

LAPORAN SKRIPSI

**PERANGKAT LUNAK *GRAPHIC USER INTERFACE*
(*GUI*) UNTUK SIMULASI GELOMBANG
ELEKTROMAGNETIK MENGGUNAKAN METODE
*FINITE DIFFERENCE TIME DOMAIN***

**Disusun oleh:
Eki Prasetya
065115202**



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS PAKUAN
BOGOR
2022**

HALAMAN PENGESAHAN

Judul : Perangkat Lunak *Graphic User Interface (GUI)* Untuk Simulasi Gelombang Elektromagnetik Menggunakan Metode *Finite Difference Time Domain*
Nama : Eki Prasetya
NPM : 065115202

Mengesahkan,

Pembimbing Pendamping
Program Studi Ilmu Komputer
FMIPA - UNPAK

Pembimbing Utama
Program Studi Ilmu Komputer
FMIPA - UNPAK

Teguh Puja Negara, M.Si.

Prof. Dr. -Ing. Soewarto Hardhienata

Mengetahui,

Ketua Program Studi Ilmu Komputer
FMIPA - UNPAK

Dekan
FMIPA - UNPAK

Arie Qur'ania, M.Kom.

Asep Denih, S.Kom., M.Sc., Ph.D.

PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI

Dengan ini saya yang bertanda tangan dibawah ini :

Nama : **Eki Prasetya**
NPM : **065115202**
Program Studi : **Ilmu Komputer**
Fakultas : **Matematika dan Ilmu Pengetahuan Alam**
Universitas Pakuan Bogor

Menyatakan bahwa sejauh yang saya ketahui, karya tulis ini bukan merupakan karya tulis yang pernah dipublikasikan atau sudah pernah dipakai untuk mendapatkan gelar sarjana di Universitas lain, kecuali pada bagian-bagian di mana sumber informasinya dicantumkan dengan cara referensi yang semestinya.

Demikian pernyataan ini saya buat dengan sebenar-benarnya. Apabila kelak dikemudian hari terdapat gugatan, penulis bersedia dikenakan sanksi sesuai dengan peraturan yang berlaku.

Bogor, Juli 2022

Eki Prasetya

PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA

Dengan ini saya yang bertanda tangan dibawah ini :

Nama : **Eki Prasetya**
NPM : **065115202**
Program Studi : **Ilmu Komputer**
Judul Skripsi : **Perangkat Lunak *Graphic User Interface (GUI)*
Untuk Simulasi Gelombang Elektromagnetik
Menggunakan Metode *Finite Difference Time Domain***

Dengan ini saya menyatakan bahwa paten dan hak cipta dari produk skripsi dan tugas akhir di atas adalah benar karya saya dengan arahan dari pembimbing dan belum diajukan dalam bentuk apapun kepada perguruan tinggi manapun.

Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam daftar Pustaka dibagian akhir Skripsi ini.

Dengan ini saya melimpahkan Paten, hak cipta dan karya tulis saya kepada Universitas Pakuan.

Bogor, Juli 2022

Eki Prasetya

RIWAYAT HIDUP



Penulis dilahirkan di Bogor pada tanggal 1 Oktober 1997 dari pasangan Bapak Hery Priyatna dan Ibu Yati Maryati sebagai anak ke-tiga dari tiga bersaudara. Penulis memulai pendidikan dari Taman Kanak-Kanak pada tahun 2002 dan menyelesaikan pada tahun 2003 di TK Amaliah Ciawi, Kabupaten Bogor, Jawa Barat. Melanjutkan pendidikan Sekolah Dasar pada tahun 2004 dan menyelesaikan pada tahun 2009 di SDN Lawanggintung 1, Kota Bogor, Jawa Barat. Kemudian penulis melanjutkan pendidikan Sekolah Menengah Pertama pada tahun yang sama dan menyelesaikan pada tahun 2012 di SMP Negeri 18, Kota Bogor, Jawa Barat. Pendidikan Dilanjutkan di Sekolah Menengah Atas pada tahun 2012 dan meraih kelulusan pada tahun 2015 di SMA Negeri 4, Kota Bogor, Jawa barat.

Setelah lulus SMA penulis melanjutkan pendidikan perguruan tinggi di Universitas Pakuan Bogor, Fakultas Matematika dan Ilmu Pengetahuan Alam, Program Studi S1 Ilmu Komputer pada tahun 2015. Pada bulan Juli 2022 penulis menyelesaikan tugas akhir yang berjudul “**Perangkat Lunak *Graphic User Interface (GUI)* Untuk Simulasi Gelombang Elektromagnetik Menggunakan Metode *Finite Difference Time Domain***” dan dinyatakan lulus sebagai Sarjana Komputer setelah melalui sidang komprehensif pada tanggal 5 Agustus 2022 dengan predikat **Cukup**.

KATA PENGANTAR

Puji syukur kehadirat Allah SWT, karena rahmat dan hidayah-Nya penulis dapat menyusun dan menyelesaikan penulisan penelitian yang berjudul “**Perangkat Lunak *Graphic User Interface (GUI)* untuk Simulasi Gelombang Elektromagnetik Menggunakan Metode *Finite Difference Time Domain***”.

Laporan ini diajukan sebagai salah satu syarat memperoleh gelar sarjana di Program Studi Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Pakuan Bogor. Pada kesempatan ini penulis ingin mengucapkan terimakasih kepada:

- Prof. Dr. -Ing. Soewarto Hardhienata selaku Pembimbing Utama, yang telah membimbing dan memberikan pengarahan dan petunjuk dalam penulisan usulan penelitian ini,
- Teguh Puja Negara, M.Si. selaku Pembimbing Pendamping, yang telah membimbing dan memberikan masukan dan petunjuk dalam penulisan usulan penelitian ini,
- Orang tua dan keluarga yang selalu memberikan dorongan semangat, doa, nasehat, kasih sayang dan menjadi sumber inspirasi terkuat bagi penulis untuk menyelesaikan usulan penelitian ini,
- Semua teman-teman yang telah membantu dan memberikan semangat dalam penyusunan skripsi ini,

Penulis menyadari masih banyak kekurangan didalam penulisan usulan penelitian ini, baik dari tata tulis dan tata bahasa, serta keterbatasan materi yang dimuat di dalamnya. Oleh karena itu penulis berharap adanya masukan dan saran yang sifatnya membangun dalam penulisan usulan penelitian ini. Akhir kata semoga Penelitian ini dapat bermanfaat sesuai dengan maksud dan tujuan khususnya bagi penulis dan bagi para pembaca umumnya.

Bogor, Juli 2022

Eki Prasetya

DAFTAR ISI

HALAMAN PENGESAHAN	I
PERNYATAAN KEASLIAN KARYA TULIS SKRIPSI.....	II
PERNYATAAN PELIMPAHAN SKRIPSI DAN SUMBER INFORMASI SERTA PELIMPAHAN HAK CIPTA	III
RIWAYAT HIDUP	IV
KATA PENGANTAR	V
DAFTAR ISI.....	VI
DAFTAR GAMBAR.....	VIII
DAFTAR TABEL.....	IX
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Tujuan.....	2
1.3 Ruang lingkup	2
1.4 Manfaat.....	2
BAB II TINJAUAN PUSTAKA	3
2.1 Tinjauan Pustaka	3
2.1.1 Persamaan Maxwell	3
2.1.2 Metode FDTD	4
2.1.3 GUI Matlab	8
2.2 Penelitian Terdahulu.....	9
2.3 Tabel Perbandingan Penelitian	10
BAB III METODE PENELITIAN	11
3.1 Tahapan Penelitian	11
3.1.1 Perencanaan Proyek Penelitian (<i>Project Planning</i>).....	12
3.1.2 Penelitian (<i>Reaserch</i>)	12
3.1.3 Disain persamaan Analitik	12
3.1.3.1 Parameter <i>Input</i>	12
3.1.3.2 Initial Condition.....	13
3.1.3.3 Persamaan Syarat Batas	13
3.1.3.4 Persamaan Medan	13
3.1.3.5 Persamaan Output.....	13
3.1.4 Disain Software	13
3.1.5 Implementasi Program	15
3.1.6 Pengujian.....	15
3.1.7 Aplikasi	16
3.2 Alat Dan Bahan	16
BAB IV PERANCANGAN DAN IMPLEMETASI	17
4.1 Perancangan.....	17
4.1.1 Code Interface GUI.....	17
4.1.1.1 Parameter <i>Input</i>	17

4.1.1.2 Tombol Start	18
4.1.1.3 Axes	19
4.1.1.4 Toolbar.....	19
4.1.2 Code Program Utama.....	19
4.1.2.1 Parameter Input.....	19
4.1.2.2 Initial Condition.....	20
4.1.2.3 Persamaan Syarat Batas	20
4.1.2.4 Persamaan Medan	20
4.1.2.5 Persamaan <i>Output</i>	21
4.2 Implementasi Program.....	22
4.2.1 Tampilan Program Utama Pada M-File	22
4.2.2 Tampilan Program GUI Pada M-File.....	22
BAB V HASIL DAN PEMBAHASAN	24
5.1 Hasil.....	24
5.2 Pengujian	25
5.2.1 Uji Coba Fungsional	25
5.2.2 Uji Coba Validasi.....	27
5.2.2.1 Uji Parameter Domain Komputasi.....	27
5.2.2.2 Uji Parameter <i>Grid</i>	28
5.2.2.3 Uji Parameter PML.....	29
5.2.2.4 Uji Parameter Waktu	30
5.2.2.5 Uji Parameter Bahan.....	31
BAB VI KESIMPULAN DAN SARAN	34
6.1 Kesimpulan.....	34
6.2 Saran	34
DAFTAR PUSTAKA	35
LAMPIRAN.....	37

DAFTAR GAMBAR

Gambar 1. Gelombang Elektromagnetik	3
Gambar 2. Step FDTD Dua Dimensi.....	4
Gambar 3. Domain Komputasi Dengan PML	7
Gambar 4. Wilayah TF/SF Pada Domain Komputasi.....	8
Gambar 5. Aplikasi Analisis Spektrum dan <i>running</i> pada Windows di luar Matlab ..	9
Gambar 6. Diagram Alir Penelitian	11
Gambar 7. Diagram Alir Program Utama.....	14
Gambar 8. Diagram Alir Sistem GUI	15
Gambar 9. Code Program Utama.....	22
Gambar 10. Code Pembuatan GUI	23
Gambar 11. Tampilan Aplikasi Sebelum Dijalankan	24
Gambar 12. Tampilan Aplikasi Setelah Dijalankan	25
Gambar 13. Parameter Domain Komputasi Dengan Nilai 100	27
Gambar 14. Parameter Domain Komputasi Dengan Nilai 300	28
Gambar 15. Parameter Grid dengan Nilai 0.01	28
Gambar 16. Parameter Grid Dengan Nilai 99.....	29
Gambar 17. Parameter PML dengan Nilai 5.....	29
Gambar 18. Parameter PML dengan Nilai 8.....	30
Gambar 19. Uji Parameter Waktu Dengan Nilai 50	30
Gambar 20. Uji Parameter Waktu Dengan Nilai 150	31
Gambar 21. Parameter Bahan Udara	31
Gambar 22. Parameter Bahan Dielectric (SiO ₂)	32
Gambar 23. Parameter Bahan Lossy Dielectric (Germanium)	32
Gambar 24. Parameter Bahan Metal (Tembaga)	33

DAFTAR TABEL

Tabel 1. Perbandingan Penelitian	10
Tabel 2. Parameter dan Simbol Pada Code FDTD di Matlab.....	12
Tabel 3. Simbol Medan Pada FDTD di Matlab	13
Tabel 4. Uji Coba Fungsional	25

BAB I

PENDAHULUAN

1.1 Latar belakang

Persamaan Maxwell merupakan sekumpulan persamaan yang mendeskripsikan sifat-sifat listrik dan magnet dan hubungannya dengan sumbernya (Tjia, 1994). Solusi dari persamaan Maxwell merupakan objek yang terus dikaji, karena banyak fenomena baru dalam gelombang elektromagnetik yang muncul dan dapat dikembangkan ke dalam piranti optik (Kulyabov, 2017). Pemecahan persamaan Maxwell cukup sulit karena disamping persamaan-persamaan tersebut saling terkait (*tercouple*), bentuk persamaan tersebut juga melibatkan persamaan differensial atau integral (David, 1995). Solusi persamaan Maxwell dapat dikaji berdasarkan orientasi penjalaran gelombang elektromagnet dalam sumbu x atau y (satu dimensi), sumbu x dan y (dua dimensi), dan sumbu x, y, dan z (tiga dimensi). Beberapa metode dalam memecahkan persamaan Maxwell terus dikembangkan hingga saat ini, yaitu metode analitik maupun metode numerik. Metode analitik dapat memecahkan persamaan Maxwell tetapi dibutuhkan kemampuan matematika yang cukup tinggi yaitu: differensial, vektor, matriks dan integral. Sedangkan metode numerik dapat lebih efisien digunakan tetapi membutuhkan kemampuan membuat struktur algoritma dan ruang komputasi yang cukup besar untuk persamaan Maxwell tiga dimensi. Bagaimanapun, metode numerik lebih unggul dalam memecahkan persamaan Maxwell dari sisi efisiensi, fleksibilitas dan kemudahan dalam mensimulasikan gelombang elektromagnetik. Salah satu metode numerik yang dapat memecahkan persamaan Maxwell adalah metode *Finite Difference Time Domain (FDTD)*.

