



UNIVERSITAS INDONESIA

**Implementasi Perangkat Lunak untuk Pengolahan Sinyal dan Visualisasi Citra
Ultrasonografi berbasis Opensource**

TESIS

REZA PRIMARDIANSYAH
0906578150

**FAKULTAS TEKNIK
PROGRAM STUDI ELEKTRO
KEKHUSUSAN MULTIMEDIA**

**DEPOK
MEI 2011**



UNIVERSITAS INDONESIA

**Implementasi Perangkat Lunak untuk Pengolahan Sinyal dan Visualisasi Citra
Ultrasonografi berbasis Opensource**

TESIS

Diajukan sebagai salah satu syarat untuk memperoleh gelar Magister Teknik

REZA PRIMARDIANSYAH
0906578150

**FAKULTAS TEKNIK
PROGRAM STUDI ELEKTRO
KEKHUSUSAN MULTIMEDIA**

**DEPOK
MEI 2011**

HALAMAN PERNYATAAN ORISINALITAS

Tesis ini adalah hasil karya saya sendiri, dan semua sumber baik yang dikutip maupun yang dirujuk telah saya nyatakan dengan benar.

Nama : Reza Primardiansyah

NPM : 0906578150

Tanda Tangan : 

Tanggal : 24 Juni 2011



HALAMAN PENGESAHAN

Tesis ini diajukan oleh :
Nama : Reza Primardiansyah
NPM : 0906578150
Program Studi : Multimedia
Judul Tesis : Implementasi Perangkat Lunak untuk Pengolahan Sinyal dan Visualisasi Citra Ultrasonografi berbasis Opensource

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Magister Teknik pada program studi Multimedia, Fakultas Teknik Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Prof. Dr.-Ing. Ir. Kalamullah Ramli M.Eng. (Kalamullah)

Penguji : Prof. Dr. Ir. Riri Fitri Sari, M.Sc, MM.....(RFS)

Penguji : Dr. Ir. Anak Agung Putri Ratna, M. Eng.....(A.A.P.R.)

Penguji : Prima Dewi Purnamasari, ST., MT., M.Sc....(Prima Dewi Purnamasari)

Ditetapkan di : Depok

Tanggal : 24 Juni 2011

UCAPAN TERIMA KASIH

Puji syukur saya panjatkan kepada Allah swt. karena atas berkat dan rahmatNya saya dapat menyelesaikan tesis ini. Penulisan tesis ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gelar Magister Teknik Kekhususan Jaringan Informasi dan Multimedia pada Fakultas Teknik Universitas Indonesia . Saya menyadari bahwa tanpa bantuan dan bimbingan dari berbagai pihak, sejak masa perkuliahan sampai pada penyusunan tesis ini, sangatlah sulit bagi saya untuk menuntaskan ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Prof. Dr.-Ing. Ir. Kalamullah Ramli M.Eng selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan tesis ini.
2. Syahrul Imardi, S.T. MT. selaku pembuat implementasi sebelumnya yang telah membantu memberikan tambahan pemahaman.
3. Rekan-rekan penulis di tempat kerja yang telah bersabar memberikan keluasan waktu untuk pengerjaan kuliah dan tesis selama ini.
4. Kedua orang tua yang telah dengan sabar mengasuh, merawat, dan mengarahkan selama ini.
5. Belahan hati yang telah bersabar mendampingi selama ini.
6. Kedua buah hati yang terus menyemangati.

Akhir kata, saya berharap Allah membalas kebaikan semua pihak yang telah membantu terwujudnya tesis ini dalam bentuk kebaikan yang banyak, serta menjadikan keberkahan dalam ilmu ini dan menjadikan apa yang sudah dicapai ini dapat dikembangkan lebih jauh dan menjadikannya terpakai secara nyata.

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan di bawah ini:

Nama : Reza Primardiansyah
NPM : 0906578150
Program Studi : Jaringan Informasi dan Multimedia
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis karya : Tesis

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul:

**IMPLEMENTASI PERANGKAT LUNAK UNTUK
PENGOLAHAN SINYAL DAN VISUALISASI CITRA
ULTRASONOGRAFI BERBASIS OPENSOURCE**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di: Depok

Pada tanggal : 24 Juni 2011.

Yang menyatakan



(Reza Primardiansyah)

ABSTRAK

Nama : Reza Primardiansyah
Program Studi : Teknik Elektro
Judul : Implementasi Perangkat Lunak untuk Pengolahan Sinyal dan Visualisasi Citra Ultrasonografi berbasis Opensource

Ultrasonografi (USG) merupakan salah satu teknologi pencitraan medis yang paling banyak digunakan dalam dunia kedokteran saat ini. Beberapa faktor pemilihan teknologi USG ini adalah minimalnya faktor tingkat resiko serta persiapan fisik dan waktu pasien dan operator USG. Dari sisi harga teknologi USG juga relatif lebih murah dibandingkan teknologi pencitraan medis lainnya.

Walaupun demikian ketersediaan USG masih sedikit dibandingkan dengan banyaknya jumlah rumah sakit, klinik kesehatan dan terutama Puskesmas yang ada. Ini mengindikasikan, dari sisi harga teknologi USG masih belum dapat dijangkau masyarakat Indonesia. Salah satu faktor penyebabnya adalah sangat mahalnya lisensi platform dari perangkat lunak USG. Peneliti mengusulkan penggunaan opensource untuk mengatasi masalah tersebut.

Tujuan penelitian ini adalah untuk merancang dan mengembangkan suatu perangkat lunak berbasis open source baik dari sisi sistem operasi ataupun core module untuk USG. Eksperimen dilakukan dengan mengolah hasil data dari sinyal A-mode menjadi B-mode yang kemudian divisualisasikan.

Implementasi ini dibuat berdasarkan hasil penelitian sebelumnya yang menghasilkan implementasi dengan MATLAB. Implementasi ini menghasilkan kualitas citra visual yang dapat diterima walau memiliki kelemahan berupa artefak yang tidak tampil pada implementasi MATLAB. Kecepatan pengolahan dengan implementasi ini bervariasi dari 76,49% lebih lambat sampai 236,33% lebih cepat dari MATLAB. Waktu pengolahan data yang dibutuhkan pada kedua implementasi tersebut 500ms – 6391ms yang masih tidak memadai untuk menghasilkan kinerja yang dibutuhkan untuk penggunaan langsung.

ABSTRACT

Name : Reza Primardiansyah
Study Program : Electrical Engineering
Title : Implementation of Open Source based Software for USG Signal Processing and Visualization

USG is one of the most widely used medical imaging technique. Among the factors of the wide acceptance are the low risk factor and low preparation time for both patient and USG operator. USG technology is also more inexpensive than other imaging techniques.

However, USG equipments are still not adequately available in Indonesia, especially on Puskesmas (Public Health Centers). This indicates that USG equipments are still too expensive for people of Indonesia. One of the factors of the expensiveness is the licensing cost for USG platform software. Researcher suggests to use open source platforms to address this issue.

The purpose of this research is to design and develop an open source based software library. Experiment is done by processing A-mode data to B-mode to visualize.

This research is based on previous work that implements a library using MATLAB. Implementation of this research gives acceptable visual quality even though it has artifacts that are not present on MATLAB implementation. Processing speed with this implementation varies from 76,49% slower to 236,33% faster than MATLAB. Processing times from both implementations vary in 500ms – 6391ms range that is still inadequate for real life usage.

DAFTAR ISI

Halaman Pernyataan Orisinalitas.....	ii
Halaman Pengesahan.....	iii
Ucapan Terima Kasih.....	iv
Halaman Pernyataan Persetujuan Publikasi Tugas Akhir untuk Kepentingan Akademis.....	v
Abstrak.....	vi
Abstract.....	vii
Daftar Tabel.....	x
Daftar Gambar.....	xi
1. Pendahuluan.....	1
1.1. Latar Belakang Masalah.....	1
1.2. Perumusan Masalah.....	2
1.3. Pembatasan masalah.....	3
1.4. Tujuan dan Manfaat Penelitian.....	3
1.5. Metodologi Penelitian.....	4
1.6. Sistematika penulisan.....	4
2. Tinjauan Pustaka.....	6
2.1. Sejarah perkembangan pencitraan ultrasonografi.....	6
2.2. Gelombang akustik dan ultrasonik.....	8
2.3. Karakteristik Gelombang Ultrasonik.....	8
2.3.1. Perambatan Gelombang Ultrasonik.....	8
2.3.2. Panjang Gelombang, Frekuensi, dan Kecepatan.....	9
2.3.3. Energi dan Intensitas.....	10
2.4. Interaksi Gelombang Ultrasonik dengan Materi.....	12
2.4.1. Impedansi Akustik.....	13
2.4.2. Hamburan.....	13
2.4.3. Atenuasi.....	14
2.4.4. Pembiasan (Refraksi).....	14
2.4.5. Refleksi.....	15
2.5. Prinsip Ultrasonik.....	16
2.5.1. Transduser.....	16
2.5.2. Prinsip kerja ultrasonik.....	18
2.6. Pencitraan ultrasonografi.....	18
2.6.1. A-Mode.....	18
2.6.2. B-Mode.....	19
2.6.3. M-Mode.....	19
2.7. Pustaka pendukung implementasi perangkat lunak.....	21
2.7.1. SciPy dan NumPy.....	21
2.7.2. Matplotlib.....	21
2.7.3. wxPython.....	22
2.7.4. PyQt.....	23
3. Perancangan Software.....	25
3.1. Library Pengolah.....	25

3.1.1. Ambil RF data.....	26
3.1.2. Filter Data.....	26
3.1.3. Proses Log dan envelope.....	26
3.1.4. Tampilkan dalam bentuk colormap.....	27
3.2. Aplikasi Penguji.....	27
3.3. Tahapan pembuatan library.....	28
3.3.1. Visualisasi.....	28
3.3.2. Log dan envelope.....	28
3.3.3. Filtering.....	29
3.4. Pengujian Kinerja.....	29
4. Implementasi dan Analisis.....	30
4.1. Implementasi Pengolahan dan Penampilan Citra.....	30
4.1.1. Implementasi Penapisan (Filtering).....	30
4.1.2. Envelope dan log.....	31
4.1.3. Compound.....	31
4.1.4. Scan Conversion.....	32
4.1.5. Penampilan ke layar.....	33
4.2. Perbandingan Kinerja.....	33
4.2.1. Perbandingan Kinerja Pengolahan berkas “Layers”.....	34
4.2.2. Perbandingan Kinerja Pengolahan berkas “Body”.....	36
4.2.3. Analisa Kinerja.....	38
4.3. Analisis Visual citra.....	39
4.3.1. Metode Wawancara.....	39
4.3.2. Analisa Visual dari Hasil Pengolahan berkas “Body”.....	39
4.3.3. Analisa Visual Hasil Pengolahan dari berkas “Layers”.....	43
5. Kesimpulan.....	48
Daftar Acuan.....	49
Lampiran.....	50

DAFTAR TABEL

Tabel 2.1. Kecepatan dan Impedansi Suara dalam Berbagai Medium.....	10
Tabel 4.1. Perbandingan Kecepatan Pemrosesan Filter Low Pass Butterwoth antara Python dengan MATLAB (dalam ms).....	34
Tabel 4.2. Perbandingan Kecepatan Pemrosesan Filter Low Pass Orde 10 antara Python dengan MATLAB (dalam ms).....	35
Tabel 4.3. Perbandingan Kecepatan Pemrosesan Filter Low Pass Butterwoth antara Python dengan MATLAB (dalam ms).....	36
Tabel 4.4. Perbandingan Kecepatan Pemrosesan Filter Low Pass Orde 10 antara Python dengan MATLAB (dalam ms).....	37



DAFTAR GAMBAR

Gambar 2.1. Pembagian rentang frekuensi gelombang akustik (a) Acoustic spectrum dan (b) medical ultrasound spectrum [4].....	8
Gambar 2.2. Gelombang longitudinal [4].....	9
Gambar 2.3. Hamburan : (a) pada batas dua medium; (b) pada medium heterogen [11].....	13
Gambar 2.4. Refraksi gelombang [12].....	15
Gambar 2.5. Transduser [7].....	17
Gambar 2.6: Pola radiasi gelombang ultrasonik [13].....	17
Gambar 2.7. Sistem pulsa echo ultrasonik [14].....	18
Gambar 2.8. Visualisasi A-mode [15].....	19
Gambar 2.9. Tampilan M-Mode [15].....	20
Gambar 2.10. Proses Pencitraan USG [16].....	20
Gambar 2.11. Contoh Plot yang dihasilkan oleh Matplotlib[25].....	22
Gambar 2.12. Contoh Plot Matplotlib dalam Aplikasi wxPython [21].....	23
Gambar 2.13. Contoh Implementasi Plot Matplotlib dalam Aplikasi PyQt [23]..	24
Gambar 3.1. Flowchart tahapan pemrosesan.....	25
Gambar 3.2. Flowchart envelope.....	26
Gambar 3.3. Contoh tampilan visual yang diharapkan.....	27
Gambar 4.1. Hasil visualisasi dari berkas “Layers”.....	33
Gambar 4.2. Hasil visualisasi dari berkas “Body”.....	34
Gambar 4.3. Perbandingan Kecepatan Proses Filter Low Pass Butterworth.....	35
Gambar 4.4. Perbandingan Kecepatan Fiter low pass Orde 10.....	36
Gambar 4.5. Perbandingan Kecepatan Proses dengan Filter Butterworth.....	37
Gambar 4.6. Perbandingan Kecepatan Fiter low pass Orde 10.....	38
Gambar 4.7. Implementasi MATLAB tanpa Tapis.....	41
Gambar 4.8. Implementasi Python tanpa Tapis.....	41
Gambar 4.9. Hasil Penapisan Lulus Tengah Orde 10 yang Menghilangkan Artefak.....	42
Gambar 4.10. Perbandingan citra antara tanpa filter (a), FIR (b), dan Butterworth (c).....	43
Gambar 4.11. Hasil penapisan FIR dengan orde 2 (a),10 (b), dan 20 (c).....	44
Gambar 4.12. Hasil penapisan Butterworth dengan orde 2 (a) dan 10 (b).....	45
Gambar 4.13. Hasil Penapisan FIR Orde 2 untuk Python (a) dan MATLAB (b). .	45
Gambar 4.14. Hasil Penapisan FIR Orde 10 untuk Python (a) dan MATLAB (b)	46
Gambar 4.15. Hasil Penapisan Butterworth orde 2 untuk Python (a) dan MATLAB (b).....	46
Gambar 4.16. Hasil Penapisan Butterworth orde 10 untuk Python (a) dan MATLAB (b).....	47

BAB 1

PENDAHULUAN

1.1. LATAR BELAKANG MASALAH

Pencitraan memegang peranan yang sangat penting di bidang medis. Teknologi pencitraan medis menawarkan potensi sangat besar dalam membantu dokter melakukan pencegahan dini, penetapan diagnosis, serta pemberian *treatment* lanjutan kepada para pasien. Pada banyak kasus, teknologi ini juga telah menjadi langkah pertama dari serangkaian proses tindakan medis yang harus dilakukan dokter. Di sisi yang lain, pengolahan data citra medis berbasis komputer telah terbukti mempermudah proses yang harus dilakukan dokter dengan lebih cermat, teliti dalam waktu yang singkat. Namun sayangnya, di Indonesia nampaknya potensi tersebut belum banyak tergali baik pada tataran riset ataupun untuk keperluan real di rumah sakit[1].

Perkembangan proses pengolahan citra (*image processing*) ini didorong oleh pesatnya perkembangan dunia IT (*Information Technology*). Peralatan medis seperti CT-Scan, MRI, USG, dan lain-lain juga membutuhkan pencitraan sebagai sarana untuk memudahkan dalam mendiagnosa kelainan atau penyakit. Secara spesifik proses perkembangan pencitraan ini juga dialami pada pencitraan USG.

Pendiagnosaan menggunakan ultrasonografi memiliki banyak keuntungan, di antaranya:

1. Sensitif mendeteksi permukaan yang tidak homogen.
2. Jangkauan kedalaman yang cukup.
3. Hanya membutuhkan akses pada satu sisi tubuh.
4. Memiliki akurasi tinggi dalam menetapkan posisi refleksi dan memperkirakan ukuran ketidakhomogenan.
5. Memberikan hasil dalam waktu singkat.
6. Tanpa menggunakan radiasi pengion sehingga aman bagi tubuh manusia dan janin.
7. Lebih murah dibandingkan dengan cara lain.
8. Dapat dipindah-pindahkan.
9. Persiapan pasien tidak rumit.

10. Tidak menimbulkan rasa sakit

11. Tidak bersifat merusak.

Karena alasan inilah penggunaan USG banyak digunakan di berbagai rumah sakit dan klinik. Akan tetapi pada saat ini, USG yang digunakan rumah sakit dan klinik di Indonesia masih didatangkan dari luar negeri dengan harga mahal. Ini jelas berpengaruh kepada harga yang harus dibayar pasien yang membutuhkan pendiagnosaan menggunakan USG.

Di samping itu puskesmas yang merupakan ujung tombak layanan kesehatan masyarakat di Indonesia hampir tidak mungkin mampu mengadakan alat tersebut. Semetara itu tingginya angka kematian bayi baru lahir di Indonesia sebesar 10 dari 520 bayi tiap jam, dan kematian ibu melahirkan yang tertinggi di Asia, sebesar 307 per 100.000 kelahiran [2], disebabkan, di antaranya, tidak tersedianya teknologi USG untuk mengetahui kondisi bayi dalam kandungan.

USG dapat menjadi pilihan untuk mendiagnosa kelainan di dalam tubuh dan pemeriksaan kehamilan karena tidak mengandung resiko yang membahayakan seperti resiko yang ditimbulkan dari peralatan yang menggunakan zat radioaktif. Karena banyaknya keuntungan yang dapat diberikan oleh USG, maka pada saat ini makin banyak dilakukan penelitian dan upaya perancangan terkait dengan USG di Indonesia. Untuk itu, usaha dan penelitian berbasis open-source ini diharapkan mampu memberi kontribusi untuk penciptaan USG di Indonesia, sehingga mampu mengurangi harga yang harus dibayarkan.

1.2. PERUMUSAN MASALAH

Proses untuk menghasilkan citra dengan menggunakan gelombang ultrasonik sudah banyak berkembang, baik dari segi pengembangan teknologi transdusernya, data akuisisinya, maupun pengembangan perangkat keras pengolah dan juga perangkat lunaknya. Tetapi secara umum proses itu akan melalui proses berikut, gelombang yang dipancarkan oleh transmitter akan diterima oleh receiver (dalam hal ini keduanya tergabung dalam transduser). Gelombang yang diterima transduser berupa sinyal A-Mode yang selanjutnya diproses menjadi hasil B-Mode. Hasil B-Mode menunjukkan tingkan kecerahan citra berdasarkan tingkat *grayscale*.

