



**SISTEM CERDAS DETEKSI SUARA UNTUK  
PENGKLASIFIKASIAN PENYAKIT JANTUNG  
MENGUNAKAN JARINGAN SARAF TIRUAN**

**LAPORAN  
PROYEK AKHIR**

**Diajukan Kepada Fakultas Teknik Universitas Negeri Yogyakarta  
Untuk Memenuhi Sebagian Persyaratan  
Guna Memperoleh Gelar Ahli Madya  
Program Studi Teknik Elektronika**

**OLEH:  
MUHAMMAD NADHIL SIDQI  
NIM. 14507134031**

**PROGRAM STUDI TEKNIK ELEKTRONIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
2018**

**LEMBAR PERSETUJUAN  
PROYEK AKHIR**

**SISTEM CERDAS DETEKSI SUARA UNTUK PENGKLASIFIKASIAN  
PENYAKIT JANTUNG MENGGUNAKAN JARINGAN SARAF TIRUAN**

Oleh:

**MUHAMMAD NADHIL SIDQI**

**14507134031**

Telah diperiksa dan disetujui oleh pembimbing  
Untuk diuji

Yogyakarta, 21 Desember 2018

Mengetahui,  
Kaprodí Teknik Elektronika

Menyetujui,  
Pembimbing Proyek Akhir



Dr. Dra. Sri Waluyanti, M.Pd.  
NIP. 19581218 198603 2 001



Dessy Irmawati, S.T., M.T.  
NIP. 19791214 201012 2 002

**LEMBAR PENGESAHAN  
PROYEK AKHIR**

**SISTEM CERDAS DETEKSI SUARA UNTUK PENGKLASIFIKASIAN  
PENYAKIT JANTUNG MENGGUNAKAN JARINGAN SARAF TIRUAN**

Dipersiapkan dan Disusun Oleh:  
**MUHAMMAD NADHIL SIDQI**  
**14507134031**

Telah dipertahankan didepan Dewan Penguji Proyek Akhir  
FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA

Pada tanggal  
Dan Dinyatakan Telah Memenuhi Syarat Guna Memperoleh Gelar  
Ahli Madya Teknik

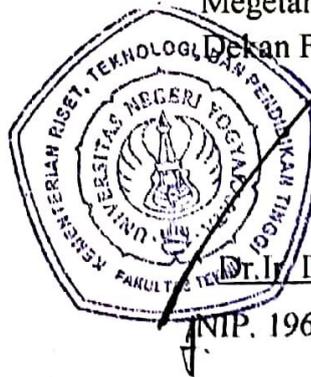
**SUSUNAN DEWAN PENGUJI**

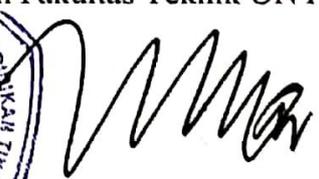
No	Jabatan	Nama Lengkap	Tanda Tangan	Tanggal
1.	Ketua Penguji	Dessy Irmawati S.T., M.T		25/01-2019
2.	Sekretaris Penguji	Bekti Wulandari S.Pd.T., M.Pd.		17/01-2019
3.	Penguji Utama	Dr. Fatchul Arifin M.T.		25/01-2019

Yogyakarta, 7 Januari 2019

Megetahui,

Dekan Fakultas Teknik UNY



  
Dr. Ir. Drs. Widarto, M.Pd.

NIP. 19631230 198812 1 001

## LEMBAR PERNYATAAN

Yang bertanda tangan dibawah ini:

Nama : Muhammad Nadhil Sidqi  
NIM : 14507134031  
Program Studi : Teknik Elektronika D-III  
Judul Proyek Akhir : Sistem Cerdas Deteksi Suara Untuk Pengklasifikasian  
Penyakit jantung Menggunakan Jaringan Saraf Tiruan

Dengan ini saya menyatakan proyek akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar ahli madya atau gelar lainya di suatu Perguruan Tinggi, dan sepanjang pengetahuan penulis juga tidak terdapat karya atau pendapat yang pernah ditulis oleh orang lain, kecuali secara tertulis diacu dalam masalah dan disebutkan dalam naskah dan disebutkan dalam daftar pustaka.

Yogyakarta, 20 Desember 2018

Yang menyatakan,



Muhammad Nadhil Sidqi

NIM.14507134031

## **MOTTO**

*“Man Jadda Wajada”*

*“Man Shabara Zhafira”*

*“Hai orang-orang yang beriman, jadikanlah sabar dan shalat sebagai penolongmu, sesungguhnya Allah beserta orang-orang yang sabar”*

*(Al-baqarah:153)*

*“Jika tidak mau mengalami lelahnya belajar maka bersiaplah menghadapi perihnya kebodohan (Imam Syafi’i)”*

*“Saat masalahmu jadi terlalu berat untuk ditangani, beristirahatlah dan hitung berkah yang sudah kau dapatkan”*

*Setiap hembusan nafas yang diberikan ALLAH padamu bukan hanya berkah, tapi juga tanggung jawab”*

*“I dont care for the state of my sad or happy because i dont know which of them is better for me”*

## **LEMBAR PERSEMBAHAN**

*Dengan rahmat Allah yang maha pengasih lagi maha penyayang. Dengan ini saya persembahkan karya ini untuk:*

### ***Ibu dan Bapak Tercinta***

*Sebagai tanda bakti, hormat dan tanda terimakasih kepada Ibu Andara Wahyuni dan Bapak Zulkarnain Ahmad. Karena atas doa restu dan dukungan serta nasihatmu yang selalu engkau berikan untuk keberhasilanku*

### ***Keluarga Besar***

*Terimakasih kepada seluruh keluarga besar ku ibu, bapak, adik, nenek, kakek, yang selalu aku sayangi dan selalu menjadi penyemangatku*

### ***Guru dan Dosen***

*Terimakasih kepada semua guru khususnya guru ngaji dan dosen yang telang mengjar saya tanpa lelah dan penuh dengan kesabaran jasa mu takan terlupakan, semoga ilmu yang saya dapat bermanfaat dan barokah dunia dan akhirat*

### ***Teman-Teman***

*Terimakasih kawan atas segala kebersamaanmu, candamu, tawamu sungguh takkan pernah terlupakan*

### ***Dosen Pembimbing***

*Kepada Ibu Dessy Irmawati, S.T, M.T. selaku pembimbing proyek akhir. Terimakasih atas bimbingan, ilmu dan nasihat yang telah engkau berikan*

# **SISTEM CERDAS DETEKSI SUARA UNTUK PENGKLASIFIKASIAN PENYAKIT JANTUNG MENGGUNAKAN JARINGAN SARAF TIRUAN**

Oleh : Muhammad Nadhil Sidqi

NIM : 14507134031

## **ABSTRAK**

Sebuah alat pengenalan dini kelainan suara jantung normal dan *murmur* sangat diperlukan untuk membantu para tenaga medis dalam mendiagnosa kelainan jantung. Suara jantung memiliki pola khusus pada setiap orang, suara jantung yang tidak normal disebut dengan *murmur*. Suara jantung *murmur* akan menimbulkan bunyi yang khas, sehingga melalui frekuensi suara dan spektrum gelombang dapat diketahui suara jantung normal atau *murmur*.

Perancangan bertujuan untuk mengidentifikasi 2 jenis suara jantung yaitu normal dan *murmur*. Sistem ini berupa *hardware* dan *software* yang terdiri dari beberapa tahapan yaitu identifikasi kebutuhan, analisis kebutuhan, perancangan sistem, langkah pembuatan alat, diagram alur program, pengambilan data dan pengujian alat. *Hardware* ini menggunakan stetoskop akuistik modul regulator tegangan, rangkaian *cut off* frekuensi dan modul perekam suara yang berfungsi untuk memberikan input suara jantung yang akan diolah oleh sistem. *Software* ini menggunakan metode Jaringan Saraf Tiruan *backpropagation* dalam mengidentifikasi suara jantung normal dan *murmur*. Metode pelatihan ini menggunakan fungsi *nprtool* yang terdapat pada *neural network toolbox* MATLAB.

Berdasarkan hasil pengujian alat diagnosa ini diperoleh akurasi pelatihan sebesar 100% dalam mengenali 2 jenis suara jantung normal dan *murmur*. Setelah dilakukan pengujian alat secara keseluruhan rangkaian sistem sudah dapat bekerja dengan baik, melihat dari *hardware* perekam suara yang sudah dapat merekam suara jantung dan diteruskan ke dalam *software* untuk diolah. Pada pengujian *software* juga sudah mampu bekerja sesuai dengan fungsinya pada masing-masing *button*, dengan mampu melakukan perekaman suara, *filtering* suara, ekstraksi ciri, dan menganalisis suara jantung sehingga didapatkan keputusan apakah masuk kategori suara jantung normal atau *murmur*.

**Kata kunci** : suara jantung, MATLAB, *murmur*, Jaringan Saraf Tiruan, *backpropagation*

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang telah melimpahkan rahmat serta karunia-Nya sehingga penulis dapat menyelesaikan laporan proyek akhir ini dengan judul “Sistem Cerdas Deteksi Suara Untuk Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Saraf Tiruan”. Pembuatan proyek akhir ini sebagai syarat untuk memperoleh gelar Ahli Madya di Universitas Negeri Yogyakarta.

Penulis menyadari dalam penyusunan laporan proyek akhir ini tidak lepas dari bantuan dan kerjasama dengan pihak lain. Berkenaan dengan hal tersebut, penulis menyampaikan ucapan terima kasih kepada:

1. Orang tua, keluarga, dan orang-orang terdekat yang telah memberikan dukungan dan doa sehingga laporan proyek akhir ini dapat diselesaikan dengan baik.
2. Dessy Irmawati, S.T, M.T. selaku dosen pembimbing proyek akhir yang telah banyak memberikan semangat, dorongan, dan bimbingan selama penyusunan laporan proyek akhir ini.
3. Dr. Fatchul Arifin, M.T. dan Dra. Sri Waluyanti, M.Pd. selaku Ketua Jurusan dan Ketua Program Studi Diploma III Jurusan Pendidikan Teknik Elektronika dan Informatika Fakultas Teknik Universitas Negeri Yogyakarta.
4. Drs. Djoko Santoso, M.Pd. selaku Dosen Penasihat Akademik
5. Dr. Widarto, M.Pd. selaku Dekan Fakultas Teknik Universitas Negeri Yogyakarta.
6. Teman-teman Teknik Elektronika UNY 2014, terkhusus Ilham Hamiddin, dan Eri Aghis, Adhinda Firdhousa yang telah memberikan bantuan yang sangat banyak sehingga proyek akhir dapat diselesaikan.

7. Semua pihak, secara langsung maupun tidak langsung yang tidak dapat disebutkan disini satu persatu atas bantuan dan perhatiannya selama penyusunan laporan proyek akhir ini.

Semoga segala bantuan yang telah diberikan semua pihak yang tertulis diatas mendapatkan balasan dari Allah SWT. Penulis menyadari dalam penyusunan laporan ini masih banyak kekurangan dan jauh dari kata sempurna. Untuk itu penulis berharap para pembaca berkenan memberikan kritik dan saran yang membangun untuk menyempurnakan laporan.

Akhir kata penulis mengucapkan terima kasih dan semoga laporan ini dapat bermanfaat bagi para pembaca.

Yogyakarta, 21 Desember 2018  
Penulis

Muhammad Nadhil Sidqi

## DAFTAR ISI

	Halaman
PROYEK AKHIR .....	i
LEMBAR PERSETUJUAN.....	ii
LEMBAR PENGESAHAN .....	iii
LEMBAR PERNYATAAN KEASLIAN .....	iv
MOTTO.....	v
LEMBAR PERSEMBAHAN .....	vi
ABSTRAK .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	x
DAFTAR TABEL .....	xii
DAFTAR GAMBAR.....	xiii
DAFTAR LAMPIRAN.....	xiv
BAB I.....	1
PENDAHULUAN.....	1
A. Latar Belakang Masalah .....	1
B. Identifikasi Masalah .....	3
C. Batasan Masalah.....	3
D. Rumusan Masalah .....	4
E. Tujuan.....	4
F. Manfaat.....	5
G. Keaslian Gagasan .....	6
BAB II.....	8
PENDEKATAN PEMECAHAN MASALAH .....	8
A. Pengertian Sistem Cerdas.....	8
B. Klasifikasi Suara Jantung .....	9
C. Jaringan Saraf Tiruan .....	11
D. <i>Fast Fourier Transform</i> (FFT) .....	19
E. Komponen Pendukung .....	21
1. Modul Mikrofon MAX 9814.....	21

2. Modul Regulator Tegangan DC LM2596.....	22
3. Stetoskop Akustik .....	23
4. Rangkaian Filter .....	33
BAB III.....	33
KONSEP RANCANGAN.....	33
A. Identifikasi Kebutuhan.....	33
B. Analisis Kebutuhan .....	33
C. Perancangan Alat .....	34
D. Pembuatan Alat.....	45
E. Pengujian Alat.....	49
F. Pengoperasian Alat .....	57
BAB IV.....	59
PENGUJIAN DAN PEMBAHASAN .....	59
A. Pengujian.....	59
B. Pembahasan .....	70
BAB V .....	75
KESIMPULAN DAN SARAN .....	75
A. Kesimpulan.....	75
B. Keterbatasan Alat.....	77
C. Saran .....	77
DAFTAR PUSTAKA .....	79
LAMPIRAN.....	82

## DAFTAR TABEL

	Halaman
Tabel 1. Tabel Pengukuran Modul Regulator Tegangan .....	50
Tabel 2. Tabel Pengujian Rangkaian Filter .....	50
Tabel 3. Tabel Pengujian GUI Home .....	51
Tabel 4. Tabel Pengujian <i>Input Sound</i> .....	51
Tabel 5. Tabel Pengujian <i>Sound Filtering</i> .....	52
Tabel 6. Tabel Pengujian <i>Input Signal</i> .....	53
Tabel 7. Tabel Pengujian <i>Signal Analyser</i> .....	54
Tabel 8. Tabel Pengujian <i>Options</i> .....	54
Tabel 9. Tabel Pengujian JST .....	55
Tabel 10. Tabel Pengujian Unjuk Kerja Alat .....	56
Tabel 11. Hasil Pengujian Tegangan <i>Power Supply</i> .....	59
Tabel 12. Hasil Pengujian Rangkaian Filter .....	60
Tabel 13. Hasil Pengujian GUI Home .....	61
Tabel 14. Hasil Pengujian <i>Input Sound</i> .....	62
Tabel 15. Hasil Pengujian <i>Sound Filtering</i> .....	63
Tabel 16. Hasil Pengujian <i>Input Signal</i> .....	65
Tabel 17. Hasil Pengujian <i>Signal Analyser</i> .....	65
Tabel 18. Hasil Pengujian <i>Options</i> .....	66
Tabel 19. Hasil Pengujian JST .....	67
Tabel 20. Hasil Pengujian Unjuk Kerja Alat .....	68

## DAFTAR GAMBAR

	Halaman
Gambar 1. Lup-Dup Gelombang Suara Jantung .....	10
Gambar 2. Susunan Saraf Manusia .....	13
Gambar 3. Sel Saraf Sederhana .....	14
Gambar 4. Arsitektur Jaringan <i>Backpropagation</i> Secara Horizontal .....	16
Gambar 5. Arsitektur Jaringan <i>Backpropagation</i> Secara Horizontal .....	16
Gambar 6. Modul MAX 9814.....	22
Gambar 7. Modul Regulator DC LM2596.....	23
Gambar 8. Stetoskop Akustik.....	24
Gambar 9. Kurva Ideal Respon Berbagai Macam Filter.....	28
Gambar 10. Rangkaian <i>Low Pass Filter</i> .....	30
Gambar 11. Rangkaian <i>High Pass Filter</i> .....	30
Gambar 12. Rangkaian <i>Band Pass Filter</i> .....	31
Gambar 13. Rangkaian <i>Band Stop Filter</i> .....	31
Gambar 14. Blok Diagram Rangkaian .....	34
Gambar 15. Rangkaian Perekam Suara .....	36
Gambar 16. <i>Flowchart</i> Sistem.....	42
Gambar 17. <i>Flowchart</i> Program .....	44
Gambar 28. Desain <i>Box</i> Perekam Suara Jantung .....	45
Gambar 29. <i>Layout</i> Rangkaian Perekam Suara Tampak Atas .....	46
Gambar 20. <i>Layout</i> Rangkaian Perekam Suara Tampak Bawah .....	46
Gambar 21. Tampilan GUI Home .....	47
Gambar 22. Tampilan GUI Utama .....	48

## DAFTAR LAMPIRAN

	Halaman
Lampiran 1. Desain <i>Box</i> .....	82
Lampiran 2. Rangkaian Alat Perekam Suara .....	83
Lampiran 3. Rangkaian Alat Perekam Suara .....	84
Lampiran 4. Daftar Komponen Alat dan Harga .....	85
Lampiran 5. <i>Source Code</i> .....	86
Lampiran 6 <i>Datasheet</i> MAX9814.....	95
Lampiran 7. <i>Datasheet</i> LM2596.....	104

# BAB I

## PENDAHULUAN

### A. Latar Belakang

Serangan jantung masih menjadi salah satu penyebab kematian tertinggi di berbagai negara di dunia. Data Riset Kesehatan Dasar menunjukkan angka kematian akibat penyakit tidak menular di Indonesia meningkat menjadi 59.5% dengan kematian akibat penyakit jantung menempati posisi ketujuh penyebab kematian di berbagai negara di dunia (Riskerdas, 2013). Gejala abnormalitas jantung (*murmur*) menjadi permasalahan yang seringkali muncul secara tiba-tiba, oleh sebab itu pengenalan secara dini terhadap kelainan jantung dapat membantu untuk terhindar dari serangan jantung. Sampai saat ini dokter masih menggunakan isyarat suara jantung untuk memantau kinerja jantung dengan menggunakan stetoskop akustik yang penggunaannya menghasilkan suara yang masih lemah, sehingga diperlukan kepekaan dan pengalaman untuk dapat mendiagnosis kelainan jantung, selain itu keterbatasan fisik juga sangat mempengaruhi hasil interpretasi sehingga hasil diagnosis sangat dipengaruhi oleh subyektifitas dokter.