Beberapa penelitian terkait metode FDTD telah digunakan dalam memecahkan persamaan gelombang oleh beberapa peneliti, diantaranya yang dilakukan oleh Azhar yang menggunakan metode FDTD untuk mensimulasikan gelombang dua dimensi dengan menganalisis simpangan berdasarkan variasi parameter integrasi yaitu M dan alfa (Azhar, 2000). Hasil simulasi menunjukkan jika parameter M dinaikan maka frekuensi gelombang ini menjadi naik tetapi amplitudonya turun. Penelitian lain dilakukan oleh Hadi Teguh Yudistira dan tim nya yang menerapkan metode FDTD untuk simulasi *phased array antenna*, antena yang dapat memancarkan dan juga menerima radiasi gelombang elektromagnetik dari sudut arah yang cukup luas tanpa menggerakkan antenanya (Hadi, 2017). Penelitian-penelitian yang dilakukan masih sebatas menghasilkan solusi numerik dan belum membangun perangkat lunak yang dapat digunakan untuk berbagai kasus.

Berdasarkan latar belakang dan penelitian yang telah dilakukan, maka dirancang sebuah perangkat lunak berbasis Matlab menggunakan metode FDTD 2 dimensi. Perangkat lunak tersebut berbentuk *Grafik User Interface (GUI)* yang dapat menampilkan simulasi gelombang elektromagnetik dalam ruang hampa maupun dalam medium. Penerapan syarat batas (*boundary condition*) dapat menunjukkan tidak adanya pantulan pada sisi domain komputasi. Perangkat lunak ini dapat dikembangkan kearah dimensi yang lebih tinggi dan diterapkan untuk simulasi perangkat optik seperti: sensor, *waveguide*, filter optik, dan lain-lain.

1.2 Tujuan

Penelitian bertujuan untuk membuat perangkat lunak *Graphic User Interface (GUI)* untuk memvisualisasikan perambatan gelombang elektromagnetik dan menganalisis karakteristiknya sehingga para peneliti dapat melihat rambatan gelombang elektromagnetik secara *visual* melalui simulasi.

1.3 Ruang lingkup

Ruang lingkup dalam penelitian ini adalah simulasi gelombang elektromagnetik menggunakan metode FDTD dua dimensi dengan syarat batas komputasi menggunakan *Perfectly Matched Layer (PML)*.

1.4 Manfaat

Manfaat yang didapat dari tugas akhir ini adalah perangkat lunak yang *open source* dapat dikembangkan untuk pengembangan piranti optik, seperti: sensor, *waveguide*, maupun *optical filter*.

BAB II TINJAUAN PUSTAKA

2.1 Tinjauan Pustaka

2.1.1 Persamaan Maxwell

Radiasi gelombang EM dideskripsikan oleh vektor medan listrik dan medan magnetik. Propagasi dari kedua vektor medan tersebut ditentukan oleh persamaan Maxwell. Sejalan dengan Hukum Newton sebagai landasan hukum mekanika klasik, maka persamaan Maxwell merupakan perumusan hukum-hukum alam yang melandasi semua fenomena elektromagnetik. Dalam papernya “*A Dynamic Theory of Electromagnetic Field*” Maxwell mengungkapkan 4 persamaan yang mendasari semua gejala makroskopik listrik dan magnet (Tjia, 1994), yakni:

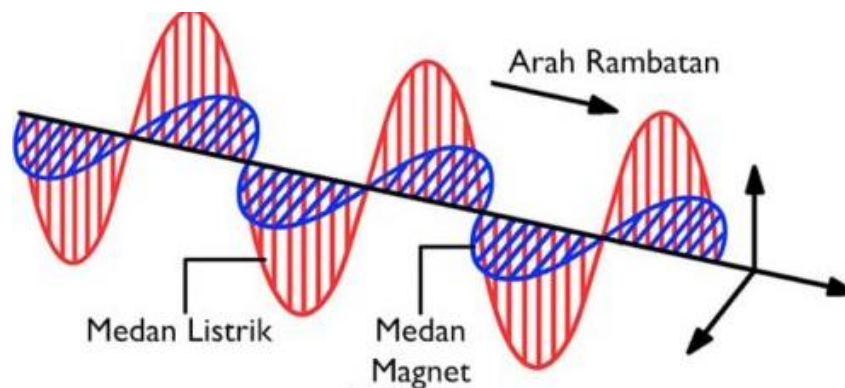
$$\nabla \times \mathbf{E} + \frac{\partial \mathbf{B}}{\partial t} = 0 \quad (1)$$

$$\nabla \times \mathbf{H} - \frac{\partial \mathbf{D}}{\partial t} = \mathbf{J} \quad (2)$$

$$\nabla \cdot \mathbf{D} = \rho \quad (3)$$

$$\nabla \cdot \mathbf{B} = 0 \quad (4)$$

Dimana \mathbf{E} dan \mathbf{H} adalah vektor medan makroskopik listrik dan magnet, \mathbf{B} dan \mathbf{D} adalah medan perpindahan listrik dan induksi magnet yang muncul sebagai respon bahan terhadap medan, ρ dan \mathbf{J} adalah rapat muatan listrik bebas dan rapat arus listrik bebas.

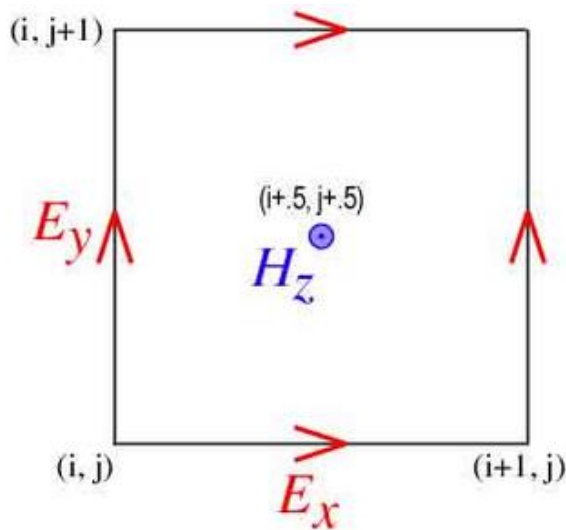


Gambar 1. Gelombang Elektromagnetik

Solusi dari empat persamaan elektromagnetik bisa didapatkan dengan cara analitik dengan mengaitkannya dengan persamaan konstitutif (Teguh, 2006), maupun dengan cara numerik menggunakan metode diskritisasi. Solusi dari persamaan Maxwell banyak dimanfaatkan dalam sains dan teknik, seperti: aplikasi medis (Daniel, 1985), radar (Teguh, 2019), pandu gelombang optik (Eugene, 2019), sensor (Teguh, 2020), antenna (Yahya, 2014), deteksi gas *hidrant* (Weitemeyer, 2011), dan lain-lain.

2.1.2 Metode FDTD

Finite Difference Time Domain (FDTD) adalah metode yang menggunakan pendekatan pada domain ruang dan waktu. Metode FDTD diperkenalkan pertama kali oleh Kane Yee pada tahun 1966 untuk menganalisis medan elektromagnet. Metode ini sekarang banyak diterapkan sebagai metode analisis yang utama. Salah satu alasan pemilihan metode ini adalah mudah untuk menganalisis permasalahan yang didasarkan pada persamaan integral yang sangat sulit dilakukan bila dipecahkan dengan *Moment Method* dan lain-lain. Selain itu untuk menggunakan metode ini tidak memerlukan dasar-dasar pengetahuan yang mendalam (Josaphat, 1998).



Gambar 2. Step FDTD Dua Dimensi

Untuk gelombang EM 2 dimensi, dikenal dua mode dari komponen medan listrik dan medan magnet, yaitu polarisasi *Transverse Electric (TE)* dan polarisasi *Magnetik (TM)*. Untuk medan magnet memiliki komponen pada arah sumbu-z, sedangkan medan listrik mempunyai komponen pada arah sumbu-x dan sumbu-y, maka disebut mode TE. Untuk medan listrik memiliki komponen pada arah sumbu-z, sedangkan medan magnet mempunyai komponen pada arah sumbu-x dan sumbu-y, maka disebut mode TM.

Persamaan Maxwell untuk mode TM dapat ditulis:

$$\frac{\partial D_z}{\partial t} = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \quad (5)$$

$$\frac{\partial H_x}{\partial t} = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \left(\frac{\partial D_z}{\partial y} \right) \quad (6)$$

$$\frac{\partial H_y}{\partial t} = \frac{1}{\sqrt{\epsilon_0 \mu_0}} \left(\frac{\partial D_z}{\partial x} \right) \quad (7)$$

sedangkan untuk mode TE

$$\frac{\partial H_z}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial E_x}{\partial y} - \frac{\partial H_y}{\partial x} \right) \quad (8)$$

$$\frac{\partial D_x}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial H_z}{\partial y} \right) \quad (9)$$

$$\frac{\partial D_y}{\partial t} = \frac{1}{\sqrt{\varepsilon_0 \mu_0}} \left(\frac{\partial H_z}{\partial x} \right) \quad (10)$$

dengan $D_x = \varepsilon_0 \varepsilon_r E_x$, dan $D_y = \varepsilon_0 \varepsilon_r E_y$ merupakan hubungan antara vektor medan dan perpindahan medan listrik dan magnet, ε_0, μ_0 adalah berturut-turut permitifitas listrik dan permeabilitas magnetik udara, ε_r adalah berturut-turut permitifitas relatif bahan. $\frac{1}{\sqrt{\varepsilon_0 \mu_0}}$ nilainya setara dengan nilai cepat rambat cahaya di ruang hampa, c_0

Pada metode FDTD, untuk memecahkan persoalan wilayah terbuka, misalnya analisa hamburan, wilayah analisa perlu dibatasi menggunakan wilayah batas khayal. Apabila wilayah batas khayal (kondisi batas serap) ini tidak sempurna, maka akan menyebabkan pantulan gelombang ke wilayah analisa yang tentunya dapat mempengaruhi nilai analisa. Pada kondisi batas serap tidak menggunakan algoritma Yee, karenanya diperlukan cara khusus untuk menurunkannya. Dalam perkembangannya banyak kondisi batas serap yang diusulkan, diantara yang diusulkan terdapat kondisi batas serap yang memiliki keakurasian tinggi yang dikenal dengan *perfectly matched layer* (PML). Untuk mode TM dengan menggunakan batas serap PML, dapat ditulis pada persamaan berikut:

$$j\omega \left(1 + \frac{\sigma_D(x)}{j\omega \varepsilon_0} \right) D_z = c_0 \left(\frac{\partial H_y}{\partial x} - \frac{\partial H_x}{\partial y} \right) \quad (11)$$

$$j\omega \left(1 + \frac{\sigma_D(x)}{j\omega \varepsilon_0} \right) H_y = c_0 \left(\frac{\partial E_z}{\partial x} \right) \quad (12)$$

$$j\omega \left(1 + \frac{\sigma_D(x)}{j\omega \varepsilon_0} \right)^{-1} H_x = -c_0 \left(\frac{\partial E_z}{\partial y} \right) \quad (13)$$

Dengan σ_D merupakan konduktivitas listrik medium. Diskritisasi persamaan (11) sampai (13) menggunakan metode *central difference*, menghasilkan persamaan berikut:

$$D_z^{n+\frac{1}{2}}(i, j) = g_{i3}(i) D_z^{n-\frac{1}{2}}(i, j) + g_{i2}(i) 0.5 \left(H_y^n \left(i + \frac{1}{2}, j \right) - H_y^n \left(i - \frac{1}{2}, j \right) - H_x^n \left(i, j + \frac{1}{2} \right) - H_x^n \left(i, j - \frac{1}{2} \right) \right) \quad (14)$$

$$H_y^{n+1}\left(i + \frac{1}{2}, j\right) = f_{i3}\left(i + \frac{1}{2}\right) H_y^n\left(i + \frac{1}{2}, j\right) + f_{i2}\left(i + \frac{1}{2}\right) 0.5 \left(E_z^{n+\frac{1}{2}}\left(i + 1, j\right) - E_z^{n+\frac{1}{2}}\left(i, j\right) \right) \quad (15)$$

$$I_{Hx}^{n+\frac{1}{2}}\left(i, j + \frac{1}{2}\right) = I_{Hx}^{n-\frac{1}{2}}\left(i, j + \frac{1}{2}\right) + \left(E_z^{n+\frac{1}{2}}\left(i, j\right) - E_z^{n+\frac{1}{2}}\left(i, j + 1\right) \right) \quad (16)$$

$$H_x^{n+1}\left(i, j + \frac{1}{2}\right) = H_x^n\left(i, j + \frac{1}{2}\right) + 0.5 \left(E_z^{n+\frac{1}{2}}\left(i, j\right) - E_z^{n+\frac{1}{2}}\left(i, j + 1\right) \right) + f_{i1}(i) I_{Hx}^{n+\frac{1}{2}}\left(i, j + \frac{1}{2}\right) \quad (17)$$

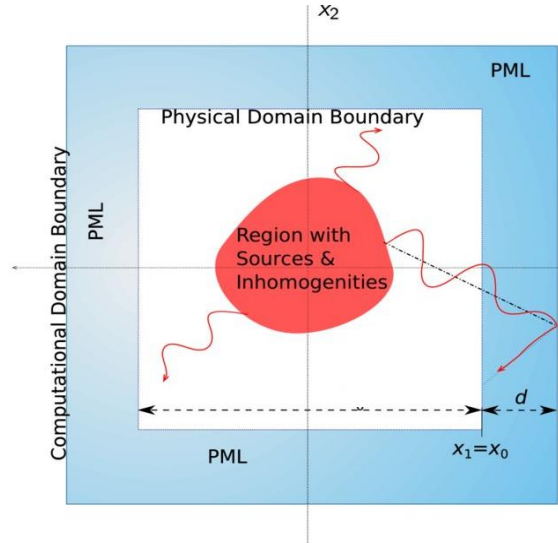
dimana $g_{i2}(i) = \frac{1}{1 + \frac{\sigma_D(i)\Delta t}{2c_0}}$, $g_{i3}(i) = \frac{1 - \frac{\sigma_D(i)\Delta t}{2c_0}}{1 + \frac{\sigma_D(i)\Delta t}{2c_0}}$, $f_{i2}\left(i + \frac{1}{2}\right) = \frac{1}{1 + \frac{\sigma_D\left(i + \frac{1}{2}\right)\Delta t}{2c_0}}$, dan

$$f_{i3}\left(i + \frac{1}{2}\right) = \frac{1 - \frac{\sigma_D\left(i + \frac{1}{2}\right)\Delta t}{2c_0}}{1 + \frac{\sigma_D\left(i + \frac{1}{2}\right)\Delta t}{2c_0}}$$

Jika gelombang bergerak ke arah sebuah batas dalam ruang terbuka dan gelombang merambat pada c (kecepatan cahaya), maka dalam satu *time step* gelombang merambat sepanjang $\frac{\Delta x}{2}$. Nilai ini menjelaskan bahwa gelombang membutuhkan dua *time steps* untuk melewati satu *cell*. Dengan demikian dapat disimpulkan bahwa syarat batas dapat diterima jika :

$$E_z|_{i=0}^n = E_x|_{i=1}^n \quad (18)$$

Dari persamaan di atas dapat dipahami bahwa hanya dengan menyimpan nilai $E_x(1)$ untuk dua *time steps* dan kemudian masukkan nilainya ke dalam $E_x(0)$, maka syarat batas telah terpenuhi.



