



TESIS - TE142599

OPTIMASI PENEMPATAN BTS DAN MCP BERBASIS ARTIFICIAL INTELLIGENT DENGAN MEMANFAATKAN DATA SPATIAL GIS

PRASETYO YULIANTORO
07111650030011

DOSEN PEMBIMBING
Dr. Ir. Endroyono, DEA
Dr. Ir. Achmad Affandi, DEA.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018



TESIS - TE142599

OPTIMASI PENEMPATAN BTS DAN MCP BERBASIS ARTIFICIAL INTELLIGENT DENGAN MEMANFAATKAN DATA SPATIAL GIS

PRASETYO YULIANTORO
07111650030011

DOSEN PEMBIMBING
Dr. Ir. Endroyono, DEA
Dr. Ir. Achmad Affandi, DEA.

PROGRAM MAGISTER
BIDANG KEAHLIAN TELEKOMUNIKASI MULTIMEDIA
DEPARTEMEN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI ELEKTRO
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2018

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (M.T)
di
Institut Teknologi Sepuluh Nopember

oleh:

Prasetyo Yuliantoro
NRP. 07111650030011

Tanggal Ujian : 05 Juli 2018
Periode Wisuda : September 2018

Disetujui oleh:

1. Dr. Ir. Endroyono, DEA
NIP: 196504041991021001

(Pembimbing I)

2. Dr. Ir. Achmad Affandi, DEA
NIP: 196510141990021001

(Pembimbing II)

3. Dr. Istas Pratomo, ST., MT.
NIP: 197903252003121001

(Penguji)

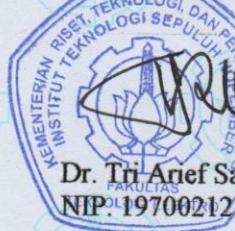
4. Dr. Ir. Achmad Maulidiyanto, MT.
NIP: 196109031989031001

(Penguji)

5. Eko Setijadi, ST., MT., Ph.D.
NIP: 197210012003121002

(Penguji)

Dekan Fakultas Teknologi Elektro



Dr. Tri-Arief Sardjono, S.T., M.T.
NIP. 197002121995121001

Halaman ini sengaja dikosongkan

PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul "**OPTIMASI PENEMPATAN BTS DAN MCP BERBASIS ARTIFICIAL INTELLIGENT DENGAN MEMANFAATKAN DATA SPATIAL GIS**" adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2018

Prasetyo Yuliantoro

NRP. 07111650030011

Halaman ini sengaja dikosongkan

OPTIMASI PENEMPATAN BTS DAN MCP BERBASIS ARTIFICIAL INTELLIGENT DENGAN MEMANFAATKAN DATA SPASIAL GIS

Nama mahasiswa : Prasetyo Yuliantoro
NRP : 07111650030011
Pembimbing : 1. Dr. Ir. Endroyono, DEA
 2. Dr. Ir. Achmad Affandi, DEA

ABSTRAK

Operator seluler berlomba untuk membangun layanan selular. Mereka mengajukan pembangunan menara baru kepada pemerintah. Pemerintah menggunakan proses manual untuk melihat usulan tersebut. Hal ini membutuhkan waktu tambahan karena pengecekan dilakukan satu persatu parameter dan hal ini mempunyai resiko terkait tingkat ketelitiannya. Pada penelitian ini, dibuat optimasi spasial menggunakan algoritma greedy remove dan algoritma genetika dengan data spasial dari pemerintah. Data spasial berupa data menara, data dasar Kota Surabaya dan data bangunan seperti makam, militer, industri dan fasilitas umum.

Penelitian ini bertujuan untuk optimalisasi penempatan menara BTS dan MCP agar menghasilkan layanan secara optimal yang memaksimalkan kapasitas dan coverage layanan komunikasi, meminimalkan jumlah menara, serta meminimalkan keberadaan menara di lokasi terlarang dengan memperhatikan keamanan dan estetika kota sesuai aturan tata-ruang. Perancangan algoritma menggunakan bahasa pemrograman PHP dengan bantuan software QGIS sebagai media data spasial.

Pengujian awal algoritma yaitu pengujian algoritma konvesional dengan memasukkan parameter perulangan sebanyak 16 kali dan jarak 200 meter. Hasil dari algoritma pertama yaitu grid optimal terletak pada perulangan ke-14 dengan waktu pencarian selama 6.25 menit. Hasil dari algoritma kedua yaitu grid optimal terletak pada perulangan ke-4 dengan waktu pencarian selama 62.5 menit. Pengujian algoritma konvensional dibagi menjadi 3 skenario. Hasil skenario pertama yaitu penempatan menara pada lokasi minim menara, hasil skenario ini yaitu menara boleh ditempatkan pada lokasi tersebut. Pada skenario kedua dan ketiga yaitu penempatan menara pada lokasi yang terdapat bangunan dan jumlah menara diatas 4. Hasil dari skenario kedua dan ketiga yaitu usulan pendirian menara ditolak dan diberikan rekomendasi untuk mendirikan menara pada sisi yang lain. Pengujian algoritma cerdas dilakukan dengan iterasi sebanyak 10000 kali. Percobaan dilakukan sebanyak 30 kali menghasilkan nilai kehandalan sebesar 96,6%. Sehingga algoritma cerdas yang disusun dinyatakan layak untuk menyelesaikan masalah terkait penempatan menara bts baru.

Kata kunci: Optimasi Menara, QGIS, Data Spasial, Algoritma Genetika, PHP.

Halaman ini sengaja dikosongkan

OPTIMIZATION OF BTS AND MCP PLACEMENT BASED ON ARTIFICIAL INTELLIGENT BY USING SPATIAL DATA GIS

By : Prasetyo Yuliantoro
Student Identity Number : 07111650030011
Supervisor(s) : 1. Dr. Ir. Endroyono, DEA
 2. Dr. Ir. Achmad Affandi, DEA

ABSTRACT

Celullar operators compete to build cellular services. They proposed the construction of new towers to the government. The government uses a manual process to see the proposal. This requires additional time because the checks are done one by one parameter and this has a risk associated with the level of accuracy. In this research, spatial optimization is made using greedy remove algorithm with spatial data from government. Spatial data are tower data, basic data of Surabaya City and building data such as grave, military, industry and public facilities.

The objective of this research is to optimize the placement of BTS and MCP towers in order to provide optimal services that maximize the capacity and coverage of communication services, minimize the number of towers, and minimize the presence of towers in illegal locations with respect to the security and aesthetics of the city according to the spatial rules. The design of algorithms using PHP programming language with the help of QGIS software as spatial data media.

Preliminary test algorithm that is by entering the recurrence parameter as much as 16 times and distance of 200 meter. The result of the first algorithm is the optimal grid lies in the 14th iteration with a total search time of 6.25 minutes. The result of the second algorithm is optimal grid lies in the 4th with total search time for 62.5 minutes. Testing algorithm is divided into 3 scenarios. The first scenario result is the placement of minarets at the location of minimal towers, the result of this scenario is the tower may be placed on the location. In the second and third scenarios the placement of the tower at the location of the building and the number of towers above 4. The result of the second scenario of the proposed tower establishment was rejected and given a recommendation to establish the tower on the other side. The intelligent algorithm test is done by iterating as much as 10000 times. The experiment was conducted 30 times yielding reliability value of 96.6%. So that the intelligent algorithm is arranged declared eligible to solve problems related to placement of new bts tower.

Key words: Tower Optimization, QGIS, Spatial Data, Genetic Algorithm, PHP.

Halaman ini sengaja dikosongkan

KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah memberikan rahmat, petunjuk, pengetahuan, serta karunia-Nya sehingga penelitian ini dapat terselesaikan. Penelitian ini disusun untuk memenuhi syarat menyelesaikan pendidikan S2 pada Bidang Studi Telekomunikasi Multimedia, Jurusan Teknik Elektro, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember Surabaya. Judul penelitian ini adalah :

“OPTIMASI PENEMPATAN BTS DAN MCP BERBASIS ARTIFICIAL INTELLIGENT DENGAN MEMANFAATKAN DATA SPATIAL GIS”

Terselesaikannya tugas akhir ini tentunya tak lepas dari dorongan dan uluran tangan berbagai pihak. Oleh karena itu, tak salah kiranya bila penulis mengungkapkan rasa terima kasih kepada beberapa pihak yang memberikan dukungan selama proses penyelesaian tugas akhir ini, antara lain :

1. Kedua orang tua penulis tersayang, Ibu Nuraeni Dwi Astuti dan Bapak Wakidi yang telah mensupport banyak hal, serta Reza Edwien Sulistyaningtyas yang telah memberi saran, menghibur dan mendoakan.
2. Bapak Dr. Ir. Endroyono, DEA serta Bapak Dr. Ir. Achmad Affandi, DEA selaku Dosen Pembimbing yang telah banyak memberikan ilmu, pengarahan dan bimbingan selama penyelesaian tugas akhir ini.
3. Teman – teman S2 Telkom maupun Lab A205 yang selalu memberikan support dan dukungannya.

Semoga tugas akhir ini dapat bermanfaat dan berguna bagi pembacanya. Penulis menyadari dalam penyusunan tugas akhir ini masih jauh dari sempurna sehingga saran, kritik dan diskusi untuk pengembangan dari tugas akhir ini sangat penulis harapkan.

Surabaya, 12 Juni 2016

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

LEMBAR PENGESAHAN	iii
PERNYATAAN KEASLIAN TESIS	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	3
1.4 Batasan Masalah	3
1.5 Kontribusi	3
1.6 Metodologi Penelitian.....	4
BAB 2 KAJIAN PUSTAKA.....	7
2.1 Kajian Penelitian Terkait, tambah terkait AI yang digunakan untuk pengambilan keputusan.....	7
2.2 Teori Dasar.....	8
2.2.1 Peraturan Pemerintah	8
2.2.2 Sistem Informasi Geografis.....	9
2.2.3 <i>Platform Open Source GIS</i>	14
2.2.4 Algoritma <i>Greedy</i>	15
2.2.5 Algoritma <i>Greedy Remove</i>	16
2.2.6 Algoritma Genetika.....	17
2.2.7 XAMPP	24
BAB 3 METODOLOGI PENELITIAN.....	27
3.1 Langkah-langkah Penelitian.....	27
3.2 Data Spatial.....	28
3.2.1 Peta Dasar Kota Surabaya.....	28
3.2.2 Data Bangunan	29

3.2.3	Koordinat Tower (BTS dan MCP)	31
3.3	Metode Pembuatan Grid berdasarkan Posisi Menara.....	32
3.4	Metode Pencarian Grid Optimal berdasarkan Posisi Menara	34
3.5	Metode Pencarian Grid Optimal berdasarkan Bobot	37
3.6	Metode Pencarian Lokasi Rekomendasi	39
3.7	Spesifikasi Kawasan Pendirian Menara	40
3.8	Metode Pencarian Kawasan Pendirian Menara Menggunakan Algoritma Genetika.....	41
3.8.1	Setting Parameter GA.....	42
3.8.2	Menentukan Populasi Awal.....	42
3.8.3	Mengurutkan Individu Berdasarkan Nilai <i>Fitness</i>	42
3.8.4	Kriteria Pemberhentian.....	44
3.8.5	Membuat Generasi Baru Menggunakan Elitisme.....	44
3.8.6	Seleksi Menggunakan Metode <i>Turney</i>	44
3.8.7	Proses <i>Crossover</i>	45
3.8.8	Proses Mutasi.....	46
3.9	Skenario Pengujian Algoritma	47
BAB 4 HASIL DAN PEMBAHASAN	49	
4.1	Hasil Pencarian Grid Optimal berdasarkan Posisi Menara	49
4.2	Konversi Data Spasial Bangunan Menjadi Koordinat	50
4.3	Hasil Pencarian Grid Optimal berdasarkan Bobot	53
4.4	Hasil Skenario Pengujian	54
4.5	Hasil Pengujian Algoritma Cerdas	56
4.6	Visualisasi Hasil Optimasi	57
BAB 5 PENUTUP	59	
5.1	Kesimpulan.....	59
5.2	Saran.....	59
DAFTAR PUSTAKA	61	
LAMPIRAN	63	
DAFTAR RIWAYAT HIDUP	93	

DAFTAR GAMBAR

Gambar 2.1 Model data dalam GIS [8]	10
Gambar 2.2 Model data dalam GIS [8]	11
Gambar 2.3 Peta UTM Dunia	12
Gambar 2.4 Struktur umum algoritma genetika.....	21
Gambar 3.1 Diagram Alir Langkah Penelitian	27
Gambar 3.2 Peta Dasar Kota Surabaya	28
Gambar 3.3 Bangunan Militer.....	29
Gambar 3.4 Pemakaman	30
Gambar 3.5 Kawasan perindustrian	30
Gambar 3.6 Bangunan Fasilitas Umum	31
Gambar 3.7 File excel data koordinat menara	31
Gambar 3.8 Layer Menara	32
Gambar 3.9 <i>Tools create grid</i> dari software QGIS	33
Gambar 3.10 Hasil generate grid	34
Gambar 3.11 Mencari jumlah menara yang berada didalam grid	35
Gambar 3.12 Proses perhitungan penentuan grid optimal	37
Gambar 3.13 Perhitungan bobot untuk setiap grid.....	39
Gambar 3.14 Gambaran grid rekomendasi	40
Gambar 3.15 Rancangan Algoritma Genetika	41
Gambar 3.16 Populasi, Individu, <i>Kromosom, Gen</i> dan <i>Allele</i>	42
Gambar 3.17 Bobot individu yang dicari.....	43
Gambar 3.18 Individu yang diambil pada proses <i>elitisme</i>	44
Gambar 3.19 Proses seleksi menggunakan metode <i>turney</i>	45
Gambar 3.20 <i>Gen x</i> dan <i>y</i> pada masing-masing individu.....	45
Gambar 3.21 Proses <i>crossover gen y₁</i> dan <i>y₂</i>	46
Gambar 3.22 Proses mutasi.....	46
Gambar 4.1 Koordinat <i>point</i> militer.....	51
Gambar 4.2 Koordinat <i>point</i> makam.....	52
Gambar 4.3 Koordinat <i>point</i> industri	52
Gambar 4.4 Koordinat <i>point</i> fasilitas umum.....	53
Gambar 4.5 Hasil generate grid untuk <i>visualisasi</i>	57

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 2.1 Deskripsi Penelitian Sebelumnya.....	7
Tabel 2.2 Zona UTM untuk Wilayah Indonesia [10].....	13
Tabel 2.3 Terminologi Algoritma Genetika.....	18
Tabel 3.1 Pemberian nilai bobot	38
Tabel 3.2 Spesifikasi kawasan yang dapat didirikan menara baru.....	40
Tabel 3.3 Parameter GA.....	42
Tabel 3.4 Tabel pembobotan kawasan	43
Tabel 3.5 Persentase Kelayakan [12]	48
Tabel 4.1 Hasil algoritma awal	49
Tabel 4.2 Hasil algoritma yang ditambahkan parameter bobot	53
Tabel 4.3 Rekomendasi koordinat skenario kedua	55
Tabel 4.4 Rekomendasi koordinat skenario ketiga	55
Tabel 4.5 Hasil algoritma GA	56

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Di era informasi, teknologi informasi dan komunikasi (TIK/ICT) menjadi kunci kemajuan. Telekomunikasi dan transmisi data mempunyai peranan yang sangat penting dalam mendukung pertumbuhan ekonomi, dibandingkan pertumbuhan ekonomi pada sektor yang lain. Para pelaku bisnis kemudian berlomba untuk menyediakan layanan komunikasi, baik melalui pengembangan jaringan kabel maupun jaringan nirkabel. Data terakhir menunjukkan, bahwa jaringan nirkabel lebih disukasi oleh pengguna dan memberikan keuntungan lebih besar dibanding investasi menggunakan jaringan kabel, termasuk jaringan serat-optik.

Penyediaan jaringan nirkabel, termasuk melalui jaringan seluler, jelas akan berakibat munculnya kebutuhan akan adanya jaringan akses nirkabel dan BTS (*Base Transceiver Station*). Jumlah BTS yang harus dibangun sangat tergantung kepada 2 (dua) hal utama, yaitu *coverage* (wilayah layanan) dari BTS dan *traffic* (atau kapasitas) yang harus dilayani di setiap wilayah layanan. Setiap operator pasti telah mempunyai team yang memahami hal ini.

Dari sudut pandang operator dan masyarakat, pembangunan BTS jelas bertujuan untuk meningkatkan wilayah dan kapasitas layanan, akan tetapi terbukti perijinan untuk membangun BTS di setiap kota dapat dikatakan cukup sulit untuk diperoleh. Hal ini diakibatkan oleh pengalaman masa-lalu, bahwa pembangunan BTS yang tidak terkendali berkibat pada penurunan kualitas tata-ruang, estetika (keindahan kota), keamanan dan aspek sosial non-teknis lainnya. Untuk itulah hampir di setiap kota menerbitkan perda [1] yang merupakan turunan dari peraturan menteri terkait [2] yang berhubungan dengan aturan perijinan pendirian menara BTS. Dan dengan semakin kompleksnya permasalahan, maka semakin dibutuhkan alat-bantu dalam penempatan menara BTS, karena semakin banyak aspek dan faktor yang harus dipertimbangkan oleh pembuat keputusan.

Pada akhirnya untuk menempatkan sebuah menara di lokasi tertentu, fokus utama jelas pada usulan provider telekomunikasi yang biasanya didasarkan pada pemenuhan wilayah layanan dan kapasitas *traffic*. Akan tetapi kota mempunyai kepentingan untuk mensinkronkan hal tersebut dengan penataan tata-ruang kota (RT/RW) [3], aspek pekerjaan umum, keselamatan, keamanan dan keindahan. Walaupun mempunyai potensi pelanggan, pembangunan sebuah menara dapat ditolak apabila di area tersebut berada di kawasan keselamatan penerbangan, kawasan strategis, kawasan sempadan terlarang dan seterusnya.

Untuk sebuah kota besar, dengan luasan yang besar dan penduduk jutaan penempatan BTS pada zona-zona yang benar secara manual jelas akan membutuhkan waktu yang lama dan dapat menimbulkan kesalahan. Untuk itu banyak teknik pemanfaatan berbasis data telah mulai digunakan, penempatan secara konvensional biasanya dilakukan menggunakan konsep pencarian (*search*), seperti menggunakan algoritma tabu and search, menggunakan teknik *greedy* [4], *greedy remove* dan sebagainya. Pada [5] diusulkan teknik optimasi penempatan BTS berdasarkan *coverage* area. Hal ini kurang optimal mengingat beberapa faktor seperti tata kota dan estetika kota tidak diperhitungkan. Oleh karena itu, pada penelitian ini penulis membuat algoritma berbasis algoritma cerdas yang dapat digunakan untuk menentukan apakah di sebuah tempat bisa dipasang menara baru, direkomendasikan menggunakan menara *existing* atau ditolak.

1.2 Rumusan Masalah

Permasalahan utama dalam penempatan menara sebenarnya bukanlah hanya terbatas boleh atau tidak, tetapi lebih kepada memberikan solusi optimal penempatan menara di sebuah kota dengan target memaksimalkan kapasitas, meminimalkan jumlah menara, serta meminimalkan keberadaan menara di lokasi terlarang. Hal tersebut dapat dibagi menjadi tiga permasalahan utama, yaitu:

- a. Bagaimana mengoptimalkan lokasi dan jumlah menara di sebuah kota.
- b. Bagaimana memberikan rekomendasi lokasi penempatan menara berdasarkan algoritma genetika.

- c. Bagaimana menyediakan sistem visualisasi spasial, menggunakan aplikasi berbasis GIS untuk memudahkan pengambil keputusan memberi rekomendasi kepada pemohon ijin.

1.3 Tujuan

Mengacu pada permasalahan diatas, tujuan penelitian ini adalah

- a. Membuat sebuah algoritma baru yang dapat mengolah data-data kondisi spasial sebuah kota dalam rangka menghasilkan layanan secara optimal yang memaksimalkan kapasitas dan *coverage* layanan komunikasi, meminimalkan jumlah menara, serta meminimalkan keberadaan menara di lokasi terlarang dengan memperhatikan keamanan dan estetika kota sesuai aturan tata-ruang.
- b. Mengembangkan algoritma cerdas yang dapat memberikan rekomendasi penempatan menara berdasarkan data spasial yang telah diolah.
- c. Mengembangkan sistem visualisasi spasial, menggunakan aplikasi berbasis GIS, untuk media visualisasi posisi menara dan data spasial kota.

1.4 Batasan Masalah

- 1. Optimasi dilakukan pada kota besar (urban) dengan mengambil contoh kota Surabaya.
- 2. Algoritma cerdas yang digunakan untuk rekomendasi penempatan menara yaitu algoritma genetika.
- 3. Software GIS yang digunakan yaitu Quantum GIS yang merupakan software *open-source*.

1.5 Kontribusi

Penelitian ini diharapkan dapat memberikan kontribusi berupa pemanfaatan algoritma cerdas yang digunakan untuk optimalisasi penempatan menara BTS dan divisualkan pada sistem informasi geografis (GIS) untuk Kota Surabaya.

1.6 Metodologi Penelitian

Tahap-tahap yang dilakukan untuk menyelesaikan penelitian ini adalah sebagai berikut:

- 1. Studi Literatur**

Mempelajari teori-teori yang berkaitan dengan masalah optimasi, *greedy*, *greedy remove*, *artificial intelligent*, *genetic algorithm* (algoritma genetika) dan Bahan literatur berupa buku teks, materi dari internet, paper referensi dan literatur lainnya.

- 2. Pengambilan Data**

Data spasial GIS Kota Surabaya didapat dari Pemerintah Kota.

- 3. Perancangan Algoritma**

Algoritma awal merupakan algoritma konvensional yang mengacu kepada sistem *greedy remove*. Data menara dari pemerintah dijadikan sebagai dasar pembuatan grid dan akan diberikan bobot sesuai bangunan diatasnya. Algoritma konvensional ini digunakan untuk menentukan grid yang memiliki bobot paling optimal. Selanjutnya merancang algoritma cerdas berbasis algoritma genetika yang digunakan untuk memberikan rekomendasi terkait penempatan menara.

- 4. Pengujian Algoritma**

Pengujian dilakukan dengan cara memasukkan koordinat tower dari pemerintah. Output dari algoritma ini yaitu menemukan grid optimal dengan banyak iterasi yang telah ditentukan. Grid optimal diuji dengan cara memasukkan koordinat tower yang diibaratkan berupa usulan dari operator. Hasil dari langkah ini yaitu keputusan berupa boleh tidak nya menara dibangun di tempat usulan, dan jika menara tidak boleh dibangun di tempat tersebut, maka algoritma akan mencari lokasi penempatan menara. Pengujian pada sisi algoritma cerdas yaitu dengan memasukkan beberapa parameter seperti menara, makam, militer, fasum dan industri. Hasil dari algoritma ini yaitu memberikan rekomendasi penempatan menara dengan jumlah menara kurang dari tiga dan lokasi berada pada kawasan fasilitas umum. Output ini diharapkan dapat membantu pemerintah dalam hal pengambilan keputusan terkait pembangunan menara baru.

5. Kesimpulan

Hasil dari pengujian algoritma konvensional dan algoritma cerdas dianalisa dan digunakan untuk membuat kesimpulan dari penelitian yang telah dilakukan.

Halaman ini sengaja dikosongkan

BAB 2

KAJIAN PUSTAKA

2.1 Kajian Penelitian Terkait tambah terkait AI yang digunakan untuk pengambilan keputusan

Beberapa penelitian yang mendukung optimasi penempatan BTS telah dilakukan, diantaranya [4], [5] dan [6]. Pada penelitian [4] dijelaskan bahwa beberapa algoritma optimasi telah digunakan, diantaranya yaitu *Random Greedy Add*, *Random Greedy Remove*, *Multistart Tabu Search using RGA*, *Multistart Tabu Search using RGR*, dan *Branch and bound Algorithm*. Dari kelima algoritma tersebut, algoritma yang paling *reliable* untuk digunakan yaitu algoritma *branch and bound* (BnB).

Berikut tabel 2.1 yang berisikan penjelasan singkat mengenai kelebihan masing-masing penelitian

Tabel 2.1 Deskripsi Penelitian Sebelumnya

No	Peneliti	Judul	Deskripsi
[4]	G. Kalata (2014)	<i>Optimising Base Station Location For UMTS Branch and Bound Cellular Network</i>	Penggunaan algoritma
[5]	Haitham K.Ali (2015)	<i>Geographic Information System (GIS) Spatial Analyst Techniques a Reference for Determining the Position of Cellular System</i>	Mengatur ulang penempatan menara BTS, mengurangi jumlah menara BTS dan mengurangi wilayah dengan cakupan sinyal yang lemah.
[6]	C O Alenogheha (2013)	<i>Artificial Intelligent Based Technique for BTS Placement</i>	Mengoptimalkan penempatan menara BTS dengan AI dan dengan panduan dari regulasi pemerintah setempat.

