



TUGAS AKHIR - SM141501

**IMPLEMENTASI METODE PEWARNAAN GRAF
MENGUNAKAN ALGORITMA WELCH
POWELL UNTUK SIMULASI PENERAPAN
FREKUENSI RADIO DI JAWA TIMUR**

**FADHLAN SEPTIANTO EKA PUTRA
NRP 1213 100 110**

**Dosen Pembimbing
Dr. Darmaji, S.Si, MT
Drs. Soetrisno, Ml.Komp.**

**DEPARTEMEN MATEMATIKA
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember
Surabaya
2017**

“Halaman ini sengaja dikosongkan”



FINAL PROJECT - SM141501

**IMPLEMENTATION OF GRAPH COLORING
METHOD USING WELCH POWELL
ALGORITHM FOR SIMULATION OF RADIO
FREQUENCY IN EAST JAVA**

**FADHLAN SEPTIANTO EKA PUTRA
NRP 1213 100 110**

Supervisors

**Dr. Darmaji, S.Si, MT
Drs. Soetrisno, Ml.Komp.**

**DEPARTMENT OF MATHEMATICS
Faculty of Mathematics and Science
Institute Technology of Sepuluh Nopember
Surabaya
2017**

“Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN

**IMPLEMENTASI METODE PEWARNAAN GRAF
MENGUNAKAN ALGORITMA WELCH POWELL UNTUK
SIMULASI PENERAPAN FREKUENSI RADIO DI JAWA TIMUR**

**IMPLEMENTATION OF GRAPH COLORING METHOD USING
WELCH POWELL ALGORITHM FOR SIMULATION OF RADIO
FREQUENCY IN EAST JAVA**

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika dan Ilmu Pengetahuan Alam
Institut Teknologi Sepuluh Nopember Surabaya

FADHLAN SEPTIANTO EKA PUTRA
NRP. 1213 100 110

Menyetujui,

Dosen Pembimbing II,

Dosen Pembimbing I,

Drs. Soetrisno, MI, Komp.

Dr. Darmah, S.Si, MT

NIP. 19571103 198603 1003

Mengetahui,

NIP. 1961015 199402 1 001

Kepala Departemen Matematika
FMIPA ITS

Dr. Imam Mukhlash, S.Si, MT

NIP. 19700831 199403 1 003

Surabaya, Juli 2017

“Halaman ini sengaja dikosongkan”

**IMPLEMENTASI METODE PEWARNAAN GRAF
MENGUNAKAN ALGORITMA WELCH POWELL
UNTUK SIMULASI PENERAPAN FREKUENSI RADIO
DI JAWA TIMUR**

Nama Mahasiswa : Fadhlan Septianto Eka Putra
NRP : 1213 100 110
Departemen : Matematika
Dosen Pembimbing : Dr. Darmaji, S.Si, MT
Drs. Soetrisno, MI.Komp.

Abstrak

Interferensi frekuensi radio merupakan permasalahan umum yang di hadapi oleh instansi pemerintah dalam pengelolaan frekuensi radio FM. Interferensi menjadi kelemahan dalam pengelolaan frekuensi radio FM dimana permasalahan yang terjadi adalah adanya ‘tubrukan’ pada dua stasiun radio dengan frekuensi yang sama. Interferensi memberikan dampak negatif yang signifikan terhadap kinerja jaringan nirkabel. Akibatnya penerimaan suara pada stasiun radio FM di Jawa Timur yang mengalami interferensi menghasilkan kualitas suara yang buruk. Pada penelitian ini, dilakukan penggunaan algoritma welch powell dalam menyelesaikan masalah interferensi dan mengoptimalkan pengalokasian frekuensi radio FM. Hasil dari penelitian ini menunjukkan Algoritma Welch Powell dapat menyelesaikan masalah interferensi dan mengoptimalkan pengalokasian frekuensi di Jawa Timur tanpa adanya frekuensi yang bertubrukan, dengan menggunakan data real stasiun radio FM yang ada di Jawa Timur, Algoritma Welch Powell menghasilkan pengalokasian frekuensi yang lebih optimal. Karena jumlah frekuensi yang dihasilkan dengan algoritma Welch Powell lebih minimum dibandingkan jumlah frekuensi yang digunakan data real.

Kata Kunci : Interferensi, Pewarnaan Graf, Algoritma Welch Powell

**IMPLEMENTATION OF GRAPH COLORING METHOD
USING WELCH POWELL ALGORITHM FOR
SIMULATION OF RADIO FREQUENCY IN EAST JAVA**

Abstract

Name of Student : Fadhlan Septianto Eka Putra
NRP : 1213 100 110
Department : Mathematics
Supervisors : Dr. Darmaji, S.Si, MT
Drs. Soetrisno, MI.Komp.

Radio frequency interference is a common problem faced by government agencies in FM radio frequency management. Interference becomes a weakness in FM radio frequency management where the problem that occurs is the existence of 'collision' on two radio stations with the same frequency. Interference has a significant negative impact on the performance of wireless networks. As a result, voice reception at FM radio stations in East Java experiencing interference produces poor sound quality. In this research, we used the welch powell algorithm to solve the interference problem and to optimize the FM radio frequency allocation. The results of this study show that Welch Powell algorithm can solve interference problem and optimize frequency allocation in East Java without any collision frequency, using real data of FM radio station in East Java, Welch Powell algorithm resulted in a more optimal frequency allocation. Since the number of frequencies generated with the Welch Powell algorithm is less than the number of frequencies used in real data.

Keywords : *Interference, Coloring Graph, Welch Powell Algorithm*

“Halaman ini sengaja dikosongkan”

KATA PENGANTAR

Segala puji syukur penulis panjatkan ke hadirat Allah SWT, karena dengan ridlo-Nya penulis dapat menyelesaikan Tugas Akhir yang berjudul

“IMPLEMENTASI METODE PEWARNAAN GRAF MENGUNAKAN ALGORITMA WELCH POWELL UNTUK SIMULASI PENERAPAN FREKUENSI RADIO DI JAWA TIMUR”

yang merupakan salah satu persyaratan akademis dalam menyelesaikan Program Sarjana Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Institut Teknologi Sepuluh Nopember Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika ITS.
2. Dr. Darmaji, S.Si, MT selaku Dosen Wali yang telah memberikan arahan akademik selama penulis menempuh pendidikan di Departemen Matematika ITS.
3. Dr. Darmaji, S.Si, MT selaku Dosen Pembimbing 1 dan Drs. Soetrisno, MI.Komp selaku Dosen Pembimbing 2 yang telah memberikan bimbingan dan motivasi kepada penulis dalam mengerjakan Tugas Akhir ini sehingga dapat terselesaikan dengan baik.
4. DR. Budi Setiyono, S.Si, MT selaku Dosen Penguji 1, Drs. I Gst Ngr Rai Usadha, M.Si selaku Dosen Penguji 2, dan Dr. Dwi Ratna Sulistyaningrum, S.Si, MT selaku Dosen Penguji 3 yang telah memberikan kesempatan kepada penulis untuk dapat menyelesaikan tugas akhir ini.
5. Seluruh jajaran Dosen dan Staff Departemen Matematika ITS.

6. Kedua orang tua, serta keluarga yang senantiasa memberikan bantuan, dukungan serta do'a yang tak terhingga.
7. Kepada adik saya, Adhifa Putri Lestari yang senantiasa memberikan motivasi dalam menyelesaikan tugas akhir saya.
8. Diput, Bhara, Jonbon dan Chyntia yang selalu memberi semangat selama pengerjaan Tugas Akhir ini.
9. Chyntia Kumalasari Puteri yang telah berjuang bersama-sama sejak awal penyusunan Tugas Akhir ini.
10. Teman-teman angkatan 2013 yang saling mendukung dan memotivasi.
11. Farhan, Fintanto, Oby, Ardi dan Gede selaku sahabat perjuangan yang senantiasa membantu dan memotivasi penulis dalam menyelesaikan Tugas Akhir ini.
12. Semua pihak yang tak bisa penulis sebutkan satu-persatu, terima kasih telah membantu sampai terselesaikannya Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih jauh dari kesempurnaan. Oleh karena itu, penulis mengharapkan kritik dan saran dari pembaca. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang berkepentingan.

Surabaya, Juli 2017

Penulis

DAFTAR ISI

HALAMAN JUDUL.....	i
LEMBAR PENGESAHAN ... Error! Bookmark not defined.	
KATA PENGANTAR	xxiii
DAFTAR ISI.....	xxv
DAFTAR GAMBAR.....	xxvii
DAFTAR TABEL.....	xxix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Batasan Masalah	5
1.4 Tujuan	5
1.5 Manfaat	6
1.6 Sistematika Penulisan Tugas Akhir	6
BAB II TINJAUAN PUSTAKA	9
2.1 Teori Graf.....	9
2.2 Terminologi Graf	10
2.3 Pewarnaan Graf.....	11
2.4 Pewarnaan Simpul.....	11
2.5 Algoritma <i>Welch Powell</i>	15
2.6 Spektrum Frekuensi Radio	19
2.7 Gelombang Radio	19
2.8 Frekuensi Radio FM.....	20
BAB III METODE PENELITIAN	23
3.1 Tahapan Penelitian	23
3.2 Diagram Alir Penelitian	25
BAB IV PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK	29
4.1 Analisis Masalah Interferensi Radio	29
4.2 Deskripsi Umum Perangkat Lunak	30
4.3 Analisis Kebutuhan Perangkat Lunak.....	30

4.4	Desain Sistem Perangkat Lunak.....	32
4.3.1	Waterfall Diagram.....	32
4.3.2	Usecase Diagram.....	34
4.3.3	Activity Diagram.....	35
4.3.4	Data Flow Diagram.....	36
4.3.5	Class Diagram.....	37
4.4	Desain Antarmuka Perangkat Lunak.....	38
4.4.1	Desain Halaman Utama.....	38
4.4.2	Desain Login <i>User</i>	39
4.4.3	Desain Halaman Home.....	40
4.4.4	Desain Representasi Graf.....	40
4.4.5	Desain Pewarnaan Graf.....	41
4.4.6	Desain Rekomendasi Frekuensi Stasiun Radio FM.....	42
4.5	Implementasi Program.....	43
4.5.1	Tampilan Halaman Pertama.....	43
4.5.2	Tampilan Login.....	44
4.5.3	Tampilan Halaman Utama.....	45
4.5.4	Tampilan Representasi Graf.....	49
4.5.5	Tampilan Pewarnaan Graf.....	50
4.5.6	Tampilan Rekomendasi Frekuensi.....	55
BAB V PENGUJIAN DAN PEMBAHASAN HASIL.....		61
5.1.	Pembahasan.....	61
5.2.	Pengujian Program.....	66
5.2.1	Pengujian dengan Kasus 1.....	66
5.2.2	Pengujian dengan Kasus 2.....	70
5.2.3	Pengujian dengan Kasus 3.....	74
5.2.4	Pengujian dengan Data Real.....	78
BAB VI PENUTUP.....		85
6.1	Kesimpulan.....	85
6.2	Saran.....	85
DAFTAR PUSTAKA.....		87
LAMPIRAN A.....		89
LAMPIRAN B.....		96

DAFTAR GAMBAR

Gambar 2. 1. Contoh Graf Berbobot dan Terhubung.....	11
Gambar 2. 2 Contoh Pewarnaan Simpul.....	12
Gambar 2. 3. Sebuah Graf dengan 5 Simpul.....	13
Gambar 2. 4 Graf yang Diwarnai dengan 5 Warna.....	13
Gambar 2. 5. Graf yang Diwarnai dengan 4 Warna.....	14
Gambar 2. 6 Graf yang Memiliki $\chi(G) = 3$	15
Gambar 2. 7. Flowchart Algoritma Welch Powell.....	17
Gambar 2. 8. Contoh pewarnaan simpul dengan menggunakan pewarnaan simpul	18
Gambar 2. 9. Frekuensi Radio FM.....	21
Gambar 3. 1 Diagram Alir Metode Penelitian	26
Gambar 3. 2 Diagram Alir Implementasi.....	27
Gambar 4. 1 Waterfall Diagram.....	32
Gambar 4. 2 Usecase Diagram.....	34
Gambar 4. 3 Activity Diagram.....	35
Gambar 4. 4 Data Flow Diagram Level 0.....	36
Gambar 4. 5. Data Flow Diagram Level 1	37
Gambar 4. 6. Class Diagram	38
Gambar 4. 7 Desain Antarmuka Halaman Utama.....	39
Gambar 4. 8 Desain Login User.....	39
Gambar 4. 9 Desain Antarmuka Halaman Home.....	40
Gambar 4. 10 Desain Antarmuka Representasi Graf	41
Gambar 4. 11 Desain Antarmuka Pewarnaan Graf	42
Gambar 4. 12 Desain Antarmuka Halaman Hasil Rekomendasi Pewarnaan Graf	43
Gambar 4. 13 Tampilan Halaman Utama	44
Gambar 4. 14 Tampilan Login.....	44
Gambar 4. 15 Program Halaman Utama	46
Gambar 4. 16. Flowchart Input Graf.....	47
Gambar 4. 17 Program Representasi Graf	49
Gambar 4. 18 Program Pewarnaan Graf	51
Gambar 4. 19. Flowchart Pewarnaan Graf.....	52

Gambar 4. 20 Program Rekomendasi Frekuensi	55
Gambar 4. 21. Flowchart Penentuan Frekuensi Radio FM...	56
Gambar 5. 1. Posisi Stasiun Radio Tes Manual.....	61
Gambar 5. 2. Model Graf Tes Manual	62
Gambar 5. 3. Gambar Graf Sebelum di Welch Powell.....	62
Gambar 5. 4. Gambar Graf Tahap Pertama	63
Gambar 5. 5. Gambar Graf Tahap Kedua	63
Gambar 5. 6. Gambar Graf Tahap Ketiga.....	64
Gambar 5. 7. Gambar Graf Tahap Keempat	64
Gambar 5. 8. Pewarnaan Graf Tes Manual.....	65
Gambar 5. 9. Frekuensi Radio Tes Manual	65
Gambar 5. 10 Input Kasus 1	67
Gambar 5. 11 Representasi Graf Kasus 1	67
Gambar 5. 12 Program Pewarnaan Graf Kasus 1	69
Gambar 5. 13 Program Rekomendasi Frekuensi Kasus 1	70
Gambar 5. 14 Input Kasus 2	71
Gambar 5. 15 Representasi Graf Kasus 2	72
Gambar 5. 16 Pewarnaan Graf Kasus 2	73
Gambar 5. 17 Rekomendasi Frekuensi Kasus 2.....	74
Gambar 5. 18 Input Kasus 3	75
Gambar 5. 19 Representasi Graf Kasus 3	76
Gambar 5. 20 Pewarnaan Graf Kasus 3	77
Gambar 5. 21 Rekomendasi Frekuensi Kasus 3.....	78
Gambar 5. 22 Input Kasus Data Real.....	79
Gambar 5. 23 Representasi Graf Kasus Data Real	80
Gambar 5. 24 Pewarnaan Graf Kasus Data Real	81

DAFTAR TABEL

Tabel 2. 1 Jenis Bidang Frekuensi Radio.....	20
Tabel 4. 1 Kebutuhan Fungsional	31
Tabel 4. 2 Kebutuhan Non Fungsional	31
Tabel 5. 1 Rekomendasi Frekuensi Kasus Data Real	82

BAB I

PENDAHULUAN

1.1 Latar Belakang

Radio adalah salah satu media informasi yang masih digunakan sampai saat ini. Di provinsi Jawa Timur terdapat stasiun berjumlah 115 radio FM [8]. Interval frekuensi radio FM yang terbatas, menuntut pemerintah harus dapat mengatur pengalokasian frekuensi radio FM agar tidak terjadi interferensi. Terdapat radio yang mempunyai frekuensi sama namun berbeda kota, yaitu radio RRI Surabaya dan radio Purnama Cipta Prawira di Blitar. Kedua radio tersebut mempunyai frekuensi yang sama yaitu pada frekuensi 95.30 dan kedua radio tersebut berjarak 100 Km. Hal tersebut dapat mengakibatkan interferensi pada radius yang bertubrukan. Jangkauan frekuensi radio FM tidak sejauh frekuensi radio AM. Jangkauan frekuensi radio AM mencapai 120 Km dikarenakan bantuan oleh lapisan ionosfer (berada diantara 80 Km sampai 100 Km dari permukaan bumi). Namun frekuensi radio FM dapat mencapai jarak maksimum *Coverage Area* 60 Km sampai 70 Km [8]. Terdapat gelombang frekuensi FM yang dihantarkan melalui lapisan troposfer, sehingga siaran FM bisa diterima sampai jarak yang jauh [8]. Dengan jarak *Coverage Area* yang mencapai 60 Km. Terdapat kemungkinan terjadinya interferensi pada daerah tertentu. Interferensi adalah sebuah emisi koheren yang memiliki konten spektral relatif sempit, misalnya radio emisi dari pemancar lain di sekitar frekuensi yang sama dan menghambat penerimaan sinyal. Definisi dari spektral relatif sempit adalah terbatasnya panjang spektrum frekuensi pada frekuensi FM.

Spektrum frekuensi radio adalah susunan pita frekuensi radio yang mempunyai frekuensi lebih kecil dari 3000 GHz. Spektrum frekuensi radio merambat dan terdapat dalam dirgantara (ruang udara dan antariksa). Spektrum frekuensi radio merupakan sumber daya alam terbatas, namun saat ini peminatnya semakin meningkat sementara jumlah ketersediaan spektrum tidak bertambah. Spektrum frekuensi radio digunakan untuk pemancar frekuensi radio. Dalam hal penggunaan spektrum frekuensi radio diatur oleh badan balai monitoring frekuensi radio FM. Interferensi memberikan dampak negatif yang signifikan terhadap kinerja jaringan nirkabel. Dalam hal ini penulis menganalisis masalah interferensi dari sudut pandang teori graf dengan menggunakan metode pewarnaan graf.

Optimalisasi frekuensi radio adalah mengoptimalkan pengalokasian frekuensi radio FM sehingga stasiun pemancar radio sebanyak-banyaknya, dengan jumlah pemakaian alokasi frekuensi sesedikit mungkin. Optimalisasi frekuensi radio bertujuan menghindari interferensi frekuensi radio. Permasalahan interferensi radio di jawa timur adalah terdapat stasiun radio FM yang mempunyai frekuensi yang sama dan menyebabkan terjadinya tubrukan radius pancaran frekuensi radio di daerah tersebut. Dampak dari interferensi tersebut yaitu dapat mengakibatkan *noise* pada frekuensi radio. *Noise* adalah sinyal-sinyal yang tidak diinginkan yang selalu ada dalam suatu sistem transmisi. *Noise* mengakibatkan gangguan kualitas sinyal untuk proses penerimaan dan pengiriman suara [2]. Metode pewarnaan graf dapat diimplementasikan ke dalam permasalahan interferensi frekuensi radio di jawa timur. Dengan mensimulasikan pengguna radio yang relatif

terhadap posisi stasiun radio diasumsikan sebagai node dan interferensi yang terjadi diasumsikan menjadi lintasan. Pada penelitian ini penulis menggunakan algoritma Welch Powell untuk optimalisasi frekuensi radio agar tidak ada frekuensi sama yang mengalami interferensi. Algoritma Welch Powell adalah teknik pewarnaan simpul, dimana simpul yang bertetangga akan memiliki warna yang berbeda. Warna merepresentasikan frekuensi yang dimiliki suatu pemancar.

Pewarnaan graf pada penelitian ini menggunakan pewarnaan simpul. Pewarnaan simpul adalah pemberian warna terhadap simpul sedemikian sehingga dua simpul yang berdampingan mempunyai warna yang berlainan. Jumlah minimum warna yang dibutuhkan disebut bilangan khromatik dari graf G , dimana khromatik dari graf G adalah jumlah warna minimum yang diperlukan untuk mewarnai graf G , dan dilambangkan dengan $\chi(G)$. Dengan menggunakan Algoritma Welch Powell yang digunakan dalam pewarnaan graf ternyata dapat diterapkan pada masalah penggunaan frekuensi radio dan dapat menghindari interferensi yang terjadi di spektrum frekuensi radio FM.

Permasalahan interferensi tidak hanya terjadi pada jaringan nirkabel seperti telepon seluler, namun juga terjadi pada jaringan nirkabel siaran radio. Maaly A. Hassan dan Andrew Chickadel pada tahun 2011 melakukan penelitian yang berjudul "*A Review of Interference Reduction in Wireless Networks Using Graph Coloring Methods*". Pada penelitian tersebut dibahas mengenai Interferensi yang terjadi pada jaringan nirkabel. Penggunaan beberapa metode seperti metode *weighted coloring*, metode *dynamic channel assignment*, metode *location-oblivious* untuk menyelesaikan permasalahan interferensi. Dalam pembahasan penelitian

tersebut dijelaskan bahwa metode *dynamic channel assignment* adalah metode yang optimal dalam pembahasan penelitian tersebut. Selanjutnya, pada tahun 2010, telah dilakukan penelitian oleh Margaretha Siahaan yang berjudul “Studi Pewarnaan Graf pada Radio *Frequency Assignment Problem*”. Pada penelitian ini dibahas mengenai sebuah model *frequency assignment problem* untuk mengoptimalkan bidang frekuensi dengan menggunakan pewarnaan simpul dan teknik kromatik polinomial dengan mencari jumlah warna paling sedikit untuk pewarnaan graf. *Frequency Assignment Problem* pada dasarnya merupakan cara untuk mencari jarak terkecil antara dua buah pemancar agar pemancar tersebut tidak saling berinterferensi dan dapat memancarkan gelombang melalui saluran-saluran yang tersedia. *Frequency assignment problem* pertama kali muncul di tahun 1960. Perkembangan dari layanan nirkabel seperti telepon selular pertama memulai proses kelangkaan frekuensi dalam spectrum radio. *Frequency Assignment Problem* yang juga dikenal sebagai *channel assignment problem*, berkembang dengan cukup cepat. Teknik pewarnaan graf dapat diimplementasikan untuk menemukan nomor warna yang paling kecil yang menghubungkan kedua pemancar yang direpresentasikan sebagai node.