Gambar 3. Domain Komputasi Dengan PML

Untuk mensimulasikan gelombang datar dalam program FDTD 2D, ruang permasalahan dibagi menjadi dua bagian yaitu: medan total (*total field*) dan medan hambur (*scattered field*) sebagaimana ada gambar 4. Medan listrik yang masuk dapat digunakan untuk menganalisa wilayah analisa, tetapi perhitungan medan hambur perlu dilakukan dengan penurunan persamaan dengan syarat batas serap. Pada umumnya medan hambur lebih kecil dibanding gelombang masuk. Walaupun pantulan pada dinding syarat batas serap terjadi dalam wilayah analisa, pantulan ini harus lebih kecil daripada gelombang masuk. Keseluruhan medan listrik-magnet dapat dibagi menjadi 3:

- Nilai D_z pada titik j_a dan j_b

$$D_z(i, j_a) = D_z(i, j_a) + 0.5H_{x_inc} \left(j_a - \frac{1}{2} \right) \quad (19)$$

$$D_z(i, j_b) = D_z(i, j_b) + 0.5H_{x_inc} \left(j_b + \frac{1}{2} \right) \quad (20)$$

- Nilai H_x diluar titik j_a dan j_b

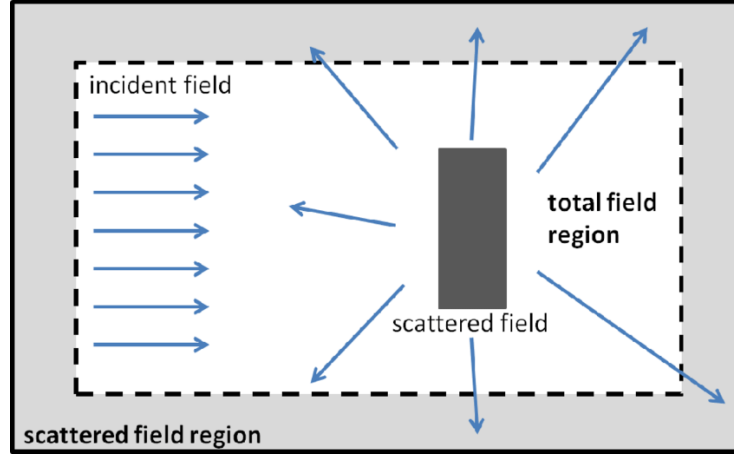
$$H_x \left(i, j_a - \frac{1}{2} \right) = H_x \left(i, j_a - \frac{1}{2} \right) + 0.5E_{z_inc} (j_a) \quad (21)$$

$$H_x \left(i, j_b + \frac{1}{2} \right) = H_x \left(i, j_b + \frac{1}{2} \right) + 0.5E_{z_inc} (j_b) \quad (22)$$

- Nilai H_y diluar titik i_a dan i_b

$$H_x\left(i_a - \frac{1}{2}, j\right) = H_x\left(i_a - \frac{1}{2}, j\right) + 0.5E_{z_inc}(j) \quad (23)$$

$$H_x\left(i_b + \frac{1}{2}, j\right) = H_x\left(i_b + \frac{1}{2}, j\right) + 0.5E_{z_inc}(j) \quad (24)$$



Gambar 4. Wilayah TF/SF Pada Domain Komputasi

Sumber gelombang input merupakan persamaan gelombang dalam domain waktu yang dapat ditulis sebagai persamaan gelombang berjalan (sinusoidal), maupun paket gelombang (gaussian). Persamaan gelombang sinusoidal dan gaussian dapat ditulis:

$$E(t) = E_0 \sin(\omega t) \quad (25)$$

$$E(t) = E_0 e^{-4\pi \left(\frac{t-t_0}{\tau}\right)^2} \quad (26)$$

dimana adalah ω_0 frekuensi sudut gelombang, adalah waktu gelombang, adalah waktu awal gelombang, adalah τ konstanta penyebaran gelombang.

2.1.3 GUI Matlab

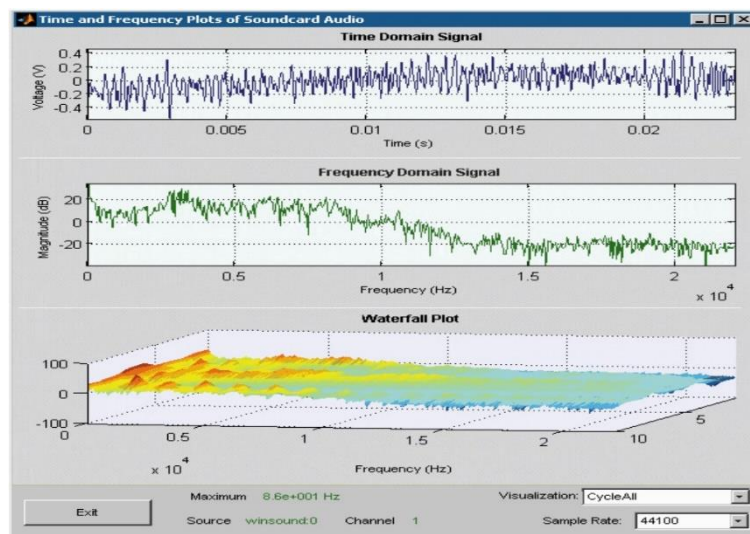
Metode komputasi yang umum digunakan untuk simulasi gelombang elektromagnetik adalah Matlab, dikarenakan bahasa Matlab adalah bahasa matriks yang merupakan formulasi dalam pemecahan secara numerik persamaan Maxwell. Matlab adalah suatu bahasa pemrograman komputer yang digunakan untuk memecahkan persoalan-persoalan (khususnya bidang teknik), seperti halnya bahasa-bahasa program lainnya yang sudah dikenal, misalnya Fortran, Basic, Pascal, C language dan lain-lain. Jadi Matlab adalah salah satu *High level language* yang dikembangkan setelah bahasa-bahasa tersebut diatas. *Syntax* Matlab mirip dengan *Basic* dan *C*, namun memiliki kemampuan/kelebihan dari bahasa-bahasa yang

disebutkan terakhir, diantaranya data yang berupa bilangan biasa, atau bilangan kompleks atau bahkan suatu matrix (baik riil maupun kompleks), pengoperasiannya sudah disiapkan operasinya built-in didalam Matlab itu sendiri. Jadi kita tidak perlu lagi mendefinisikan dan membuat definisi-definisi / rutin-rutin untuk operasi terhadap data-data tersebut. Dengan demikian pembuatan program dalam Matlab akan jauh lebih mudah dan lebih singkat dibanding dengan bahasa-bahasa program lainnya (Maula, 2001).

Program simulasi gelombang elektromagnetik dapat dibuat dengan *GUIDE* yang memuat untuk seluruh model menggunakan metode FDTD. *GUIDE* atau *GUI builder* merupakan sebuah *graphical user interface (GUI)* yang dibangun dengan obyek grafik seperti tombol (*button*), kotak *text*, *slider*, *menu* dan lain-lain. Aplikasi yang menggunakan GUI umumnya lebih mudah dipelajari dan digunakan karena orang yang menjalankannya tidak perlu mengetahui perintah yang ada dan bagaimana kerjanya. *GUIDE* Matlab mempunyai kelebihan tersendiri dibandingkan dengan bahasa pemrogram lainnya, diantaranya:

- GUI Matlab mempunyai fungsi *built-in* yang siap digunakan dan pemakai tidak perlu repot membuatnya sendiri,
- Ukuran file, baik FIG-file maupun M-file, yang dihasilkan relatif kecil,
- Kemampuan grafisnya cukup andal dan tidak kalah dibandingkan dengan bahasa pemrograman lainnya.

Agar GUI Matlab dapat lebih mudah digunakan, Matlab menyediakan *compiler* yang memungkinkan *user* menggunakan GUI diluar matlab.



Gambar 5. Aplikasi Analisis Spektrum dan *running* pada Windows di luar Matlab

2.2 Penelitian Terdahulu

Penelitian terdahulu yang membahas tentang simulasi gelombang menggunakan metode numerik adalah sebagai berikut :

1. Azhar KA. Pada tahun 2000 melakukan penelitian yang ditulis pada Jurnal Penelitian Sains dengan judul penelitian: “Penerapan Metode Beda Hingga untuk Penyelesaian Persamaan Gelombang Kawasan Waktu”.
2. Hadi TY, Hermawan KD, Andian BS. Pada tahun 2000 melakukan penelitian yang ditulis pada Jurnal Penelitian Sains dengan judul penelitian: *Journal of*

Science and Applicative Technology dengan judul penelitian: “Penggunaan Metode *Finite Difference Time Domain (FDTD)* Dalam Simulasi *Phased Array Antenna*”.

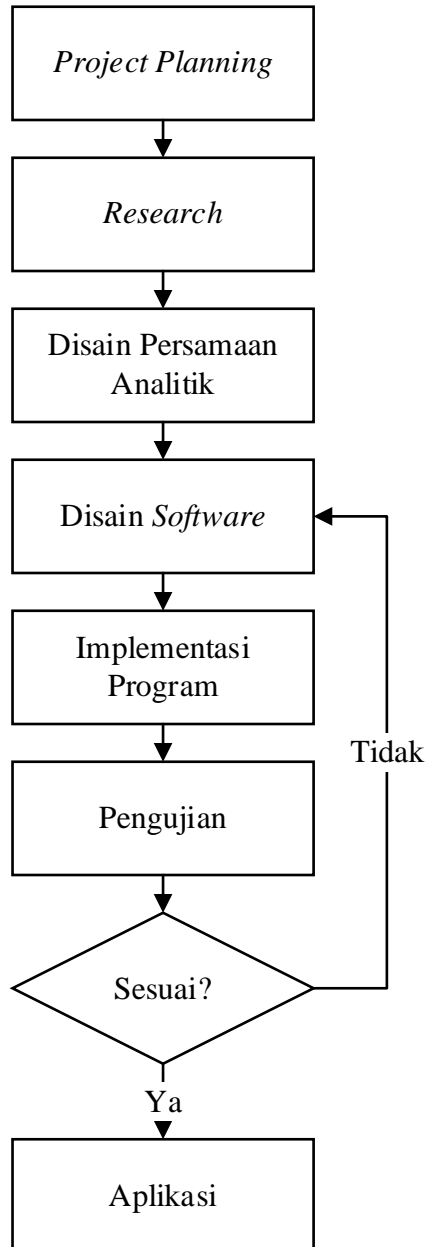
2.3 Tabel Perbandingan Penelitian

Tabel 1. Perbandingan Penelitian

Nama	Objek Penelitian	Metode yang digunakan	Instrumen yang digunakan	Display (Tampilan)
Azhar KA	Gelombang Elektromagnetik	FDTD	Software Matlab	MFile
Hadi TY, Hermawan KD, Andian BS	Penerapan Gelombang Elektromagnetik	FDTD	Software Matlab	MFile
Eki Prasetya	Gelombang Elektromagnetik	FDTD	Software Matlab	GUI

BAB III METODE PENELITIAN

3.1 Tahapan Penelitian



Gambar 6. Diagram Alir Penelitian

Tahap pertama dalam penelitian ini adalah *projet planning* yang mencakup: penentuan tema dan judul, ruang lingkup, tujuan dan manfaat penelitian. Dilanjutkan tahap kedua, yaitu *research* yang merupakan kegiatan mencari, mengumpulkan dan menganalisis studi penelitian terkait dalam bentuk jurnal, buku, maupun skripsi sebelumnya. Tahap ketiga adalah pembentukan persamaan numerik dari persamaan analitik FDTD yang terdapat pada literatur. Tahap kelima merupakan perancangan algoritma dan penerapannya pada bahasa pemrograman Matlab. Tahap berikutnya

adalah pengujian, yaitu *running program* untuk melihat hasil yang diharapkan. Jika hasilnya kurang sesuai, dipastikan ada kesalahan pada algoritma.