Pada penelitian [6] dijelaskan bahwa AI dapat digunakan untuk penempatan BTS. Penelitian tersebut menggunakan algoritma GA (*genetic algorithm*) dengan skenario lingkungan sub-urban. Pada penelitian [5] dijelaskan bahwa GIS dapat digunakan untuk menentukan posisi BTS. Penelitian tersebut dilakukan di Iraq, dengan menara BTS awal sejumlah 22. Setelah dilakukan optimasi dengan cara menaikkan radius pancaran dan dianalisa terkait area yang belum tercover, maka didapat hasil yaitu optimum menara BTS menjadi 19 unit.

2.2 Teori Dasar

2.2.1 Peraturan Pemerintah

Pada peraturan pemerintah yang menjadi acuan untuk topik penelitian yaitu Peraturan Pemerintah Daerah Kota Surabaya Nomor 5 Tahun 2013 tentang Penyelenggaraan Menara Telekomunikasi Bersama. Pada PP daerah tersebut dibahas bahwa untuk menjamin keamanan, kenyamanan dan keselamatan masyarakat, perlu dilakukan penataan terhadap penyelenggaraan menara telekomunikasi oleh Pemerintah Daerah. Penataan yang dibahas dalam UU ini diantaranya mengenai zona penempatan lokasi menara, maksimal penempatan menara, dan juga mengenai estetika kawasan daerah dan memperhatikan kelestarian lingkungan.

Zona penempatan lokasi menara sesuai pasal 1 ayat 22 adalah zona penempatan titik-titik lokasi menara yang telah ditentukan untuk pembangunan menara telekomunikasi bersama. Zona ini berada dalam radius maksimum 200 (dua ratus) meter dari titik koordinat yang telah ditentukan. Pembangunan menara telekomunikasi memperhatikan aspek-aspek kaidah perencanaan jaringan selular yaitu ketersediaan *coverage* area pada area potensi *generated traffic* dan ketersediaan kapasitas *traffic* telekomunikasi selular [1].

Pembahasan mengenai zona maksimal penempatan menara telekomunikasi dibahas pada pasal 5 ayat (3) yang isinya yaitu setiap zona penempatan lokasi menara telekomunikasi sebagaimana dimaksud pada ayat (1) paling banyak ditempatkan 4 (empat) bangunan menara telekomunikasi [1]. Pada peraturan tersebut, suatu zona hanya boleh didirikan maksimal 4 menara didalamnya, menara dalam konteks penelitian ini yaitu bangunan menara

telekomunikasi. Pada pasal 5 ayat 2 dijelaskan bahwa menara atau bangunan menara telekomunikasi harus mampu menopang perangkat telekomunikasi yang dimiliki oleh paling sedikit 3(tiga) penyelenggara telekomunikasi selular [1]. Sehingga jika menara dipasang secara maksimal, 4 bangunan menara dapat menopang sebanyak 12 perangkat telekomunikasi yang dimiliki oleh penyelenggara telekomunikasi seluler. Ketinggian yang dibahas pada peraturan ini yaitu jika ketinggian menara lebih dari 6 meter, maka menara didirikan diatas bangunan (*roff top*).

Sesuai dengan perundungan yang berlaku, MCP (Microcell pole) adalah bangunan menara tunggal dengan ketinggian maksimal 18 (delapan belas) meter dari permukaan tanah yang digunakan untuk menempatkan peralatan komunikasi yang diletakkan pada BTS Hotel/BTS Room/Data Center yang terhubung dengan kabel serat optik. Spesifikasi menara tunggal microcell terdapat pada lampiran.

Beberapa poin diatas dijadikan sebagai bahan pertimbangan atau sebagai parameter yang mempengaruhi pengambilan keputusan penempatan menara BTS. Parameter-parameter tersebut dijadikan sebagai pembobot dan diterapkan pada software QGIS untuk dijadikan sebagai data spasial yang nantinya digunakan pada algoritma cerdas.

2.2.2 Sistem Informasi Geografis

Sistem Informasi Geografis (*Geographic Information Systems - GIS*) pada umumnya adalah sistem informasi khusus yang mengelola data yang memiliki informasi spasial. SIG juga merupakan sejenis perangkat lunak yang dapat digunakan untuk pemasukan, penyimpanan, manipulasi, menampilkan, dan keluaran informasi geografis berikut atribut – atributnya [7].

SIG digunakan untuk pengolahan data dan melakukan pengaturan serta dapat memperlihatkan data yang tepat, dengan kata lain SIG dapat digunakan untuk analisis data dan menghasilkan data baru yang berguna, pada gilirannya SIG dapat membantu untuk pengambilan keputusan.

SIG dibagi menjadi dua kelompok yaitu sistem manual (*analog*) dan sistem otomatis (yang berbasis digital komputer). Perbedaan yang paling dasar yaitu terletak pada cara pengelolannya. Pada sistem manual, beberapa komponen peta seperti peta, foto udara, laporan statistik dan survei lapangan digabungkan

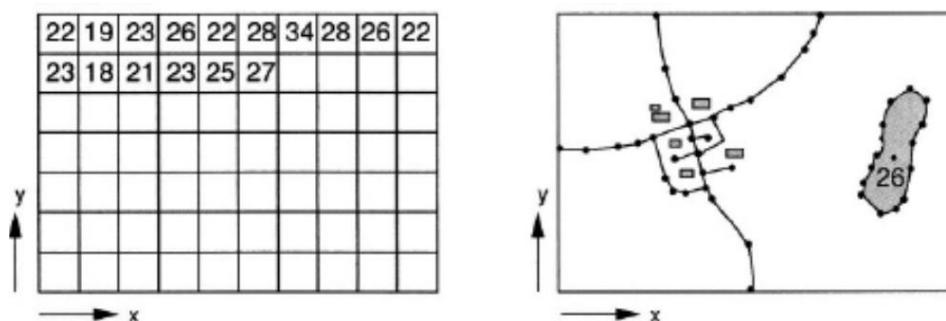
secara *overlay* atau tumpang tindih. Sedangkan SIG otomatis menggunakan komputer untuk sistem pengolah data melalui proses digitasi. Sumber data digital dapat berupa citra satelit atau foto udara digital serta foto udara yang terdigitalisasi. Data lain dapat berupa peta dasar terdigitalisasi.

2.2.2.1 Jenis data dalam SIG

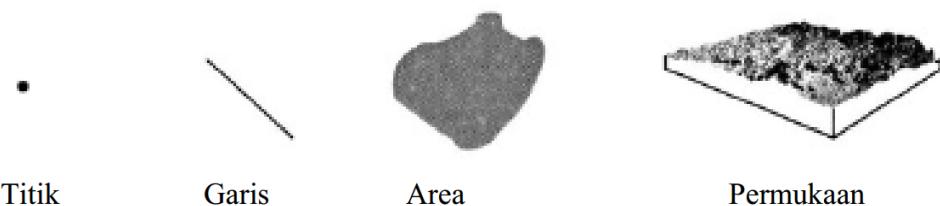
Secara umum, dalam SIG ada dua jenis data yang diolah, yaitu data spasial dan data atribut. Model data spasial memerlukan metoda—metoda tertentu untuk efektifitas dan efisiensi dalam pengolahannya. Data spasial sendiri merupakan data keruangan yang berorientasi geografis dan memiliki sistem koordinat tertentu sebagai dasar referensinya. Ada dua model data yang umum digunakan dalam data spasial, yaitu mode data raster dan model data vektor (Gambar 2.1).

Model data raster menggambarkan ruang dua dimensi dalam bentuk *matriks* yang terdiri atas grid sel (pixel) segi empat yang teratur dalam baris dan kolom. Tiap pixel menggambarkan bagian-bagian permukaan bumi, dimana jenis dan nilai cirinya berbentuk segi empat yang diberi label dan direkam. Resolusi data raster ditentukan oleh ukuran *pixel*-nya.

Model data vektor adalah suatu harga ruang dua dimensi yang diwakili oleh suatu harga *kontinyu* dan teliti yang merupakan tampilan dari suatu posisi *feature* geografi. Dalam model data vektor, daerah pada data diasumsikan sebagai ruang koordinat kontinu dimana posisi-posisi obyek yang dapat ditentukan sesuai dengan kenampakan aslinya.



Gambar 2.1 Model data dalam GIS [8]



Gambar 2.2 Model data dalam GIS [8]

Secara umum, bumi dan semua fitur berada dalam ruang dan waktu 3 dimensi, akan tetapi dalam aplikasi SIG data yang dipakai adalah hasil proyeksi benda 3 dimensi ke dalam bidang datar 2 dimensi. *Elevasi*, sebagai dimensi ketiga biasanya disimpan dalam layer terpisah yang merepresentasikan permukaan dalam ruang 3 dimensi. Dimensi data spasial dapat berupa titik, garis, area, dan permukaan seperti pada gambar 2.2 yang dapat dijelaskan sebagai berikut

- Titik/*point* : berbentuk titik dengan dimensi nol dengan koordinat x,y,z.
- Gars/*line* : merupakan kumpulan koordinat berbasis satu yang mempunyai titik awal dan titik akhir
- Area/*polygon* : merupakan kumpulan koordinat berdimensi dua, yang membentuk area dengan titik awal dan titik akhir yang bertemu pada satu titik
- Permukaan/*surface* : merupakan area dengan koordinat x,y,z berdimensi 3.

2.2.2.2 Sumber data dalam SIG

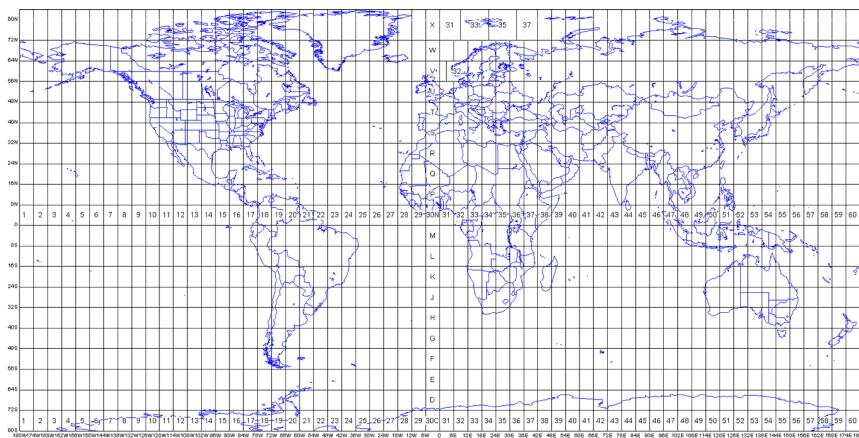
Sumber data dalam SIG bisa berasal dari pengamatan langsung (teristris), penginderaan jauh, maupun data dari penelitian terdahulu berupa peta, tabel, maupun data statistik. Berikut ini penjelasan masing-masing sumber data:

1. Survei lapangan : pengambilan data dengan cara langsung kelapangan dan melakukan pengukuran fisik (*land marks*), pengambilan sampel, dan pengumpulan data-data non fisik.
2. Sensus : pengumpulan data dengan pendekatan kuisioner, wawancara dan pengamatan; pengumpulan data secara nasional dan periodik (sensus jumlah penduduk, sensus kepemilikan tanah).

3. Statistik : merupakan cara pengumpulan data periodik/ per interval waktu pada stasiun pengamatan dan analisis data geografi tersebut.
4. *Tracking*: pengambilan data dalam periode tertentu dengan tujuan pemantauan atau pengamatan perubahan, contoh: kebakaran hutan, gunung meletus, debit air sungai.
5. Penginderaan jarak jauh: merupakan ilmu dan seni untuk mendapatkan informasi suatu obyek, wilayah atau fenomena melalui analisis data yang diperoleh dari sensor pengamat tanpa harus kontak langsung dengan obyek, wilayah atau fenomena yang diamati.

2.2.2.3 Sistem Proyeksi GIS [9]

Proyeksi peta adalah teknik-teknik yang digunakan untuk menggambarkan sebagian atau keseluruhan permukaan tiga dimensi ke permukaan dua dimensi. Dalam sistem proyeksi peta, proyeksi diartikan sebagai metode atau cara dalam usaha mendapatkan bentuk ubahan dari dimensi tertentu menjadi bentuk dimensi yang sistematik. Sistem UTM (*Universal Transvers Mercator*) dengan system koordinat WGS 84 sering digunakan pada pemetaan wilayah Indonesia. UTM menggunakan silinder yang membungkus *ellipsoid* dengan kedudukan sumbu silindernya tegak lurus sumbu tegak *ellipsoid* (sumbu perputaran bumi) sehingga garis singgung *ellipsoid* dan silinder merupakan garis yang berhimpit dengan garis bujur pada *ellipsoid*. Gambar 2.3 dibawah ini merupakan proyeksi peta UTM Dunia.



Gambar 2.3 Peta UTM Dunia

Sistem koordinat yang saat ini digunakan di Indonesia yaitu sistem bujur-lintang dan sistem koordinat UTM. Tidak semua sistem koordinat cocok untuk dipakai di semua wilayah. Sistem koordinat bujur-lintang tidak cocok digunakan di tempat-tempat yang berdekatan dengan kutub sebab garis bujur akan menjadi terlalu pendek. Tetapi, kedua sistem koordinat tersebut cocok digunakan di Indonesia.

Sistem koordinat bujur-lintang (atau dalam bahasa Inggris disebut Latitude-Longitude), terdiri dari dua komponen yang menentukan, yaitu :

- Garis dari atas ke bawah (*vertikal*) yang menghubungkan kutub utara dengan kutub selatan bumi, disebut juga garis lintang (*latitude*)
- Garis mendatar (*horizontal*) yang sejajar dengan garis khatulistiwa, disebut juga garis bujur (*longitude*)

2.2.2.4 Sistem Koordinat UTM (*Universal Transverse Mercator*)

Seluruh wilayah yang ada di permukaan bumi dibagi menjadi 60 zona bujur. Zona 1 dimulai dari lautan teduh (pertemuan antara garis 180 Bujur Barat dan 180 Bujur Timur), menuju ke timur dan berakhir di tempat berawalnya zona 1. Masing-masing zona bujur memiliki lebar 6 derajat atau sekitar 667 kilometer. Berikut ini merupakan tabel zona proyeksi UTM untuk wilayah Indonesia:

Tabel 2.2 Zona UTM untuk Wilayah Indonesia [10]

Universal Transverse Mercator			
No Zona	Mer. Sentral	Mer. Batas	
		Barat	Timur
46	93°	90°	96°
47	99°	96°	102°
48	105°	102°	108°
49	111°	108°	114°
50	117°	114°	120°
51	123°	120°	126°
52	129°	126°	132°
53	135°	132°	138°
54	141°	138°	144°

Garis lintang UTM dibagi menjadi 20 zona lintang dengan panjang masing-masing zona adalah 8 (derajat) atau sekitar 890 km. Zona lintang dimulai dari 80 LS - 72 LS diberi nama zona C dan berakhir pada zona X yang terletak pada koordinat 72 LU - 84 LU. Huruf (I) dan (O) tidak dipergunakan dalam penamaan zona lintang. Dengan demikian penamaan setiap zona UTM adalah koordinasi antara kode angka (garis bujur) dan kode huruf (garis lintang).

2.2.3 Platform Open Source GIS

Terdapat beberapa software pengolah data sistem informasi geografis yang berbayar maupun yang gratis, diantaranya yaitu ArcView, ArcGis, Quantum GIS, GRASS, ILWIS, RAT dan NEST. Dalam penelitian ini, peneliti menggunakan software yang bersifat gratis atau *open source* yaitu Quantum GIS. Quantum GIS dikembangkan pertama kali oleh Gary Sherman pada awal tahun 2002, dan hal tersebut menjadi proyek inkubator *Open Source Geospatial Foundation* pada tahun 2004 dan resmi dirilis pada Januari 2009.

Dalam pengolahan data geografis, QGIS mempunyai banyak keunggulan, dikarenakan software ini bersifat *open source* maka dukungan dari pengguna baik dari dokumentasi maupun dari forum resmi di internet sangat banyak. Dukungan tersebut berupa *plugin-plugin* yang dapat digunakan untuk pengolahan data informasi geografis. Selain plugin yang selalu dikembangkan, dalam QGIS terdapat *tool-tool* yang bermanfaat, diantara yaitu *research tool*, *Geoprocessing tools*, *Geometry tools*, *Analysis tools* dan *processing tools*.

Pada *research tools* terdapat *tool* pembantu, yaitu *tool* untuk pembuatan grid, pembuatan *random point*, dan seleksi secara *random*. Pembuatan grid dilakukan berdasarkan layer yang terpilih. Pembuatan *random point* dilakukan untuk memunculkan titik-titik lokasi secara acak. Untuk pemilihan secara acak, dilakukan untuk menentukan lokasi yang diseleksi secara acak yang di *generate* oleh sistem.

Pada *Geoprocessing tools* terdapat tools yang dapat digunakan untuk menggabungkan data-data spasial. *Tools* ini hampir sama dengan data pengolahan gambar, yaitu terdapat *intersection* yang digunakan untuk memangkas data

berdasarkan data yang lain, *union* digunakan untuk mengabungkan dua data menjadi satu atau menggabungkan dua layer menjadi satu dan *clip* yang digunakan untuk mengambil suatu data pada layer aktif berdasarkan layer yang dipilih.

Pada *Analysis tools* terdapat beberapa *tool* yang digunakan untuk pengukuran, diantaranya tools yang berfungsi sebagai penghitung jarak antara koordinat satu dengan koordinat yang lain, terdapat *distance matrix* yaitu tool yang digunakan untuk memetakan jarak dan nanti disimpan dalam bentuk *matriks*, dan terdapat tools *sum line lengths* yang berguna untuk menghitung total baris yang dibuat di QGIS. Baris tersebut dapat berupa jalan, sungai, maupun data spasial yang lain.

Pada *processing tools* terdapat macam-macam *tool* yang berisi proses-proses pengolahan data. Pengolahan data dapat berupa perhitungan *matrix*, perhitungan jumlah titik, perhitungan grid dan lain-lain. Hampir semua proses pengolahan data berada pada tool ini.

2.2.4 Algoritma *Greedy*

Algoritma *greedy* merupakan metode yang populer untuk memecahkan masalah optimasi. Dalam permasalahan optimasi, terdapat 2 kriteria yang akan dicapai, yang pertama adalah memaksimalkan sesuatu dan yang lain nya yaitu meminimalkan sesuatu tersebut. Dalam hal ini dapat dikatakan bahwa algoritma *greedy* digunakan untuk mencari solusi maksimum. Arti dari kata *greedy* sendiri yaitu rakus, tamak atau loba dengan prinsip “*take what you can get now!*”. Cara memanfaatkan algoritma ini dengan cara membentuk solusi langkah per langkah dengan setiap langkah yang terdapat banyak pilihan yang perlu dieksplorasi. Oleh karena itu, pada setiap langkah harus dibuat keputusan yang terbaik dalam menentukan pilihan. Pada setiap langkah, kita membuat pilihan optimum lokal (*local optimum*) dengan harapan bahwa langkah selanjutnya mengarah ke solusi optimum global (*global optimum*).

Beberapa elemen-elemen penyusun sebuah algoritma *Greedy* antara lain,

1. Himpunan kandidat, yaitu himpunan seluruh kemungkinan solusi yang masih memungkinkan untuk dapat mengarah ke solusi.

2. Himpunan solusi, yaitu sebuah himpunan bagian dari himpunan kandidat yang dianggap menjadi solusi. Sebuah solusi yang terbentuk akan terbentuk dari hasil seleksi oleh fungsi seleksi, fungsi kelayakan, dan fungsi obyektif.
3. Fungsi seleksi, yaitu fungsi yang digunakan untuk melakukan pemilihan di tiap langkah. Pilihan yang akan diambil merupakan pilihan yang dapat memberikan keuntungan paling maksimal.
4. Fungsi kelayakan, yaitu fungsi yang digunakan untuk melakukan verifikasi sebuah kelayakan sebuah solusi dengan batasan-batasan yang didefinisikan pada permasalahan.
5. Fungsi obyektif, yaitu fungsi yang menyatakan obyektif (*goal*) dari algoritma *Greedy*.

Langkah dasar dari perhitungan algoritma *greedy* yaitu

1. $x(i)$ merupakan himpunan kandidat, berisi beberapa kandidat yang nantinya akan membagi bilangan yang akan dipecah.
2. y merupakan bilangan yang akan dipecah.
3. hasil(i) merupakan hasil pembagian dari y dengan $x(i)$.

Sisa dari pembagian tersebut akan dibagi dengan bilangan himpunan $x(i)$ sampai mendapatkan hasil 0 atau optimum.

2.2.5 Algoritma *Greedy Remove*

Prosedur algoritma *greedy remove* ini mirip dengan prosedur sebelumnya. Algoritma ini dimulai dari solusi awal yang terdiri dari semua node dan kemudian menghapus (*remove*) node sesuai fungsi *greedy* seperti pada algoritma sebelumnya [4]. Namun, fungsi *greedy* disini agak berbeda. Hal ini termasuk istilah tambahan, yang mengukur jumlah koneksi tambahan yang dapat dilayani oleh solusi. Keuntungan dari prosedur ini yaitu solusi yang layak pasti ada karena solusinya termasuk semua kandidat yang harus memenuhi batasan.

Batasan dalam algoritma *greedy remove* yaitu jika semua koneksi memiliki nilai masing-masing, solusi yang diambil yaitu solusi yang memiliki nilai dibawah nilai *greedy* yang sudah ditetapkan yaitu $\geq x$, maka solusi dari *greedy remove* terdapat pada $< x$.

2.2.6 Algoritma Genetika

Algoritma genetika merupakan algoritma yang berasal dari proses genetika pada suatu makhluk hidup. Algoritma ini diilhami dari teori evolusi Darwin dimana algoritma ini bekerja dengan sejumlah individu dalam suatu populasi. Algoritma ini menerapkan proses seleksi alam dimana siapa yang kuat atau baik, dia akan bertahan. Penggunaan algoritma genetika dalam pemrograman yaitu pada kecerdasan buatan atau lebih dikenal dengan istilah *artificial intelligence*.

Pada awal tahun 1975 John Holland mulai menyampaikan algoritma ini. Algoritma genetika ini digunakan untuk menyelesaikan sebuah problematika dengan cara berusaha untuk mendapatkan *chromosome-chromosome* yang baik, melalui proses memanipulasi materi dan sifat (*gen*) dari *chromosome*. Hasil dari evaluasi *chromosome* adalah informasi yang berupa nilai kebugaran (*fitness*) setiap *chromosome* yang dihasilkan, dan digunakan untuk seleksi *chromosome* pada proses berikutnya. *Chromosome* dengan nilai kebugaran atau nilai *fitness* terbaik yang bertahan hidup dan selalu direproduksi.

Individu-individu yang lahir sebelum dinyatakan sebagai individu akan mengalami proses yang berhubungan dengan perubahan nilai *gen*, perubahan tersebut ada yang berupa perubahan yang berbeda dengan induk aslinya dan ada pula yang menyerupai sifat induk aslinya, hal ini ditentukan oleh kemungkinan (*probabilitas*) kawin silang dan mutasi.

Setelah proses mutasi gen inilah individu-individu baru dinyatakan telah lahir. Setelah proses tersebut individu-individu hasil mutasi akan bergabung menjadi satu dan akan membentuk populasi baru yang menggantikan populasi lama. Proses satu pengulangan ini dinamakan dengan satu generasi. Generasi akan terus berlanjut samapi proses *fitness* terpenuhi. Selain berhenti karena proses *fitness*, generasi juga akan berhenti ketika kriteria generasi atau banyak nya generasi sudah terpenuhi. Banyaknya generasi dibatasi dengan tujuan agar proses pencarian nilai *fitness* ketika terjebak dalam *local optimum*, proses akan berhenti mengikuti prosedur yang telah ditetapkan.

Tabel 2.3 Terminologi Algoritma Genetika

Terminologi	Keterangan
Chromosome	Merupakan kumpulan <i>gen</i> yang membentuk satu kesatuan nilai, dari model <i>chromosome</i> inilah algoritma genetika dapat dibedakan menjadi algoritma genetika <i>chromosome biner</i> , <i>chromosome float</i> dan <i>chromosome kombinatorial</i> . Model algoritma genetika sangat ditentukan nilai pada setiap <i>gen</i> -nya.
Gen	Merupakan elemen terkecil yang menyatakan nilai parameter dari individu, didalam algoritma genetika terdapat berbagai macam jenis gen yaitu <i>gen biner</i> , <i>gen float</i> atau <i>gen kombinasi</i> .
Individu	Merupakan satu kesatuan <i>chromosome</i> yang menyatakan satu kemungkinan penyelesaian
Allele	Merupakan nilai yang dimasukkan pada <i>gen</i>
Populasi	Merupakan sekumpulan individu yang menjadi dasar dari pengolahan seleksi alamiah
Generasi	Merupakan pengulangan (<i>loop</i>) dari proses seleksi alamiah dimulai dari satu populasi hingga membentuk populasi yang baru.

(1) *Terminologi* algoritma genetika

Algoritma genetika menggunakan mekanisme genetika yang ada pada proses alami dan sistem buatan secara *numerik*. Rancangan atau proses yang digunakan diambil dari disiplin ilmu biologi dan ilmu komputer. *Terminologi* dalam algoritma genetika adalah seperti Tabel 2.3.