Pengolahan citra dan penampilan citra ini berkembang dari tahun ke tahun mengikuti perkembangan teknologi komputer, tetapi dasar dari semua proses penampilan citra tersebut tetap pengolahan dari sinyal A-Mode dan B-Mode.

Dalam penelitian ini, akan dibuat implementasi pengolahan sinyal dan visualisasi citra USG berbasis opensource dengan menggunakan bahasa Python. Diharapkan dihasilkan sebuah library yang memiliki kemampuan dan kinerja memadai untuk digunakan dalam operasional sehari-hari.

1.3. PEMBATASAN MASALAH

Dalam penelitian ini, masalah yang akan diteliti dibatasi sesuai judul yang diajukan. Judul yang diajukan adalah “Implementasi Perangkat Lunak untuk Pengolahan Sinyal dan Visualisasi Citra Ultrasonografi berbasis Opensource”.

1. Implementasi dilakukan menggunakan bahasa pemrograman Python.
2. Data awal yang didapatkan yang berupa sinyal A-Mode diubah menjadi citra B-Mode menggunakan library yang dibuat.
3. Penampilan citra B-Mode dibatasi berupa citra 2-dimensi.
4. Peneliti tidak merancang perangkat keras untuk transduser ataupun menganalisa karakteristiknya.
5. Peneliti tidak merancang perangkat keras untuk akuisisi data, ataupun menganalisa kinerjanya.
6. Tidak membahas proses akuisisi data dari DAQ, ataupun analisa sinyal yang diolahnya.
7. Tidak membahas proses yang terjadi dalam sistem sehingga menjadi data A-mode.
8. Tidak membuat library pembaca file binary yang digunakan untuk pengujian.

1.4. TUJUAN DAN MANFAAT PENELITIAN

Tujuan penelitian ini adalah :

1. Mempelajari dasar dari teknik rekonstruksi visual hasil citra ultrasonik.
2. Mengimplementasikan teknik rekonstruksi yang sudah dibuat dalam penelitian sebelumnya ke dalam program yang open source sehingga mudah diimplementasikan dan dikembangkan lebih lanjut.

Manfaat penelitian ini adalah:

1. Menghasilkan pustaka perangkat lunak opensource yang dapat digunakan untuk kepentingan medis sehari-hari
2. Memberi kontribusi dalam pengembangan teknologi USG di Indonesia.

1.5. METODOLOGI PENELITIAN

Metode penelitian yang akan dilakukan terdiri dari beberapa tahap :

1. Studi kepustakaan.

Studi kepustakaan digunakan penulis untuk memperoleh teori-teori dasar sebagai sumber penulisan tesis.

2. Implementasi

Implementasi dilakukan berdasarkan hasil penelitian yang sudah dilakukan oleh peneliti sebelumnya. Implementasi ini dilakukan menggunakan bahasa Python dengan menggunakan library Matplotlib, SciPy, NumPy. Aplikasi pengujian dibuat menggunakan wxPython atau PyQt.

1.6. SISTEMATIKA PENULISAN

Sistematika pada penulisan laporan seminar ini dibagi menjadi 4 bab. Penulisan bab-bab dilakukan sebagai berikut:

Bab 1. Pendahuluan. Pada bab ini merupakan penjelasan secara umum yang menjelaskan latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penelitian, metodologi penelitian, serta sistematika penelitian.

Bab 2. Tinjauan Pustaka. Pada Bab ini penulis menguraikan sejarah dari perkembangan teknologi ultrasonik dan ultrasonografi secara umum dan dalam bidang medis. Menguraikan teori-teori dasar yang digunakan pada penulisan, implementasi, dan analisa dalam penelitian ini. Penjelasan dasar prinsip ultrasonografi, transduser, prinsip kerja ultrasonik ataupun pencitraan ultrasonografi itu sendiri. Mengulas tentang pustaka-pustaka yang akan digunakan dalam implementasi.

Bab 3. Perancangan Sistem Perangkat Lunak. Bab ini berisi perangkat keras sebagai sistem modulnya, perangkat lunak sebagai sistem utamanya, tahapan-tahapan

implementasinya, dan simulasi dengan data A-mode yang sudah ada untuk diolah menjadi citra visual.

Bab 4. Penutup. Bab ini berisi hasil yang diharapkan dari perancangan simulasi dan eksperimen.



BAB 2

TINJAUAN PUSTAKA

2.1. SEJARAH PERKEMBANGAN PENCITRAAN ULTRASONOGRAFI

Perkembangan penggunaan ultrasonik dalam berbagai bidang ilmu kedokteran saat ini berawal dari ditemukannya cara mengukur jarak di dalam air menggunakan gelombang suara. Pada saat itu dikenal istilah “sonar” (*sound navigation and ranging*). Sekitar tahun 1794, Lazzaro Spallansani, seorang ahli biologi Italia, mendemonstrasikan kemampuan seekor kelelawar menentukan arah terbang dan mencari mangsa dalam gelap dengan menggunakan gelombang suara berfrekuensi tinggi (ultrasonik).

Penerapan prinsip kerja gelombang ultrasonik dalam bidang kedokteran sekitar tahun 1920an mulai diterapkan. Hasil penelitian William Fry dan Russel Meyers membuktikan bahwa gelombang ultrasonik dapat digunakan untuk menghancurkan sel-sel basal ganglia pada penderita Parkinson. Kemampuan gelombang ultrasonik dalam menghancurkan ini kemudian secara luas diterapkan pula pada penyembuhan penyakit-penyakit lainnya. Pada tahun 1929 dan 1935, Solokov mempelajari penggunaan gelombang ultrasonik untuk mendeteksi obyek logam. Mulhauser pada tahun 1931 mendapat paten menggunakan gelombang ultrasonik dengan dua transduser untuk mendeteksi kecacatan pada logam.

Di awal tahun 1940 gelombang ultrasonik dinilai memungkinkan untuk digunakan sebagai alat mendiagnosis suatu penyakit setelah keberhasilan Karl Theodore Dussik dan Freiderich menemukan lokasi sebuah tumor otak dan pembuluh darah pada otak besar dengan mengukur transmisi pantulan gelombang ultrasonik melalui tulang tengkorak. Dengan menggunakan transduser, hasil pemindaian masih berupa gambar dua dimensi yang terdiri dari barisan titik-titik berintensitas rendah.

George Ludwik menyempurnakan temuan Dussik tersebut, dan berhasil menemukan lokasi batu ginjal. Pantulan gelombang ultrasonik tersebut dilihat pada layar osiloskop. Selanjutnya diketahui bahwa gelombang ultrasonik tersebut memerlukan panjang gelombang tertentu untuk mengidentifikasi obyek jaringan tubuh yang beragam kepadatannya. Pada 1949 John Julian Wild berkolaborasi dengan John Reid untuk menginvestigasi sel-sel kanker dengan alat ultrasonik. Beberapa jenis alat yang

dibuat untuk kepentingan investigasi tersebut antara lain adalah B-mode ultrasound, transduser jenis A-mode transvaginal dan transrectal.

Pada 1950, Wild bersama French dan Neal berhasil mengadakan riset dalam hal deteksi tumor-tumor otak. Di tahun yang sama Ludwig dan Struthers juga mengumumkan keberhasilan riset mereka dalam deteksi ultrasonik kandung kemih. Tahun 1952, Howry dan Bliss mengeluarkan hasil penelitian mereka dalam visualisasi ultrasonik untuk jaringan kulit dan otot.

Penelitian yang dilakukan Ian Donald pada 1955 terhadap kista ovarium dengan alat Metal Flaw Detector mulai membuka peluang dilakukannya berbagai penelitian lanjutan. Penelitian tentang hamburan balik gelombang ultrasonik dari hati yang normal dan abnormal sudah dimulai pada tahun 1981. Dari analisa amplitudo, diperoleh bahwa sirosis hati memiliki karakter rata-rata amplitudo dan distribusi amplitudo yang lebih besar daripada hati yang normal.

Beberapa hasil penelitian di bidang kandungan antara lain dengan ditemukannya metode penentuan ukuran janin (*fetal biometry*), teknologi transduser digital, transduser dua dimensi dan tiga dimensi, dan penentuan jenis kelamin janin dalam kandungan (*Fetal Anatomic Sex Assignment/FASA*).

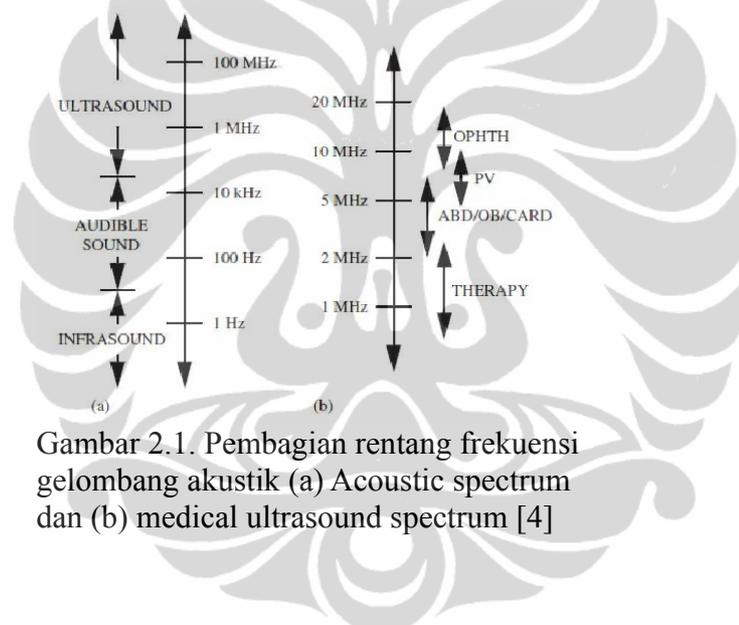
Penemuan metode penentuan fetal biometry dimulai pada 1980an. Berdasarkan tampilan gambar pada layar USG, beberapa parameter yang biasa dijadikan standar penentuan ukuran dan berat janin antara lain diameter kepala janin (*biparietal diameter/BPD*), keliling lingkaran kepala janin (*head circumference/HC*), panjang tulang paha (*femur length/FL*), dan lingkaran perut (*abdominal circumference/AC*). Metode ini dapat membantu para dokter ahli kandungan menentukan kenormalan pertumbuhan janin.

Teknologi transduser digital sekitar 1990an memungkinkan sinyal gelombang ultrasonik yang diterima menghasilkan tampilan gambar suatu jaringan tubuh dengan lebih jelas. Penemuan komputer pada pertengahan 1990 jelas sangat membantu teknologi ini.

2.2. GELOMBANG AKUSTIK DAN ULTRASONIK

Gelombang bunyi adalah gelombang yang dirambatkan sebagai gelombang mekanik yang dapat menjalar dalam medium padat, cair, dan gas. Gelombang bunyi ini merupakan getaran molekul-molekul zat dan saling beradu satu sama lain namun demikian zat tersebut terkoordinasi menghasilkan gelombang serta mentransmisikan energi bahkan tanpa terjadi perpindahan partikel [3]. Apabila gelombang bunyi merambat mencapai batas permukaan maka gelombang tersebut akan mengalami transmisi dan refleksi.

Gelombang ultrasonik merupakan gelombang mekanik dengan frekuensi di atas 20kHz. Diagram rentang frekuensi suara tampak pada gambar di bawah ini.

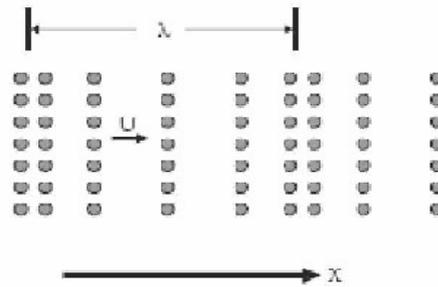


Gambar 2.1. Pembagian rentang frekuensi gelombang akustik (a) Acoustic spectrum dan (b) medical ultrasound spectrum [4]

2.3. KARAKTERISTIK GELOMBANG ULTRASONIK

2.3.1. Perambatan Gelombang Ultrasonik

Jenis perambatan gelombang ada dua, yaitu transversal dan longitudinal. Pada gelombang transversal arah getar tegak lurus dengan arah perambatan, sedangkan pada gelombang longitudinal arah getar searah dengan arah perambatan. Perambatan gelombang suara, termasuk gelombang ultrasonik, disebabkan oleh gerakan bolak-balik partikel medium searah dengan arah rambatannya sehingga gelombang suara merupakan gelombang longitudinal.



Gambar 2.2. Gelombang longitudinal [4]

2.3.2. Panjang Gelombang, Frekuensi, dan Kecepatan

Panjang gelombang (λ) adalah jarak yang ditempuh gelombang suara dalam periode satu getaran. Frekuensi (f) adalah banyaknya gelombang yang bergetar dalam satu detik, dalam satuan Hertz(Hz). Gelombang ultrasonik merupakan gelombang suara dengan frekuensi di atas 20 kHz. Frekuensi ultrasonik yang digunakan untuk diagnosis berkisar 1 sampai 10 MHz [5][6]. Periode adalah waktu yang dibutuhkan gelombang menempuh satu panjang gelombang, besarnya sama dengan $1/f$. Kecepatan ultrasonik (v) adalah jarak yang dilalui gelombang per satuan waktu dan senilai dengan panjang gelombang dibagi dengan periode, atau dengan kata lain:

$$c = \lambda \cdot f \quad (2.1)$$

dimana c (m/s) adalah kecepatan gelombang dalam medium, λ (m) adalah panjang gelombang, dan f (Hz) adalah frekuensi.

Kecepatan gelombang ultrasonik di dalam jaringan tubuh manusia diberikan dalam tabel berikut [8]

Tabel 2.1. Kecepatan dan Impedansi Suara dalam Berbagai Medium

Material	Velocity (m/s)	Acoustic Impedance (Rayls)
Air	331	0
Fat	1450	1.38
Water (50 C)	1540	1.54
Brain	1541	1.68
Blood	1570	1.61
Liver	1549	1.65
Kidney	1561	1.62
Muscle	1585	1.7
Lens of Eye	1620	1.84
Skull Bone	4080	7.8
Aluminium	6400	18

Sumber : Welch Allyn, 2004

2.3.3. Energi dan Intensitas

Daya adalah energi yang ditransfer, dinyatakan dalam satuan Watt. Intensitas adalah daya yang melewati suatu area tertentu, dinyatakan dalam Watt / m². Intensitas menunjukkan kuantitas energi ultrasonik yang diaplikasikan pada permukaan tertentu dalam tubuh pasien. Jika gelombang ultrasonik merambat dalam suatu medium, maka partikel medium mengalami perpindahan energi[9]. Besarnya energi gelombang ultrasonik yang dimiliki partikel medium adalah:

$$E = E_p + E_k \quad (2.2)$$

dimana E_p adalah energi potensial dan E_k adalah energi kinetik, keduanya dalam satuan Joule. Untuk menghitung intensitas gelombang ultrasonik perlu mengetahui energi yang dibawa oleh gelombang ultrasonik. Intensitas gelombang ultrasonik (I) adalah energi yang melewati luas permukaan medium 1 m/s atau watt/m [10]. Untuk sebuah permukaan, intensitas gelombang ultrasonik diberikan dalam persamaan:

$$I = \frac{1}{2} \cdot \rho \cdot V \cdot A \cdot (2\pi f)^2 = \frac{1}{2} \cdot Z \cdot (A\omega)^2 \quad (2.3)$$

dimana ρ adalah massa jenis medium (kg/m³), f adalah frekuensi (Hz), v adalah kecepatan gelombang (m/s), V adalah volume (m³), A adalah amplitudo maksimum (m), Z adalah impedansi akustik (kg/m.s), dan ω adalah frekuensi sudut (rad/s).

Gelombang ultrasonik merambat membawa energi dari satu medium ke medium lainnya. Energi yang dipindahkan sebagai energi getaran dari partikel ke partikel pada medium tersebut. Besarnya energi yang dibawa partikel tersebut adalah :

$$E = \frac{1}{2} k A \quad (2.4)$$

dimana

$$k = \text{konstanta} = 4 \pi^2 \cdot m / T^2 = 4 \pi^2 m f^2 \quad (2.5)$$

T = periode (s)

A = amplitudo gerak (m)

m = massa partikel (kg)

Kemudian :

$$E = 2 \pi^2 m f^2 A^2 \quad (2.6)$$

jika

$$m = \text{massa (kg)} = \rho V = \rho S l = \rho S v t$$

$$V = \text{volume} = \text{luas} \cdot \text{tebal} = S l \quad (\text{m}^3)$$

S = luas permukaan penampang lintang yang dilalui gelombang (m^2)

l = $v \cdot t$ = jarak yang ditempuh gelombang dalam waktu t (m)

v = laju gelombang (m/s)

t = waktu (s)

Maka :

$$E = 2 \pi^2 r S v t f^2 A^2 \quad (2.7)$$

Dari persamaan di atas diperoleh hasil bahwa energi yang dibawa oleh gelombang ultrasonik sebanding dengan kuadrat amplitudo. Besar daya yang dibawa gelombang ultrasonik (P) menjadi :

$$P = 2 \pi^2 r S v f^2 A^2 \quad (2.8)$$

Intensitas gelombang ultrasonik adalah daya yang dibawa melalui luas permukaan yang tegak lurus terhadap aliran energi, maka :

$$I = 2 \pi^2 r v f^2 A^2 \quad (2.9)$$

Persamaan di atas menyatakan hubungan secara eksplisit bahwa intensitas gelombang ultrasonik sebanding dengan kuadrat amplitudo (A) dan dengan kuadrat frekuensi (f).

Gelombang ultrasonik yang keluar dari sumber transduser mengalir ke semua arah. Gelombang ultrasonik merambat keluar, energi yang dibawanya tersebar ke permukaan yang makin lama semakin luas. Karena merambat dalam arah tiga dimensi, maka luas permukaan merupakan luasan permukaan bola dengan radius r yaitu $S = 4\pi r^2$

Berarti intensitas gelombang ultrasonik menjadi

$$I = P/S \quad (2.10)$$

Jika keluaran daya P dari sumber konstan, maka intensitas berkurang sebanding dengan kuadrat jarak dari sumber sehingga

$$I = I_0/r^2 \quad (2.11)$$

Jika diambil dua titik dengan jarak r_1 dan r_2 dari sumber, maka $I_1 = P/4\pi r_1^2$ dan

$$I_2 = P/4\pi r_2^2 \quad \text{sehingga}$$

$$\frac{I_2}{I_1} = \frac{r_1^2}{r_2^2} \quad (2.12)$$

jika amplitudo gelombang berkurang terhadap jarak, maka amplitudo mengecil sebesar $1/r$ [9],

Jika kita ambil dua jarak yang berbeda dari sumber transduser maka

$$\frac{A_2}{A_1} = \frac{r_1}{r_2} \quad (2.13)$$

Ketika gelombang ultrasonik dua kali lipat lebih jauh dari sumber transduser, maka amplitudo akan menjadi setengahnya [9].

