	Pagar Merbau	3
19	Simpang Empat	3
20	Air Batu	5
21	Sei Dadap	5
22	Meranti	3
23	Air Joman	5
24	Kisaran Barat	8
25	Kisaran Timur	7
26	Rantau Selatan	20
27	Kota Pinang	4
28	Kampung Rakyat	5
29	Limapuluh	4
30	Medang Deras	5
31	Tanjung Tiram	5
32	Gunung Sitoli Alo'a	3
33	Gunung Sitoli Idanoi	5

Tabel 4.5: Data Jarak antar Permintaan dan Calon Kandidat Lokasi 1

I/J	1	2	3	4	5	6	7	8
1	0	10,68	17,03	20,85	12,27	20,01	29,56	40,29
2	10,68	0	9,86	16,23	15,35	23,92	33,52	26,73
3	17,03	9,86	0	6,94	13,37	16,50	28,55	35,73
4	20,85	16,23	6,94	0	12,59	16,50	23,43	29,38
5	12,27	15,35	13,37	12,59	0	8,64	18,40	28,35
6	20,01	23,92	20,14	16,50	8,64	0	9,79	20,32
7	29,65	33,52	28,55	23,43	18,40	9,79	0	11,47
8	40,29	42,54	35,73	29,38	28,35	20,32	11,47	0
9	36,65	35,10	26,23	19,29	24,69	20,22	17,78	14,86
10	27,03	24,42	15,50	8,57	15,95	15,02	18,34	21,76
11	24,57	21,40	12,40	5,47	14,17	14,94	19,84	24,41
12	32,44	23,39	15,60	15,75	27,77	32,17	38,04	41,35
13	49,23	39,89	32,38	31,45	43,96	47,07	51,05	51,30
14	43,01	37,85	27,99	22,29	32,82	31,50	31,59	28,71
15	45,88	41,91	32,14	25,18	34,79	31,91	30,06	25,06
16	52,44	48,47	38,67	32,40	41,23	37,87	34,99	28,27
17	45,23	43,33	34,13	27,25	33,24	28,31	24,02	16,89
18	46,45	45,90	37,25	30,32	34,21	27,94	21,79	12,50
19	141,86	149,02	145,23	139,89	133,77	125,50	116,69	111,63
20	129,57	137,08	133,74	128,66	121,93	113,82	105,30	101,02
21	129,40	136,57	132,76	127,43	121,31	113,04	104,32	99,42
22	122,15	128,21	123,36	117,53	112,86	104,32	94,98	88,48
23	133,91	140,36	135,83	130,15	125,01	116,54	107,35	101,28
24	124,42	131,23	127,12	121,66	115,99	107,55	98,58	93,20
25	126,62	133,32	129,08	123,55	117,99	109,59	100,55	94,96
26	200,83	210,49	210,28	206,87	197,07	190,49	184,17	183,58
27	234,09	240,58	243,17	239,56	230,03	223,27	216,65	215,52
28	231,56	240,58	239,12	234,92	226,30	219,00	211,63	209,23
29	103,56	108,64	120,94	96,78	93,42	84,78	75,12	67,42
30	89,93	86,45	79,89	79,89	78,60	70,07	60,29	50,82
31	191,49	183,76	164,61	164,61	176,89	105,94	139,32	122,02
32	251,49	258,84	271,93	271,62	261,03	264,17	269,15	279,64
33	261,55	269,48	283,32	282,23	270,94	273,52	277,89	288,05

Tabel 4.6: Data Jarak antar Permintaan dan Calon Kandidat Lokasi 2

I/J	9	10	11	12	13	14	15	16
1	36,65	27,03	24,57	32,44	49,23	43,01	45,88	52,44
2	35,10	24,42	21,40	23,39	39,89	37,85	41,91	48,47
3	26,23	15,50	12,40	15,60	32,38	27,99	32,14	38,67
4	19,29	8,57	5,47	15,75	31,45	22,29	25,18	32,40
5	24,69	15,90	14,17	27,77	43,96	32,82	34,79	41,23
6	20,22	15,02	14,49	32,17	47,07	31,50	31,91	37,87
7	17,78	18,34	19,84	38,04	51,05	31,59	30,06	34,99
8	14,86	21,76	24,34	41,35	51,30	28,71	25,06	28,27
9	0	10,75	13,84	27,66	36,45	14,29	12,31	17,75
10	10,75	0	3,10	20,03	32,87	17,02	18,93	25,45
11	13,84	3,10	0	18,20	32,18	18,71	21,32	27,91
12	27,66	20,03	18,20	0	16,80	21,02	26,89	32,44
13	36,45	32,87	32,18	16,80	0	23,83	29,63	32,29
14	14,29	17,02	18,71	21,02	23,83	0	6,23	11,49
15	12,31	18,93	21,32	26,89	29,63	6,32	0	6,60
16	17,75	25,45	27,91	32,44	32,29	11,49	6,60	0
17	8,57	18,92	21,93	32,83	38,06	14,24	8,92	11,42
18	11,08	21,75	24,85	37,86	44,00	20,19	14,92	16,41
19	125,88	113,16	135,52	152,98	161,59	137,79	132,28	131,45
20	115,56	122,27	124,50	142,23	151,67	128,01	122,73	122,42
21	113,55	120,74	123,08	140,58	149,37	125,60	120,15	119,49
22	102,11	110,24	112,82	129,63	137,00	113,18	107,44	106,08
23	115,09	123,01	125,53	142,53	150,18	126,36	120,56	119,36
24	107,32	114,80	117,21	134,55	143,02	119,22	113,70	93,20
25	109,03	116,60	19,05	136,29	144,53	120,71	115,14	114,20
26	198,39	202,47	220,39	222,13	234,62	212,20	207,82	208,95
27	230,37	234,89	236,49	254,68	266,74	243,99	239,40	240,18
28	224,05	229,55	231,41	249,55	260,48	237,17	232,18	232,31
29	80,35	89,09	91,82	108,00	144,50	90,74	84,87	83,21
30	62,07	71,70	74,36	89,55	94,54	71,06	64,99	62,69
31	140,83	156,83	159,15	181,04	187,28	151,54	143,46	148,53
32	284,40	276,97	275,06	281,77	287,59	293,77	295,66	301,90
33	293,72	286,81	285,07	292,84	308,96	303,75	305,27	311,38