3.1.1 Perencanaan Proyek Penelitian (*Project Planning*)

Tahap pertama dalam penelitian ini adalah *project planning* yang mencakup: penentuan tema dan judul, ruang lingkup, tujuan dan manfaat penelitian, membuat kerangka awal penelitian, estimasi kebutuhan alat dan bahan, estimasi anggaran, dan kemungkinan penerapan dari aplikasi yang akan dirancang.

3.1.2 Penelitian (*Reaserch*)

kegiatan mencari, mengumpulkan dan menganalisis studi penelitian terkait dalam bentuk jurnal, buku, maupun skripsi sebelumnya.

3.1.3 Disain persamaan Analitik

Persamaan analitik disusun secara terstruktur dari persamaan Maxwell dengan dilengkapi kondisi syarat batas, agar gelombang tidak mengalami pantulan akibat domain komputasi.

3.1.3.1 Parameter *Input*

Input dalam simulasi gelombang EM dapat dilihat pada tabel berikut:

Tabel 2. Parameter dan Simbol Pada Code FDTD di Matlab

No.	Parameter	Simbol	Keterangan
1	Parameter gelombang	f	Frekuensi gelombang
2	Parameter <i>grid</i>	ddx	Resolusi size
3	Parameter PML dan TF/SF	npml	Kondisi batas serap
		ia	Batas kiri sumbu x untuk gelombang total dan gelombang yang di hamburkan
		ib	Batas kanan sumbu x untuk gelombang total dan gelombang yang di hamburkan
		ja	Batas bawah umbu y untuk gelombang total dan gelombang yang di hamburkan
		jb	Batas atas sumbu y untuk gelombang total dan gelombang yang di hamburkan
4	Parameter bahan	e0	Permitivitas listrik di udara
		e	Permeabilitas relatif bahan
		Sigma	Konduktivitas listrik

5	Parameter domain komputasi	KE	Batas komputasi sumbu y
		IE	Batas komputasi sumbu x
6	Parameter waktu	NSTEPS	Waktu penjalaran gelombang
		dt	Resolusi waktu

3.1.3.2 Initial Condition

Initial condition merupakan kondisi awal yang diberikan kepada medan listrik maupun medan magnet sebelum bergerak.

Tabel 3. Simbol Medan Pada FDTD di Matlab

No.	Parameter	Simbol	Keterangan
1	TM	ez	Medan listrik arah sumbu z
		dz	Vektor medan listrik arah sumbu z
		hx	Medan magnet arah sumbu x
		hy	Medan magnet arah sumbu y
2	TE	hz	Medan magnet arah sumbu z
		bz	Vektor medan magnet arah sumbu z
		ex	Medan listrik arah sumbu x
		ey	Medan listrik arah sumbu y

3.1.3.3 Persamaan Syarat Batas

Persamaan syarat batas menggunakan *Perfectly Matched Layer* (PML) yang menunjukkan kontinuitas medan diluar wilayah komputasi, yaitu pada persamaan (27).

3.1.3.4 Persamaan Medan

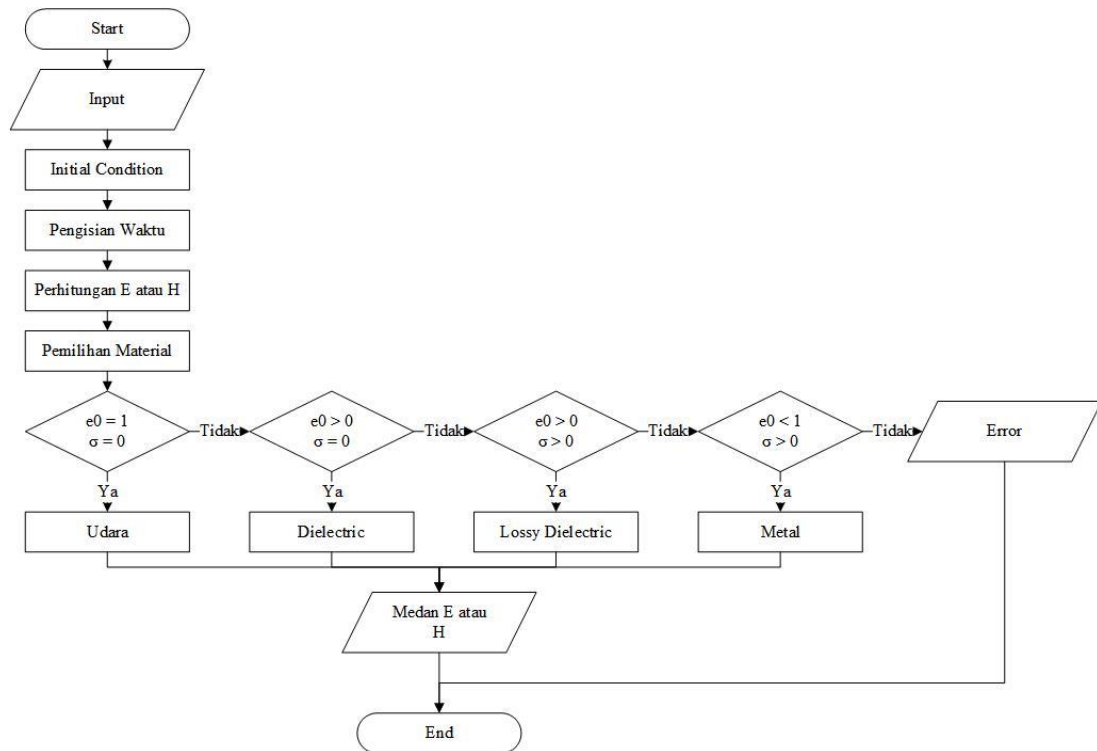
Medan listrik dan medan magnet untuk kasus TM dan dilengkapi dengan syarat batas PML dan wilayah TF/SF telah ditunjukkan melalui persamaan (14)-(17).

3.1.3.5 Persamaan Output

Output dari simulasi gelombang EM adalah medan listrik dan medan magnet dalam domain ruang dan waktu $E_z(i, j, t)$.

3.1.4 Disain Software

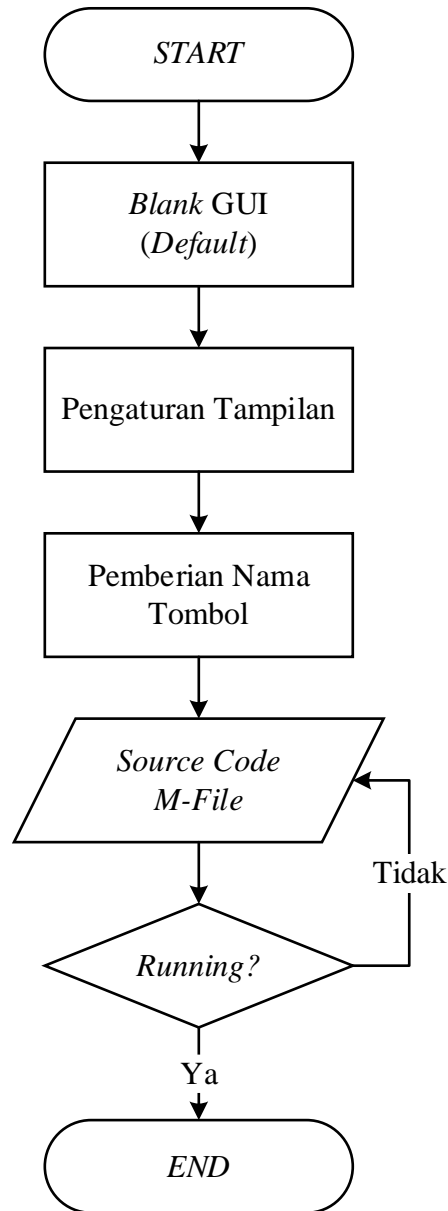
Disain software dirancang untuk menuliskan *listing* program dalam Matlab yang terbagi dalam dua tahap, yaitu program utama dan program *interface*.



Gambar 7. Diagram Alir Program Utama

Listing program utama disusun berdasarkan gambar 7. Input tersusun atas domain komputasi, parameter PML, dan parameter gelombang. ketika waktu diisi, maka perhitungan gelombang dimulai dan merambat dalam domain komputasi. Gelombang mengenai struktur sesuai dengan parameter material yang diberikan. Karakteristik perambatan gelombang dapat langsung di jalankan dan ditampilkan melalui M-File.

Pada perancangan GUI, pada pemrograman Matlab telah disediakan fasilitas menu GUI Matlab yang memungkinkan user untuk membuat sebuah *interface* untuk perhitungan. Pada dasarnya, membangun sebuah interface GUI Matlab dapat dilakukan dengan tiga tahap: pembentukan tampilan, pengaturan menu, dan pemanggilan program utama. Langkah kerja tersebut dapat disusun berdasarkan diagram alir pada gambar 8.



Gambar 8. Diagram Alir Sistem GUI

3.1.5 Implementasi Program

Algoritma yang telah disusun dapat dijalankan dalam dua cara, yaitu *running* program dalam program utama dan *running* melalui *interface* GUI.

3.1.6 Pengujian

Pada tahap ini dilakukan pengujian perangkat lunak yang telah dirancang, yang meliputi:

- Uji Fungsional

Uji coba fungsional dilakukan dengan cara menguji setiap button, fungsi atau form yang dibuat, apakah sudah berfungsi dengan baik atau tidak,

- Uji Validasi

Validasi data dilakukan dengan cara menguji semua hasil data sesuai dengan literature atau penelitian sebelumnya,

3.1.7 Aplikasi

Setelah pengujian dilakukan, program dapat digunakan untuk mensimulasikan gelombang terhadap objek rod dengan bahan yang dapat divariasikan.

3.2 Alat Dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah Komputer *Intel Core i5* (16GB) dan *software* Matlab R2019b (9.0.7).

BAB IV PERANCANGAN DAN IMPLEMENTASI

4.1 Perancangan

Pada tahapan ini dibuat rancangan tampilan aplikasi FDTD. Program terdiri atas dua bagian yaitu source code FDTD pada M-File Matlab dan *source code* GUI untuk menampilkan simulasi gelombang.

4.1.1 Code Interface GUI

Interface GUI dibuat menggunakan fitur GUIDE yang langsung terhubung dengan M-File, GUI terdiri dari:

4.1.1.1 Parameter *Input*

Tampilan parameter input dapat dituliskan sebagai berikut:

```
function input_ddx_Callback(hObject, eventdata, handles)
function input_ddx_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function input_npml_Callback(hObject, eventdata, handles)
function input_npml_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function input_ia_Callback(hObject, eventdata, handles)
function input_ia_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function input_ib_Callback(hObject, eventdata, handles)
function input_ib_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function input_ja_Callback(hObject, eventdata, handles)
function input_ja_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function input_jb_Callback(hObject, eventdata, handles)
function input_jb_CreateFcn(hObject, eventdata, handles)
```

```

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

function input_epsz_Callback(hObject, eventdata, handles)
function input_epsz_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_e0_Callback(hObject, eventdata, handles)
function input_e0_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_sigma_Callback(hObject, eventdata, handles)
function input_sigma_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_KE_Callback(hObject, eventdata, handles)
function input_KE_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_IE_Callback(hObject, eventdata, handles)
function input_IE_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_steps_Callback(hObject, eventdata, handles)
function input_steps_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_dt_Callback(hObject, eventdata, handles)
function input_dt_CreateFcn(hObject, eventdata, handles)
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

4.1.1.2 Tombol Start

Tampilan tombol *start* dapat dituliskan sebagai berikut:

```
function pushbutton1_Callback(hObject, eventdata, handles)
```

4.1.1.3 Axes

Tampilan *axes* dapat dituliskan sebagai berikut:

```
axes(handles.axes1);
cla;

t = 1:100;
v_ddx = str2num(get(handles.input_ddx, 'String'));
v_npml = str2num(get(handles.input_npml, 'String'));
v_ia = str2num(get(handles.input_ia, 'String'));
v_ib = str2num(get(handles.input_ib, 'String'));
v_ja = str2num(get(handles.input_ja, 'String'));
v_jb = str2num(get(handles.input_jb, 'String'));
v_epsz = str2num(get(handles.input_epsz, 'String'));
v_e0 = str2num(get(handles.input_e0, 'String'));
v_sigma = str2num(get(handles.input_sigma, 'String'));
v_KE = str2num(get(handles.input_KE, 'String'));
v_IE = str2num(get(handles.input_IE, 'String'));
v_step = str2num(get(handles.input_steps, 'String'));
v_dt = str2num(get(handles.input_dt, 'String'));

clc
```

4.1.1.4 Toolbar

Menu toolbar seperti *save*, *zoom in*, *zoom out*, *pan*, dan *3d rotate* merupakan menu bawaan GUIDE, akan tetapi untuk tombol *save* dibutuhkan tambahan code seperti berikut:

```
function savebutton_ClickedCallback(hObject, eventdata, handles)
[file,path] = uiputfile('*.jpg');
filename = fullfile(path,file);
saveas(gelelektro,filename);
```

4.1.2 Code Program Utama

Program utama merupakan keseluruhan program FDTD, yang ditulis pada M-File Matlab. Untuk kasus TE, program terdiri atas:

4.1.2.1 Parameter Input

Dalam bahasa Matlab, parameter input dapat dituliskan sebagai berikut:

```
IE      = 300;           % Jumlah grid sumbu_x.
JE      = 300;           % Jumlah grid sumbu_y
ia      = 7;             % Batas awalsumbu_x untuk TF/SF
ib      = IE-ia-1;      % Batas akhir sumbu_x untuk TF/SF
ja      = 7;             % Batas awalsumbu_y untuk TF/SF
jb      = JE-ja-1;      % Batas awalsumbu_y untuk TF/SF
ddx     = 0.01;          % Ukuran grid
dt      = ddx/6e8;      % Step waktu
epsz    = 8.8e-12;      % Permittivitas listrik udara
T       = 0;             % Waktu awal
```