(2) Parameter Algoritma genetika

Sebelum mengevaluasi nilai *fitness* masing-masing individu, terlebih dahulu kita kenal beberapa parameter yang digunakan dalam algoritma GA. Parameter-parameter tersebut digunakan untuk melihat kompleksitas dari GA. Parameter yang digunakan tersebut adalah :

Jumlah Generasi

Merupakan jumlah perulangan atau iterasi dilakukannya *crossover*, *mutasi* dan seleksi. Jumlah generasi ini mempengaruhi kestabilan *output* dan lama iterasi

(waktu proses GA). Jumlah generasi yang besar dapat mengarah ke suatu solusi yang optimal, namun akan membutuhkan waktu yang lama. Sedangkan jika jumlah iterasinya sedikit dapat mengakibatkan solusi akan terjebak pada nilai *local optimum*.

Ukuran Populasi

Ukuran populasi mempengaruhi kinerja dan efektifitas dari GA. Jika ukuran populasi kecil maka populasi tidak menyediakan cukup materi untuk mencangkup ruang permasalahan, sehingga pada umumnya kinerja GA menjadi buruk. Dalam hal inilah dibutuhkan ruang populasi yang lebih besar untuk merepresentasikan keseluruhan ruang permasalahan. Selain itu populasi yang besar dapat mencegah terjadinya konvergensi pada wilayah lokal. Banyak aplikasi GA yang menggunakan populasi antara 50-100.

Seleksi

Proses ini digunakan untuk menyeleksi atau memilih individu yang dapat melanjutkan ke proses selanjutnya. Proses seleksi ini memilih individu terbaik yang terdapat pada populasi yang nantinya akan digunakan untuk membuat populasi baru dan menggantikan populasi lama. Jika suatu individu terpilih dalam proses ini, maka individu ini akan dimasukkan kedalam populasi baru, dan jika individu tidak terpilih, maka individu akan dibuang dan akan digantikan dengan individu yang lain.

Probabilitas Kawin Silang (*cross over*)

Probabilitas ini digunakan untuk mengendalikan operator kawin silang. Dalam hal ini, dalam populasi terdapat individu yang melakukan kawin silang. Semakin besar nilai probabilitas kawin silang maka semakin cepat struktur baru diperkenalkan dalam populasi. Namun jika probabilitas kawin silang terlalu besar maka struktur dengan nilai objektif yang baik dapat hilang dengan lebih cepat dari seleksi. Sebaliknya probabilitas kawin silang kecil maka akan menghalangi proses pencarian dalam GA. Banyak aplikasi GA yang menggunakan nilai probabilitas kawin silang atau probabilitas *crossover* antara 0.65-1.

Probabilitas Mutasi

Mutasi digunakan untuk meningkatkan variasi populasi. Probabilitas mutasi ini digunakan untuk menentukan tingkat mutasi terjadi. Probabilitas yang rendah akan menyebabkan *gen-gen* yang berpotensi tidak dicoba dan sebaliknya tingkat mutasi yang tinggi akan menyebabkan keturunan akan semakin mirip dengan induknya dalam GA mutasi menjalankan aturan penting yaitu :

- a. Mengganti *gen* yang hilang selama proses seleksi
- b. Menyediakan *gen-gen* yang tidak muncul pada saat inisialisasi awal populasi.

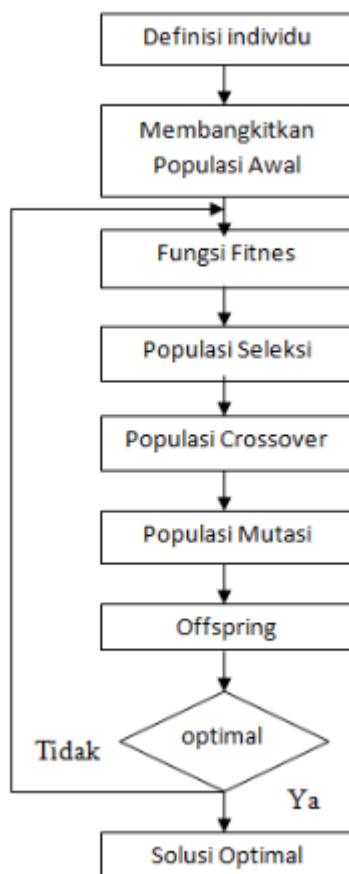
Panjang Chromosome

Panjang *chromosome* berbeda-beda sesuai dengan model permasalahan. Titik solusi dari ruang permasalahan dikodekan dalam bentuk *chromosome/ string* yang terdiri dari komponen genetik terkecil yaitu *gen*. Pengkodean dapat memakai bilangan seperti string *biner*, *integer*, *floating point* dan abjad.

Mekanisme secara umum dari GA digambarkan oleh Goldberg. Proses kerjanya diawali dengan inisialisasi satu rangkaian nilai random yang disebut dengan populasi. Setiap individu dalam satu populasi dinamakan *chromosome*. Setiap individu dievaluasi dengan menggunakan nilai kebugaran atau *fitness*. Untuk menghasilkan generasi selanjutnya sebagai individu baru yang disebut dengan *offspring*, dibentuk melalui persilangan dua individu dengan menggunakan operator kawin silang (*crossover*) dan memodifikasi sebuah *chromosome* atau disebut juga mutasi.

(3) Struktur umum dalam Algoritma Genetika

Dalam algoritma genetika, terdapat beberapa urutan proses yang perlu dilakukan diantaranya seperti yang terlihat pada gambar dibawah ini :



Gambar 2.4 Struktur umum algoritma genetika

1. Definisi Individu

Definisi individu adalah proses pertama yang harus dilakukan dalam pembuatan atau perancangan algoritma genetika. Dalam hal ini individu menyatakan salah satu solusi (yang mungkin) dari suatu permasalahan yang diangkat. Definsi individu atau jumlah individu disebut juga merepresentasikan *chromosome* yang akan diproses nanti, dilakukan dengan mendefinisikan jumlah dan tipe dari gen yang digunakan dan tentunya dapat mewakili solusi permasalahan yang diangkat.

2. Membangkitkan Populasi Awal

Proses ini dilakukan dengan membangkitkan populasi secara acak, dimana populasi tersebut berisi beberapa *chromosome* yang telah didefinisikan sebelumnya. Dalam proses ini perlu diperhatikan syarat-syarat yang harus dipenuhi

untuk menunjukkan suatu solusi dari permasalahan dan jumlah *chromosome* yang digunakan dalam satu populasi. Jika *chromosome* yang digunakan terlalu sedikit, maka individu yang dapat digunakan untuk proses *crossover* dan mutasi akan sangat terbatas. Sehingga menyia-nyiakan proses yang ada. Tetapi jika jumlah *chromosome* yang digunakan terlalu banyak akan memperlambat proses algoritma genetika yang dilakukan. Jumlah *chromosome* yang dianjurkan lebih besar dari jumlah *gen* yang ada dalam satu *chromosome*, tetapi juga harus disesuaikan dengan permasalahan, apabila jumlah *gen*-nya terlalu banyak, tidak juga dianjurkan seperti itu.

3. Fungsi Evaluasi Kebugaran (*Fitness*)

Fungsi kebugaran menggambarkan hasil atau solusi yang telah dikodekan. Selama proses GA, induk harus mengalami *crossover* dan mutasi untuk memperoleh keturunan. Jika GA dirancang dengan baik, maka populasi akan mengalami sebuah konvergensi dan mengarah pada solusi yang minimum. Pada permulaan minimisasi, kemungkinan nilai kebugaran akan memiliki rentang yang lebar. seiring dengan bertambahnya generasi beberapa *chromosome* mendominasi populasi dan mengakibatkan rentang nilai kebugaran makin kecil. Hal ini dapat mengakibatkan konvergensi dini. Dalam hal ini hanya proses mutasi yang mampu menghasilkan *chromosome* yang relatif baru dan merupakan cara untuk menghindari *chromosome* tertentu mendominasi populasi.

4. Proses Seleksi

Salah satu proses dalam GA adalah proses seleksi yaitu proses pemilihan *chromosome* yang akan dilakukan proses kawin silang dengan *chromosome* dari individu lain. Dasar proses seleksi adalah teori evolusi Darwin tentang proses seleksi pada makhluk hidup.

Operasi seleksi dilakukan dengan cara memperhatikan nilai *fitness* dari tiap-tiap individu. Manakah yang dapat dipergunakan untuk generasi selanjutnya. Seleksi ini digunakan untuk mendapatkan calon induk yang baik, semakin tinggi nilai *fitnessnya* maka semakin besar juga kemungkinan individu tersebut terpilih. Terdapat beberapa macam cara seleksi untuk mendapatkan calon induk yang baik,

diantaranya adalah seleksi *roulette wheel*, *steady state*, *tournament* dan *rank*. Beberapa penjelasan tentang keempat metode seleksi diatas adalah sebagai berikut:

a. Roulette Wheel

Calon induk yang akan dipilih berdasarkan nilai *fitness* yang dimilikinya, semakin baik individu tersebut yang ditunjukkan dengan semakin besar nilai *fitnessnya* akan mendapatkan kemungkinan yang lebih besar untuk terpilih sebagai induk misalkan saja *roulette wheel* merupakan tempat untuk menampung seluruh *chromosome* dari tiap populasi, maka besarnya tempat dari *roulette wheel* tersebut menunjukkan seberapa besar nilai *fitness* yang dimiliki oleh suatu *chromosome*, semakin besar nilai *fitness* tersebut, maka semakin besar pula tempat yang tersedia.

b. Steady state

Metode *steady state* tidak banyak digunakan dalam proses seleksi karena dilakukan dengan mempertahankan individu yang terbaik. Pada setiap generasi akan dipilih beberapa *chromosome* dengan nilai *fitness* yang terbaik sebagai induk, sedangkan *chromosome-chromosome* yang memiliki nilai *fitness* terburuk akan digantikan dengan *offspring* yang baru. Sehingga pada generasi selanjutnya akan terdapat beberapa populasi yang bertahan.

c. Tournament

Dalam metode seleksi *tournament* sejumlah n individu dipilih secara random dan kemudian menentukan *fitnessnya*. Kebanyakan metode ini digunakan pada *binary*, dimana hanya dua individu yang dipilih.

d. Rank

Seleksi ini memperbaiki proses seleksi yang sebelumnya yaitu *roulate whell* karena pada seleksi tersebut kemungkinan salah satu *chromosome* mempunyai nilai *fitness* yang mendominasi hingga 90% bisa terjadi, sehingga nilai *fitness* yang lain akan mempunyai kemungkinan yang sangat kecil sekali untuk terpilih.

5. Proses *Crossover*

Crossover adalah salah satu operator penting dalam algoritma genetika, metode dan tipe *crossover* yang dilakukan tergantung dari *encoding* dan permasalahan yang diangkat. Proses *crossover* atau dapat dikatan kawin silang merupakan proses dimana terjadi perpindahan *gen* antara individu satu dengan individu yang lain.

6. Proses Mutasi

Prosedur mutasi sangatlah sederhana. Untuk semua *gen* yang ada, jika bilangan random yang dibangkitkan kurang dari probabilitas mutasi yang ditentukan maka ubah *gen* tersebut menjadi nilai kebalikannya (dalam *binary encoding*, 0 diubah menjadi 1 dan 1 diubah menjadi 0). Biasanya *Pmut* diset sebagai $1/n$, dimana n adalah jumlah *gen* dalam *chromosome*. Dengan probabilitas mutasi sebesar ini berarti mutasi hanya terjadi pada sekitar satu *gen* saja. Pada AG sederhana, nilai probabilitas mutasi adalah tetap selama evolusi.

7. Proses *Elitisme*

Untuk menjaga nilai individu yang terbaik yang terpilih, dilakukan proses *elitisme*. Proses ini menggambarkan bahwa individu terbaik dipertahankan dan individu yang lain dilakukan proses mutasi atau *crossover*. Dilakukan proses *elitisme* karena seleksi individu dilakukan secara random, maka tidak ada jaminan bahwa suatu individu bernilai *fitness* tertinggi akan selalu terpilih. Kalaupun individu bernilai *fitness* tertinggi terpilih, mungkin saja individu tersebut akan rusak (nilai *fitnessnya* menurun) karena proses pindah silang.

2.2.7 XAMPP

XAMPP adalah perangkat lunak yang mendukung banyak sistem operasi, dan merupakan kompilasi dari beberapa program. Fungsinya adalah sebagai server yang berdiri sendiri (*localhost*). Nama XAMPP sendiri merupakan singkatan dari X (empat sistem operasi apapun), Apache, MySQL, PHP dan Perl. Dengan menginstall XAMPP maka tidak perlu lagi melakukan instalasi dan konfigurasi web

server Apache, PHP dan MySQL secara manual. XAMPP akan menginstalasi dan mengkonfigurasikannya secara otomatis.

Dalam satu paket XAMPP tersedia :

1. Apache Cgi-Bin
2. FTP
3. Mercury Mail (SMTP)
4. PHP
5. MySql
6. Perl
7. PHP Myadmin
8. Webalizer

a. Apache

Tugas utama apache adalah menampilkan halaman web yang benar, sesuai dengan program PHP yang telah dibuat. Apache bersifat *open source*, artinya setiap orang boleh menggunakan, mengambil, dan mengubah kode programnya. Sampai saat ini Apache telah mengalami beberapa perkembangan versi.

b. PHP

PHP (*Personal Home Page*) merupakan bahasa pemrograman yang digunakan untuk membuat web yang bersifat *server-side scripting*. PHP memungkinkan kita untuk membuat halaman web yang bersifat dinamis, yakni dimana isi informasi website berubah-ubah, dan interaktif dua arah baik dari pemilik maupun pengguna website. PHP dapat dijalankan pada berbagai macam *Operating System*, seperti Windows, Linux, dan Mac OS.

Sistem manajemen *database* yang sering digunakan bersama PHP adalah MySQL. Namun selain itu, PHP juga mendukung sistem manajemen *database* Oracle, Microsoft Access, Interbase, d-Base, PostgreSQL, dan lain-lain. Sama seperti Apache, PHP juga bersifat *open source*.

Pada penelitian ini, php digunakan sebagai bahasa pemrograman pengolah algoritma. Pada *script* php yang dibuat, terdapat fungsi-fungsi yang dibuat. Argumen fungsi digunakan untuk menyerdehanakan fungsi yang akan ditulis pada *script*. Dapat dikatakan bahwa fungsi dibuat agar sekumpulan tugas yang dilakukan berulang-ulang dapat ditulis dalam satu kode saja, sehingga jika ada tugas yang

sama ingin dikerjakan lagi, fungsi yang telah dibuat tinggal dipanggil kedalam baris kode.

c. Mysql

SQL merupakan kepanjangan dari *Structured Query Language* yang artinya bahasa terstruktur yang digunakan untuk mengolah *database*. MySQL merupakan sistem manajemen database yang bersifat *open source*.

MySQL digunakan untuk membuat dan mengelola *database* beserta isinya, seperti menambahkan, mengubah, dan menghapus data. MySQL juga bersifat *relational*, artinya data-data yang dikelola akan diletakkan pada beberapa tabel terpisah, sehingga proses manipulasi data akan menjadi lebih cepat.

d. PHPMyAdmin

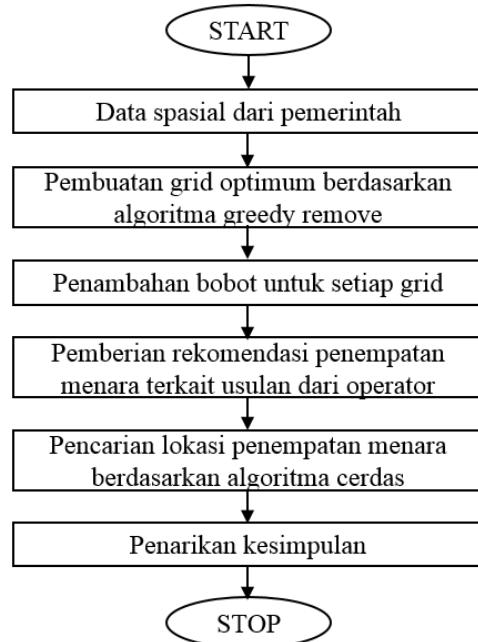
Salah satu perangkat lunak yang digunakan untuk mengelola *database* dalam MySQL adalah PHPMyAdmin. Dengan PHPMyAdmin kita dapat dengan mudah membuat tabel, mengisi data, dan banyak lagi hal lainnya tanpa harus hafal perintahnya, namun cukup dengan mengisi tabel-tabel yang telah tersedia.

BAB 3

METODOLOGI PENELITIAN

3.1 Langkah-langkah Penelitian

Pada bagian ini akan dijelaskan mengenai langkah-langkah yang dilakukan di dalam penelitian ini. Langkah-langkah ini secara umum dijelaskan menggunakan diagram alir seperti ditunjukkan pada Gambar 3.1. Penelitian diawali dengan mengumpulkan data spasial yang didapat dari pemerintah kota, selanjutnya masuk ke bagian pembuatan grid optimum. Pembuatan grid optimum ini berdasarkan algoritma *greedy remove*. Untuk dapat memberikan rekomendasi mengenai kawasan yang diperbolehkan dalam hal pendirian menara, Grid ditambahkan dengan bobot. Langkah selanjutnya yaitu mencari lokasi (grid) untuk penempatan BTS dan MCP secara optimal berdasarkan algoritma cerdas. Langkah terakhir menuliskan kesimpulan.



Gambar 3.1 Diagram Alir Langkah Penelitian

3.2 Data Spatial

Data spatial yang didapat dari pemerintah Kota Surabaya berupa peta dasar kota surabaya, data bangunan diatas nya dan terakhir data koordinat tower. Data-data ini menggunakan *geo* referensi berupa WGS84 dengan representasi koordinat berupa *latitude* dan *longitude*. Data peta dasar kota berupa garis pantai, jalan lokal, jalan tol, jalan utama, jalur kereta api, jalur pejalan kaki, bangunan dan sungai. Data bangunan yang didapat dari pemerintah yaitu bangunan militer, makam, industri, fasilitas umum dan ruang terbuka hijau (RTH). Data bangunan tersebut akan digunakan untuk mengambil rekomendasi apakah koordinat yang diusulkan oleh operator boleh dibangun menara ataukah harus digeser.

3.2.1 Peta Dasar Kota Surabaya

Data peta dasar yang didapat dari Pemerintah Kota Surabaya yaitu data jalan utama, data jalan lokal, data jalur kereta api, jalur pejalan kaki, garis pantai. Data-data tersebut dimasukkan ke dalam software QGIS agar data tersebut dapat terbaca. Gambar 3.2 menunjukkan gambaran peta dasar Kota Surabaya di ambil dari software QGIS.



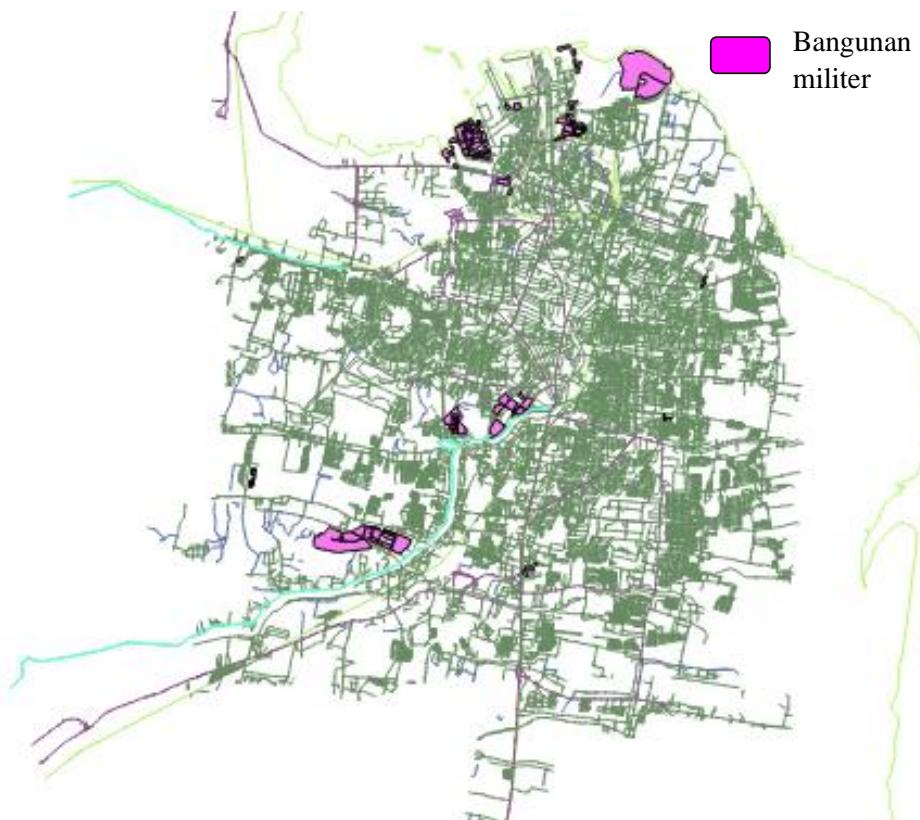
Gambar 3.2 Peta Dasar Kota Surabaya

3.2.2 Data Bangunan

Data bangunan yang didapat dari Pemerintah Kota Surabaya seperti yang dijelaskan pada bab sebelumnya yaitu bangunan militer, makam, bangunan industri dan fasilitas umum. Data bangunan selanjutnya di masukkan kedalam software QGIS dengan layer diatas peta dasar Kota Surabaya.

3.2.2.1 Bangunan Militer

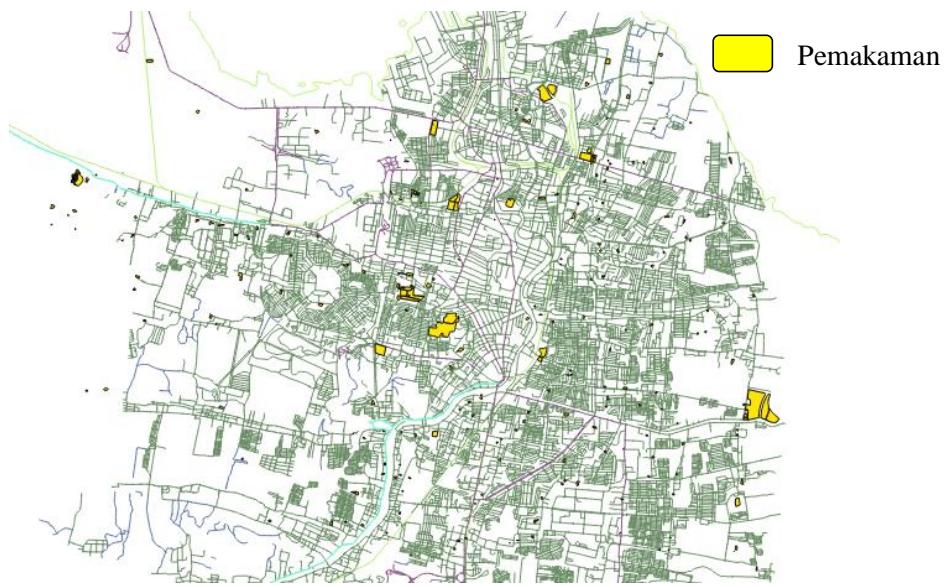
Berikut ini layer data bangunan militer yang ditempatkan diatas layer peta dasar Kota Surabaya ditunjukkan pada gambar 3.3.



Gambar 3.3 Bangunan Militer

3.2.2.2 Pemakaman

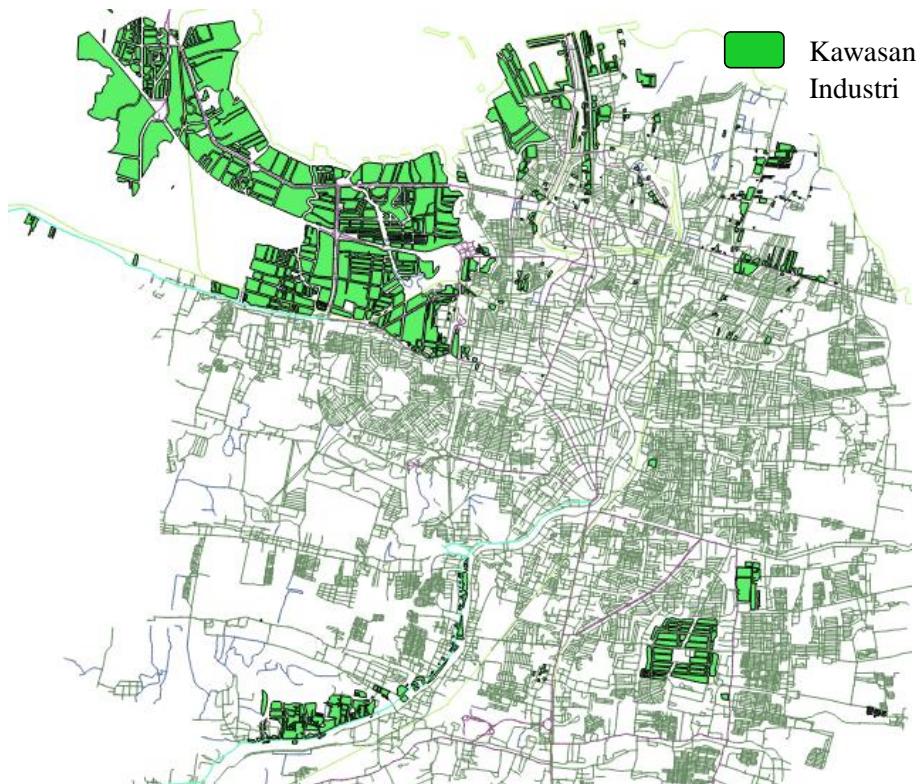
Berikut ini layer data kawasan pemakaman yang ditempatkan diatas layer peta dasar Kota Surabaya ditunjukkan pada gambar 3.4.