Berdasarkan penelitian diatas, penerapan pewarnaan graf sudah banyak digunakan dalam kehidupan sehari-hari. Oleh karena itu, pada Tugas Akhir ini dilakukan menggunakan pewarnaan graf menggunakan Algoritma *Welch Powell* untuk simulasi interferensi frekuensi radio di Jawa Timur. Sehingga dapat menghindari interferensi frekuensi radio di Jawa Timur.

1.2 Rumusan Masalah

Berdasarkan latar belakang tersebut, maka permasalahan yang dirumuskan dalam Tugas Akhir ini adalah sebagai berikut :

1. Bagaimana mengimplementasikan pewarnaan graf untuk menghindari terjadinya interferensi frekuensi radio di Jawa Timur.
2. Bagaimana mengoptimalkan slot frekuensi radio FM menggunakan pewarnaan graf dengan Algoritma Welch Powell yang relatif terhadap posisi dan *coverage area* stasiun radio FM di Jawa Timur.

1.3 Batasan Masalah

Pada penelitian ini, penulis membuat batasan masalah sebagai berikut :

1. Tidak dilakukan proses perbandingan dengan metode lainnya.
2. Gelombang radio merambat pada bidang datar.
3. Graf yang dihasilkan adalah graf sederhana.

1.4 Tujuan

Berdasarkan permasalahan yang telah dirumuskan sebelumnya, tujuan penelitian Tugas Akhir ini adalah :

1. Mengimplementasikan pewarnaan graf untuk menghindari terjadinya interferensi frekuensi radio di Jawa Timur.
2. Mengoptimalkan slot frekuensi radio FM menggunakan pewarnaan graf dengan Algoritma Welch Powell yang relatif terhadap posisi dan *coverage area* stasiun radio FM di Jawa Timur.

1.5 Manfaat

Setelah diperoleh peninjauan penurunan interferensi siaran radio di Jawa Timur, maka Tugas Akhir ini dapat memberikan manfaat sebagai berikut :

1. Diperoleh suatu perangkat lunak untuk mengurangi terjadinya interferensi gelombang radio di Jawa Timur.
2. Sebagai salah satu referensi penggunaan pewarnaan graf untuk permasalahan penggunaan frekuensi secara lebih efisien tanpa terjadi interferensi antara dua pemancar atau lebih.

1.6 Sistematika Penulisan Tugas Akhir

Sistematika dari penulisan Tugas Akhir ini adalah sebagai berikut :

1. BAB I PENDAHULUAN

Bab ini menjelaskan tentang gambaran umum dari penulisan Tugas Akhir ini yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, manfaat penelitian, dan sistematika penulisan.

2. BAB II TINJAUAN PUSTAKA

Bab ini menjelaskan tentang tinjauan pustaka dari referensi penunjang serta penjelasan permasalahan yang dibahas dalam Tugas Akhir ini, meliputi Pengertian Graf, Terminologi Graf, Jenis-jenis Graf, Pewarnaan Graf, Bilangan Kromatik, Algoritma *Welch Powell*, dan Kompleksitas Algoritma.

3. BAB III METODOLOGI PENELITIAN

Bab ini berisi metodologi atau urutan pengerjaan yang dilakukan dalam menyelesaikan Tugas Akhir, meliputi studi literatur, pengumpulan data, pengolahan data,

implementasi, pengujian, penarikan simpulan dan saran, serta penyusunan laporan tugas akhir.

4. **BAB IV PERANCANGAN DAN IMPLEMENTASI**

Bab ini menjelaskan mengenai perancangan dan implementasi sistem, proses pembuatan sistem secara utuh sehingga terbentuk simulasi interferensi frekuensi radio di Jawa Timur.

5. **BAB V PENGUJIAN DAN PEMBAHASAN HASIL**

Bab ini akan menampilkan hasil uji coba serta pembahasan terkait simulasi interferensi frekuensi radio di Jawa Timur yang telah dibuat..

6. **BAB VI PENUTUP**

Bab ini berisi kesimpulan yang diperoleh dari pembahasan masalah sebelumnya serta saran yang diberikan untuk pengembangan selanjutnya .

BAB II TINJAUAN PUSTAKA

2.1 Teori Graf

Teori graf merupakan pokok bahasan yang memiliki banyak terapan dalam kehidupan sehari-hari sampai saat ini. Graf diperkenalkan pertama kali pada tahun 1736 melalui tulisan *Leonhard Euler* yang berisi tentang upaya pemecahan masalah jembatan *Königsberg* yang sangat terkenal di Eropa.

Pada abad ke-18, ada tujuh jembatan yang menghubungkan 4 daerah di Königsberg, yang sekarang bernama Kaliningrad, sebuah kota di Rusia. Kota ini merupakan tempat tinggal para bangsawan abad ke-16. Sungai Pregel membagi Königsberg menjadi empat daerah karena mengalir tepat melalui kota. Penduduk setempat berusaha mencari tahu apakah ada cara yang memungkinkan mereka untuk menjalani seluruh jembatan dari sebuah titik awal dari sebuah daerah tanpa harus melewati jembatan yang sama lebih dari sekali. Solusi Euler diberikan dengan merepresentasikan jembatan dalam kota Königsberg sebagai sebuah graf dengan keempat daerah sebagai simpulnya dan tujuh jembatan sebagai tujuh sisi graf. Euler menyatakan bahwa ketujuh jembatan tidak mungkin dilalui sekali saja karena derajat seluruh simpulnya tidak genap. Ini dianggap sebagai literature pertama yang menyatakan teori graf.

Graf didefinisikan sebagai pasangan himpunan sehingga ditulis dengan notasi $G = (V, E)$, dimana $V =$ himpunan tidak kosong dari simpul-simpul (*vertices* atau *node*) dan $E =$ himpunan sisi (*edges* atau *arcs*) yang menghubungkan sepasang simpul dalam V [5]. Secara

sederhana graf merupakan kumpulan simpul-simpul yang dihubungkan dengan garis atau busur.

Setiap sisi berhubungan dengan satu atau dua simpul. Dua buah simpul dikatakan berhubungan atau bertangga (*adjacent*) jika ada sisi yang menghubungkan keduanya. Berdasarkan orientasi yang ada pada sisinya, graf dapat dikelompokkan menjadi dua yaitu graf berarah (*direct graph*) yaitu graf yang setiap sisinya diberikan arah sehingga untuk dua simpul v_i dan v_j , maka $(v_i, v_j) \neq (v_j, v_i)$ dan graf tak berarah (*undirected graph*) yaitu graf yang sisinya tidak mengandung arah sehingga untuk dua simpul v_i dan v_j maka $\{v_i, v_j\} = \{v_j, v_i\}$. Selain itu juga dikenal graf berbobot yaitu graf yang sisinya memiliki bobot atau nilai.

2.2 Terminologi Graf

Terminologi graf adalah istilah dari beberapa sifat graf yang digunakan dalam penelitian tugas akhir ini. Berikut akan dijelaskan mengenai terminologi (istilah) yang berkaitan dengan graf dan pewarnaan graf.

a. Bertetangga (*adjacent*)

Dua buah simpul pada graf tak berarah dikatakan bertetangga jika keduanya dihubungkan dengan sebuah sisi, dengan kata lain, u bertetangga dengan v jika (u, v) adalah sebuah sisi pada graf G .

b. Berinsidensi (*incident*)

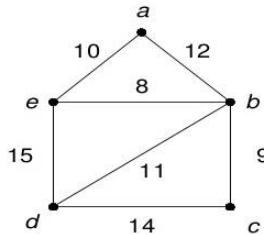
Untuk sembarang sisi $e = (u, v)$, sisi e dikatakan berinsidensi dengan simpul u dan simpul v .

c. Derajat (*degree*)

Derajat suatu simpul pada graf tak berarah adalah jumlah sisi yang berinsidensi dengan simpul tersebut.

d. Terhubung (connected)

Suatu graf disebut graf terhubung jika untuk setiap pasang simpul u dan v dalam himpunan V terdapat lintasan dari u ke v (yang berarti ada lintasan dari v ke u). Jika dua buah simpul terhubung maka pasti simpul yang pertama dapat mencapai simpul yang kedua.



Gambar 2. 1. Contoh Graf Terhubung.

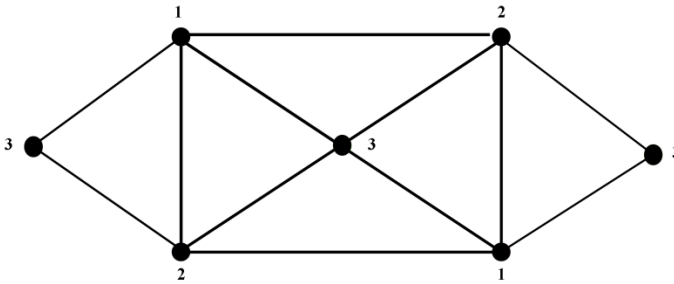
2.3 Pewarnaan Graf

Pewarnaan graf merupakan pemberian warna pada graf sedemikian sehingga graf tersebut memiliki warna yang berbeda-beda pada simpul ataupun lintasan. Ada 3 macam pewarnaan graf, yaitu pewarnaan simpul, pewarnaan sisi, dan pewarnaan wilayah. Berikut akan dibahas pewarnaan simpul saja.

2.4 Pewarnaan Simpul

Pewarnaan sebuah graf sering disebut dengan pewarnaan graf $G = (V, E)$ simpul graf. Pewarnaan simpul pada graf adalah memberi warna pada simpul-simpul suatu graf sedemikian sehingga tidak ada dua simpul bertetangga yang memiliki warna yang sama [1]. Kita dapat memberikan

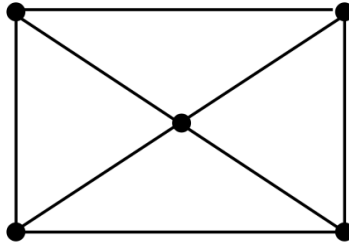
sembarang warna pada simpul-simpul asalkan berbeda dengan simpul-simpul tetangganya. Terminologi dalam menggunakan warna untuk label simpul kembali kepada pemetaan warna. Label warna seperti merah dan biru hanya dipakai ketika jumlah dari warna-warna itu kecil dan normalnya, diketahui bahwa warna-warna tersebut digambar dari integer-integer $\{1,2,3,\dots\}$.



Gambar 2. 2 Contoh Pewarnaan Simpul

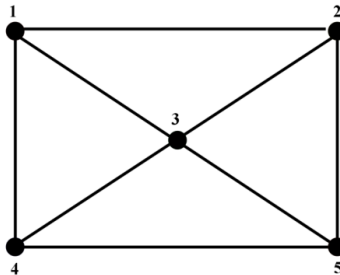
Sebuah pewarnaan yang menggunakan n warna biasanya disebut dengan n -coloring. Ukuran terkecil banyaknya warna yang dapat diberikan kepada sebuah graf G dinamakan dengan bilangan kromatik, Bilangan kromatik merupakan jumlah warna minimum yang dapat digunakan untuk mewarnai simpul, sedemikian hingga tidak ada simpul bertetangga dengan warna sama. Bilangan khromatik ditulis dengan simbol $\chi(G)$. Graf lengkap K_n memiliki $\chi(G) = n$ karena semua simpul saling terhubung satu sama lain.

Contoh lain mengenai bilangan kromatik yaitu terdapat sebuah graf G dengan simpul sebanyak 5. Berikut ini adalah gambar graf G tersebut.



Gambar 2. 3. Sebuah Graf dengan 5 Simpul

Berdasarkan Gambar 2.3 di atas, graf tersebut dapat diwarnai dengan 5 warna yang berbeda, yaitu merah, biru, kuning, hijau, dan magenta. Di bawah ini adalah hasil graf yang telah diwarnai dengan 5 warna yang berbeda.

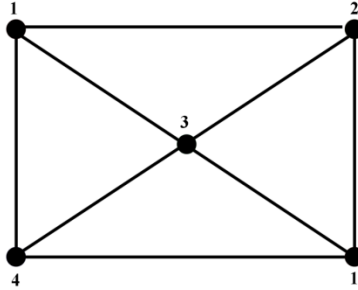


Gambar 2. 4 Graf yang Diwarnai dengan 5 Warna

Lima warna pada Gambar 2. 4 di atas belum bisa disebut sebagai **bilangan kromatik** atau $\chi(G)$ karena terdapat jumlah warna minimum lain yang bisa digunakan untuk mewarnai graf pada Gambar 2. 5 di atas.

Karena jumlah warna minimum lain yang bisa digunakan untuk mewarnai graf tersebut. Maka, graf di atas

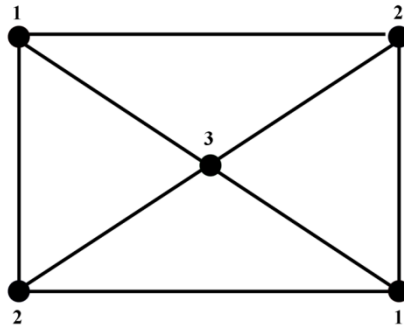
dapat diwarnai dengan 4 warna yang berbeda, yaitu merah, kuning, biru, dan hijau. Berikut ini adalah hasil graf yang telah diwarnai dengan 4 warna yang berbeda.



Gambar 2. 5. Graf yang Diwarnai dengan 4 Warna

Empat warna pada Gambar 2. 5 di atas belum bisa disebut sebagai **bilangan kromatik** atau $\chi(G)$ karena terdapat jumlah warna minimum lain yang bisa digunakan untuk mewarnai graf pada Gambar 2. 6 di bawah.

Pada penjelasan sebelumnya, graf dapat diwarnai dengan jumlah warna yang lebih kecil, yaitu dapat diwarnai dengan 3 warna yang berbeda. tiga warna tersebut yaitu merah, biru, dan hijau. Di bawah ini adalah hasil graf yang telah diwarnai dengan 3 warna yang berbeda.



Gambar 2. 6 Graf yang Memiliki $\chi(G) = 3$

Tiga warna yang berbeda pada Gambar 2. 6 di atas adalah jumlah warna minimum yang dapat digunakan untuk mewarnai simpul tersebut. Sehingga dapat disimpulkan bahwa graf tersebut memiliki **bilangan kromatik** atau $\chi(G) = 3$.

2.5 Algoritma Welch Powell

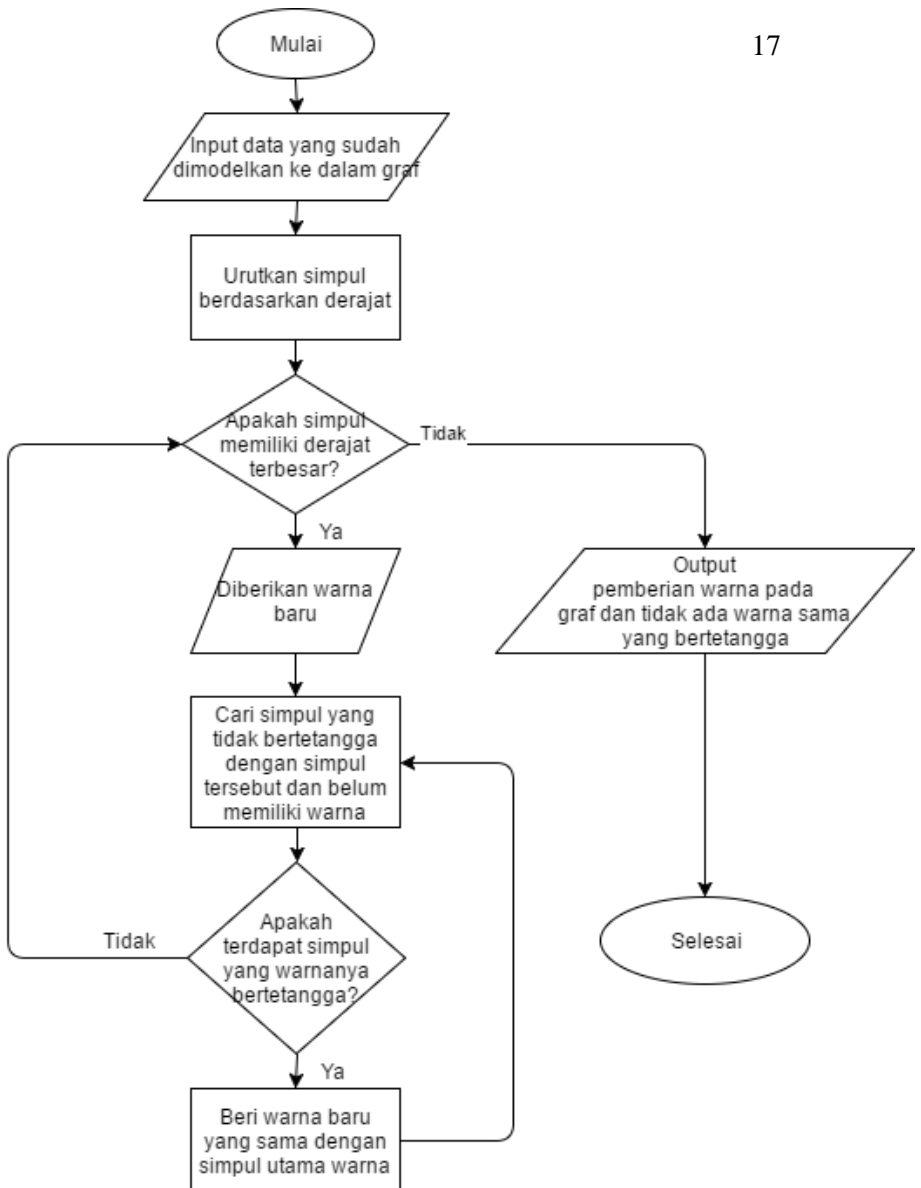
Algoritma ini dapat digunakan untuk mewarnai sebuah graf G secara mangkus. Dengan memberikan batas atas untuk nilai $\chi(G)$, algoritma ini tidak selalu memberikan jumlah warna minimum yang diperlukan.

Langkah-langkah algoritma:

- a. Input posisi stasiun radio dan radius pancaran, yang akan direpresentasikan menjadi simpul.
- b. Urutkan simpul-simpul dari G dalam derajat yang menurun (urutan seperti ini mungkin tidak unik karena beberapa simpul mungkin memiliki derajat yang sama).

- c. Carilah simpul yang memiliki derajat tertinggi untuk diberi warna.
- d. Gunakan satu warna untuk mewarnai simpul pertama (yang mempunyai derajat tertinggi) dan simpul yang tidak bertetangga langsung dengan simpul pertama.
- e. Mulai lagi dengan simpul derajat tertinggi berikutnya di dalam daftar terurut yang belum diwarnai dan ulangi proses pewarnaan simpul dengan warna kedua.
- f. Ulangi ke tahap kedua sampai semua simpul telah diwarnai

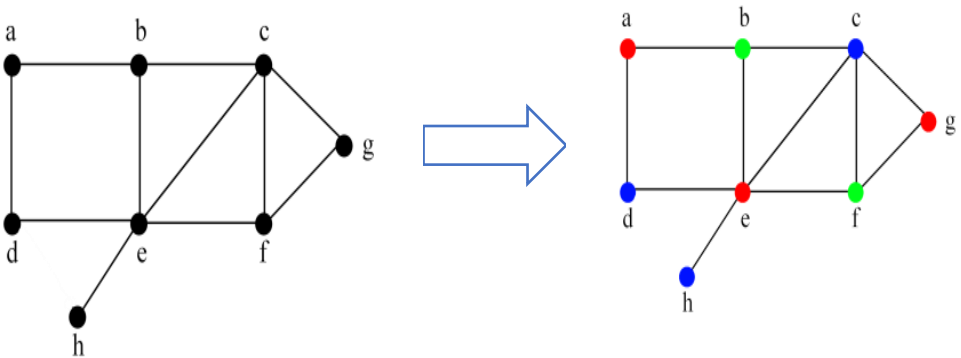
Gambar flowchart algoritma Welch Powell ditunjukkan pada Gambar 2.7.



Gambar 2. 7. Flowchart Algoritma Welch Powell

Salah satu contoh aplikasi teknik pewarnaan simpul dengan menggunakan algoritma Welch Powell.

- (1). Diawali dengan mengurutkan simpul-simpul dari graf G dalam derajat menurun dari yang terbesar, dalam hal ini berturut-turut E, C, F, B, A, D, G, H.
- (2). Menentukan satu warna yaitu merah untuk mewarnai simpul pertama yaitu E dan simpul-simpul lain yang tidak *adjacent* dengan simpul E yaitu G, dan A.
- (3). Dilanjutkan pewarnaan simpul biru untuk derajat tertinggi berikutnya yaitu C dan simpul-simpul lain yang tidak *adjacent* dengan simpul C yaitu D dan H.
- (4). Ulangi langkah tersebut hingga diperoleh pewarnaan hijau yang berarti semua simpul telah diwarnai (lihat Gambar 2.8).



Gambar 2. 8. Contoh pewarnaan simpul dengan menggunakan Algoritma Welch Powell.

2.6 Spektrum Frekuensi Radio

Spektrum frekuensi radio adalah susunan pita frekuensi radio yang mempunyai frekuensi lebih kecil dari 3000 GHz sebagai satuan getaran gelombang elektromagnetik yang merambat dan terdapat dalam dirgantara (ruang udara dan antariksa). Pengalokasian Spektrum Frekuensi Radio di Indonesia mengacu kepada alokasi frekuensi radio internasional untuk region 3. Region 3 (wilayah 3) sesuai dengan peraturan radio yang ditetapkan oleh *International Telecommunication Union* (ITU) atau Himpunan Telekomunikasi Internasional. Penetapan jalur Spektrum frekuensi radio ini bertujuan untuk menghindari terjadinya interferensi (*Interference*) dan untuk menetapkan protokol demi keserasian antara pemancar dan penerima.