Jantung merupakan organ yang memiliki peranan penting dalam metabolisme dan suplai oksigen ke seluruh tubuh. Organ ini bekerja dengan cara mengumpulkan darah di seluruh tubuh yang kurang akan oksigen dan dipompa ke dalam paru-paru, sehingga darah mengambil oksigen dari paru-paru untuk disebarkan ke seluruh tubuh dan akan membuang karbondioksida. Selanjutnya

jantung mengumpulkan darah dari paru-paru yang banyak mengandung oksigen dan memompanya untuk disebarkan keseluruh jaringan tubuh. Aktivitas listrik dari jantung diketahui dari perekaman dan pencatatan detak jantung. Fungsi perekaman ini adalah untuk mengetahui frekuensi detak jantung yang dinyatakan dengan satuan Hz. Frekuensi ini memberikan informasi mengenai bagaimana kondisi jantung, ada tidaknya gangguan pembentukan impuls dan gangguan fungsi jantung. Suara jantung normal mempunyai rentang frekuensi 20 Hz hingga 200 Hz, sedangkan suara jantung *murmur* mempunyai rentang frekuensi hingga 1000 Hz. Salah satu jenis regurgitasi menyebabkan *murmur* dalam rentang 100 hingga 600 Hz dan bahkan untuk jenis *murmur* tertentu hingga 1000 Hz.

Alat diagnostik untuk membantu dokter dalam menentukan detak jantung serta mengklasifikasikannya sebagai jantung sehat atau sakit yang ada selama ini sangat beragam baik dari metode pengukuran, komponen dan program aplikasi yang dipakai. Hal ini berakibat pada harga alat pendeteksi kelainan jantung yang berada di pasaran. Diagnosa kelainan jantung dari yang ada selama ini terkesan sebagai suatu piranti yang sangat mahal, dan hanya ada pada rumah sakit tertentu.

Pada saat ini komponen yang digunakan dalam mendapatkan data suara detak jantung terbatas hanya menggunakan stetoskop akustik sehingga diperlukan modifikasi untuk mendapatkan rekaman frekuensi suara detak jantung yaitu dengan komponen stetoskop akustik yang telah dipasang *mic condenser* dan komponen elektronika lainnya, kemudian dihubungkan dengan laptop untuk mendapatkan gambaran grafis dari frekuensi rekaman suara detak jantung yang kemudian dikumpulkan menjadi satu *database*.

*Database* ini kemudian akan diidentifikasi apakah pasien memiliki kelainan suara jantung atau tidak. Klasifikasi kelainan suara jantung dan jantung normal ini menggunakan klasifikasi dengan metode Jaringan Saraf Tiruan (*artificial neural network*). Jaringan Saraf Tiruan ini (JST) dapat mengklasifikasikan dengan baik suara jantung normal atau *murmur*. Jaringan Saraf Tiruan ini memiliki pola kerja yang efektif dalam menganalisis sinyal multiple serta lebih sensitif daripada program interpretasi dan kardiologis dalam mendiagnosa kelainan suara jantung.

## **B. Identifikasi Masalah**

Berdasarkan uraian latar belakang masalah diatas, maka dapat dibuat suatu identifikasi masalah sebagai berikut:

1. Belum ada alat elektronik pengenalan dini terhadap kelainan jantung untuk mendiagnosa jantung *murmur* dan jantung normal.
2. Alat untuk mendapatkan data suara jantung hanya terbatas pada stetoskop yang tidak bisa digunakan untuk perekaman suara detak jantung.
3. Belum adanya sistem diagnostik berbasis suara untuk membantu dokter dalam mendiagnosa kelainan jantung.

## **C. Batasan Masalah**

Berdasarkan indentifikasi masalah di atas maka perlu adanya batasan masalah agar ruang lingkup masalah menjadi jelas. Pada proyek akhir membahas mengenai perekaman suara jantung proses pengenalan dini kelainan suara jantung. Permasalahan utama yang akan dibahas adalah proses pengenalan dini terhadap kelainan jantung untuk mendiagnosa suara jantung *murmur* dan jantung normal. Proses menggunakan stetoskop yang telah dimodifikasi untuk pengambilan data

suara jantung dan sistem cerdas berupa software untuk mendiagnosa dini kelainan suara jantung.

#### **D. Rumusan Masalah**

Berdasarkan batasan masalah, maka dapat dirumuskan permasalahan sebagai berikut:

1. Bagaimana perancangan *hardware* dari Sistem Cerdas Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Saraf Tiruan?
2. Bagaimana perancangan *software* menggunakan Matlab R2018a untuk mengklasifikasikan penyakit jantung?
3. Bagaimana Unjuk kerja Sistem Cerdas Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Saraf Tiruan?

#### **E. Tujuan**

Adapun tujuan dari pembuatan proyek akhir yang berjudul “Sistem Cerdas Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Saraf Tiruan” untuk memenuhi persyaratan guna memperoleh gelar ahli madya adalah sebagai berikut:

1. Merealisasikan rancangan *hardware* dari sistem cerdas dengan menggunakan stetoskop akuistik yang dimodifikasi menggunakan *mic condenser*.
2. Merealisasikan rancangan *software* sistem cerdas menggunakan Jaringan Saraf Tiruan.
3. Mengetahui unjuk kerja dari sistem cerdas pengklasifikasian penyakit jantung menggunakan Jaringan Saraf Tiruan.

## **F. Manfaat**

Pada pembuatan proyek akhir ini diharapkan dapat bermanfaat bagi mahasiswa, lembaga pendidikan, dan kedokteran. Adapun manfaat yang diharapkan dari pembuatan tugas akhir ini antara lain:

### **1. Bagi Mahasiswa**

- a. Sebagai tolak ukur individual setelah mendapatkan ilmu dari bangku kuliah dan kehidupan sehari-hari untuk diimplementasikan dalam bentuk suatu aplikasi.
- b. Dapat mengaplikasikan ilmu yang didapat selama di bangku kuliah dan menerapkannya secara nyata.
- c. Dapat digunakan sebagai bahan pembelajaran dan penambah wawasan tentang membuat *Software* dan sistem Jaringan Saraf Tiruan dengan menggunakan MATLAB, serta sebagai kajian untuk pengembangan selanjutnya.
- d. Sebagai bentuk kontribusi terhadap Universitas dan pengabdian kepada masyarakat dalam bentuk suatu karya yang bermanfaat.

### **2. Bagi Universitas**

Dapat dijadikan tolak ukur daya serap mahasiswa yang bersangkutan selama menempuh pendidikan di perguruan tinggi dan mampu menerapkan ilmu secara praktis pada bidang-bidang yang sesuai dengan program studi yang dipelajari.

### **3. Bagi Dunia Kedokteran**

Dapat membantu peran *user* terutama perawat, dokter dan pasien dalam mendiagnosa kondisi kesehatan jantung, kemudian mampu mengenali dini kelainan jantung terhadap pasien, sehingga apabila ditemukan kelainan pada jantung maka dapat ditangani secara efisien, cepat dan akurat.

#### **G. Keaslian Gagasan**

Karya tugas akhir ini asli dari gagasan pribadi yang terinspirasi dari tingkat kematian yang disebabkan oleh serangan jantung. Alat ini diharapkan dapat membantu perawat maupun dokter dalam mengenali dini kelainan jantung pada pasien. Adapun alat dan jurnal yang menjadi acuan penulis membuat alat ini, yang pertama yaitu MEDISKOP “Medical Elektronik Stetoskop” yang dirancang oleh Dionita Rani Karyono, dan Imah Nur Chasanah dari Prodi Ilmu Keperawatan Fakultas Kedokteran, lalu Muhammad Fadhil Ainuri dari Teknik Elektro Fakultas Teknik dan Abdullah Ibnu Hasan dari Prodi Elektronika dan Instrumentasi FMIPA Universitas Gajah Mada, yang kedua yaitu jurnal “Simulasi Pengenalan Kelainan Jantung Dengan Menggunakan Metode Jaringan Syaraf Tiruan” yang dibuat oleh Hermanto Sitinjak dari Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, dan yang ketiga yaitu jurnal “Analisa Suara Jantung Normal Menggunakan *Discrete Wavelet Transform* (DWT) dan *Fast Fourier Transform* (FFT)” yang dibuat oleh Putri Madona dari Jurusan Teknik Elektro Program Studi Teknik Elektronika Politeknik Caltex Riau.

Kekurangan dari alat MEDISKOP di atas yaitu :

1. Belum ada *user interface* lanjutan setelah perekaman data suara jantung diambil.
2. Belum adanya system cerdas untuk mengolah data rekaman suara jantung tersebut untuk mengenali adanya kelainan jantung.
3. Belum ada keputusan untuk mengkategorikan pasien apakah da kelainan jantung atau normal.

Sedangkan kelebihan dari proyek akhir ini yaitu :

1. Menggunakan stetoskop akustik yang dihubungkan menggunakan *mic condenser* dan komponen elektronika lainnya yang dihubungkan ke laptop untuk melakukan perekaman data pasien.
2. Menggunakan sistem cerdas dengan menggunakan metode Jaringan Saraf Tiruan sebagai pengolah data rekaman suara jantung.
3. Menggunakan GUI sebagai *user interface* dari hasil pengolahan dari JST.

Dari acuan diatas penulis ingin membuat perangkat keras dari sistem cerdas pengklasifikasian penyakit jantung dengan menggunakan jaringan saraf tiruan guna membantu peran *user* terutama perawat, dokter dalam mendiagnosa kondisi kesehatan jantung.

## BAB II

### PENDEKATAN PEMECAHAN MASALAH

#### A. Pengertian Sistem Cerdas

Sistem Cerdas merupakan bagian dari bidang Ilmu Komputer/Informatika dan Rekayasa Cerdas untuk pengembangan berbagai metode bekemampuan tinggi yang diilhami oleh fenomena alam untuk menyelesaikan berbagai masalah kompleks di dunia nyata. Masalah-masalah kompleks tersebut juga bisa berkaitan dengan pengolahan *big data* dan perancangan *embedded systems*. Sebuah Sistem Cerdas digunakan untuk mampu menghasilkan sistem dan perangkat lunak berbasis algoritma cerdas, baik *inter* maupun *multi disipliner*, dengan kemampuan analisis dan teknis sehingga menghasilkan karya inovatif dan teruji.

Kecerdasan Buatan adalah ilmu rekayasa yang membuat suatu mesin mempunyai intelegensi tertentu khususnya program komputer yang “cerdas” (John Mc Carthy, 1956). Kecerdasan buatan merupakan kawasan penelitian, aplikasi dan intruksi yang terkait dengan pemrograman komputer untuk melakukan sesuatu hal yang dalam pandangan manusia adalah cerdas. (H.A Simon, 1987). Kecerdasan merupakan bagian kemampuan komputasi untuk mencapai tujuan dalam dunia. Ada bermacam-macam jenis dan derajat kecerdasan untuk manusia, hewan dan mesin. Kecerdasan buatan merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia. (Rich and Knight, 1991).

Kecerdasan buatan (AI) merupakan cabang dari ilmu komputer yang dalam mempresentasi pengetahuan lebih banyak menggunakan bentuk symbol-simbol

daripada bilangan, dan memproses informasi berdasarkan metode heuristic (Metode Heuristik adalah teknik yang dirancang untuk memecahkan masalah yang mengabaikan apakah solusi dapat dibuktikan benar, tapi yang biasanya menghasilkan solusi yang baik atau memecahkan masalah yang lebih sederhana yang mengandung atau memotong dengan pemecahan masalah yang lebih kompleks.) atau dengan berdasarkan sejumlah aturan (Encyclopedia Britannica).

Kecerdasan adalah kemampuan untuk belajar atau mengerti dari pengalaman. Memahami pesan yang kontradiktif dan ambigu, menanggapi dengan cepat dan baik atas situasi yang baru, menggunakan penalaran dalam memecahkan masalah dan menyelesaikannya secara efektif (Winston dan Pendergast, 1994).

## **B. Klasifikasi Suara Jantung**

Suara jantung normal memiliki dua irama. Jika didengarkan, suaranya seperti “lup-dup” yang berulang. Suara ini berasal dari aktivitas katup jantung yang membuka dan menutup ketika darah mengalir melalui jantung.

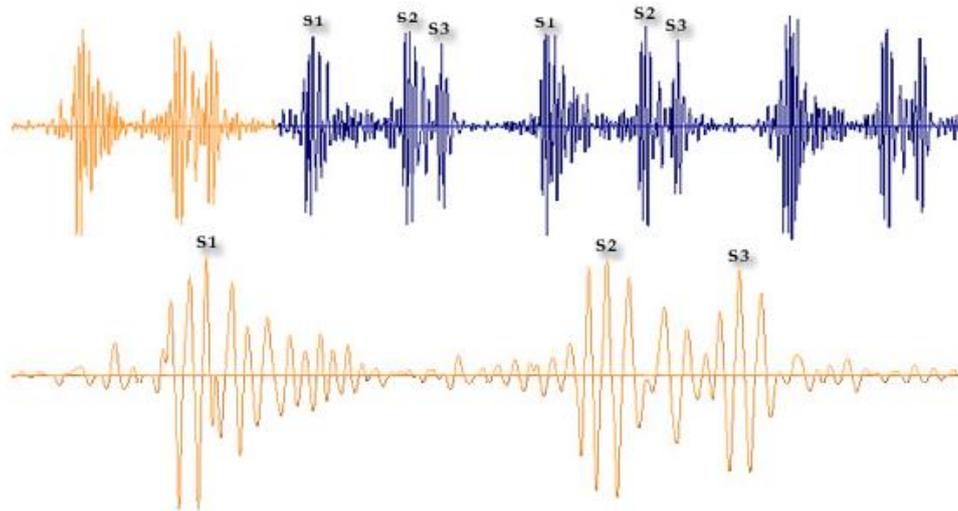
Suara jantung bisa didengar secara lebih jelas ketika dokter melakukan pemeriksaan dengan menggunakan stetoskop akuistik. Bunyi jantung bisa menjadi alat ukur untuk mengetahui kondisi kesehatan seseorang, termasuk kesehatan jantung. Dari hasil pemeriksaan itu, dokter akan menentukan langkah penanganan yang tepat.

Anatomi jantung terdiri dari empat ruangan, yaitu atrium (serambi) kanan dan kiri di bagian atas, serta ventrikel (bilik) kanan dan kiri di bagian bawah. Di antara setiap atrium dan ventrikel terdapat katup kecil yang berfungsi untuk menjaga

agar darah tetap mengalir ke arah yang benar. Secara umum, suara detak jantung berasal dari:

- 1) Kekuatan jaringan yang menghubungkan katup dan otot jantung.
- 2) Getaran yang terjadi ketika katup jantung membuka dan menutup.
- 3) Darah yang mengalir tidak normal atau terlalu cepat di dalam jantung.

Biasanya suara jantung normal atau tidak, diketahui ketika seseorang sedang sakit dan memeriksakan diri ke dokter, atau pada saat melakukan *medical check-up*. Jika bunyi jantung Anda bukan “lup-dup” atau ada bunyi tambahan, kemungkinan jantung Anda mengalami kelainan. Gambar 1 menunjukkan gambaran lup-dup pada gelombang suara jantung.



Gambar 1. Lup-Dup Gelombang Suara Jantung

(dikutip dari:[http://www.stethographics.com/main/physiology\\_hs\\_introduction.html](http://www.stethographics.com/main/physiology_hs_introduction.html). 06 Desember 2018).

Keterangan: S1 = Lup, S2 = Dup

## 1. Penyebab Bunyi Jantung Abnormal

Salah satu bentuk bunyi jantung yang tidak normal adalah bising jantung, atau dikenal sebagai *murmur* jantung. *Murmur* jantung terjadi ketika katup jantung tidak menutup dengan benar, sehingga darah dapat mengalir kembali dan menimbulkan bising jantung. Meski tak selalu menimbulkan bahaya, namun pada kasus tertentu *murmur* jantung harus ditangani secara khusus.

*Murmur* jantung seringkali disebabkan oleh penyakit katup jantung. Meski begitu, *murmur* jantung juga bisa terjadi karena adanya peningkatan aliran darah yang melintasi katup jantung. Kondisi ini biasanya terjadi karena pengaruh penyakit lain yang bukan merupakan gangguan jantung seperti anemia dan hipertiroidisme.

Selain *murmur* jantung, ada pula bunyi jantung lain yang dianggap abnormal, misalnya *friction rub*, yang menimbulkan suara gesekan, dapat menandakan adanya peradangan pada perikardium (membran pembungkus jantung). Serta *gallop*, yakni ketika bunyi jantung menyerupai bunyi derap langkah kuda, kerap berkaitan dengan kondisi gagal jantung.

Bunyi jantung bias membantu dokter untuk mendeteksi adanya penyakit pada otot jantung dan penyakit jantung bawaan. Untuk memastikan, dokter akan melakukan pemeriksaan lanjutan, misalnya dengan ekokardiografi untuk melihat aliran darah atau CT-scan jantung untuk melihat kondisi anatomis jantung.