Intensitas relatif digambarkan dalam satuan desibel (dB) sebagai

$$\text{Intensitas Relatif} = 10 \cdot \log\left(\frac{I_2}{I_1}\right) \quad (2.14)$$

2.4. INTERAKSI GELOMBANG ULTRASONIK DENGAN MATERI

Interaksi gelombang ultrasonik dengan jaringan mempengaruhi sinyal yang diterima receiver. Ini karena gelombang ultrasonik mempunyai sifat memantul, diteruskan, dan diserap oleh medium.

2.4.1. Impedansi Akustik

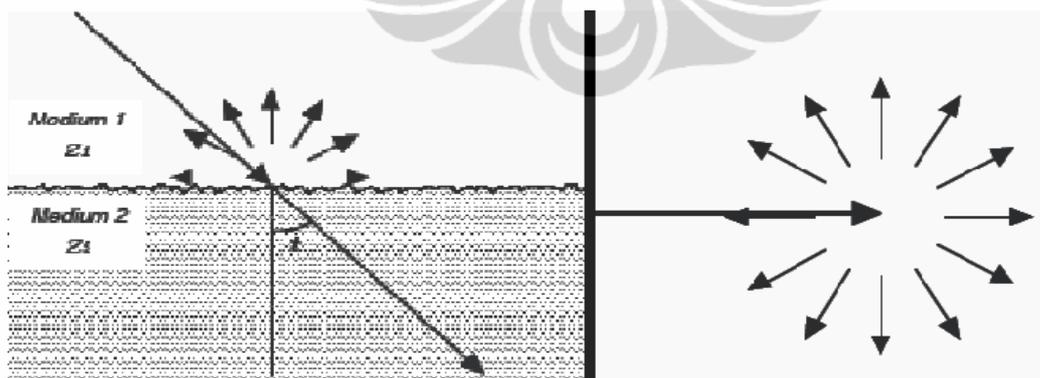
Ketika medium yang berdekatan memiliki impedansi akustik yang hampir sama, hanya sedikit energi yang direfleksikan. Impedansi akustik memiliki peran menetapkan transmisi dan refleksi gelombang di batas antara medium yang memiliki impedansi akustik yang berbeda. Impedansi akustik suatu materi didefinisikan sebagai perkalian antara massa jenis (ρ) dan kecepatan gelombang akustik (V)

$$Z = \rho V \quad (2.15)$$

dimana Z adalah impedansi akustik ($\text{kg/m}^2\text{s}$), ρ adalah massa jenis (kg/m^3) dan V adalah laju gelombang (m/s).

2.4.2. Hamburan

Fenomena hamburan mengikuti prinsip Huygen yang mengatakan bahwa setiap titik pada muka gelombang dapat dianggap sebagai sumber radiasi bola. Peristiwa hamburan yang terjadi ketika gelombang ultrasonik berinteraksi dengan batas antara dua medium, jika batas itu relatif rata maka pulsa ultrasonik dapat memantul yang disebut dengan *specular reflection* di mana semua pulsa akan diantulkan ke arah yang sama. Permukaan yang tidak rata menyebabkan gelombang pantulan dihamburkan ke segala arah seperti gambar di bawah ini. Hamburan juga terjadi dalam medium yang heterogen. Hamburan ini menyebabkan hanya sedikit gelombang pantul yang ditangkap kembali transduser untuk ditampilkan dalam bentuk citra.



Gambar 2.3. Hamburan : (a) pada batas dua medium; (b) pada medium heterogen [11]

2.4.3. Atenuasi

Gelombang suara yang merambat melewati suatu medium mengalami pelemahan intensitas seiring bertambahnya kedalaman. Penyebabnya adalah adanya penghamburan dan penyerapan[12]. Penyerapan merupakan perubahan bentuk energi ke bentuk lain seperti panas. Hal ini menyebabkan pulsa tersebut mengalami kehilangan energi. Kombinasi kedua efek tersebut disebut sebagai atenuasi atau pelemahan.

Besar energi yang diserap sebanding dengan koefisien pelemahan dan tebal medium yang dilalui. Setiap medium memiliki koefisien pelemahan yang berbeda. Semakin kecil koefisien pelemahan maka semakin baik medium tersebut sebagai penghantar. Penyerapan energi gelombang ultrasonik akan mengakibatkan berkurangnya amplitudo gelombang. Atenuasi berguna untuk menjelaskan fenomena berkurangnya intensitas gelombang ultrasonik. Besar amplitudo setelah mengalami atenuasi adalah

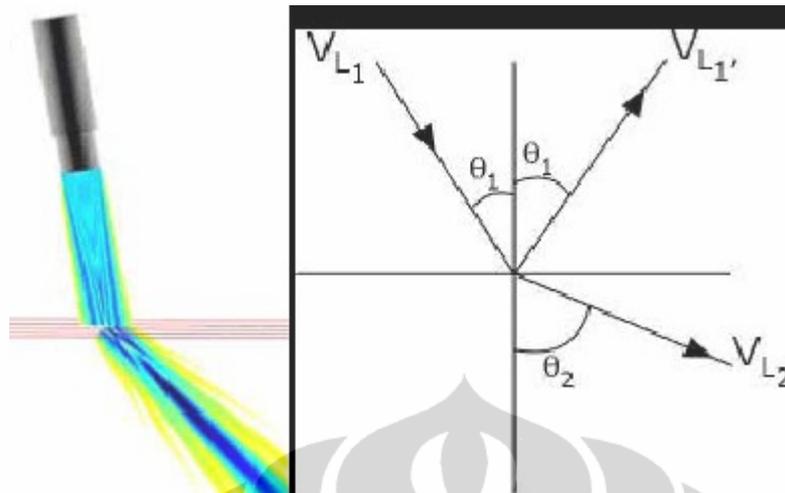
$$A = A_0 e^{-\alpha z} \quad (2.16)$$

di mana A_0 adalah amplitudo awal, A adalah amplitudo yang tereduksi setelah gelombang berjalan sejauh z , dan α adalah koefisien atenuasi. Secara umum, atenuasi sebanding dengan kuadrat frekuensi gelombang.

2.4.4. Pembiasan (Refraksi)

Refraksi adalah perubahan arah gelombang ultrasonik yang ditransmisikan pada batas antara medium yang berbeda ketika berkas gelombang tidak datang tegak lurus terhadap batas jaringan. Ketika gelombang ultrasonik melalui dua medium yang berbeda dengan sudut tertentu maka gelombang mengalami refraksi. Refraksi terjadi pada dua medium yang memiliki perbedaan impedansi akustik.

Hukum Snellius menggambarkan hubungan antara sudut (sudut datang dan sudut bias) dan kecepatan gelombang. Persamaan hukum Snellius menggambarkan perbandingan antara kecepatan gelombang di medium pertama (V_{L_1}) dan kecepatan gelombang di medium 2 (V_{L_2}) dengan sinus sudut datang (θ_1) dan sinus sudut bias (θ_2) seperti ditunjukkan pada gambar berikut



Gambar 2.4. Refraksi gelombang [12]

$$\frac{\sin(\theta_1)}{VL_1} = \frac{\sin(\theta_2)}{VL_2} \quad (2.17)$$

Ketika refraksi terjadi, dapat menyebabkan artifak (gangguan pada citra yang dihasilkan).

2.4.5. Refleksi

Pada pencitraan ultrasonografi, citra dihasilkan melalui berkas suara yang direfleksikan. Berkas gelombang yang dipancarkan tidak memperbesar apapun pada formasi citra, tapi transmisi harus cukup kuat menghasilkan gema-gema di tingkat yang lebih dalam. Prosentase suara yang direfleksikan di antara muka jaringan tergantung pada impedansi. Apabila gelombang ultrasonik mengenai permukaan antara dua jaringan yang memiliki perbedaan impedansi akustik, maka sebagian gelombang ini dipantulkan dan sebagian lagi diteruskan. Pulsa terpantul ini ditangkap oleh receiver untuk diolah menjadi citra. Refleksi yang sangat kuat terjadi pada batas organ.

Perbandingan amplitudo pantulan (R) terhadap amplitudo datang (A_0) bergantung pada impedansi akustik (Z) kedua medium tersebut dengan hubungan

$$\frac{R}{A_0} = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (2.18)$$

dengan Z_1 dan Z_2 adalah impedansi akustik dari kedua medium ($\text{kg/m}^2\text{s}$). Sebagian gelombang ultrasonik tersebut akan ditransmisikan (diteruskan). Perbandingan antara amplitudo transmisi (T) dengan amplitudo gelombang datang (A_0) adalah

$$\frac{T}{A_0} = \frac{2Z_1}{Z_1 + Z_2} \quad (2.19)$$

Koefisien intensitas pantulan, RI , didefinisikan sebagai perbandingan dari intensitas pantulan dan intensitas yang datang

$$RI = \frac{I_r}{I_i} = \left(\frac{Z_2 - Z_1}{Z_2 + Z_1} \right)^2 \quad (2.20)$$

dan koefisien intensitas transmisi adalah

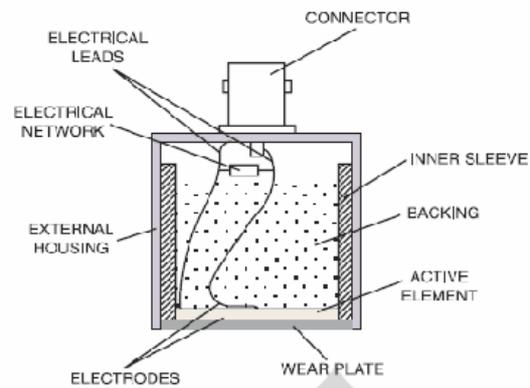
$$RT = \frac{I_t}{I_i} = \frac{4Z_2Z_1}{(Z_2 + Z_1)^2} \quad (2.21)$$

pada bagian tubuh yang lunak, hanya sebagian kecil pulsa yang direfleksikan. Untuk medium yang keras seperti tulang, energi yang direfleksikan sangat besar. Amplitudo pulsa dilemahkan oleh adanya absorpsi medium dan energi yang direfleksikan. Hal ini menyebabkan gelombang pantul yang dikirmkan kembali ke transduser sangat kecil dibandingkan dengan pulsa awal yang dihasilkan transduser.

2.5. PRINSIP ULTRASONIK

2.5.1. Transduser

Gelombang ultrasonik dihasilkan dan diterima oleh sebuah transduser. Transduser merupakan sebuah perangkat yang akan mentransformasikan suatu bentuk energi ke bentuk energi lain. Transduser ultrasonik akan mengubah energi listrik menjadi energi mekanik atau sebaliknya [12]. Pengiriman energi bisa berupa listrik, mekanik, kimia, optik, atau panas. Bagan transduser dapat dilihat pada gambar berikut.



Gambar 2.5. Transduser [7]

Transduser ultrasonik bekerja berdasar prinsip piezoelektrik di mana perlakuan mekanik yang diterima akan hasilkan muatan listrik. Sebaliknya, penerapan tegangan listrik menyebabkan perubahan ketebalan pada bahan. Bahan yang biasa dipakai adalah zirconate tatanate (PZT). Pada transduser, elemen ini mengubah sinyal listrik menjadi mekanik untuk hasilkan gelombang ultrasonik dan mengubah getaran mekanik dari pantulan gelombang ultrasonik yang diterima menjadi sinyal listrik[4]. Pola radiasi yang dipancarkan melalui transduser yang berada di depannya tergantung pada diameter transduser dan panjang gelombangnya sehingga transduser yang sama dapat memiliki pola radiasi yang berlainan jika medium yang dilalui juga berlainan. Pola radiasi suatu transduser ultrasonik merupakan gabungan antara gelombang bidang datar dan gelombang bola seperti tampak pada gambar berikut

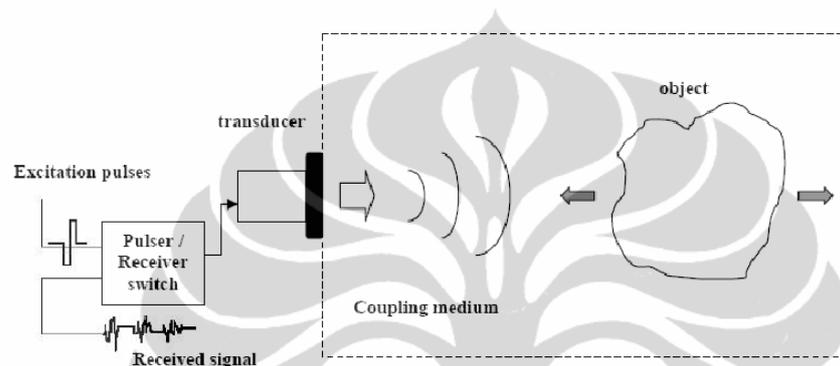


Gambar 2.6: Pola radiasi gelombang ultrasonik [13]

Terlihat bahwa dari permukaan transduser sampai jarak tertentu yang disebut medan dekat, gelombang ultrasonik yang dipancarkan merupakan gelombang bidang datar.

2.5.2. Prinsip kerja ultrasonik

Ultrasonik bekerja dengan cara memancarkan gelombang suara frekuensi tinggi ke tubuh pasien melalui transduser. Gelombang suara ini menembus tubuh dan mengenai batas-batas antar jaringan, seperti antara cairan dan otot, antara otot dan tulang. Sebagian gelombang suara ini dipantulkan kembali ke transduser, sebagian lain terus menembus bagian tubuh lainnya sampai kemudian juga dipantulkan seperti pada gambar



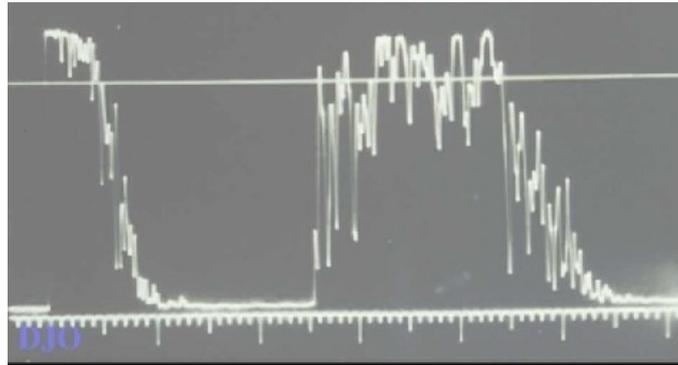
Gambar 2.7. Sistem pulsa echo ultrasonik [14]

Gelombang-gelombang suara pantulan ini ditangkap kembali oleh transduser dan diteruskan ke mesin ultrasonik yang akan menghitung berapa jarak jaringan pemantul dengan probe berdasarkan kecepatan suara di dalam jaringan. Lalu mesin ultrasonik menampilkan pantulan gelombang suara itu di layar dalam bentuk sinyal hasil pantul dari gelombang tersebut kemudian dideteksi dengan transduser yang mengubah gelombang akustik ke sinyal elektronik untuk diolah dan ditampilkan.

2.6. PENCITRAAN ULTRASONOGRAFI

2.6.1. A-Mode

A-Mode adalah singkatan dari Amplitude-mode. Contoh visualisasi sinyal A-mode tampak pada gambar berikut di mana pada sumbu X adalah waktu di mana gelombang pantul diterima dan sumbu Y adalah amplitudonya.



Gambar 2.8. Visualisasi A-mode [15]

A-mode display digunakan untuk gambarkan hubungan amplitudo pulsa pantul dengan kedalaman jaringan tubuh. Posisi sinyal echo di kedalaman jaringan dipengaruhi oleh interval waktu pulsa yang dikirim dan diterima. Cara ini biasanya untuk mengukur jarak pada tubuh dan ukuran organ internal. Pulsa ultrasonik merambat ke dalam jaringan tubuh sampai pada batas A jaringan yang memiliki impedansi akustik berbeda. Hal ini menyebabkan sebagian pulsa ultrasonik dipantulkan dan diterima oleh receiver sehingga menghasilkan echo A. Sebagian pulsa ultrasonik yang telah melewati batas A akan diteruskan sampai pada batas B sehingga dihasilkan echo C. Proses yang sama berlanjut untuk seterusnya.

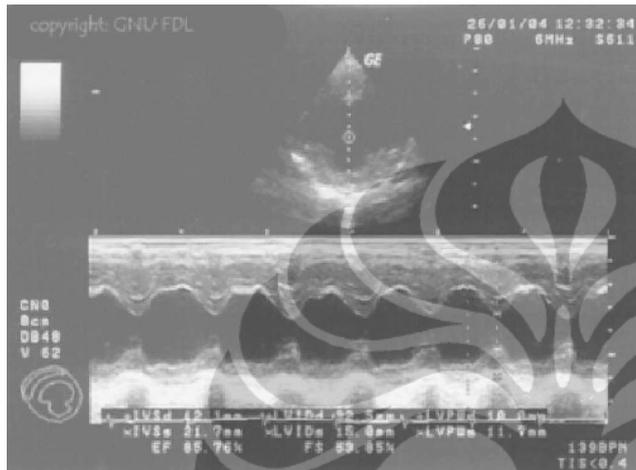
2.6.2. B-Mode

B-mode adalah singkatan dari brightness-mode, sebuah moda di mana gelombang echo ditampilkan sebagai warna. Warna menyesuaikan dari amplitudo (dalam grayscale dari hitam sampai putih). Mode ini dipergunakan di sonografi. Dalam ultrasound B-mode, satu array linear dari transduser secara simultan memindai (scan) satu benda melalui tubuh yang dapat dipandang sebagai suatu gambar dua dimensi pada layar. Probe ultrasound mengandung lebih dari 100 transduser sesuai urutan bentuk landasan pemakaian scanner yang paling umum.