2.7 Gelombang Radio

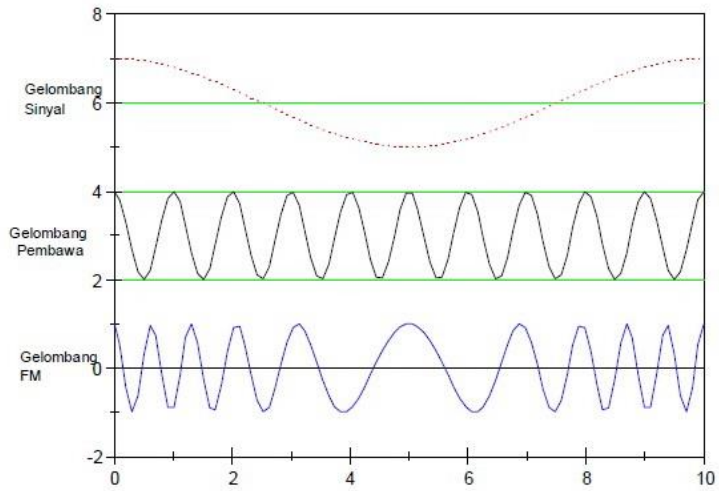
Gelombang radio merupakan gelombang elektromagnetik dengan panjang gelombang lebih dari 10³ meter. Gelombang radio muncul secara alami oleh adanya petir atau oleh objek-objek luar angkasa lainnya. Frekuensi gelombang radio berada di daerah kHz sampai dengan beberapa ratus MHz. Setiap frekuensi radio yang berbeda memiliki karakteristik propagasi yang berbeda di atmosfer bumi. Propagasi adalah proses perambatan gelombang radio dari pemancar ke penerima. Transmisi sinyal dengan media non-kawat memerlukan antenna untuk meradiasikan sinyal radio ke udara bebas dalam bentuk gelombang elektromagnetik.

Tabel 2. 1 Jenis Bidang Frekuensi Radio

Nama Band (Jalur)	Singkatan	Frekuensi	Panjang Gelombang
<i>Tremendously low frequency</i>	TLF	< 3Hz	>100.000 km
<i>Extremely Low Frequency</i>	ELF	3 – 30 Hz	10.000 – 100.000 km
<i>Super Low Frequency</i>	SLF	30 – 300 Hz	1.000 – 10.000 km
<i>Ultra Low Frequency</i>	ULF	300 – 3.000 Hz	100 – 1.000 km
<i>Very Low Frequency</i>	VLF	3 – 30 kHz	10 – 100 km
<i>Low Frequency</i>	LF	30–300 kHz	1 – 10 km
<i>Medium frequency</i>	MF	300 – 3.000 kHz	100 – 1.000 m
<i>High Frequency</i>	HF	3 – 30 MHz	10 – 100 m
<i>Very High Frequency</i>	VHF	30 – 300 MHz	1 – 10 m
<i>Ultra High Frequency</i>	UHF	300 – 3.000 MHz	10 – 100 cm
<i>Super High Frequency</i>	SHF	3 – 30 GHz	1 – 10 cm
<i>Extremely High Frequency</i>	EHF	30 – 300 GHz	1 – 10 mm
<i>Tremendously High Frequency</i>	THF	300 – 3.000 GHz	0.1 – 1 mm

2.8 Frekuensi Radio FM

Frekuensi radio FM adalah proses mengirimkan sinyal Frekuensi rendah dengan menggunakan FM (*Frequency Modulation*). Metode modulasi frekuensi memvariasikan frekuensi untuk mengantarkan gelombang frekuensi radio. Frekuensi radio FM pertama kali ditemukan oleh *Edwin Howard Amstrong* pada tahun 1936, dimana frekuensi radio FM memiliki kelebihan tahan terhadap *noise* dibandingkan Frekuensi AM. Jarak tempuh frekuensi radio FM mencapai 60 Km dari pusat pemancar sinyal. Jarak tempuh frekuensi FM lebih kecil dibandingkan frekuensi AM. Frekuensi radio AM dapat mencapai jarak 100 Km dari pusat pemancar sinyal, dikarenakan sinyal frekuensi radio AM dipantulkan oleh ionosfer [8]. Interval frekuensi radio FM adalah 15 Hz sampai 50 Hz. Frekuensi radio FM dialokasikan pada range frekuensi 88 MHz sampai 108 MHz. Range frekuensi tersebut relatif baik terhadap gangguan atmosfer maupun cuaca.



Gambar 2.9. Frekuensi Radio FM

BAB III

METODE PENELITIAN

Pada bab ini dijelaskan langkah-langkah yang digunakan dalam penyusunan Tugas Akhir. Disamping itu, dijelaskan pula prosedur dan proses pelaksanaan tiap-tiap langkah yang dilakukan dalam menyelesaikan Tugas Akhir.

3.1 Tahapan Penelitian

Langkah-langkah yang digunakan dalam menyelesaikan Tugas Akhir ini yang diringkas dalam Gambar 3.1 adalah sebagai berikut :

1. Studi Literatur

Studi Literatur ini dilakukan untuk identifikasi permasalahan dengan mencari referensi yang menunjang penelitian yang berupa Tugas Akhir, jurnal internasional, buku, maupun artikel yang berhubungan dengan topik Tugas Akhir ini.

2. Pengumpulan Data

Pengumpulan data merupakan tahap untuk mengumpulkan data yang diperlukan dalam pengerjaan Tugas Akhir. Data tersebut berupa frekuensi radio FM di Jawa Timur. Data tersebut digunakan untuk menunjukkan terjadinya interferensi frekuensi radio di Jawa Timur. Dari data tersebut dicari stasiun radio yang memiliki frekuensi sama di Jawa Timur. Stasiun radio FM yang memiliki frekuensi sama akan dihitung jarak antara stasiun radio tersebut, jika stasiun radio yang memiliki frekuensi sama berjarak saling berdekatan. Stasiun radio tersebut mengalami interferensi pada daerah yang bertubrukan radius pancaran.

3. Perancangan & Implementasi

Pada tahap ini penulis membuat implementasi agar mudah dipahami oleh pembaca yang alurnya digambarkan pada Gambar 3.2. Implementasi tersebut dibuat dengan bahasa pemrograman Java dan menggunakan aplikasi NETBEANS.

Pada tahap ini dilakukan proses dimana persoalan diterjemahkan melalui suatu rangkaian aktivitas sesuai model proses, metode, dan alat bantu yang akan digunakan. Tahap-tahap yang dilakukan seperti yang digambarkan pada Gambar 3.2 adalah sebagai berikut :

a. Menentukan posisi simulasi stasiun radio FM

Pada tahap ini ditentukan lokasi stasiun radio FM yang ada di Jawa Timur. Posisi tersebut ditentukan oleh *user* dengan menggunakan simulasi stasiun radio.

b. Menentukan daerah interferensi

Untuk menentukan daerah yang mengalami interferensi. Tahap penentuan posisi stasiun radio FM di Jawa Timur harus sudah ditentukan. Dari posisi stasiun radio FM ditinjau stasiun radio yang radius pancarannya saling bertubrukan.

c. Melakukan simulasi interferensi dalam bentuk graf.

Tahap mensimulasikan interferensi dalam pemetaan graf. Posisi stasiun radio FM akan disimulasikan menjadi simpul dan radius frekuensi yang bertubrukan akan disimulasikan menjadi lintasan.

d. Melakukan prosedur *Coloring Graph*

Setelah melakukan simulasi interferensi dalam bentuk graf. Tahapan selanjutnya menguji algoritma *Welch-Powell* untuk mendapatkan metode pewarnaan

graf. Pada tahapan ini graf yang sudah direpresentasikan dalam bentuk graf akan diwarnai. Pewarnaan graf dengan menggunakan Algoritma Welch-Powell untuk mendapatkan nilai optimal.

e. Representasi hasil implementasi

Hasil pewarnaan graf dengan menggunakan algoritma *Welch-Powell* akan mendapatkan nilai optimal. Nilai optimal yang didapat berupa nilai bilangan khromatik terkecil. Nilai tersebut akan menjadi rekomendasi dan kesimpulan yang digunakan dalam simulasi interferensi frekuensi radio di Jawa Timur.

4. Pengujian

Setelah dibuat implementasi, akan dilakukan pengujian untuk memeriksa apakah hasil implementasi sudah sesuai atau terjadi eror.

5. Kesimpulan dan Saran

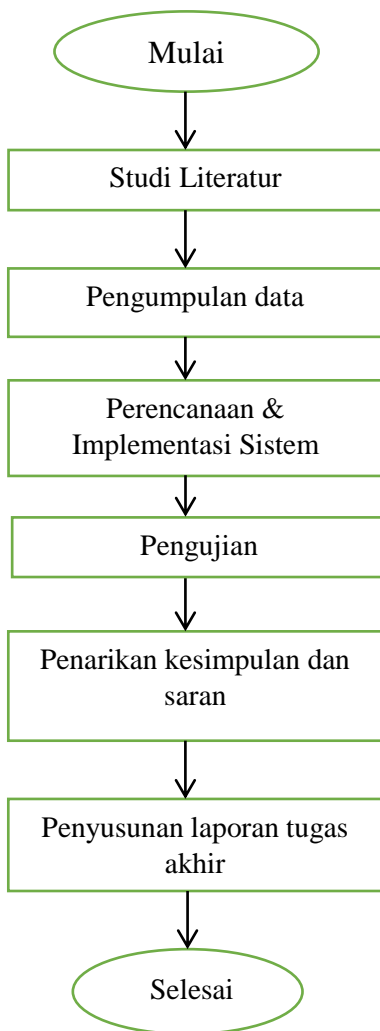
Setelah dilakukan analisis dan pembahasan maka dapat ditarik suatu kesimpulan dan saran sebagai masukan untuk pengembangan penelitian lebih lanjut.

6. Penyusunan Laporan Tugas Akhir

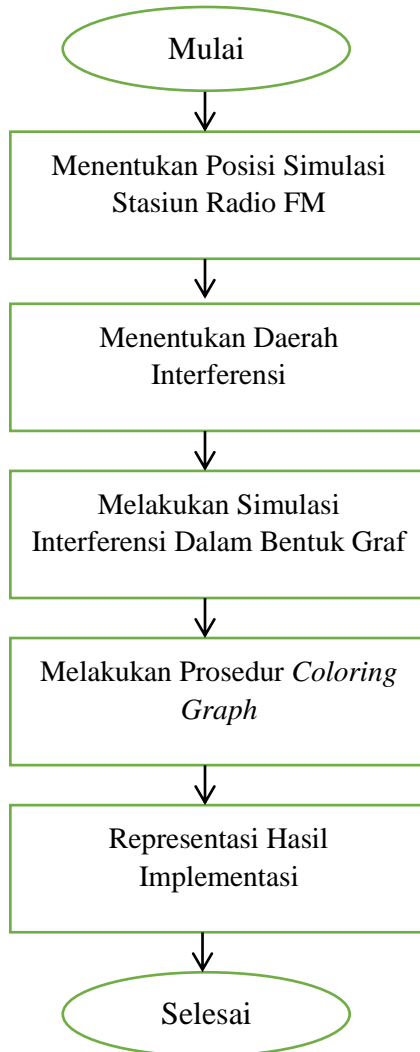
Setelah semua proses selesai dilakukan maka tahap terakhir adalah penyusunan laporan Tugas Akhir.

3.2 Diagram Alir Penelitian

Berdasarkan uraian tersebut diatas, penelitian Tugas Akhir ini dapat dinyatakan dalam diagram alir pada Gambar 3.1.



Gambar 3. 1 Diagram Alir Metode Penelitian



Gambar 3. 2 Diagram Alir Implementasi

BAB IV

PERANCANGAN DAN IMPLEMENTASI PERANGKAT LUNAK

4.1 Analisis Masalah Interferensi Radio

Interferensi merupakan permasalahan yang menjadi tinjauan dalam penelitian ini. Interferensi disebabkan oleh radio yang memiliki frekuensi sama saling berdekatan, mengakibatkan dampak negatif yang signifikan terhadap kinerja jaringan nirkabel. Optimalisasi frekuensi radio adalah mengoptimalkan pengalokasian frekuensi radio FM sehingga pengguna pemancar frekuensi sebanyak-banyaknya, dengan jumlah pemakaian alokasi frekuensi sesedikit mungkin. Optimalisasi frekuensi radio bertujuan menghindari interferensi frekuensi radio pada radius yang bertubrukan. Algoritma Welch Powell yang digunakan dalam pewarnaan graf ternyata dapat diterapkan pada masalah interferensi penggunaan frekuensi radio. Misalkan pemancar dengan suatu frekuensi tertentu adalah suatu simpul. Pemancar yang memungkinkan akan berinterferensi dihubungkan dengan sisi, kita sebut kedua simpul yang dihubungkan sisi sebagai simpul yang bertetangga. Sesuai dengan algoritma Welch Powell, simpul yang bertetangga akan memiliki warna yang berbeda. Warna di sini adalah gambaran dari frekuensi yang dimiliki suatu pemancar. Artinya, jika warna pemancar (simpul) adalah sama, maka pemancar tersebut memiliki frekuensi radio yang sama. Jika pengaturan penggunaan frekuensi seperti ini digunakan, maka akan tercapai tujuan mengoptimalkan frekuensi radio, dengan mengakomodasikan pengguna frekuensi radio sebanyak-banyaknya dengan jumlah frekuensi yang digunakan sesedikit mungkin.

4.2 Deskripsi Umum Perangkat Lunak

Perangkat lunak simulasi stasiun radio merupakan suatu aplikasi yang membantu Balai Monitoring provinsi Jawa Timur menyelesaikan masalah tentang penempatan stasiun radio dan penentuan frekuensi radio FM menggunakan metode pewarnaan graf dengan Algoritma *Welch Powell*. Perangkat lunak ini terdiri dari 3 faktor yang saling mempengaruhi, yaitu lokasi, radius pemancar, dan frekuensi. Ketiga faktor tersebut mempengaruhi terjadinya interferensi pada daerah tertentu. Selain itu, Ketiga faktor diatas juga saling berelasi satu sama lain, sehingga diperlukan pengaturan dalam simulasi stasiun radio FM agar mengurangi terjadinya interferensi satu sama lain.

4.3 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak dilakukan untuk mengidentifikasi kebutuhan dan komponen apa saja yang diperlukan untuk perancangan perangkat lunak. Dengan adanya analisis, suatu sistem diharapkan dapat diuraikan secara utuh menjadi komponen-komponen dasar dengan tujuan identifikasi, evaluasi, dan pemenuhan kebutuhan yang diharapkan. Analisis kebutuhan perangkat lunak simulasi stasiun radio FM secara umum dengan menyesuaikan prosedur metode pewarnaan graf dan disesuaikan dengan deskripsi umum perangkat lunak. Perangkat lunak ini harus memiliki antarmuka pengguna (*Interface*) yang interaktif, mudah digunakan, mudah dipelajari, memberi respon yang cepat. Analisis kebutuhan perangkat lunak penjadwalan mata kuliah ini dapat dibagi menjadi tiga, yaitu :

1. Kebutuhan Fungsional

Kebutuhan fungsional dari perangkat lunak simulasi stasiun radio FM dinyatakan pada Tabel 4.1 di bawah ini.

Tabel 4. 1 Kebutuhan Fungsional

No.	Kebutuhan Fungsional
1.	Mengelola Posisi Stasiun Radio FM
2.	Mengelola Frekuensi Stasiun Radio FM

2. Kebutuhan Non Fungsional

Kebutuhan non fungsional dari perangkat lunak simulasi stasiun radio FM dinyatakan pada Tabel 4.2 di bawah ini.

Tabel 4. 2 Kebutuhan Non Fungsional

No.	Kebutuhan Non Fungsional
1.	Ketersediaan Perangkat Lunak
2.	Antarmuka perangkat Lunak bersifat interaktif

3. Kebutuhan Perangkat Lunak

Kebutuhan lain dari perangkat lunak simulasi stasiun radio FM, yaitu :

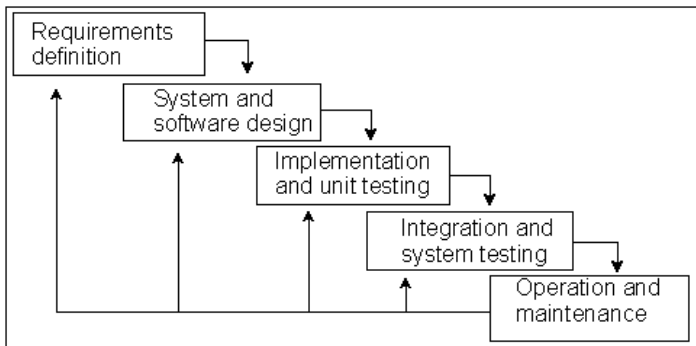
1. Sistem Operasi Windows 8.1 Ultimate 64-bit
2. Hard Disk 1 TB
3. Processor AMD A8
4. RAM 4 GB
5. NetBeans IDE 8.1
6. Power Designer 16.5
7. Balsamiq Mockup 3

4.4 Desain Sistem Perangkat Lunak

Berikut ini adalah desain sistem perangkat lunak simulasi stasiun radio FM di provinsi Jawa Timur.

4.3.1 Waterfall Diagram

Metode *waterfall* merupakan metode yang sering digunakan oleh penganalisa sistem pada umumnya. Inti dari metode *waterfall* adalah pengerjaan dari suatu sistem dilakukan secara berurutan atau secara linear. Pada tugas akhir ini penulis menggunakan model waterfall diagram untuk membuat software simulasi stasiun radio di Jawa Timur. Model *waterfall* diagram memiliki 5 langkah yaitu: *Requirements definition*, *System and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance* seperti gambar 4.1 berikut:



Gambar 4. 1 *Waterfall Diagram*

Berikut adalah penjelasan dari tahap-tahap yang dilakukan di dalam *waterfall* model:

1) *Requirements definition*

Pada tahap *user requirements* penulis melakukan analisa terhadap kebutuhan sistem. Proses pencarian kebutuhan diintensifkan dan difokuskan pada *software* untuk mengetahui sifat dari program yang akan dibuat. Setelah mendapatkan analisa sistem yang akan dibuat maka penulis melakukan tahap selanjutnya.

2) *System and software design*

Pada tahap *system and software design* penulis mengubah kebutuhan-kebutuhan sistem menjadi representasi ke dalam *software design* (GUI). Desain harus dapat mengimplementasikan kebutuhan yang telah disebutkan pada tahap sebelumnya. Setelah melakukan perancangan sistem, selanjutnya desain yang digunakan akan diimplementasikan pada tahap berikutnya.

3) *Implementation and unit testing*

Pada tahap *implementation and unit testing* penulis melakukan pembuatan sistem ke dalam bahasa pemrograman *java*. Penulis menggunakan Netbeans IDE 8.1 sebagai *tools* dalam membuat program dengan *java*. Desain yang sudah selesai akan diimplementasikan fungsinya. Setelah tahap pembuatan sistem sudah selesai, maka penulis melakukan testing unit pada setiap tombol dan pemrosesan pada program. Jika terdapat fungsi tombol dan pemrosesan yang masih mengalami *error* maka penulis harus memperbaiki fungsi tersebut untuk bisa melanjutkan tahap selanjutnya.

4) *Integration and system testing*

Pada tahap *integration and system testing* penulis melakukan uji coba program ke *user*. *User* melakukan

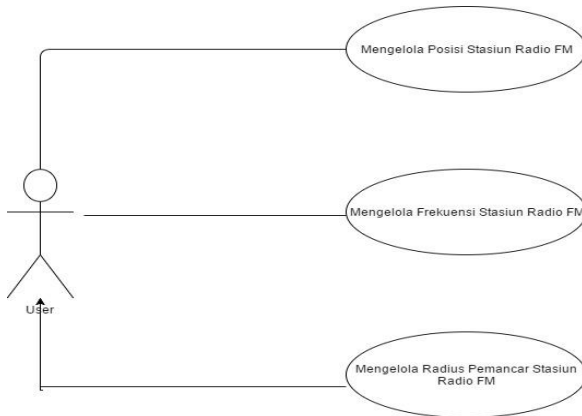
pengujian kemampuan dan keefektifan program untuk mengkaji jika terjadi *error* atau tidak. Setelah melakukan pengujian dan tidak terjadi *error* maka melakukan tahap terakhir

5) *Operation and maintenance*

Pada tahap *Operation and maintenance*, perangkat lunak yang sudah diuji oleh *user*, akan diimplementasikan oleh *user* untuk simulasi stasiun radio FM di Jawa Timur. *Programer* akan tetap melakukan pemeliharaan untuk pemeliharaan dan perkembangan software.