## C. Jaringan Saraf Tiruan (*Neural Network*)

Jaringan Saraf Tiruan atau *Neural Network* pertama kali dikemukakan pada tahun 1943 oleh ahli saraf Warren McCulloch dan *logician* Walter Pitts dalam

jurnal yang berjudul “Bagaimana neuron bekerja?”. Jurnal tersebut memperagakan *neural network* yang sederhana dengan menggunakan rangkaian listrik, akan tetapi teknologi yang tersedia pada saat itu belum memungkinkan untuk dikembangkan lebih jauh.

#### 1. Pengertian Jaringan Saraf Tiruan (*Neural Network*)

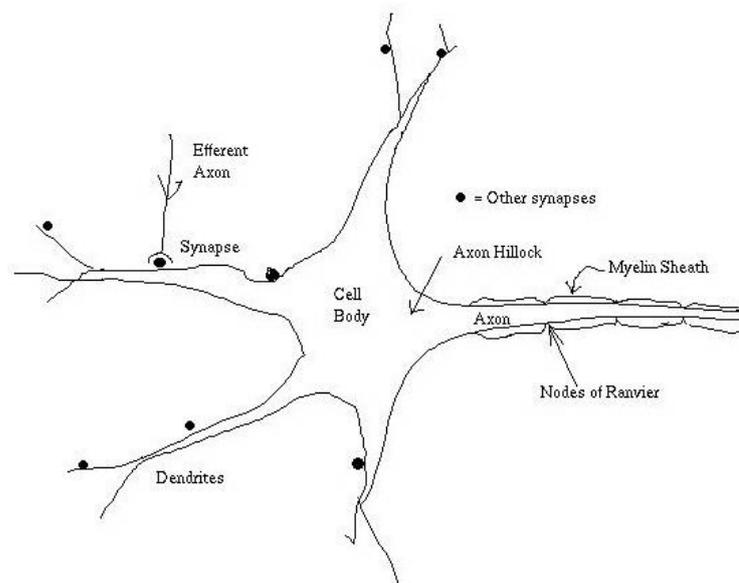
Jaringan Saraf Tiruan atau *Neural Network* adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistem sel saraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistem pemrosesan informasi. Jaringan Saraf Tiruan (*Neural Network*), seperti manusia, belajar dari suatu contoh. Jaringan Saraf Tiruan (*Neural Network*) dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran.

Jaringan Saraf Tiruan atau *Neural Network* berkembang secara pesat pada beberapa tahun terakhir. Jaringan Saraf Tiruan telah dikembangkan sebelum adanya suatu komputer konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun.

#### 2. Inspirasi Jaringan Saraf Tiruan (*Neural Network*)

Jaringan Saraf Tiruan (*Neural Network*) terinspirasi dari penelitian kecerdasan buatan, terutama percobaan untuk menirukan fault-tolerance dan kemampuan untuk belajar dari sistem saraf biologi dengan model struktur *low-level* dari otak. Otak terdiri dari sekitar (10.000.000.000) sel saraf yang saling berhubungan. Sel saraf mempunyai cabang struktur input (dendrites), sebuah inti sel dan percabangan struktur *output* (axon). Axon dari sebuah sel terhubung

dengan dendrites yang lain melalui sebuah synapse. Ketika sebuah sel saraf aktif, kemudian menimbulkan suatu *signal electrochemical* pada axon. *Signal* ini melewati synapses menuju ke sel saraf yang lain. Sebuah sel saraf lain akan mendapatkan signal jika memenuhi batasan tertentu yang sering disebut dengan nilai ambang atau (*threshold*). Gambar 2 menunjukkan bagian-bagian susunan sel saraf manusia.



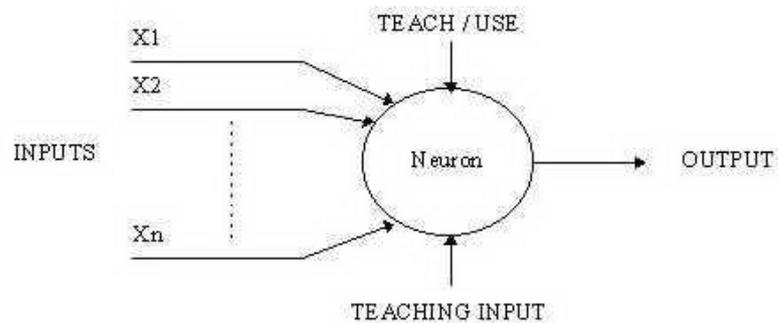
Gambar 2. Susunan Saraf Manusia

(dikutip dari: <http://elektronika-dasar.web.id/jaringan-saraf-tiruan-neural-network/#>. 22 November 2018).

### 3. Perbandingan Jaringan Saraf Tiruan dengan Metode Konvensional

Jaringan Saraf Tiruan memiliki pendekatan yang berbeda untuk memecahkan masalah bila dibandingkan dengan sebuah komputer konvensional. Umumnya komputer konvensional menggunakan pendekatan algoritma (komputer

konvensional menjalankan sekumpulan perintah untuk memecahkan masalah). Jika suatu perintah tidak diketahui oleh komputer konvensional maka komputer konvensional tidak dapat memecahkan masalah yang ada. Sangat penting mengetahui bagaimana memecahkan suatu masalah pada komputer konvensional dimana komputer konvensional akan sangat bermanfaat jika dapat melakukan sesuatu dimana pengguna belum mengetahui bagaimana melakukannya. Gambar 3 menunjukkan gambaran sel saraf sederhana.



Gambar 3. Sel Saraf Sederhana

(dikutip dari: <http://elektronika-dasar.web.id/jaringan-saraf-tiruan-neural-network/#>. 22 November 2018).

Jaringan Saraf Tiruan (*Neural Network*) dan suatu algoritma komputer konvensional tidak saling bersaing namun saling melengkapi satu sama lain. Pada suatu kegiatan yang besar, sistem yang diperlukan biasanya menggunakan kombinasi antara keduanya (biasanya sebuah komputer konvensional digunakan untuk mengontrol Jaringan Saraf Tiruan (*Neural Network*) untuk menghasilkan efisiensi yang maksimal. Jaringan Saraf Tiruan (*Neural Network*) tidak

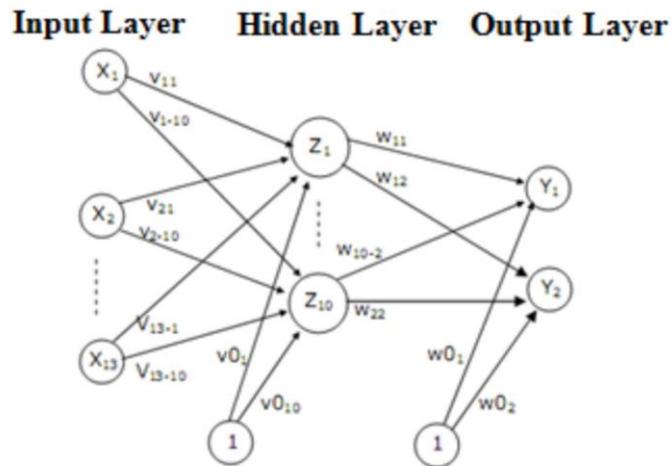
memberikan suatu keajiban tetapi jika digunakan secara tepat akan menghasilkan sesuatu hasil yang luarbiasa.

#### 4. Jenis Metode Jaringan Saraf Tiruan

##### a. Metode *Backpropagation*

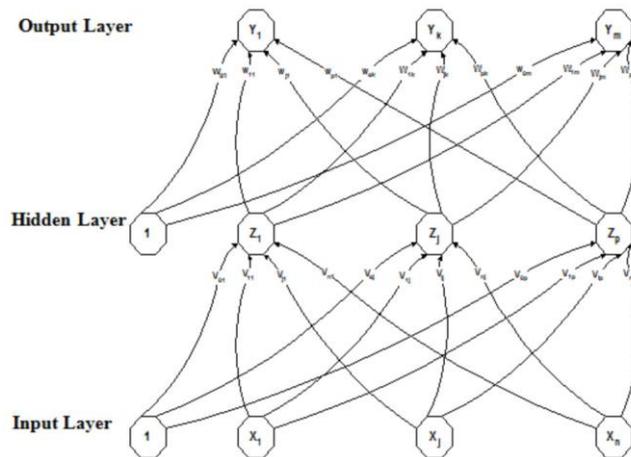
*Backpropagation* adalah algoritma pembelajaran untuk memperkecil tingkat error dengan cara menyesuaikan bobot melalui perbedaan output dan target yang diinginkan. *Backpropagation* juga merupakan sebuah metode sistematis untuk pelatihan multilayer JST. *Backpropagation* dikatakan sebagai algoritma multilayer karena *Backpropagation* memiliki tiga layer dalam proses pelatihannya, yaitu *input* layer, *hidden* layer dan *output* layer, dimana *backpropagation* ini merupakan perkembangan dari *single layer network* (jaringan layer tunggal) yang memiliki dua layer, yaitu *input* layer dan *output* layer, dengan adanya *hidden* layer pada *backpropagation* dapat menyebabkan besarnya tingkat *error* pada *backpropagation* lebih kecil dibanding tingkat *error* pada *single layer network*. Hal tersebut dikarenakan *hidden* layer pada *backpropagation* berfungsi sebagai tempat untuk mengupdate dan menyesuaikan bobot, sehingga nilai bobot baru yang bias diarahkan mendekati dengan target *output* yang diinginkan. Arsitektur *backpropagation* terdiri dari tiga layer, yaitu *input* layer, *hidden* layer dan *output* layer. Pada *input* layer tidak terjadi proses komputasi, namun pada *input* layer terjadi pengiriman sinyal input ke X ke *hidden* layer, pada *hidden* dan *output* layer tersebut berdasarkan fungsi aktivasi tertentu. Dalam algoritma *backpropagation* ini digunakan fungsi aktivasi sigmoid biner, karena output yang

diharapkan bernilai 0 sampai 1. Gambar 4 dan 5 menunjukkan arsitektur jaringan *backpropagation* secara horizontal.



Gambar 4. Arsitektur Jaringan *Backpropagation* Secara Horizontal

(dikutip dari: <https://novikaginanto.wordpress.com/2012/11/14/backpropagation/>. 19 Desember 2018)



Gambar 5. Arsitektur Jaringan *Backpropagation* Secara Horizontal

(dikutip dari: <https://novikaginanto.wordpress.com/2012/11/14/backpropagation/>. 19 Desember 2018)

Pada gambar 4 dan 5 tiga layer *backpropagation* adalah *input layer*, *hidden layer* dan *output layer*. Pada *input layer*, inputan divariabelkan dengan  $X_n$ , pada *hidden layer* terdapat bobot ( $V_{ij}$ ) dan bias ( $V_{oj}$ ), serta  $Z$  sebagai data *hidden layer*, pada *output layer* juga demikian, terdapat bobot ( $w_{ij}$ ) dan bias ( $W_{oj}$ ) dengan data *output* divariabelkan dengan  $Y$ .

b. Algoritma *Backpropagation*

Algoritma *backpropagation* adalah sebuah algoritma untuk memperkecil tingkat *error* dengan menyesuaikan bobot berdasarkan perbedaan output dan target yang diinginkan. Secara umum algoritma terdiri dari 3 langkah utama, yaitu:

- 1) Pengambilan *input*
- 2) Penelusuran *error*
- 3) Penyesuaian bobot

Pada pengambilan *input*, terlebih dahulu dilakukan inisialisasi bobot, kemudian masuk ke dalam algoritma proses *backpropagation* yang terdiri dari komputasi maju yang bertujuan untuk menelusuri besarnya *error* dan komputasi balik untuk mengupdate dan menyesuaikan bobot. Dalam mengupdate bobot dapat dilakukan dengan dua cara, yaitu tanpa momentum dan dengan momentum. Namun dalam mengupdate bobotnya dilakukan tanpa memperhatikan besarnya momentum. Dalam metode *backpropagation*, algoritma yang harus dilakukan adalah inisialisasi bobot, komputasi *feed forward* dan *backpropagation* serta inisialisasi kondisi stopping berdasarkan nilai batas error atau jumlah batas epoch.

*Epoch* merupakan rangkaian langkah dalam pembelajaran ANN. Satu epoch diartikan sebagai satu kali pembelajaran ANN.

Pada dasarnya proses algoritma backpropagation terdiri dari komputasi maju (*feed forward*) dan komputasi balik (*backpropagation*). Komputasi maju digunakan sebagai sebuah algoritma untuk menghitung nilai aktivasi yang ada pada semua neuron baik yang ada pada *hidden layer* ataupun *output layer*. Berikut merupakan algoritma yang digunakan dalam komputasi maju.

- 1) Neuron yang berada pada lapisan tersembunyi melakukan perhitungan nilai total masukan dari lapisan tersembunyi dengan cara menjumlahkan perkalian sinyal masukan dengan bobot antara lapisan masukan (*input layer*) dan lapisan tersembunyi (*hidden layer*) dan nilai bias.
- 2) Neuron tersebut melakukan perhitungan nilai aktivasi.
- 3) Sinyal keluaran neuron yang berada pada lapisan keluaran melakukan perhitungan untuk menghitung nilai keluaran.
- 4) Neuron keluaran tersebut menghitung nilai aktivasi dengan menerapkan fungsi aktivasi pada lapisan keluarannya.

Backpropagation merupakan sebuah algoritma yang berfungsi untuk melakukan perhitungan balik dari neuron keluaran agar memiliki nilai bobot yang sesuai dalam *jarangan neural network*, dengan komputasi balik ini nilai *error* atau kesalahan bias dikurangi dengan cukup baik. Berikut ini adalah gambaran dari algoritma komputasi balik yang digunakan pada *Artificial Neural Network*.

#### **D. Fast Fourier Transform (FFT)**

Pada tahun 1960, J. W. Cooley dan J. W. Tukey, berhasil merumuskan suatu teknik perhitungan algoritma Fourier Transform yang cepat dan efisien. Teknik perhitungan algoritma ini dikenal dengan sebutan *Fast Fourier Transform* atau yang disebut dengan istilah FFT yang diperkenalkan oleh J.S.Bendat dan A.G.Piersol pada 1986. *Fast Fourier Transform* (FFT) adalah sumber dari suatu algoritma untuk menghitung *Discrete Fourier Transform* (transformasi fourier diskri atau DFT) dengan cepat, efisien dan inversnya.

*Fast Fourier Transform* (FFT) diterapkan dalam berbagai bidang pengolahan sinyal digital dan memecahkan persamaan diferensial parsial menjadi algoritma-algoritma untuk penggandaan bilangan integer dalam jumlah banyak. Ada pun kelas dasar dari algoritma FFT yaitu *Decimation In Time* (DIT) dan *Decimation In Frequency* (DIF). Garis besar dari kata *Fast* diartikan karena formulasi FFT jauh lebih cepat dibandingkan dengan metode perhitungan algoritma *Fourier Transform* sebelumnya.

Metode FFT memerlukan sekitar 10000 operasi algoritma matematika untuk data dengan 1000 observasi, 100 kali lebih cepat dibandingkan dengan metode sebelumnya.

Penemuan FFT dan perkembangan personal komputer, teknik FFT dalam proses analisa data menjadi sangat populer, dan merupakan salah satu metode

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt$$

baku dalam analisa data. Satu bentuk transformasi yang umum digunakan untuk merubah sinyal dari domain waktu ke domain frekuensi adalah Transformasi Fourier:

Persamaan dari bentuk sinyanya  $x(t)$ .

FFT dalam pengolahan isyarat meliputi Periode dan frekuensi:

#### 1. Periode

Secara umum periode didefinisikan sebagai waktu yang dibutuhkan untuk sebuah isyarat atau gelombang mencapai suatu gelombang penuh dan dapat menentukan nilai periodesitasnya. Perlu dicermati bahwa pengertian ini berlaku untuk isyarat monokromatis, isyarat yang dimaksud adalah gelombangnya bersifat tunggal, pasti memiliki sebuah periode. Dengan demikian isyarat itu dikenal dengan istilah priodis, pengamatan dapat dilakukan dengan memantau gelombang kita dapat mengetahui nilai nilai yang terkandung dalam isyarat serta periodenya.

#### 2. Frekuensi

Ada periode, maka ada frekuensi diartikan sebagai jumlah gelombang yang terjadi dalam 1 detik. Frekuensi didefinisikan secara sederhana sebagai kebalikan dari waktu. Sehingga waktu yang satuannya adalah detik (*second*) akan menjadi Hertz (1-per *second*) hanya akan memiliki tepat satu nilai spektrum. Yang dikenal

dengan spektrum frekuensi. Pengertian frekuensi ini juga berlaku untuk gelombang monokromatis.