Tabel 4.7: Data Jarak antar Permintaan dan Calon Kandidat Lokasi 3

I/J	17	18	19	20	21	22	23	24
1	45.23	46.45	141.86	129.57	129.40	122.15	133.91	124.42
2	43.33	45.90	149.02	137.08	136.57	128.21	140.36	131.23
3	34.13	37.25	145.23	133.74	132.76	123.36	135.83	127.12
4	27.25	30.32	139.89	128.66	127.43	117.53	130.15	121.66
5	33.24	34.21	133.77	121.93	121.31	112.86	125.01	115.92
6	28.31	27.94	125.50	113.82	113.04	104.32	116.54	107.55
7	24.02	21.79	116.69	105.30	104.32	94.98	107.35	98.58
8	16.89	12.50	111.63	101.02	99.42	88.48	101.28	93.20
9	8.57	11.08	125.88	115.56	113.55	102.11	115.09	107.32
10	18.92	21.75	133.16	122.27	120.74	110.24	123.01	114.80
11	21.93	24.85	135.52	124.50	123.08	112.82	125.53	117.21
12	32.83	37.86	152.98	142.23	140.58	129.63	142.53	134.55
13	38.06	44.00	161.59	151.67	149.37	137.00	150.18	143.02
14	14.24	20.19	137.79	128.01	125.60	113.18	126.36	119.22
15	8.92	14.92	132.28	122.73	120.15	107.44	120.56	113.7
16	11.42	16.41	131.45	122.42	119.49	106.08	119.36	93.20
17	0	6.02	123.58	113.90	111.41	98.95	112.12	112.90
18	6.02	0	117.60	107.88	105.42	93.08	106.22	99.03
19	123.58	117.60	0	13.66	12.47	26.89	14.77	18.58
20	113.90	107.88	13.66	0	7.01	24.17	18.21	12.76
21	111.41	105.42	12.47	7.01	0	17.78	11.46	6.98
22	98.95	93.08	26.89	24.17	17.78	0	13.31	11.45
23	112.12	106.22	14.77	18.21	11.46	13.31	0	10.96
24	112.90	99.03	18.58	12.76	6.98	11.45	10.96	0
25	106.48	100.52	17.33	13.83	7.17	10.63	8.20	2.78
26	198.94	193.06	90.91	92.62	98.37	116.09	105.57	105.14
27	230.49	224.55	116.28	120.59	125.54	142.76	131.01	132.49
28	223.28	217.27	103.54	110.26	114.19	130.44	117.83	121.13
29	76.57	70.81	49.73	43.91	39.00	23.17	36.46	32.02
30	57.17	51.74	71.26	64.78	60.31	44.68	57.93	53.36
31	132.27	136.09	32.42	32.06	25.32	8.53	17.69	19.62
32	292.36	290.93	301.57	289.97	293.91	305.05	307.12	297.19
33	301.46	299.66	302.32	291.26	287.45	307.62	308.79	299.20