Gambar 3.4 Pemakaman

3.2.2.3 Industri

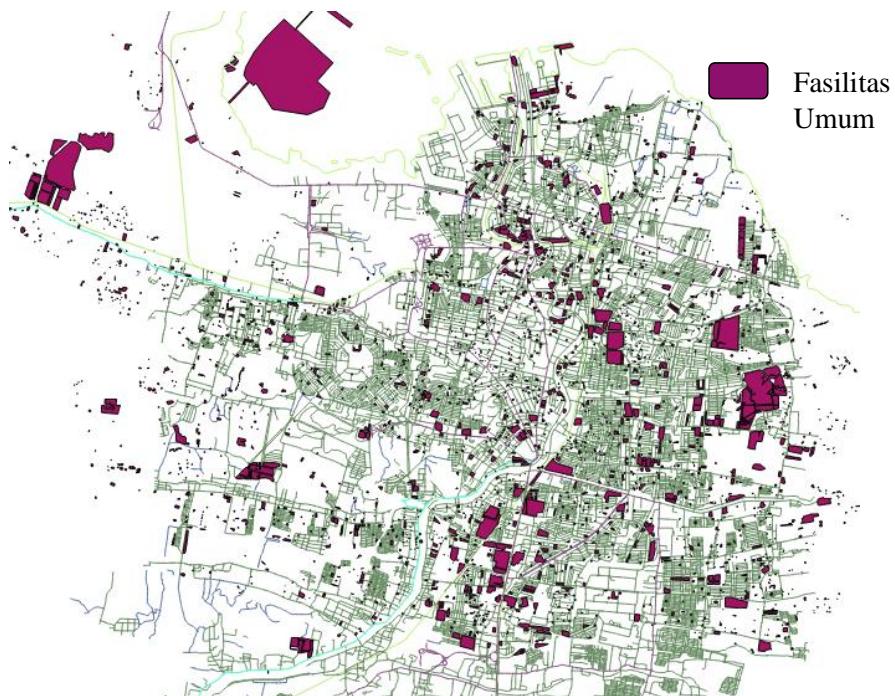
Berikut ini layer data kawasan industri yang ditempatkan diatas layer peta dasar Kota Surabaya ditunjukan pada gambar 3.5.



Gambar 3.5 Kawasan perindustrian

3.2.2.4 Fasilitas Umum

Berikut ini layer data bangunan dengan kategori fasilitas umum yang ditempatkan diatas layer peta dasar Kota Surabaya ditunjukkan pada gambar 3.6.



Gambar 3.6 Bangunan Fasilitas Umum

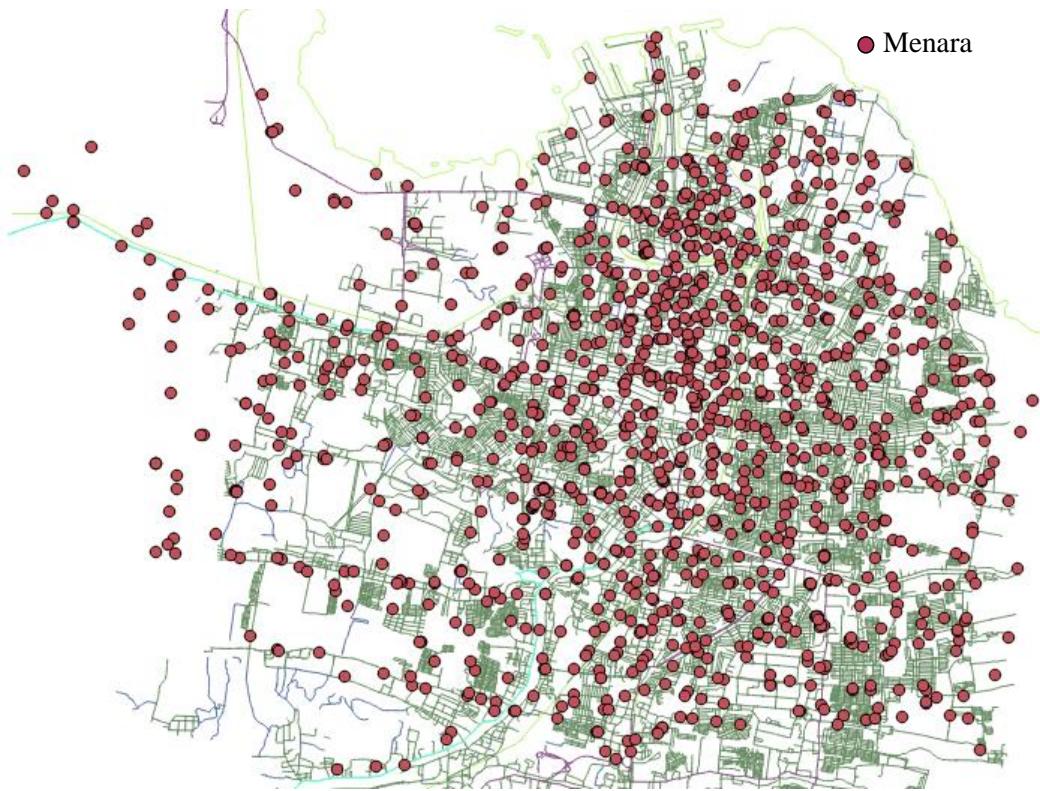
3.2.3 Koordinat Tower (BTS dan MCP)

Data koordinat tower berisi data menara yang berada di Kota Surabaya. Data tersebut berjumlah 1254 titik sebagai wujud representasi menara milik operator. Data menara memuat informasi berupa koordinat latitude dan longitude. Gambar 3.7 merupakan cuplikan sebagian data koordinat menara.

A	B	C	D	E	F
No.	Provider	Alamat Persil	Latt	Long	Tinggi
1	TBG	Masjid Agung Al Akbar Surabaya	-7.335380	112.715000	42
2	TBG	TANDES Jl. Manukan Tama No 48	-7.266371	112.666148	42
3	TBG	WISMA PERMAI SURABAYA	-7.321250	112.807000	42
4	TBG	PENJARINGAN SARI	-7.317426	112.793705	42
5	TBG	MULYOSARI TIMUR	-7.264560	112.797688	42
6	TBG	BIBIS SURABAYA	-7.207507	112.750461	52
7	TBG	TANJUNG SADARI	-7.224673	112.719629	42
8	TBG	PETI KEMAS SURABAYA	-7.205906	112.721067	42
9	TBG	KREMBANGAN UTARA SURABAYA	-7.263579	112.660643	42
10	TBG	MOJO KLANGGRU	-7.272895	112.787978	42
11	TBG	MENGANTI 2	-7.329730	112.704000	42
12	TBG	SIDOTOTO LOR	-7.231309	112.753597	12
13	TBG	MULYOSARI TIMUR	-7.268270	112.803030	25
14	TBG	I T S Jl.Kanoman Gg.II Gebang Lor Kel.Gebang putih Kec.Sukolilo	-7.280336	112.790450	42
15	TBG	NGAGEL MULYO Jl. Bratang Gede Gg.3 No.83 A, RT. 02/07, Kel Ngagel Rejo	-7.298330	112.753360	32

Gambar 3.7 File excel data koordinat menara

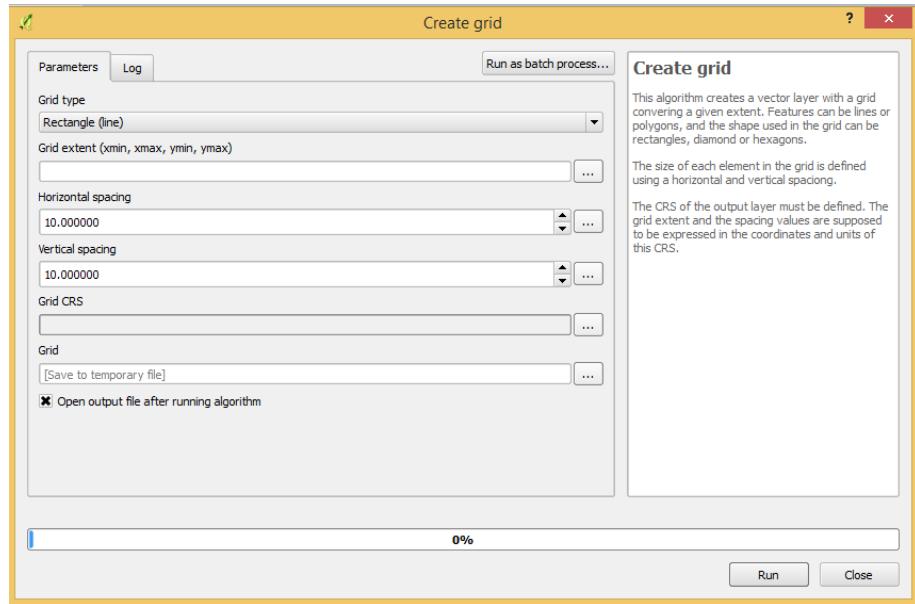
Gambar 3.8 merupakan gambar layer menara yang ditempatkan diatas layer peta dasar Kota Surabaya.



Gambar 3.8 Layer Menara

3.3 Metode Pembuatan Grid berdasarkan Posisi Menara

Pada tahap ini, layer yang akan digunakan yaitu layer peta dasar dan layer menara. Pembuatan grid digunakan untuk membagi-bagi jumlah tower perwilayah. Pembuatan grid dapat dilakukan dengan 2 cara, yaitu pembuatan grid dari aplikasi QGIS dan pembuatan grid dengan perhitungan secara manual. Pembuatan grid secara manual masuk kedalam subbab pemodelan algoritma *greedy*. Pembuatan grid menggunakan software QGIS dilakukan dengan bantuan *vector creation tools* kemudian pilih *create grid*. Gambar 3.9 menunjukkan tampilan dari *tools create grid* pada aplikasi QGIS.



Gambar 3.9 *Tools create grid* dari software QGIS

Pada *tools create grid* diatas, terdapat beberapa kolom yang harus diisi diantaranya *grid type*, *grid extent*, *horizontal spacing*, *vertical spacing*, dan *grid*. Pilihan dari kolom *grid type* yaitu *rectangle (line)*, *rectangle (polygon)*, *diamond (polygon)* dan *hexagon (polygon)*. Untuk pembuatan grid pada tahap ini, *type* yang digunakan yaitu *type rectangle (line)*. Kolom *grid extent* adalah kolom pembatas atas, bawah, kanan dan kiri. Selain menuliskan batas secara manual, pada kolom ini dapat dipilih layer yang akan digunakan sebagai pembatas. Layer yang dipilih yaitu layer menara. Pada kolom *horizontal spacing* dan *vertical spacing* diisi jarak batas antara garis satu dengan garis yang lain. Kolom ini menggunakan satuan derajat, sehingga jika ingin memasukkan satuan meter harus diubah dahulu menjadi derajat. Pada penelitian ini, penulis menggunakan jarak grid sebesar 200 meter. Hal ini sesuai dengan ukuran atau range dari menara jenis MCP (*micro cell poles*). Berikut ini konversi dari meter menjadi derajat.

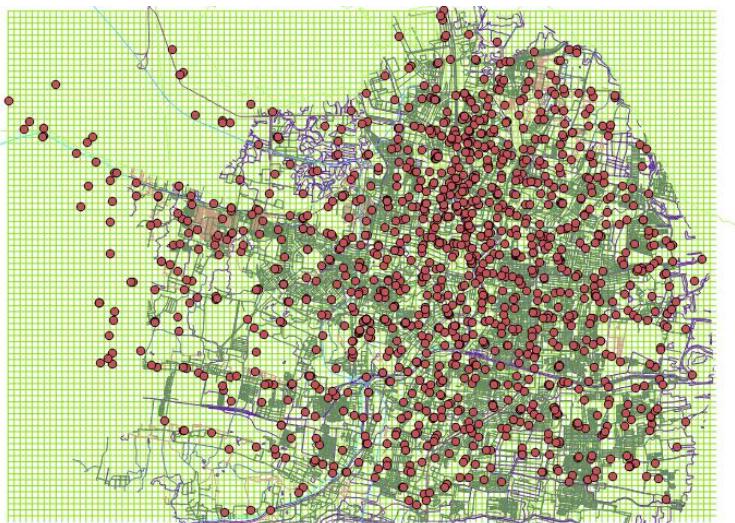
$$200 \text{ meter} / 30,92(\text{meter/sec}) = 6,468 \text{ sec}$$

$$6,468 \text{ sec} / 60 (\text{sec/min}) = 0,1078 \text{ min}$$

$$0,1078 \text{ min} / 60 (\text{min/deg}) = 0,00179^\circ \approx 0,0018^\circ \quad (3.1)$$

Dari perhitungan 3.1 menghasilkan nilai $0,0018^\circ$ yang akan digunakan untuk mengisi kolom *horizontal* dan *vertical spacing*. Pada kolom terakhir yaitu

kolom *grid* digunakan untuk menyimpan file hasil generate. Hasil dari generate software QGIS ditunjukkan pada gambar 3.10.



Gambar 3.10 Hasil generate grid

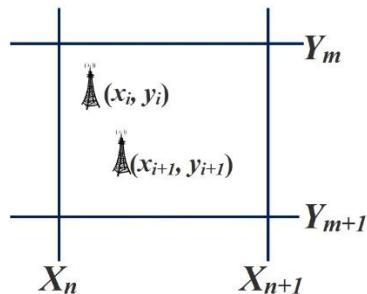
3.4 Metode Pencarian Grid Optimal berdasarkan Posisi Menara

Pemerintah masih menggunakan metode konvensional dalam hal perijinan penempatan menara. Metode konvensional tersebut diantaranya yaitu jika terdapat proposal pengajuan menara baru dari operator, pemerintah melihat lokasi tersebut dan melihat tower/menara disekelilingnya, apakah disekitar tempat tersebut terdapat banyak menara atau tidak. Jika banyak menara, maka pemerintah berhak menolak proposal tersebut. Jika terdapat sedikit menara, pemerintah memberikan ijin pendirian menara di lokasi yang diusulkan. Hal ini sama dengan konsep greedy remove, dimana lokasi yang terdapat paling banyak menara, dihindari untuk dipasang atau didirikan menara baru. Dan untuk lokasi yang terdapat sedikit menara, dipilih atau diperbolehkan untuk mendirikan menara.

Algoritma greedy remove yang digunakan pada penelitian ini ditujukan untuk mencari grid optimal. Grid optimal didapat dari perbandingan jumlah grid dengan hasil penjumlahan isi grid. Jumlah grid dalam hal ini yaitu grid yang didalam nya terdapat 4 tower atau lebih. Sedangkan penjumlahan isi grid yaitu jumlah dari isi grid yang didalam nya terdapat 4 tower atau lebih. Digunakan parameter 4 tower karena mengacu kepada peraturan pemerintah terkait jumlah maximum tower pada suatu wilayah [3].

Perancangan algoritma GR (greedy remove) dimulai dari input file koordinat. Selanjutnya yaitu menentukan berapa iterasi yang akan digunakan. Semakin banyak iterasi akan menghasilkan nilai akhir yang lebih akurat, akan tetapi waktu yang dibutuhkan lebih lama dikarenakan proses iterasi tersebut. Dalam perancangan ini, koordinat yang diinputkan berupa koordinat latitude dan longitude. Untuk mempermudah proses perhitungan, koordinat latitude dan longitude dirubah menjadi Y_n dan X_n . Selanjutnya mencari berapa banyak baris yang nantinya akan dirubah menjadi grid. Banyaknya garis dapat dicari dengan mencari nilai terendah dan nilai tertinggi dari latitude dan longitude. Nilai terendah dijumlahkan jarak grid sampai nilai tersebut melewati nilai tertinggi. Dari perhitungan tersebut, terdapat sejumlah garis yang merepresentasikan jarak antara grid satu dengan grid yang lainnya. Selanjutnya yaitu mencari terdapat berapa banyak titik yang masuk kedalam grid tersebut.

Menghitung titik yang terdapat di dalam grid, dapat dilakukan dengan cara menggabungkan hasil perhitungan sebelumnya menjadi seperti gambar 3.11 dibawah ini



Gambar 3.11 Mencari jumlah menara yang berada didalam grid

Berikut ini rumus yang digunakan untuk mencari banyaknya menara yang berada di dalam grid.

$$BX = \text{count}(X)$$

$$BY = \text{count}(Y) \quad (3.2)$$

```

for(m = 1; m ≤ BY; m++) {
    for(n = 1; n ≤ BX; n++) {
        GRIDp = count(xi ≤ Xn && xi > Xn+1
                      && yi ≤ Ym && yi > Ym+1)
    }
}

```

Dimana

$i = 1, 2, 3, \dots I$ jumlah menara

$p = 1, 2, 3, \dots P$ total grid yang telah digenerate

Keterangan

BX = menghitung banyaknya garis X (*longitude*) yang telah digenerate

BY = menghitung banyaknya garis Y (*latitude*) yang telah digenerate

x_i, y_i = koordinat tower

Setelah didapatkan nilai pergrid, selanjutnya menghitung berapa banyak grid yang mempunyai nilai lebih dari 3. Selain mencari nilai tersebut, dicari juga hasil kwadrat menara didalam nya. Perhitungan ini ditunjukkan pada rumus (3.4) dan (3.5).

$$LBH_q = \text{count 'grid' where } GRID_p \geq 4 \quad (3.4)$$

$$K_r = (\text{select 'grid' where } GRID_p \geq 4)^2$$

$$JML_q = \text{sum}(K_1, K_2, \dots, K_R) \quad (3.5)$$

Dimana

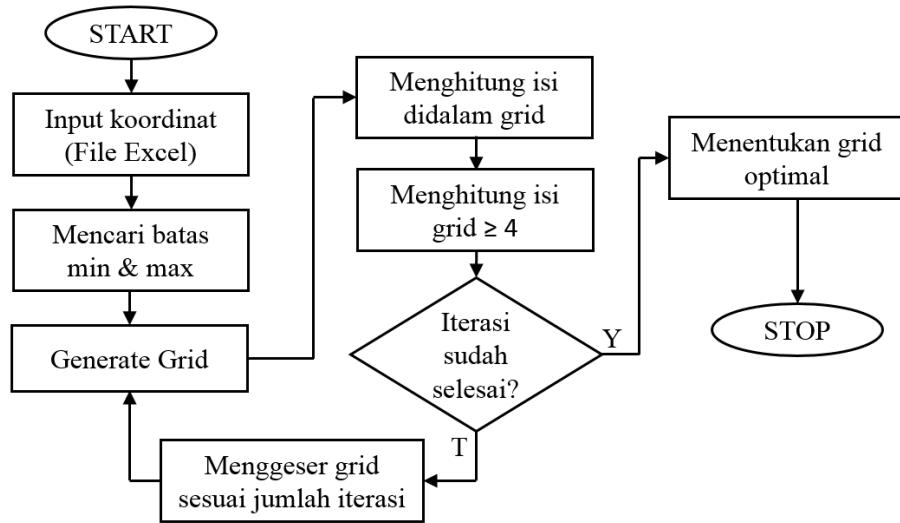
$q = 1, 2, 3, \dots Q$ total iterasi

$r = 1, 2, 3, \dots R$ banyaknya data

Langkah selanjutnya yaitu melakukan iterasi sesuai dengan banyak nya iterasi yang telah ditentukan di awal. Iterasi dimulai dari generate grid sampai langkah terakhir yaitu menghasilkan nilai LBH_q dan JML_q . Iterasi dilakukan untuk mencari grid paling optimal, sehingga untuk membedakan nilai satu dengan yang lainnya grid digeser sejauh G , hal ini ditunjukkan pada rumus (3.6).

$$G = \text{Jarak Grid / Iterasi} \quad (3.6)$$

Nilai awal dari koordinat yang telah diurutkan, ditambahkan dengan G kemudian melakukan proses generate grid seperti langkah sebelumnya. Setelah semua iterasi berhasil dilakukan, selanjutnya memilih pada iterasi keberapa grid paling optimal ditemukan. Grid optimal adalah grid yang memiliki nilai $LBHq$ dan $JMLq$ terbanyak. Gambar 3.12 menunjukkan proses perhitungan yang telah disusun.



Gambar 3.12 Proses perhitungan penentuan grid optimal

3.5 Metode Pencarian Grid Optimal berdasarkan Bobot

Metode pencarian tahap ini hampir sama dengan tahap sebelumnya. Yang membedakannya yaitu penambahan parameter bobot untuk setiap data spasial yang ada. Parameter ini mengacu kepada peraturan Walikota Surabaya [3] tentang pedoman pengendalian menara telekomunikasi di Kota Surabaya, serta peraturan Walikota Surabaya [11] tentang rencana tata ruang wilayah Kota Surabaya tahun 2014-2034. Data spasial dalam hal ini data bangunan dan data menara diberi bobot, pembobotan ini mengacu kepada Perwali Kota Surabaya, data ini terdapat pada lampiran. Bobot tertinggi terletak pada bobot menara dimana terdapat bangunan menara eksisting lebih dari 4, maka grid tersebut dilarang untuk didirikan menara. Dalam hal ini, bobot grid yang terdapat 4 menara atau lebih didalamnya diberikan bobot sebesar 10. Pemberian bobot berikutnya yaitu bobot untuk pemakaman, dimana makam merupakan ruang terbuka hijau, dan pada peraturan, suatu kota harus mempunyai minimal 20% RTH. Pada peraturan pembangunan menara sesuai

lampiran, makam merupakan kawasan yang dilarang didirikan menara, oleh karena itu pada daerah atau grid yang didalamnya terdapat makam, diberikan bobot sebesar 10.

Pemberian bobot selanjutnya yaitu pembobotan untuk bangunan militer, bobot untuk bangunan militer diberikan bobot sebesar 5. Hal ini dikarenakan bangunan militer termasuk kedalam zona kawasan peruntukan khusus. Pada zona ini, menara boleh dibangun dengan syarat yaitu harus sesuai dengan ketentuan instansi tersebut. Untuk bobot bangunan industri dan fasilitas umum diberikan bobot sebanyak 2, hal ini dikarenakan bangunan industri masuk kedalam zona industri gudang dan fasilitas umum masuk kedalam zona fasilitas umum. Pada bangunan industri, semua jenis menara diperbolehkan didirikan pada tempat tersebut. Hal ini sama dengan fasilitas umum bukan milik/aset pemerintah juga diperbolehkan untuk didirikan menara. Tetapi bangunan industri dan fasilitas umum mempunyai dimensi masing-masing yang berbeda ukurannya, sehingga diberikan bobot sebesar 2 dengan tujuan menunjukkan bahwa di tempat tersebut terdapat bangunan fasilitas umum dan industri.

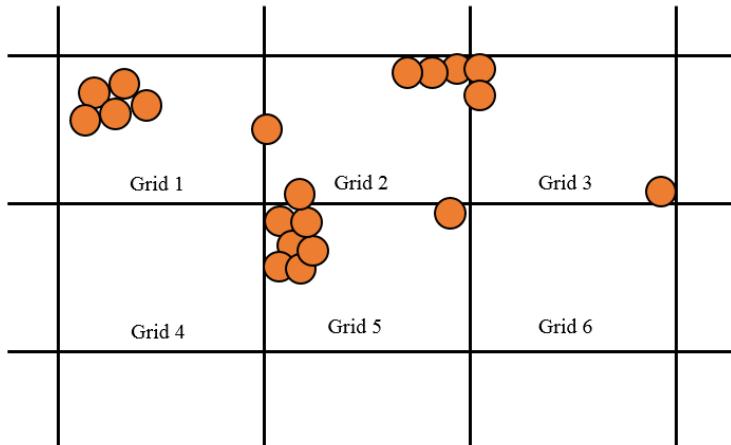
Tabel 3.1 menunjukkan nilai yang digunakan dalam penelitian ini.

Tabel 3.1 Pemberian nilai bobot

Keterangan	Nilai Bobot
Jumlah menara = 0	0
Jumlah menara < 2	1
Jumlah menara ≤ 3	5
Jumlah menara ≥ 4	10
Pemakaman	10
Bangunan Militer	5
Industri	2
Fasilitas Umum	2

Metode tahap ini dimulai dari generate grid, kemudian mencari bobot dalam grid. Rumus mencari bobot sama dengan mencari isi menara dalam grid (3.2)(3.3), yang membedakan yaitu jika grid tersebut terdapat data menara atau data bangunan semua dimasukkan ke dalam perhitungan.