## **E. Komponen Pendukung**

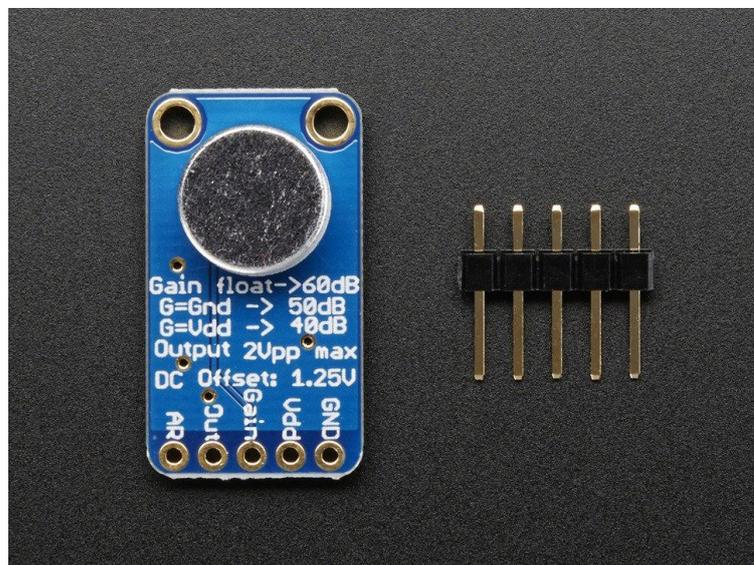
### **1. Modul Mikrofon MAX 9814**

Modul mikrofon ini, mempunyai kemampuan untuk mengatur penguatan secara otomatis, sehingga suara terdekat yang keras, akan otomatis disesuaikan agar tidak terdengar begitu keras, dan suara terjauh yang samar-samar, akan otomatis dikuatkan agar dapat terdengar oleh *user*. Semua ini menggunakan IC MAX 9814 yang mempunyai fitur AGC (*Automatic Gain Control*).

Modul mikrofon ini menyediakan beberapa fitur yang memudahkan *user* dalam merekam suara yang dibutuhkan. *Default max gain* yang dimiliki modul ini adalah 60dB, tapi bisa diatur ke 40dB atau 50dB dengan menghubungkan pin Gain modul dengan pin Vdd modul atau pin GND modul. *Attack./Release ratio* juga dapat diubah, yang nilai secara *default* nya adalah 1:4000, tapi bisa diatur hingga mencapai 1:2000 atau 1:500. Modul ini mempunyai tegangan *output* maksimal sekitar 2Vpp dengan bias 1,25V, sehingga dapat dengan mudah digunakan dengan berbagai model *Analog/Digital Converter* yang mana rata-rata mempunyai input tegangan hingga 3,3 Volt. Jika ingin langsung menghubungkan modul dengan jack audio, pin *output* hanya tinggal dihubungkan secara seri dengan kapasitor 1uF. Gambar komponen MAX 9814 dapat dilihat pada gambar 6.

Berikut spesifikasi modul mikrofon MAX 9814 ini :

- a. Tegangan supply : 2,7 – 5,5 Volt dengan arus 3mA
- b. *Output* tegangan : 2Vpp dalam 1,25V bias
- c. Respon frekuensi : 20Hz – 20 KHz
- d. *Attack/Release ratio* yang dapat diubah-ubah
- e. *Automatic Gain*, yang dapat dipilih antara 40dB, 50dB dan 60dB
- f. *Low input noise density* sebesar  $30\text{nV}/\sqrt{\text{Hz}}$
- g. *Low THD* : 0,04%



Gambar 6. Modul MAX 9814

(dikutip dari: <https://www.adafruit.com/product/1713>. 29 November 2018).

## 2. Modul Regulator Tegangan DC LM2596

Modul regulator tegangan ini merupakan jenis *step down* regulator, dimana tegangan input yang dimiliki modul ini adalah 4-40 Volt, dengan *output* tegangan

yang dihasilkan antara 1,23 – 35 Volt. Modul ini dapat menyuplai arus hingga 2 Ampere dan 3 Ampere jika ditambah dengan *heatsink*. Efisiensi yang dimiliki modul ini adalah 92% dengan tegangan *dropout* minimal sebesar 2Volt.

Dilengkapi dengan pembatas arus, dan pelindung *short circuit*. Modul ini digunakan karena cocok untuk berbagai project yang membutuhkan tegangan dan arus yang stabil, dari berbagai input tegangan yang bisa didapatkan. Gambar komponen modul regulator tegangan LM2596 dapat dilihat pada gambar 7.



Gambar 7. Modul Regulator DC LM2596

(dikutip dari: <https://www.addicore.com/lm2596-module-p/ad281.htm>. 29

November 2018).

### 3. Stetoskop Akustik

Stetoskop dalam bahasa Yunani yaitu *stethos* yang artinya dada dan *skopein* artinya memeriksa, maka Stetoskop adalah sebuah alat medis akustik untuk memeriksa suara dalam tubuh. Gambar komponen stetoskop akustik dapat dilihat pada gambar 8.



Gambar 8. Stetoskop Akustik

(dikutip dari: <https://www.medicalogy.com/blog/jenis-jenis-stetoskop/>  
. 29 November 2018).

Stetoskop banyak digunakan untuk mendengar suara jantung dan pernapasan, meskipun dia juga digunakan untuk mendengar intestine dan aliran darah dalam arteri dan "vein". Ketika kita sakit dan diperiksa ke dokter kemudian dokter dengan menggunakan Stetoskop yang menggantung di leher dan menempatkan ujung datar dingin stetoskop pada bagian tubuh kita dan kita mungkin tidak tahu apa yang didengarkan dokter tersebut pada tubuh kita.

Seorang dokter profesional yang terlatih menggunakan stetoskop bisa mengetahui kondisi kesehatan pasien hanya dengan mendengarkan suara jantung, paru-paru, perut, maupun organ-organ lain. mengindikasikan apakah fungsi organ tubuh pasien normal atau abnormal.

Stetoskop bukan hanya memeriksa organ seperti jantung dan organ pernafasan, stetoskop juga digunakan untuk mendengarkan intestine serta aliran darah dalam vena dan arteri. Fungsi stetoskop kurang lebih adalah untuk memeriksa keadaan paru-paru, jantung, tekanan darah, pemeriksaan prenatal dan gangguan perut.

Keuntungan menggunakan stetoskop adalah professional medis dapat lebih mudah mendengar suara yang dihasilkan organ terutama jantung, paru-paru dan usus. Fungsi yang abnormal dalam tubuh dapat dengan segera diketahui dengan penggunaan stetoskop yang tepat. Penggunaan yang salah dari stetoskop dapat menyebabkan kerugian seperti rusaknya telinga pendengar karena suara yang terlalu keras. Stetoskop juga harus sering dibersihkan. Penggunaan stetoskop langsung bersentuhan dengan kulit sehingga bias menyebarkan virus dan kuman.

Stetoskop akustik merupakan stetoskop yang paling banyak digunakan oleh professional medis. Cara kerja stetoskop ini adalah dengan menyalurkan suara yang berasal dari bagian dada ke telinga pendengar melalui tabung kosong yang berisi udara. Bagian yang disebut "*chestpiece*" bias diletakan dibadan pasien yang digunakan untuk memperjelas suara. Bagian tersebut biasanya terdiri dari dua sisi. Sebuah "*bell*" atau "*diaphragm*" juga diletakan pada tubuh pasien. "*Diaphragm*" akan bergetar karena suara dari dalam tubuh dan selanjutnya tercipta tekanan gelombang akustik. Gelombang akustik inilah yang berjalan ke telinga pendengar. Kinerja "*bell*" sama dengan "*diaphragm*". Getaran dari tubuh pasien akan menghasilkan tekanan gelombang akustik yang akhirnya sampai ke telinga pendengar. Perbedaan "*bell*" dan "*diaphragm*" terletak pada frekuensi suara yang

disalurkan. Diaphragm menyalurkan suara dengan frekuensi rendah. Stetoskop ini disebut juga sebagai stetoskop dua sisi. Pada awal abad ke-20 stetoskop ini diciptakan oleh Sprague dan Rappaport. Kelemahan dari stetoskop ini adalah tingkat suara yang sangat rendah dan menyebabkan sulit dalam mendiagnosis.

#### 4. Rangkaian Filter

Pada dasarnya, sebuah filter listrik adalah sirkuit yang dapat dirancang untuk memodifikasi, membentuk kembali atau menolak semua frekuensi yang tidak diinginkan dari sinyal listrik dan menerima atau lulus hanya mereka sinyal dicari oleh desainer sirkuit. Dengan kata lain mereka "filter-out" sinyal yang tidak diinginkan dan filter yang ideal akan memisahkan dan melewatkan sinyal masukan sinusoidal berdasarkan frekuensi mereka.

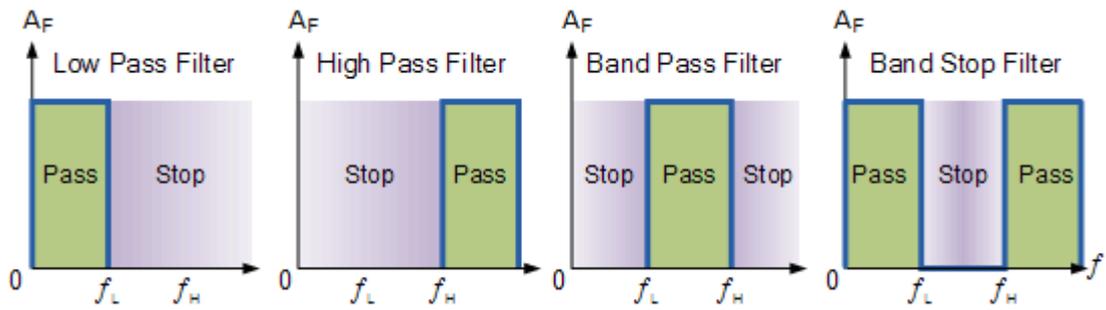
Dalam aplikasi frekuensi rendah (sampai 100kHz), filter pasif umumnya dibangun menggunakan sederhana RC (Resistor-Capacitor) jaringan, sedangkan filter frekuensi yang lebih tinggi (di atas 100kHz) biasanya terbuat dari RLC (Resistor-Induktor-Capacitor)

Filter pasif terdiri dari komponen pasif seperti resistor, kapasitor dan induktor dan tidak memiliki unsur penguatan (transistor, op-amp, dll) sehingga tidak memiliki gain sinyal, oleh karena itu tingkat *output* mereka selalu kurang dari input filter sehingga dinamai sesuai dengan rentang frekuensi dari sinyal bahwa mereka memungkinkan untuk melewati mereka, sementara memblokir atau "pelemahan" sisanya. Desain filter yang paling sering digunakan adalah:

- a. *Low pass Filter* - *low pass* filter hanya memungkinkan sinyal frekuensi rendah dari 0Hz ke frekuensi cut-off nya, titik  $f_c$  untuk lulus sementara memblokir mereka lebih tinggi.
- b. *High pass Filter* - *pass filter* tinggi hanya memungkinkan sinyal frekuensi tinggi dari frekuensi cut-off nya, titik  $f_c$  dan lebih tinggi hingga tak terbatas untuk melewati sementara memblokir mereka lebih rendah.
- c. *Band Pass Filter* - *band pass* filter memungkinkan sinyal yang masuk dalam setup pita frekuensi tertentu antara dua titik untuk melewati sementara memblokir baik rendah dan lebih tinggi frekuensi kedua sisi pita frekuensi ini.

Sederhana Pertama-order filter pasif (order 1) dapat dibuat dengan menghubungkan bersama-sama satu resistor dan satu kapasitor secara seri di sinyal input, ( $V_{in}$ ) dengan *output* dari filter, ( $V_{out}$ ) diambil dari persimpangan dua ini komponen. Tergantung pada cara sekitar kita menghubungkan resistor dan kapasitor berkaitan dengan sinyal *output* menentukan jenis konstruksi penyaring menghasilkan baik Filter *Low pass* atau Filter *High pass*.

Sebagai fungsi filter setiap adalah untuk memungkinkan sinyal dari sebuah band diberikan frekuensi untuk lulus tidak berubah sementara pelemahan atau melemahkan semua orang lain yang tidak ingin, kita dapat mendefinisikan karakteristik respon amplitudo dari sebuah filter yang ideal dengan menggunakan kurva respon frekuensi ideal empat jenis filter yang dasar seperti yang ditunjukkan. Gambar 9 menunjukkan kurva ideal respon berbagai macam filter.



Gambar 9. Kurva Ideal Respon Berbagai Macam Filter

(dikutip dari: <http://www.tespenku.com/2018/01/low-pass-filter-lpf-filter-pasif-rc.html>. 29 November 2018).

Filter dapat dibagi menjadi dua jenis yang berbeda: filter aktif dan filter pasif. Filter aktif berisi memperkuat perangkat untuk meningkatkan kekuatan sinyal sementara pasif tidak mengandung memperkuat perangkat untuk memperkuat sinyal. Karena ada dua komponen pasif dalam filter pasif desain sinyal *output* memiliki amplitudo kecil dari sinyal input yang sesuai, oleh karena itu filter RC pasif melemahkan sinyal dan memiliki keuntungan kurang dari satu, (kesatuan).

#### 5. Macam-macam rangkaian filter :

Berdasarkan sifat penguatannya, filter bisa diklasifikasikan :

##### a. Filter aktif

- 1) Komponen penyusunnya : ohm-Amp,kapasitor,dan resistor.
- 2) Keuntungannya : ukurannya yang lebih kecil, ringan, lebih murah, dan lebih fleksibel dalam perancangannya.
- 3) Kekurangan : kebutuhan catu daya eksternal,lebih sensitif terhadap perubahan lingkungan,dan frekuensi kerja yang sangat dipengaruhi oleh karakteristik komponen aktifnya.

b. Filter pasif :

- 1) Komponen penyusunnya : induktor,kapasitor,dan resistor.
- 2) Kelebihan : dapat dipergunakan untuk frekuensi tinggi.
- 3) Kekurangan : dimensi lebih besar daripada filter aktif.

c. Berdasarkan daerah frekuensi yang dilewatkan, filter analog dibagi menjadi:

- 1) LPF (*Low pass Filter*)
- 2) BPF (*Band Pass Filter*)
- 3) HPF (*High pass Filter*)
- 4) BSF/BRF (*Band Stop Filter/Band Reject Filter*)

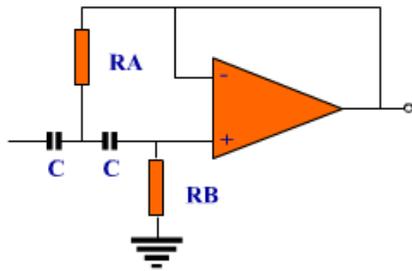
d. Berdasarkan bentuk respon frekuensi terhadap gain:

- 1) Filter *Bessel (Maximally Flat Time Delay)*
- 2) Filter *Cauer (Eliptic)*
- 3) Filter Butterworth (*Maximally Flat*)
- 4) Filter Chebyshev (*Tchebycheff*)

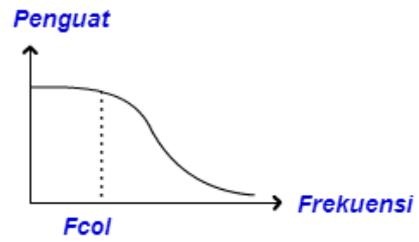
Filter-filter tersebut merupakan dasar untuk mendesain bermacam-macam kegunaan yang kita kenal dalam kehidupan sehari-hari yaitu :*equalizer,crossover,* dan lain-lain.

Rangkaian filter aktif tersebut dapat *low pass filter,band pass filter,band pass filter,dan high pass filter.* Rangkaian berikut adalah gambar macam-macam rangkaian filter dan respon frekuensinya. Gambar 10, 11 12, 13 menunjukkan macam macam rangkaian filter.

a. Rangkaian *low pass filter*



Rangkaian low pass filter

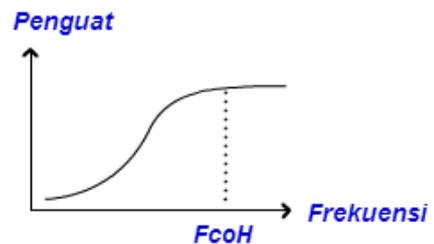
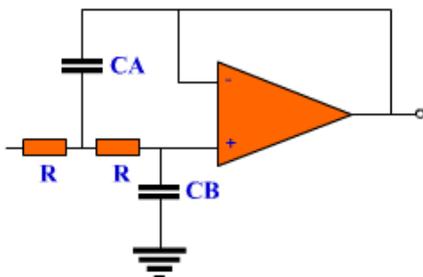


Frekuensi respon low pass filter

Gambar 10. Rangkaian *low pass filter*

(dikutip dari: [https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam\\_filter.htm](https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam_filter.htm). 25 November 2018).

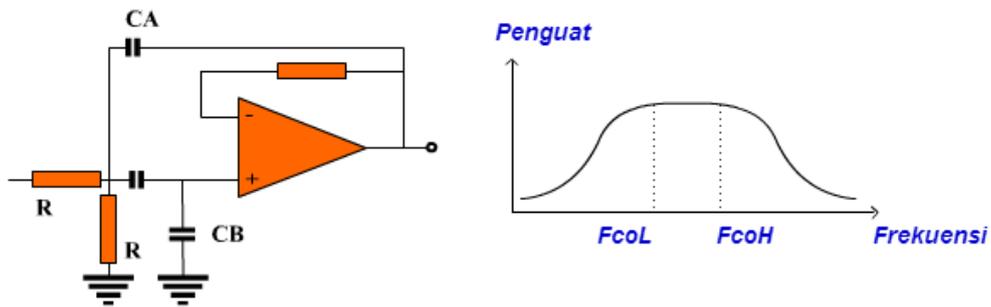
b. Rangkaian *high pass filter*



Gambar 11. Rangkaian *high pass filter*

(dikutip dari: [https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam\\_filter.htm](https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam_filter.htm). 25 November 2018).

c. Rangkaian *band pass* filter

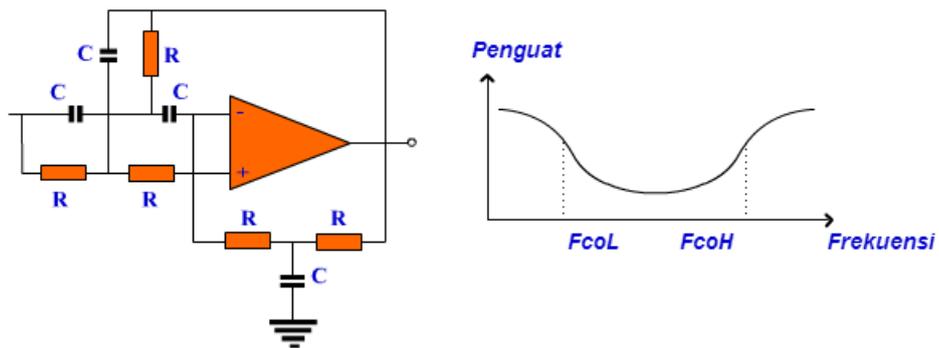


Gambar 12. Rangkaian *band pass* filter

(dikutip dari: [https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam\\_filter.htm](https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam_filter.htm).