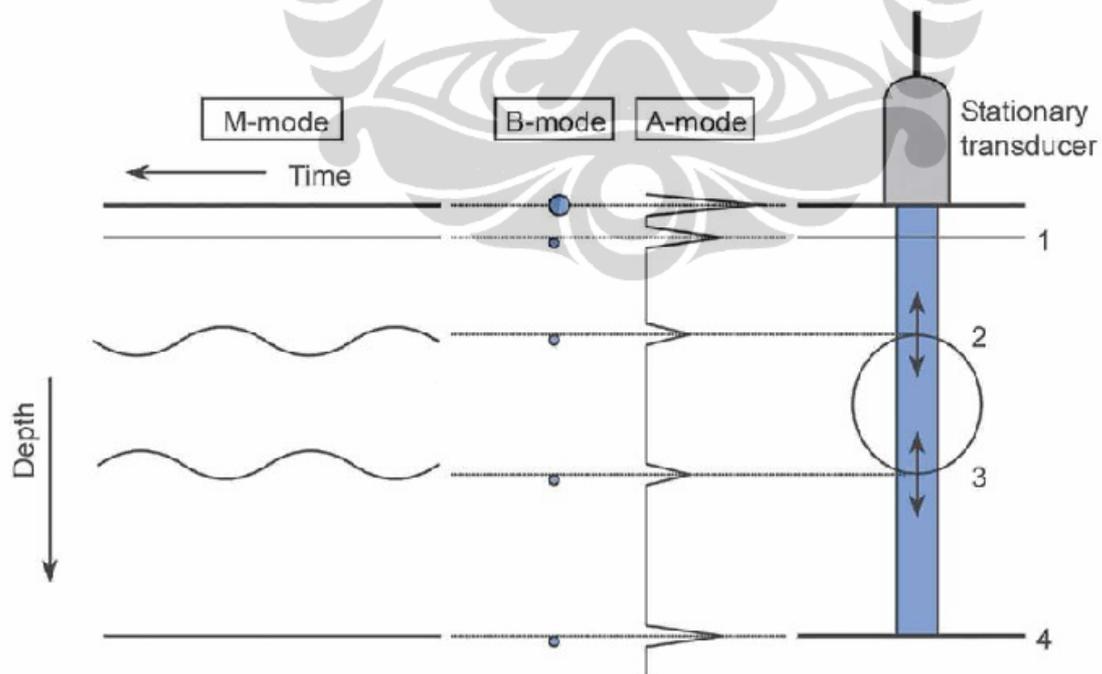
2.6.3. M-Mode

M-mode adalah singkatan dari Motion mode di mana amplitudo dan frekuensi saling berganti pada sumbu XY. Diagram ini biasanya khusus untuk detak jantung. Diagram ini sering terlihat dengan B-Mode. Suatu urutan secara cepat dari scan B-mode yang

mana gambar mengikuti satu dengan yang lainnya sesuai pada layar yang memungkinkan dokter untuk melihat dan menentukan ukuran dari gerak sebagai batas anggota tubuh yang menghasilkan gerakan pantulan pada probe. M-mode ultrasound telah dijadikan untuk penggunaat tertentu dalam mempelajari gerak jantung. Contoh tampilan M-mode dan proses pencitraannya dapat dilihat pada gambar berikut



Gambar 2.9. Tampilan M-Mode [15]



Gambar 2.10. Proses Pencitraan USG [16]

2.7. PUSTAKA PENDUKUNG IMPLEMENTASI PERANGKAT LUNAK

Implementasi perangkat lunak ini dilakukan dalam bahasa Python. Versi bahasa yang dipilih adalah versi 2.6 atau lebih tetapi tidak dalam versi 3 karena belum banyaknya penggunaan Python 3. Digunakan beberapa pustaka (*library*) tambahan untuk membuat perangkat lunak ini.

2.7.1. SciPy dan NumPy

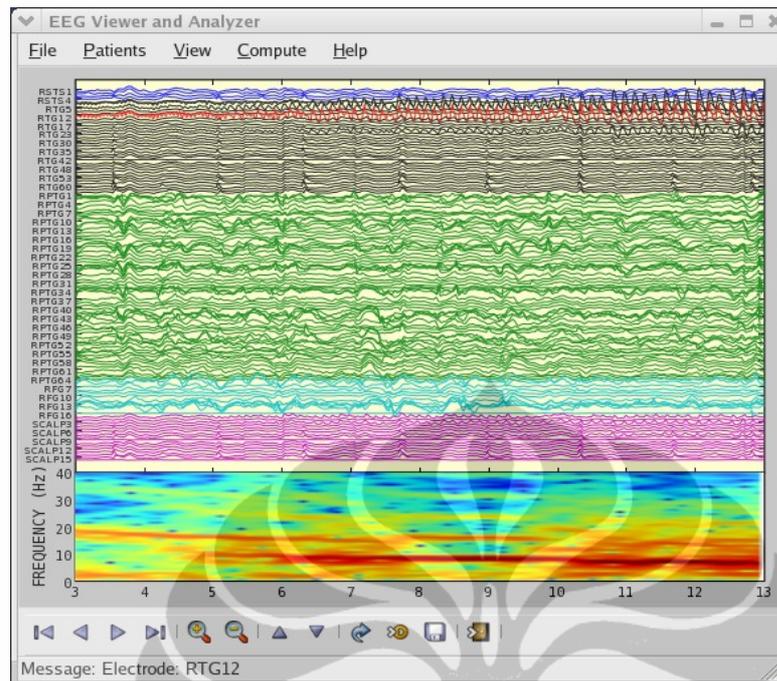
Untuk pengolahan numerik data sinyal USG digunakan pustaka numerik SciPy dan NumPy. SciPy adalah pustaka berisi algoritma dan peralatan matematis untuk bahasa Python. Di antara modul SciPy adalah optimisasi, integrasi, FFT. SciPy didistribusikan dengan lisensi BSD sehingga termasuk pustaka opensource.

Struktur data mendasar yang digunakan SciPy didapatkan dari modul NumPy. NumPy adalah modul tambahan untuk bahasa Python yang memberikan dukungan bagi larik (*array*) dan matriks multidimensi yang besar, juga sejumlah besar pustaka fungsi matematis yang mengolah data-data tersebut. Modul NumPy dirilis dengan lisensi BSD-new sehingga juga termasuk pustaka opensource.[17]

2.7.2. Matplotlib

Matplotlib adalah pustaka Python yang berguna untuk menampilkan data SciPy dalam bentuk plot atau chart. Matplotlib didistribusikan dengan lisensi BSD-style sehingga termasuk pustaka opensource.[18]

Contoh penampilan plot dari Matplotlib dapat dilihat pada gambar di bawah ini.



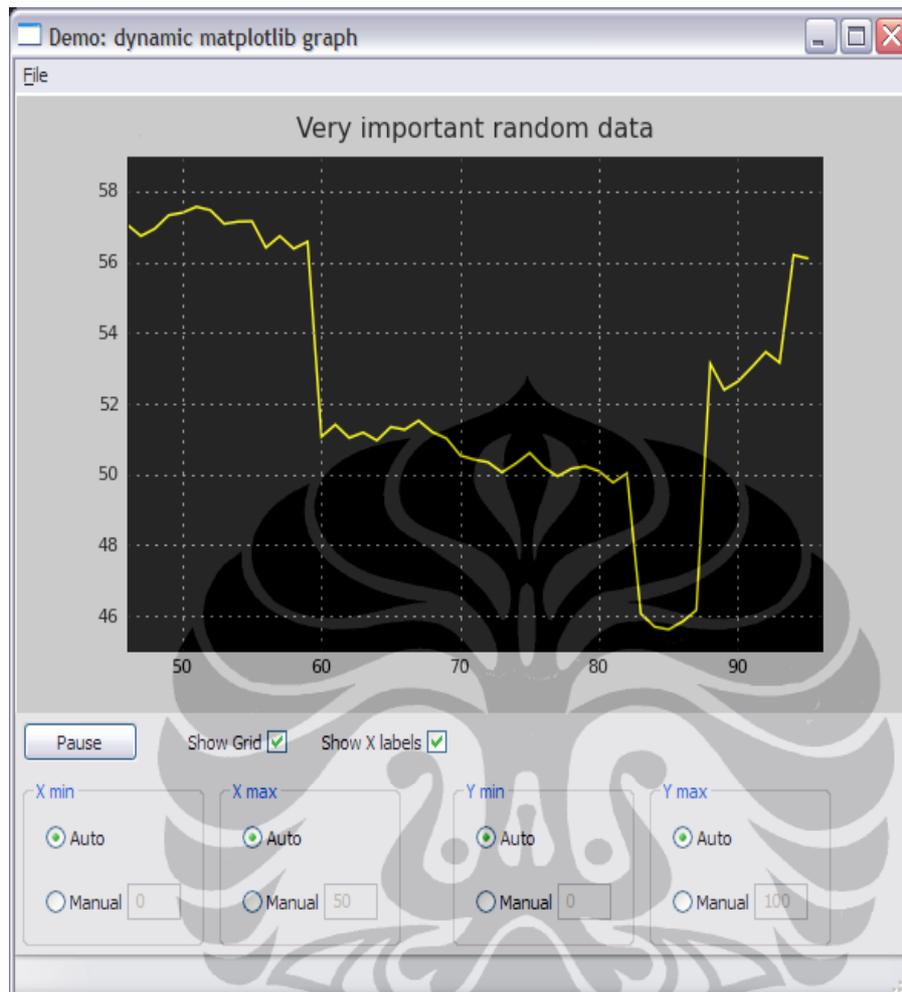
Gambar 2.11. Contoh Plot yang dihasilkan oleh Matplotlib[25]

2.7.3. wxPython

Salah satu pustaka pembuat aplikasi GUI adalah wxWidgets. Pustaka cross platform ini dimulai tahun 1992 dengan nama wxWindows oleh Julian Smart, dan berubah menjadi wxWidgets pada 2004. Di antara aplikasi yang menggunakan wxWidgets adalah BitTorrent, TortoiseCVS, dan Amaya. wxWidgets mendukung berbagai platform OS termasuk MS Windows, Linux, dan OSX.[19] wxWidgets memiliki lisensi LGPL dengan pengecualian bahwa “derived works in binary form may be distributed on the user's own terms“ sehingga dapat dipakai baik oleh aplikasi open source maupun aplikasi closed source. [20]

wxPython merupakan pustaka yang memungkinkan aplikasi yang dibuat dalam bahasa Python mengakses fungsi-fungsi wxWidgets. Pustaka wxPython ini memiliki lisensi yang sama dengan wxWidgets sehingga tergolong pula sebagai perangkat lunak open source.

Contoh penampilan data oleh Matplotlib dalam aplikasi GUI yang menggunakan wxPython dapat dilihat pada gambar berikut



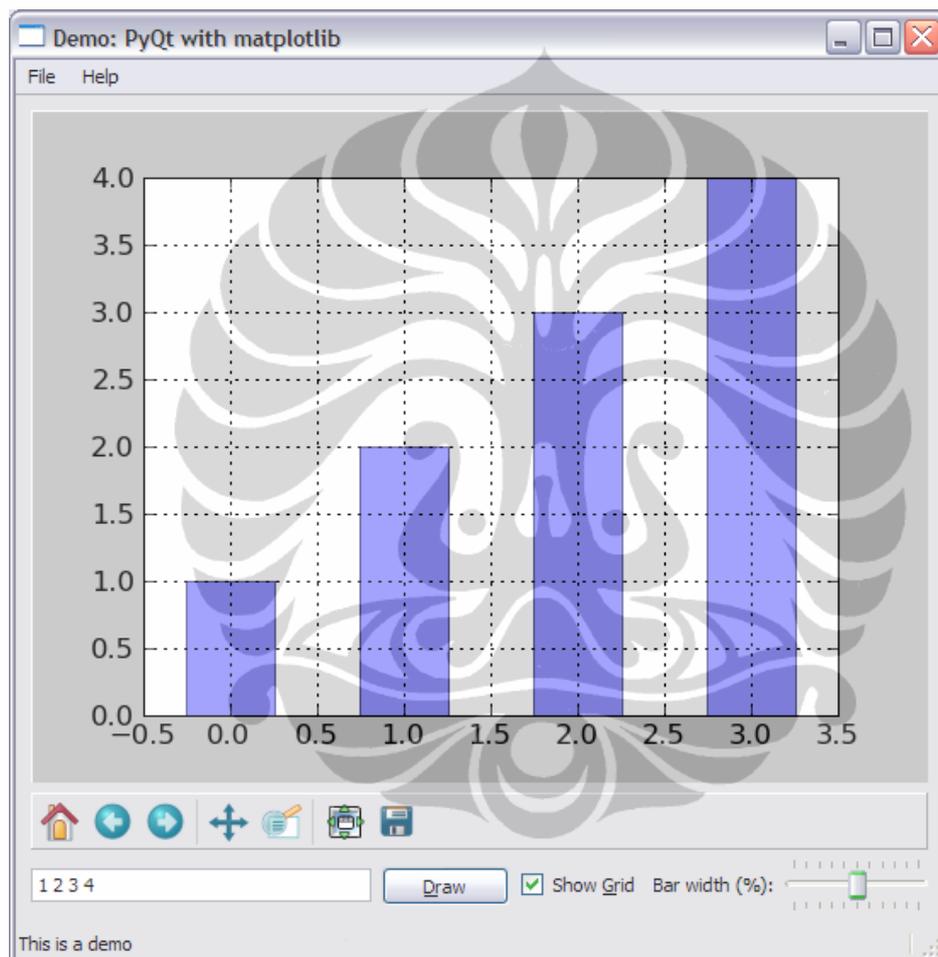
Gambar 2.12. Contoh Plot Matplotlib dalam Aplikasi wxPython [21]

2.7.4. PyQt

Salah satu pustaka pembuat aplikasi GUI lain adalah Qt. Qt adalah framework aplikasi yang cross platform yang dibuat oleh perusahaan Trolltech sejak tahun 1992 sebelum akhirnya Trolltech diakuisisi oleh Nokia. Qt mendukung berbagai platform sistem operasi termasuk MS Windows, Linux, OSX, dan Symbian. Qt digunakan dalam berbagai aplikasi termasuk Skype, Mathematica, VirtualBox, dan KDE. Qt memiliki beberapa lisensi termasuk LGPL 2.1 sehingga termasuk pustaka opensource. [22]

PyQt adalah pustaka yang memungkinkan aplikasi yang dibuat dalam bahasa Python mengakses fungsi-fungsi Qt. Dengan kata lain PyQt adalah Python binding dari Qt. Di

samping lisensi komersial, PyQt memiliki lisensi GPL sehingga dapat digolongkan sebagai pustaka open source. Tetapi PyQt tidak memiliki alternatif lisensi LGPL sehingga tidak dapat digunakan oleh aplikasi yang closed source. Dalam penelitian ini PyQt merupakan salah satu alternatif pustaka pembuat aplikasi pengujian. Matplotlib mendukung penampilan data dalam aplikasi PyQt. Contohnya dapat dilihat pada gambar berikut.



Gambar 2.13. Contoh Implementasi Plot Matplotlib dalam Aplikasi PyQt [23]

BAB 3 PERANCANGAN SOFTWARE

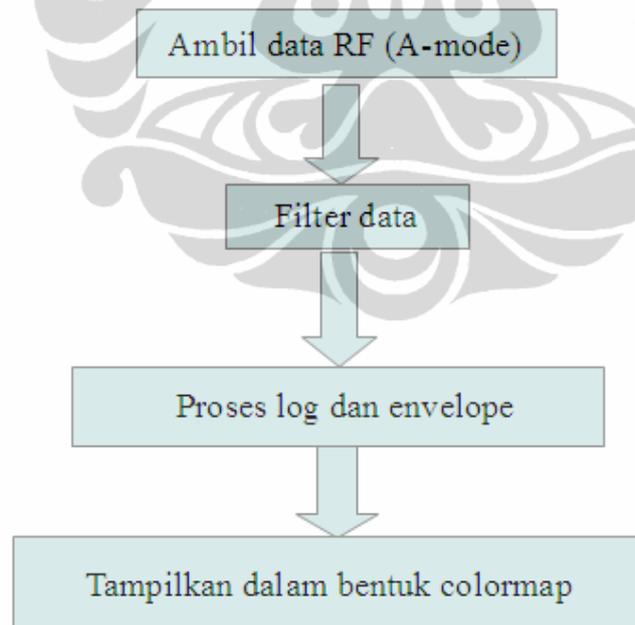
Sistem ini diimplementasikan sebagai kelanjutan dari hasil penelitian yang dilakukan oleh Syahrul Imardi[24]. Penelitian tersebut dilakukan dengan menggunakan aplikasi MATLAB yang membaca dan menampilkan data USG hasil pembacaan yang tersimpan dalam *file* biner dalam format ANTARES.

Penelitian ini akan membuat pustaka pengolah dan visualisasi hasil pembacaan USG. Software ini akan dibangun menggunakan bahasa Python dengan pustaka visualisasi Matplotlib.

3.1. LIBRARY PENGOLAH

Library pengolah akan menyediakan fungsi-fungsi yang memfilter dan mengubah data A-mode menjadi data B-mode, juga menyediakan fungsi yang memetakan data B-mode tersebut menjadi citra visual yang akan ditampilkan. Data A-mode diterima dalam bentuk Array NumPy, begitu pula dengan data koordinat XY yang dipakai untuk menampilkan data B-mode.

Tahap-tahap pengolahan data tampak pada gambar berikut.



Gambar 3.1. *Flowchart* tahapan pemrosesan

3.1.1. Ambil RF data

RF data berjenis A-mode diterima dari aplikasi pemanggil library. Data ini pada prakteknya didapatkan oleh aplikasi dari pembacaan transduser. Data transduser harus dikumpulkan (*buffer*) dalam bentuk *array* NumPy untuk kemudian dikirimkan sebagai parameter fungsi penampil data.

3.1.2. Filter Data

Pada tahap ini dapat dilakukan proses filter data. Proses filter ini digunakan untuk menghilangkan atau meredam *noise* yang didapatkan dari proses pemindaian (*scanning*) oleh transduser. Proses filter tersebut dapat berupa *low pass filter* maupun *high pass filter*. Fungsi yang melakukan filter data menerima data A-mode dalam bentuk *array* NumPy dalam parameteranya. Frekuensi acuan untuk proses filter juga dimasukkan sebagai parameter. Fungsi ini mengembalikan data A-mode yang sudah terfilter dalam bentuk *array* NumPy.

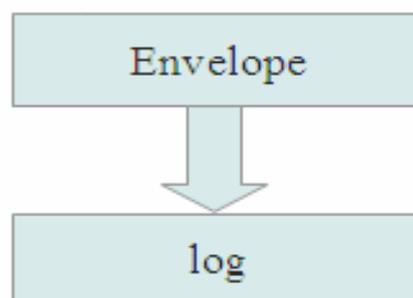
Dalam prakteknya dapat digunakan proses kalibrasi untuk menentukan besaran angka frekuensi yang diperlukan untuk proses filter ini.