4.3.2 Usecase Diagram

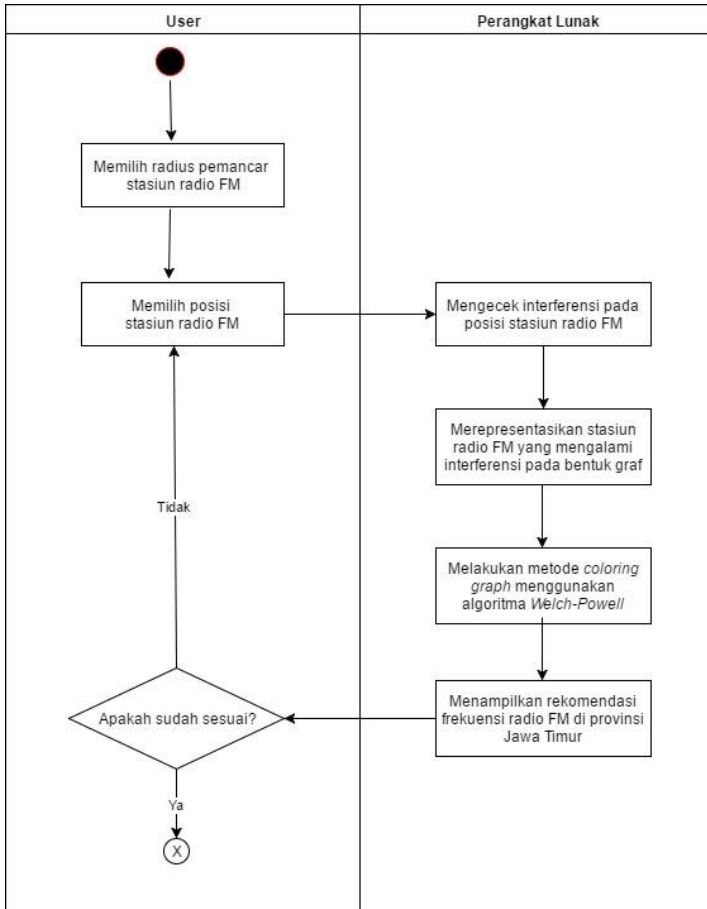
Usecase diagram menggambarkan aktivitas yang dapat dilakukan *user*. Berdasarkan kebutuhan yang telah dijelaskan sebelumnya, dapat dibuat *usecase* diagram perangkat lunak simulasi stasiun radio FM di provinsi Jawa Timur yang dinyatakan pada Gambar 4.2 di bawah ini.



Gambar 4. 2 *Usecase* Diagram

4.3.3 Activity Diagram

Berdasarkan *use case diagram* yang ditunjukkan pada gambar 4.3, kemudian dibuat suatu *activity diagram* untuk menggambarkan alur proses dari setiap interaksi.



Gambar 4. 3 *Activity Diagram*

Pada gambar 4.3 sistem dimulai dengan memilih radius pemancar stasiun radio FM dan menentukan posisi stasiun radio FM di provinsi Jawa Timur. Apabila semua parameter telah di-input kan, maka sistem akan melakukan pengecekan interferensi pada posisi stasiun radio FM yang memiliki radius bertubrukan dengan stasiun radio FM lainnya. Selanjutnya sistem akan menampilkan hasil dari pola interferensi stasiun radio FM dalam bentuk graf. Setelah itu melakukan metode pewarnaan graf dengan menggunakan algoritma *Welch-Powell*. Hasil dari metode pewarnaan graf akan di rekomendasikan untuk *user* dalam menentukan frekuensi radio. Apabila hasil dari proses perangkat lunak sudah sesuai, maka proses telah selesai.

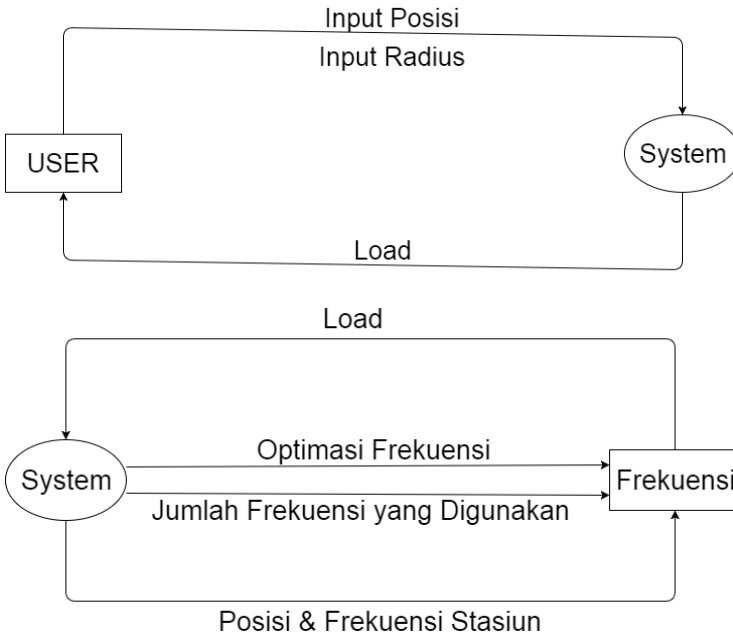
4.3.4 Data Flow Diagram

Data flow diagram adalah representasi grafik yang menggambarkan aliran informasi dan transformasi informasi yang diaplikasikan sebagai data yang mengalir dari masukan (*input*) dan keluaran (*output*) [11]. Kemudian dibuat suatu *data flow diagram* untuk menggambarkan alur proses. DFD *Level 0* biasa disebut dengan diagram sistem inti. Pada diagram sistem ini menjelaskan sebuah proses umum yang nantinya akan di dekomposisi menjadi proses-proses yang lebih detail. Seperti gambar 4.4 berikut



Gambar 4. 4 *Data Flow Diagram Level 0*

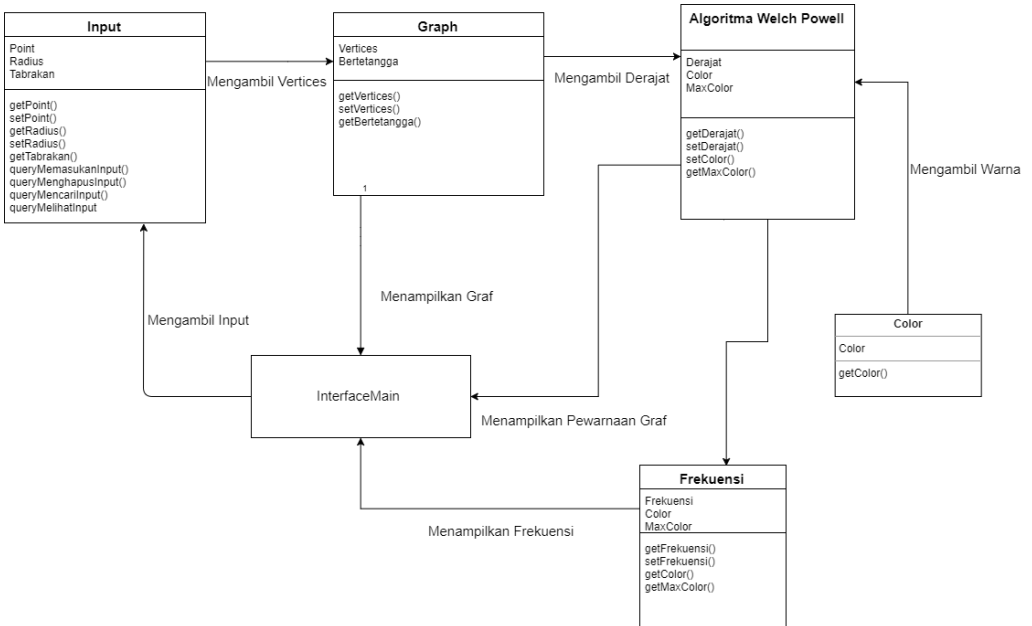
DFD *level 1* menggambarkan dekomposisi dari proses-proses diagram sistem inti menjadi lebih detail.



Gambar 4. 5. *Data Flow Diagram Level 1*

4.3.5 Class Diagram

Diagram kelas menggambarkan struktur system dari segi pendefinisian kelas-kelas yang dibuat untuk membangun sistem. Kelas-kelas yang ada pada struktur system harus dapat melakukan fungsi-fungsi sesuai kebutuhan system sehingga pembuat perangkat lunak dapat membuat program sesuai dengan rancangan diagram kelas. Seperti gambar 4.6 berikut.



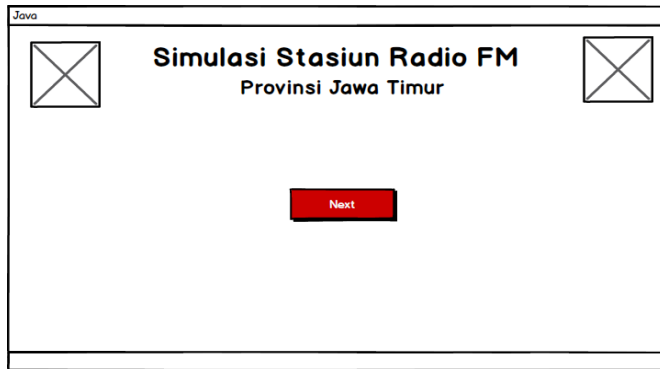
Gambar 4. 6. *Class Diagram*

4.4 Desain Antarmuka Perangkat Lunak

Berikut adalah desain atau perancangan antarmuka perangkat lunak simulasi stasiun radio FM di provinsi Jawa Timur.

4.4.1 Desain Halaman Utama

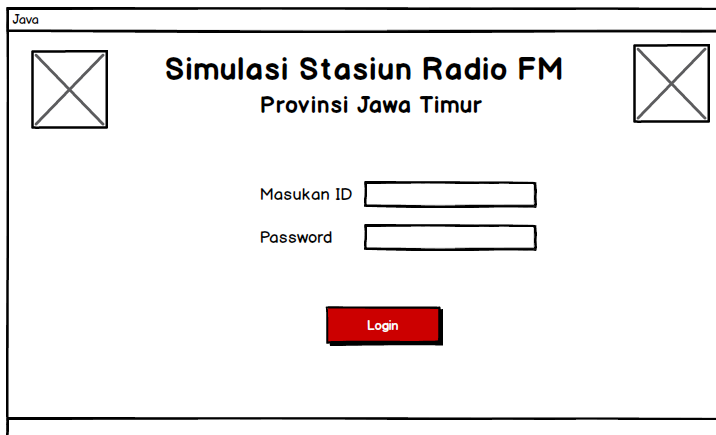
User akan melihat tampilan halaman utama setelah membuka perangkat lunak. *User* akan keluar dari program jika memilih tombol *exit* yang ada pada halaman utama. Desain antarmuka halaman utama dinyatakan pada Gambar 4.7.



Gambar 4. 7 Desain Antarmuka Halaman Utama

4.4.2 Desain Login User

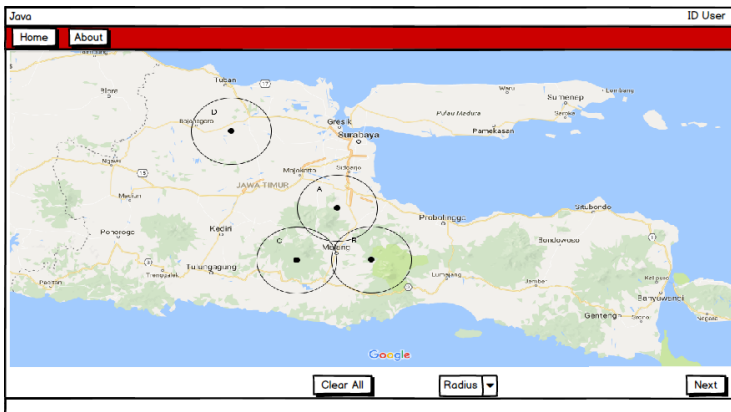
User akan *login* terlebih dahulu pada program simulasi stasiun radio FM. *User* yang dimaksud adalah pihak balai monitoring provinsi Jawa Timur. setelah memasukan ID dan *password* maka *User* memilih tombol login untuk dapat masuk ke menu home program.



Gambar 4. 8 Desain *Login User*

4.4.3 Desain Halaman Home

Setelah *user* menekan tombol *login* pada halaman sebelumnya, selanjutnya *user* akan melihat halaman *home*. Pada halaman *home* terdapat tombol *about*, yang berfungsi untuk memberi petunjuk penggunaan simulasi program. *User* akan menentukan posisi stasiun radio pada *layout* yang terdapat pada program. Terdapat *combo box* radius yang berfungsi untuk menentukan radius pancaran stasiun radio. *User* dapat menghapus semua stasiun radio yang sudah ada dengan memilih tombol *clear all*. Setelah menentukan posisi stasiun radio, maka *user* memilih tombol *next* untuk tahap selanjutnya yaitu representasi graf. Desain antarmuka halaman *home* dinyatakan pada Gambar 4.9 .

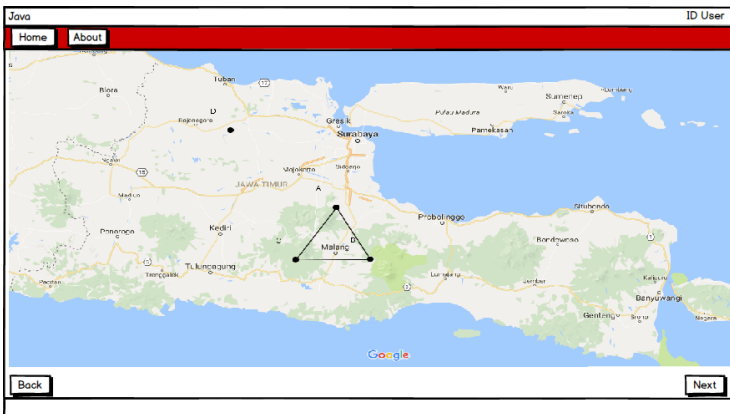


Gambar 4. 9 Desain Antarmuka Halaman Home

4.4.4 Desain Representasi Graf

Setelah *user* menekan tombol *next* pada halaman home, selanjutnya *user* akan melihat representasi graf dari posisi stasiun radio yang sudah diinputkan. Jika *user* memilih

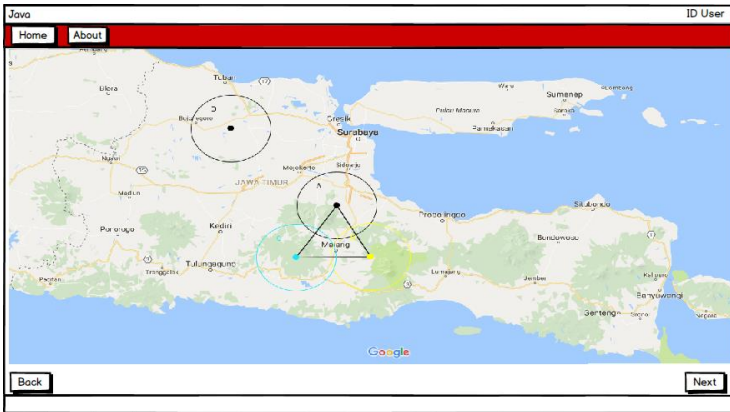
tombol *back* maka *user* akan kembali ke halaman sebelumnya. Terdapat tombol *about* yang berfungsi memberi petunjuk pada halaman representasi graf. Tombol *home* berfungsi untuk kembali ke halaman utama. Desain *antarmuka* halaman representasi graf dinyatakan pada Gambar 4.10.



Gambar 4. 10 Desain *Antarmuka* Representasi Graf

4.4.5 Desain Pewarnaan Graf

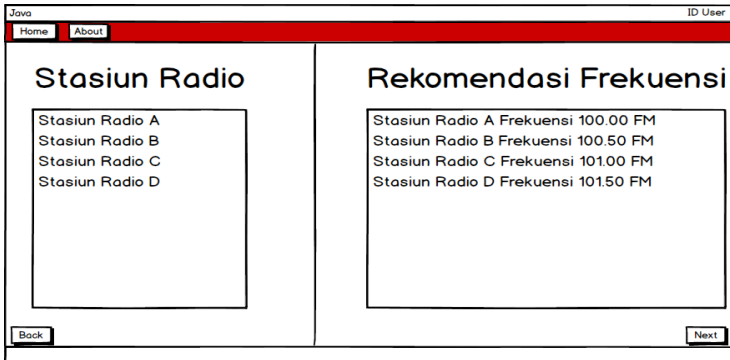
Setelah *user* menekan tombol *next* pada halaman home, selanjutnya *user* akan melihat representasi graf dengan menggunakan metode pewarnaan graf dari posisi stasiun radio yang sudah diinputkan. Jika *user* memilih tombol *back* maka *user* akan kembali ke halaman sebelumnya. Terdapat tombol *about* yang berfungsi memberi petunjuk pada halaman representasi graf. Tombol *home* berfungsi untuk kembali ke halaman utama. Desain *antarmuka* pewarnaan graf dinyatakan pada Gambar 4.11.



Gambar 4. 11 Desain *Antarmuka* Pewarnaan Graf

4.4.6 Desain Rekomendasi Frekuensi Stasiun Radio FM

User akan melihat tampilan hasil rekomendasi dari pewarnaan graf setelah memilih tombol *next* pada halaman sebelumnya. Jika *user* memilih tombol *back*, maka *user* akan kembali ke halaman pewarnaan graf. *User* dapat menyimpan hasil rekomendasi jika memilih tombol *save* pada halaman rekomendasi frekuensi radio. Desain antarmuka halaman hasil rekomendasi frekuensi stasiun radio dinyatakan pada Gambar 4.12.



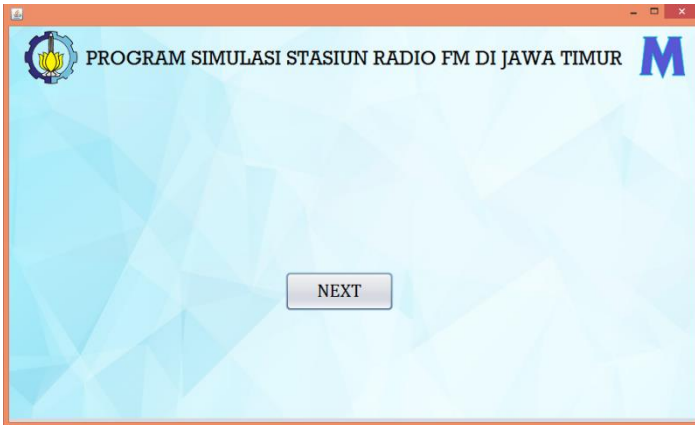
Gambar 4. 12 Desain *Antarmuka* Halaman Hasil Rekomendasi Pewarnaan Graf

4.5 Implementasi Program

Implementasi program menjelaskan tentang hasil implementasi rancangan dan pengujian perangkat lunak simulasi stasiun radio di provinsi Jawa Timur dalam bahasa pemrograman Java. Pengujian meliputi menginput data oleh *user*, serta hasil pewarnaan graf sampai terbentuknya rekomendasi frekuensi stasiun radio di Jawa Timur. Perangkat lunak simulasi stasiun radio di provinsi Jawa Timu ini diimplementasikan dalam bahasa pemrograman Java dengan menggunakan *software* Netbeans IDE 8.1

4.5.1 Tampilan Halaman Pertama

Gambar 4.13 merupakan tampilan yang pertama kali muncul saat perangkat lunak penjadwalan dijalankan. Pada halaman utama perangkat lunak terdapat dua tombol, yaitu START yang akan menampilkan tampilan selanjutnya



Gambar 4. 13 Tampilan Halaman Utama

4.5.2 Tampilan Login

Gambar 4.14 merupakan tampilan login setelah memilih tombol next pada antarmuka sebelumnya. *User* diwajibkan untuk mengisi USERNAME dan PASSWORD untuk dapat masuk ke dalam sistem. Tombol NEXT akan menampilkan tampilan utama.



Gambar 4. 14 Tampilan Login

Untuk keseluruhan *source code* fungsi login dapat dilihat pada gambar dibawah.

```
if(user.equals(username.getText().toString()  
g()) &&  
pass.equals(password.getText().toString()  
)) {  
        AntarmukaMain baru = new  
AntarmukaMain(user,null);  
        baru.setVisible(true);  
        this.setVisible(false);  
    }  
    else {  
new PopUp().show_error("Username atau  
Password salah","Failed");  
    }  
}
```

4.5.3 Tampilan Halaman Utama

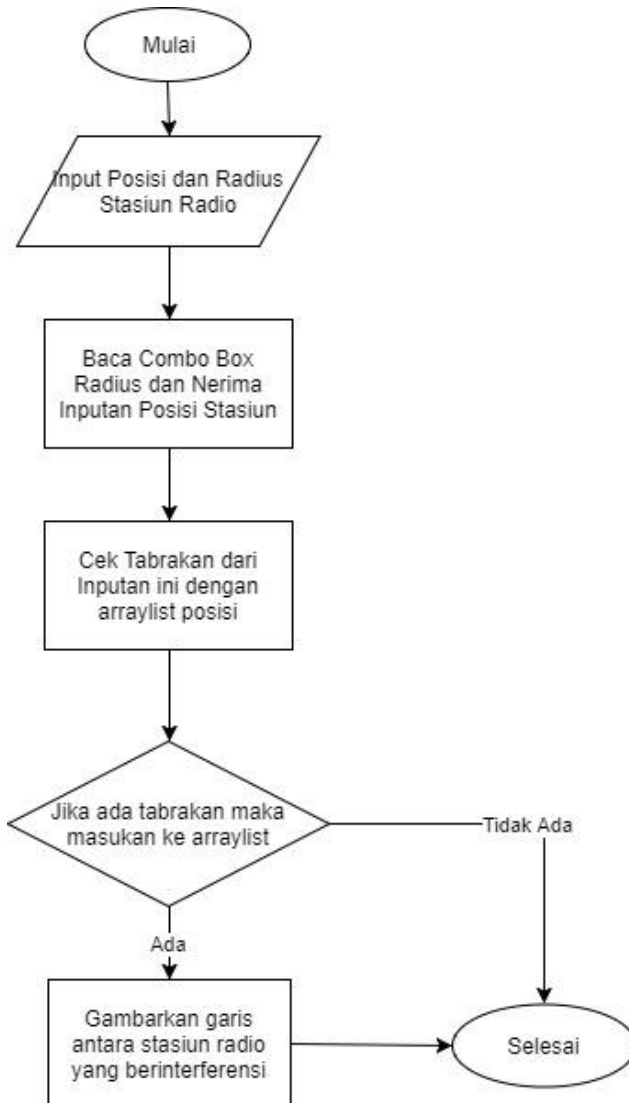
Pada halaman utama *user* akan menginputkan posisi stasiun radio FM. Terdapat beberapa tombol dan ikon pada halaman utama. Pada halaman utama terdapat gambar ilustrasi peta Jawa Timur dan tidak bersifat spasial. *User* dapat menginputkan posisi stasiun radio dengan menekan tombol *mouse* kiri dan fungsi tombol *mouse* kanan untuk menghapus posisi stasiun radio. *User* dapat mengubah radius pancaran stasiun radio dengan memilih *combo box* radius Terdapat tombol clear yang berfungsi untuk menghilangkan setiap inputan posisi radio. Pada tampilan input warna awal berupa hitam, menunjukkan bahwa inputan awal berupa frekuensi yang sama. Setelah *user* melakukan penginputan posisi stasiun radio *user* dapat menekan tombol *next* pada halaman

utaman, selanjutnya *user* akan melihat tampilan halaman representasi graf. Jika *user* memilih tombol *back*, maka *user* akan kembali ke halaman utama. Desain antarmuka halaman home dinyatakan pada Gambar 4.15.