Tabel 4.8: Data Jarak antar Permintaan dan Calon Kandidat Lokasi 4

I/J	25	26	27	28	29	30	31	32	33
1	126.62	200.83	234.09	231.56	103.63	89.85	191.49	251.11	261.55
2	133.32	210.49	243.65	240.58	108.64	93.29	183.76	258.84	269.48
3	129.08	210.28	243.17	239.12	102.94	86.45	171.67	267.65	278.32
4	123.55	206.87	239.56	234.93	96.78	79.89	164.61	271.92	282.23
5	117.99	197.07	230.03	226.30	93.42	78.60	176.89	261.03	270.94
6	109.59	190.49	223.27	219.0	84.78	70.07	105.94	264.17	273.52
7	100.55	184.17	216.65	211.63	75.12	60.29	139.32	269.15	277.89
8	94.96	183.58	215.52	209.23	67.42	50.82	122.02	279.64	288.05
9	109.03	198.39	230.37	224.05	80.35	62.07	140.83	284.40	293.72
10	116.60	202.47	234.89	229.55	89.09	71.70	156.83	276.97	286.81
11	119.05	203.96	236.49	231.41	91.82	74.63	159.13	275.06	285.07
12	136.29	222.13	254.68	249.55	108.0	89.55	181.04	281.77	292.84
13	144.53	234.62	266.74	260.48	114.50	94.54	187.28	297.59	308.96
14	120.71	212.20	243.99	237.17	90.74	71.06	151.54	293.77	303.75
15	115.14	207.82	239.4	232.18	84.87	64.99	143.46	295.66	305.27
16	114.20	208.95	240.18	232.31	83.21	62.69	148.53	301.90	311.38
17	106.48	198.94	230.49	223.28	76.57	57.17	132.27	292.36	301.46
18	100.52	193.06	224.55	217.27	70.81	51.74	136.09	290.93	299.66
19	17.33	90.91	116.28	103.54	49.73	71.26	32.42	301.57	302.32
20	13.83	92.62	120.59	110.26	43.91	64.78	32.06	289.97	291.26
21	7.17	98.37	125.54	114.19	39.00	60.31	25.32	293.91	287.45
22	10.63	116.09	142.76	130.44	23.17	44.68	8.53	305.05	307.62
23	8.20	105.57	131.01	117.83	36.46	57.93	17.69	307.12	308.79
24	2.78	105.14	132.49	121.13	32.02	53.36	19.62	297.19	299.20
25	0	105.4	132.36	120.52	32.54	54.03	18.24	299.82	301.77
26	105.47	0	33.59	41.08	136.37	156.61	122.91	270.59	265.20
27	132.36	33.59	0	24.50	164.42	185.36	148.69	291.78	284.47
28	120.52	41.08	24.50	0	153.06	174.48	135.37	310.11	303.83
29	32.54	136.37	164.42	153.06	0	21.55	21.47	304.50	308.63
30	54.03	156.61	185.36	174.48	21.55	0	42.14	307.39	312.91
31	18.24	122.91	148.69	135.37	21.47	42.14	0	312.87	315.65
32	299.82	270.59	291.78	310.11	304.50	307.39	312.87	0	20.86
33	301.77	265.20	284.47	303.83	308.63	312.91	315.65	20.86	0

SURAT PERNYATAAN KEASLIAN SKRIPSI

Dengan ini saya yang bertanda tangan di bawah ini, mahasiswa Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Jakarta:

Nama : Firdha Defita Sari

No. Registrasi : 3125111193

Program Studi : Matematika

Menyatakan bahwa skripsi ini yang saya buat dengan judul ” **Pengoptimalan Penempatan Fasilitas Kesehatan Menggunakan Algoritma Memetika (Studi Kasus : Fasilitas Kesehatan bagi Penderita HIV/AIDS di Sumatera Utara)**” adalah :

1. Dibuat dan diselesaikan oleh saya sendiri.
2. Bukan merupakan duplikat skripsi yang pernah dibuat oleh orang lain atau jiplakan karya tulis orang lain.

Pernyataan ini dibuat dengan sesungguhnya dan saya bersedia menanggung segala akibat yang timbul jika pernyataan saya tidak benar.

Jakarta, Agustus 2017

Yang membuat pernyataan

Firdha Defita Sari

DAFTAR RIWAYAT HIDUP



Firdha Defita Sari. Lahir di Jakarta, 7 Desember 1993. Anak kedua dari pasangan Bapak Safiudin dan Ibu Rita. Saat ini bertempat tinggal di Pinang, Kunciran, Tangerang.

Email : firdhadefta@yahoo.com

Riwayat Pendidikan : Penulis mengawali pendidikan di TK Aisyah 6 Jakarta Barat, kemudian melanjutkan pendidikan di SD Negeri Kemanggisan 03 Jakarta Barat 1999 - 2005. Setelah itu, penulis melanjutkan ke SMP Negeri 88 Jakarta dan dilanjutkan di SMA Negeri 35 Jakarta lulus tahun 2011. Di Tahun yang sama penulis melanjutkan ke Universitas Negeri Jakarta (UNJ), jurusan Matematika, melalui jalur tertulis SMPTN. Di pertengahan tahun 2017 penulis telah memperoleh gelar Sarjana Sains untuk Jurusan Matematika, Program Studi Matematika, FMIPA, UNJ.

Riwayat Organisasi : Selama di bangku perkuliahan, penulis aktif dalam organisasi kemahasiswaan yaitu sebagai staff Lembaga Legislatif Mahasiswa Jurusan Matematika UNJ dan Badan Perwakilan Mahasiswa FMIPA.

Riwayat Pekerjaan : Penulis mulai menjadi pengajar matematika sejak tahun 2013. Pada tahun 2011 sampai sekarang, penulis menjadi pengajar privat matematika. Penulis juga pernah bekerja sebagai *freelancer* di perusahaan yang bergerak dibidang riset.