25 November 2018).

d. Rangkaian *band stop* filter



Gambar 13. Rangkaian *band stop* filter

(dikutip dari: [https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam\\_filter.htm](https://m-edukasi.kemdikbud.go.id/medukasi/produk-files/kontenonline/online2007/rangkaianfilter/macam_filter.htm).

25 November 2018).

Kecuraman peredaman pada kurva frekuensi respon ditentukan oleh orde dari filter yang digunakan.

- Orde 1 memiliki kecepatan redaman (Roll-off) 6 db/Oktav
- Orde 2 memiliki kecepatan redaman (Roll-off) 12 db/Oktav
- Orde 3 memiliki kecepatan redaman (Roll-off) 18 db/Oktav

## **BAB III**

### **KONSEP RANCANGAN**

#### **A. Identifikasi Kebutuhan**

Untuk merancang Sistem Cerdas Dekteksi Suara untuk Pengklasifikasian Penyakit Jantung Menggunakan JST, harus diperhatikan beberapa komponen yaitu:

##### **1. *Hardware***

- a. alat perekam suara jantung sebagai alat pendukung
- b. stetoskop untuk mengambil data yang akan diuji
- c. penghubung antara alat perekam suara jantung dan komputer sebagai transfer data

##### **2. *Software***

- a. sebuah program untuk mengklasifikasi penyakit jantung
- b. sistem cerdas untuk pengklasifikasian penyakit jantung
- c. media penampil agar *user* dapat berinteraksi langsung dengan sistem.

#### **B. Analisis Kebutuhan**

Berdasarkan identifikasi kebutuhan diatas, maka diperoleh beberapa analisis kebutuhan terhadap Sistem Cerdas Dekteksi Suara untuk Pengklasifikasian Penyakit Jantung Menggunakan JST yang akan dirancang.

##### **1. *Hardware***

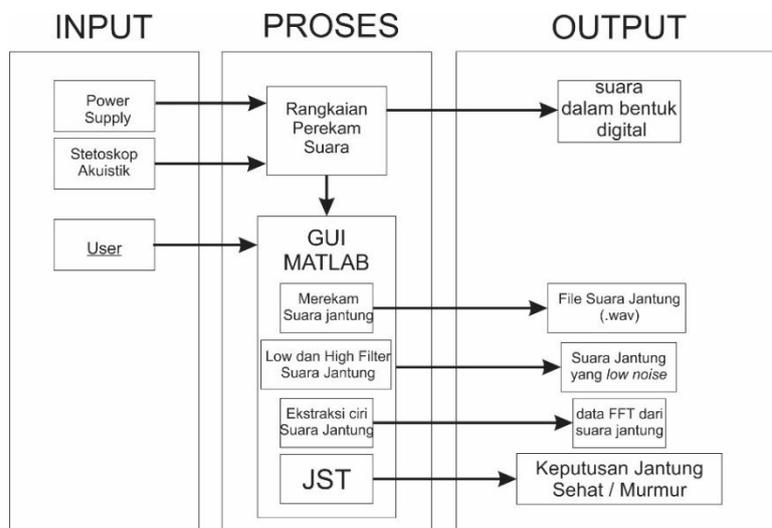
- a. Modul perekam suara MAX 9812 digunakan untuk mendapatkan suara jantung dari stetoskop.

- b. Modul penstabil tegangan *step-down DC to DC* dengan IC LM2596, untuk menstabilkan tegangan dari PC ke modul perekam suara MAX 9812
- c. Rangkaian filtering suara tambahan, sebagai filtering suara jantung di kisaran 20 Hz sampai 1000 Hz
- d. Stetoskop untuk mengambil suara jantung pasien.
- e. Kabel audio yang telah dimodifikasi sebagai media transfer data dari alat menuju PC

**2. Software**

- a. *Software* Matlab merupakan *software* yang digunakan untuk membuat aplikasi pengklasifikasian penyakit jantung.
- b. Menggunakan fungsi MATLAB JST (Jaringan Saraf Tiruan) yang dapat membuat sistem cerdas pengklasifikasian penyakit jantung.
- c. Menggunakan fungsi Matlab GUI yang memudahkan dalam pembuatan media penampil yang dapat berinteraksi langsung dengan *user*.

**C. Perancangan Alat**



Gambar 14. Blok Diagram Rangkaian

Penjelasan mengenai diagram blok sistem pada gambar 14 di atas adalah sebagai berikut:

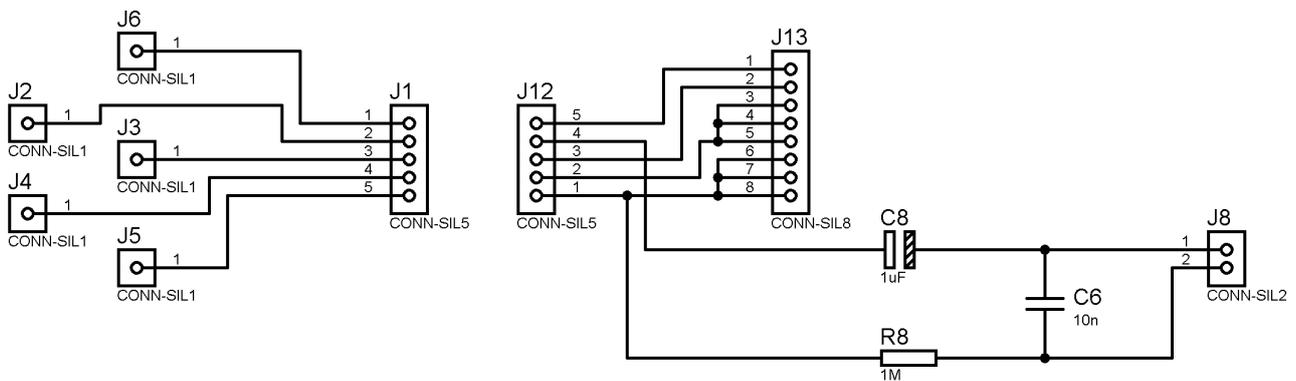
- 1) *Power supply* berfungsi sebagai input yang menyediakan tegangan bagi rangkaian perekam suara.
- 2) Stetoskop akustik berfungsi sebagai perekam suara jantung yang akan ditempelkan pada jantung pasien, yang nantinya akan menjadi input bagi rangkaian perekam suara.
- 3) *User* akan memberikan input di GUI matlab yang nantinya akan memproses setiap fungsi yang telah dirancang sebelumnya.
- 4) Rangkaian perekam suara akan merekam suara jantung dengan input dari *user* melalui GUI. Fungsi yang digunakan ialah merekam suara, fungsi tersebut akan merekam suara jantung melalui rangkaian perekam suara, dan memprosesnya untuk dijadikan file digital dalam format (.wav).
- 5) Fungsi *Low* dan *High Filter* Suara Jantung dalam GUI dapat memfilter suara yang telah direkam oleh rangkaian perekam suara, agar menjadi lebih bagus untuk dimasukkan kedalam proses selanjut nya
- 6) Fungsi Ekstraksi Ciri Suara Jantung dalam GUI ini akan membuat suara yang telah difilter sebelumnya, menjadi memiliki ciri tersendiri yang dapat dibedakan dengan suara lain nya, ekstraksi ciri ini menggunakan fungsi FFT (*Fast Fourier Transform*).
- 7) Fungsi JST (Jaringan Saraf Tiruan), dimana disini, suara-suara jantung yang telah di ekstraksi ciri dengan FFT, akan di latih dengan data jantung lain

nya, agar tercipta suatu sistem yang dapat membedakan antara suara jantung normal dengan suara jantung *murmur*.

### 1. *Hardware*

Berdasarkan blok diagram rangkaian maka dapat dijelaskan bagaimana Perancangan *Hardware* Sistem Cerdas Dekteksi Suara untuk Pengklasifikasian Penyakit Jantung Menggunakan JST.

*Power supply* menyediakan tegangan bagi rangkaian perekam suara yang di dalamnya terdapat modul perekam suara MAX9812, modul penstabil tegangan, dan stetoskop akuistik. Rangkaian perekam suara ini terdiri dari *female header*, konektor *stereo*, dan rangkaian komponen untuk *low pass filter*. Berikut adalah rancangan rangkaian untuk perekam suara jantung.



Gambar 15. Rangkaian Perekam Suara.

Pada gambar 15, merupakan rancangan rangkaian perekam suara jantung yang akan dibuat, dengan J1 adalah header untuk konektor *stereo*, J12 adalah header untuk modul perekam suara MAX9812, J13 adalah header untuk dihubungkan dengan modul penstabil tegangan, dan J8 adalah header untuk *output*

suara. Rangkaian diatas dilengkapi dengan rangkaian *low pass* filter, dengan *cut-off* sebesar 1500 Hz, yang mana diatas batas akhir frekuensi jantung *murmur* yang sebesar 1000 Hz.

Nilai frekuensi 1500 Hz diambil menimbang, untuk mengeliminasi *noise* yang dihasilkan lingkungan sekitar tanpa mengurangi detail-detail yang ada pada suara jantung yang terekam, maka diambil jarak potong frekuensi yang cukup jauh yaitu sekitar 500 Hz dari batas akhir frekuensi jantung *murmur*, dan nantinya juga, dalam GUI matlab yang dirancang, suara akan difilter lebih lanjut dengan menggunakan fungsi dalam matlab sendiri, sehingga dengan mengambil *cut-off* frekuensi di sekitar 1500 Hz, dianggap sudah mencukupi untuk rangkaian perekam suara, sebagai *low-pass* filter analog yang mengeliminasi *noise-noise* yang ada disekitar alat ketika proses perekaman jantung dimulai.

Komponen diatas yang merupakan bagian dari *low-pass* filter adalah R8 dan C6, dengan R8 dihubungkan secara seri dengan *Ground* rangkaian, dan C6 yang dihubungkan secara parallel dengan R8.

Menentukan nilai komponen untuk menghasilkan *cut-off* sebesar 1500 Hz adalah dengan rumus berikut :

$$f = \frac{1}{2\pi \times R \times C}$$

Keterangan :

F = Frekuensi *cut-off*

R = Nilai Resistor

C = Nilai Kapasitor

Dengan menentukan R dengan nilai yang mudah dicari yaitu 4700 Ohm, sehingga hanya perlu mencari nilai C dengan perhitungan sebagai berikut :

$$C = \frac{1}{2 \times 3.14 \times 1500 \times 4700}$$

$$C = 22.58 \times 10^{-9}$$

$$C \approx 22nF$$

Nilai C sudah ditemukan yaitu 22nF, sehingga nilai komponen untuk menghasilkan *cut-off* sebesar 1500 Hz adalah dengan R sebesar 4700 Ohm dan C sebesar 22 nF.

Fungsi dari C8 sendiri yang mempunyai besaran 1uF adalah dengan mengikuti *datasheet* modul yang jika ingin modul dihubungkan langsung menggunakan kabel *stereo*, maka harus ada kapasitor yang berfungsi sebagai pelindung *sound driver stereo* dari tegangan DC *bias* yang dihasilkan modul.

## 2. *Software*

Berdasarkan blok diagram pada gambar 12 maka dapat dijelaskan bagaimana Perancangan *Software* Sistem Cerdas Dekteksi Suara untuk Pengklasifikasian Penyakit Jantung Menggunakan JST.

*User* berperan untuk memasukkan input di GUI matlab yang telah dirancang sebelumnya, sedangkan didalam GUI matlab nya sendiri terbagi menjadi 4 fungsi utama, yaitu merekam suara jantung, *high* and *low* filter suara jantung, ekstraksi ciri suara jantung, dan JST. Pada fungsi pertama, jika dipilih, GUI akan menerima sinyal suara jantung dari rangkaian perekam suara, yang nantinya keluaran dari GUI ini adalah file audio dengan format (.wav). Pada fungsi kedua, jika dipilih

akan memfilter suara yang telah direkam sebelumnya, suara ini akan difilter dengan *high pass* filter dan *low pass* filter, dengan rentang frekuensi yang diloloskan antara 20 hingga 1000 Hz . Pada fungsi ketiga jika dipilih, GUI akan melakukan ekstraksi ciri pada suara yang telah difilter sebelumnya, ekstraksi ciri ini menggunakan metode FFT yaitu *Fast Fourier Transform*, dimana suara akan diproses untuk dijadikan dalam rentangan spektrum frekuensi, rentangan inilah yang menjadi ciri setiap suara jantung. Pada fungsi keempat, jika dipilih, JST yang sebelumnya dilatihkan terlebih dahulu dengan data-data suara jantung, dapat mengenali antara suara jantung normal dengan *murmur*.

Berikut adalah Algoritma sitem dan program yang akan dirancang :

a. Algoritma sistem

- 1) Start
- 2) Menghubungkan rangkaian perekam suara ke PC
- 3) Menempatkan stetoskop pada dada pasien
- 4) Masuk ke GUI matlab di PC
- 5) Mulai perekaman suara jantung di GUI
- 6) Memfilter suara jantung yang telah direkam
- 7) Ekstraksi ciri suara jantung dengan FFT
- 8) Melatihkan hasil ekstraksi ciri suara jantung dengan JST yang sudah dibuat sebelumnya
- 9) Apakah data suara jantung yang dilatihkan sehat ?
- 10) Jika ya, GUI matlab akan menampilkan status jantung sehat
- 11) Jika tidak, GUI matlab akan menampilkan status jantung *murmur*

12) Finish

b. Algoritma program

1) Start .

2) Inisialisasi variable global GUI matlab.

3) *User* memilih pushbutton mana yang akan ditekan.

4) Jika *push button* 1 ditekan, mengeset  $F_s = 11025$ , dan waktu = 10 detik, setelah itu proses perekaman suara jantung akan dimulai, dan setelah selesai dalam waktu 10 detik, gelombang suara akan ditampilkan di GUI matlab.

5) Jika *push button* 2 ditekan, mengeset  $[y, F_s] = \text{audioread}(\text{File\_jantung})$ , setelah itu proses menampilkan suara jantung dimulai, dengan mengubah suara jantung ke domain frekuensi dan amplitudo, setelah selesai mengubahnya, gelombang suara jantung dan bentuk suara jantung dalam domain frekuensi dan amplitudo akan ditampilkan di GUI matlab.

6) Jika *push button* 3 ditekan, mengeset  $y = \text{hasil\_FFT\_suara\_jantung}$ , dan  $JST = \text{net}$ , setelah itu proses pengklasifikasian suara jantung dimulai, jika  $\text{output} == [1,0]$ , maka GUI matlab akan menampilkan tulisan “Jantung Normal”, sedangkan jika  $\text{output} == [0,1]$ , akan menampilkan tulisan “Jantung *Murmur*”.