Gambar 4.15 Program Halaman Utama

Untuk flowchart input graf dapat dilihat pada Gambar 4.16.



Gambar 4. 16. Flowchart Input Graf

Untuk *source code* fungsi *input mouse listener* dapat dilihat pada gambar dibawah.

```
points = new ArrayList();
    label_huruf = new ArrayList();
    loadImage();
    addMouseListener(new
MouseListener() {
        @Override
        public void
mousePressed(MouseEvent e) {
            if(enable) {
                if(e.getButton() ==
MouseEvent.BUTTON1) {
                    now = new
Point(e.getX(), e.getY());
                    points.add(new
Koordinat(now, radiusLuar));
                }
                else if(e.getButton()
== MouseEvent.BUTTON3) {
                    hapus = new
Point(e.getX(), e.getY());
                    delete(hapus);
                }
                repaint();
            }
        }
    });
```

4.5.4 Tampilan Representasi Graf

Pada halaman representasi graf, *user* akan melihat hasil representasi graf dari inputan yang sudah dilakukan oleh *user* sebelumnya. Program akan mengecek setiap node yang mengalami interferensi, jika terdapat node yang mengalami interferensi maka program akan membuat lintasan pada stasiun yang mengalami interferensi. Terdapat tombol *back* pada tampilan representasi graf yang berfungsi untuk kembali ke halaman sebelumnya. Setelah *user* melihat hasil representasi graf, *user* dapat memilih tombol *next* untuk melanjutkan tampilan selanjutnya. Desain antarmuka halaman representasi graf dinyatakan pada Gambar 4.17.



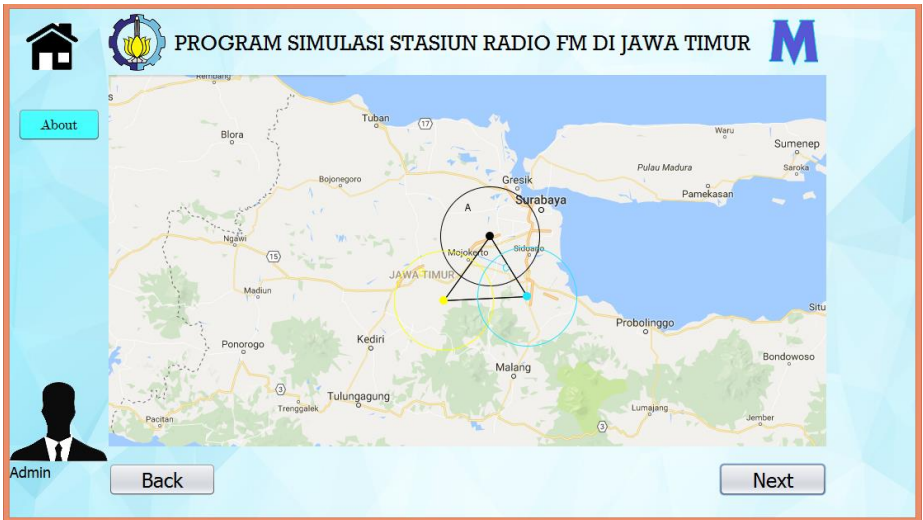
Gambar 4. 17 Program Representasi Graf

Untuk *source code* fungsi hasil representasi graf dapat dilihat pada gambar dibawah

```
public void gambarGaris(Graphics g) {
    for(int i = 0; i <
tabrakan.size(); i++) {
        int x1 =
points.get(tabrakan.get(i).x).x;
        int y1 =
points.get(tabrakan.get(i).x).y;
        int x2 =
points.get(tabrakan.get(i).y).x;
        int y2 =
points.get(tabrakan.get(i).y).y;
        g.drawLine(x1,y1,x2,y2);
    }
}
```

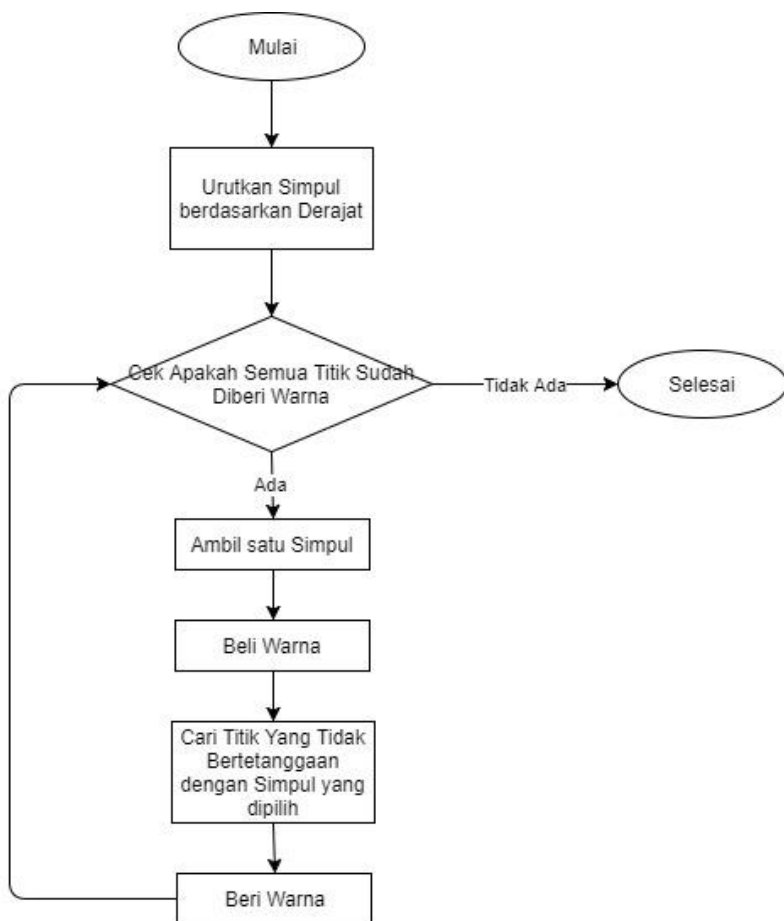
4.5.5 Tampilan Pewarnaan Graf

Pada halaman pewarnaan graf, *user* akan melihat hasil pewarnaan graf dari inputan yang sudah dilakukan oleh *user* sebelumnya. Program akan mengecek setiap graf yang bertetangga tidak boleh memiliki warna yang sama, bilangan khromatik dari pewarnaan graf adalah minimum. Terdapat tombol *back* pada tampilan representasi graf yang berfungsi untuk kembali ke halaman sebelumnya. Setelah *user* melihat hasil representasi graf, *user* dapat memilih tombol *next* untuk melanjutkan tampilan selanjutnya. Desain *antarmuka* halaman pewarnaan graf dinyatakan pada Gambar 4.19.



Gambar 4. 18 Program Pewarnaan Graf

Untuk flowchart pewarnaan graf dapat dilihat pada Gambar 4.20.



Gambar 4. 19. Flowchart Pewarnaan Graf

Untuk *source code* fungsi hasil pewarnaan graf dapat dilihat pada gambar dibawah

```
public void colourVertices(){
    Collections.sort(vertices, new
VertexComparator()); // arrange vertices
in order of descending valence
    color = new int[vertices.size()];
    int warna = 1;
    for (int i = 0; i <
vertices.size(); i++) {

if(vertices.get(i).neighbors==null){

color[vertices.get(i).node]=warna;
    continue;
    }

if(color[vertices.get(i).node] != 0)
continue;

    color[vertices.get(i).node] =
warna;

    int[] dilarang = new
int[vertices.size()];

dilarang[vertices.get(i).node] = 1;
    for(int j = 1; j <
vertices.get(i).neighbors.length; j++) {

dilarang[vertices.get(i).neighbors[j]] =
1;

```

```
        }
        while(true) {
            int temp =
cari(dilarang);
            if(temp == -1) break;
            color[temp] = warna;
            dilarang[temp] = 1;
            for(int j = 0; j <
vertices.get(cari1(temp)).neighbors.length; j++) {

dilarang[vertices.get(cari1(temp)).neighbors[j]] = 1;

            }
        }
        warna++;
    }
}
```

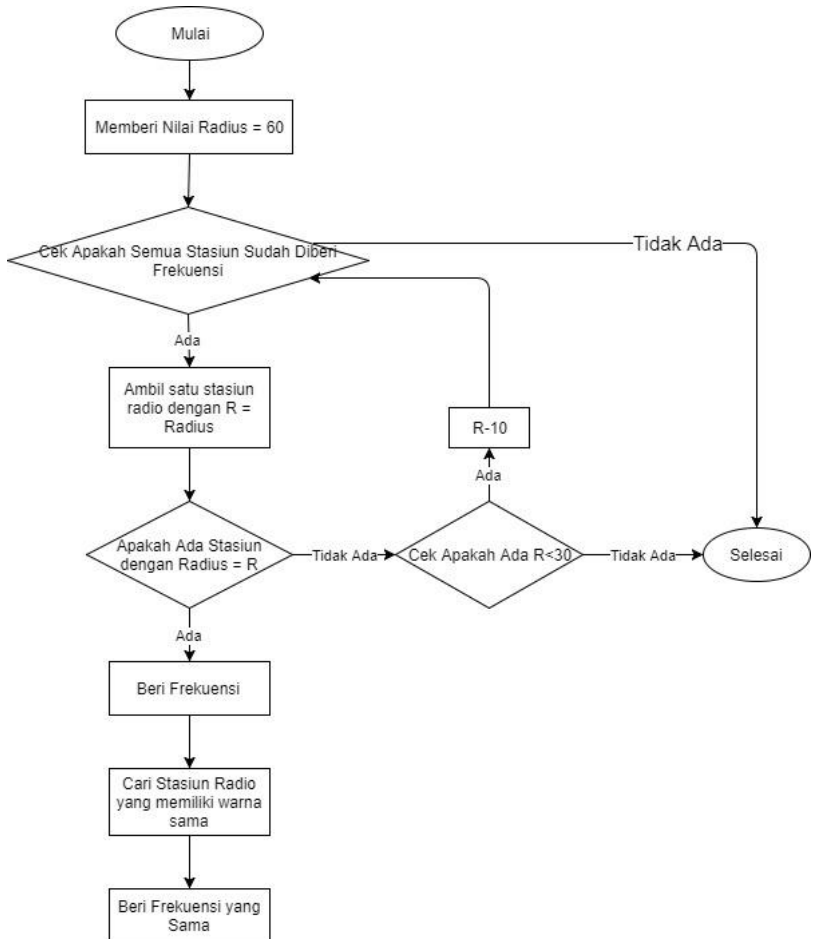

4.5.6 Tampilan Rekomendasi Frekuensi

Pada tampilan rekomendasi frekuensi radio FM. Interval yang digunakan untuk membedakan frekuensi radio FM sebesar 0.4 MHz. *User* akan melihat rekomendasi frekuensi dari inputan posisi stasiun radio. Jika posisi stasiun radio tidak mengalami interferensi maka radio memiliki frekuensi sama, namun jika posisi stasiun radio mengalami interferensi maka salah satu radio diubah frekuensi yang berbeda. Hasil rekomendasi tersebut bergantung pada inputan posisi stasiun radio. Setelah *user* melihat hasil rekomendasi frekuensi radio FM, *user* dapat menyimpan posisi stasiun radio dengan memilih tombol *save*. Desain antarmuka rekomendasi frekuensi stasiun radio FM dinyatakan pada Gambar 4.20.



Gambar 4. 20 Program Rekomendasi Frekuensi

Untuk flowchart input graf dapat dilihat pada gambar 4.22 dibawah.



Gambar 4. 21. Flowchart Penentuan Frekuensi Radio FM

Untuk *source code* fungsi rekomendasi frekuensi dapat dilihat pada gambar dibawah

```
public InterfaceKesimpulan(InterfaceMain
main,float mulai,float beda) {
    initComponents();
    String textnya_kiri = "";
    String textnya_kanan = "";
    this.main = main;
    if(main.Input.points != null) {
        int n =
main.Input.points.size();
        int[] array = new int[n];
        int now = 60;
        double[] frekuensi = new
double[n];
        boolean ketemu = false;
        int ke = 0,pengali = 0;
        while(true) {
            ketemu = false;
            for(int i=0;i<n;i++) {
                if(array[i] == 1)
continue;
            if(main.Input.points.get(i).radius ==
now) {
                    ketemu = true;
                    ke = i;
                    break;
                }
            }
            if(ketemu) {
                frekuensi[ke] =
mulai+beda*pengali;
                array[ke] = 1;
            }
        }
    }
}
```

```

        int temp_warna =
main.Input.points.get(ke).warna;
        for(int j=0;j<n;j++)
    {
        if(array[j] == 1)
continue;

if(main.Input.points.get(j).warna ==
temp_warna) {
        frekuensi[j]
= mulai+beda*pengali;
        array[j] = 1;
        }
        }
        pengali++;
    }
    else {
        if(now > 30) now-=10;
        else break;
    }
}

        for(int i=0;i<n;i++) {
            String temp_kiri =
"Stasiun Radio "+text(i)+"\n";
            double frekuensinya =
Double.parseDouble(new
DecimalFormat("##.##").format(frekuensi[i]
));
            String temp_kanan =
"Stasiun Radio "+text(i)+" frekuensi

```

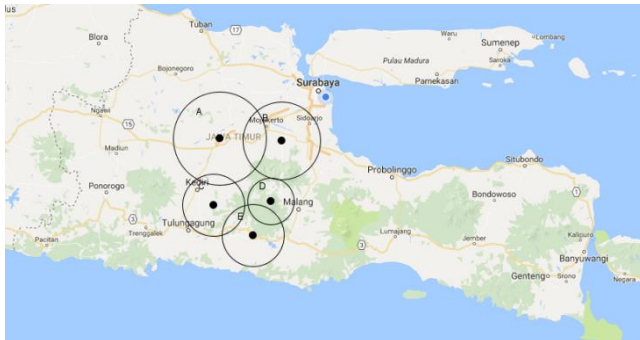
```
" + frekuensinya + " FM\n";  
        textnya_kiri += temp_kiri;  
  
textnya_kanan += temp_kanan;  
    }
```


BAB V PENGUJIAN DAN PEMBAHASAN HASIL

Bab ini menjelaskan mengenai pengujian yang dilakukan terhadap simulasi penerapan frekuensi. Hasil pengujian kemudian dibahas untuk mengetahui unjuk kerja sistem secara keseluruhan dalam menjalankan fungsi yang diharapkan.

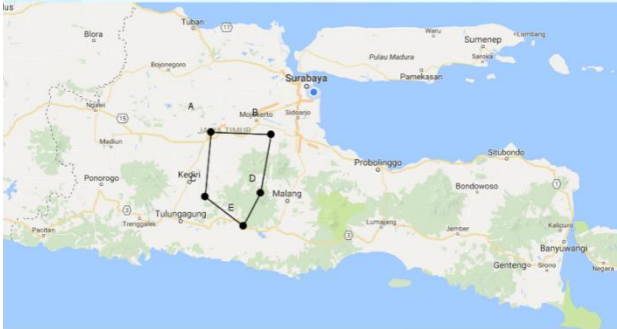
5.1. Pembahasan

Berdasarkan implementasi dari program, maka akan dilakukan pengujian terhadap sistem simulasi secara manual. Tahap pertama adalah menentukan posisi stasiun radio dengan *coverage area* yang berbeda-beda, dimana setiap stasiun radio memiliki frekuensi sama. Seperti pada Gambar 5.1



Gambar 5. 1. Posisi Stasiun Radio Tes Manual

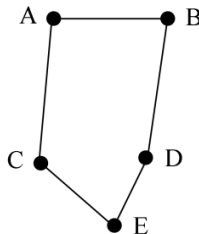
Pada Gambar 5.1 posisi stasiun radio menunjukkan terdapat beberapa stasiun radio yang mengalami interferensi *coverage area*. Setelah dilakukan penentuan posisi stasiun radio, selanjutnya adalah merepresentasikan stasiun radio tersebut ke dalam model graf



Gambar 5. 2. Model Graf Tes Manual

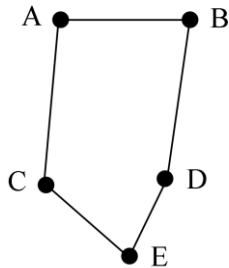
Pada Gambar 5.2 posisi stasiun radio yang mengalami interferensi dimodelkan memiliki *edge*. Model graf yang dihasilkan dari posisi stasiun radio menggambarkan graf terhubung, karena setiap stasiun radio mengalami interferensi. Setelah dilakukan representasi graf stasiun radio, selanjutnya adalah pewarnaan graf menggunakan Algoritma Welch Powell. Langkah langkah Algoritma Welch Powell:

- a. Dicari simpul dengan derajat tertinggi dalam graf tersebut.
- b. Karena pada graf tersebut setiap simpul memiliki derajat yang sama, maka simpul pertama yang akan di eksekusi.
- c. Pemberian warna pada simpul A dengan warna hitam.



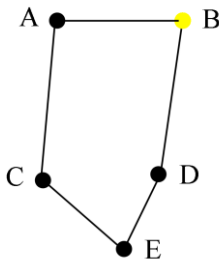
Gambar 5. 3. Gambar Graf Sebelum di Welch Powell

- d. Cari simpul yang tidak bertetangga dengan simpul A, yaitu simpul D maka simpul D diberi warna hitam juga.



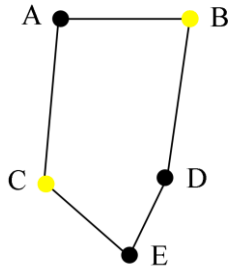
Gambar 5. 4. Gambar Graf Tahap Pertama

- e. Pemberian warna pada simpul B dengan warna kuning.



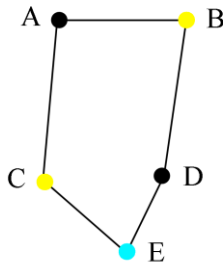
Gambar 5. 5. Gambar Graf Tahap Kedua

- f. Cari simpul yang tidak bertetangga dengan simpul B, yaitu simpul C maka simpul C diberi warna kuning.



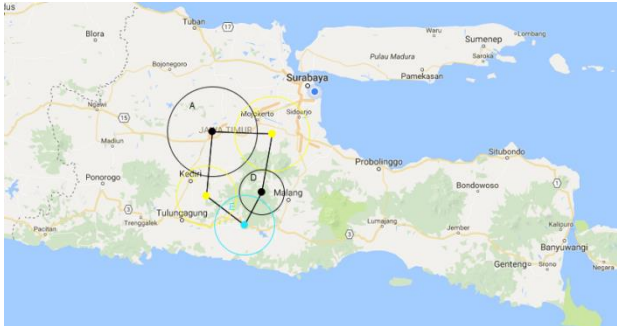
Gambar 5. 6. Gambar Graf Tahap Ketiga

- g. Cari simpul yang masih belum diberi warna, yaitu simpul E
- h. Karena simpul E bertetangga dengan simpul C dengan warna kuning dan simpul D dengan warna hitam
- i. Maka simpul E diberi warna baru yaitu warna biru muda.



Gambar 5. 7. Gambar Graf Tahap Keempat

- j. Karena setiap simpul sudah diberi warna maka tahapan Algoritma Welch-Powell sudah selesai.



Gambar 5. 8. Pewarnaan Graf Tes Manual

Pada Gambar 5.8 posisi stasiun radio yang sudah di modelkan ke dalam bentuk graf diproses ke dalam metode pewarnaan graf dengan menggunakan Algoritma Welch Powell. Posisi stasiun radio yang sudah dimodelkan ke dalam bentuk graf menghasilkan 3 warna berbeda. Selanjutnya warna tersebut akan direpresentasikan ke dalam frekuensi stasiun radio agar tidak mengalami interferensi.

Stasiun Radio A frekuensi 87.7 FM
 Stasiun Radio B frekuensi 88.1 FM
 Stasiun Radio C frekuensi 88.1 FM
 Stasiun Radio D frekuensi 87.7 FM
 Stasiun Radio E frekuensi 88.5 FM

Gambar 5. 9. Frekuensi Radio Tes Manual

Pada Gambar 5.9 banyaknya frekuensi yang dihasilkan adalah 3 frekuensi yang berbeda. Sesuai dengan pewarnaan graf menggunakan Algoritma Welch Powell yang menghasilkan 3 warna berbeda, maka frekuensi yang dihasilkan juga 3 frekuensi berbeda. Algoritma Welch Powell dikatakan optimal karena dapat menghasilkan warna yang minimal dari model graf tersebut.

5.2. Pengujian Program

Pada pengujian program, *user* akan menguji beberapa poin permasalahan yang dapat diterapkan di dalam program. Hasil implementasi pengujian tersebut akan menjadi kajian dalam pengembangan dan pemeliharaan perangkat lunak. Program akan diuji dengan 4 kasus. Dimana setiap Kasus memiliki hasil inputan yang berbeda beda. kasus 1 adalah setiap inputan posisi stasiun radio FM tidak mengalami interferensi. kasus 2 adalah setiap inputan posisi stasiun radio FM mengalami interferensi. kasus 3 adalah terdapat beberapa stasiun radio yang mengalami interferensi. kasus data real adalah dimana input program berdasarkan posisi stasiun radio di Jawa Timur.