```

nsteps = 800; % Waktu total

fil = zeros(IE,1); % Parameter PML
fi2 = ones(IE,1); % Parameter PML
fi3 = ones(IE,1); % Parameter PML
gi2 = ones(IE,1); % Parameter PML
gi3 = ones(IE,1); % Parameter PML
fj1 = zeros(IE,1); % Parameter PML
fj2 = ones(IE,1); % Parameter PML
fj3 = ones(IE,1); % Parameter PML
gj2 = ones(IE,1); % Parameter PML
gj3 = ones(IE,1); % Parameter PML
ihx = zeros(IE,JE); % Parameter PML
ihy = zeros(IE,JE); % Parameter PML
iz = zeros(IE,JE); % Parameter PML

```

4.1.2.2 Initial Condition

Initial condition untuk medan listrik dan medan magnet adalah sebagai berikut:

```

ez_low_m2 = 0; % Medan megnet pada PML
ez_low_m1 = 0; % Medan megnet pada PML
ez_high_m2 = 0; % Medan megnet pada PML
ez_high_m1 = 0; % Medan megnet pada PML
ez_inc = zeros(IE+1,1); % Vektor medan listrik awal
hx_inc = zeros(IE,1); % Vektor medan magnet awal
dz = zeros(IE,JE); % Perpindahan medan listrik
hy = zeros(IE,JE); % Vektor medan magnet
hx = zeros(IE,JE); % Vektor medan magnet

```

4.1.2.3 Persamaan Syarat Batas

Persamaan syarat batas menggunakan *Perectly Matched Layer* (PML) yang dapat ditulis dalam bahasa Matlab sebagai berikut:

```

ez_inc(1) = ez_low_m2;
ez_low_m2 = ez_low_m1;
ez_low_m1 = ez_inc(2);
ez_inc(JE) = ez_high_m2;
ez_high_m2 = ez_high_m1;
ez_high_m1 = ez_inc(JE-1);

```

4.1.2.4 Persamaan Medan

Medan listrik dan medan magnet yang telah dirumuskan ditulis dalam code Matlab adalah sebagai berikut:

Medan awal

```

for j = 2:JE
    ez_inc(j) = ez_inc(j) + 0.5*(hx_inc(j-1)-hx_inc(j));
end

for j = 1:JE
    hx_inc(j) = hx_inc(j) + 0.5*(ez_inc(j)-ez_inc(j+1));
end

```

Perambatan medan

```
    for j =2:IE;
        for i =2:IE;
dz(i,j)=gi3(i)*gj3(j)*dz(i,j) + gi2(i)*gj2(j)*.5*(hy(i,j)- hy(i-1,j)-
hx(i,j)+hx(i,j-1));
        end
    end

for j = 1:JE-1
    for i = 1:IE-1
        de = ez(i+1,j) - ez(i,j);
        ihy(i,j) = ihy(i,j) + fj1(j)*de;
        hy(i,j) = fi3(i)*hy(i,j) + fi2(i)*.5*(de + ihy(i,j));
    end
end

for j = 1:JE-1
    for i = 1:IE
        de = ez(i,j) - ez(i,j+1);
        ihx(i,j) = ihx(i,j) + fi1(i)*de;
        hx(i,j) = fj3(j)*hx(i,j) + fj2(j)*.5*(de + ihx(i,j));
    end
end
```

Medan pada TF/SF

```
    for i = ia:ib;
        dz(i,ja)=dz(i,ja)+0.5*hx_inc(ja-1);
        dz(i,jb)=dz(i,jb)-0.5*hx_inc(jb);
    end

    for j =ja:jb
        hy(ia-1,j)=hy(ia-1,j)-.5*ez_inc(j);
        hy(ib,j)=hy(ib,j)+.5*ez_inc(j);
    end

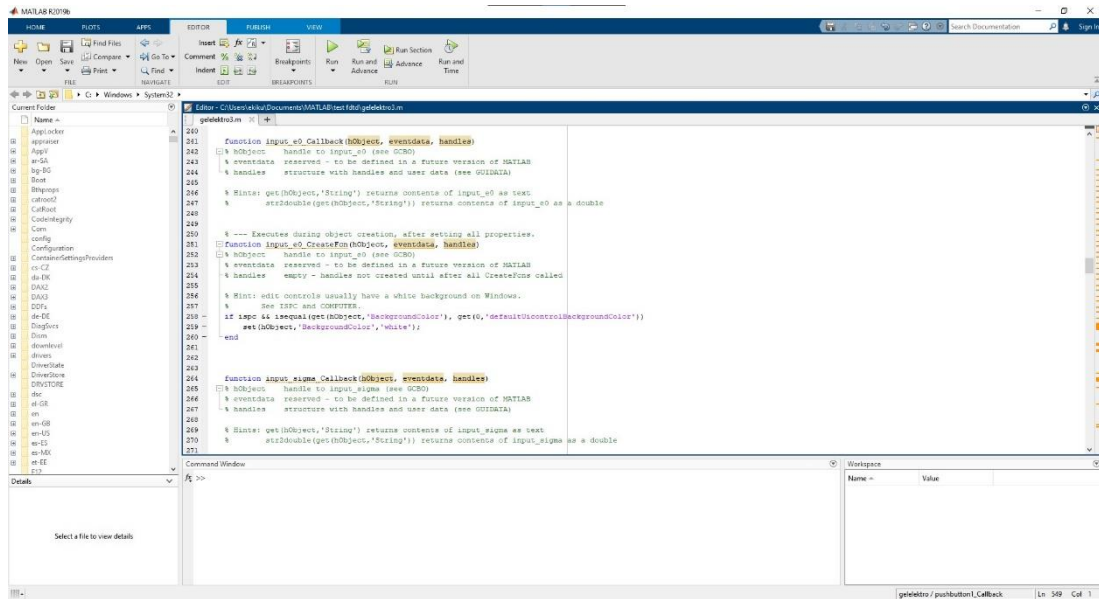
for i =ia:ib
        hx(i,ja-1)=hx(i,ja-1)+.5*ez_inc(ja);
        hx(i,jb)=hx(i,jb)-.5*ez_inc(jb);
    end
```

4.1.2.5 Persamaan *Output*

Output dari simulasi gelombang EM pada kasus TM dapat berupa medan listrik arah sumbu x atau sumbu y (H_x, H_y), maupun medan magnet arah sumbu z (E_z), yang dapat ditulis pada Matlab:

```
for j=1:JE
    for i=1:IE
        ez(i,j)=ga(i,j)*dz(i,j);
    end
end

figure(1)
```

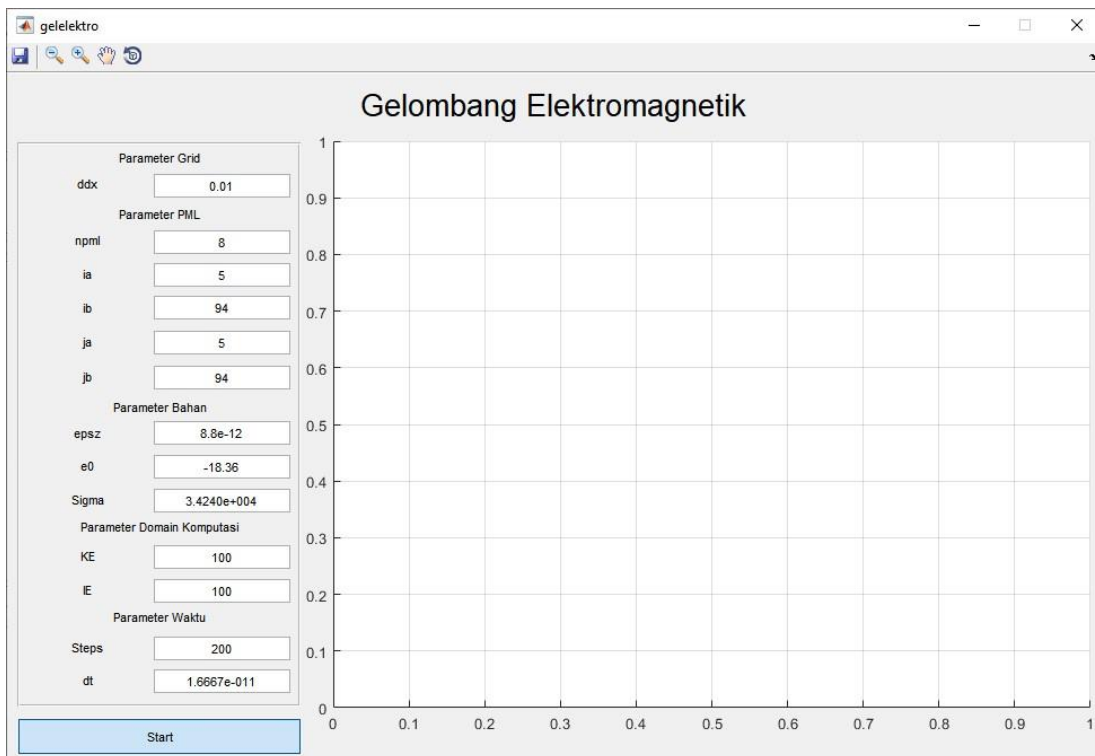



Gambar 10. Code Pembuatan GUI

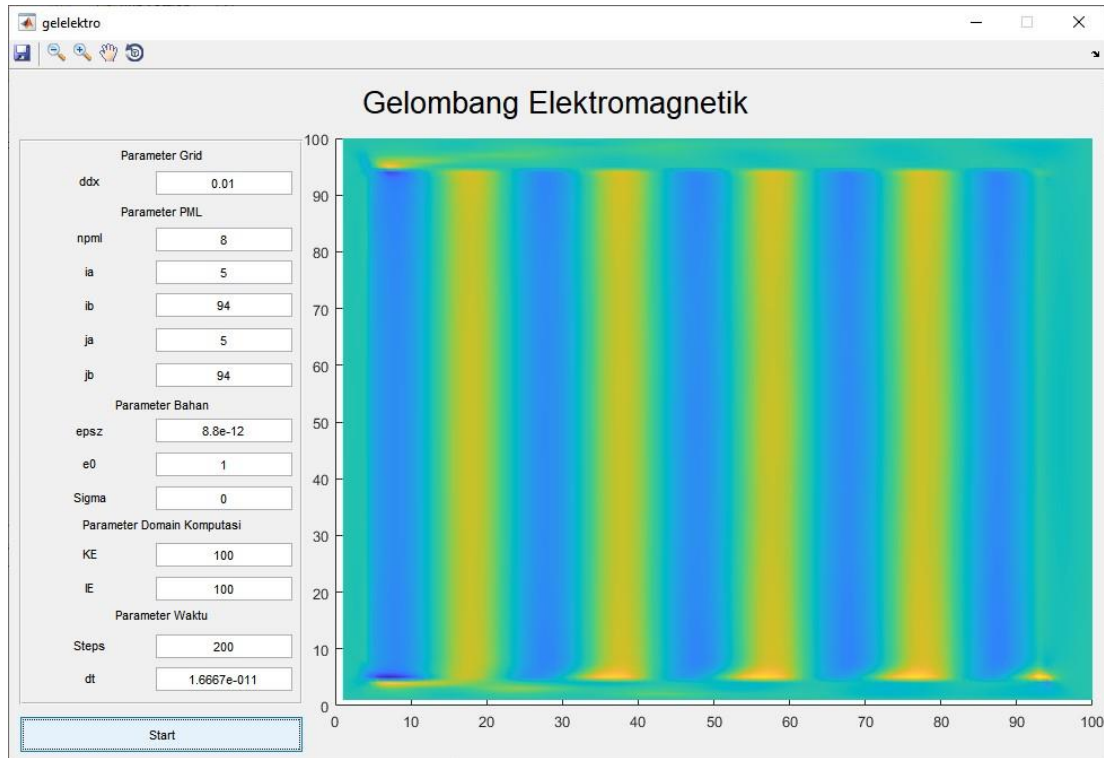
BAB V HASIL DAN PEMBAHASAN

5.1 Hasil

Hasil akhir dari penelitian ini adalah sebuah *interface* yang dapat mensimulasikan perambatan medan listrik atau medan magnet dalam ruang kosong (udara) maupun ketika berinteraksi dengan struktur. Berikut adalah hasil dari tampilan aplikasi, pada bagian input terdapat parameter *default* sebagai titik awal parameter, pada bagian *output* terdapat kotak kosong yang setelah dijalankan akan membentuk gelombang rambat sesuai dengan parameter yang di *inputkan*.



Gambar 11. Tampilan Aplikasi Sebelum Dijalankan



Gambar 12. Tampilan Aplikasi Seteah Dijalankan

5.2 Pengujian

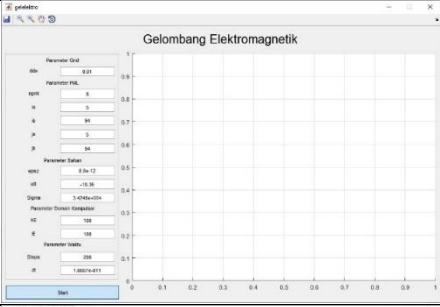
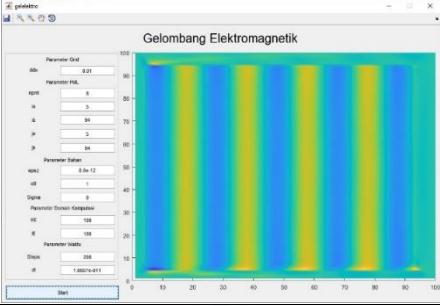
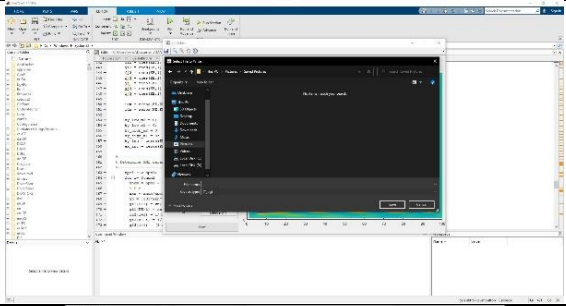
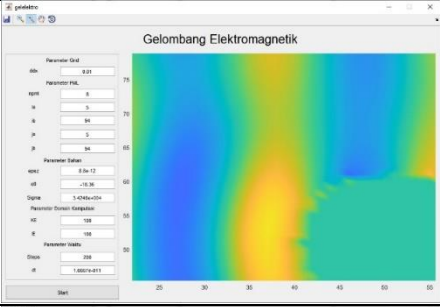
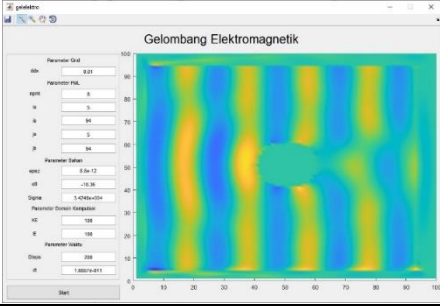
Hasil dari program aplikasi GUI dapat di uji kebenarannya melalui serangkaian uji coba. Interpretasi grafik juga dapat di analisis sesuai dengan teori gelombang dan penelitian sebelumnya.

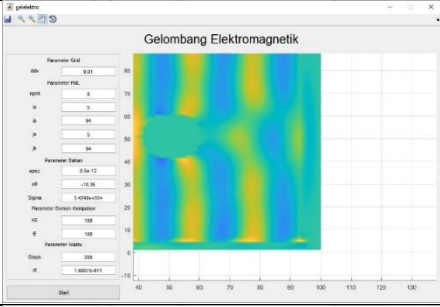
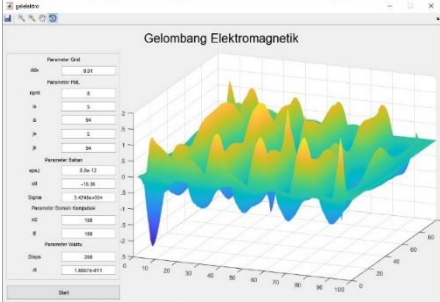
5.2.1 Uji Coba Fungsional

Uji coba fungsional dikakukan untuk mengetahui apakah sistem yang dibuat sudah berfungsi dengan baik, pada tahap ini dilakukan percobaan untuk mengetahui apakah fungsi dari setiap tombol, *edit text* dan yang lainnya dapat berfungsi dengan baik.

Tabel 4. Uji Coba Fungsional

No.	Fungsi	Tampilan	Keterangan
1	<i>Running program</i>		Berhasil

2	<i>Edit input</i>		Berhasil
3	Tombol <i>start</i>		Berhasil
4	Simpan file <i>output</i>		Berhasil
5	Tombol <i>Zoom In</i>		Berhasil
6	Tombol <i>Zoom Out</i>		Berhasil

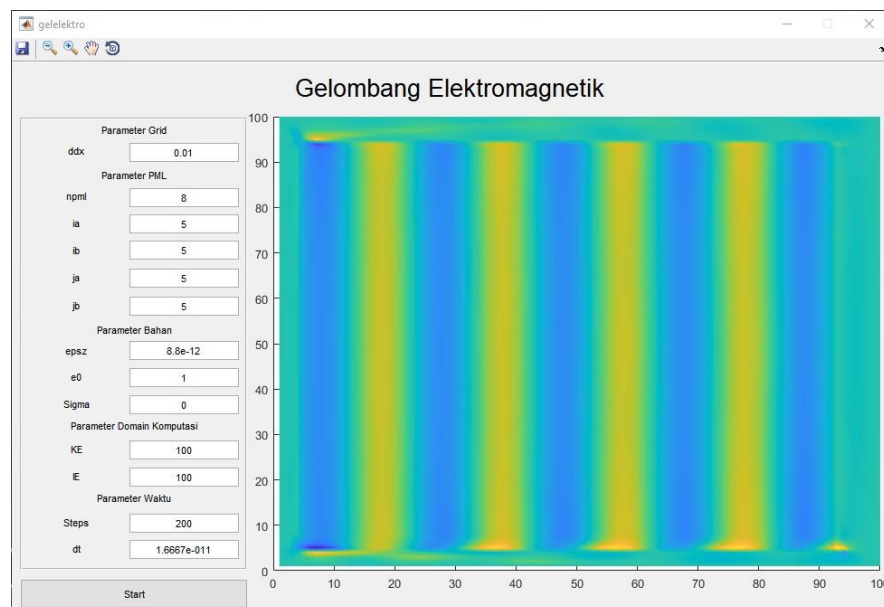
7	Tombol <i>Pan</i>		Berhasil
8	Tombol <i>Rotate 3D</i>		Berhasil

Fungsi tombol seperti *zoom in*, *zoom out*, *pan*, dan *3d rotate* baru dapat digunakan setelah gelombang selesai disimulasikan. Hasil simulasi gelombang berupa gambar dapat disimpan pada computer dengan cara klik tombol *save* pada aplikasi lalu pilih lokasi dan beri nama pada file.

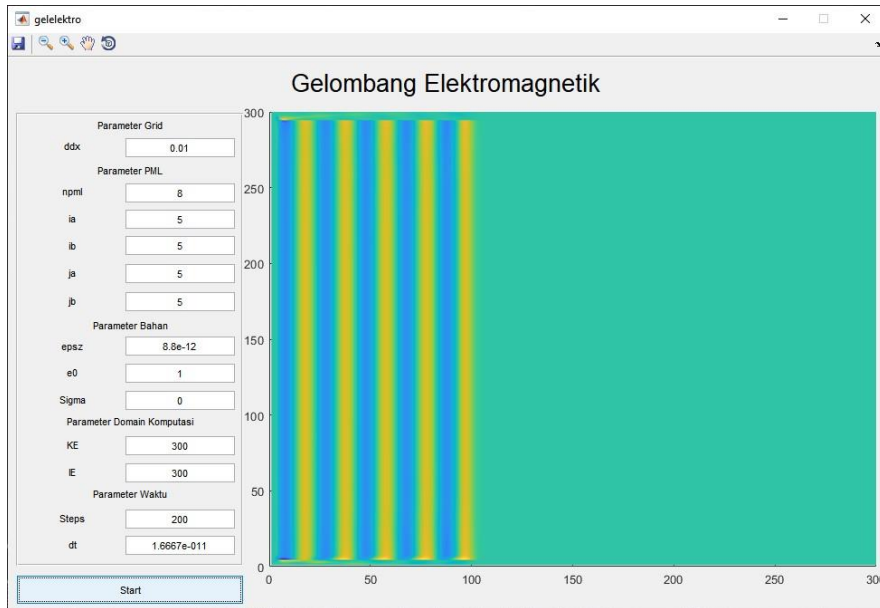
5.2.2 Uji Coba Validasi

Uji coba validasi adalah tahap dimana pengecekan apakah aplikasi sudah sesuai yang diharapkan atau tidak. Dibawah ini merupakan uji coba aplikasi pada beberapa parameter.

5.2.2.1 Uji Parameter Domain Komputasi

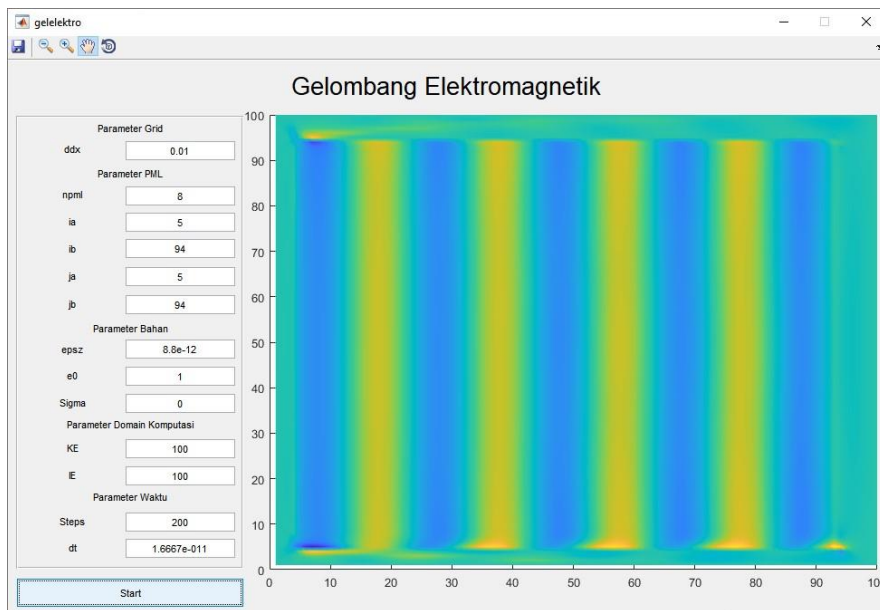


Gambar 13. Parameter Domain Komputasi Dengan Nilai 100

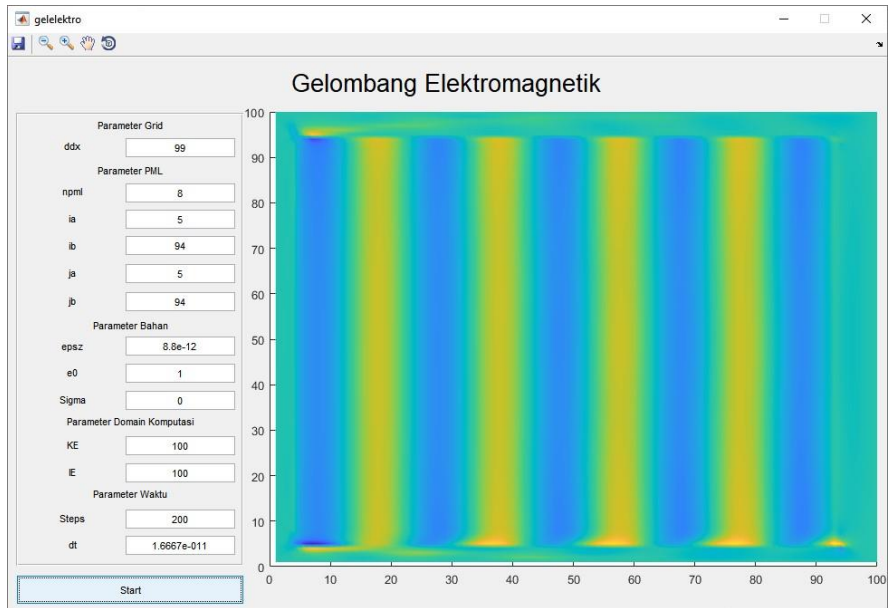


Gambar 14. Parameter Domain Komputasi Dengan Nilai 300

5.2.2.2 Uji Parameter *Grid*

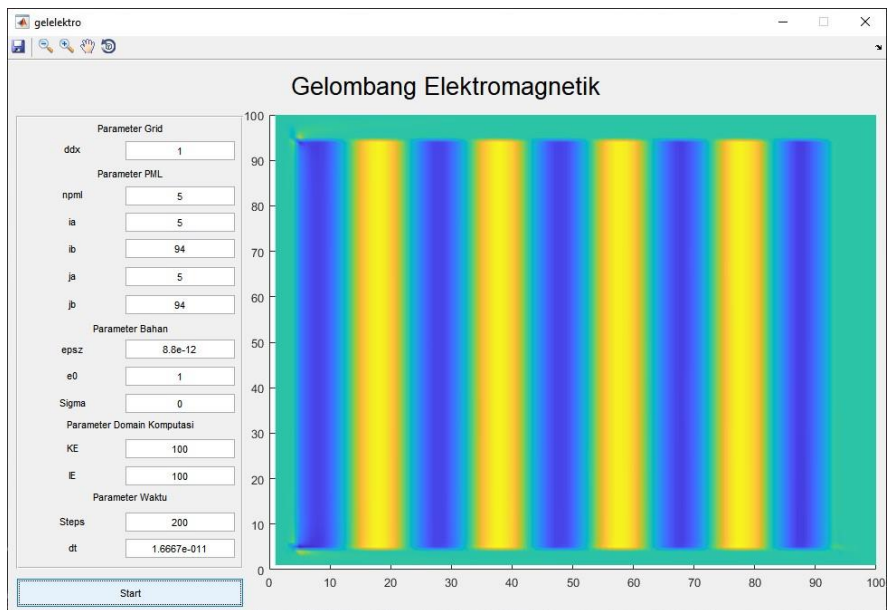


Gambar 15. Parameter *Grid* dengan Nilai 0.01

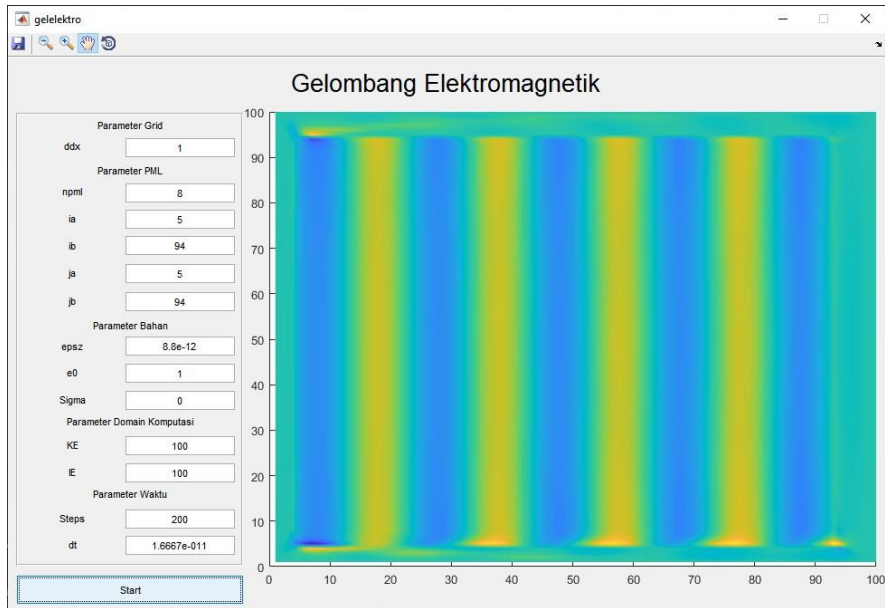


Gambar 16. Parameter *Grid* Dengan Nilai 99

5.2.2.3 Uji Parameter PML

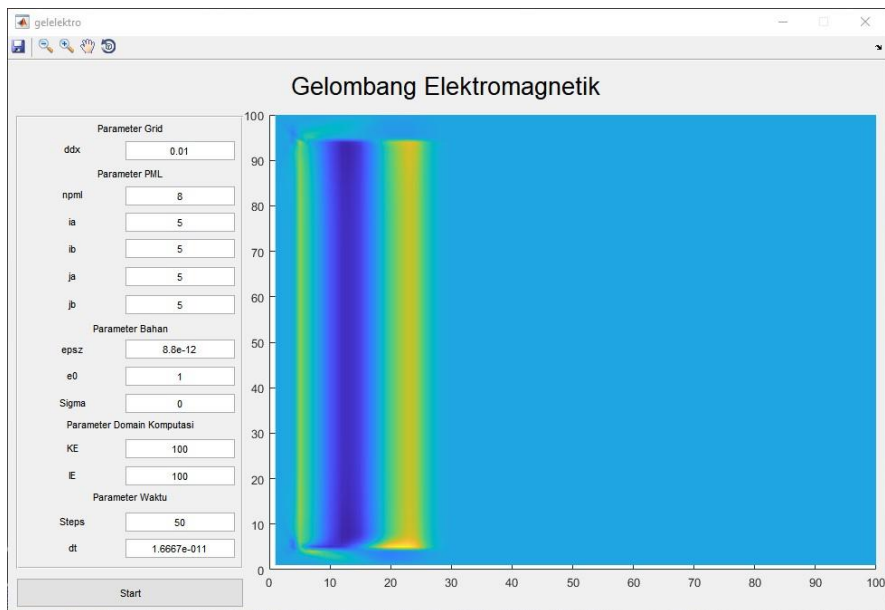


Gambar 17. Parameter PML dengan Nilai 5

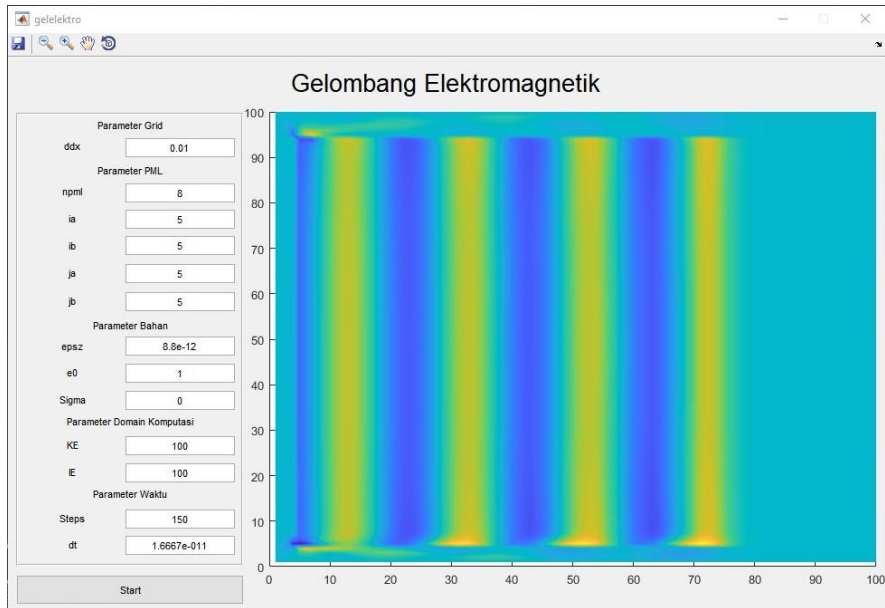


Gambar 18. Parameter PML dengan Nilai 8

5.2.2.4 Uji Parameter Waktu

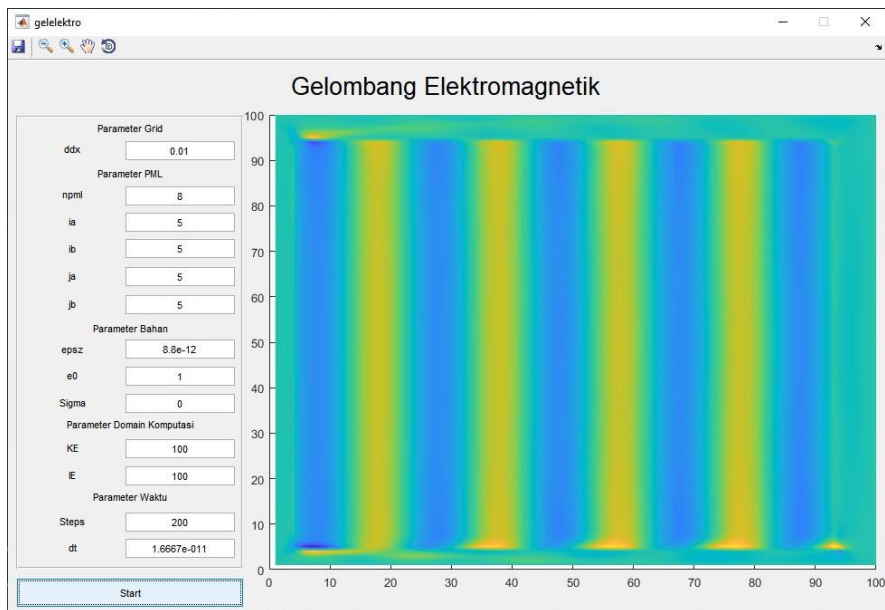


Gambar 19. Uji Parameter Waktu Dengan Nilai 50

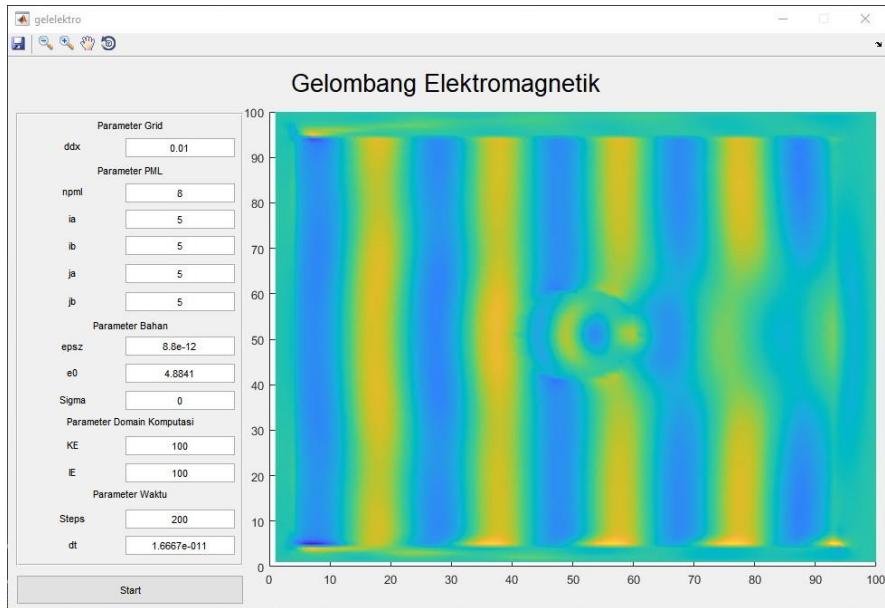


Gambar 20. Uji Parameter Waktu Dengan Nilai 150

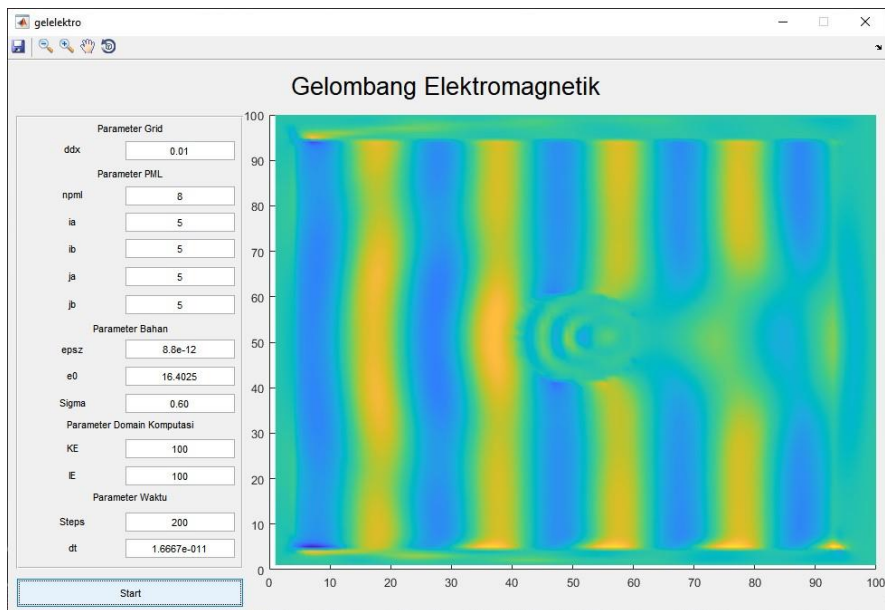
5.2.2.5 Uji Parameter Bahan



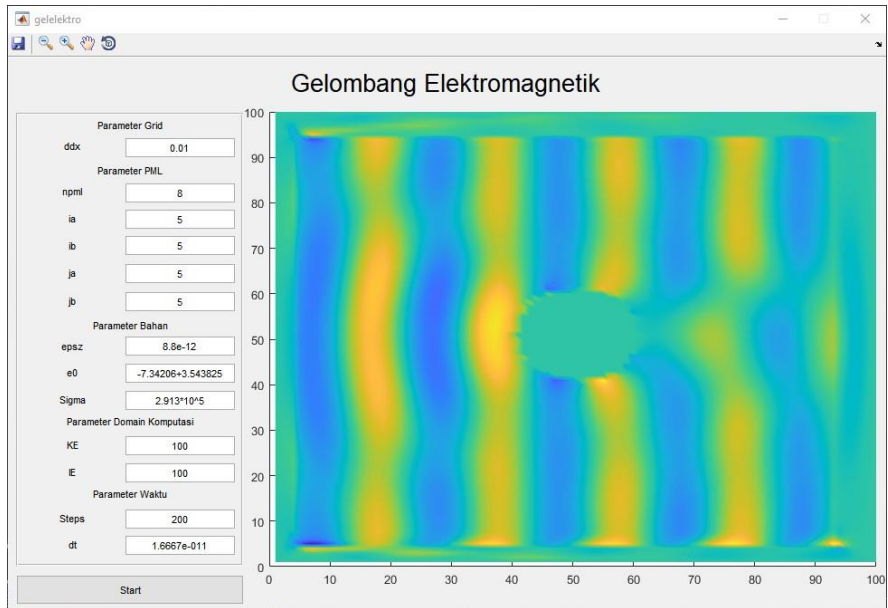
Gambar 21. Parameter Bahan Udara



Gambar 22. Parameter Bahan *Dielectric* (SiO2)



Gambar 23. Parameter Bahan *Lossy Dielectric* (Germanium)



Gambar 24. Parameter Bahan Metal (Tembaga)