3.1.3. Proses Log dan envelope

Sinyal A-mode yang sudah terfilter kemudian harus melalui proses untuk mendapatkan sinyal *envelope*. Pada tulisan ini digunakan Hilbert Transform untuk memprosesnya.

Setelah sinyal *envelope* didapatkan, digunakan proses konversi log sinyal untuk dapatkan sinyal B-Mode yang dapat ditampilkan.

Tahapan proses ini ditampilkan pada gambar berikut.



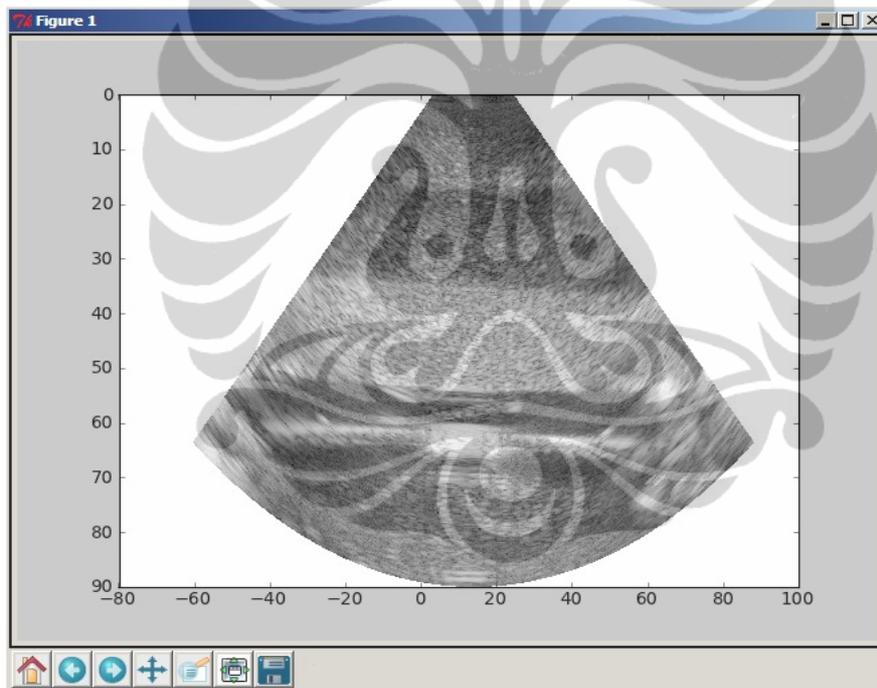
Gambar 3.2. Flowchart envelope

3.1.4. Tampilkan dalam bentuk colormap

Karena RF data diterima dalam bentuk array/matriks 2 dimensi, perlu dipetakan lokasinya sehingga visualisasinya tampak dalam satuan ukuran jarak (mm atau cm). Karenanya fungsi untuk melakukan penampilan membutuhkan dua buah array tambahan sebagai peta lokasi dari masing-masing indeks matriks, masing-masing untuk sumbu X dan sumbu Y. Array tambahan tersebut harus disediakan oleh aplikasi pemanggil. Data untuk peta lokasi dapat didapatkan dari proses kalibrasi.

Pada sistem ini dapat digunakan fungsi pcolor dari library Matplotlib. Tetapi jika data terlalu besar yang akibatkan pcolor terlalu banyak memakan memori dan swap file, dapat digunakan fungsi pcolormesh.

Contoh tampilan visual dapat dilihat pada gambar berikut.



Gambar 3.3. Contoh tampilan visual yang diharapkan

3.2. APLIKASI PENGUJI

Aplikasi penguji akan dibuat dengan menggunakan library wxPython atau PyQt. Aplikasi ini akan melakukan pembacaan data dari file dan kemudian memanggil library pengolah dan penampil yang telah diuraikan di atas.

Berbeda dengan penelitian sebelumnya yang membaca data dari *file binary* ANTARES, pada penelitian ini tidak diimplementasikan pembacaan *file binary* tersebut karena penelitian ini bertujuan untuk mengolah dan menampilkan hasil pembacaan dari alat USG yang tidak terbatas pada pembacaan *file*. Untuk ujicoba, digunakan pembaca *file* berformat MAT yang digunakan oleh MATLAB. NumPy telah menyediakan fungsi tersebut. Dengan demikian pengujian dilakukan sebagai berikut:

1. Menggunakan program MATLAB yang menggunakan *library* pembaca data biner tersebut untuk menerima data yang disimpan dalam variabel MATLAB. Data biner tersebut disimpan dalam file MAT.
2. Aplikasi pengujian membaca data biner tersebut dari file MAT.
3. Aplikasi pengujian mengolah dan menampilkan gambar B-Mode.

3.3. TAHAPAN PEMBUATAN LIBRARY

Karena penelitian ini merupakan upaya *porting* dari hasil penelitian sebelumnya yang menghasilkan *library* dalam program MATLAB, maka dilakukan proses *porting* bertahap dari fungsi-fungsi tersebut. Indikator kesuksesan masing-masing tahapan adalah tampilnya visualisasi seperti yang diharapkan.

3.3.1. Visualisasi

Untuk memudahkan pengujian, *porting* dilakukan dari tahap terakhir yaitu visualisasi. Pada tahap ini dilakukan proses visualisasi dari data B-mode yang dihasilkan oleh program yang sudah ada yang dibuat dengan MATLAB tersebut di atas beserta data peta lokasinya. Data B-mode dan peta lokasi tersebut dituliskan ke file .MAT dan dibaca oleh program pengujian dan ditampilkan dengan menggunakan pustaka ini.

3.3.2. Log dan envelope

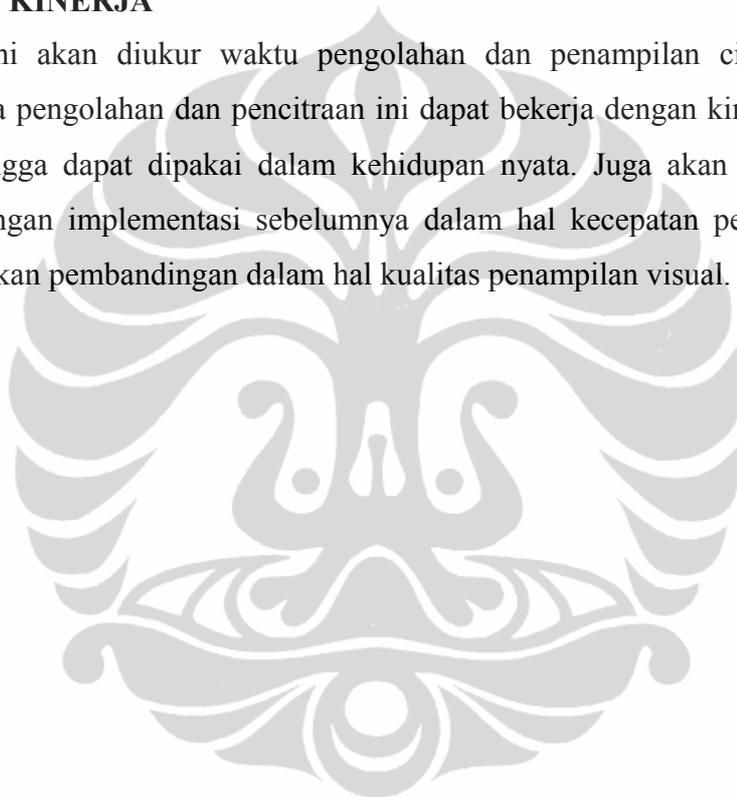
Untuk menguji tahap ini data A-mode yang sudah mengalami proses filter disimpan dalam file .MAT yang lalu dibaca program pengujian. Pada data A-mode tersebut dilakukan proses log dan envelope untuk kemudian ditampilkan dengan *library* yang sudah diuji di atas. Peta lokasi juga dibaca dari file .MAT tersebut.

3.3.3. Filtering

Untuk menguji tahap ini data RF sebagai data A-mode yang belum mengalami proses filter disimpan dalam file .MAT yang lalu dibaca program penguji. Pada data A-mode tersebut dilakukan proses filter untuk kemudian dilakukan padanya proses log dan *envelope* untuk kemudian ditampilkan dengan pustaka yang sudah diuji di atas. Peta lokasi juga dibaca dari file .MAT tersebut.

3.4. PENGUJIAN KINERJA

Pada pengujian ini akan diukur waktu pengolahan dan penampilan citra untuk memastikan bahwa pengolahan dan pencitraan ini dapat bekerja dengan kinerja yang cukup cepat sehingga dapat dipakai dalam kehidupan nyata. Juga akan dilakukan perbandingan dengan implementasi sebelumnya dalam hal kecepatan pengolahan. Juga akan diupayakan perbandingan dalam hal kualitas penampilan visual.



BAB 4 IMPLEMENTASI DAN ANALISIS

4.1. IMPLEMENTASI PENGOLAHAN DAN PENAMPILAN CITRA

4.1.1. Implementasi Penapisan (Filtering)

Pada implementasi pustaka ini diberikan beberapa opsi penapisan yang dapat dipakai, yaitu tanpa penapisan, tapis FIR, dan tapis Butterworth. Pemilihan tipe penapisan dilakukan pada fungsi `filt` berikut.

```
def filt(rfData, fOrder, fCutoffLow, fCutoffHigh, fType):  
    if fType == 0:  
        return rfData  
    if fType == 1:  
        return filtFIR(rfData, fOrder, fCutoffLow, fCutoffHigh)  
    if fType == 2:  
        return filtButterworth(rfData, fOrder, fCutoffLow, fCutoffHigh)
```

Penapisan dengan FIR dilakukan oleh fungsi berikut

```
def filtFIR(rfData, fOrder, fCutoffLow, fCutoffHigh):  
    filterCutoffs = array([fCutoffLow, fCutoffHigh])  
    filterCutoffs = filterCutoffs[filterCutoffs>0]  
    numtaps = filterOrder + 1  
  
    if filterCutoffLow == 0:          # low pass filter  
        B_fir = firwin(numtaps, filterCutoffs)  
        return lfilter(B_fir, 1, rfData)  
    else:                             # band pass or high pass filter  
        B_fir = firwin(numtaps, filterCutoffs, pass_zero=False)  
        return lfilter(B_fir, 1, rfData)
```

Dan penapisan dengan Butterworth dilakukan dengan fungsi berikut

```

def filtButterworth(rfData, fOrder, fCutoffLow, fCutoffHigh):
    filterCutoffs = array([filterCutoffLow, filterCutoffHigh])
    filterCutoffs = filterCutoffs[filterCutoffs>0]

    if filterCutoffHigh == 0:          # high pass filter
        b,a = butter(filterOrder, filterCutoffs, btype='high')
        return lfilter(b,a, unfilteredRfData)
    if filterCutoffLow == 0:          # low pass filter
        b,a = butter(filterOrder, filterCutoffs, btype='low')
        return lfilter(b,a, unfilteredRfData)
    else:
        b,a = butter(filterOrder, filterCutoffs, btype='band')
        return lfilter(b,a, unfilteredRfData)

```

4.1.2. Envelope dan log

Pengambilan sinyal envelope dan pengkompresian dengan log dilakukan oleh fungsi berikut

```

def logenv(filteredRfData, const=10, factor=1):
    return log(const+factor*abs(hilbert(filteredRfData)));

```

4.1.3. Compound

Jika pada perangkat terdapat beberapa *focal point*, terjadi duplikasi data sinyal yang diterima. Karenanya data tersebut harus dipilah untuk mendapatkan data dari masing-masing *focal point*. Data yang sudah dipilah dapat dipilih yang memiliki kecerahan terbesar atau dapat diambil rata-ratanya. Pemrosesan ini dilakukan dengan fungsi berikut

```

def compound(b_modeArray, numFocalZones=1, compoundMethod=1):
    if numFocalZones <= 1:
        return b_modeArray
    row,col = shape(b_modeArray)
    depth = col/numFocalZones
    newCol = numFocalZones
    result = reshape(b_modeArray, (row, depth, newCol))
    if compoundMethod == 1:
        result = squeeze(amax(result,axis=2));
    else:
        result = squeeze(mean(result,axis=2));
    return result

```

4.1.4. Scan Conversion

Data yang sudah diolah tersebut tersimpan dalam matriks dua dimensi. Untuk dapat mengetahui ukuran benda yang diukur, tiap indeks matriks harus dipetakan ke dalam bentuk ukuran jarak seperti milimeter (mm). Pembuatan peta indeks ini dilakukan oleh fungsi berikut

```

def scanconv(thetaRad,numSamplesPerVector,IndicesOfBVectors,positionX,
positionZ,minDataVectorRange):
    LastBVector = length(IndicesOfBVectors);
    AxialSize = size(BmodeArray,1);
    LateralSize = size(BmodeArray,2);
    SkipNVectors = LastBVector/LateralSize;
    AxialPosition = (1540000/2/40e6 *
cos(thetaRad(IndicesOfBVectors(1:SkipNVectors:LastBVector),1))
*((0:numSamplesPerVector-33) + minDataVectorRange)
+PositionZ(IndicesOfBVectors(1:SkipNVectors:LastBVector),transpose(ones(1,numSamplesPerVector-32)) - MinDepth));
    LateralPosition = (1540000/2/40e6 *
sin(thetaRad(IndicesOfBVectors(1:SkipNVectors:LastBVector),1)) *
((0:numSamplesPerVector-33) + minDataVectorRange) +
PositionX(IndicesOfBVectors(1:SkipNVectors:LastBVector),transpose(ones(1,numSamplesPerVector-32))));

```

4.1.5. Penampilan ke layar

Sinyal B-mode yang dihasilkan harus ditampilkan ke layar sebagai citra jadi. Proses ini dilakukan oleh fungsi berikut

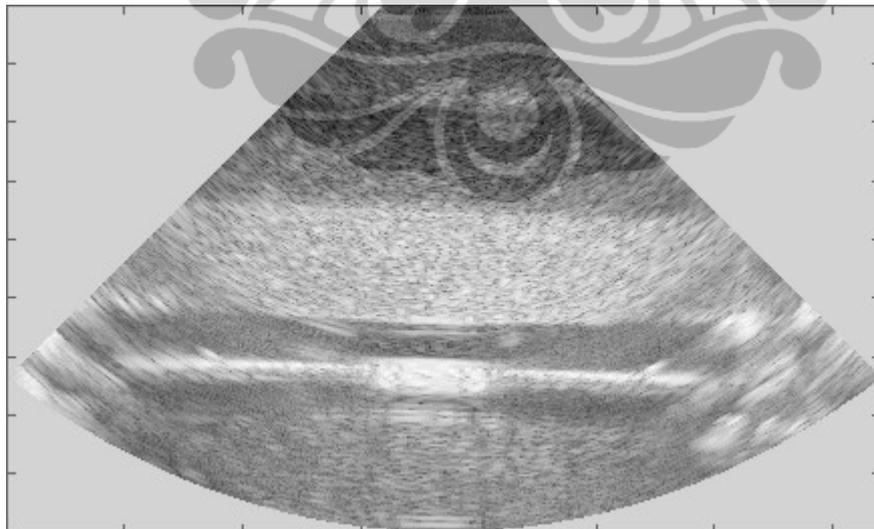
```
def showData(AxialPosition,LateralPosition,data,flip):
    figure()
    plt.ion()
    plt.gray()
    plt.pcolormesh(LateralPosition,AxialPosition, data)
    if flip:
        ax=plt.gca()
        ax.set_ylim(ax.get_ylim()[::-1])
```

Variabel flip digunakan untuk membalikkan sumbu vertikal citra sehingga angka nol berada di bagian atas gambar untuk menunjukkan kedalaman.

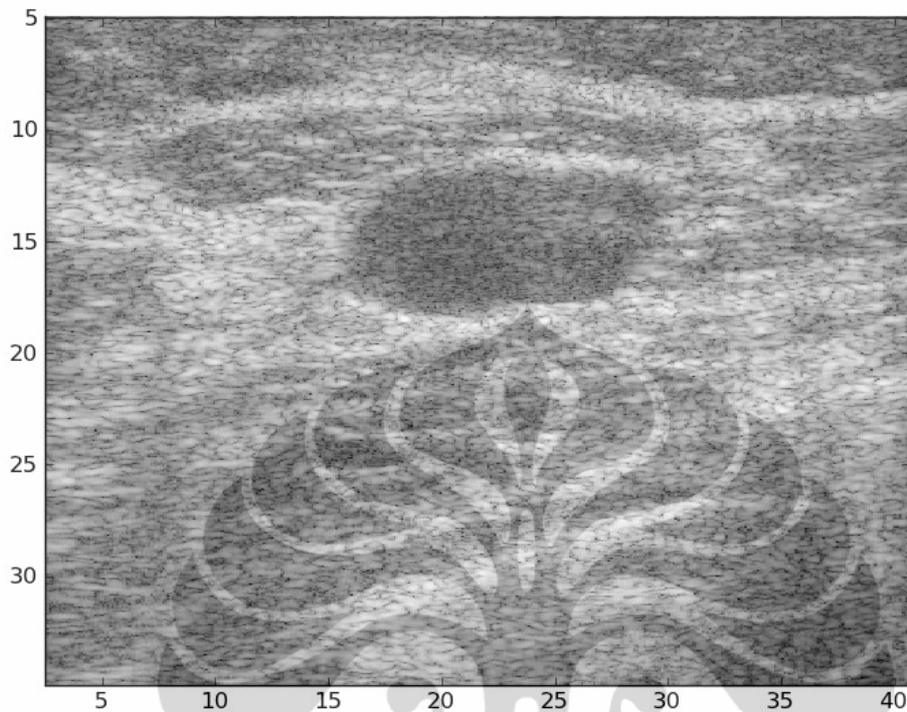
4.2. PERBANDINGAN KINERJA

Untuk memutuskan apakah implementasi ini dapat diterima, perlu dibandingkan kecepatan proses antara implementasi ini dengan implementasi sebelumnya.

Pada penulisan ini digunakan dua berkas data sinyal yang digunakan untuk pengujian. Keduanya ditampakkan pada gambar-gambar berikut.