5.2.1 Pengujian dengan Kasus 1

Hasil input yang digunakan pada pengujian kasus 1 adalah posisi stasiun radio radio FM tidak mengalami interferensi. Berdasarkan penjelasan tersebut, simulasi yang digunakan dengan radius pemancar sama pada setiap stasiun radio. Dengan syarat tertentu, radius pemancar stasiun radio tidak boleh bertubrukan.

5.2.1.1 Input Kasus 1

Berdasarkan penjelasan pengujian dengan kasus 1, maka posisi stasiun radio yang dipilih haruslah memiliki radius pemancar yang berbeda-beda dan tidak boleh mengalami tubrukan radius pancaran stasiun radio FM dengan syarat radius pemancar bernilai sama. Pada input posisi stasiun radio node berwarna hitam, diasumsikan bahwa pada tahap input setiap stasiun radio memiliki frekuensi yang sama.

Sehingga hasil input ditunjukkan pada Gambar 5.10.



Gambar 5. 10 Input Kasus 1

5.2.1.2 Representasi Graf Kasus 1

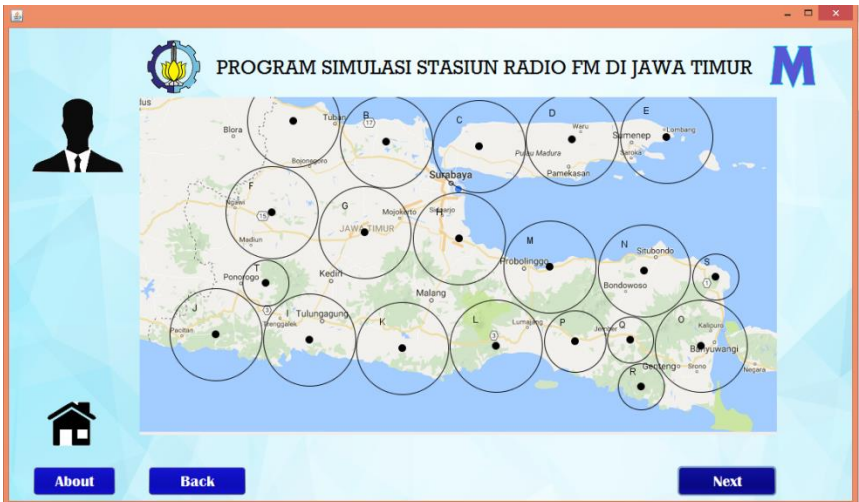
Berdasarkan hasil input, selanjutnya inputan tersebut akan diolah dan diproses menjadi sebuah graf. Pemberian input posisi stasiun radio FM kasus 1 menghasilkan tidak ada node yang memiliki lintasan dengan node yang lain. Hal tersebut dikarenakan setiap node tidak ada yang mengalami interferensi. Dan hasil representasi graf dapat dilihat pada Gambar 5.11.



Gambar 5. 11 Representasi Graf Kasus 1

5.2.1.3 Hasil Pewarnaan Graf Kasus 1

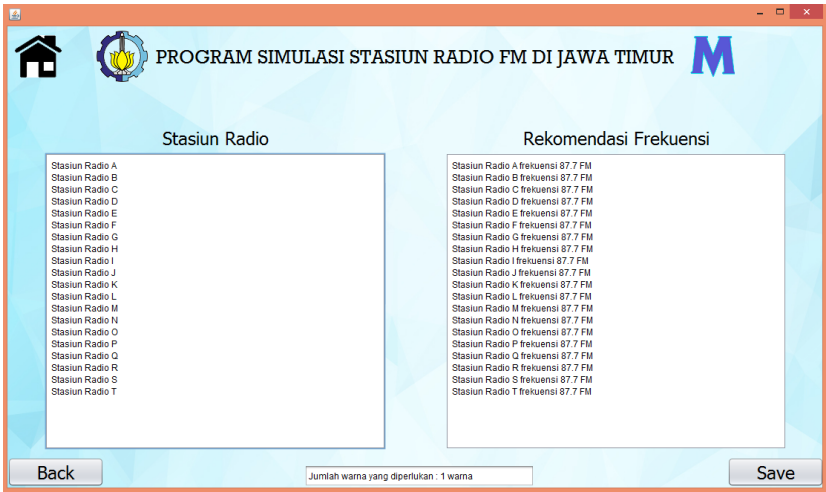
Berdasarkan hasil representasi graf, selanjutnya inputan tersebut akan diolah dan diproses menjadi hasil pewarnaan graf. Pemberian input posisi stasiun radio FM kasus 1 menghasilkan tidak ada node yang memiliki lintasan dengan node yang lain. Hal tersebut dikarenakan setiap node tidak ada yang mengalami interferensi. Maka hasil pewarnaan graf pada kasus 1 hanya memiliki 1 warna, hal tersebut menunjukkan bahwa pada kasus 1 tidak dipermasalahkan jika memiliki frekuensi yang sama. Dan hasil pewarnaan graf dapat dilihat pada Gambar 5.12.



Gambar 5. 12 Program Pewarnaan Graf Kasus 1

5.2.1.4 Hasil Rekomendasi Frekuensi Kasus 1

Berdasarkan hasil pewarnaan graf, selanjutnya inputan tersebut akan diolah dan diproses menjadi hasil rekomendasi frekuensi radio FM. Pemberian input posisi stasiun radio FM kasus 1 menghasilkan setiap stasiun radio FM memiliki frekuensi yang sama. Hal tersebut dikarenakan setiap node tidak ada yang mengalami interferensi. Maka hasil rekomendasi frekuensi pada kasus 1 hanya memiliki 1 frekuensi. Dan hasil rekomendasi frekuensi dapat dilihat pada Gambar 5.13.



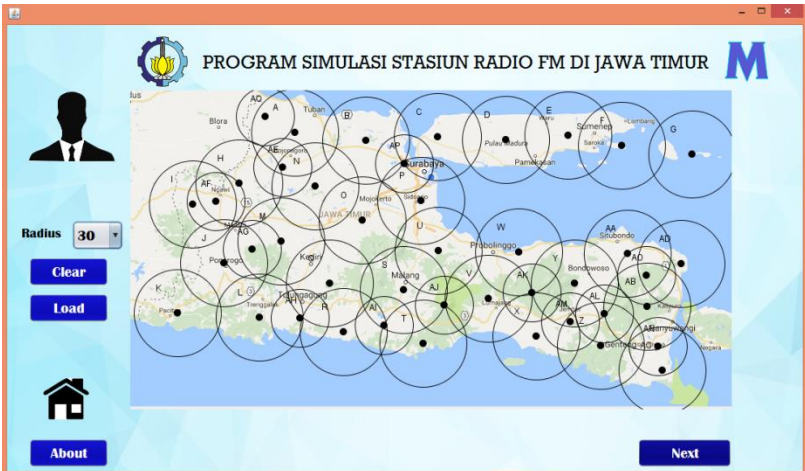
Gambar 5. 13 Program Rekomendasi Frekuensi Kasus 1

5.2.2 Pengujian dengan Kasus 2

Hasil input yang digunakan pada pengujian kasus 2 adalah setiap posisi stasiun radio radio FM mengalami interferensi. Berdasarkan penjelasan tersebut, simulasi yang digunakan dengan radius pemancar yang sama pada setiap stasiun radio. Dengan syarat tertentu, radius pemancar stasiun radio mengalami tubrukan atau interferensi.

5.2.2.1 Input Kasus 2

Berdasarkan penjelasan pengujian dengan kasus 2, maka posisi stasiun radio yang dipilih haruslah mengalami interferensi dengan radius pemancar yang berbeda-beda. Pada inputan posisi stasiun radio node berwarna hitam, diasumsikan bahwa pada tahap input setiap stasiun radio memiliki frekuensi yang sama. Sehingga hasil inputan ditunjukkan pada Gambar 5.14.



Gambar 5. 14 Input Kasus 2

5.2.2.2 Representasi Graf Kasus 2

Berdasarkan hasil input, selanjutnya input tersebut akan diolah dan diproses menjadi sebuah graf. Pemberian input posisi stasiun radio FM kasus 2 menghasilkan setiap node memiliki lintasan dengan node yang lain. Hal tersebut dikarenakan setiap node mengalami interferensi. Dan hasil representasi graf dapat dilihat pada Gambar 5.15.



Gambar 5. 15 Representasi Graf Kasus 2

5.2.2.3 Hasil Pewarnaan Graf Kasus 2

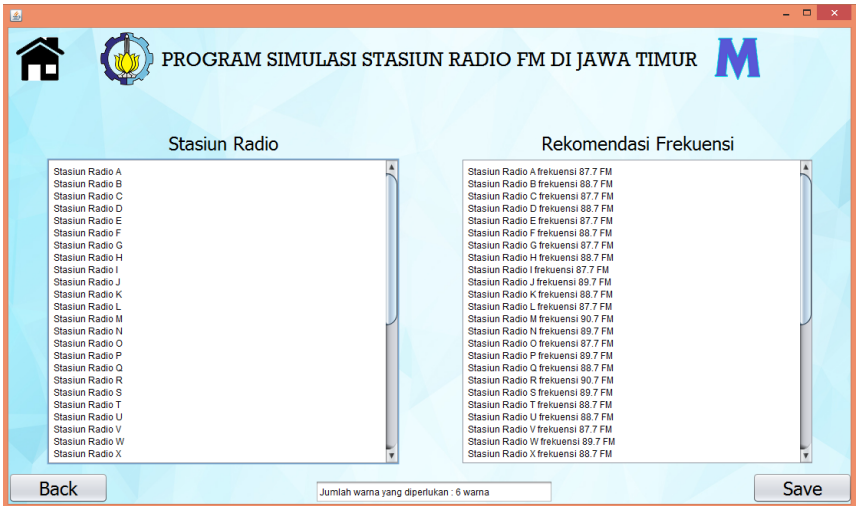
Berdasarkan hasil representasi graf, selanjutnya inputan tersebut akan diolah dan diproses menjadi hasil pewarnaan graf. Pemberian input posisi stasiun radio FM kasus 2 menghasilkan setiap node memiliki lintasan dengan node yang lain. Hal tersebut dikarenakan setiap node mengalami interferensi. Maka hasil pewarnaan graf pada kasus 2 memiliki n warna, hal tersebut menunjukkan bahwa pada kasus 2 bermasalah jika memiliki frekuensi yang sama.. Dan hasil pewarnaan graf dapat dilihat pada Gambar 5.16.



Gambar 5. 16 Pewarnaan Graf Kasus 2

5.2.2.4 Hasil Rekomendasi Frekuensi Kasus 2

Berdasarkan hasil pewarnaan graf, selanjutnya inputan tersebut akan diolah dan diproses menjadi hasil rekomendasi frekuensi radio FM. Pemberian input posisi stasiun radio FM kasus 2 menghasilkan setiap stasiun radio FM memiliki frekuensi yang berbeda-beda. Hal tersebut dikarenakan setiap node mengalami interferensi. Maka hasil rekomendasi frekuensi pada kasus 2 memiliki n frekuensi. Dan hasil rekomendasi frekuensi dapat dilihat pada Gambar 5.17.



Gambar 5. 17 Rekomendasi Frekuensi Kasus 2

5.2.3 Pengujian dengan Kasus 3

Hasil input yang digunakan pada pengujian kasus 3 adalah terdapat beberapa posisi stasiun radio radio FM yang mengalami interferensi dan tidak mengalami interferensi. Berdasarkan penjelasan tersebut, simulasi yang digunakan dengan radius pemancar yang berbeda-beda pada setiap stasiun radio. Dengan syarat tertentu, beberapa radius pemancar stasiun radio mengalami tubrukan dan tidak mengalami tubrukan.

5.2.3.1 Input Kasus 3

Berdasarkan penjelasan pengujian dengan kasus 3, maka posisi stasiun radio yang dipilih terdapat beberapa pemancar stasiun radio FM yang mengalami interferensi dan beberapa tidak mengalami interferensi dengan radius pemancar berbeda-beda. Pada input posisi stasiun radio node berwarna hitam, diasumsikan bahwa pada tahap input setiap stasiun radio memiliki frekuensi yang sama. Sehingga hasil inputan ditunjukkan pada Gambar 5.18.



Gambar 5. 18 Input Kasus 3

5.2.3.2 Representasi Graf Kasus 3

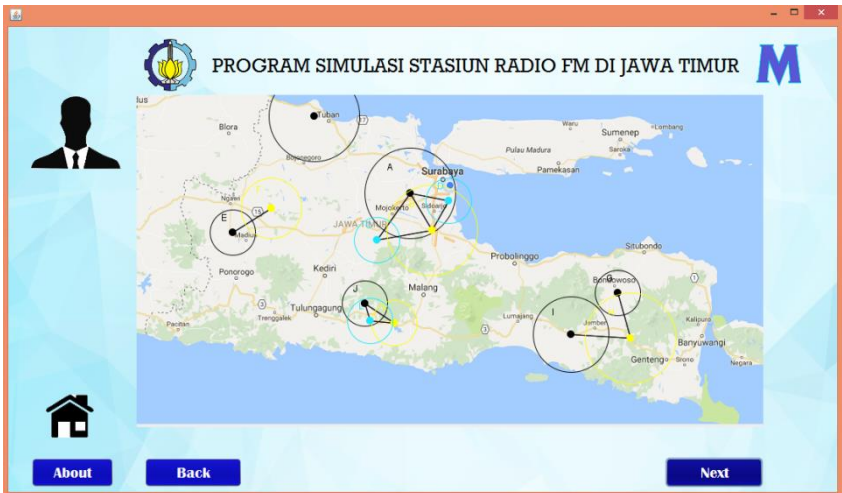
Berdasarkan hasil input, selanjutnya input tersebut akan diperoleh dan diproses menjadi sebuah graf. Pemberian input posisi stasiun radio FM kasus 3 menghasilkan terdapat node yang memiliki lintasan dengan node yang lain dan beberapa tidak memiliki lintasan. Hal tersebut dikarenakan beberapa node mengalami interferensi dan beberapa node tidak mengalami interferensi. Dan hasil representasi graf dapat dilihat pada Gambar 5.19.



Gambar 5. 19 Representasi Graf Kasus 3

5.2.3.3 Hasil Pewarnaan Graf Kasus 3

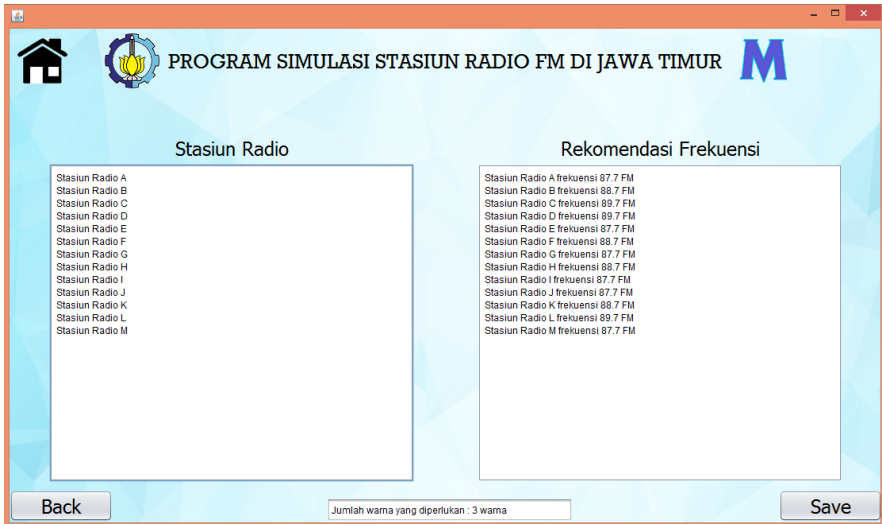
Berdasarkan hasil representasi graf, selanjutnya inputan tersebut akan diolah dan diproses menjadi hasil pewarnaan graf. Pemberian input posisi stasiun radio FM kasus 3 menghasilkan beberapa node memiliki lintasan dengan node yang lain dan beberapa node yang lain tidak memiliki lintasan. Hal tersebut dikarenakan setiap node mengalami interferensi. Maka hasil pewarnaan graf pada kasus 3 memiliki n warna. Dan hasil pewarnaan graf dapat dilihat pada Gambar 5.20.



Gambar 5. 20 Pewarnaan Graf Kasus 3

5.2.3.4 Hasil Rekomendasi Kasus 3

Berdasarkan hasil pewarnaan graf, selanjutnya inputan tersebut akan diolah dan diproses menjadi hasil rekomendasi frekuensi radio FM. Pemberian input posisi stasiun radio FM kasus 3 menghasilkan beberapa stasiun radio FM memiliki frekuensi yang berbeda-beda. Hal tersebut dikarenakan beberapa node mengalami interferensi dan beberapa node tidak mengalami interferensi. Maka hasil rekomendasi frekuensi pada kasus 3 memiliki n frekuensi. Dan hasil rekomendasi frekuensi dapat dilihat pada Gambar 5.21.



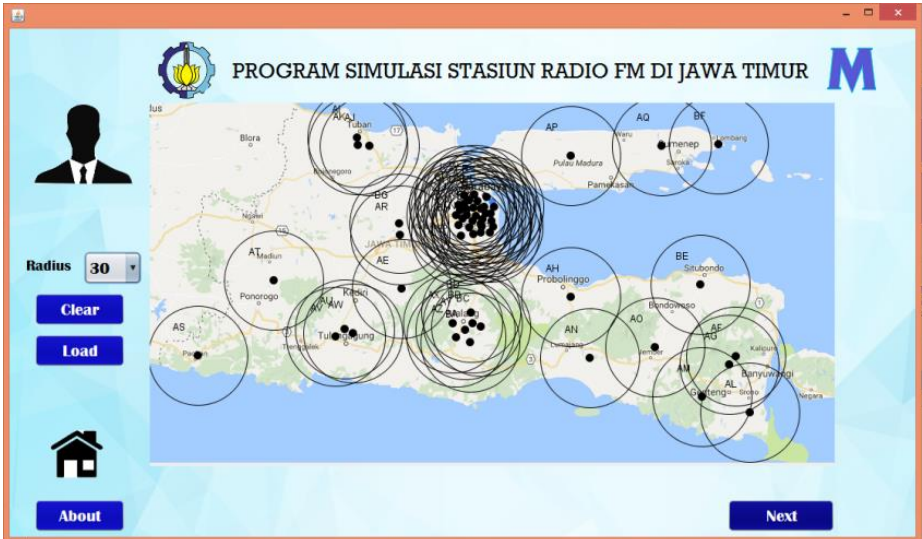
Gambar 5. 21 Rekomendasi Frekuensi Kasus 3

5.2.4 Pengujian dengan Data Real

Hasil input yang digunakan pada pengujian Kasus 1 adalah posisi stasiun radio FM yang ada di Jawa Timur tahun 2006. Berdasarkan penjelasan tersebut, simulasi yang digunakan dengan radius pemancar sama pada setiap stasiun radio.

5.2.4.1 Input Kasus Data Real

Berdasarkan penjelasan pengujian dengan Kasus data real, maka posisi stasiun radio yang dipilih pemancar stasiun radio FM di Jawa Timur. Pada input posisi stasiun radio node berwarna hitam, diasumsikan bahwa pada tahap input setiap stasiun radio memiliki frekuensi yang sama. Sehingga hasil input ditunjukkan pada Gambar 5.22.



Gambar 5. 22 Input Kasus Data Real

5.2.4.2 Representasi Graf Kasus Data Real

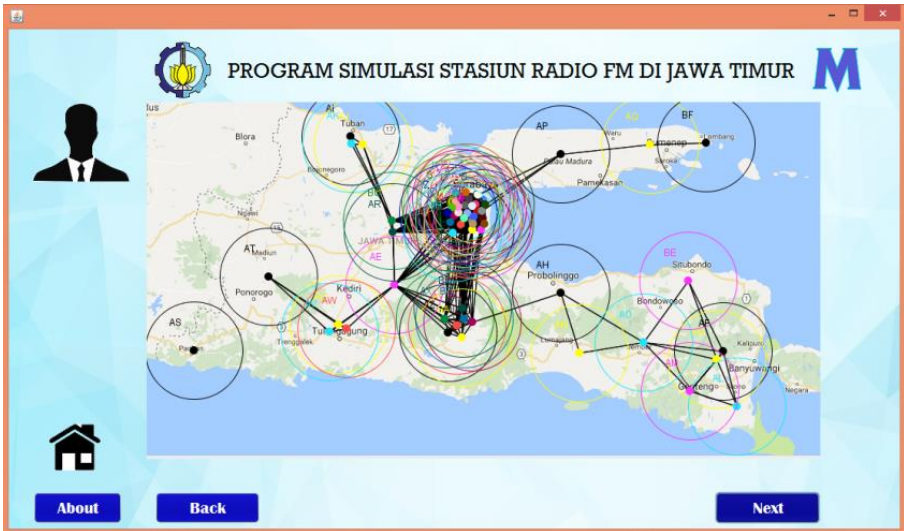
Berdasarkan hasil input, selanjutnya input tersebut akan diperoleh dan diproses menjadi sebuah graf. Pemberian input posisi stasiun radio FM kasus data real menghasilkan beberapa node yang memiliki lintasan dengan node yang lain. Hal tersebut dikarenakan beberapa node mengalami interferensi. Dan hasil representasi graf dapat dilihat pada Gambar 5.23.



Gambar 5. 23 Representasi Graf Kasus Data Real

5.2.4.3 Hasil Pewarnaan Graf Kasus Data Real

Berdasarkan hasil representasi graf, selanjutnya input tersebut akan diolah dan diproses menjadi hasil pewarnaan graf. Pemberian input posisi stasiun radio FM kasus data real menghasilkan terdapat node memiliki lintasan dengan node yang lain. Hal tersebut dikarenakan beberapa node mengalami interferensi. Maka hasil pewarnaan graf pada kasus data real memiliki n warna, hal tersebut menunjukkan bahwa pada Kasus data real memiliki warna yang bervariasi. Dan hasil pewarnaan graf dapat dilihat pada Gambar 5.24.