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Perangkat Lunak *Graphic User Interface* (GUI) menggunakan metode *Finite Difference Time Domain* telah berhasil mensimulasikan gelombang elektromagnetik karena lebih unggul dalam memecahkan persamaan Maxwell dari sisi efisiensi, fleksibilitas dan kemudahan dalam mensimulasikan gelombang elektromagnetik. Metode FDTD lebih mudah karena berbasis numerik, dibandingkan metode lain yang berbasis analitik, seperti *transfer matriks* dan *couple mode*. Pada uji coba fungsional, sistem yang di buat sudah berfungsi dengan baik dan sesuai dengan fungsi tiap *button*. Pada uji coba validasi, perangkat lunak diuji berdasarkan parameternya berupa parameter *Grid*, parameter PML, parameter bahan, parameter domain komputasi, dan parameter waktu. Sistem sudah menunjukkan hasil yang sesuai dengan nilai yang di *inputkan* dan telah dibandingkan dengan hasil sebelumnya. Sistem interface ini dapat menjelaskan fenomena gelombang khususnya ketika berinteraksi dengan struktur

6.2 Saran

Penelitian ini masih memiliki kekurangan, maka dari itu diperlukan pengembangan di penelitian selanjutnya untuk meningkatkan kualitas perangkat lunak ini. Adapun saran dari penulis, yaitu:

1. Pada perangkat lunak ini jika ada objek pada rambatan gelombang maka ukuran dan posisi objek tidak berubah walaupun ukuran domain sudah diubah, diharapkan kedepannya ukuran dan posisi objek dapat diubah sesuai kebutuhan.
2. Pada saat ini perangkat lunak hanya bisa di akses melalui komputer, diharapkan kedepannya dapat didistribusikan untuk *website* maupun aplikasi *mobile*.

DAFTAR PUSTAKA

- Azhar, KA.** 2000. Penerapan Metode Beda Hingga untuk Penyelesaian Persamaan Gelombang Kawasan Waktu. *Jurnal Penelitian Sains*, Volume 7, Halaman 49-58. Universitas Sriwijaya.
- Daniel RL, Keith DP, John WS.** 1985. Finite Element Solution of Maxwell's Equations for Hyperthermia Treatment Planning. *Journal of Computational Physics*. Volume 58. Nomor 2, Halaman 246-269. New Hampshire. USA.
- David, JG.** 1995. Introduction to Electrodynamics, Halaman 304-314. New Delhi. Prantice Hall of India.
- Dennis, MS.** 2000. Electromagnetic simulation using the FDTD method, Halaman 1-9. University of Idaho. Wiley-IEEE Press.
- Eugene S, Maxim S.** 2019. Numerical Method for Electromagnetic Wave Propagation Problem in a Cylindrical Anisotropic Waveguide with Longitudinal Magnetization. *Advanced Electromagnetics*. Penza State University.
- Hadi, TY. Hermawan, KD. Andian, BS.** 2017. Penggunaan Metode Finite Difference Time Domain (FDTD) Dalam Simulasi Phased Array Antenna. *Journal of Science and Applicative Technology*, Halaman 19-24. Institut Teknologi Sumatera.
- Josaphat, TSS.** 1998. Analisa Hantaran Gelombang Listrik dengan Menggunakan Metode Finite Difference Time Domain (FDTD). Tutorial Riset Unggulan Terpadu (RUT) IV Pengembangan Sistem Radar Bawah Tanah Pulsa CHIRP. Bandung. BPPT.
- Kulyabov, DS.** 2017. Geometrization of Maxwell's Equations in the Construction of Optical Devices, Vol 25, No.1, Halaman 81-90. RUDN University.
- Teguh, PN.** 2006. Kristal Fotonik Asimetrik Omnidirectional Satu Dimensi Dengan Defek Geometris Skripsi. Bogor. Institut Pertanian Bogor.
- Teguh PN.** 2019. Simulasi Gelombang Elektromagnetik pada Waveguide Menggunakan Metode Finite Difference Time Domain untuk Aplikasi Radar, Volume 1, Nomor 1, Halaman 1-6. Yogyakarta.
- Teguh, PN.** 2020. Design of an Optical Biosensor Based on Photonic Crystal at THz Frequency. *Journal of Physics: Conference Series*. Volume 1567. Halaman 1-6. Semarang, Indonesia.
- Tjia, MO.** 1994. Gelombang, Halaman 123-129. Bandung. Debara Publisher

- Widjaja, MS.** 2001. Program Pembentukan Matrix Admitansi Pada Sistem Tenaga Listrik Menggunakan Matlab, Volume 1, Nomor 1, Halaman 1-16. Universitas Trisakti.
- Weitemeyer KA, Constable SC, Key KW, Behrens J.** 2006. First Results from a Marine Controlled-Source Electromagnetic Survey to Detect Gas Hydrates Offshore Oregon. Volume 33, Nomor 3, Halaman1-4. Geophysical Research Letters.
- Yahya, RS.** 2014. From Maxwell's Equations to Modern Antenna Marvels: From Tiniest Capsule Antennas to Largest Space Antennas. Volume 1, Halaman 575-576. Beijing, China. XXXIth URSI General Assembly and Scientific Symposium (URSI GASS).

LAMPIRAN

1. GUI

a. Inisialisasi