Perbedaan ini terletak pada jumlah atau isi *point* setiap grid. seperti ditunjukkan pada gambar 3.13 bahwa jika didalam grid tersebut terdapat *point* atau titik bangunan, maka di grid tersebut diberi bobot bangunan yang bersangkutan.



Gambar 3.13 Perhitungan bobot untuk setiap grid

Pada gambar 3.13 ditunjukkan hasil dari konversi bangunan dari bentuk *polygon* menjadi bentuk titik/*point*. Pada grid dengan isi titik lebih dari sama dengan satu, maka grid tersebut diberi bobot bangunan tersebut. Hal ini ditunjukkan pada grid 1, grid 2, grid 3 dan grid 5. Pada grid 4 dan 6, tidak terdapat titik didalam grid, sehingga grid tersebut tidak diberi bobot bangunan atau dapat dikatakan bobot nya 0(nol).

Setelah semua bobot terhitung, langkah selanjutnya yaitu menjumlahkan semua bobot tersebut. Jika grid mempunyai bobot ≥ 10 maka grid tersebut dilarang untuk mendirikan menara. Jika grid mempunyai bobot antara 5 sampai dengan 10, maka grid tersebut diperbolehkan untuk mendirikan menara dengan syarat tertentu. Jika grid mempunyai bobot dibawah 5, maka grid tersebut diperbolehkan untuk mendirikan menara.

3.6 Metode Pencarian Lokasi Rekomendasi

Metode pencarian lokasi rekomendasi digunakan untuk memberi pilihan pada operator jika koordinat usulan masuk pada grid yang memiliki bobot lebih dari 10. Gambar 3.14 merupakan ilustrasi dari grid rekomendasi tersebut. X merupakan lokasi usulan pihak operator dan $R_1 - R_8$ merupakan grid rekomendasi yang tedapat pada sekitar koordinat usulan.

R ₁	R ₂	R ₃
R ₄	x	R ₅
R ₆	R ₇	R ₈

Gambar 3.14 Gambaran grid rekomendasi

3.7 Spesifikasi Kawasan Pendirian Menara

Usulan pendirian menara tidak terbatas hanya pada usulan operator, melainkan pemerintah berhak untuk mengajukan lokasi atau kawasan yang dapat didirikan menara baru. kawasan yang berhak ditempati tersebut mempunyai spesifikasi seperti pada tabel 3.2.

Tabel 3.2 Spesifikasi kawasan yang dapat didirikan menara baru

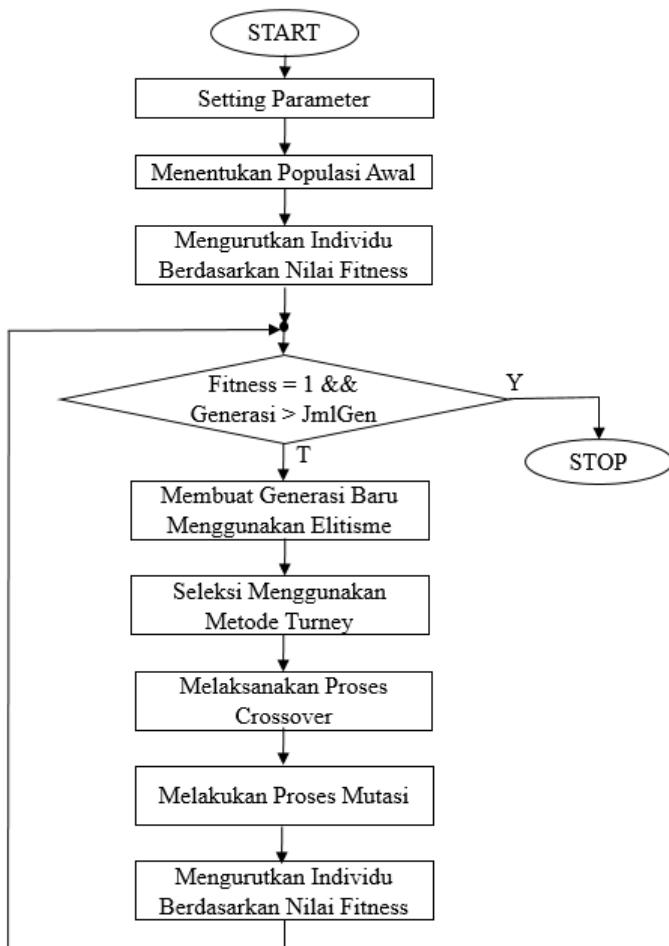
Spesifikasi	Target
Menara	< 4
Menara tetangga	0
Bangunan Makam	Tidak ada
Bangunan Militer	Tidak ada

Pada tabel diatas, spesifikasi menara target kurang dari 4 dikarenakan calon menara diusahakan dapat memenuhi kebutuhan jaringan pada kawasan yang minim menara. Selain kawasan minim menara, target berikutnya yaitu kawasan yang disekelilingnya belum terdapat menara sama sekali. Selain tidak terdapat menara, target yang harus dicapai yaitu pada kawasan tersebut tidak boleh terdapat bangunan makam dan militer. Pada rancangan ini, penulis menggunakan algoritma cerdas untuk memecahkan masalah diatas. Algoritma cerdas yang digunakan adalah algoritma genetika.

Algoritma genetika ditunjukkan untuk memecahkan masalah yang ada. Spesifikasi yang telah dibuat dirumuskan menjadi permasalahan yang harus dipecahkan oleh GA. Hasil dari algoritma GA yaitu mencari solusi dari permasalahan yang telah dirumuskan. Solusi yang diharapkan yaitu menara diletakkan pada posisi dimana hanya terdapat satu menara eksisting didalamnya.

3.8 Metode Pencarian Kawasan Pendirian Menara Menggunakan Algoritma Genetika

Algoritma genetika digunakan untuk memecahkan masalah pada tabel 3.2. algoritma ini dimulai dari setting parameter awal yang akan digunakan. Setelah parameter didapat, langkah selanjutnya yaitu menentukan populasi awal. Pada populasi terdapat individu yang mempunyai nilai *fitness* masing-masing. Nilai *fitness* tersebut diurutkan dari yang terkecil hingga yang terbesar. Setelah itu akan dicek apakah nilai *fitness* sesuai dengan yang diharapkan atau belum, jika belum maka proses akan berlanjut melalui proses *elitisme*, seleksi, *crossover*, mutasi sampai dengan pengurutan nilai *fitness*. Proses ini akan berakhir ketika kriteria pemberhentian terpenuhi. Proses diatas dijelaskan melalui flowchart pada gambar 3.15.



Gambar 3.15 Rancangan Algoritma Genetika

3.8.1 Setting Parameter GA

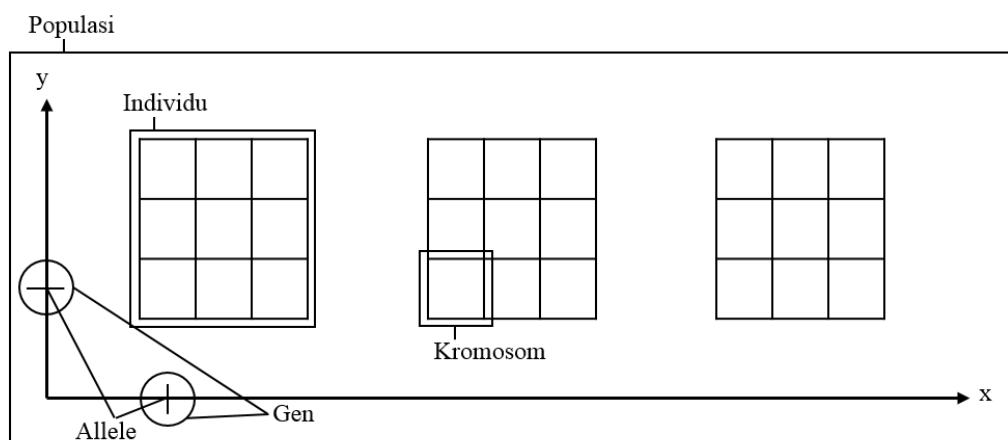
Berikut tabel 3.3 merupakan parameter GA yang digunakan pada penelitian ini.

Tabel 3.3 Parameter GA

No	Parameter GA	Target Kawasan	Jumlah
1	Jumlah Populasi	Kawasan 3x3	3
2	Jumlah <i>Elitisme</i>	Kawasan terbaik	1
3	Jumlah Seleksi	Penemuan kawasan baru	3
4	Probabilitas <i>Crossover</i>	Probabilitas kawin silang antara kawasan satu dengan kawasan yang lain	0,75
5	Probabilitas Mutasi	Probabilitas ditemukannya kawasan baru	0,75

3.8.2 Menentukan Populasi Awal

Populasi terdiri dari *kromosom-kromosom* yang mewakili individu, individu-individu tersebut tergabung menjadi satu menjadi populasi awal. Populasi awal didapat dari pembangkitan data individu secara random sebanyak parameter yang telah ditentukan. Didalam individu terdapat *kromosom*, *gen* dan *allele* seperti ditunjukkan pada gambar 3.16.



Gambar 3.16 Populasi, Individu, Kromosom, Gen dan Allele

3.8.3 Mengurutkan Individu Berdasarkan Nilai *Fitness*

Setiap individu yang terdapat pada populasi mempunyai nilai *fitness* masing-masing. Nilai *fitness* atau fungsi evaluasi pada proses ini merupakan

pemberian pinalti atau pembobotan pada masalah yang ada. Pembobotan dilakukan untuk menentukan wilayah kawasan yang dilarang untuk mendirikan menara. Tabel 3.4 merupakan tabel pembobotan pada setiap kawasan/grid.

Tabel 3.4 Tabel pembobotan kawasan

No	Kawasan / Grid	Bobot
1	Menara <i>existing</i> pada titik tengah individu /target kawasan lebih dari 1	10000
2	Target kawasan berada pada sisi samping luar area pencarian	1000
3	Menara <i>existing</i> pada target kawasan lebih dari samadengan 4	10
4	Menara <i>existing</i> pada target kawasan berjumlah 2 atau 3	5
5	Menara <i>existing</i> berisi 1 menara	1
6	Tidak terdapat menara <i>existing</i>	0
7	Terdapat bangunan militer	5
8	Terdapat bangunan pemakaman	8
9	Terdapat bangunan industri	2
10	Terdapat bangunan fasilitas umum	2

Pada tahap ini, nilai *fitness* yang dicari yaitu nilai minimum pada total bobot suatu individu. Total bobot yang dicari ditunjukkan pada gambar 3.17.

C ₁	C ₂	C ₃
C ₄	C ₅	C ₆
C ₇	C ₈	C ₉

Gambar 3.17 Bobot individu yang dicari

Berdasarkan gambar diatas, bobot yang dicari mulai dari bobot C₁ sampai dengan C₈, sehingga rumus nilai fitness nya :

$$\begin{aligned}
 F(C) &= \sum C_i \\
 &= C_1 + C_2 + \dots + C_8
 \end{aligned}$$

Dimana:

C = nilai bobot masing-masing kawasan

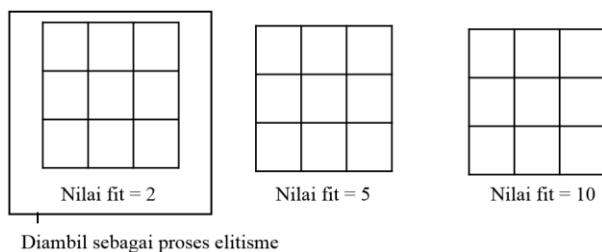
3.8.4 Kriteria Pemberhentian

Kriteria pemberhentian yaitu ketika nilai *fitness* sudah terpenuhi atau jumlah generasi yang telah terjadi melebihi nilai iterasi yang telah ditentukan. Nilai *fitness* yang ditetapkan pada algoritma ini yaitu 1, dan banyaknya iterasi yang digunakan sebanyak 10000 generasi.

Nilai *fitness* yang terpenuhi harus 1, karena peletakan menara disarankan berada pada kawasan yang mempunyai 1 menara *existing*. Pada saat proses generasi berakhir, apabila tidak ditemukan kriteria individu yang diharapkan (berupa nilai *fitness* 1), maka individu yang diambil adalah individu yang mempunyai nilai *fitness* terendah. Jika kriteria pemberhentian belum menemui kedua syarat, maka algoritma berlanjut ke proses berikutnya.

3.8.5 Membuat Generasi Baru Menggunakan Elitisme

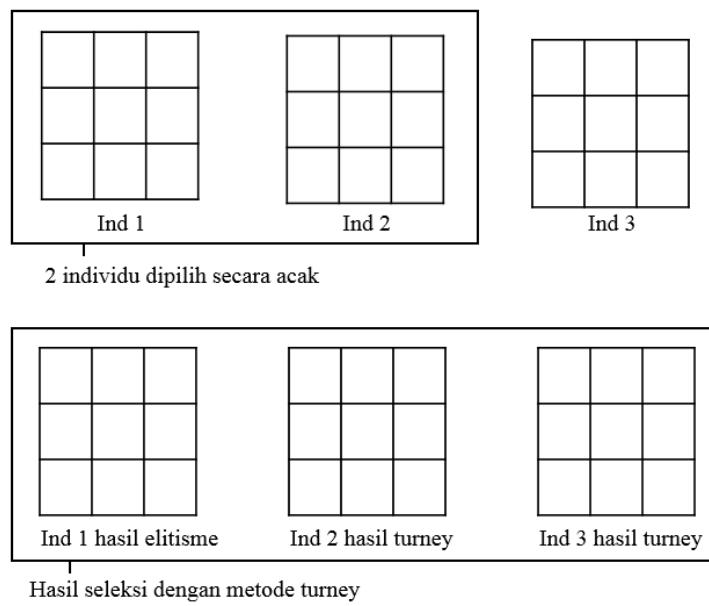
Pembuatan generasi baru ini bertujuan untuk membentuk populasi baru yang nantinya akan menggantikan populasi lama. Tetapi, pada proses ini nilai terbaik pada populasi lama dipertahankan. Hal ini sesuai dengan konsep *elitisme* yaitu mempertahankan individu terbaik dari individu yang ada. Pengambilan individu pada proses elitisme hanya mengambil 1 individu dengan nilai *fitness* minimum, hal ini ditunjukkan pada gambar 3.18.



Gambar 3.18 Individu yang diambil pada proses *elitisme*

3.8.6 Seleksi Menggunakan Metode *Turney*

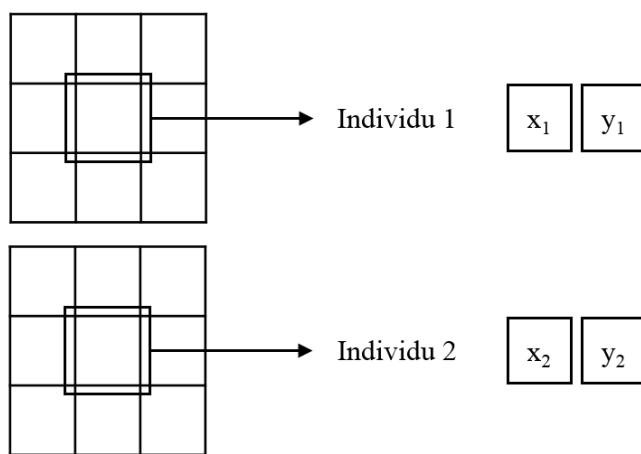
Pada proses ini terdapat proses perbandingan antara 2 individu hasil yang dipilih secara random dan dipilih nilai *fitness* minimumnya. Hasil dari proses *turney* ini digabungkan dengan hasil dari proses *elitisme* yang telah dilakukan. Proses ini berhenti ketika jumlah seleksi terpenuhi. Proses ini ditunjukkan pada gambar 3.19.



Gambar 3.19 Proses seleksi menggunakan metode *turney*

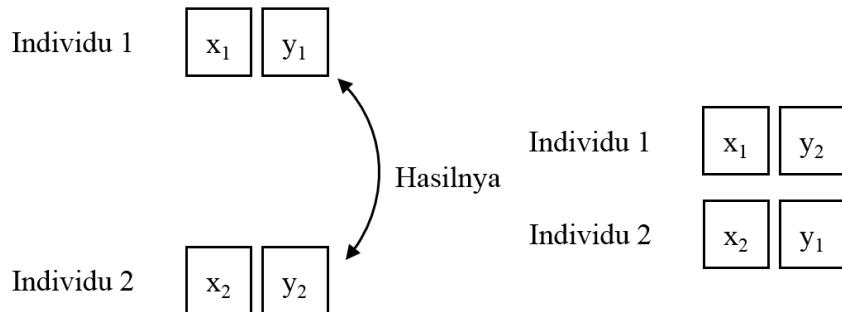
3.8.7 Proses *Crossover*

Pada proses *crossover*, hasil dari seleksi dipilih 2 secara random untuk dijadikan sebagai induk. Parameter *crossover* yang telah ditetapkan sebelumnya sebesar 0.75, sehingga jika nilai probabilitas *crossover* terpenuhi, maka proses *crossover* akan dilakukan. Proses ini melakukan *crossover* atau kawin silang dari 2 individu yang telah terpilih. Masing-masing individu memuat *gen x* dan *gen y*, tetapi *crossover* hanya dilakukan pada titik tengah individu seperti ditunjukkan pada gambar 3.20.



Gambar 3.20 *Gen x* dan *y* pada masing-masing individu

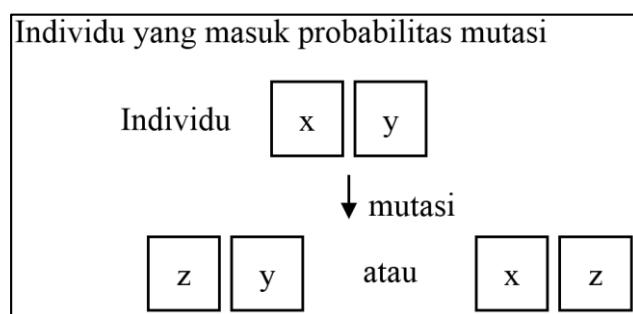
Pada proses *crossover* yang digunakan kali ini yaitu meng-kawin silangkan antar *gen y*, sehingga *gen y₁* akan bertukar ke *gen y₂*. Proses ini ditunjukkan pada gambar 3.21.



Gambar 3.21 Proses *crossover gen y₁* dan *y₂*

3.8.8 Proses Mutasi

Proses *crossover* menghasilkan 2 individu baru yang mempunyai sifat sama seperti induknya. Proses selanjutnya yaitu proses mutasi. Proses ini mengambil masing-masing individu dari hasil *crossover* dengan probabilitas 0,75. Proses ini, individu mengalami perubahan *gen x* atau *y* yang menghasilkan individu baru yang berbeda dari induk sebelumnya. Perubahan *x* dan *y* mempunyai perbandingan probabilitas sama yaitu 50:50, sehingga mutasi dapat terjadi pada *gen x* saja atau pada *gen y* saja. Proses ini ditunjukkan pada gambar 3.22.



Gambar 3.22 Proses mutasi

Dimana

z = nilai gen baru

Proses seleksi sampai dengan mutasi menghasilkan 3 individu baru. Individu tersebut berupa hasil dari *elitisme*, hasil dari mutasi pertama dan hasil dari

mutasi kedua. Individu-individu ini membentuk populasi baru yang menggantikan populasi lama. Populasi baru ini kemudian diurutkan sesuai proses pengurutan individu.

3.9 Skenario Pengujian Algoritma

Dalam skenario ini, pengujian dilakukan untuk melakukan tes ketepatan pembobotan dan ketepatan memberikan saran atau rekomendasi untuk algoritma konvensional yang telah dibuat. Pengujian ini dilakukan 3 kali, yaitu:

1. Memasukkan koordinat (lokasi penempatan menara) yang disekitarnya terdapat menara kurang dari 4.
2. Memasukkan koordinat di area daerah khusus, daerah yang terdapat makam dan bangunan militer.
3. Memasukkan koordinat (lokasi penempatan menara) yang disekitarnya terdapat menara lebih dari 4.

Dalam skenario pengujian algoritma cerdas (GA), algoritma diuji keandalannya. Pengujian dilakukan dengan cara melakukan *iterasi* atau proses generate generasi baru sebanyak 10000 dan algoritma dijalankan sebanyak 30 kali. Analisa keandalan dilakukan dengan presentase keberhasilan menemukan nilai *fitness* berbanding dengan banyaknya pengujian. Berikut rumus keandalan

$$P = \frac{T}{N} \times 100\%$$

Dimana :

P = persentase hasil pengujian

T = total nilai benar

N = banyaknya N perulangan

Berdasarkan persentase hasil pengujian tersebut selanjutnya dilakukan penarikan kesimpulan untuk memberikan nilai kelayakan mengenai algoritma genetika yang telah disusun. Besar persentase kelayakan seperti pada tabel 3.5

Tabel 3.5 Persentase Kelayakan [12]

Percentase (%)	Keterangan
76 – 100	Layak
51 – 75	Cukup Layak
26 – 50	Kurang Layak
< 25	Tidak Layak

Adapun keterangan dari tabel 3.5 diatas seperti berikut

- a. Apabila algoritma yang diuji coba mencapai hasil persentase 76% sampai dengan 100%, maka algoritma yang telah dibuat tergolong masuk kedalam kualifikasi layak serta tidak perlu direvisi.
- b. Apabila algoritma yang diuji coba mencapai hasil persentase 51% sampai dengan 75%, maka algoritma tersebut tergolong kualifikasi cukup layak dan memungkinkan revisi dalam algoritma yang telah disusun.
- c. Apabila algoritma yang diuji coba mencapai hasil persentase 26% sampai dengan 50%, maka algoritma yang telah dibuat tergolong masuk ke dalam kualifikasi kurang layak dan memerlukan revisi cukup banyak pada bagian algoritmanya.
- d. Apabila algoritma yang diuji coba mencapai hasil persentase kurang dari 25%, maka algoritma yang telah dibuat tergolong masuk ke dalam kualifikasi tidak layak dan wajib untuk merevisi algoritma secara keseluruhan.

BAB 4

HASIL DAN PEMBAHASAN

Pada bagian ini, akan diuraikan mengenai hasil algoritma yang telah disusun. Algoritma tahap awal ditulis menggunakan bahasa PHP dan dijalankan menggunakan bantuan XAMPP. Selanjutnya yaitu mengkonversi data bangunan ke data koordinat agar dapat dilakukan perhitungan dan pembobotan. Langkah berikutnya yaitu menjalankan skenario pengujian berdasarkan algoritma cerdas yang telah disusun. Tujuan utama dari skenario ini yaitu memberikan hasil dari koordinat usulan, hasil tersebut dapat berupa persetujuan atau penolakan. Jika usulan koordinat ditolak, maka akan diberikan rekomendasi terkait lokasi yang diperbolehkan untuk mendirikan menara.

4.1 Hasil Pencarian Grid Optimal berdasarkan Posisi Menara

Hasil dari pencarian ini yaitu grid yang memiliki jumlah menara paling banyak di dalam nya. Optimasi dilakukan dengan mencari nilai kwadrat dari menara di dalam masing-masing grid, dan dipilih nilai yang paling tinggi.

Proses algoritma dimulai dari memasukkan file excel yang berisikan koordinat menara. Proses selanjutnya seperti metode pada bab 3, yaitu membuat grid, dan menghitung isi dalam grid tersebut. Pada pengujian algoritma yang pertama ini, penulis membuat perulangan sebanyak 16 kali dan jarak grid nya sebesar 200 meter. Tabel 4.1 dibawah ini merupakan hasil dari algoritma yang telah dijalankan.

Tabel 4.1 Hasil algoritma awal

Id	LBH	Tower	JML
1	23	102	476
2	22	97	447
3	21	90	396
4	24	104	468
5	21	93	429
6	20	89	413
7	18	81	381
8	19	85	397

Id	LBH	Tower	JML
9	18	81	381
10	17	78	374
11	20	89	409
12	22	98	450
13	23	103	487
14	25	112	528
15	21	94	444
16	23	102	476

Algoritma yang telah dijalankan menghasilkan grid yang optimal pada pergeseran ke 14. Hal ini dapat dikatakan optimal karena pada pergeseran tersebut nilai *JML*, *Tower*, dan *LBH* merupakan nilai tertinggi. Pada kasus ini, pergeseran ke 14 dapat dikatakan bahwa pergeseran tersebut merupakan “*Global Optimum*”. Dapat dikatakan global optimum karena pergeseran tersebut mempunyai nilai tertinggi pada setiap hasilnya. Pada kasus yang lain, jika terdapat jumlah *Tower* yang sama, hal mendasar yang dijadikan acuan optimasi yaitu nilai *JML*, karena nilai tersebut merupakan hasil kuadrat dari isi *LBH*.