Gambar 4.1. Hasil visualisasi dari berkas “Layers”



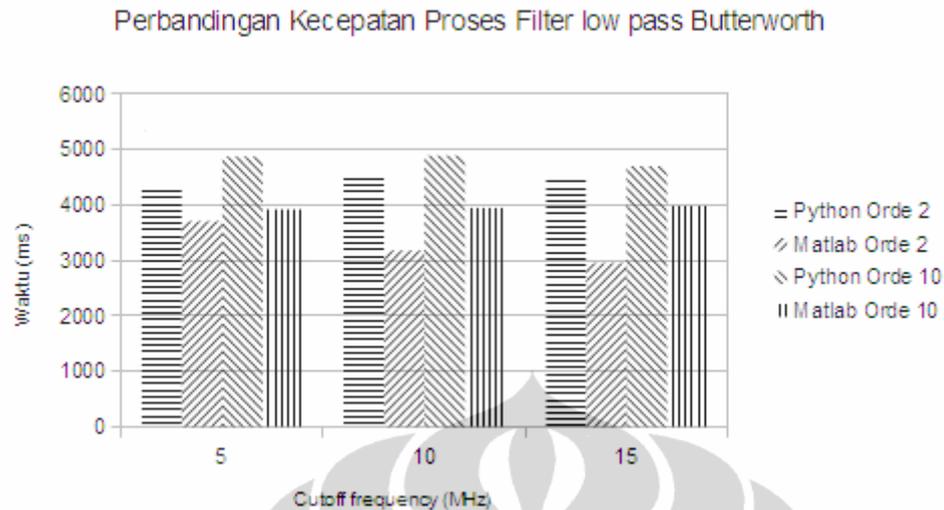
Gambar 4.2. Hasil visualisasi dari berkas “Body”

4.2.1. Perbandingan Kinerja Pengolahan berkas “Layers”

Di bawah ini tabel perbandingan antara kinerja implementasi filter low pass Butterworth pada Python dan Matlab.

Tabel 4.1. Perbandingan Kecepatan Pemrosesan Filter Low Pass Butterwoth antara Python dengan MATLAB (dalam ms)

Cutoff Freq	Python Orde 2	Matlab Orde 2	Python Orde 10	Matlab Orde 10
5	4359	3713	4865	3927
10	4578	3179	4891	3953
15	4531	2955	4703	3986



Gambar 4.3. Perbandingan Kecepatan Proses Filter Low Pass Butterworth

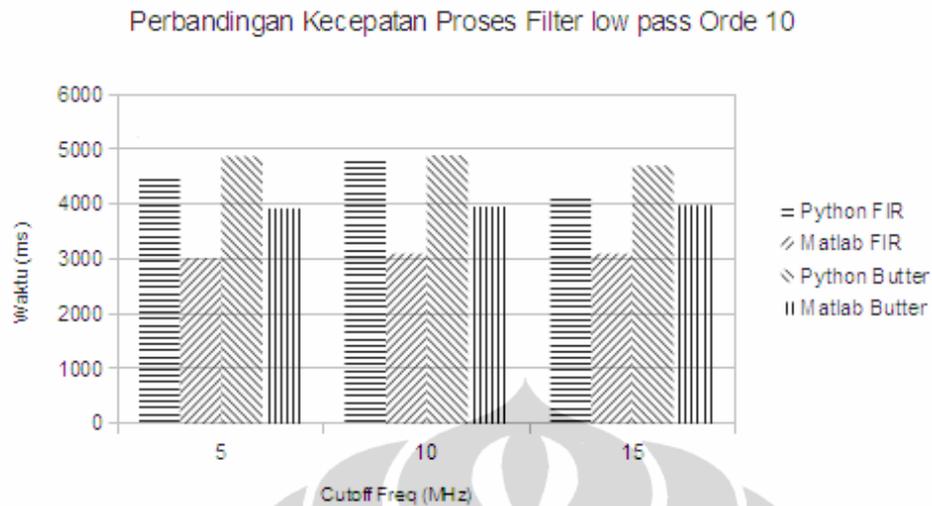
Tampak bahwa terjadi peningkatan waktu proses seiring peningkatan orde filter. Juga terjadi peningkatan waktu proses seiring dengan peningkatan *cutoff frequency* walau tidak signifikan. Juga terlihat bahwa kecepatan pemrosesan Python lebih lambat daripada Matlab tetapi peningkatan waktu prosesnya terhadap peningkatan orde filter terlihat lebih kecil.

Di samping itu dibandingkan juga kecepatan antara filter FIR dan Butterworth untuk melihat filter yang mana yang paling cepat mengolah data. Di bawah ini perbandingan kecepatan pemrosesan filter low pass pada orde 10.

Terlihat bahwa kecepatan proses FIR lebih baik daripada Butterworth, terutama pada frekuensi cutoff yang lebih tinggi.

Tabel 4.2. Perbandingan Kecepatan Pemrosesan Filter Low Pass Orde 10 antara Python dengan MATLAB (dalam ms)

Cutoff Freq	Python FIR	Matlab FIR	Python Butter	Matlab Butter
5	4563	3015	4875	3927
10	4891	3096	4891	3953
15	4203	3101	4703	3986



Gambar 4.4. Perbandingan Kecepatan Fiter low pass Orde 10

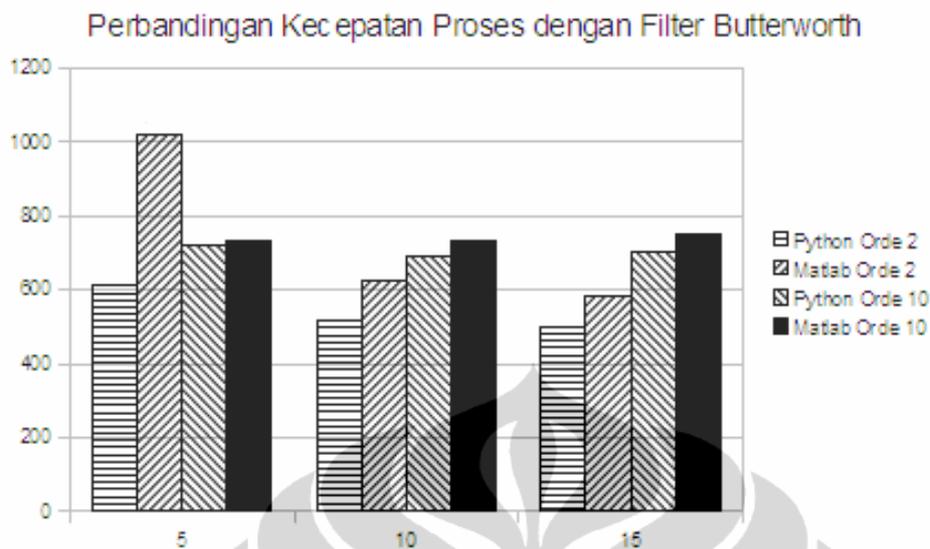
Seperti tampak pada lampiran, kecepatan proses tertinggi dari Python adalah sebesar 3640 ms, dibandingkan dengan kecepatan proses tertinggi dari MATLAB yang sebesar 2677 ms. Tetapi kecepatan Python dapat turun sampai 6391 ms seperti yang tampak pada penggunaan tapis lulus bawah Butterworth orde 20 dibandingkan dengan MATLAB yang hanya mencapai 4526 ms. Perbandingan waktu olah Python terhadap MATLAB bervariasi dari Python 44,56% lebih lambat dari MATLAB sampai Python 8.06% lebih lambat dari MATLAB.

4.2.2. Perbandingan Kinerja Pengolahan berkas “Body”

Di bawah ini tabel perbandingan antara kinerja implementasi filter low pass Butterworth pada Python dan Matlab.

Tabel 4.3. Perbandingan Kecepatan Pemrosesan Filter Low Pass Butterwoth antara Python dengan MATLAB (dalam ms)

Cutoff Freq	Python Orde 2	Matlab Orde 2	Python Orde 10	Matlab Orde 10
5	610	1019	719	733
10	516	627	688	735
15	500	585	703	751

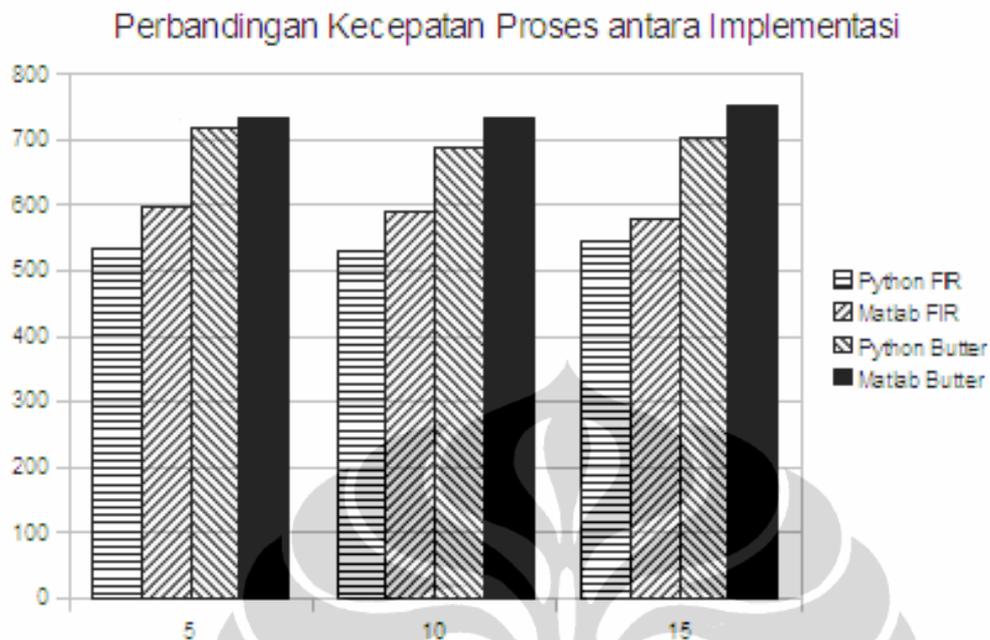


Gambar 4.5. Perbandingan Kecepatan Proses dengan Filter Butterworth

Tampak bahwa terjadi peningkatan waktu proses seiring peningkatan orde filter. Tetapi tidak terjadi peningkatan waktu proses seiring dengan peningkatan *cutoff frequency* seperti yang terjadi pada berkas “Layers” di atas. Juga terlihat bahwa dalam kondisi ini kecepatan pemrosesan Python lebih cepat daripada Matlab. Di samping itu dibandingkan juga kecepatan antara filter FIR dan Butterworth untuk melihat filter yang mana yang paling cepat mengolah data. Di bawah ini perbandingan kecepatan pemrosesan filter low pass pada orde 10.

Tabel 4.4. Perbandingan Kecepatan Pemrosesan Filter Low Pass Orde 10 antara Python dengan MATLAB (dalam ms)

Cutoff Freq	Python FIR	Matlab FIR	Python Butter	Matlab Butter
5	532	596	719	733
10	531	591	688	735
15	547	580	703	751



Gambar 4.6. Perbandingan Kecepatan Fiter low pass Orde 10

Terlihat bahwa kecepatan proses FIR lebih baik daripada Butterwoth. Dan juga terlihat bahwa dalam hal ini kecepatan proses Python lebih baik daripada Matlab. Seperti tampak pada lampiran, kecepatan proses tertinggi dari Python adalah sebesar 484 ms, sedangkan kecepatan proses tertinggi dari MATLAB sebesar 500 ms. Tetapi kecepatan Python dapat turun sampai 3875 ms seperti yang tampak pada penggunaan tapis lulus atas Butterwoth orde 20 dibandingkan dengan MATLAB yang hanya mencapai 1944 ms. Perbandingan waktu olah Python terhadap MATLAB bervariasi mulai dari Python 236% lebih cepat dari MATLAB sampai Python 79,46% lebih lambat dari MATLAB.

4.2.3. Analisa Kinerja

Pada kedua berkas tersebut didapatkan perbedaan waktu pengolahan yang jauh berbeda. Hal ini mungkin disebabkan oleh beberapa hal yaitu:

- Perbedaan ukuran data yang diolah
- Perbedaan jumlah *focal point* di mana berkas “Layers” memiliki 3 *focal point* sedangkan berkas “Body” memiliki 1 *focal point*.

Kedua ujicoba tidak memberikan kecepatan pengolahan yang memadai untuk dipakai dalam praktek karena kecepatan pengolahan tercepat dari data di atas adalah 500ms sehingga didapatkan kinerja yang tidak melebihi 2 fps (*frame per second*, gambar per detik)

4.3. ANALISIS VISUAL CITRA

Perbandingan kualitas visual citra dilakukan dengan wawancara dengan seorang dokter yang memahami kebutuhan pembacaan citra USG.

4.3.1. Metode Wawancara

Pada wawancara ini pada dokter diperlihatkan serangkaian citra hasil pemrosesan yang sudah dilakukan sebelumnya yang disimpan pada berkas (file) berformat PNG (*Portable Network Graphic*). Format ini dipilih karena merupakan format citra yang *lossless*. Berkas-berkas tersebut dikelompokkan menurut implementasi (MATLAB atau Python), metode penapisan (*high pass*, *low pass*, atau *band pass*), dan jenis tapis (FIR atau Butterworth) sehingga didapatkan nama folder seperti “matlab-highpass-butter”. Dalam satu folder terdapat beberapa berkas yang berbeda berdasarkan frekuensi *cutoff* dan orde tapisnya. Komentar dokter tersebut dicatat dan berkas yang dipandanginya terbaik dari tiap orde dipisahkan ke *subfolder* baru untuk kemudian dibandingkan kembali dan diambil yang terbaik. Hasil yang terbaik dari tiap *folder* tersebut diatas dikelompokkan dalam satu folder baru untuk dibandingkan.

Dalam perbandingan ini digunakan orde tapis 2, 10, dan 20. Pada tapis lulus bawah (*low pass filter*) dan lulus atas (*high pass filter*) digunakan frekuensi *cutoff* 5, 10, dan 15 MHz. Pada tapis lulus tengah (*band pass filter*) digunakan frekuensi *cutoff* bawah 5 MHz dan frekuensi *cutoff* atas 10 MHz.

4.3.2. Analisa Visual dari Hasil Pengolahan berkas “Body”

Seluruh tapis FIR orde 2 memberikan hasil yang hampir tidak dapat dibedakan dengan citra yang tidak ditapis. Hal ini berlaku baik pada implementasi MATLAB maupun implementasi Python ini.

Pada tapis lulus bawah , frekuensi *cutoff* 5 Mhz memberikan hasil yang buruk pada seluruh jenis filter kecuali FIR orde 2 seperti yang sudah diulas di atas. Sedangkan pada tapis lulus atas hasil yang buruk dihasilkan pada frekuensi *cutoff* 15 Mhz.

Tapis lulus atas Butterworth pada orde tinggi memberikan hasil pencitraan yang terlalu gelap sehingga mengakibatkan adanya jaringan yang tidak dapat dibedakan dari rongga tubuh (*pocket*).

Tapis lulus atas FIR memberikan hasil yang optimal pada orde 10 dan pada frekuensi 5 MHz. Penapisan orde 10 frekuensi 10 MHz memberikan hasil yang terlalu gelap dan tidak dapat diterima.

Pada tapis lulus bawah Butterworth, terdapat perbedaan kesimpulan antara tapis MATLAB dengan Python. Tapis MATLAB orde 20 memberikan hasil yang tidak dapat diterima sedangkan tapis Python orde 20 memberikan hasil yang lebih optimal daripada orde 10. Kedua implementasi memberikan hasil yang terbaik pada frekuensi 15 MHz.

Pada tapis lulus bawah FIR didapatkan hasil yang optimal pada frekuensi 15 MHz. Tetapi juga terdapat perbedaan kesimpulan antara implemetasi MATLAB dengan Python. Pada MATLAB tidak nampak perbedaan yang signifikan antara orde 2, 10, dan 20 sedangkan pada Python didapatkan hasil yang terbaik pada orde 2.

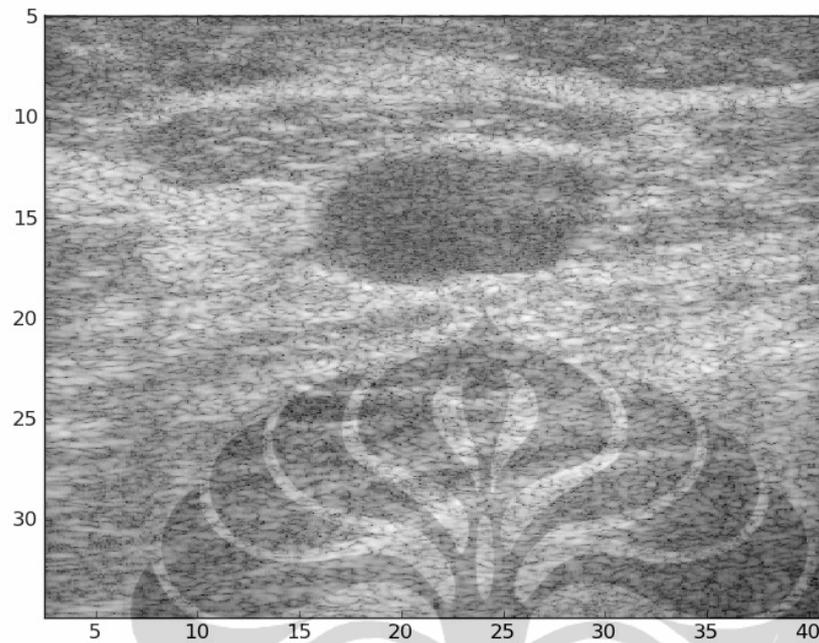
Pada tapis lulus tengah Butterworth didapatkan hasil terbaik pada orde tapis 2. Sedangkan pada tapis FIR didapatkan hasil terbaik pada orde tapis 10.

Dari seluruh hasil terbaik implementasi Python didapatkan hasil terbaik adalah hasil penapisan lulus tengah FIR. Sedangkan dari seluruh hasil implementasi MATLAB tidak didapatkan perbedaan signifikan.

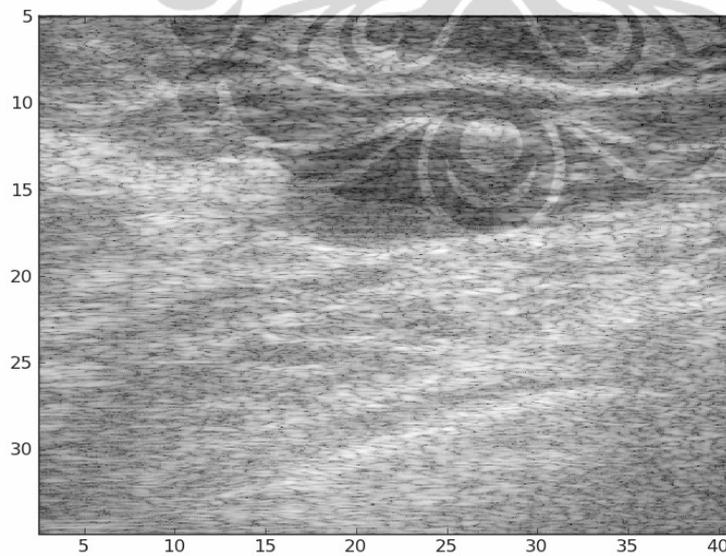
Dalam proses perbandingan tersebut di atas, didapatkan catatan bahwa ada kekurangan dari hasil proses menggunakan Python, yaitu :

1. Artefak berupa garis-garis horizontal yang mengaburkan citra
2. Ketidaktegasan dinding kanan pocket yang terdapat di tengah gambar.