Gambar 5. 24 Pewarnaan Graf Kasus Data Real

5.2.4.4 Hasil Rekomendasi Frekuensi Kasus Data Real

Berdasarkan hasil pewarnaan graf, selanjutnya input tersebut akan diolah dan diproses menjadi hasil rekomendasi frekuensi radio FM. Pemberian input posisi stasiun radio FM kasus data real menghasilkan beberapa stasiun radio FM memiliki frekuensi yang berbeda-beda. Hal tersebut dikarenakan beberapa node mengalami interferensi dan beberapa node tidak mengalami interferensi. Maka hasil rekomendasi frekuensi pada kasus data real memiliki 33 frekuensi. Rekomendasi frekuensi dikatakan optimum karena dengan jumlah stasiun pemancar yang banyak hanya memerlukan beberapa alokasi frekuensi untuk dapat diterapkan. Hasil rekomendasi frekuensi menggunakan perangkat lunak menunjukkan bahwa jumlah frekuensi yang digunakan sebanyak 33 frekuensi berbeda dan lebih optimum dibandingkan data real yang berjumlah 54 frekuensi berbeda. Hasil rekomendasi frekuensi dapat dilihat pada tabel 4.3

Tabel 5. 1 Rekomendasi Frekuensi Kasus Data Real

Stasiun Radio A frekuensi 96.5 FM	Stasiun Radio AE frekuensi 88.9 FM
Stasiun Radio B frekuensi 96.9 FM	Stasiun Radio AF frekuensi 87.7 FM
Stasiun Radio C frekuensi 93.3 FM	Stasiun Radio AG frekuensi 88.1 FM
Stasiun Radio D frekuensi 97.3 FM	Stasiun Radio AH frekuensi 87.7 FM
Stasiun Radio E frekuensi 97.7 FM	Stasiun Radio AI frekuensi 87.7 FM
Stasiun Radio F frekuensi 93.7 FM	Stasiun Radio AJ frekuensi 88.1 FM
Stasiun Radio G frekuensi 98.1 FM	Stasiun Radio AK frekuensi 88.5 FM
Stasiun Radio H frekuensi 90.5 FM	Stasiun Radio AL frekuensi 88.5 FM
Stasiun Radio I frekuensi 98.5 FM	Stasiun Radio AM frekuensi 88.9 FM
Stasiun Radio J frekuensi 90.9 FM	Stasiun Radio AN frekuensi 88.1 FM
Stasiun Radio K frekuensi 94.1 FM	Stasiun Radio AO frekuensi 88.5 FM
Stasiun Radio L frekuensi 88.9 FM	Stasiun Radio AP frekuensi 87.7 FM
Stasiun Radio M frekuensi 89.3 FM	Stasiun Radio AQ frekuensi 88.1 FM
Stasiun Radio N frekuensi 87.7 FM	Stasiun Radio AR frekuensi 92.9 FM
Stasiun Radio O frekuensi 88.1 FM	Stasiun Radio AS frekuensi 87.7 FM
Stasiun Radio P frekuensi 91.3 FM	Stasiun Radio AT frekuensi 87.7 FM
Stasiun Radio Q frekuensi 94.5 FM	Stasiun Radio AU frekuensi 88.1 FM
Stasiun Radio R frekuensi 94.9 FM	Stasiun Radio AV frekuensi 88.5 FM
Stasiun Radio S frekuensi 95.3 FM	Stasiun Radio AW frekuensi 89.3 FM
Stasiun Radio T frekuensi 98.9 FM	Stasiun Radio AX frekuensi 89.7 FM
Stasiun Radio U frekuensi 99.3 FM	Stasiun Radio AY frekuensi 89.3 FM
Stasiun Radio V frekuensi 99.7 FM	Stasiun Radio AZ frekuensi 87.7 FM
Stasiun Radio W frekuensi 100.1 FM	Stasiun Radio BA frekuensi 88.1 FM
Stasiun Radio X frekuensi 95.7 FM	Stasiun Radio BB frekuensi 90.1 FM
Stasiun Radio Y frekuensi 90.1 FM	Stasiun Radio BC frekuensi 90.5 FM
Stasiun Radio Z frekuensi 91.7 FM	Stasiun Radio BD frekuensi 92.9 FM
Stasiun Radio AA frekuensi 92.1 FM	Stasiun Radio BE frekuensi 88.9 FM
Stasiun Radio AB frekuensi 100.5 FM	Stasiun Radio BF frekuensi 87.7 FM
Stasiun Radio AC frekuensi 96.1 FM	Stasiun Radio BG frekuensi 89.7 FM
Stasiun Radio AD frekuensi 92.5 FM	Stasiun Radio BH frekuensi 88.5 FM
Stasiun Radio A frekuensi 96.5 FM	Stasiun Radio AE frekuensi 88.9 FM
Stasiun Radio B frekuensi 96.9 FM	Stasiun Radio AF frekuensi 87.7 FM

Stasiun Radio C frekuensi 93.3 FM	Stasiun Radio AG frekuensi 88.1 FM
Stasiun Radio D frekuensi 97.3 FM	Stasiun Radio AH frekuensi 87.7 FM
Stasiun Radio E frekuensi 97.7 FM	Stasiun Radio AI frekuensi 87.7 FM
Stasiun Radio F frekuensi 93.7 FM	Stasiun Radio AJ frekuensi 88.1 FM

BAB VI PENUTUP

Pada bab ini berisi tentang beberapa kesimpulan yang dihasilkan berdasarkan penelitian yang telah dilaksanakan. dan saran yang dapat digunakan jika penelitian ini dikembangkan.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program, maka dapat diambil kesimpulan sebagai berikut :

1. Metode pewarnaan graf telah berhasil diterapkan untuk menghindari interferensi frekuensi radio FM di Jawa Timur.
2. Algoritma *Welch Powell* dapat mengoptimalkan pemakaian frekuensi radio FM yang relatif terhadap posisi dan *coverage area* stasiun radio FM tanpa adanya interferensi.
3. Untuk pengujian pada Kasus 1 dimana semua pengguna tidak mengalami interferensi, Kasus 2 dimana semua pengguna mengalami, Kasus 3 dimana beberapa pengguna mengalami interferensi, menunjukkan bahwa perangkat lunak dapat digunakan untuk mengakomodasi stasiun radio sebanyak-banyaknya, dengan jumlah frekuensi seminimal mungkin.
4. Untuk pengujian pada kasus data real dimana data yang digunakan adalah data asli tahun 2006 menunjukkan bahwa perangkat lunak dapat menghasilkan pengalokasian frekuensi yang lebih optimal dibandingkan data frekuensi yang digunakan pada tahun 2006.

6.2 Saran

Ada beberapa hal yang penulis sarankan untuk pengembangan penelitian selanjutnya :

1. Perlu dilakukan studi lebih lanjut dengan gelombang radio merambat tidak pada bidang datar.
2. Perlu dilakukan studi lebih lanjut dengan perbandingan metode pewarnaan graf lainnya.

DAFTAR PUSTAKA

- [1] Hassan, A .M & Chickade, A. (2011). **A Review Of Interference Reduction In Wireless Networks Using Graph Coloring Methods**. Department of Computer Science, Villanova University, Villanova, USA..
- [2] Ning, Z & Yao, Y. (2015). **Interference-Aware Spectrum Sensing Mechanismsin Cognitive Radio Networks**. school of Information Science and Engineering, Northeastern University, Shenyang 110819, PR China.
- [3] Siahaan, M. (2011). **Studi Pewarnaan Graf Pada Radio Frequency Assignment Problem**. Program Studi Teknik Informatika. Sekolah Teknik Elektro dan Informatika Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia.
- [4] He, Y & Hongxi, Y. (2016). **Multiuser-Diversity-Based Interference Alignment In Cognitive Radionetwork**. School of Information and Communication Engineering, Dalian University of Technology, Dalian, Liaoning, PR China.
- [5] Munir, R. (2008). Diktat Kuliah IF2031: “**Struktur Diskrit**”. Departemen Teknik Informatika, Institut Teknologi Bandung.
- [6] Muhammad, N S. (2013). **Optimasi Radio Frequency Unit Berbasis Interferensi Kanal Menggunakan Pendekatan Upper Bound Bilangan Kromatik**. Program Studi Matematika Fakultas Sains dan Teknologi UIN Sunan Kalijaga. Jl. Marsda Adisucipto No. 1 Yogyakarta 55281
- [7] Marcus, M. (2012). **Clarifying Harmful Interference Will Facilitate Wireless Innovation**. The Institute of Electrical and Electronics Engineers-United States of America (IEEE-USA). 2001 L Street, NW Suite 700 Washington, DC 20036.

- [8] Ditjen Postel, (2016). “**Master Plan Penetapan Frekuensi Kanal Radio Siaran Fm**”. DEPKOMINFO, Jakarta.
- [9] Park. T & Lee. C Y. (1995). “**Application Of The Graph Coloring Algorithm To The Frequency Assignment Problem**”. Korea Advanced Institute of Science and Technology. April 12, 1995.
- [10] Priyono, R. (2010). “**Penerapan Pewarnaan Graf Dalam Penggunaan Frekuensi Radio**”. Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia. 29 April 2010.
- [11] Rosa A.S & Muhammad. S, (2015). “**Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek**”, Informatika Bandung, Indonesia. September 2015.

LAMPIRAN A

Data

- a. *Data Frekuensi Radio di Jawa Timur*
 Gambar data nama dan frekuensi radio di Jawa Timur.

XVI. PROPINSI JAWA TIMUR
 A. RADIO SIARAN KELAS B
 1 KOTA SURABAYA

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSISTING (MHZ)	MASTERPLAN KANAL	FREKUENSI (MHZ)
1	PT. Radio Zodiac	PM6FUC	87,80	2	87,7
2	PT. Radio Kota Buaya Mandiri	PM6FKA	87,95	6	88,1
3	PT. Radio Metro Gema Mega	PM6FNO	88,30	10	88,5
4	PT. Radio Surabaya Pesona Femina	PM6FUE	88,65	18	89,3
5	PT. Radio Harini Jaya Mandiri	PM6FNV	89,35	22	89,7
6	PT. Radio Media Caraka Angkasa	PM6FUG	90,40	26	90,1
7	PT. Radio Global Nada Prima	PM6FUQ	90,75	34	90,9
8	PT. Radio Suara Suzana Bhakti	PM6FUF	91,10	38	91,3
9	PT. Radio Kreasi Indah Dunia Suara	PM6FPV	92,15	50	92,5
10	PT. Radio Ekalaras Vioaksana Torya	PM6FKM	93,90	58	93,3
11	PT. Radio Suara Digital Indonesia	PM6FNW	94,25	69	94,4
12	PT. Radio Deviana Jelita	PM4FUT	94,60	73	94,8
13	RRI		95,30	77	95,2
14	PT. Radio Bahtera Swarayudha	PM6FKD	95,65	89	96,4
15	PT. Radio Mercury Masa Depan Sukses	PM6FKU	96,00	85	96,0
16	PT. Radio Shinta Warta Gemilang	PM6FKO	97,40	101	97,6
17	PT. Radio Salvatore Surabaya	PM6FKL	97,75	105	98,0
18	RRI		99,15	117	99,2
19	PT. Radio Fiskaria Jaya Suara Surabaya	PM6FJK	100,55	125	100,0
20	PT. Radio Laras Pancar Istana Suara	PM6FJI	100,90	136	101,1
21	PT. Radio Erabima Sakti Selaras	PM6FJJ	101,25	184	105,9
22	PT. Radio Stratosfir	PM6FJM	103,00	144	101,9
23	PT. Radio Widjaya	PM6FJQ	103,35	160	103,5
24	PT. Radio Bisnis Surabaya	PM6FNR	104,40	168	104,3
25	PT. Radio Cakra Awigra	PM6FJG	104,75	172	104,7
26	PT. Radio Wahana Informasi Gemilang	PM6FZB	105,45	176	105,1
27	PT. Radio Cakrawala Bhakti	PM6FZA	105,80	140	101,5
28	PT. Radio Camar	PM6FKC	106,50	156	103,1
29	PT. Radio Merdeka Lokatama	PM6FJD	106,85	192	106,7
30	PT. Radio Suara Mahasiswa Turun Bekerja	PM6FJF	107,90	152	102,7

1 BANGKALAN, KAB. BANGKALAN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Amanna Ibnu Kholil	PM6FNN	89,00	81	95,6

2 BANYUWANGI / GLAGAH, KAB. BANYUWANGI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Suara Banyuwangi	PM6FKB	90,40	22	89,7
2	PT. Radio Suara Mandala	PM6FPY	96,70	89	96,4
3	PT. Radio Suara Habibulloh	PM6FUD	99,50	73	94,8
4	PT. Radio Visi Inti Swara	PM6FPI	102,65	140	101,5
5	PT. Radio Genta Bawana Sakti	PM6FJV	107,55	156	103,1

3 MUNCAR / ROGOJAMPI, KAB. BANYUWANGI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Swarawangi Timur	PM6FNQ	91,80	81	95,6
2	PT. Radio Sri Tanjung Setia	PM6FHX	93,20	148	102,3

4 SEMPU / GENTENG / GAMBIRAN, KAB. BANYUWANGI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Gema Ujung Timur	PM6FNU	89,70	30	90,5
2	PT. Radio Mediagama FM	PM6FHT	98,80	113	98,8
3	PT. Radio Millenium Century FM	PM6FHQ	100,55	97	97,2
4	PT. Radio Suara Hati Insan Masyarakat	PM6FKF	105,45	180	105,5
5	PT. Radio Suara Gandrung	PM6FPZ	106,50	164	103,9

5 BANGOREJO / PASANGGARAN, KAB. BANYUWANGI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Suara Kenedes	PM6FHR	96,35	38	91,3
2	PT. Radio Suara Citra Bahana Angkasa	PM6FHZ	104,75	172	104,7

6 KALIBARU, KAB. BANYUWANGI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Swasta Merpati FM	PM6FKG	101,25	105	98,0

7 BLITAR (KOTA), KAB. BLITAR

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Mayangkara Ria	PM4FUR	97,05	135	101,0
2	PT. Radio Suara Patria	PM6FKK	97,40	151	102,6

8 WLINGI, KAB. BLITAR

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Purnama Cipta Prawira	PM6FNP	95,30	76	95,1

9 BOJONEGORO, KAB. BOJONEGORO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Bathara Suara Sakti	PM6FKI	97,05	91	96,6
2	PT. Radio Puspa Jaya	PM6FKS	98,45	142	101,7
3	PT. Radio Swara Bojonegoro Indah	PM6FNM	106,15	158	103,3

10 BONDOWOSO, KAB. BONDOWOSO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Romantika Suara Bondowoso	PM6FNS	95,30	111	98,6

10 BONDOWOSO, KAB. BONDOWOSO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Romantika Suara Bondowoso	PM6FNS	95,30	111	98,6

11 GRESIK, KAB. GRESIK

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	PT. Radio Giri Swara Indah Sakti	PM6FJS	104,05	109	98,4

12 JEMBER, KAB. JEMBER

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHZ)	MASTERPLAN	
				KANAL	FREKUENSI (MHZ)
1	RRI		91,10	20	89,5
2	PT. Radio Suara Akbar	PM6FKV	94,60	71	94,6
3	PT. Radio Kirana Indah Sentra Suara	PM6FJE	97,05	87	96,2
4	PT. Radio Soka Adiswara	PM6FNX	100,90	146	102,1

5	PT. Radio Prosa Lirik Nada	PM6FJN	101,60	138	101,3
---	----------------------------	--------	--------	-----	-------

13 JOMBANG, KAB. JOMBANG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	Koperasi Radio Nada Kencana	PM6FHV	92,60	40	91,5

14 PARE, KAB. KEDIRI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Sigma Rasitania	PM6FJH	107,55	194	106,9

15 KOTA KEDIRI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Suara Mahameru	PM6FNT	92,85	48	92,3
2	PT. Radio Wijang Songko	PM6FKQ	98,10	115	99,0
3	PT. Radio Swara Sabdotomo	PM6FJX	102,85	60	93,5
4	PT. Radio Swara Andhika Jaya	PM6FJT	106,50	182	105,7

16 LAMONGAN, KAB. LAMONGAN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Praja Media Swara	PM6FPG	107,20	164	103,9

17 LUMAJANG, KAB. LUMAJANG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Swara Semeru Pemail	PM6FHU	90,40	32	90,7
2	PT. Radio Gloria Paramitha Sampurna	PM6FHS	97,40	99	97,4

18 MADIUN (KOTA), KAB. MADIUN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	RRRI		99,15	122	99,7
2	PT. Radio Duta Cakrawala Serasi	PM6FJU	101,60	130	100,5
3	PT. Radio Moderato	PM6FUJ	105,45	197	107,2

19 MAGETAN, KAB. MAGETAN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI	
			FREKUENSI	MASTERPLAN

NO	PENYELENGGARA	CALLSIGN	EKSITING (MHz)	KANAL	FREKUENSI (MHz)
1	PT. Radio Suara Sarangan Indah	PM6FPS	90,40	31	90,6
2	PT. Radio Mitra Caraka Informasi Beragam	PM6FUN	100,90	98	97,3

20 MALANG (KOTA), KAB. MALANG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSITING (MHz)	MASTERPLAN	
		KANAL		FREKUENSI (MHz)	
1	PT. Radio Clarinta Makabu Utama	PM6FUI	89,00	12	88,7
2	PT. Radio Chakra Bhuwana	PM6FKW	90,05	20	89,5
3	PT. Radio Andhika Lugas Suara	PM6FKT	91,80	36	91,1
4	RRI		94,95	71	94,6
5	PT. Radio Suara Pionir	PM6FJC	96,70	87	96,2
6	PT. Radio Kharisma Dian Suara	PM6FJB	98,80	103	97,8
7	PT. Radio Suara Millenium	PM6FPF	99,50	79	95,4
8	PT. Radio Kalimaya Bhaskara	PM6FJL	100,20	146	102,1
9	PT. Radio Suara Rebana Turen	PM6FKY	101,80	138	101,3
10	PT. Radio Gita Loka Suara	PM6FJZ	106,15	154	102,9
11	PT. Radio Mitra Adiswara	PM6FHW	107,20	170	104,5

21 SINGOSARI, KAB. MALANG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSITING (MHz)	MASTERPLAN	
		KANAL		FREKUENSI (MHz)	
1	PT. Radio Puspita Hutama Nusantara	PM6FJR	103,70	162	103,7

22 MOJOKERTO, KAB. MOJOKERTO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSITING (MHz)	MASTERPLAN	
		KANAL		FREKUENSI (MHz)	
1	PT. Radio Anupadhatu Citra Indonesia	PM6FJA	93,20	65	94,0
2	KSU Radio Satria	PM6FHP	96,70	46	92,1
3	PT. Radio Maja Pari Indah Suara Mojokerto	PM6FJY	99,50	132	100,7

23 NGAWI, KAB. NGAWI

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSITING (MHz)	MASTERPLAN	
		KANAL		FREKUENSI (MHz)	
1	PT. Radio Bahana Ngawi Perkasa	PM6FPT	106,15	170	104,5

24 PASURUAN (KOTA), KAB. PASURUAN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSITING (MHz)	MASTERPLAN	
		KANAL		FREKUENSI (MHz)	
1	PT. Radio Pasuruan Warna Pesona	PM6FKP	91,45	61	93,6

25 PANDAAN / PRIGEN, KAB. PASURUAN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Parisuka Notasirama	PM6FJO	102,30	128	100,3
2	PT. Radio Star Wibawa Anugrah	PM6FKZ	107,55	180	105,5

26 PROBOLINGGO (KOTA), KAB. PROBOLINGGO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Angkasa Jaya Permai	PM6FKR	98,15	75	95,0

27 SAMPANG, KAB. SAMPANG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Pas Navaka	PM6FHN	96,35	66	94,1

28 SIDOARJO, KAB. SIDOARJO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	RRJ		96,35	188	106,3
2	PT. Radio Gitaya Gegana	PM6FKH	99,85	121	99,6

29 KRIAN, KAB. SIDOARJO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Aneka Getar Estetika	PM6FKE	93,55	54	92,9

30 WARU, KAB. SIDOARJO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Kartika Bahari Dirgantara Jaya	PM6FJP	101,95	113	98,8

31 SITUBONDO, KAB. SITUBONDO

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Bahana Swara Asem Bagus	PM6FUA	92,50	56	93,1

32 SUMENEP, KAB. SUMENEP

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	RRI		101,25	138	101,3

33 TUBAN, KAB. TUBAN

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Fajar Vidia Karya	PM6FUB	96,35	87	96,2
2	PT. Radio Sinar Natas Angin	PM6FZC	100,20	138	101,3
3	PT. Radio Tuban Pantai Pesona	PM6FHO	103,70	154	102,9

34 TULUNGAGUNG, KAB. TULUNGAGUNG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Paramita Jaya Perkasa	PM6FKX	96,35	93	96,8
2	PT. Radio Suara Tulung Agung Jaya	PM6FJW	104,40	160	103,5

35 NGUNUT, KAB. TULUNGAGUNG

NO	PENYELENGGARA	CALLSIGN	PENGALIHAN FREKUENSI		
			FREKUENSI EKSTING (MHz)	MASTERPLAN	
				KANAL	FREKUENSI (MHz)
1	PT. Radio Pandowo Anugrah Sakti	PM6FKJ	105,80	168	104,3