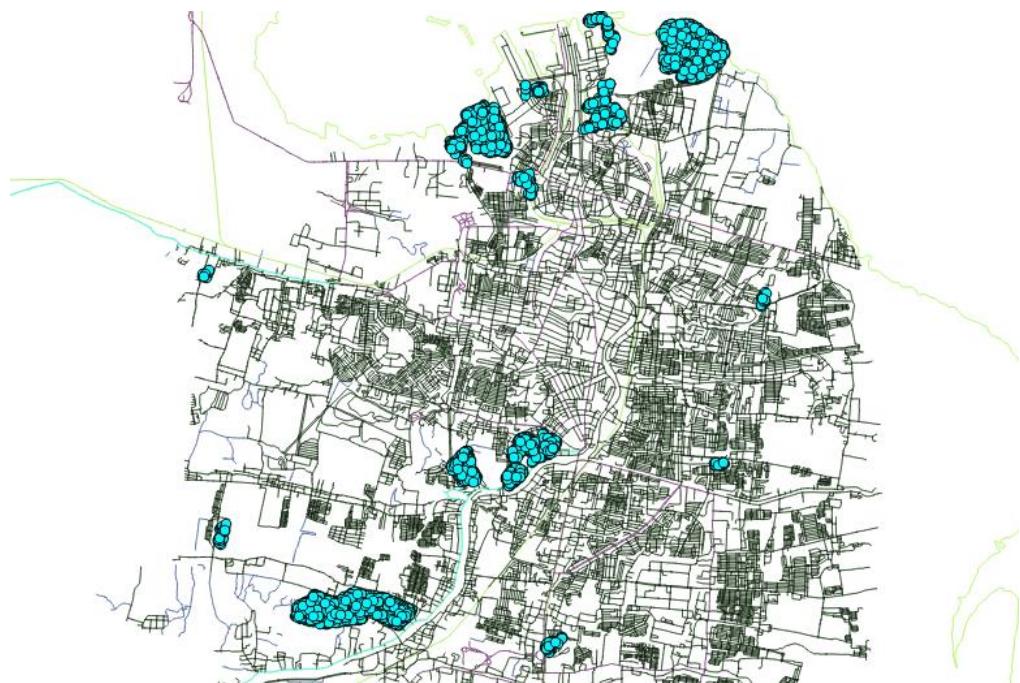
Hasil optimum yang didapatkan, dapat menghemat waktu pencarian. Hal ini didasari dari jumlah grid yang didalamnya terdapat menara lebih dari 4. Total grid sewaktu pembuatan yaitu 9860 grid. jika nilai tersebut dikurangi nilai *LBH* maka, $9860 - 25 = 9835$. Sehingga program hanya akan mencari sisa grid sebanyak 9835 grid atau dapat dikatakan hal ini dapat menghemat waktu pencarian sebesar 0,25%.

4.2 Konversi Data Spasial Bangunan Menjadi Koordinat

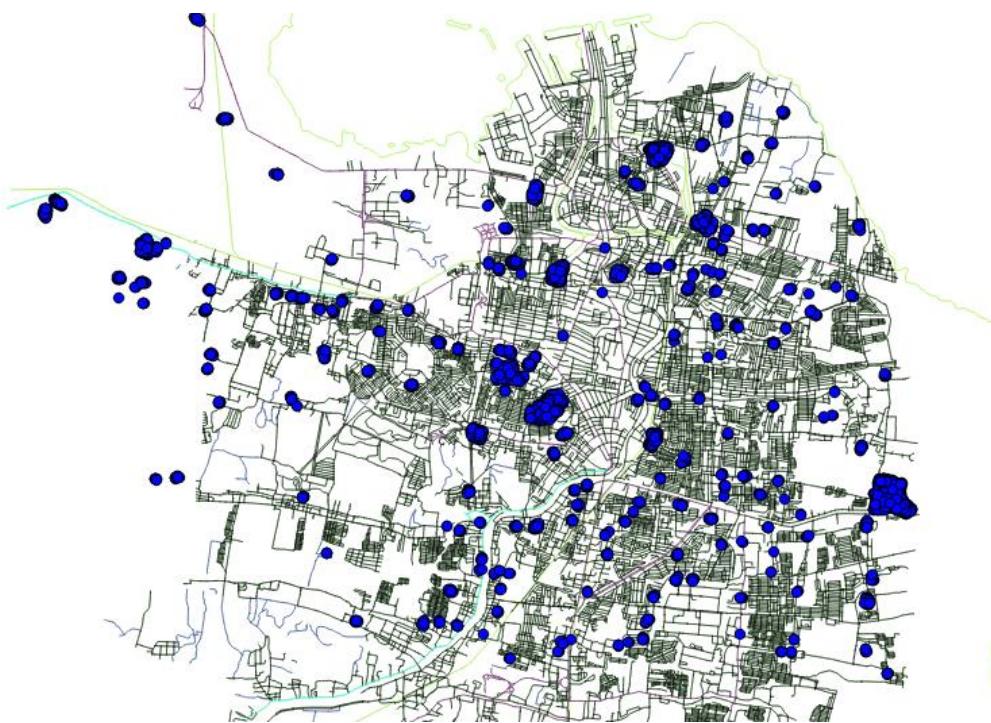
Sebelum masuk kebagian pembobotan, data bangunan perlu dirubah menjadi titik koordinat. Hal ini dilakukan untuk mempermudah perhitungan pada sisi algoritmanya. Data bangunan sebelumnya berupa data spasial type *polygon*, cara merubah menjadi titik koordinat yaitu dengan bantuan ”*QGIS geoalgorithm*” yaitu tools *random point in layer bound*.

Pada tools tersebut terdapat beberapa kolom yang harus diisi. Kolom pertama yaitu input layer diisi dengan layer salah satu bangunan yang bertipe

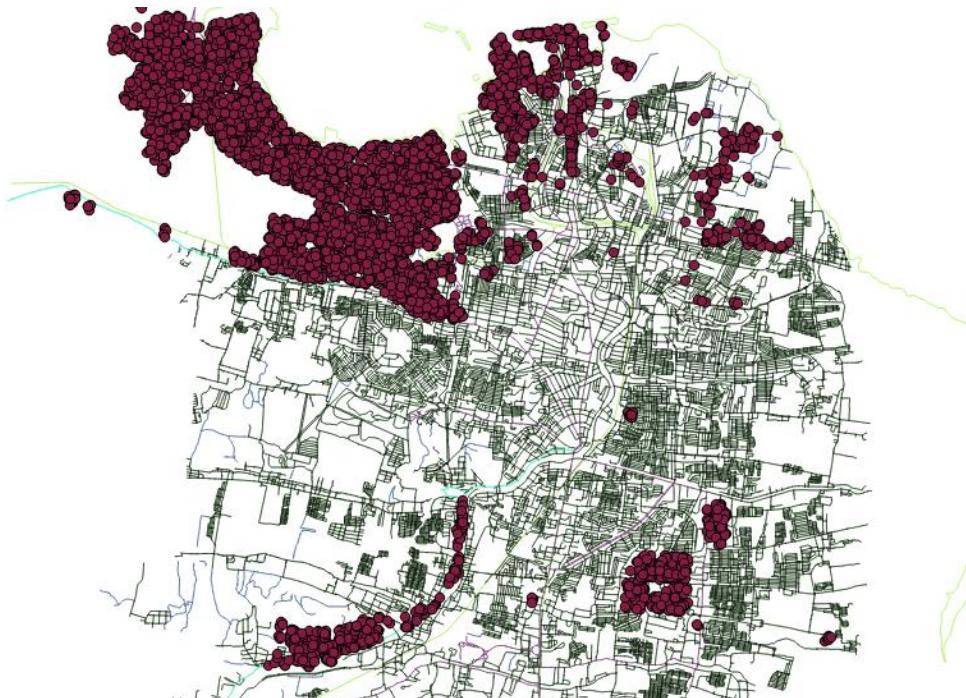
polygon. Pada kolom kedua, terdapat jumlah point number yang harus diisi. Pada penelitian ini, penulis menggunakan jumlah point sebanyak 5000 untuk bangunan militer, makam dan industri. Sedangkan untuk bangunan fasilitas umum, penulis menggunakan point sebanyak 15000 dikarenakan jumlah *polygon* yang banyak dan menyebar pada seluruh peta. Gambar 4.1 merupakan hasil dari *geo algorithm* yang telah dijalankan untuk koordinat atau layer militer. Gambar 4.2 merupakan hasil dari *geo algorithm* yang telah dijalankan untuk koordinat atau layer makam. Gambar 4.3 merupakan hasil dari *geo algorithm* yang telah dijalankan untuk koordinat atau layer industri. Gambar 4.4 merupakan hasil dari *geo algorithm* yang telah dijalankan untuk koordinat atau layer fasilitas umum.



Gambar 4.1 Koordinat *point* militer



Gambar 4.2 Koordinat *point* makam



Gambar 4.3 Koordinat *point* industri



Gambar 4.4 Koordinat *point* fasilitas umum

4.3 Hasil Pencarian Grid Optimal berdasarkan Bobot

Pada metodologi telah dijelaskan mengenai nilai bobot untuk masing-masing data bangunan. Metode yang digunakan pada algoritma tahap ini hampir sama dengan algoritma tahap awal. Input tersebut berupa input data-data bangunan yang telah dijadikan titik koordinat, perulangan yang akan dilakukan dan jarak masing-masing grid dalam meter. Pada tampilan tersebut terdapat *checkbox* berupa rubah data, hal ini digunakan untuk memaksimalkan penelitian. Data bangunan yang telah diinputkan, dimasukkan ke dalam database tersendiri. Sehingga jika penulis ingin mengganti nilai perulangan, penulis tidak perlu merubah data bangunan yang sudah ada.

Tabel 4.2 Hasil algoritma yang ditambahkan parameter bobot

Id	GNA	GA	TotalGrid
1	182	9763	9945
2	183	9762	9945
3	180	9765	9945
4	190	9755	9945
5	179	9766	9945
6	173	9772	9945

Id	GNA	GA	TotalGrid
7	173	9772	9945
8	174	9771	9945
9	171	9774	9945
10	173	9772	9945
11	183	9677	9860
12	189	9671	9860
13	188	9672	9860
14	187	9673	9860
15	177	9683	9860
16	182	9678	9860

Tabel 4.2 merupakan hasil dari algoritma yang telah ditambahkan dengan nilai bobot. Pada gambar tersebut terdapat *id*, *GNA*, *GA* dan total *grid*, *id* merupakan keterangan dari pergeseran atau perulangan yang telah dilakukan, *GNA* merupakan *grid not available* yang berarti grid yang tidak boleh ditempati atau tidak boleh mendirikan menara. *GNA* merupakan grid dengan nilai bobot lebih dari 10. *GA* merupakan *grid available* yang berarti grid boleh ditempati. Yang terakhir yaitu *total grid* merupakan total dari semua grid yang telah dicari atau dihitung. Hasil tersebut menggunakan pergeseran sebanyak 16 kali dengan jarak masing-masing grid sebesar 200 meter.

Dari algoritma tersebut menghasilkan bobot paling banyak terdapat pada pergeseran ke 4. Sehingga pergeseran ini menjadi pergeseran optimal atau grid optimal. Berikut ini merupakan batas dari grid optimal tersebut,

$$\begin{aligned}
 latawal &= -7.19601 \\
 latakhir &= -7.35028 \\
 longawal &= 112.6024775 \\
 longakhir &= 112.8138
 \end{aligned}$$

4.4 Hasil Skenario Pengujian

Terdapat 3 skenario pengujian yang telah disusun pada bab sebelumnya. Skenario ini diujikan pada algoritma yang telah diberi bobot. Pada skenario pertama, posisi koordinat yang digunakan yaitu -7.273480, 112.730371.

Hasil skenario pertama yaitu menara boleh dibangun pada grid tersebut. Hal ini dikarenakan pada grid atau pada lokasi tersebut menara eksisting berjumlah kurang dari 4, dan pada lokasi tersebut tidak terdapat bangunan yang memerlukan perlakuan khusus.

Pada skenario kedua, posisi koordinat yang digunakan yaitu -7.334150, 112.695293. Hasil skenario kedua yaitu menara tidak boleh dibangun pada grid atau pada lokasi tersebut. Hal ini dikarenakan pada grid tersebut terdapat bangunan dan total bobot dari grid tersebut diatas 10. Oleh karena itu, terdapat rekomendasi agar menara digeser sesuai dengan tabel 4.3.

Tabel 4.3 Rekomendasi koordinat skenario kedua

Min	Max	Min	Max	Jumlah
Latitude	Latitude	Longitude	Longitude	tower
-7.33101	-7.33281	112.6924775	112.6942775	0
-7.33101	-7.33281	112.6942775	112.6960775	1
-7.33101	-7.33281	112.6960775	112.6978775	0
-7.33281	-7.33461	112.6924775	112.6942775	0
-7.33281	-7.33461	112.6960775	112.6978775	0
-7.33461	-7.33641	112.6924775	112.6942775	0
-7.33461	-7.33641	112.6942775	112.6960775	0
-7.33461	-7.33641	112.6960775	112.6978775	0

Hasil skenario terakhir dipilih titik koordinat -7.270155, 112.756425. Pada skenario ketiga, dipilih koordinat yang terdapat menara berjumlah 4 atau lebih didalam nya. Dari skenario tersebut menghasilkan bahwa koordinat yang dipilih merupakan lokasi atau tempat yang dilarang untuk mendirikan menara, dan diberikan rekomendasi untuk menggeser lokasi menara sesuai rekomendasi pada tabel 4.4.

Tabel 4.4 Rekomendasi koordinat skenario ketiga

Min	Max	Min	Max	Jumlah
Latitude	Latitude	Longitude	Longitude	tower
-7.26801	-7.26981	112.7536775	112.7554775	0
-7.26801	-7.26981	112.7554775	112.7572775	0
-7.26801	-7.26981	112.7572775	112.7590775	0
-7.26981	-7.27161	112.7536775	112.7554775	0

<i>Min</i>	<i>Max</i>	<i>Min</i>	<i>Max</i>	<i>Jumlah</i>
<i>Latitude</i>	<i>Latitude</i>	<i>Longitude</i>	<i>Longitude</i>	<i>tower</i>
-7.26981	-7.27161	112.7572775	112.7590775	0
-7.27161	-7.27341	112.7536775	112.7554775	0
-7.27161	-7.27341	112.7554775	112.7572775	0
-7.27161	-7.27341	112.7572775	112.7590775	1

4.5 Hasil Pengujian Algoritma Cerdas

Pengujian dilakukan dengan iterasi sebanyak 10000 kali dan dilakukan dalam 30 kali percobaan. Parameter yang disetting sesui dengan parameter pada tabel 3.3. Tabel 4.5 merupakan hasil dari algoritma yang telah dijalankan.

Tabel 4.5 Hasil algoritma GA

RUN	ROW	COL	ITERASI	Neighbour Weight
1	74	113	1968	0
2	56	18	519	0
3	62	43	787	0
4	25	13	6027	0
5	58	53	7061	0
6	58	53	3265	0
7	62	43	504	0
8	42	18	1509	0
9	24	97	1822	0
10	42	18	7032	0
11	72	35	1773	0
12	30	23	2395	0
13	58	53	2640	0
14	55	44	6267	0
15	24	97	53	0
16	58	24	3125	0
17	56	18	356	0
18	72	35	1611	0
19	55	42	5812	0
20	24	97	4799	0
21	42	18	0	0
22	62	43	390	0
23	55	42	7579	0
24	58	53	7723	0

RUN	ROW	COL	ITERASI	Neighbour Weight
25	42	18	7900	0
26	74	113	350	0
27	30	23	259	0
28	58	24	354	0
29	62	43	4619	0
30	53	47	10000	1

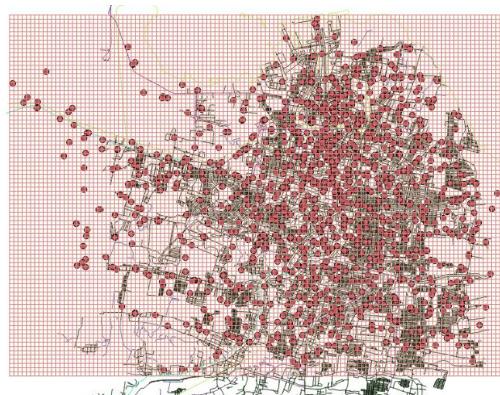
Dari hasil pengujian sebanyak 30 kali, didapatkan pada pengujian ke 30 nilai *fitness* minimum tidak terpenuhi. Karena batas iterasi sudah terlampaui, maka nilai yang diambil adalah nilai *fitness* terkecil. Persentase kehandalan untuk algoritma ini sebesar

$$P = \frac{29}{30} \times 100\% = 96,6\%$$

Nilai persentase kehandalan yang diperoleh sebesar 96,6%, sehingga algoritma dinyatakan layak untuk menyelesaikan masalah terkait penempatan menara bts baru.

4.6 Visualisasi Hasil Optimasi

Pada subbab 4.3 didapatkan hasil berupa batas-batas latitude dan longitude. Hasil tersebut dimasukkan kedalam aplikasi *quantum GIS*, yang nantinya dapat digunakan sebagai media visualisasi menara dan data spasial kota. Hal ini ditunjukkan pada gambar 4.5.



Gambar 4.5 Hasil generate grid untuk visualisasi

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Berdasarkan tujuan, hasil dan analisa dari penelitian yang telah dilakukan, disimpulkan bahwa :

1. Pembuatan algoritma baru berdasarkan konsep *greedy remove* telah dibuat. Hasil dari algoritma ini yaitu algoritma dapat mengolah data spasial dengan hasil grid optimum didapatkan pada batas *latitude* awal -7.196101, batas *latitude* akhir -7.35028 dan batas *longitude* awal 112.6024775, batas *longitude* akhir 112.8138.
2. Pengembangan algoritma cerdas menggunakan algoritma genetika dijalankan sebanyak 30 kali. Menghasilkan nilai keberhasilan sebanyak 29 kali dan gagal sebanyak 1 kali, sehingga nilai kehandalan diperoleh sebesar 96%. Sehingga algoritma genetika dinyatakan layak untuk menyelesaikan masalah terkait penempatan menara BTS baru.
3. Pengembangan sistem visualisasi spasial menggunakan aplikasi Quantum GIS. Hasil optimasi berupa batas-batas *latitude* dan *longitude* digunakan untuk menggenerate grid. Hal ini dapat digunakan sebagai media *visulisasi* posisi menara dan data spasial kota.

5.2 Saran

Pada penelitian selanjutnya terkait optimasi penempatan menara BTS dan MCP, dapat dilakukan penambahan atau pengembangan seperti berikut:

1. Penambahan parameter data bangunan, seperti taman, lapangan olah raga dan ruang terbuka hijau.
2. Penambahan parameter pada menara seperti ketinggian menara dan perhitungan *coverage* menara.
3. Penambahan target wilayah untuk dijadikan sebagai solusi pada algoritma genetika.

Halaman ini sengaja dikosongkan

DAFTAR PUSTAKA

- [1] Pemerintah Daerah Kota Surabaya, "Peraturan Daerah Nomor 05 Tahun 2013", Surabaya, 2013.
- [2] Menteri Komunikasi dan Informatika, "Pedoman Pembangunan dan Penggunaan Menara Bersama Telekomunikasi", vol. 02/PER/M.KOMINFO/3/2008, 2008.
- [3] WALIKOTA SURABAYA, "Pedoman Pengendalian Menara Telekomunikasi di Kota Surabaya", vol. 48 Tahun 2017, Surabaya, 2017.
- [4] G. Kalata, I. Pozniak-Koszalka, L. Koszalka and A. Kasprzak, "Optimising base station location for UMTS cellular networks," *J. Phys. Conf. Ser.*, vol. 570, no. 62005, p. 6, 2014.
- [5] H. K. Ali, H. D. Muhammed and J. S. Abdaljabar, "Geographic Information System (Gis) Spatial Analyst Techniques a Reference for Determining the Position of Cellular Systems," *European Scientific Journal*, Vols. 11, No. 18, p. 174–189, 2015.
- [6] C. O. Alenoghena, J. O. Emagbetere and A. M. Aibinu, "Artificial Intelligence based technique for BTS placement," in *5th International Conference on Mechatronics (ICOM' 13)*, 2013.
- [7] E. Prahasta, Sistem Informasi Geografis, Bandung: CV Informatika, 2005.
- [8] M. Neteler and H. Mitasova, OPEN SOURCE GIS: AGRASS GIS Approach, New York: Springer, 2008.
- [9] J. Bossler, J. Jensen, R. McMaster and C. Rizos, Manual of Geospatial Science and Technology, London: CRC Press, 2004.
- [10] A. S. Soedomo, Sistem dan Transformasi Koordinat, Bandung: ITB, 2004.
- [11] Walikota Surabaya Provinsi Jawa Timur, Rencana Tata Ruang Wilayah Kota Surabaya Tahun 2014-2034, vol. 12, Surabaya, 2014.
- [12] S. Arikunto, Prosedur Penelitian Suatu Pendekatan Praktik, Jakarta: Rineka Cipta, 2010.

Halaman ini sengaja dikosongkan

LAMPIRAN

Algoritma tahap pertama

```
<?php
error_reporting(0);
//koneksi ke database, username,password dan namadatabase menyesuaikan
include "db.php";
$time_start = microtime(true);
ini_set("max_execution_time", 0);
require "excel_reader.php";
=====

===== header awal
//Input file menara dulu
echo "Input excel dan input berapa banyak iterasi yang dibutuhkan";
echo "<br>";
echo "<br>";
?>
<!-- tambahkan inputan untuk excel -->
<form name="input" action="paket1.php" method="post" onSubmit="return validateForm()"
enctype="multipart/form-data">
    input file excel : <input type="file" id="fileexcel" name="fileexcel" /><br>
    berapa perulangan :<input type="text" name="iterasi"><br>
    berapa jarak grid nya :<input type="text" name="jarak"> meter <br>
    <input type="submit" name="submit" value="Submit">
</form>

<script type="text/javascript">
// validasi form (hanya file .xls yang diijinkan)
function validateForm()
{
    function hasExtension(inputID, exts) {
        var fileName = document.getElementById(inputID).value;
        return (new RegExp('(' + exts.join('|').replace(/\./g, '\\.') + ')$')).test(fileName);
    }

    if(!hasExtension('fileexcel', ['.xls'])){
        alert("Hanya file XLS (Excel 2003) yang diijinkan.");
        return false;
    }
}
</script>
```

```

<?php
    // hapus file xls yang udah dibaca
    //unlink($_FILES['fileexcel']['name']);

    // input excel
    // excel masuk ke database, masuk ke database t_dataawal, database ini merupakan patokan awal semuanya.

    // kalau normal, dari sini data diurutkan untuk membuat grid, maka data ini juga harus diurutkan terlebih dahulu

    // setelah data urut masuk ke dalam database t_laturut dan t_longurut, maka data baru di modifikasi!!

    if(isset($_POST['submit'])){
        //cek paketkompliv2 untuk tahu mana yang harus di create, dan mana yang harus di delete.

        $deltbldb=mysql_query("SHOW TABLES FROM tesisv3");

        while($dataassoc=mysql_fetch_assoc($deltbldb)){
            mysql_query("DROP TABLE ".$dataassoc["Tables_in_"].thesisv3);
        }

        //=====
        ===== buat database
        // database -> t_dataawal , t_laturut , t_longurut
        $dbtblnew1 = "CREATE TABLE `thesisv3`.`t_dataawal` ( `id` INT NOT NULL AUTO_INCREMENT ,`latitude` DOUBLE NOT NULL ,`longitude` DOUBLE NOT NULL ,PRIMARY KEY (`id`)) ENGINE = InnoDB";
        $qdbtblnew1 = mysql_query($dbtblnew1);

        //=====
        =====
        //=====

        $target = basename($_FILES['fileexcel']['name']);
        move_uploaded_file($_FILES['fileexcel']['tmp_name'], $target);

        $data = new Spreadsheet_Excel_Reader($_FILES['fileexcel']['name'],false);

        // menghitung jumlah baris file xls
        $baris = $data->rowcount($sheet_index=0);

        // import data excel mulai baris ke-2 (karena tabel xls ada header pada baris 1)
        for ($i=2; $i<=$baris; $i++){
            {
                // membaca data (kolom ke-1 sd terakhir)

```

```

$lat      = $data->val($i, 1);
$long     = $data->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query = "INSERT into `tesisv3`.`t_dataawal` values(null,'$lat','$long')";
$hasil = mysql_query($query);
}

if(!$hasil){
// jika import gagal
die(mysql_error());
} else{
// jika impor berhasil
echo "<br>";
echo "Data berhasil diimpor.";
echo "<br>";
}
=====

$iterasi = $_POST['iterasi'];
$jarak = $_POST['jarak'];

//perubahan dari meter menjadi degree
$jarakMtoD = $jarak/30.92;
$jarakMtoD1 = $jarakMtoD/60;
$jarakMtoD2 = $jarakMtoD1/60;
$jarakdes = round($jarakMtoD2, 4); //fungsi pembulatan dengan 4 angka dibelakang koma
//echo $jarakdes;echo "<br>";

//pembagian jarak spasi dengan iterasi
$jarakits = $jarakdes/$iterasi;
=====

$querymaxlat = mysql_query("SELECT MAX(`latitude`) AS latitude FROM
`tesisv3`.`t_dataawal` ");
$arraymaxlat = mysql_fetch_array($querymaxlat);
$maxlat = $arraymaxlat['latitude'];
$hasilbataslatawal = $maxlat + $jarakdes;
// echo "$maxlat adalah maxlat";
// echo "<br>";
// echo "$hasilbataslatawal adalah hasil yang sudah ditambahkan";
// echo "<br>";