Perbedaan tersebut nampak pada gambar-gambar berikut :



Gambar 4.7. Implementasi MATLAB tanpa Tapis



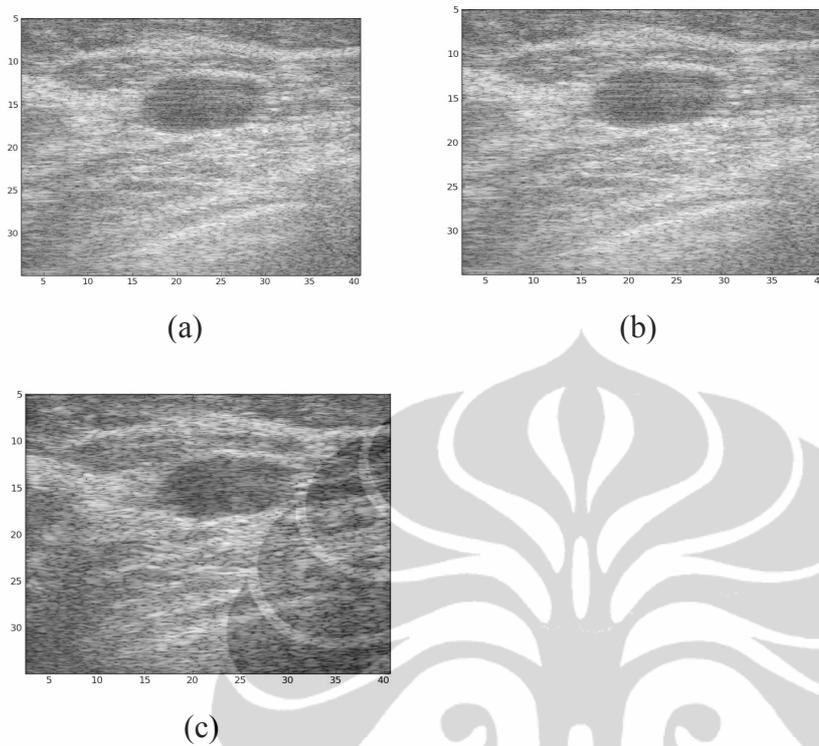
Gambar 4.8. Implementasi Python tanpa Tapis

Artefak tersebut dapat dihilangkan dengan penapisan. Gambar berikut menampilkan hasil penapisan lulus tengah FIR dengan frekuensi *cutoff* bawah 5 MHz dan *cutoff* atas 10 MHz.



Gambar 4.9. Hasil Penapisan Lulus Tengah Orde 10 yang Menghilangkan Artefak

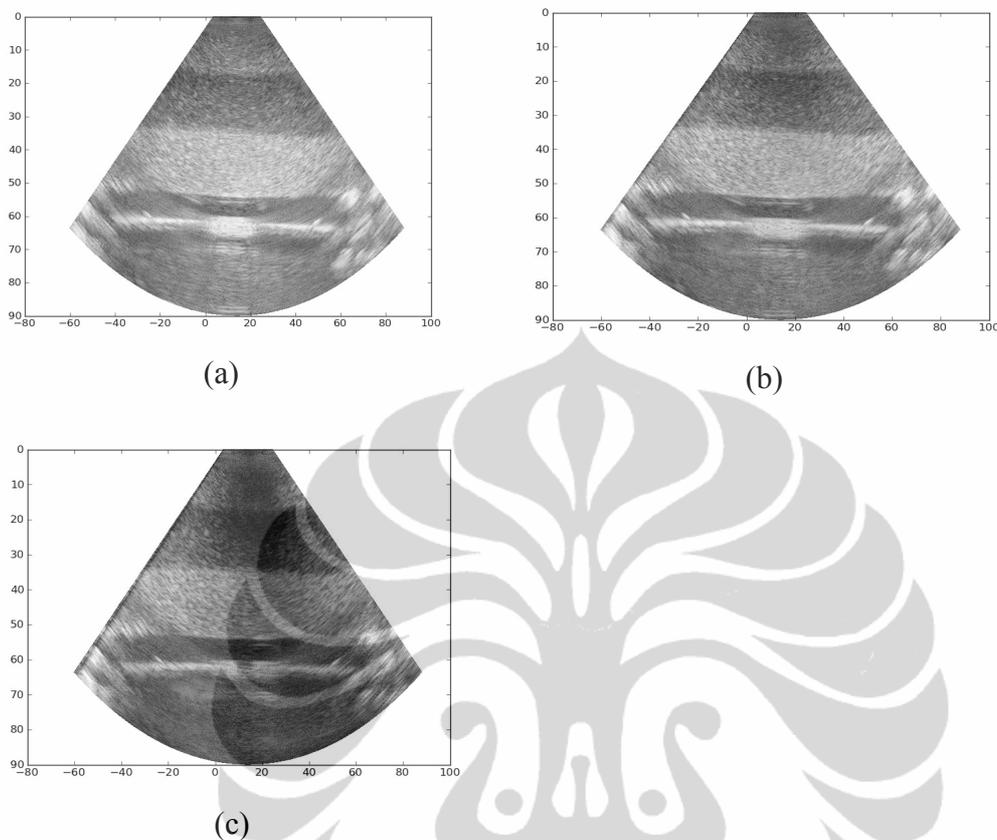
Di bawah ini adalah contoh citra yang dihasilkan masing-masing adalah citra yang diproses tanpa penapisan (*filtering*), dengan band pass filter FIR orde 2 dengan frekuensi 5 MHz dan 10 MHz, dan dengan band pass filter Butterworth frekuensi 5 dan 10 MHz.



Gambar 4.10. Perbandingan citra antara tanpa filter (a), FIR (b), dan Butterworth (c)

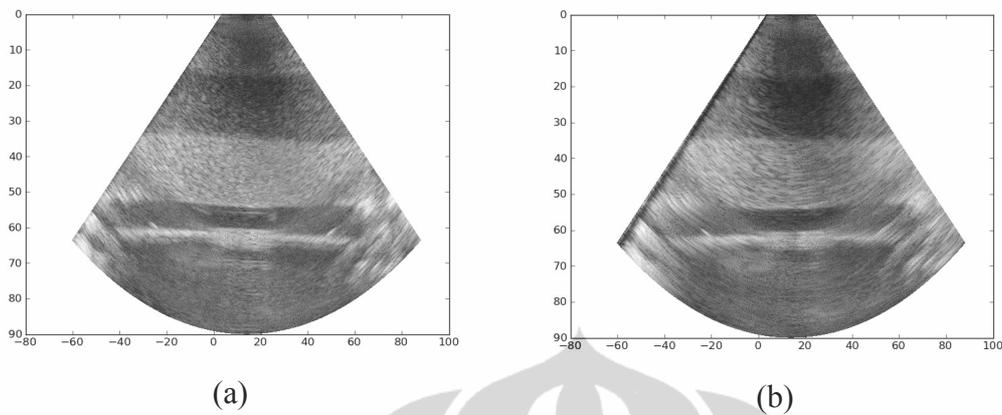
4.3.3. Analisa Visual Hasil Pengolahan dari berkas “Layers”

Selain menganalisa hasil pengolahan berkas “Body”, dibandingkan juga hasil pengolahan dari berkas “Layers”. Berikut ini akan dibandingkan hasil penapisan lulus tengah FIR dengan frekuensi *cutoff* bawah 5 MHz dan *cutoff* atas 10 MHz. Pada gambar-gambar berikut ditampakkan hasil penapisan FIR dengan orde 2,10, dan 20. Tampak bahwa peningkatan orde memberikan hasil kontras yang lebih tajam.



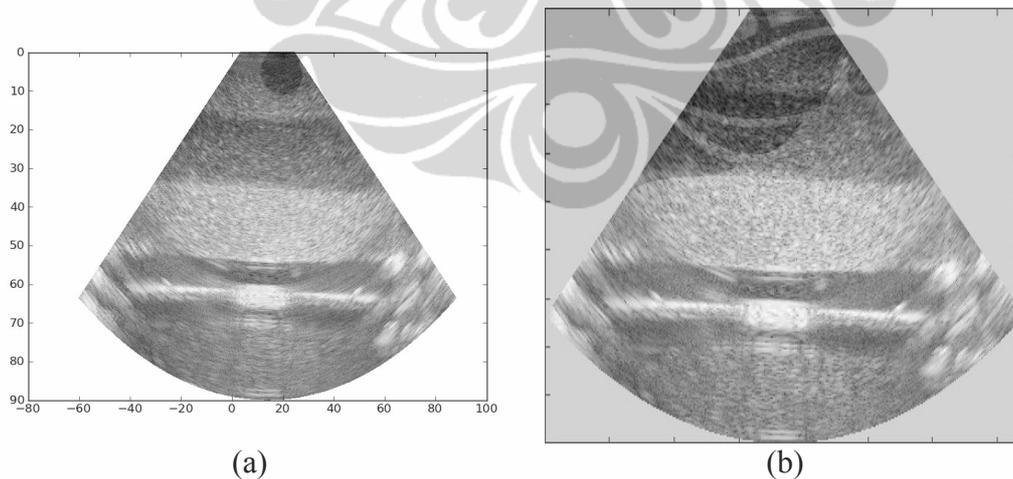
Gambar 4.11. Hasil penapisan FIR dengan orde 2 (a),10 (b), dan 20 (c)

Dilakukan juga perbandingan visual dari implementasi menggunakan tapis Butterworth. Pada gambar berikut ditampakkan hasil penapisan Butterworth dengan orde 2 dan 10. Juga tampak peningkatan ketajaman kontras seiring peningkatan orde.



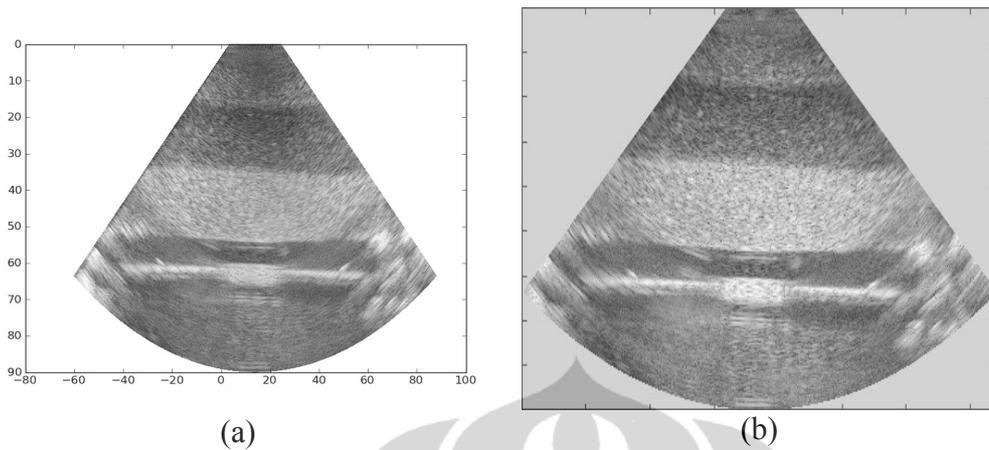
Gambar 4.12. Hasil penapisan Butterworth dengan orde 2 (a) dan 10 (b)

Berikutnya dibandingkan hasil tapis lulus tengah antara implementasi ini dengan MATLAB. Pada analisa ini digunakan hasil pencitraan dengan penapisan lulus tengah dengan frekuensi cutoff bawah 5 MHz dan cutoff atas 10 MHz. Pada gambar berikut ditampakkan hasil penapisan FIR dengan orde 2 berturut-turut hasil dari implementasi Python dengan hasil implementasi MATLAB. Tidak tampak adanya perbedaan kualitas gambar pada masing-masing pasangan.



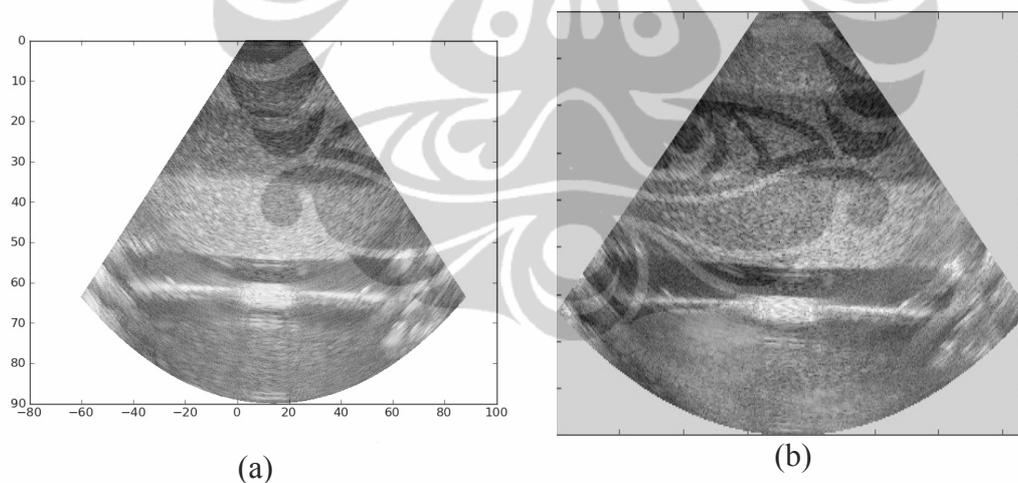
Gambar 4.13. Hasil Penapisan FIR Orde 2 untuk Python (a) dan MATLAB (b)

Pada gambar berikut ditampakkan hasil penapisan FIR dengan orde 10, berturut-turut hasil dari implementasi Python dengan hasil implementasi MATLAB. Juga tidak tampak adanya perbedaan kualitas gambar pada masing-masing pasangan.



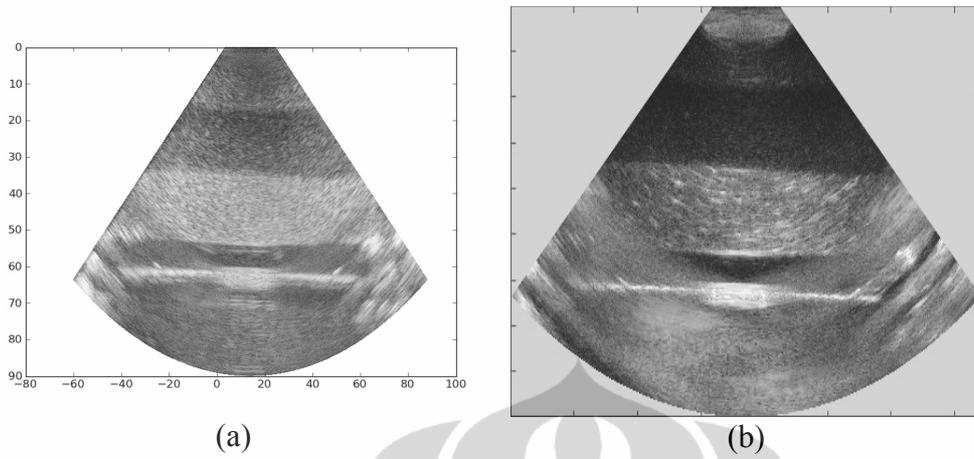
Gambar 4.14. Hasil Penapisan FIR Orde 10 untuk Python (a) dan MATLAB (b)

Pada gambar berikut ditampakkan hasil penapisan Butterworth dengan orde 2, berturut-turut hasil dari implementasi Python dengan hasil implementasi MATLAB. Terlihat bahwa implementasi MATLAB menghasilkan citra yang lebih tajam daripada implementasi Python.

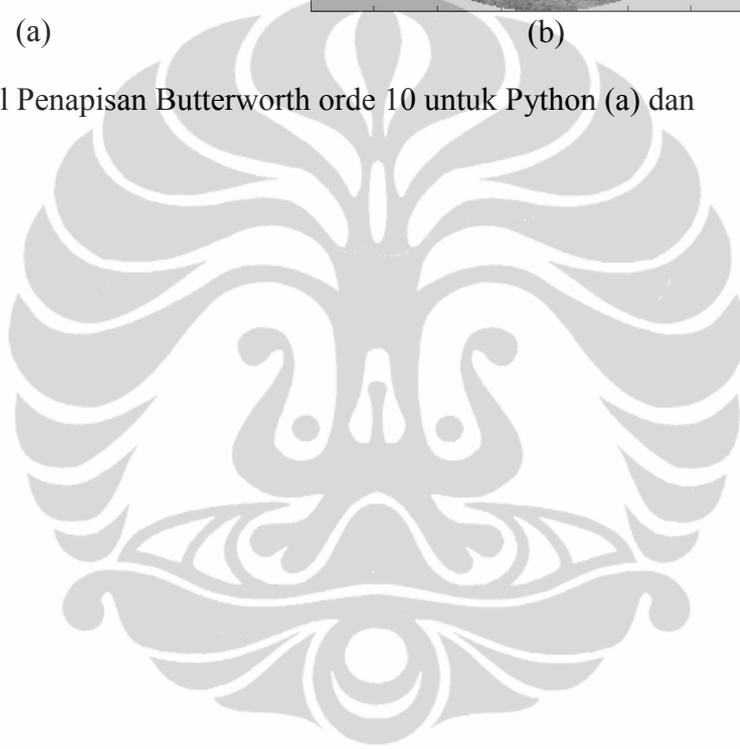


Gambar 4.15. Hasil Penapisan Butterworth orde 2 untuk Python (a) dan MATLAB (b)

Pada gambar berikut ditampakkan hasil penapisan Butterworth dengan orde 10, berturut-turut hasil dari implementasi Python dengan hasil implementasi MATLAB. Terlihat bahwa implementasi MATLAB tampak terlalu gelap sehingga implementasi Python lebih dapat dibaca.



Gambar 4.16. Hasil Penapisan Butterworth orde 10 untuk Python (a) dan MATLAB (b)



BAB 5

KESIMPULAN

Pada tulisan ini telah diuraikan implementasi pustaka open source pengolah data USG dalam bahasa Python.