LAMPIRAN B

Source Code

1. Menu Home

```

public class InterfaceMain extends javax.swing.JFrame {
    public static int state = 0;
    public static Input Input = new Input();
    private static InterfaceMain a;
    private String user = "";
    private ArrayList<Koordinat> points = null;
    /**
     * Creates new form InterfaceMain
     */
    public InterfaceMain() {
        initComponents();
        new Thread(new
CekRadius(jComboBox1, Input)).start();
    }

    public InterfaceMain(String
user, ArrayList<Koordinat> points) {
        initComponents();
        new Thread(new
CekRadius(jComboBox1, Input)).start();
        this.user = user;
        lbl_username.setText(this.user);
        if(points != null) {
            Input.clear();
            Input.points = points;
            Input.repaint();
            setView();
        }
    }

    private void initComponents() {

        jPanel1 = new javax.swing.JPanel();
        jLabel2 = new javax.swing.JLabel();
        jLabel3 = new javax.swing.JLabel();
        jLabel4 = new javax.swing.JLabel();
        jPanel2 = Input;
        lbl_username = new javax.swing.JLabel();
        btn_next = new javax.swing.JButton();
        btn_clear = new javax.swing.JButton();
    }
}

```

```

        btn_back = new javax.swing.JButton();
        jComboBox1 = new javax.swing.JComboBox<>();
        jLabel5 = new javax.swing.JLabel();
        Home = new javax.swing.JLabel();
        jButton1 = new javax.swing.JButton();
        Foto = new javax.swing.JLabel();
        Load1 = new javax.swing.JButton();
        jLabel1 = new javax.swing.JLabel();
        public void
mouseClicked(java.awt.event.MouseEvent evt) {
            jButton1MouseClicked(evt);
        }
    });
    jPanel1.add(jButton1);
    jButton1.setBounds(30, 560, 110, 40);

    public void setView() {
        Input.maxColor = 0;
        if(state == 0) {
            btn_next.setVisible(true);
            btn_back.setVisible(false);
            btn_clear.setVisible(true);
            jComboBox1.setVisible(true);
            jLabel5.setVisible(true);
            Input.enable = true;
            Load1.setVisible(true);
        }
        else if(state == 1) {
            this.points = Input.points;
            btn_next.setVisible(true);
            btn_back.setVisible(true);
            btn_clear.setVisible(false);
            jComboBox1.setVisible(false);
            jLabel5.setVisible(false);
            Input.enable = false;
            Load1.setVisible(false);
        }
        else if(state == 2){
            btn_next.setVisible(true);
            btn_back.setVisible(true);
            btn_clear.setVisible(false);
            jComboBox1.setVisible(false);
            jLabel5.setVisible(false);
            Input.enable = false;
            Load1.setVisible(false);
        }
    }

    private void

```

```

btn_backMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    state--;
    setView();
    Input.state--;
    Input.repaint();
    jPanell = Input;
}

private void
btn_clearMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    int dialog = new PopUp().show_yesno("pesan",
"judul");
    if(dialog == 0) {
        Input.clear();
        Input.repaint();
        jPanell = Input;
    }
}

private void
btn_nextMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    if(state == 3) {
        InterfaceKesimpulan baru = new
InterfaceKesimpulan(this);
        baru.setVisible(true);
        this.setVisible(false);
    }
    else {
        state++;
        setView();
        Input.state++;
        if(state == 1) Input.cariTabrakan();
        else Input.print();
        Input.repaint();
        jPanell = Input;
    }
}

private void
HomeMouseClicked(java.awt.event.MouseEvent evt) {
    // TODO add your handling code here:
    state = 0;
    Input.state = 0;
    setView();
    Input.repaint();
    jPanell = Input;
}

```

```

java.awt.EventQueue.invokeLater(new Runnable() {
    public void run() {
        a = new InterfaceMain();
        a.setVisible(true);
        a.setResizable(false);
        Input.enable = true;
    }
});
}

```

2. *Input*

```

import java.awt.Color;
import java.awt.Dimension;
import java.awt.EventQueue;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Image;
import java.awt.Point;
import java.awt.RenderingHints;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
import java.io.IOException;
import java.net.URL;
import java.util.ArrayList;
import javax.imageio.ImageIO;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
/**
 *
 * @author fadhlan
 */
public class Input extends JPanel {
    ArrayList<Koordinat> points;
    private Point now, hapus;
    public ArrayList<Point> tabrakan = new ArrayList<>();
    int radiusLuar = 30, radiusPusat = 5;
    public int state = 0; int maxColor = 0;
    public boolean enable = true, clicked = false;
    private GetColor color = new GetColor();
    private static final String abjad =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    private ArrayList<JLabel> label_huruf;
    public static int CANVAS_WIDTH = 640;
    public static int CANVAS_HEIGHT = 480;

    private String imgFileName =
"SimulasiStasiunRadio/Peta.png";

```

```

private Image img;

public Input(){
    points = new ArrayList();
    label_huruf = new ArrayList();
    loadImage();
    addMouseListener(new MouseAdapter() {
        @Override
        public void mousePressed(MouseEvent e) {
            if(enable) {
                if(e.getButton() ==
MouseEvent.BUTTON1) {
                    now = new Point(e.getX(),
e.getY());
                    points.add(new
Koordinat(now, radiusLuar));
                }
                else if(e.getButton() ==
MouseEvent.BUTTON3) {
                    hapus = new Point(e.getX(),
e.getY());
                    delete(hapus);
                    System.out.println("hapus");
                }
                repaint();
            }
        }
    });
}

public void paintComponent(Graphics g){
    super.paintComponent(g);
    Graphics2D g2 = (Graphics2D) g;

    g2.drawImage(img, 0, 0, null);

    g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
RenderingHints.VALUE_ANTIALIAS_ON);
    g2.setColor(Color.black);

    if(state == 1 || state == 2) gambarGaris(g);
    boolean x = true;
    for(int i=0; i < points.size(); i++) {
        int temp_warna;
        if(state == 0 || state == 1) temp_warna = 0;
        else {
            temp_warna = points.get(i).warna;
            if(maxColor<temp_warna){
                maxColor=temp_warna;
            }
        }
    }
}

```

```

        }
        gambarPusat(g,i,points.get(i),temp_warna);
        if(state != 1)
gambarLingkaran(g,points.get(i).point,points.get(i).radius,
temp_warna);
        if(state == 2 && x) {
            x=false;
System.out.println(Integer.toString(temp_warna));
        }
    }
    public void cariTabrakan() {
        tabrakan.clear();
        for(int i=0;i<points.size();i++) {
            for(int j=i+1;j<points.size();j++) {
if(berpotongan(points.get(i).point,points.get(j).point,po
ints.get(i).radius,points.get(j).radius)) {
                tabrakan.add(new Point(i,j));
                tabrakan.add(new Point(j,i));
            }
        }
    }
}

private void delete(Point hapus) {
    int i=0,k=-1;
    for(i=0;i<points.size();i++) {
        if(dekat(points.get(i).point,hapus)) {
            points.remove(points.get(i));
            break;
        }
    }
}

public void clear() {
    points.clear();
    this.label_huruf.clear();
}

private boolean dekat(Point a,Point b) {
    if((a.x - b.x)*(a.x - b.x)+(a.y - b.y)*(a.y -
b.y) <= 4*radiusPusat*radiusPusat) return true;
    else return false;
}

private String text(int ke) {

```

```

        String hasil = "";
        hasil += abjad.charAt(ke%26);
        while(ke >= 26) {
            ke /= 26;
            hasil = abjad.charAt(ke%26-1) + hasil;
        }
        return hasil;
    }

    public void gambarPusat(Graphics g,int ke,Koordinat
point,int warna) {
        g.setColor(Color.decode(color.cariwarna(warna)));
        g.fillOval(point.point.x-
this.radiusPusat,point.point.y-
this.radiusPusat,this.radiusPusat*2,this.radiusPusat*2);
        g.drawString(text(ke), point.x-point.radius/2,
point.y-point.radius/2);
        g.setColor(Color.BLACK);
    }

    public void gambarLingkaran(Graphics g,Point
pusat,int radius,int warna) {
        g.setColor(Color.decode(color.cariwarna(warna)));
        g.drawOval(pusat.x-radius,pusat.y-
radius,radius*2,radius*2);
    }

    public void gambarGaris(Graphics g) {
        for(int i = 0; i < tabrakan.size(); i++) {
            int x1 = points.get(tabrakan.get(i).x).x;
            int y1 = points.get(tabrakan.get(i).x).y;
            int x2 = points.get(tabrakan.get(i).y).x;
            int y2 = points.get(tabrakan.get(i).y).y;
            g.drawLine(x1,y1,x2,y2);
        }
    }

    private boolean berpotongan(Point pusat1,Point
pusat2,int radius1, int radius2) {
        if((pusat1.x - pusat2.x)*(pusat1.x -
pusat2.x)+(pusat1.y - pusat2.y)*(pusat1.y - pusat2.y) <=
(radius1+radius2)*(radius1+radius2)) return true;
        else return false;
    }

    public void print() {
        ArrayList<Point> x = tabrakan;
        ArrayList<String> data = new ArrayList();
        for(int i=0;i<points.size();i++) {
            String a = i+":";
            for(int j=0;j<x.size();j++) {

```



```

        if(x.get(j).x == i) {
            a+="-"+x.get(j).y;
        }
    }
    System.out.println(a);
    data.add(a);
}
Graph2 graph = new Graph2(data);
graph.colourVertices();
for(int i=0;i<points.size();i++) {
    points.get(i).warna = graph.color[i]-1;
}
}
public void loadImage() {
    URL imgUrl =
getClass().getClassLoader().getResource(imgFileName);
    if (imgUrl == null) {
        System.err.println("Couldn't find file: " +
imgFileName);
    } else {
        try {
            img = ImageIO.read(imgUrl);
            CANVAS_WIDTH = img.getWidth(this);
            CANVAS_HEIGHT = img.getHeight(this);
//System.out.println(Integer.toString(width)+"
"+Integer.toString(height));
        } catch (IOException ex) {
            ex.printStackTrace();
        }
    }
    setPreferredSize(new Dimension(CANVAS_WIDTH,
CANVAS_HEIGHT));
}
public static void main(String[] args) {
    EventQueue.invokeLater(new Runnable() {
        @Override
        public void run() {
            JFrame frame = new JFrame();
            frame.add(new Input());

frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
            frame.setSize(CANVAS_WIDTH,
CANVAS_HEIGHT);
            frame.setVisible(true);
        }
    });
}
}

```

```
}

```

3. Graph

```
import java.util.List;
import java.io.IOException;
import java.nio.charset.Charset;
import java.nio.file.FileSystems;
import java.nio.file.Files;
import java.nio.file.Path;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
/**
 *
 * @author fadhlan
 */
public class Graph2 {
    public List<Vertex> vertices; //Array list tetangga
    public int[] color;//Array color
    public Graph2(ArrayList<String> lines) {
        vertices = new ArrayList();
        if (lines != null) {
            for (int i = 0; i < lines.size(); i++)
                //Baca banyaknya isi Arraylist
                int node =
                Integer.parseInt(lines.get(i).split(":")[0]);
                String[] adj;
                if(lines.get(i).split(":").length == 2) {
                    adj =
                    lines.get(i).split(":")[1].split("-");
                    int[] neighbors = new
                    int[adj.length];
                    for (int j = 1; j < adj.length; j++)
                    {
                        neighbors[j] =
                        Integer.parseInt(adj[j]);
                    }
                    vertices.add(new Vertex(node,
                    neighbors));
                }
                else vertices.add(new Vertex(node,
                null));
            }
        }
    }
}
```

```

private int cari(int[] dilarang) {
    boolean ketemu = false;
    int i = 0;
    for (i = 0; i < vertices.size(); i++) {
        if(dilarang[i] == 0 &&
color[vertices.get(caril(i)).node] == 0) {
            ketemu = true;
            break;
        }
    }
    if(ketemu) return i;
    else return -1;
}

private int caril(int key) { //nyari node key da di
vertices berapa.
    for (int i = 0; i < vertices.size(); i++) {
        if(vertices.get(i).node == key) return i;
    }
    return -1;
}

public void colourVertices(){
    Collections.sort(vertices, new
VertexComparator()); // arrange vertices in order of
descending valence
    color = new int[vertices.size()]; //membuat warna
    int warna = 1;
    for (int i = 0; i < vertices.size(); i++) {
        if(vertices.get(i).neighbors==null){
            color[vertices.get(i).node]=warna;
            continue;
        }
        if(color[vertices.get(i).node] != 0)
continue;
        color[vertices.get(i).node] = warna; //diwarna
1
        int[] dilarang = new int[vertices.size()];
        dilarang[vertices.get(i).node] = 1;
        for(int j = 1; j <
vertices.get(i).neighbors.length; j++) {
            dilarang[vertices.get(i).neighbors[j]] =
1; //warna tetangga menjadi 1
        }
        while(true) {
            int temp = cari(dilarang);
            if(temp == -1) break;
            color[temp] = warna;

```

```

        dilarang[temp] = 1;
        for(int j = 0; j <
vertices.get(caril(temp)).neighbors.length; j++) {
dilarang[vertices.get(caril(temp)).neighbors[j]] =
1;//diberi warna yang berbeda dengan tetangganya
        }
        warna++;
    }
}

private static List<String> readGraphData(String
_path){
    Path path =
FileSystems.getDefault().getPath(_path, "");
    try {
        return Files.readAllLines(path,
Charset.defaultCharset());
    } catch (IOException e) {
        System.err.println("I/O Error");
        return null;
    }
}

private static class Vertex{//class isi tentang isi
node dan tetangganya
    private int node;
    private int[] neighbors;

    public Vertex(int node, int[] neighbors){
        this.node = node;
        this.neighbors = neighbors;
    }
}

class VertexComparator implements
Comparator<Vertex>{//membandingkan skala prioritas
pewarnaan graf
    @Override
    public int compare(Vertex a, Vertex b) {
        if(a.neighbors == null) return -1;
        else if(b.neighbors == null) return 1;
        else if(a.neighbors.length <
b.neighbors.length) {
            return 1;
        }
        else if(a.neighbors.length ==
b.neighbors.length) return 0;

```

```

        else return -1;
    }
}
public static void main(String[] args) {
    Graph2 graph = new Graph2((ArrayList<String>)
readGraphData("F:\\tes.txt"));
    graph.colourVertices();
}
}

```

4. *Data Warna*

```

/**
 *
 * @author fadhlan
 */
public class GetColor {
    private static final String[] color = new String[] {
"#000000", "#FFFF00", "#1CE6FF", "#FF34FF", "#FF4A46", "#00894
1", "#006FA6", "#A30059",
"#FFDBE5", "#7A4900", "#0000A6", "#63FFAC", "#B79762", "#004D4
3", "#8FB0FF", "#997D87",
"#5A0007", "#809693", "#FEFF66", "#1B4400", "#4FC601", "#3B5DF
F", "#4A3B53", "#FF2F80",
"#61615A", "#BA0900", "#6B7900", "#00C2A0", "#FFAA92", "#FF90C
9", "#B903AA", "#D16100",
"#DDEFFF", "#000035", "#7B4F4B", "#A1C299", "#300018", "#0AA6D
8", "#013349", "#00846F",
"#372101", "#FFB500", "#C2FFED", "#A079BF", "#CC0744", "#C0B9B
2", "#C2FF99", "#001E09",
"#00489C", "#6F0062", "#0CBD66", "#EEC3FF", "#456D75", "#B77B6
8", "#7A8741", "#788D66",
"#885578", "#FAD09F", "#FF8A9A", "#D157A0", "#BEC459", "#45664
8", "#0086ED", "#886F4C",
"#34362D", "#B4A8BD", "#00A6AA", "#452C2C", "#636375", "#A3C8C
9", "#FF913F", "#938A81",
"#575329", "#00FECE", "#B05B6F", "#8CD0FF", "#3B9700", "#04F75
7", "#C8A1A1", "#1E6E00",

```

```

"#7900D7", "#A77500", "#6367A9", "#A05837", "#6B002C", "#77260
0", "#D790FF", "#9B9700",

"#549E79", "#FFF69F", "#201625", "#72418F", "#BC23FF", "#99ADC
0", "#3A2465", "#922329",

"#5B4534", "#FDE8DC", "#404E55", "#0089A3", "#CB7E98", "#A4E80
4", "#324E72", "#6A3A4C",

"#83AB58", "#001C1E", "#D1F7CE", "#004B28", "#C8D0F6", "#A3A48
9", "#806C66", "#222800",

"#BF5650", "#E83000", "#66796D", "#DA007C", "#FF1A59", "#8ADBB
4", "#1E0200", "#5B4E51",

"#C895C5", "#320033", "#FF6832", "#66E1D3", "#CFCDAC", "#D0AC9
4", "#7ED379", "#012C58"
};
public GetColor(){
}
public String cariwarna(int n){
return color [n];
}

```

5. *Rekomendasi Frekuensi*

```

private void
jButton1MouseClicked(java.awt.event.MouseEvent evt) {
// TODO add your handling code here:
float mulai = (float) 87.7;
float sampai = (float) 108.7;
float beda = (float) 0.4;
double jml = (double) ((sampai-mulai)/beda);
System.out.println(jml+" "+main.Input.maxColor);
if(main.Input.maxColor*beda+mulai>=105 && jml >=
main.Input.maxColor){
new PopUp().show_notif("", "Peringatan");
InterfaceKesimpulan baru = new
InterfaceKesimpulan(main,mulai,beda);
baru.setVisible(true);
this.setVisible(false);
}
else if(jml >= main.Input.maxColor) {
InterfaceKesimpulan baru = new
InterfaceKesimpulan(main,mulai,beda);
baru.setVisible(true);
}
}

```

```

        this.setVisible(false);
    }
    else {
        new PopUp().show_error("Range dan interval
tidak mencukupi atau inputan titik terlalu
banyak", "Failed");
    }
}
import javax.swing.JTextArea;

/**
 *
 * @author fadhlan
 */
public class InterfaceKesimpulan extends
javax.swing.JFrame {
    public InterfaceMain main = null;
    private static final String abjad =
"ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    JTextArea kiri = new JTextArea();
    JTextArea kanan = new JTextArea();
    /**
     * Creates new form InterfaceKesimpulan
     */
    public InterfaceKesimpulan() {
        initComponents();
    }

    public InterfaceKesimpulan(InterfaceMain main, float
mulai, float beda) {
        initComponents();
        String textnya_kiri = "";
        String textnya_kanan = "";
        this.main = main;
        if(main.Input.points != null) {
            int n = main.Input.points.size();
            int[] array = new int[n];
            int now = 60;
            double[] frekuensi = new double[n];
            boolean ketemu = false;
            int ke = 0, pengali = 0;
            while(true) {
                ketemu = false;
                for(int i=0; i<n; i++) {
                    if(array[i] == 1) continue;
                    if(main.Input.points.get(i).radius ==
now) {
                        ketemu = true;
                        ke = i;

```

```

                break;
            }
        }
        if(ketemu) {
            frekuensi[ke] = mulai+beda*pengali;
            array[ke] = 1;
            int temp_warna =
main.Input.points.get(ke).warna;
            for(int j=0;j<n;j++) {
                if(array[j] == 1) continue;
                if(main.Input.points.get(j).warna
== temp_warna) {
                    frekuensi[j] =
mulai+beda*pengali;
                    array[j] = 1;
                }
            }
            pengali++;
        }
        else {
            if(now > 30) now-=10;
            else break;
        }
    }

    for(int i=0;i<n;i++) {
        String temp_kiri = "Stasiun Radio
"+text(i)+"\n";
        double frekuensinya =
Double.parseDouble(new
DecimalFormat("##.#").format(frekuensi[i]));
        String temp_kanan = "Stasiun Radio
"+text(i)+" frekuensi "+frekuensinya+" FM\n";
        textnya_kiri+=temp_kiri;
        textnya_kanan+=temp_kanan;
    }
    int max_warna = main.Input.maxColor+1;
    String text_warna = "Jumlah warna yang diperlukan
: "+max_warna+" warna";
    warna.setText(text_warna);
    warna.setEditable(false);

    kiri.setText(textnya_kiri);
    kiri.setEditable(false);
    kiri.setCaretPosition(0);
    kanan.setText(textnya_kanan);
    kanan.setEditable(false);
    kanan.setCaretPosition(0);

```



```
}

private int cari(int[] dilarang,int n) {
    boolean ketemu = false;
    int i = 0;
    for (i = 0; i < n; i++) {
        if(dilarang[i] == 0) {
            ketemu = true;
            break;
        }
    }
    if(ketemu) return i;
    else return -1;
}

private String text(int ke) {
    String hasil = "";
    hasil += abjad.charAt(ke%26);
    while(ke >= 26) {
        ke /= 26;
        hasil = abjad.charAt(ke%26-1) + hasil;
    }
    return hasil;
}
```


BIODATA PENULIS



Penulis bernama lengkap **Fadhlhan Septianto Eka Putra**, lahir di Bekasi, 14 September 1995. Anak Pertama dari pasangan Lestariyono dan Palekah, serta memiliki adik perempuan Adhifa Putri Lestari. Penulis mengikuti pendidikan dasar dari Sekolah Dasar hingga Sekolah Menengah Atas di Kota Bekasi. Penulis menempuh pendidikan di SD Negeri Pengasinan 8, SMPN Bekasi, dan SMA Negeri 8 Bekasi. Setelah Lulus dari SMAN 8 Bekasi pada tahun 2013 yang lalu, penulis melanjutkan pendidikan tingginya di Institut Teknologi Sepuluh Nopember (ITS) Surabaya dengan mengambil Departemen Matematika dengan bidang minat Ilmu Komputer. Selama mengikuti perkuliahan di ITS, penulis turut aktif dalam beberapa kegiatan kemahasiswaan sebagai staff Departemen Sains, Teknologi, dan Keprofesian Himatika ITS Periode 2014/2015, dan Ketua Divisi Keilmiahan Aplied Science Department Himatika ITS Periode 2015/2016. Selain aktif dalam beberapa kegiatan kemahasiswaan, penulis juga mengikuti Kerja Praktek di PT.Berlian Sistem Informasi dan ditempatkan di divisi ERP Medium Project, penulis mengerjakan project Budget Add On di PT.Berlian Sistem Informasi. Informasi lebih lanjut mengenai Tugas Akhir ini dapat ditujukan ke penulis melalui email: septiantofadhlhan@gmail.com.