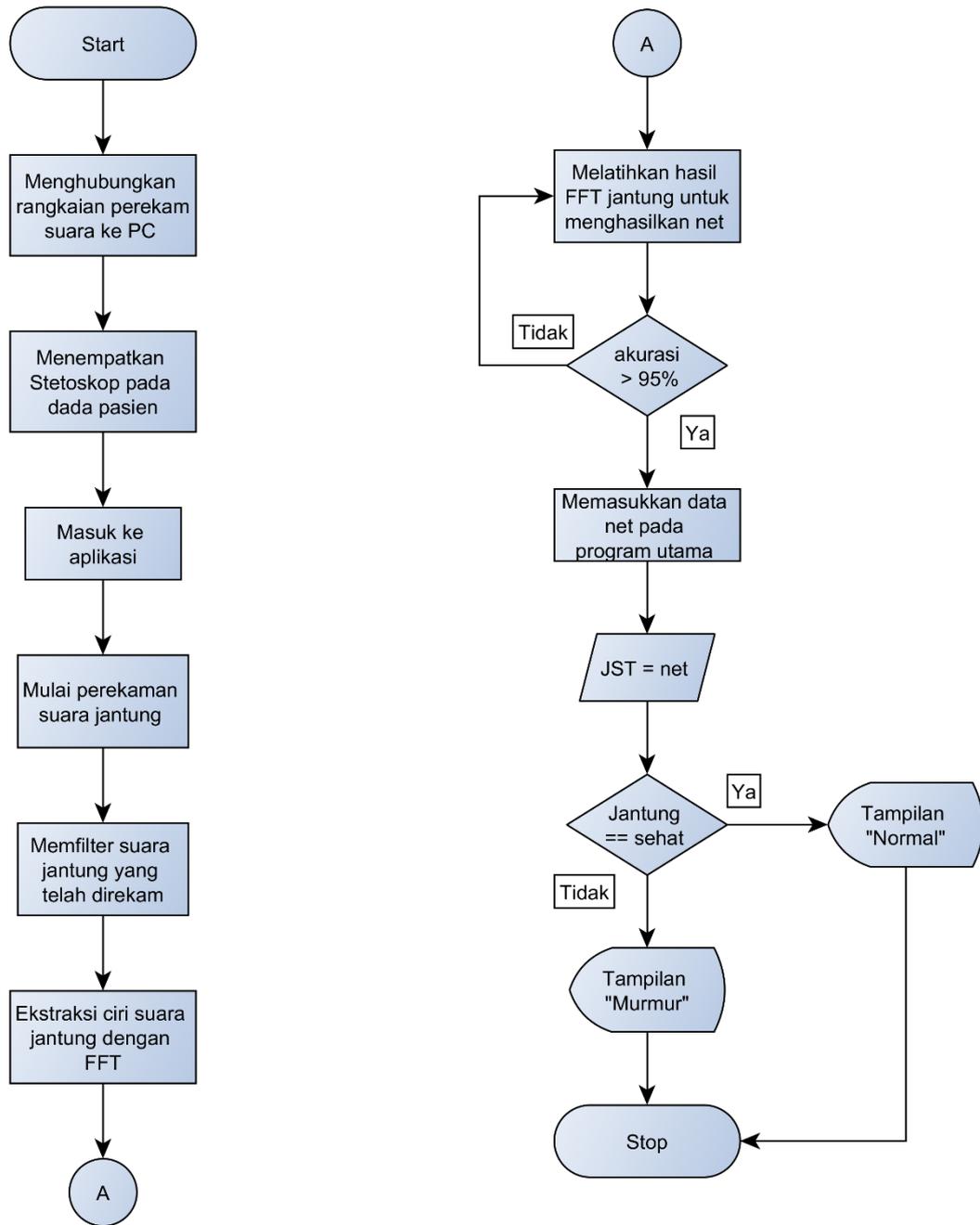
7) Jika pushbutton4 ditekan, mengeset  $y = \text{suara\_jantung}$ , dan filter = Bessel orde 10, setelah itu proses filter suara jantung dimulai, dengan menggunakan filter untuk menjalankan fungsi *band pass*, dan memasukkan fungsi untuk memotong frekuensi, setelah suara jantung terfilter, GUI matlab akan

menampilkan perbandingan suara jantung sebelum dan setelah difilter dalam domain frekuensi dan amplitudo.

- 8) Jika *push button* 5 ditekan, mengeset  $y = \text{suara\_jantung\_setelah\_direkam}$ , setelah itu  $y$  akan diputar.
- 9) Jika *push button* 6 ditekan, mengeset  $y = \text{suara\_jantung\_setelah\_direkam}$ , setelah itu proses penyimpanan suara jantung setelah direkam dimulai, dengan format penyimpanan file suara adalah wav.
- 10) Jika *push button* 7 ditekan, proses *reset* program dimulai, dengan menghapus semua variable yang telah ada sebelumnya.
- 11) Jika *push button* 8 ditekan, mengeset  $y = \text{suara\_jantung\_setelah\_difilter}$ , setelah itu  $y$  akan diputar.
- 12) Jika *push button* 9 ditekan, mengeset  $y = \text{suara\_jantung\_setelah\_difilter}$ , setelah itu proses penyimpanan suara suara jantung setelah difilter dimulai, dengan format penyimpanan file suara adalah wav.
- 13) Jika *push button* 10 ditekan, GUI matlab akan menampilkan “Apakah *user* ingin kembali ke menu utama”, setelah itu tampilan akan dialihkan ke menu utama.
- 14) Finish

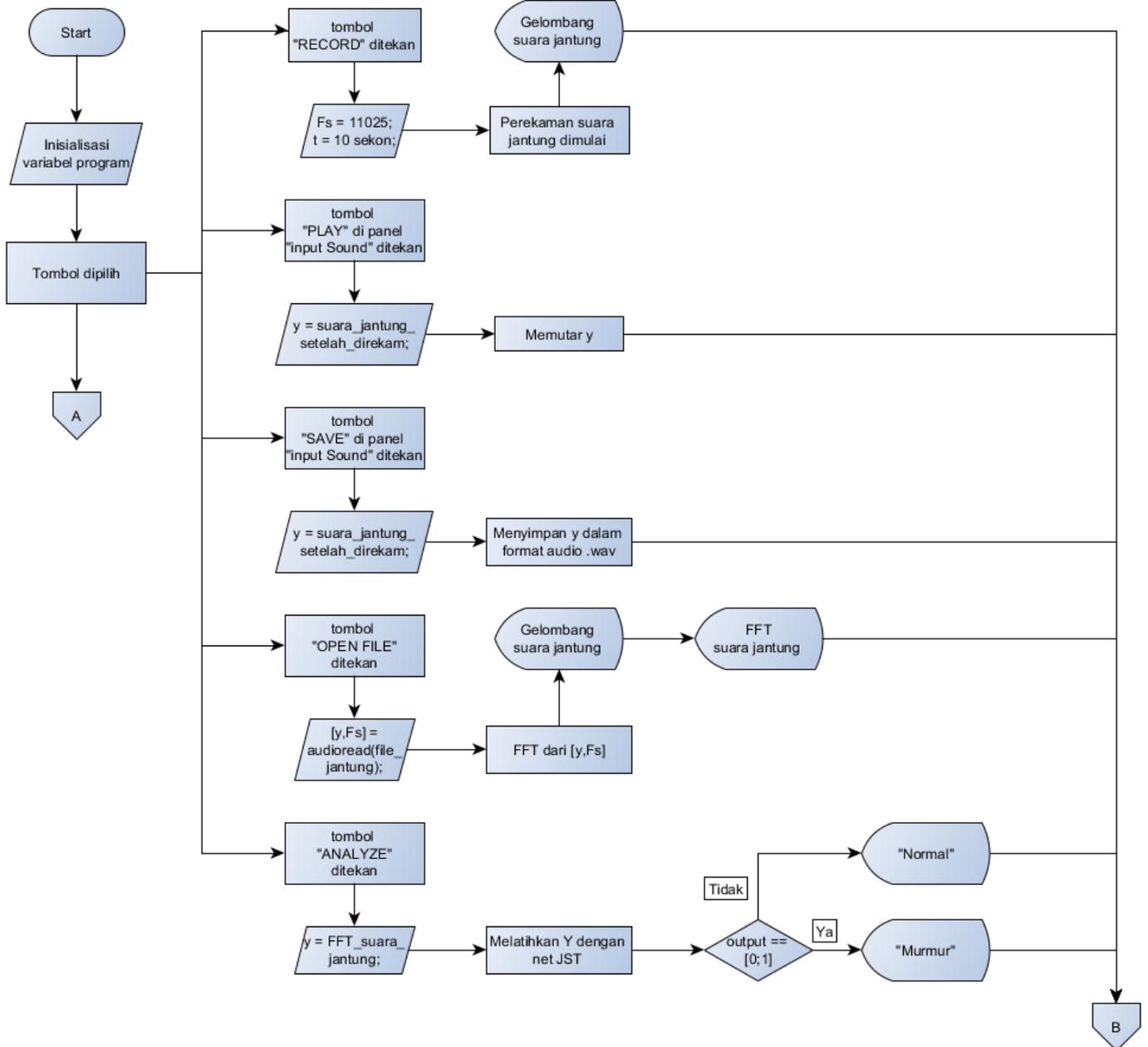
Selanjutnya adalah *flowchart* Sistem dan Program yang akan dirancang :

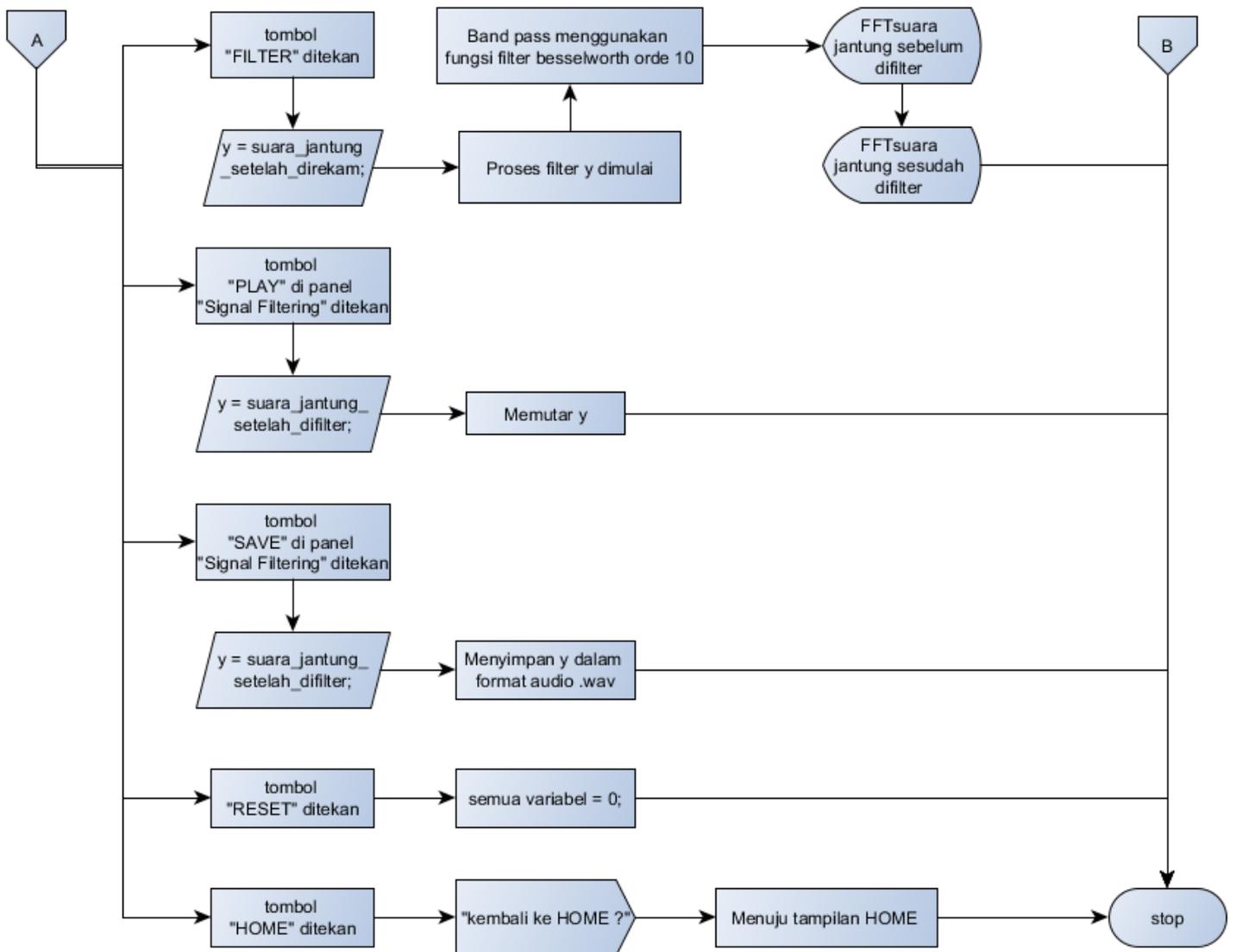
c. *Flowchart Sistem*



Gambar 16. *Flowchart Sistem*

d. Flowchart Program





Gambar 17. Flowchart Program

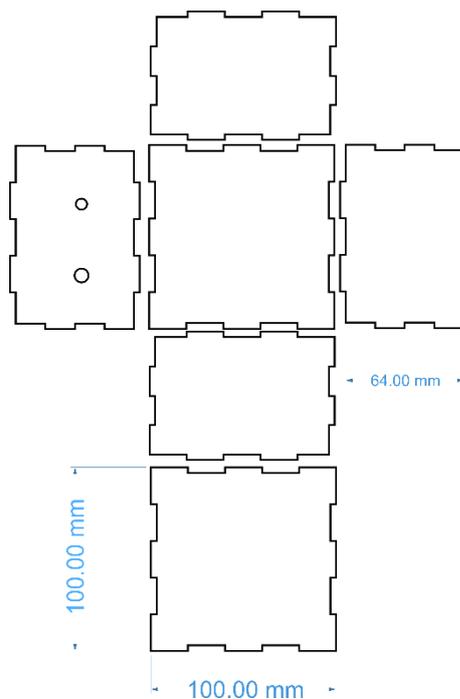
#### D. Pembuatan Alat

Pemubuatan alat dari perancangan proyek akhir ini terdiri dari 2 bagian, yaitu *Hardware* dan *Software*. Berikut penjelasannya :

##### 1. *Hardware*

Pada bagian ini, rangkaian yang akan dibuat adalah rangkaian perekam suara, yang terdiri dari modul perekam suara MAX9812, regulator tegangan *step-down DC to DC* LM2596, dan rangkaian *low-pass* filter. Langkah pembuatan dalam merangkai rangkaian perekam suara jantung yaitu pertama-tama dengan mendesain box yang nantinya menjadi tempat untuk meletakkan modul perekam suara, dan regulator tegangan.

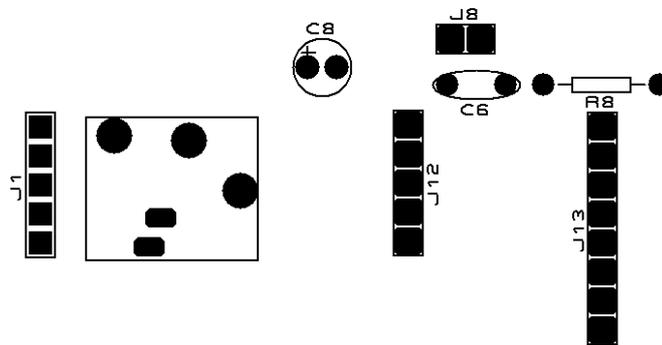
Berikut adalah gambar desain kotak yang telah dirancang :



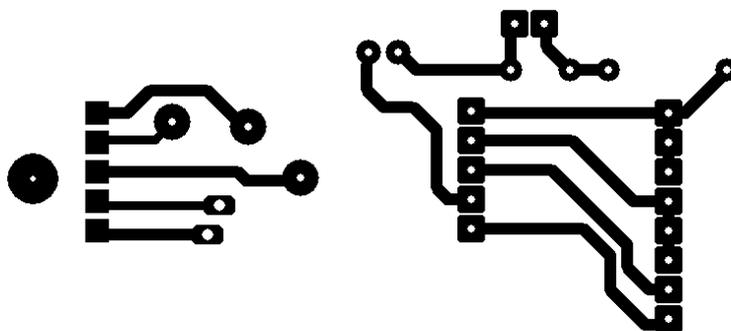
Gambar 18. Desain Box Perekam Suara Jantung

Gambar 18 merupakan desain box alat perekam suara jantung, didesain dengan menyesuaikan ukuran modul perekam suara MAX9812, modul regulator, dan rangkaian *low-pass* filter suara agar dapat dimasukkan di dalamnya. Dengan dimensi panjang x lebar x tinggi adalah 10 cm x 10 cm x 6,4 cm.

Setelah desain box perekam alat suara di atas, selanjutnya adalah desain rangkaian perekam suara yang akan menghubungkan modul perekam suara MAX9812, dengan *Low-pass* filter, berikut merupakan layout tampak bawah dan atas rangkaian perekam suara.