```

```

$queryminlat = mysql_query("SELECT MIN(`latitude`) AS latitude FROM `tesisv3`.`t_dataawal` ");
$arrayminlat = mysql_fetch_array($queryminlat);
$minlat = $arrayminlat['latitude'];
$hasilbataslataakhir = $minlat - $jarakdes;
// echo "$minlat adalah minlat";
// echo "<br>";
// echo "$hasilbataslata adalah hasil yang sudah dikurangkan";
// echo "<br>";

$queryminlong = mysql_query("SELECT MIN(`longitude`) AS longitude FROM `tesisv3`.`t_dataawal` ");
$arrayminlong = mysql_fetch_array($queryminlong);
$minlong = $arrayminlong['longitude'];
//echo "$minlong";echo "<br>";
$hasilbataslongawal = $minlong - $jarakdes;

$querymaxlong = mysql_query("SELECT MAX(`longitude`) AS longitude FROM `tesisv3`.`t_dataawal` ");
$arraymaxlong = mysql_fetch_array($querymaxlong);
$maxlong = $arraymaxlong['longitude'];
$hasilbataslongakhir = $maxlong + $jarakdes;
=====

=====
for ($i=1; $i <= $iterasi ; $i++) {
    $ctbatas = "CREATE TABLE `tesisv3`.`t_batas$i` ( `id` INT NOT NULL AUTO_INCREMENT , `latawal` double NOT NULL , `lataakhir` double NOT NULL , `longawal` double NOT NULL , `longakhir` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
    $qctbatas = mysql_query($ctbatas);

    $insertbatas = "INSERT INTO `tesisv3`.`t_batas$i`(id, latawal, lataakhir, longawal, longakhir) VALUES (NULL, '$hasilbataslataawal', '$hasilbataslataakhir', '$hasilbataslongawal', '$hasilbataslongakhir')";
    $qinsertbatas = mysql_query($insertbatas);
    $hasilbataslongawal=$hasilbataslongawal+$jarakits; //pergeseran ke kanan
}
=====

=====
for ($i=1; $i <= $iterasi ; $i++) {
    //pembuatan grid nya disini
    //create tabel sekalian t_grid
    $createtabellat = "CREATE TABLE `tesisv3`.`t_gridlat$i` ( `id` INT NOT NULL AUTO_INCREMENT , `lat` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
}

```

```

$createtabellong = "CREATE TABLE `tesisv3`.`t_gridlong$i` (`id` INT NOT NULL
AUTO_INCREMENT , `long` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
$qcreatelat = mysql_query($createtabellat);
$qcreatelong = mysql_query($createtabellong);

$a = "SELECT * from t_batas$i";
$qa = mysql_query($a);
$dataaa = mysql_fetch_array($qa);
$latawala = $dataaa['latawal'];
$latakhira = $dataaa['latakhir'];
$longawala = $dataaa['longawal'];
$longakhira = $dataaa['longakhir'];

while ($latawala >= $latakhira) {
    $inputlat = "INSERT INTO `tesisv3`.`t_gridlat$i` (`id`, `lat`) VALUES (NULL, $latawala);";
    $aksiinputlat = mysql_query($inputlat) or die ("Error in query: $inputlat ".mysql_error());
    $latawala = $latawala - $jarakdes;
}

while ($longawala <= $longakhira) {
    $inputlong = "INSERT INTO `tesisv3`.`t_gridlong$i` (`id`, `long`) VALUES (NULL,
$longawala);";
    $aksiinputlong = mysql_query($inputlong) or die ("Error in query: $inputlong ".
.mysql_error());
    $longawala = $longawala + $jarakdes;
}

=====

for ($i=1; $i <= $iterasi ; $i++) {
    $cthasil = "CREATE TABLE `tesisv3`.`t_hasil` (`id` INT NOT NULL
AUTO_INCREMENT , `LBH` INT NOT NULL , `tower` INT NOT NULL , `JML` INT NOT NULL ,
PRIMARY KEY (`id`)) ENGINE = InnoDB;";
    $qccthasil = mysql_query($cthasil);

    $sql = "SELECT count(`lat`) as total FROM `t_gridlat$i` where `lat` != 0 ";
    $sql2 = mysql_query($sql);
    $sql3 = mysql_fetch_array($sql2);
    $totallat = $sql3['total'];
    $a = "SELECT count(`long`) as total FROM `t_gridlong$i` where `long` != 0
";
    $a2 = mysql_query($a);
}

```

```

$a3 = mysql_fetch_array($a2);
$totallong = $a3['total'];

for ($k=1; $k <= $totallat; $k++) {
    //echo "i = $i";
    $q1 = "SELECT * FROM `t_gridlat$i` WHERE `id`=$k";
    $q2 = mysql_query($q1);
    $q3 = mysql_fetch_array($q2);
    //echo "id $i = ".$q3['latga']."<br>";
    $kk=$k+1;
    $w1 = "SELECT * FROM `t_gridlat$i` WHERE `id`=$kk";
    $w2 = mysql_query($w1);
    $w3 = mysql_fetch_array($w2);

    for ($j=1; $j <= $totallong; $j++) {
        //echo " ";
        //echo "<br>";
        //hitung disini
        $q1a = "SELECT * FROM `t_gridlong$i` WHERE
'id'=$j";
        $q2a = mysql_query($q1a);
        $q3a = mysql_fetch_array($q2a);
        $jj = $j + 1;
        $w1a = "SELECT * FROM `t_gridlong$i` WHERE
'id'=$jj";
        $w2a = mysql_query($w1a);
        $w3a = mysql_fetch_array($w2a);
        //echo $w3a['longitude'];
        //echo $j;
        //echo "<br>";
        $cek = "SELECT count(*) as jumlah From `t_dataawal`
where `latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >= ".$q3a['long']."' &&
`longitude` < ".$w3a['long']."'";

        $sqlcek = mysql_query($cek);
        $arraycek = mysql_fetch_array($sqlcek);
        $jmlmenara = $arraycek['jumlah'];
        if ($jmlmenara >= 4) {
            $itungLbh4 = $itungLbh4 + 1;
            $totmenara = $totmenara + $jmlmenara;
            $kwadratmenara = $jmlmenara*$jmlmenara;
            $totkwadrat = $kwadratmenara + $totkwadrat;
        }
    }
}

```

```

        // $jajal = "INSERT INTO `hasil$i` VALUES (NULL,
'$jmlmenara', '$jmlmakam', '$jmlmiliter', '$jmlindustri', '$jmlfasum')";
        // $aksi = mysql_query($jajal);
    }
    //echo "<br>";
}
$insHasil = "INSERT INTO `tesisv3`.`t_hasil` VALUES (null, $itungLbh4,
$totmenara, $totkwadrat)";
$aksi = mysql_query($insHasil);
echo "selesai perulangan ke-$i";
echo "<br>";
$itungLbh4 = 0;
$totmenara = 0;
$kwadratmenara = 0;
$totkwadrat = 0;
}

=====

//sort untuk menentukan optimasi
//SELECT * FROM `t_hasil` ORDER BY `JML` DESC, `tower` DESC LIMIT 1
//tampilkan hasil grid yang telah di generate
$optimsort = "SELECT * FROM `t_hasil` ORDER BY `JML` DESC, `tower` DESC
LIMIT 1";
$qoptimsort = mysql_query($optimsort);
$aroptimsort = mysql_fetch_array($qoptimsort);
$idoptim = $aroptimsort['id'];
//echo "$idoptim";
//echo "<br>";

$querytabel=mysql_query("select*from `tesisv3`.`t_hasil` ");
echo "
<table border=1><tr>
<th width=50 >id</th>
<th width=100>LBH</th>
<th width=100>Tower</th>
<th width=100>JML</th>
<th width=100>Hasil</th>
</tr>";
while ($data=mysql_fetch_array($querytabel)) {
echo "<tr>";

```

```

echo "<td>".$data['id']."</td>";
echo "<td>".$data['LBH']."</td>";
echo "<td>".$data['tower']."</td>";
echo "<td>".$data['JML']."</td>";
echo "<td>";
if ($data['id']==$idoptim) {
    echo "Dipilih";
}
"</td>";
}
echo "</table>";

?>
<p>Keterangan: <br>
LBH => Merupakan grid yang mempunyai tower lebih dari sama dengan 4<br>
Tower => Merupakan jumlah semua tower LBH<br>
JML => Merupakan hasil jumlah dari kwadrat isi LBH<br>
</p>

<?php

} //penutup isset

//=====
===== header akhir

$time_end = microtime(true);
$execution_time = ($time_end - $time_start)/60;
//echo "<br>";
//echo "waktu mulai = $time_start";
echo "<br>";
echo "<br>";
echo '<b>Total waktu yang dibutuhkan:</b> '.$execution_time.' Menit';

?>

```

Algoritma tahap kedua

```

<?php
error_reporting(0);
//koneksi ke database, username,password dan namadatabase menyesuaikan
include "db2.php";
$time_start = microtime(true);
ini_set("max_execution_time", 0);
require "excel_reader.php";
//=====
===== header awal

```

```

?>
<form name="myForm" id="myForm" onSubmit="return validateForm()" action="paket2.php"
method="post" enctype="multipart/form-data">
    input menara : <input type="file" id="fileexcel" name="menara" /><br/>
    input makam : <input type="file" id="fileexcel" name="makam" /><br/>
    input militer : <input type="file" id="fileexcel" name="militer" /><br/>
    input industri : <input type="file" id="fileexcel" name="industri" /><br/>
    input fasum : <input type="file" id="fileexcel" name="fasum" /><br/>
    <label><input type="checkbox" name="drop" value="1" /> <u>Rubah data?</u> </label><br/>
    berapa perulangan : <input type="text" name="iterasi"><br>
        berapa jarak space nya : <input type="text" name="jarak"> meter <br>
        <input type="submit" name="submit" value="Import" /><br/>
</form>

<script type="text/javascript">
// validasi form (hanya file .xls yang diijinkan)
function validateForm()
{
    function hasExtension(inputID, exts) {
        var fileName = document.getElementById(inputID).value;
        return (new RegExp('(^|' + exts.join('|').replace(/\./g, '\\.)) + '$')).test(fileName);
    }
}
</script>

<?php
if(isset($_POST['submit'])){
//===== delete tabel
$deltbl=db=mysql_query("SHOW TABLES FROM tesisv3tes");
while($dataassoc=mysql_fetch_assoc($deltbl)){
    mysql_query("DROP TABLE ".$dataassoc["Tables_in_"].thesisv3tes);
}

//=====
//=====
===== buat database
// database -> t_dataawal , t_laturut , t_longurut
$drop = isset( $_POST["drop"] ) ? $_POST["drop"] : 0 ;
if($drop == 1){
//detele tabel dulu, baru create
$deltbltes=mysql_query("DROP TABLE `thesisv3tesi`.`t_fasum` , `thesisv3tesi`.`t_industri` ,
`thesisv3tesi`.`t_makam` , `thesisv3tesi`.`t_menara` , `thesisv3tesi`.`t_militer` ");
//=====
$dbtblnew1 = "CREATE TABLE `thesisv3tesi`.`t_menara` ( `id` INT NOT NULL
AUTO_INCREMENT , `latitude` DOUBLE NOT NULL , `longitude` DOUBLE NOT NULL ,
PRIMARY KEY ( `id` )) ENGINE = InnoDB";
$dbtblnew2 = "CREATE TABLE `thesisv3tesi`.`t_makam` ( `id` INT NOT NULL
AUTO_INCREMENT , `latitude` DOUBLE NOT NULL , `longitude` DOUBLE NOT NULL ,
PRIMARY KEY ( `id` )) ENGINE = InnoDB";
$dbtblnew3 = "CREATE TABLE `thesisv3tesi`.`t_militer` ( `id` INT NOT NULL
AUTO_INCREMENT , `latitude` DOUBLE NOT NULL , `longitude` DOUBLE NOT NULL ,
PRIMARY KEY ( `id` )) ENGINE = InnoDB";
$dbtblnew4 = "CREATE TABLE `thesisv3tesi`.`t_industri` ( `id` INT NOT NULL
AUTO_INCREMENT , `latitude` DOUBLE NOT NULL , `longitude` DOUBLE NOT NULL ,
PRIMARY KEY ( `id` )) ENGINE = InnoDB";
$dbtblnew5 = "CREATE TABLE `thesisv3tesi`.`t_fasum` ( `id` INT NOT NULL
AUTO_INCREMENT , `latitude` DOUBLE NOT NULL , `longitude` DOUBLE NOT NULL ,
PRIMARY KEY ( `id` )) ENGINE = InnoDB";
$qdbtblnew1 = mysql_query($dbtblnew1);
$qdbtblnew2 = mysql_query($dbtblnew2);
$qdbtblnew3 = mysql_query($dbtblnew3);
$qdbtblnew4 = mysql_query($dbtblnew4);
$qdbtblnew5 = mysql_query($dbtblnew5);
}
}

```

```

//=====
=====

$target1 = basename($_FILES['menara']['name']) ;
$target2 = basename($_FILES['makam']['name']) ;
$target3 = basename($_FILES['militer']['name']) ;
$target4 = basename($_FILES['industri']['name']) ;
$target5 = basename($_FILES['fasum']['name']) ;

move_uploaded_file($_FILES['menara']['tmp_name'], $target1);
move_uploaded_file($_FILES['makam']['tmp_name'], $target2);
move_uploaded_file($_FILES['militer']['tmp_name'], $target3);
move_uploaded_file($_FILES['industri']['tmp_name'], $target4);
move_uploaded_file($_FILES['fasum']['tmp_name'], $target5);

$data1 = new Spreadsheet_Excel_Reader($_FILES['menara']['name'],false);
$data2 = new Spreadsheet_Excel_Reader($_FILES['makam']['name'],false);
$data3 = new Spreadsheet_Excel_Reader($_FILES['militer']['name'],false);
$data4 = new Spreadsheet_Excel_Reader($_FILES['industri']['name'],false);
$data5 = new Spreadsheet_Excel_Reader($_FILES['fasum']['name'],false);

// menghitung jumlah baris file xls
$baris1 = $data1->rowcount($sheet_index=0);
$baris2 = $data2->rowcount($sheet_index=0);
$baris3 = $data3->rowcount($sheet_index=0);
$baris4 = $data4->rowcount($sheet_index=0);
$baris5 = $data5->rowcount($sheet_index=0);

// import data excel mulai baris ke-2 (karena tabel xls ada header pada baris 1)
//=== 1
for ($i=2; $i<=$baris1; $i++)
{
// membaca data (kolom ke-1 sd terakhir)
$lat1      = $data1->val($i, 1);
$long1     = $data1->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query1 = "INSERT into `tesisv3tesi`.`t_menara` values(null,'$lat1','$long1')";
$hasil1 = mysql_query($query1);
}

//=== 2
for ($i=2; $i<=$baris2; $i++)
{
// membaca data (kolom ke-1 sd terakhir)
$long2      = $data2->val($i, 1);
$lat2      = $data2->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query2 = "INSERT into `tesisv3tesi`.`t_makam` values(null,'$lat2','$long2')";
$hasil2 = mysql_query($query2);
}

//=== 3
for ($i=2; $i<=$baris3; $i++)
{
// membaca data (kolom ke-1 sd terakhir)
$long3      = $data3->val($i, 1);
$lat3      = $data3->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query3 = "INSERT into `tesisv3tesi`.`t_militer` values(null,'$lat3','$long3')";
$hasil3 = mysql_query($query3);
}

```

```

//===== 4
for ($i=2; $i<=$baris4; $i++)
{
// membaca data (kolom ke-1 sd terakhir)
$long4      = $data4->val($i, 1);
$lat4       = $data4->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query4 = "INSERT into `tesisv3tesi`.`t_industri` values(null,'$lat4','$long4')";
$hasil4 = mysql_query($query4);
}

//===== 5
for ($i=2; $i<=$baris5; $i++)
{
// membaca data (kolom ke-1 sd terakhir)
$long5      = $data5->val($i, 1);
$lat5       = $data5->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query5 = "INSERT into `tesisv3tesi`.`t_fasum` values(null,$lat5','$long5')";
$hasil5 = mysql_query($query5);
}

//===== selesai
if(!$hasil1){
// jika import gagal
die(mysql_error());
}else{
// jika impor berhasil
echo "Data menara berhasil diimpor.";
echo "<br>";
}
if(!$hasil2){
// jika import gagal
die(mysql_error());
}else{
// jika impor berhasil
echo "Data makam berhasil diimpor.";
echo "<br>";
}
if(!$hasil3){
// jika import gagal
die(mysql_error());
}else{
// jika impor berhasil
echo "Data militer berhasil diimpor.";
echo "<br>";
}
if(!$hasil4){
// jika import gagal
die(mysql_error());
}else{
// jika impor berhasil
echo "Data industri berhasil diimpor.";
echo "<br>";
}
if(!$hasil5){
// jika import gagal
die(mysql_error());
}else{
// jika impor berhasil
echo "Data fasum berhasil diimpor.";
echo "<br>";
}
}

```

```

};

//mulai script generate grid dll
//=====

$iterasi = $_POST['iterasi'];
$jarak = $_POST['jarak'];

//perubahan dari meter menjadi degree
$jarakMtoD = $jarak/30.92;
$jarakMtoD1 = $jarakMtoD/60;
$jarakMtoD2 = $jarakMtoD1/60;
$jarakdes = round($jarakMtoD2, 4); //fungsi pembulatan dengan 4 angka dibelakang koma
//echo $jarakdes;echo "<br>";

//pembagian jarak spasi dengan iterasi
$jarakits = $jarakdes/$iterasi;
//=====

$querymaxlat = mysql_query("SELECT MAX(`latitude`) AS latitude FROM `tesisv3tesi`.`t_menara` ");
$arraymaxlat = mysql_fetch_array($querymaxlat);
$maxlat = $arraymaxlat['latitude'];
$hasilbataslatawal = $maxlat + $jarakdes;
// echo "$maxlat adalah maxlat";
// echo "<br>";
// echo "$hasilbataslatawal adalah hasil yang sudah ditambahkan";
// echo "<br>";
$queryminlat = mysql_query("SELECT MIN(`latitude`) AS latitude FROM `tesisv3tesi`.`t_menara` ");
$arrayminlat = mysql_fetch_array($queryminlat);
$minlat = $arrayminlat['latitude'];
$hasilbataslataakhir = $minlat - $jarakdes;
// echo "$minlat adalah minlat";
// echo "<br>";
// echo "$hasilbataslataakhir adalah hasil yang sudah dikurangkan";
// echo "<br>";

$queryminlong = mysql_query("SELECT MIN(`longitude`) AS longitude FROM
`tesisv3tesi`.`t_menara` ");
$arrayminlong = mysql_fetch_array($queryminlong);
$minlong = $arrayminlong['longitude'];
//echo "$minlong";echo "<br>";
$hasilbataslongawal = $minlong - $jarakdes;

$querymaxlong = mysql_query("SELECT MAX(`longitude`) AS longitude FROM
`tesisv3tesi`.`t_menara` ");
$arraymaxlong = mysql_fetch_array($querymaxlong);
$maxlong = $arraymaxlong['longitude'];
$hasilbataslongakhir = $maxlong + $jarakdes;
//=====

for ($i=1; $i <= $iterasi ; $i++) {
    $ctbatas = "CREATE TABLE `tesisv3tesi`.`t_batas$i` (`id` INT NOT NULL AUTO_INCREMENT ,
`latawal` double NOT NULL ,`lataakhir` double NOT NULL ,`longawal` double NOT NULL ,`longakhir` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
    $qctbatas = mysql_query($ctbatas);

    $insertbatas = "INSERT INTO `tesisv3tesi`.`t_batas$i`(`id`, latawal, lataakhir, longawal, longakhir)
VALUES (NULL, '$hasilbataslatawal', '$hasilbataslataakhir', '$hasilbataslongawal',
'$hasilbataslongakhir')";
    $qinsertbatas = mysql_query($insertbatas);
    $hasilbataslongawal=$hasilbataslongawal+$jarakits; //pergeseran ke kanan
}

```

```

//=====
=====

for ($i=1; $i <= $iterasi ; $i++) {
    //pembuatan grid nya disini
    //create tabel sekalian t_grid
    $createtabellat = "CREATE TABLE `tesisv3tes`.`t_gridlat$i` (`id` INT NOT NULL
    AUTO_INCREMENT , `lat` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
    $createtabellong = "CREATE TABLE `tesisv3tes`.`t_gridlong$i` (`id` INT NOT NULL
    AUTO_INCREMENT , `long` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
    $qcreatelat = mysql_query($createtabellat);
    $qcreatelong = mysql_query($createtabellong);

    $a = "SELECT * from t_batas$i";
    $qa = mysql_query($a);
    $dataaa = mysql_fetch_array($qa);
    $latawala = $dataaa['latawala'];
    $latakhir = $dataaa['latakhir'];
    $longawala = $dataaa['longawala'];
    $longakhira = $dataaa['longakhir'];

    while ($latawala >= $latakhir) {
        $inputlat = "INSERT INTO `tesisv3tes`.`t_gridlat$i` (`id`, `lat`) VALUES (NULL, $latawala);";
        $aksiinputlat = mysql_query($inputlat) or die ("Error in query: $inputlat. ".mysql_error());
        $latawala = $latawala - $jarakdes;
    }

    while ($longawala <= $longakhira) {
        $inputlong = "INSERT INTO `tesisv3tes`.`t_gridlong$i` (`id`, `long`) VALUES (NULL, $longawala);";
        $aksiinputlong = mysql_query($inputlong) or die ("Error in query: $inputlong. ".mysql_error());
        $longawala = $longawala + $jarakdes;
    }
}

//=====
=====

for ($i=1; $i <= $iterasi ; $i++) {
    $cthasil = "CREATE TABLE `tesisv3tes`.`t_hasil` (`id` INT NOT NULL
    AUTO_INCREMENT , `GNA` INT NOT NULL , `GA` INT NOT NULL , `TG` INT NOT NULL ,
    PRIMARY KEY (`id`)) ENGINE = InnoDB";
    $qcthasil = mysql_query($cthasil);

    $sql = "SELECT count(`lat`) as total FROM `tesisv3tes`.`t_gridlat$i` where `lat` != 0
";
    $sql2 = mysql_query($sql);
    $sql3 = mysql_fetch_array($sql2);
    $totallat = $sql3['total'];
    $totallatfix = $totallat-1;
    $a = "SELECT count(`long`) as total FROM `tesisv3tes`.`t_gridlong$i` where `long` != 0 ";
    $a2 = mysql_query($a);
    $a3 = mysql_fetch_array($a2);
    $totallong = $a3['total'];
    $totallongfix = $totallong-1;

    for ($k=1; $k <= $totallatfix; $k++) {
        //echo "i = $i";
        $q1 = "SELECT * FROM `tesisv3tes`.`t_gridlat$i` WHERE `id`=$k";
        $q2 = mysql_query($q1);
        $q3 = mysql_fetch_array($q2);
        //echo "id $i = ".$q3['latga']."<br>";
        $kk=$k+1;
        $w1 = "SELECT * FROM `tesisv3tes`.`t_gridlat$i` WHERE `id`=$kk";
        $w2 = mysql_query($w1);
        $w3 = mysql_fetch_array($w2);
}

```

```

for ($j=1; $j <= $totallongfix; $j++) {
    //echo " ";
    //echo "<br>";
    //hitung disini
    $q1a = "SELECT * FROM `tesisv3tes`.`t_gridlong$i` WHERE
`id`=$j";
    $q2a = mysql_query($q1a);
    $q3a = mysql_fetch_array($q2a);
    $jj = $j + 1;
    $w1a = "SELECT * FROM `tesisv3tes`.`t_gridlong$i` WHERE
`id`=$jj";
    $w2a = mysql_query($w1a);
    $w3a = mysql_fetch_array($w2a);
    //echo $w3a['longitude'];
    //echo $j;
    //echo "<br>";
//===== cek menara
$cek = "SELECT count(*) as jumlah From
`tesisv3tesi`.`t_menara` where `latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >=
".$q3a['long']."' && `longitude` < ".$w3a['long']."' ";
$sqlcek = mysql_query($cek);
$arraycek = mysql_fetch_array($sqlcek);
$jmlmenara = $arraycek['jumlah'];
if ($jmlmenara <= 1) {
    $botmenara = 0;
} elseif ($jmlmenara <= 3) {
    $botmenara = 5;
} else {
    $botmenara = 10;
}
//===== cek makam
$cekmakam = "SELECT count(*) as jumlah From `tesisv3tesi`.`t_makam`"
where `latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >= ".$q3a['long']."' &&
`longitude` < ".$w3a['long']."' ";
$sqlcekmakam = mysql_query($cekmakam);
$arraycekmakam = mysql_fetch_array($sqlcekmakam);
$jmlmakam = $arraycekmakam['jumlah'];
if ($jmlmakam > 0) {
    $botmakam = 8;
} else {
    $botmakam = 0;
}
//===== cek militer
$cekmiliter = "SELECT count(*) as jumlah From `tesisv3tesi`.`t_militer`"
where `latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >= ".$q3a['long']."' &&
`longitude` < ".$w3a['long']."' ";
$sqlcekmiliter = mysql_query($cekmiliter);
$arraycekmiliter = mysql_fetch_array($sqlcekmiliter);
$jmlmiliter = $arraycekmiliter['jumlah'];
if ($jmlmiliter > 0) {
    $botmiliter = 5;
} else {
    $botmiliter = 0;
}
//===== cek industri
$cekindustri = "SELECT count(*) as jumlah From `tesisv3tesi`.`t_industri`"
where `latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >= ".$q3a['long']."' &&
`longitude` < ".$w3a['long']."' ";
$sqlcekindustri = mysql_query($cekindustri);
$arraycekindustri = mysql_fetch_array($sqlcekindustri);
$jmlindustri = $arraycekindustri['jumlah'];
if ($jmlindustri > 0) {