```
function varargout = gelelektro(varargin)

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',  gui_Singleton, ...
                  'gui_OpeningFcn', @gelelektro_OpeningFcn, ...
                  'gui_OutputFcn',  @gelelektro_OutputFcn, ...
                  'gui_LayoutFcn',  [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

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

% --- Executes just before gelelektro is made visible.
function gelelektro_OpeningFcn(hObject, eventdata, handles,
varargin)

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

% Update handles structure
guidata(hObject, handles);

cla;
if strcmp(get(hObject,'Visible'),'off')
    %plot(rand(20));
end

% --- Outputs from this function are returned to the command line.
```

```

function varargout = gelelektro_OutputFcn(hObject, eventdata,
handles)

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

```

b. Input

```

function input_ddx_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_ddx_CreateFcn(hObject, eventdata, handles)

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

function input_npml_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_npml_CreateFcn(hObject, eventdata, handles)

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

function input_ia_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_ia_CreateFcn(hObject, eventdata, handles)

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

```



```

end

function input_ib_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_ib_CreateFcn(hObject, eventdata, handles)

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

function input_ja_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_ja_CreateFcn(hObject, eventdata, handles)

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

function input_jb_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_jb_CreateFcn(hObject, eventdata, handles)

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

function input_epsz_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_epsz_CreateFcn(hObject, eventdata, handles)

```

```

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

function input_e0_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_e0_CreateFcn(hObject, eventdata, handles)

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

function input_sigma_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_sigma_CreateFcn(hObject, eventdata, handles)

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

function input_KE_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_KE_CreateFcn(hObject, eventdata, handles)

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

function input_IE_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.

```

```

function input_IE_CreateFcn(hObject, eventdata, handles)

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

function input_steps_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_steps_CreateFcn(hObject, eventdata, handles)

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

function input_dt_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function input_dt_CreateFcn(hObject, eventdata, handles)

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

```

c. Tombol

```

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)

% --- Executes on button press in savebutton.
function savebutton_ClickedCallback(hObject, eventdata, handles)

[file,path] = uiputfile('*.jpg');
filename = fullfile(path,file);
saveas(gelelektro,filename);

```

2. FDTD

a. Input & Handles

```
axes(handles.axes1);
cla;

t = 1:100;
v_ddx = str2num(get(handles.input_ddx, 'String'));
v_npml = str2num(get(handles.input_npml, 'String'));
v_ia = str2num(get(handles.input_ia, 'String'));
v_ib = str2num(get(handles.input_ib, 'String'));
v_ja = str2num(get(handles.input_ja, 'String'));
v_jb = str2num(get(handles.input_jb, 'String'));
v_epsz = str2num(get(handles.input_epsz, 'String'));
v_e0 = str2num(get(handles.input_e0, 'String'));
v_sigma = str2num(get(handles.input_sigma, 'String'));
v_KE = str2num(get(handles.input_KE, 'String'));
v_IE = str2num(get(handles.input_IE, 'String'));
v_step = str2num(get(handles.input_steps, 'String'));
v_dt = str2num(get(handles.input_dt, 'String'));

%clear all
clc
```

b. Inisialisasi

```
KE = v_KE;
ia = v_ia;
ja = v_ja;
ib = v_ib;
jb = v_jb;

ex = zeros(KE, KE); % Electric field vector E_x.
ez = zeros(KE, KE); % Electric field vector E_z.
dx = zeros(KE, KE); % Electric flux vector D_x.
dz = zeros(KE, KE); % Electric flux vector D_z.
hy = zeros(KE, KE); % Magnetic field vector.
hy_hat = zeros(KE, KE); % Parameter vector.
sx = zeros(KE, KE); % Medium vector.
sz = zeros(KE, KE); % Medium vector.
ix = zeros(KE, KE); % Parameter vector.
iz = zeros(KE, KE); % Parameter vector.
kc = KE/2; % Center of computational domain.
```

```

% PML vectors.
fi2 = ones(KE,1);
fi3 = ones(KE,1);
gi1 = zeros(KE,1);
gi2 = ones(KE,1);
gi3 = ones(KE,1);
fj2 = ones(KE,1);
fj3 = ones(KE,1);
gj1 = zeros(KE,1);
gj2 = ones(KE,1);
gj3 = ones(KE,1);

idx = zeros(KE,KE);
idz = zeros(KE,KE);

hy_low_m2 = 0;
hy_low_m1 = 0;
hy_high_m2 = 0;
hy_high_m1 = 0;
hy_inc = zeros(KE+1,1);
ex_inc = zeros(KE,1);

%-----
% Determine PML parameters.
%-----
npml = v_npml; % input('Enter the number of PML
cells: ');
for i= 0:npml
    xnum = npml - i;
    % D_z
    xxn = xnum/npml;
    xn = .333*xxn^3;
    gi1(i+1) = xn;
    gi1(KE-i) = xn;
    gi2(i+1) = 1/(1+xn);
    gi2(KE-i) = 1/(1+xn);
    gi3(i+1) = (1-xn)/(1+xn);
    gi3(KE-i) = (1-xn)/(1+xn);
    % for H_x and H_y
    xxn = (xnum - .5)/npml;
    xn = .25*xxn^3;

```

```

    fi2(i+1) = 1/(1+xn);
    fi2(KE-1-i) = 1/(1+xn);
    fi3(i+1) = (1-xn)/(1+xn);
    fi3(KE-1-i) = (1-xn)/(1+xn);
end

% We leave the Z+ boundary without a PML.
gj1 = gi1; %gj1(1:npml+1) = gi1(1:npml+1);
gj2 = gi2; %gj2(1:npml+1) = gi2(1:npml+1);
gj3 = gi3; %gj3(1:npml+1) = gi3(1:npml+1);
fj2 = fi2; %fj2(1:npml+1) = fi2(1:npml+1);
fj3 = fi3; %fj3(1:npml+1) = fi3(1:npml+1);

%-----
% Determine step sizes ddx and dt.
%-----
ddx = v_ddx; % Cell size. Eqn. (1.19). Here
lcm = .01.
dt = v_dt; % Calculate the time step size,
Eqn. (1.7).
%epsilon_0 = 8.8e-12;
epsilon_0 = v_epsz;
i0=KE/2;
j0=KE/2;
r=10;
for i=1:KE
    for j=1:KE
        d(i,j)=sqrt((i0+1-i)^2 + (j0+1-j)^2);
        if d(i,j) <= r
            e0(i,j)=-18.36;
            sigma(i,j)=3.4240e+004;%
            e0(i,j)=v_e0;%
            sigma(i,j)=v_sigma;
        else
            e0(i,j)=1;
            sigma(i,j)=0;
        end
    end
end
end

gb = (dt*sigma)./epsilon_0;
gbc = zeros(KE,KE);

```

```

ga = 1./(e0 + gb + gbc);

%-----
% Initialize variables.
%-----

T = 0; % Initialize time step.
show_output = 1; % Input('Enter 1 for plot at
each iteration; 0 otherwise: ');
NSTEPS = v_step; % Number of time steps.

```

c. FDTD Loop

```

for n = 1:NSTEPS
    T = T + 1;

    %-----
    % Determine the Hy field from Ex and Ez.
    %-----

    for j = 2:KE
        hy_inc(j) = hy_inc(j) + 0.5*(ex_inc(j-1)-ex_inc(j));
    end

    hy_inc(1) = hy_low_m2;
    hy_low_m2 = hy_low_m1;
    hy_low_m1 = hy_inc(2);
    hy_inc(KE) = hy_high_m2;
    hy_high_m2 = hy_high_m1;
    hy_high_m1 = hy_inc(KE-1);

    pulse = sin(2*pi*1500*1e6*dt*T);
    hy_inc(3)=hy_inc(3)+pulse;

    for i = 2:KE
        for j = 2:KE
            hy_hat_temp = gi3(i)*hy_hat(i,j) + gi2(i)*.5*(ez(i,j)-
ez(i-1,j)-ex(i,j)+ex(i,j-1));
            hy(i,j) = gj3(j)*hy(i,j) + gj2(j)*(hy_hat_temp -
hy_hat(i,j));
            hy_hat(i,j) = hy_hat_temp;
        end
    end
end

```

```

for i = ia:ib;
    hy(i,ja)=hy(i,ja)+0.5*ex_inc(ja-1);
    hy(i,jb)=hy(i,jb)-0.5*ex_inc(jb);
end
%-----
% Determine the Dx and Dz fields from Hy.
%-----
for j = 1:KE
    ex_inc(j) = ex_inc(j) + 0.5*(hy_inc(j)-hy_inc(j+1));
end

for i = 1:KE-1
    for j = 1:KE-1
        curl_h = hy(i+1,j) - hy(i,j);
        idz(i,j) = idz(i,j) + gj1(j)*curl_h;
        dz(i,j) = fi3(i)*dz(i,j) + fi2(i)*.5*curl_h +
fi2(i)*idz(i,j);

        curl_h = hy(i,j) - hy(i,j+1);
        idx(i,j) = idx(i,j) + gil(i)*curl_h;
        dx(i,j) = fj3(j)*dx(i,j) + fj2(j)*.5*curl_h +
fj2(j)*idx(i,j);
    end
end

for j =ja:jb
    dz(ia-1,j)=dz(ia-1,j)-.5*hy_inc(j);
    dz(ib,j)=dz(ib,j)+.5*hy_inc(j);
end

for i =ia:ib
    dx(i,ja-1)=dx(i,ja-1)+.5*hy_inc(ja);
    dx(i,jb)=dx(i,jb)-.5*hy_inc(jb);
end

ez = ga.*dz;
ex = ga.*dx;

```

d. Output Information

```

surf(ex);
view(0,90);

```



```
shading interp;  
a=ex;  
pause(0.1);  
end
```