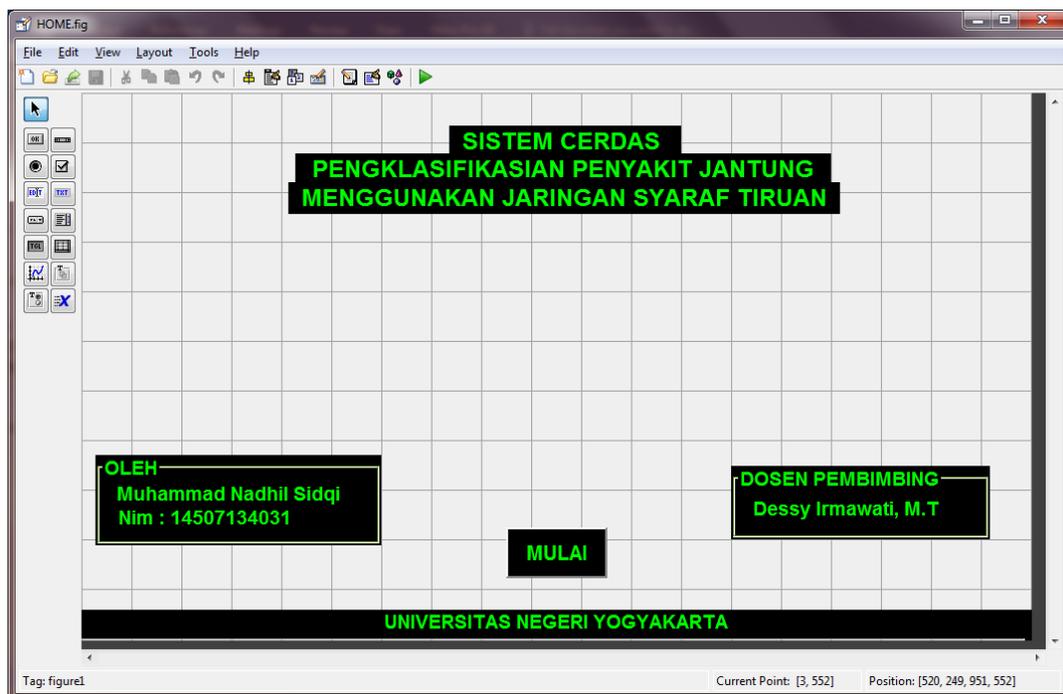
Gambar 19. *Layout* Rangkaian Perekam Suara Tampak Atas



Gambar 20. *Layout* Rangkaian Perekam Suara Tampak Bawah

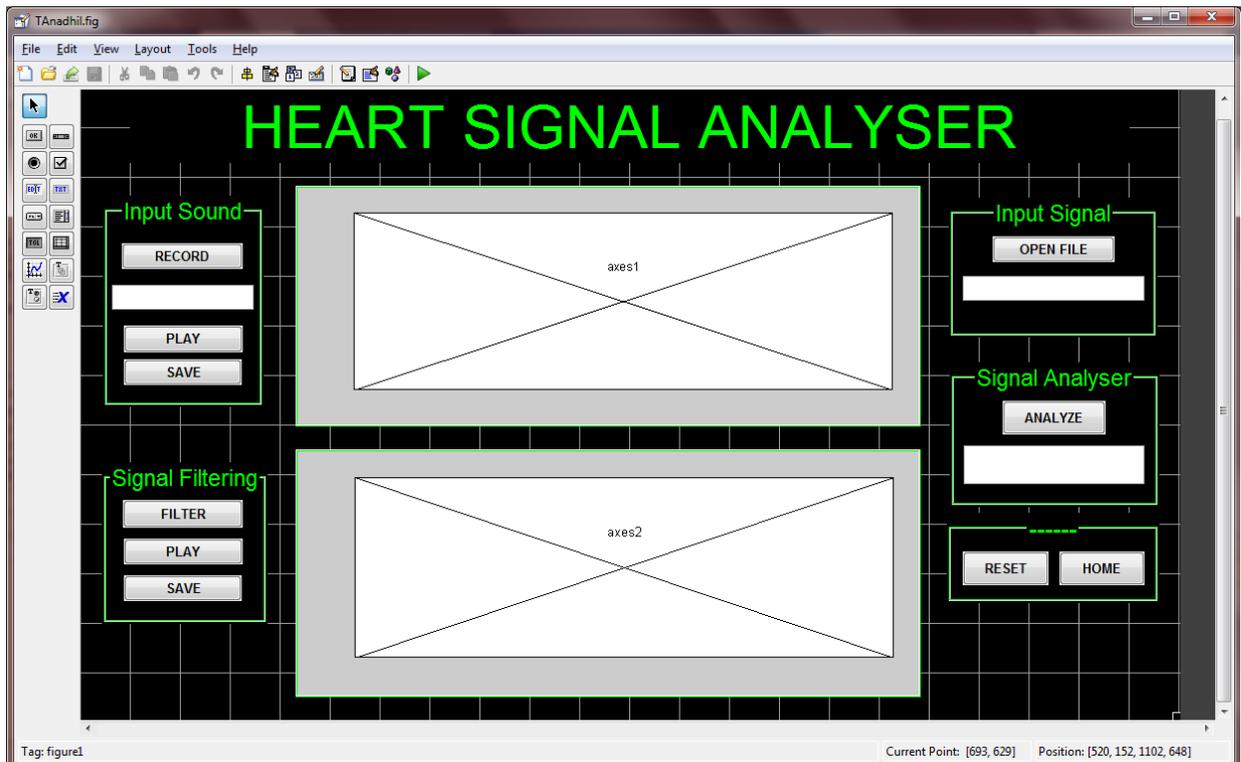
## 2. Software

Pembuatan aplikasi matlab dibagi menjadi 2, yaitu, tampilan GUI HOME, dan tampilan GUI utama. Tampilan GUI pembuka dibuat sebagai pembuka tampilan sebelum menuju ke tampilan GUI utama. Berikut adalah desain tampilan GUI HOME, dan tampilan GUI utama.



Gambar 21. Tampilan GUI HOME

Pada tampilan GUI HOME diatas, dirancang menggunakan fungsi “*guide*” pada matlab, yang nantinya akan diarahkan menuju menu pembuatan GUI. Komponen-komponen yang digunakan diatas hanya *push button* “MULAI” yang ketika di klik akan membuka jendela baru yaitu tampilan GUI utama.



Gambar 22. Tampilan GUI utama

Pada tampilan GUI utama diatas menggunakan berbagai macam komponen penyusun, yaitu: “Push Button”, “Edit Text”, “Axes”, dan “Panel”. Penamaan komponen-komponen diatas jika disesuaikan dengan rancangan *software* akan menjadi sebagai berikut:

- a. Tombol “*RECORD*” dalam panel “*Input Sound*” merupakan pushbutton1
- b. Tombol “*PLAY*” dalam panel “*Input Sound*” merupakan pushbutton5
- c. Tombol “*SAVE*” dalam panel “*Input Sound*” merupakan pushbutton6
- d. Tombol “*FILTER*” dalam panel “*Signal Filtering*” merupakan pushbutton4
- e. Tombol “*PLAY*” dalam panel “*Signal Filtering*” merupakan pushbutton8
- f. Tombol “*SAVE*” dalam panel “*Signal Filtering*” merupakan pushbutton9

- g. Tombol “*OPEN FILE*” dalam panel “*Input Signal*” merupakan pushbutton2
- h. Tombol “*ANALYZE*” dalam panel “*Signal Analyser*” merupakan pushbutton3
- i. Tombol “*RESET*” merupakan pushbutton7
- j. Tombol “*HOME*” merupakan pushbutton10

Penambahan komponen “Edit Text” pada panel “*Input Sound*” berfungsi untuk menunjukkan status perekaman suara jantung, komponen ini akan menampilkan tulisan “start *recording*...” ketika tombol “*RECORD*” ditekan, dan dalam waktu 10 detik tulisan akan berganti “end of *recording*”.

Penambahan komponen “Edit Text” pada panel “*Input Signal*” berfungsi untuk menunjukkan nama file yang diambil untuk diproses lebih lanjut.

Penambahan komponen “Edit Text” pada panel “*Signal Analyser*” berfungsi untuk menampilkan status jantung yang diuji dengan menampilkan tulisan “Normal” jika jantung normal dan “*Murmur*” jika jantung *murmur*.

Penambahan “Axes” berfungsi untuk menampilkan gelombang suara jantung dan data FFT dari suara jantung yang telah diekstraksi ciri.

## **E. Pengujian Alat**

Pengujian alat dilakukan untuk mendapatkan data penelitian dengan menggunakan dua buah uji, yaitu.

### 1. Uji Fungsional

Pengujian ini dilakukan dengan cara menguji setiap bagian alat berdasarkan karakteristik dan fungsi masing-masing. Pengujian ini dilakukan untuk mengetahui apakah setiap bagian dari perangkat telah berkerja sesuai dengan fungsi dan keinginan

a. *Hardware*

1) Pengujian tegangan catu daya

Tabel 1. Tabel pengukuran tegangan modul regulator tegangan

No	Vin (Volt)	V output (Volt)	
		Tanpa beban	Dengan beban
1.			

2. Pengujian Rangkaian Filter (*low pass*) Perekam Suara

Tabel 2. Tabel Pengujian Rangkaian Filter

Jenis Filter	Filter ( <i>low pass</i> ) 1000 Hz
Tanpa Filter	
Dengan Filter	

b. *Software*

Pengujian GUI Home

Tabel 3. Pengujian GUI Home

Tombol	Tampilan GUI	Keterangan
MULAI		

Pengujian GUI Utama

Tabel 4. Pengujian *Input Sound*

1) Panel *Input Sound*

Tombol	Tampilan GUI	Keterangan
<i>RECORD</i>		

<i>PLAY</i>		
<i>SAVE</i>		

Tabel 5. Pengujian *Sound Filtering*

2) Panel *Sound Filtering*

Tombol	Tampilan GUI	Keterangan
<i>FILTER</i>		

<i>PLAY</i>		
<i>SAVE</i>		

Tabel 6. Pengujian *Input Signal*

3) Panel *Input Signal*

Tombol	Tampilan GUI	Keterangan
<i>OPEN FILE</i>		

Tabel 7. Pengujian *Signal Analyser*

4) Panel *Signal Analyser*

Tombol	Tampilan GUI	Keterangan
<i>ANALYZE</i>		

Tabel 8. Pengujian *Options*

5) Panel *Options*

Tombol	Tampilan GUI	Keterangan
<i>RESET</i>		

<i>HOME</i>		
-------------	--	--

### Pengujian JST

Tabel 9. Pengujian JST

No.	Jumlah hidden layer	Hasil <i>training</i> JST	Hasil <i>net</i> JST pada data latih	Hasil <i>net</i> JST pada data uji

### F. Uji unjuk kerja

Pengujian unjuk kerja alat dilakukan dengan cara melihat unjuk kerja alat. Hal-hal yang perlu diamati antara lain: rangkaian perekam suara, GUI HOME, dan GUI utama.

a. Pengujian unjuk kerja alat

Tabel 10. Pengujian unjuk kerja alat

Tombol	Tampilan GUI	Keterangan
Merekam Suara Jantung		
Filter Suara Jantung		
Ekstraksi Ciri Suara Jantung		

JST		
-----	--	--

### G. Pengoperasian Alat

Pengoperasian alat ini dapat dilakukan dengan cara sebagai berikut.

1. Hubungkan kabel power USB dari alat ke PC.
2. Hubungkan kabel *stereo* dari alat ke PC.
3. Pastikan suara sudah dapat masuk di PC dengan mengecek kondisi microphone yang ada di PC.
4. Membuka aplikasi.
5. Klik tombol “MULAI”
6. Jika ingin merekam suara jantung, terlebih dahulu tempatkan stetoskop pada bagian jantung pasien.
7. Klik tombol “*RECORD*” pada panel “*Input Sound*” untuk mengambil data suara jantung pasien.
8. Jika ingin memainkan file yang telah terekam klik tombol “*PLAY*” pada panel yang sama.
9. Jika ingin menyimpan file yang telah terekam klik tombol “*SAVE*” pada panel yang sama

10. Klik tombol “*FILTER*” untuk memfilter suara yang telah terekam
11. Jika ingin mengambil file suara jantung yang sudah direkam klik tombol “*OPEN FILE*” untuk memasukkan file audio jantung.
12. Jika ingin memainkan file yang sudah terfilter, klik tombol “*PLAY*” pada panel “*Signal Filtering*”
13. Jika ingin menyimpan file yang sudah terfilter, klik tombol “*SAVE*” pada panel yang sama
14. Klik tombol “*ANALYZE*” untuk menentukan file yang ada, apakah termasuk jantung yang normal atau *murmur*.
15. Klik tombol “*RESET*” untuk membersihkan semua data yang ada dalam aplikasi
16. Klik tombol “*HOME*” untuk kembali ke HOME
17. Selesai

## BAB IV

### PENGUJIAN DAN PEMBAHASAN

Pengujian terhadap alat dilakukan untuk mengetahui kinerja dari masing-masing komponen yang digunakan. Pengujian ini diharapkan mendapatkan hasil yang baik dan komponen penyusun alat secara keseluruhan bekerja sesuai fungsinya.

#### A. Pengujian

##### 1. Pengujian tegangan *power supply*

Pengujian *power supply* dan baterai dibutuhkan karena merupakan sebuah rangkaian yang memberikan sumber tegangan ke seluruh rangkaian dalam sistem cerdas ini. Pengukuran catu daya dilakukan dengan cara diukur pada saat ada dan tanpa beban. Pengukuran ini dilakukan untuk memastikan tegangan yang dihasilkan *power supply* tidak melebihi kapasitas maksimal tegangan kerja pada seluruh rangkaian. Tabel 11 merupakan pengukuran tegangan *power supply* pada saat ada dan tanpa beban.

Tabel 11. Hasil pengujian tegangan *power supply*

No	Vin (Volt)	V output (Volt)	
		Tanpa beban	Dengan beban
1.	5	4.5	4.5

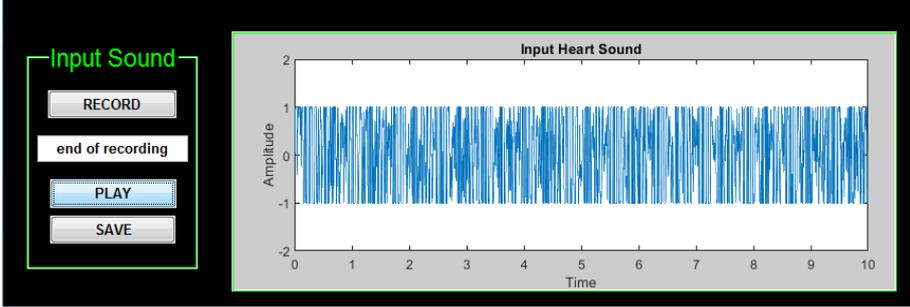
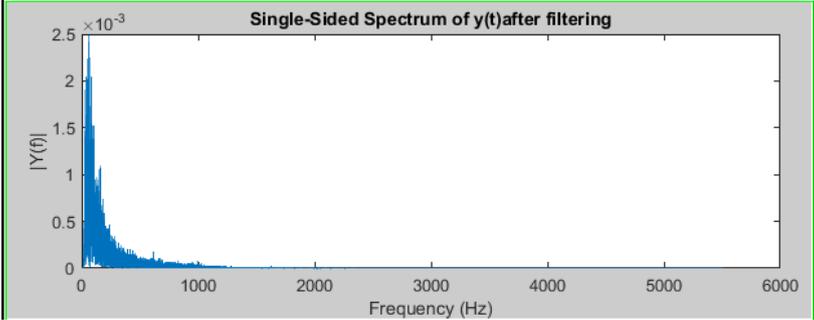
Tabel 11 menunjukkan hasil pengukuran tegangan untuk kondisi tanpa beban dan dengan beban. Modul ini mempunyai keluaran arus maximal sebesar 2 Ampere, dengan suhu operasi maksimal hingga mencapai 85° C. Regulator

berfungsi dengan normal dan sudah mengeluarkan tegangan yang sesuai dengan yang tertera di datasheet komponen. Input yang diberikan untuk modul ini adalah 5 V DC dari PC, kemudian tegangan diturunkan menjadi 4.5 V DC Kondisi tegangan seperti ini sudah dapat menyuplai tegangan dengan baik,

## 2. Pengujian Rangkaian Filter (*low pass*) Perekam Suara

Pengujian ini dibutuhkan untuk melihat, apakah rangkaian filter perekam suara pada hardware sudah bekerja dengan baik atau belum, dengan cara mengukur frekuensi suara sebelum di filter dan setelah di filter, yang mana akan menunjukkan hasil apakah frekuensi terfilter atau tidak pada rangkaian perekam suara ini.

Tabel 12. Hasil pengujian rangkaian filter perekam suara

Jenis Filter	Filter ( <i>low pass</i> ) 1000 Hz
Tanpa Filter	
Dengan Filter	

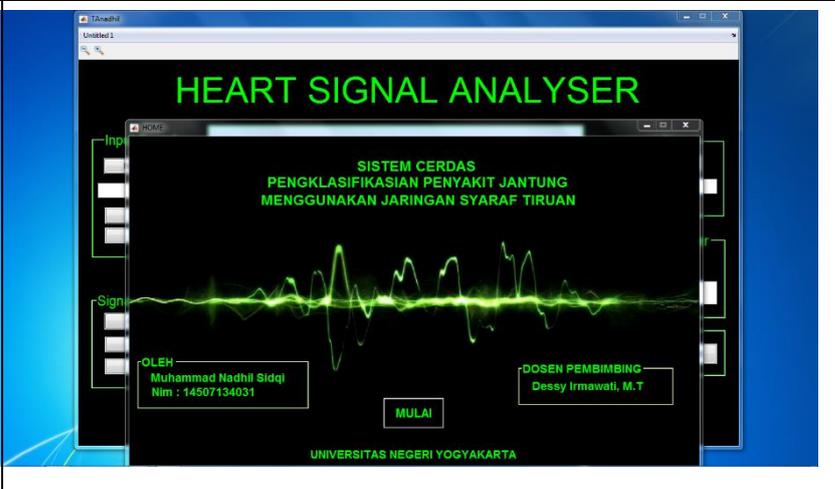
Tabel 12 merupakan hasil pengujian untuk perbandingan rangkaian perekam suara. Factor yang diuji adalah ketika perekam suara tanpa filter (*low pass*) dan dengan filter (*low pass*). *Output* yang dihasilkan sudah sesuai dengan kebutuhan perekaman suara. Ketika rangkaian perekam suara tanpa filter *output* frekuensi yang dihasilkan masih tinggi, ketika rangkaian perekam suara menggunakan filter *otput* yang frekuensi yang dihasilkan sudah sesuai dengan kebutuhan frekuensi suara dalam perekaman.

### 3. Pengujian Aplikasi

Pengujian ini dilakukan untuk melihat hasil perancangan *software*, apakah sudah bekerja dengan baik atau belum, dengan menghubungkan *output* perekam suara yang sudah terangkai sedemikian rupa.