Didapatkan hasil sebagai berikut:

1. Implementasi ini menghasilkan kualitas citra visual yang dapat diterima walau memiliki kelemahan berupa artefak yang tidak tampil pada implementasi MATLAB.
2. Kecepatan pengolahan dengan filter FIR lebih cepat 0,4% - 536,3% daripada dengan filter Butterworth.
3. Kecepatan pengolahan dengan Python ini bervariasi dari 76,49% lebih lambat sampai 236,33% lebih cepat dari MATLAB.
4. Waktu pengolahan data yang dibutuhkan pada kedua implementasi tersebut 500ms – 6391ms yang masih tidak memadai untuk menghasilkan kinerja yang dibutuhkan untuk penggunaan langsung.
5. Dibutuhkan pengembangan lebih lanjut untuk mempercepat pengolahan data.

DAFTAR ACUAN

- [1] Agung Alfiansyah, Workshop on Medical Image Processing, Submit, 11/24/2008, diunduh 20 februari 2010 dari Website dikti.org. http://www.dikti.org/?quicktabs_1=2#quicktabs-1.
- [2] Kalamullah Ramli, et.al. Pengembangan Perangkat Lunak Untuk Akuisisi data, Visualisasi dan Analisis Citra Ultrasonografi Berbasis Open Source, Bahan Presentasi proposal penelitian kerjasama UI, BPPT dan Edwar Technology, 2009, pp 3.
- [3] Halliday David and Robert Resnick. Physics. Third Edition. John Wiley & Sons, Inc, 1978. Translated in Indonesian Language by Pantur silaban and Erwin Sucipto,. Fisika. Jakarta : Erlangga, 1992. pp 656-693.
- [4] William D. O'Brien. (2007). Review Ultrasound – biophysics mechanisms Journal Science Direct, Progress in Biophysics and Molecular Biology. 93 (2007) 214–216.
- [5] Pauly H, Schwan P. Mecanism of absorbtio of ultrasound in liver tissues. J Acoust Soc Am 1971;2: pp 692-699.
- [6] Parker KJ. Attenuation measurement uncertainties caused by spekle statistics. J Acoust Soc Am 1986;80:pp 727-734.
- [7] Ultrasonic Transducer Technical Notes. Olympus NDT (2006). Pp 40-43
- [8] Welch Allyn, Physician Office Ultrasonic Imaging. Thesis, Syracuse University Coolege of Law Technology Transfer Reseach Center, 2004. pp 15-16.
- [9] Douglas C Giancoli, Physics (Principles With Application). Fifth Edition. Prentice-Hall International, Inc, 1998. Translated in Indonesian Language by Yuhilza Hanum. Fisika. Edisi 5. Jakarta : Erlangga, 2001. pp 407-444.
- [10] John R. Cameron, James G. Skofronick, Medical Physics, New York: John Wiley & Sons Inc, 1978, pp 253-287.
- [11] Adarsh K. Modeling Ultrasonic Fields In Complex Geometries, Project Report, Department Of Mechanical Engineering, Indian Institute Of Technology Madras, Chennai, 2005. pp 12.
- [12] Jerrold T Bushberg, et.al., The Essential Physics of Medical Imaging, Chapter 16 : Ultrasound (2nd ed.). Philadelphia: Lippincott Williams & Wilkins, 2002.

- [13] Xiaoyang Cheng, Minimally Invasive Capacitive Micromachined Ultrasonic Transducers Array For Biomedical Applications, Dissertation, The University Of New Mexico, Albuquerque, New Mexico, 2008, pp 66-73.
- [14] Peter N Burns, Introduction to The Physical Principles of Ultrasound Imaging and Doppler. Journal Medical Biophysics-MBP1007/1008, 2005, pp 5.
- [15] Doppler Ultrasound , Bahan presentasi, diunduh pada 10 April 2010 dari : <http://fygo.dk/files/ukursus/Ultrasound%20Doppler.pdf> .
- [16] Alejandro Frangi, Introduction to Biomedical Imaging, Bahan presentasi Computational Imaging Lab, Department of Information & Communication Technology, pp 14, diunduh, 5 mei 2010 dari web Pompeu Fabra University, <http://www.cilab.upf.edu>.
- [17] Berkeley Software Distribution, June, 13 2011.
<http://en.wikipedia.org/wiki/BSD>
- [18] Matplotlib License, June, 13 2011.
<http://matplotlib.sourceforge.net/users/license.html>
- [19] wxWidgets General FAQ, June, 13 2011.
<http://www.wxwidgets.org/docs/faqgen.htm>
- [20] wxWidgets License, June, 13 2011.
<http://www.wxwidgets.org/about/newlicen.htm>
- [21] Bendersky, Eli. Matplotlib with wxPython GUIs. June, 13 2011.
<http://eli.thegreenplace.net/2008/08/01/matplotlib-with-wxpython-guis/>
- [22] Qt (Framework), June, 13 2011.
http://en.wikipedia.org/wiki/Qt_%28framework%29
- [23] Bendersky, Eli. Matplotlib with PyQt GUIs. June, 13 2011.
<http://eli.thegreenplace.net/2009/01/20/matplotlib-with-pyqt-guis/>
- [24] Imardi, Syahrul, Pengembangan Perangkat Lunak untuk Visualisasi dan Analisis Citra Ultrasonografi Berbasis Open Source, Universitas Indonesia, 2010
- [25] Matplotlib Screenshots. June, 13 2011.
<http://matplotlib.github.com/users/screenshots.html>

LAMPIRAN

LAMPIRAN 1

KECEPATAN PROSES BODY.MAT

Filter	Orde	F Pancung (MHz)	Waktu Proses (ms)		Ratio (P/M)	Speedup (%)
			Python	Matlab		
Low Pass FIR	2	5	500	848	0.590	69.60%
		10	484	522	0.927	7.85%
		15	484	520	0.931	7.44%
	10	5	532	596	0.893	12.03%
		10	531	591	0.898	11.30%
		15	547	580	0.943	6.03%
	20	5	594	598	0.993	0.67%
		10	609	592	1.029	-2.79%
		15	594	597	0.995	0.51%
Low Pass Butter	2	5	610	1019	0.599	67.05%
		10	516	627	0.823	21.51%
		15	500	585	0.855	17.00%
	10	5	719	733	0.981	1.95%
		10	688	735	0.936	6.83%
		15	703	751	0.936	6.83%
	20	5	1672	804	2.080	-51.91%
		10	1688	797	2.118	-52.78%
		15	1640	787	2.084	-52.01%
High Pass FIR	2	5	484	1234	0.392	154.96%
		10	515	567	0.908	10.10%
		15	516	500	1.032	-3.10%
	10	5	547	557	0.982	1.83%
		10	594	598	0.993	0.67%
		15	547	538	1.017	-1.65%
	20	5	594	587	1.012	-1.18%
		10	609	604	1.008	-0.82%
		15	594	597	0.995	0.51%
High Pass Butter	2	5	578	1944	0.297	236.33%
		10	562	601	0.935	6.94%
		15	578	603	0.959	4.33%
	10	5	938	807	1.162	-13.97%
		10	969	727	1.333	-24.97%
		15	937	720	1.301	-23.16%
	20	5	2765	806	3.431	-70.85%
		10	2735	801	3.414	-70.71%
		15	3875	796	4.868	-79.46%
Min					0.297	-79.46%
Max					4.868	236.33%

KECEPATAN PROSES BODY.MAT

Filter	Ord	Python				Matlab			
		Avg	StdDev	Max	Min	Avg	StdDev	Max	Min
Low Pass FIR	2	541.7	48.3	609	484	604.9	96.4	848	520
		489.3	9.2	500	484	630.0	188.8	848	520
	10	536.7	9.0	547	531	589.0	8.2	596	580
	20	599.0	8.7	609	594	595.7	3.2	598	592
Low Pass Butter	2	970.7	527.7	1688	500	759.8	122.9	1019	585
		542.0	59.4	610	500	743.7	239.4	1019	585
	10	703.3	15.5	719	688	739.7	9.9	751	733
	20	1666.7	24.4	1688	1640	796.0	8.5	804	787
High Pass FIR	2	555.6	44.4	609	484	642.4	224.4	1234	500
		505.0	18.2	516	484	767.0	405.8	1234	500
	10	562.7	27.1	594	547	564.3	30.7	598	538
	20	599.0	8.7	609	594	596.0	8.5	604	587
High Pass Butter	2	1548.6	1236.9	3875	562	867.2	412.1	1944	601
		572.7	9.2	578	562	1049.3	774.8	1944	601
	10	948.0	18.2	969	937	751.3	48.3	807	720
	20	3125.0	649.7	3875	2735	801.0	5.0	806	796
Keseluruhan		904.1	766.3	3875	484	718.6	258.5	1944	500
FIR				609	484			1234	500
Butterworth				3875	500			1944	585
FIR Speedup				536.3%	3.3%			57.5%	17.0%

LAMPIRAN 2

KECEPATAN PROSES LAYERS.MAT

Filter	Ord	F Pancung (MHz)	Waktu Proses (ms)		Ratio (P/M)	Speedup (%)
			Python	Matlab		
Low Pass FIR	2	5	4344	3445	1.261	-20.70%
		10	4000	2757	1.451	-31.08%
		15	4500	2677	1.681	-40.51%
	10	5	4563	3015	1.513	-33.93%
		10	4391	3096	1.418	-29.49%
		15	4203	3101	1.355	-26.22%
	20	5	4640	3124	1.485	-32.67%
		10	4844	3102	1.562	-35.96%
		15	4891	3150	1.553	-35.60%
Low Pass Butter	2	5	4359	3713	1.174	-14.82%
		10	4578	3179	1.440	-30.56%
		15	4531	2955	1.533	-34.78%
	10	5	4875	3927	1.241	-19.45%
		10	4891	3953	1.237	-19.18%
		15	4703	3986	1.180	-15.25%
	20	5	6391	4238	1.508	-33.69%
		10	6125	4293	1.427	-29.91%
		15	5484	4526	1.212	-17.47%
High Pass FIR	2	5	4750	3525	1.348	-25.79%
		10	4812	2794	1.722	-41.94%
		15	3640	2736	1.330	-24.84%
	10	5	3985	3059	1.303	-23.24%
		10	3906	3057	1.278	-21.74%
		15	3891	3025	1.286	-22.26%
	20	5	5703	3162	1.804	-44.56%
		10	4234	3182	1.331	-24.85%
		15	4203	3174	1.324	-24.48%
High Pass Butter	2	5	3687	3390	1.088	-8.06%
		10	3656	2744	1.332	-24.95%
		15	3656	2711	1.349	-25.85%
	10	5	4188	3593	1.166	-14.21%
		10	4235	3562	1.189	-15.89%
		15	4218	3575	1.180	-15.24%
	20	5	6312	3950	1.598	-37.42%
		10	6328	3931	1.610	-37.88%
		15	6281	3947	1.591	-37.16%
Min					1.088	-44.56%
Max					1.804	-8.06%

KECEPATAN PROSES LAYERS.MAT

Filter	Ord	Python				Matlab			
		Avg	StdDev	Max	Min	Avg	StdDev	Max	Min
Low Pass FIR	2	4486.2	288.9	4891	4000	3051.9	225.0	3445	2677
		4281.3	255.8	4500	4000	2959.7	422.2	3445	2677
	10	4385.7	180.1	4563	4203	3070.7	48.3	3101	3015
	20	4791.7	133.4	4891	4640	3125.3	24.0	3150	3102
Low Pass Butter	2	5104.1	729.9	6391	4359	3863.3	513.2	4526	2955
		4489.3	115.3	4578	4359	3282.3	389.4	3713	2955
	10	4823.0	104.2	4891	4703	3955.3	29.6	3986	3927
	20	6000.0	466.2	6391	5484	4352.3	152.9	4526	4238
High Pass FIR	2	4347.1	640.7	5703	3640	3079.3	231.3	3525	2736
		4400.7	659.5	4812	3640	3018.3	439.7	3525	2736
	10	3927.3	50.5	3985	3891	3047.0	19.1	3059	3025
	20	4713.3	857.2	5703	4203	3172.7	10.1	3182	3162
High Pass Butter	2	4729.0	1207.1	6328	3656	3489.2	475.8	3950	2711
		3666.3	17.9	3687	3656	2948.3	382.9	3390	2711
	10	4213.7	23.8	4235	4188	3576.7	15.6	3593	3562
	20	6307.0	23.9	6328	6281	3942.7	10.2	3950	3931
Keseluruhan		4666.6	807.8	6391	3640	3370.9	499.7	4526	2677
FIR				5703	3640			3525	2677
Butterworth				6391	3656			4526	2711
FIR Speedup				12.1%	0.4%			28.4%	1.3%

LAMPIRAN 3

**HASIL WAWANCARA TENTANG PERBANDINGAN
KUALITAS CITRA HASIL PENGOLAHAN USG**

Matlab High Pass Butterworth

Orde dan Frekuensi Cutoff	Komentar
Orde 2, 15 MHz	Terlalu gelap sehingga jaringan di sudut kiri atas tidak dapat dibedakan dengan pocket di tengah
Orde 10, 15 MHz	Tidak dapat diterima
Orde 20, 15 MHz	Tidak dapat diterima
Antara Orde 2, 5 MHz dengan Orde 2, 10 MHz	5 MHz lebih baik karena 10 MHz terlalu gelap
Antara Orde 10, 5 MHz dengan Orde 10, 10 MHz	5 MHz lebih baik karena 10 MHz terlalu banyak noise
Antara Orde 20, 5 MHz dengan Orde 20, 10 MHz	5 MHz lebih baik karena 10 MHz terlalu banyak noise
Antara orde 2, 10, dan 20, frekuensi 5 MHz	Terbaik adalah orde 2 5 MHz

Matlab High Pass FIR

Orde dan Frekuensi Cutoff	Komentar
Orde 2 seluruh frekuensi	Tidak dapat dibedakan dengan yang tidak ditapis
Orde 10 15 MHz	Tidak dapat diterima
Orde 20 15 MHz	Tidak dapat diterima
Antara orde 10, 5 MHz dengan orde 10, 10 MHz	5 MHz karena 10 MHz terlalu gelap
Antara orde 20, 5 MHz dengan orde 20, 10 MHz	5 MHz karena 10 MHz terlalu banyak yang jadi gelap walau ada kontras yang lebih terlihat (khususnya di sisi kiri bawah)
Antara orde 20, 5 MHz dengan orde 10, 10 MHz	Orde 20, 5 MHz karena orde 10, 10 MHz terlalu banyak yang jadi gelap walau ada kontras yang lebih terlihat (khususnya di sisi kiri bawah)
Antara orde 2,10, 20 frekuensi 5 MHz	Orde 20 lebih sedikit noisenya.

Python High Pass Butterworth

Orde dan Frekuensi Cutoff	Komentar
-	Seluruh hasil penapisan buruk. Yang paling sedikit keburukannya adalah orde 2 5 MHz.

Python High Pass FIR

Orde dan Frekuensi Cutoff	Komentar
Orde 2, semua frekuensi	Tidak dapat dibedakan dari yang tidak ditapis
Orde 10, 10 MHz dan 15 MHz	Tidak dapat diterima

Orde dan Frekuensi Cutoff	Komentar
Orde 20, 10 MHz dan 15 MHz	Tidak dapat diterima
Orde 2, 10, 20 frekuensi 5 MHz	Orde 10 paling baik

Matlab Low Pass Butterworth

Orde dan Frekuensi Cutoff	Komentar
Orde 2	Frekuensi 10 dan 15 MHz sangat mirip. Keduanya lebih baik dari 5 MHz.
Orde 20	Keseluruhannya tidak dapat diterima. 10 MHz masih lebih baik tetapi ada noise vertikal di atas
Orde 10, 15 MHz	Tidak dapat diterima
Antara orde 10 5 MHz dengan 10 MHz	10 MHz lebih baik karena 5 MHz masih terlalu banyak noise.

Matlab Low Pass FIR

Orde dan Frekuensi Cutoff	Komentar
Orde 2 seluruh frekuensi	Tidak dapat dibedakan
Orde 10, 10 MHz dan 15 MHz	Tidak banyak perbedaan. Keduanya lebih baik dari 5 MHz
Orde 20, 10 MHz dan 15 MHz	Tidak banyak perbedaan. Keduanya lebih baik dari 5 MHz
Antara orde 10, 10 MHz dan 15 MHz, orde 20 10 MHz dan 15 MHz	Tidak ada perbedaan signifikan

Python Low Pass Butterworth

Orde dan Frekuensi Cutoff	Komentar
Frekuensi 5 MHz seluruh orde	Tidak dapat diterima.
Antara orde 2 10 MHz dengan 15 MHz	Frekuensi 15 MHz lebih baik tetapi tetap tidak bagus karena ada artefak di pocket yang seharusnya gelap.
Antara orde 10 10 MHz dengan 15MHz	Frekuensi 15 MHz lebih baik, tetapi artefaknya mengganggu
Antara orde 20 10 MHz dengan 15MHz	Frekuensi 15 MHz lebih baik, tetapi artefaknya mengganggu
Antara orde 10 15 MHz dengan orde 20 15 MHz	Orde 20 15 MHz lebih baik, tetapi artefaknya mengganggu

Python Low Pass FIR

Orde dan Frekuensi Cutoff	Komentar
Orde 2	tidak dapat dibedakan

Antara orde 10 5 MHz, 10 MHz, 15 MHz	15 MHz terbaik tetapi artefaknya mengganggu
Antara orde 20 5 MHz, 10 MHz, 15 MHz	15 MHz terbaik tetapi artefaknya mengganggu
Antara orde 2, 10, dan 20 frekuensi 15 MHz	Orde 2 terbaik tetapi artefaknya mengganggu

Band Pass filter dengan frekuensi cutoff 5 MHz dan 10 MHz

Implementasi	Komentar
Matlab Butterworth	Orde 2 terbaik
Matlab FIR	Orde 10 terbaik tetapi tidak jauh berbeda dengan orde 2
Python Butterworth	Terbaik orde 2. Orde 10 dan 20 tidak dapat diterima
Python FIR	Terbaik orde 10. Tidak ada artefak, kontrasnya cukup. Orde 2 ada artefak yang mengganggu

Perbandingan dari hasil-hasil terbaik antar implementasi

Implementasi	Komentar
High Pass Butterworth	Matlab lebih jelas border dari pocket gelap tersebut
High Pass FIR	Matlab lebih jelas border dari pocket gelap tersebut
Low Pass Butterworth	Matlab lebih baik, artefak Python mengganggu
Low Pass FIR	Python lebih baik tetapi artefaknya mengganggu
Band Pass Butterworth	Matlab lebih bagus tetapi masih banyak noise
Band Pass FIR	Matlab lebih bagus tetapi Python dapat diterima

Kesimpulan akhir

Citra hasil bandpass FIR dari implementasi Python paling dapat diterima. perbedaan hasil terbaik Matlab tidak signifikan.

Disetujui



dr. Noha Roshadiansyah