```

```

$botindustri = 2;
} else {
$botindustri = 0;
}
//===== cek fasum
$cекfasum = "SELECT count(*) as jumlah From `tesisv3tes`.`t_fasum`  

where `latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >= ".$q3a['long']."' &&  

`longitude` < ".$w3a['long']."' ";
$sqlcekfasum = mysql_query($cekfasum);
$arraycekfasum = mysql_fetch_array($sqlcekfasum);
$jmlfasum = $arraycekfasum['jumlah'];
if ($jmlfasum > 0) {
$botfasum = 2;
} else {
$botfasum = 0;
}
//===== cek total, untuk dimasukkan di  

GNA atau GA
//echo "<br>";
//echo "$botmenara + $botmakam + $botmiliter + $botindustri +  

$botfasum"; echo "<br>";
$bottotal =
$botmenara+$botmakam+$botmiliter+$botindustri+$botfasum;//jumlah bot makam, mmiliter, industri,  

fasum
if ($bottotal >= 10) {
$GNA = $GNA + 1;
}
$totalgrid = $totalgrid + 1;

}
//echo "<br>";
}
//echo "<br>";
//echo "selesai perulangan ke-$i";
//echo "<br>";
$GA = $totalgrid - $GNA;
//echo "grid not available = $GNA";
//echo "<br>";
//echo "grid available = $GA";
//echo "<br>";
//echo "dari total grid $totalgrid";
//echo "<br>";
$insHasilGrid = "INSERT INTO `tesisv3tes`.`t_hasil` VALUES (null, $GNA, $GA,
$totalgrid)";
$aksigrid = mysql_query($insHasilGrid);
$GNA = 0;
$GA = 0;
$totalgrid = 0;
}

$optimsort = "SELECT * FROM `tesisv3tes`.`t_hasil` ORDER BY `GNA` DESC LIMIT 1";
$qoptimsort = mysql_query($optimsort);
$arroptimsort = mysql_fetch_array($qoptimsort);
$idoptim = $arroptimsort['id'];

$querytabel=mysql_query("select*from `tesisv3tes`.`t_hasil` ");
echo "
<table border=1><tr>
<th width=50 >id</th>
<th width=100>GNA</th>
<th width=100>GA</th>

```

```

<th width=100>TotalGrid</th>
<th width=100>Hasil</th>
</tr>;
while ($data=mysql_fetch_array($querytable)) {
echo "<tr>";
echo "<td>".$data['id']."</td>";
echo "<td>".$data['GNA']."</td>";
echo "<td>".$data['GA']."</td>";
echo "<td>".$data['TG']."</td>";
echo "<td>";
if ($data['id']==$idoptim) {
    echo "Dipilih";
}
"</td>";
}
echo "</table>";

}

=====
===== header akhir
$time_end = microtime(true);
$execution_time = ($time_end - $time_start)/60;
//echo "<br>";
//echo "waktu mulai = $time_start";
echo "<br>";
echo "<br>";
echo '<b>Total waktu yang dibutuhkan:</b> '.$execution_time.' Menit';
?>

```

Algoritma pencarian koordinat

```

<?php
error_reporting(0);

//koneksi ke database, username,password dan namadatabase menyesuaikan
include "db.php";

$time_start = microtime(true);
ini_set("max_execution_time", 0);
require "excel_reader.php";

=====
===== header awal

//Input file menara dulu
echo "Input excel dan input berapa banyak iterasi yang dibutuhkan";
echo "<br>";
echo "<br>";
?>
<!-- tambahkan inputan untuk excel -->
<form name="input" action="paket1.php" method="post" onSubmit="return validateForm()"
enctype="multipart/form-data">
    input file excel :<input type="file" id="fileexcel" name="fileexcel" /><br>
    berapa perulangan :<input type="text" name="iterasi"><br>

```

```

berapa jarak grid nya : <input type="text" name="jarak"> meter <br>
<input type="submit" name="submit" value="Submit">
</form>

<script type="text/javascript">
// validasi form (hanya file .xls yang diijinkan)
function validateForm()
{
    function hasExtension(inputID, exts) {
        var fileName = document.getElementById(inputID).value;
        return (new RegExp('(' + exts.join('|').replace(/\./g, '\\.') + ')$')).test(fileName);
    }

    if(!hasExtension('fileexcel', ['.xls'])){
        alert("Hanya file XLS (Excel 2003) yang diijinkan.");
        return false;
    }
}
</script>
<?php
// hapus file xls yang udah dibaca
//unlink($_FILES['fileexcel']['name']);
// input excel
// excel masuk ke database, masuk ke database t_dataawal, database ini merupakan patokan awal
semuanya.
// kalau normal, dari sini data diurutkan untuk membuat grid, maka data ini juga harus diurutkan terlebih
dahulu
// setelah data urut masuk ke dalam database t_laturut dan t_longurut, maka data baru di modifikasi!!
if(isset($_POST['submit'])){
    //cek paketkomplity2 untuk tahu mana yang harus di create, dan mana yang harus di delete.
    $delttbl=mysql_query("SHOW TABLES FROM tesisv3");

    while($dataassoc=mysql_fetch_assoc($delttbl)){
        mysql_query("DROP TABLE ".$dataassoc["Tables_in_"].thesisv3);
    }

//=====
===== buat database
// database -> t_dataawal , t_laturut , t_longurut
$dtblnew1 ="CREATE TABLE `thesisv3`.`t_dataawal` ( `id` INT NOT NULL AUTO_INCREMENT ,
`latitude` DOUBLE NOT NULL , `longitude` DOUBLE NOT NULL , PRIMARY KEY (`id`)) ENGINE
= InnoDB";

```

```

$qdtblnew1 = mysql_query($dbtblnew1);

//=====
=====

//=====

$target = basename($_FILES['fileexcel']['name']) ;
move_uploaded_file($_FILES['fileexcel']['tmp_name'], $target);

$data = new Spreadsheet_Excel_Reader($_FILES['fileexcel']['name'],false);

// menghitung jumlah baris file xls
$baris = $data->rowcount($sheet_index=0);

// import data excel mulai baris ke-2 (karena tabel xls ada header pada baris 1)
for ($i=2; $i<=$baris; $i++)
{
// membaca data (kolom ke-1 sd terakhir)
$lat      = $data->val($i, 1);
$long     = $data->val($i, 2);

// setelah data dibaca, masukkan ke tabel t_dataawal sql
$query = "INSERT into `tesisv3`.`t_dataawal` values(null,'$lat','$long')";
$hasil = mysql_query($query);

}

if(!$hasil){
// jika import gagal
die(mysql_error());
} else{
// jika impor berhasil
echo "<br>";
echo "Data berhasil diimpor.";
echo "<br>";
}

//=====
=====

$iterasi = $_POST['iterasi'];
$jarak = $_POST['jarak'];

//perubahan dari meter menjadi degree
$jarakMtoD = $jarak/30.92;
$jarakMtoD1 = $jarakMtoD/60;

```

```

$jarakMtoD2 = $jarakMtoD1/60;
$jarakdes = round($jarakMtoD2, 4); //fungsi pembulatan dengan 4 angka dibelakang koma
//echo $jarakdes;echo "<br>";

//pembagian jarak spasi dengan iterasi
$jarakits = $jarakdes/$iterasi;
//=====
=====

$querymaxlat = mysql_query("SELECT MAX(`latitude`) AS latitude FROM `tesisv3`.`t_dataawal` ");
$arraymaxlat = mysql_fetch_array($querymaxlat);
$maxlat = $arraymaxlat['latitude'];
$hasilbataslatawal = $maxlat + $jarakdes;
// echo "$maxlat adalah maxlat";
// echo "<br>";
// echo "$hasilbataslatawal adalah hasil yang sudah ditambahkan";
// echo "<br>";
$queryminlat = mysql_query("SELECT MIN(`latitude`) AS latitude FROM `tesisv3`.`t_dataawal` ");
$arrayminlat = mysql_fetch_array($queryminlat);
$minlat = $arrayminlat['latitude'];
$hasilbataslatakhir = $minlat - $jarakdes;
// echo "$minlat adalah minlat";
// echo "<br>";
// echo "$hasilbataslat adalah hasil yang sudah dikurangkan";
// echo "<br>";

$queryminlong = mysql_query("SELECT MIN(`longitude`) AS longitude FROM `tesisv3`.`t_dataawal` ");
$arrayminlong = mysql_fetch_array($queryminlong);
$minlong = $arrayminlong['longitude'];
//echo "$minlong";echo "<br>";
$hasilbataslongawal = $minlong - $jarakdes;

$querymaxlong = mysql_query("SELECT MAX(`longitude`) AS longitude FROM `tesisv3`.`t_dataawal` ");
$arraymaxlong = mysql_fetch_array($querymaxlong);
$maxlong = $arraymaxlong['longitude'];
$hasilbataslongakhir = $maxlong + $jarakdes;
//=====
=====

for ($i=1; $i <= $iterasi ; $i++) {

```

```

$ctbatas = "CREATE TABLE `tesisv3`.`t_batas`(`id` INT NOT NULL AUTO_INCREMENT ,
`latawal` double NOT NULL ,`latakhir` double NOT NULL ,`longawal` double NOT NULL ,`longakhir` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
$qctbatas = mysql_query($ctbatas);

$insertbatas = "INSERT INTO `tesisv3`.`t_batas`(`id, latawal, latakhir, longawal, longakhir) VALUES
(NULL, '$hasilbataslatawal', '$hasilbataslatakhir', '$hasilbataslongawal', '$hasilbataslongakhir')";
$qinsertbatas = mysql_query($insertbatas);
$hasilbataslongawal=$hasilbataslongawal+$jarakits; //pergeseran ke kanan
}

=====
=====

for ($i=1; $i <= $iterasi ; $i++) {
//pembuatan grid nya disini
//create tabel sekalian t_grid
$createtabellat = "CREATE TABLE `tesisv3`.`t_gridlat`(`id` INT NOT NULL
AUTO_INCREMENT ,`lat` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
$createtabellong = "CREATE TABLE `tesisv3`.`t_gridlong`(`id` INT NOT NULL
AUTO_INCREMENT ,`long` double NOT NULL , PRIMARY KEY (`id`)) ENGINE = InnoDB";
$qcreatelat = mysql_query($createtabellat);
$qcreatelong = mysql_query($createtabellong);

$a = "SELECT * from t_batas$i";
$qa = mysql_query($a);
$dataa = mysql_fetch_array($qa);
$latawala = $dataa['latawal'];
$latakhira = $dataa['latakhir'];
$longawala = $dataa['longawal'];
$longakhira = $dataa['longakhir'];

while ($latawala >= $latakhira) {
$inputlat = "INSERT INTO `tesisv3`.`t_gridlat`(`id`, `lat`) VALUES (NULL, $latawala);";
$aksiinputlat = mysql_query($inputlat) or die ("Error in query: $inputlat. ".mysql_error());
$latawala = $latawala - $jarakdes;
}

while ($longawala <= $longakhira) {
$inputlong = "INSERT INTO `tesisv3`.`t_gridlong`(`id`, `long`) VALUES (NULL, $longawala);";
$aksiinputlong = mysql_query($inputlong) or die ("Error in query: $inputlong. ".mysql_error());
$longawala = $longawala + $jarakdes;
}
}

```

```

//=====
=====

for ($i=1; $i <= $iterasi ; $i++) {
    $cthasil = "CREATE TABLE `tesisv3`.`t_hasil` (`id` INT NOT NULL
    AUTO_INCREMENT , `LBH` INT NOT NULL , `tower` INT NOT NULL , `JML` INT NOT NULL ,
    PRIMARY KEY (`id`)) ENGINE = InnoDB;";
    $qcthasil = mysql_query($cthasil);

    $sql = "SELECT count(`lat`) as total FROM `t_gridlat$i` where `lat` != 0 ";
    $sql2 = mysql_query($sql);
    $sql3 = mysql_fetch_array($sql2);
    $totallat = $sql3['total'];
    $a = "SELECT count(`long`) as total FROM `t_gridlong$i` where `long` != 0 ";
    $a2 = mysql_query($a);
    $a3 = mysql_fetch_array($a2);
    $totallong = $a3['total'];

    for ($k=1; $k <= $totallat; $k++) {
        //echo "i = $i";
        $q1 = "SELECT * FROM `t_gridlat$i` WHERE `id`=$k";
        $q2 = mysql_query($q1);
        $q3 = mysql_fetch_array($q2);
        //echo "id $i = ".$q3['latga']."<br>";
        $kk=$k+1;
        $w1 = "SELECT * FROM `t_gridlat$i` WHERE `id`=$kk";
        $w2 = mysql_query($w1);
        $w3 = mysql_fetch_array($w2);

        for ($j=1; $j <= $totallong; $j++) {
            //echo " ";
            //echo "<br>";
            //hitung disini
            $q1a = "SELECT * FROM `t_gridlong$i` WHERE `id`=$j";
            $q2a = mysql_query($q1a);
            $q3a = mysql_fetch_array($q2a);
            $jj = $j + 1;
            $w1a = "SELECT * FROM `t_gridlong$i` WHERE `id`=$jj";
            $w2a = mysql_query($w1a);
            $w3a = mysql_fetch_array($w2a);
            //echo $w3a['longitude'];
            //echo $j;
            //echo "<br>";
        }
    }
}

```

```

$cek = "SELECT count(*) as jumlah From `t_dataawal` where
`latitude` <= ".$q3['lat']."' && `latitude` > ".$w3['lat']."' && `longitude` >= ".$q3a['long']."' && `longitude` < ".$w3a['long']."' ";
$sqlcek = mysql_query($cek);
$arraycek = mysql_fetch_array($sqlcek);
$jmlmenara = $arraycek['jumlah'];
if ($jmlmenara >= 4) {
    $itungLbh4 = $itungLbh4 + 1;
    $totmenara = $totmenara + $jmlmenara;
    $kwadratmenara = $jmlmenara*$jmlmenara;
    $totkwadrat = $kwadratmenara + $totkwadrat;
}
//jajal = "INSERT INTO `hasil$i` VALUES (NULL,
'$jmlmenara', '$jmlmakam', '$jmlmiliter', '$jmlindustri', '$jmlfasum')";
//aksi = mysql_query($jajal);
}
//echo "<br>";
}
$insHasil = "INSERT INTO `tesisv3`.`t_hasil` VALUES (null, $itungLbh4,
$totmenara, $totkwadrat)";
$aksi = mysql_query($insHasil);
echo "selesai perulangan ke-$i";
echo "<br>";
$itungLbh4 = 0;
$totmenara = 0;
$kwadratmenara = 0;
$totkwadrat = 0;
}

=====

=====

//sort untuk menentukan optimasi
//SELECT * FROM `t_hasil` ORDER BY `JML` DESC, `tower` DESC LIMIT 1
//tampilkan hasil grid yang telah di generate
$optimsort = "SELECT * FROM `t_hasil` ORDER BY `JML` DESC, `tower` DESC LIMIT 1";
$qoptimsort = mysql_query($optimsort);
$roptimsort = mysql_fetch_array($qoptimsort);
$idoptim = $roptimsort['id'];
//echo "$idoptim";
//echo "<br>";

```

```

$querytabel=mysql_query("select*from `tesisv3`.`t_hasil`");
echo "
<table border=1><tr>
<th width=50 >id</th>
<th width=100>LBH</th>
<th width=100>Tower</th>
<th width=100>JML</th>
<th width=100>Hasil</th>
</tr>";
while ($data=mysql_fetch_array($querytabel)) {
echo "<tr>";
echo "<td>".$data['id']."'</td>";
echo "<td>".$data['LBH']."'</td>";
echo "<td>".$data['tower']."'</td>";
echo "<td>".$data['JML']."'</td>";
echo "<td>";
if ($data['id']==$idoptim) {
    echo "Dipilih";
}
"</td>";
}
echo "</table>";

?>
<p>Keterangan: <br>
LBH => Merupakan grid yang mempunyai tower lebih dari sama dengan 4<br>
Tower => Merupakan jumlah semua tower LBH<br>
JML => Merupakan hasil jumlah dari kwadrat isi LBH<br>
</p>

<?php

} //penutup isset

//=====
===== header akhir
$time_end = microtime(true);
$execution_time = ($time_end - $time_start)/60;
//echo "<br>";
//echo "waktu mulai = $time_start";
echo "<br>";
echo "<br>";

```

```
echo '<b>Total waktu yang dibutuhkan:</b> '.$execution_time.' Menit';
?>
```

Algoritma genetika

```
<?php
$tmp = explode('.', $_FILES['filenya']['name']);
if($tmp[count($tmp)-1] !== 'xls'){
    echo 'File harus berupa excell 2003 (xls). Klik <a href="index.php">disini</a> untuk
    kembali';
    exit(0);
}

require 'vendor/autoload.php';
$inputFileName = $_FILES['filenya']['tmp_name'];
$spreadsheet = \PhpOffice\PhpSpreadsheet\IOFactory::load($inputFileName);
$data = $spreadsheet->getActiveSheet()->toArray(null, true,true,true);
//convert data
$tmp = array();
foreach($data as $x=>$r)
{
    $y = 1;
    foreach($r as $c)
    {
        $tmp[$x][$y] = $c;
        $y += 1;
    }
}
$data = $tmp;
$jmlBaris = count($data);
$jmlKolom = count($data[1]);

function individu()
{
    GLOBAL $data;
    $x = array_rand($data,1);
    $y = array_rand($data[1],1);
```

```

        return array($x, $y);
    }

function populasi($jmlPop)
{
    $pop = array();
    while(count($pop) < $jmlPop)
    {
        $pop[] = individu();
    }
    return $pop;
}

function fitness($ind)
{
    GLOBAL $data;

    /*
    $x = $ind[0];
    $y = $ind[1];
    */

    $fit = 0;
    $individu = $data[$ind[0]][($ind[1])];
    // ketika nilai titik tengah tidak sama dengan 1 maka bobot 10000
    if($individu != 1){ $fit += 10000; }
    $tmp = array();
    // mencari tetangga dari titik tengah
    for($i=-1;$i<2;$i++)
    {
        for($j=-1;$j<2;$j++)
        {
            if(!isset($data[$ind[0]+$i]))
            {
                $tmp[] = 1000;
                continue;
            }elseif(!isset($data[$ind[0]+$i][$ind[1]+$j])){
                $tmp[] = 1000;
            }
        }
    }
}

```

```

        continue;
    }else{
        $tmp[] = $data[$ind[0]+$i][$ind[1]+$j];
    }
}

// mencari bobot tetangga dari 1
foreach($tmp as $c)
{
    $fit += $c;
}
return $fit;
}

function tourny($pop)
{
    $a = array_rand($pop,1);
    $b = array_rand($pop,1);
    return (fitness($pop[$a]) < fitness($pop[$b])) ? $pop[$a] : $pop[$b];
}

function crossover($a, $b)
{
    return array(
        array($a[0], $b[1]),
        array($b[0], $a[1]),
    );
}

function mutasi($ind)
{
    GLOBAL $jmlBaris;
    GLOBAL $jmlKolom;

    if(rand(0,1) == 0)
    {
        $rand = rand(1,$jmlBaris-1);
        return array($rand, $ind[1]);
    }
}

```

```

}else{
    $rand = rand(1,$jmlKolom-1);
    return array($ind[0], $rand);
}

return $ind;
}

function cmp($a, $b)
{
    if(fitness($a) == fitness($b))
        return 0;
    return (fitness($a) > fitness($b) ? 1 : -1);
}

function run($iter = 10000)
{
    GLOBAL $data;
    GLOBAL $jmlBaris;
    GLOBAL $jmlKolom;

    $jmlPop    = 3;
    $jmlElit   = 1;
    $jmlSeleksi = 3;
    $probCross = 0.75;
    $probMutasi = 0.75;
    $jmlIterasi = $iter;

    /*
    sample merupakan contoh populasi awal yang dihasilkan secara random
    */
    $sample = populasi($jmlPop);
    usort($sample, 'cmp');

    $generasi = 0;
    while(fitness($sample[0]) != 1 && $generasi < $jmlIterasi)
    {
        $generasi += 1;
}

```

```

//echo 'elitisme<br>';
$newPop = array();
for($i=0;$i<$jmlElit;$i++)
{
    $newPop[] = $sample[$i];
}

//echo 'seleksi tourny<br>';
$seleksi = $newPop;
while(count($seleksi) < $jmlSeleksi)
{
    $seleksi[] = tourny($sample);
}
while(count($newPop) < $jmlPop)
{
    //echo 'crossover<br>';
    $ayah = $seleksi[array_rand($seleksi,1)];
    $ibu = $seleksi[array_rand($seleksi,1)];
    $anak = array($ayah, $ibu);
    if( (rand(0,100) / 100) < $probCross )
    {
        $anak = crossover($ayah, $ibu);
    }
    //echo 'mutasi<br>';
    foreach($anak as $keturunan)
    {
        if(count($newPop) < $jmlPop)
        {
            if( (rand(0,100) / 100) < $probMutasi )
            {
                $newPop[] = mutasi($keturunan);
            }else{
                $newPop[] = $keturunan;
            }
        }else{
            break;
        }
    }
}

```

```

        }

    }

    usort($newPop, 'cmp');
    $sample = $newPop;
    //echo $generasi." fitness: ".$fitness($sample[0])."\n";
}

//echo $generasi." fitness: ".$fitness($sample[0])."\n";
return array($sample[0], $generasi, fitness($sample[0]));
}

$sample = array();
for($i=0;$i<10;$i++){
    $sample[] = run((int)$_POST['iterasi']); //masukkan iterasi
}

echo '<h3>Optimasinya</h3>';
echo '<table border="1">
<tr>
<th align="center">RUN</th>
<th align="center">ROW</th>
<th align="center">COL</th>
<th align="center">ITERASI</th>
<th align="center">NEIGHBOUR WEIGHT</th>
</tr>';
foreach($sample as $key=>$val)
{
    $r = $val[0];
    echo '<tr>
<td align="center">' . ($key + 1) . '</td>
<td align="center">' . $r[0] . '</td>
<td align="center">' . $r[1] . '</td>
<td align="center">' . $val[1] . '</td>
<td align="center">' . ($val[2] - 1) . '</td>
</tr>';
}
echo '</table>';

```

```

echo '<h3>Detailnya</h3>';
foreach($sample as $key=>$val)
{
    $r = $val[0];
    echo 'Ketemu (x,y) = ('.$r[0].','.$r[1].)';
    echo '<table border="1">';
    for($i=-1;$i<2;$i++)
    {
        if($r[0]+$i < 1 || $r[0]+$i > $jmlBaris-1){
            continue;
        }

        echo '<tr>';
        for($j=-1;$j<2;$j++)
        {
            if($r[1]+$j < 1 || $r[1]+$j > $jmlKolom-1){
                continue;
            }

            echo '<td>' . $data[$r[0]+$i][$r[1]+$j] . '</td>';
        }
        echo '</tr>';
    }
    echo '</table>';
}

```

DAFTAR RIWAYAT HIDUP

Penulis bernama lengkap Prasetyo Yuliantoro. Lahir di Yogyakarta, 20 Juli 1992, merupakan anak kedua dari dua bersaudara. Dibesarkan di kota Yogyakarta, penulis memulai pendidikan formal di SDN Kabregan pada tahun 1998, kemudian melanjutkan pendidikan di SMPN 1 Piyungan pada tahun 2004, dan pada jenjang berikutnya melanjutkan pendidikan di SMAN 1 Kalasan hingga lulus pada tahun 2010. Setamat dari SMA, penulis meneruskan pendidikan di D3 Jurusan Teknik Telekomunikasi, Universitas Telkom Bandung pada tahun 2010 sampai 2013. Pada tahun yang sama melanjutkan tingkat S1 di Institut Teknologi Sepuluh Nopember dengan jurusan Teknik Elektro dengan Prodi Telekomunikasi. Setamat dari sarjana, penulis melanjutkan pendidikan pasca sarjana strata dua (S2) di Institut Teknologi Sepuluh Nopember dengan bidang studi Teknik Telekomunikasi Multimedia. Penulis dapat dihubungi melalui alamat email : prasetyo.saka@gmail.com