#### a) Pengujian Gui Home

Tabel 13. Hasil pengujian button Gui Home

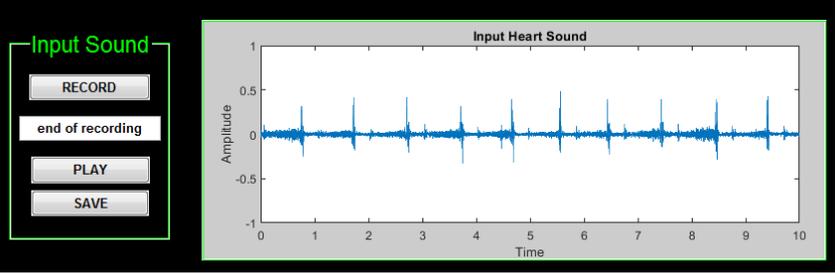
Tombol	Tampilan GUI	Keterangan
MULAI		Baik

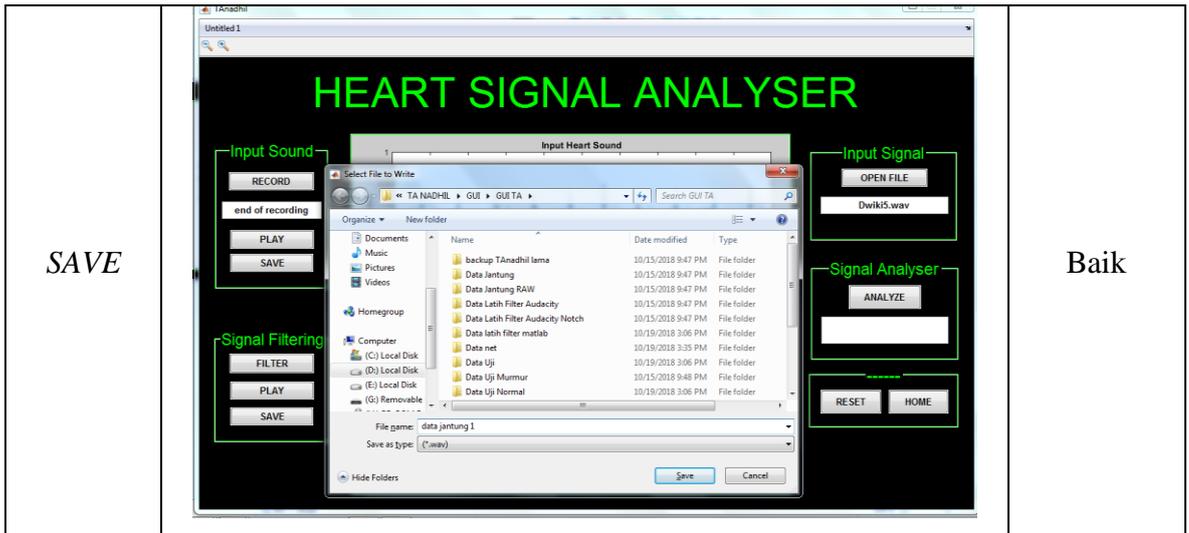
Tabel 13 menunjukkan hasil pengujian aplikasi Gui Home yang ketika *push button* “Mulai” ditekan maka akan akan masuk dari Gui Home ke Gui Utama, pengujian menunjukkan *push button* sudah berfungsi dengan baik.

b) Pengujian Gui Utama

Tabel 14. Hasil pengujian *Input Sound*

1) Panel *Input Sound*

Tombol	Tampilan GUI	Keterangan
<i>RECORD</i>		Baik
<i>PLAY</i>		Baik



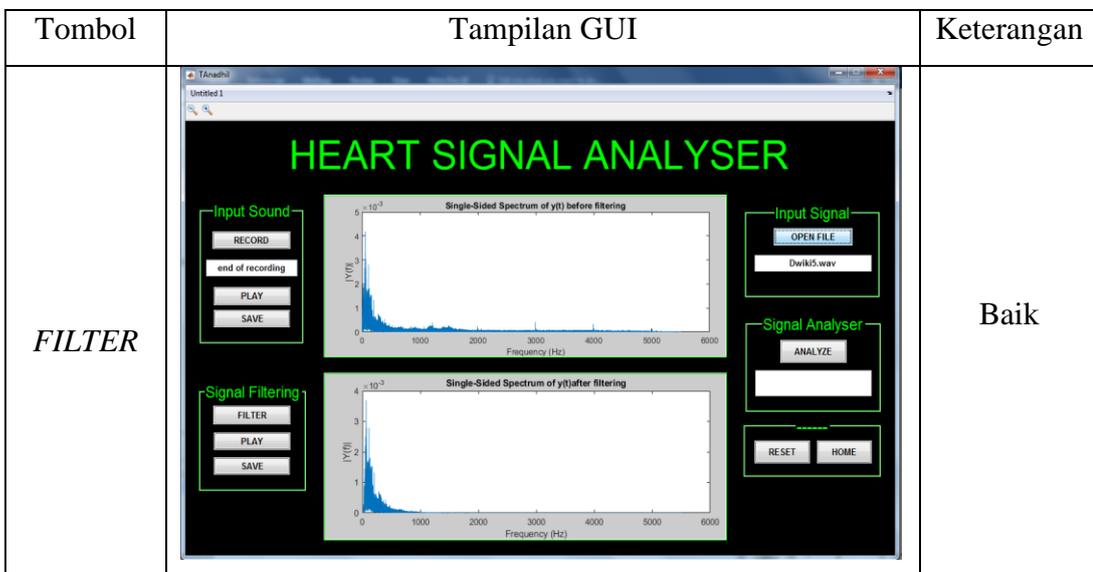
SAVE

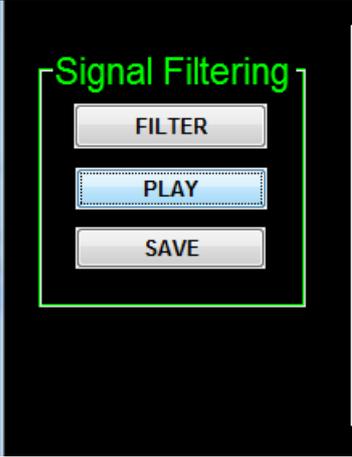
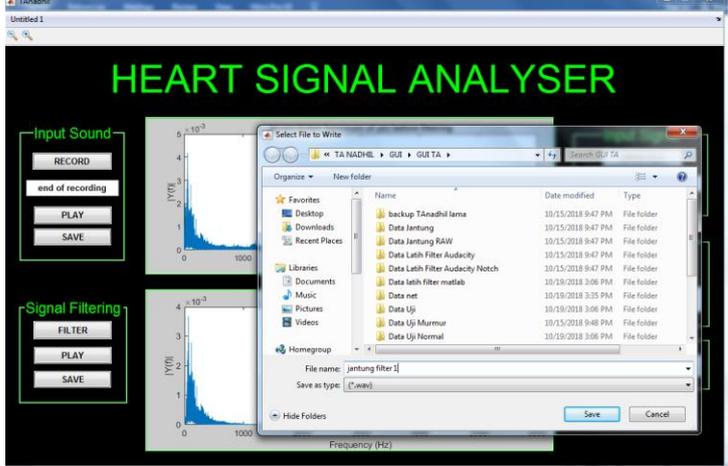
Baik

Tabel 14 menunjukkan hasil pengujian aplikasi pada panel *input sound* yang ketika *push button* “RECORD/PLAY/SAVE” ditekan akan bekerja sesuai dengan fungsinya, pengujian menunjukkan *push button* sudah berfungsi dengan baik.

Tabel 15. Hasil pengujian *Sound Filtering*

2) Panel *Sound Filtering*

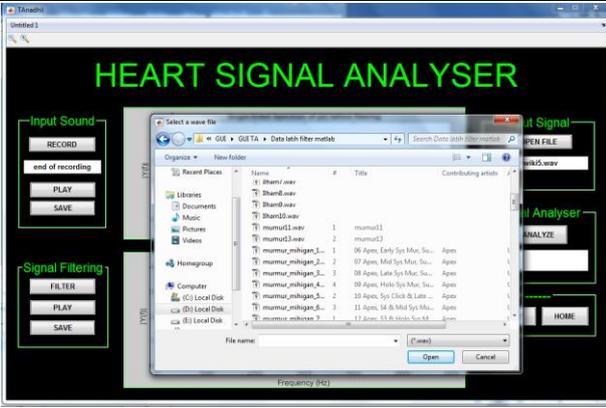


<p>PLAY</p>		<p>Baik</p>
<p>SAVE</p>		<p>Baik</p>

Tabel 15 menunjukkan hasil pengujian aplikasi pada panel *Sound Filtering* yang ketika *push button* “*FILTER/PLAY/SAVE*” ditekan akan bekerja sesuai dengan fungsinya, pengujian menunjukkan *push button* sudah berfungsi dengan baik.

Tabel 16. Hasil pengujian *Input Signal*

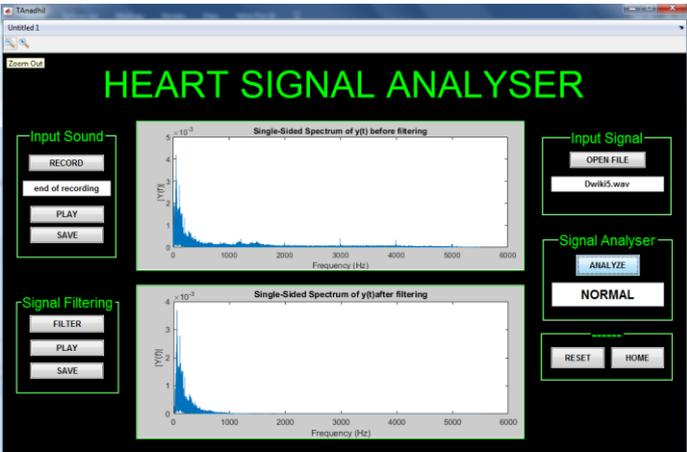
3) Panel *Input Signal*

Tombol	Tampilan GUI	Keterangan
<p><i>OPEN FILE</i></p>		<p>Baik</p>

Tabel 16 menunjukkan hasil pengujian aplikasi pada panel *Sound Filtering* yang ketika *push button* “*OPEN FILE*” ditekan akan bekerja sesuai dengan fungsinya, pengujian menunjukkan *push button* sudah berfungsi dengan baik.

Tabel 17. Hasil pengujian *Input Signal*

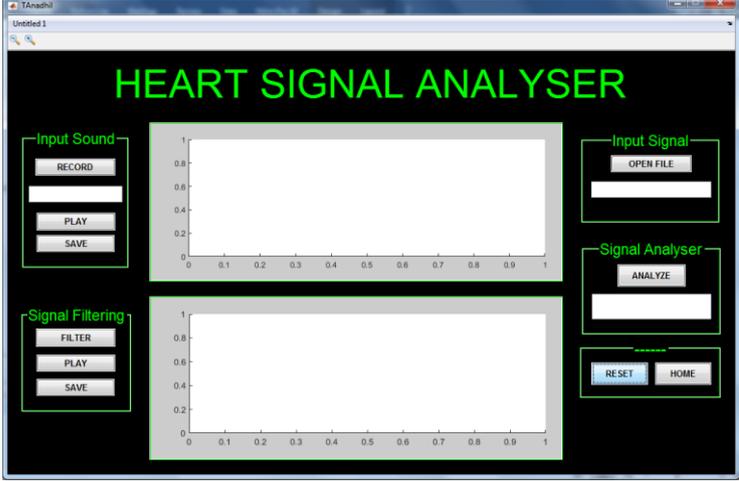
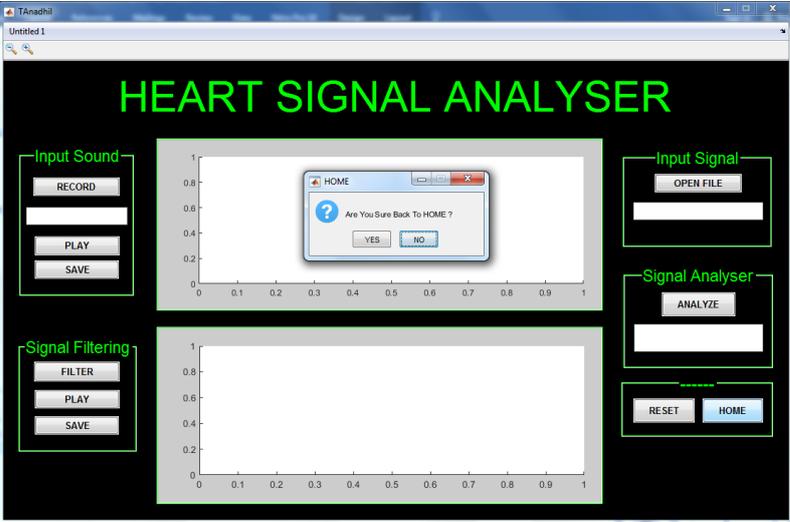
4) Panel *Signal Analyser*

Tombol	Tampilan GUI	Keterangan
<p><i>ANALYZE</i></p>		<p>Baik</p>

Tabel 17 menunjukkan hasil pengujian aplikasi pada panel *Sound Filtering* yang ketika *push button* “ANALYZE” ditekan akan bekerja sesuai dengan fungsinya, pengujian menunjukkan *push button* sudah berfungsi dengan baik.

Tabel 18. Hasil pengujian *Options*

5) Panel *Options*

Tombol	Tampilan GUI	Keterangan
<i>RESET</i>	 <p>The screenshot shows the main interface of the 'HEART SIGNAL ANALYSER'. It features two empty signal plots, one for 'Input Sound' and one for 'Signal Filtering'. On the right side, there are control panels for 'Input Signal' (with an 'OPEN FILE' button) and 'Signal Analyser' (with an 'ANALYZE' button). At the bottom right, there are two buttons: 'RESET' and 'HOME'. The 'RESET' button is highlighted with a blue glow, indicating it is the active element for this test case.</p>	Baik
<i>HOME</i>	 <p>This screenshot shows the same GUI as above, but with a modal dialog box centered on the screen. The dialog box has a question mark icon and the text 'Are You Sure Back To HOME?'. It contains two buttons: 'YES' and 'NO'. The 'HOME' button in the bottom right of the main GUI is highlighted with a blue glow, indicating it is the active element for this test case.</p>	Baik

Tabel 18 menunjukkan hasil pengujian aplikasi pada panel *Sound Filtering* yang ketika *push button* “*RESET/HOME*” ditekan akan bekerja sesuai dengan fungsinya. Pengujian menunjukkan *push button* sudah berfungsi dengan baik.

#### 4. Pengujian JST

Pengujian ini dilakukan untuk melihat bagaimana variabel jumlah *hidden layer*, dapat mempengaruhi hasil *net* yang dihasilkan oleh JST nya.

Tabel 19. Pengujian JST

No.	Jumlah hidden layer	Hasil <i>training</i> JST	Hasil <i>net</i> JST pada data latih	Hasil <i>net</i> JST pada data uji
1.	10	80%	Ada 2 data yang <i>error</i>	Ada 10 data yang <i>error</i>
2.	20	56%	Ada 4 data yang <i>error</i>	Ada 16 data yang <i>error</i>
3.	25	100%	Tidak ada hasil data yang <i>error</i>	Tidak ada hasil data yang <i>error</i>

Tabel 19 menunjukkan hasil pengujian JST, hasil *training* JST yang didapatkan tingkat akurasi yang baik yaitu 100% dengan melihat jumlah *hidden layer* 25 dan hasil *net* yang tidak ada *error* pada data latih maupun data uji

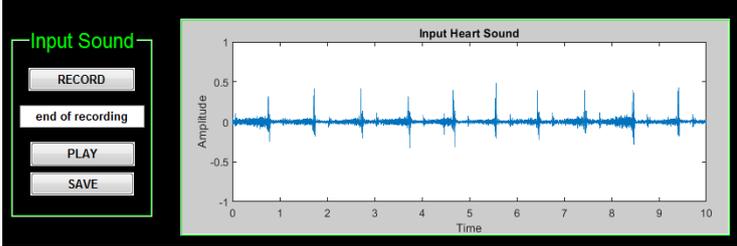
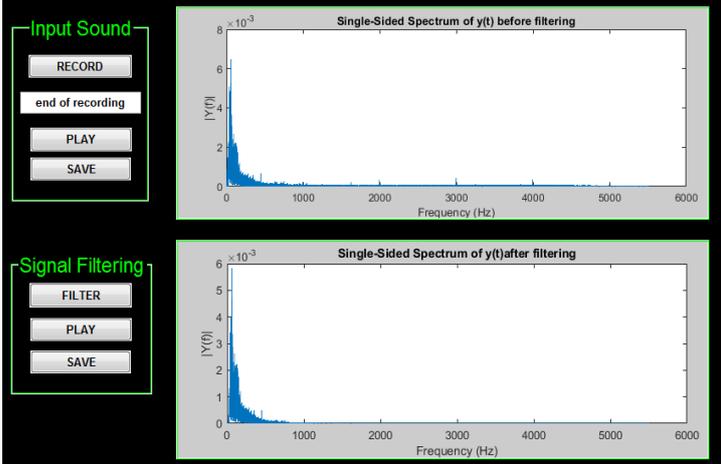
#### 5. Pengujian Unjuk Kerja

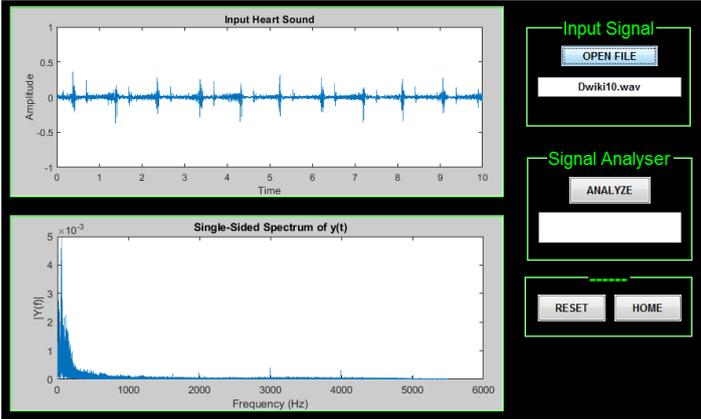
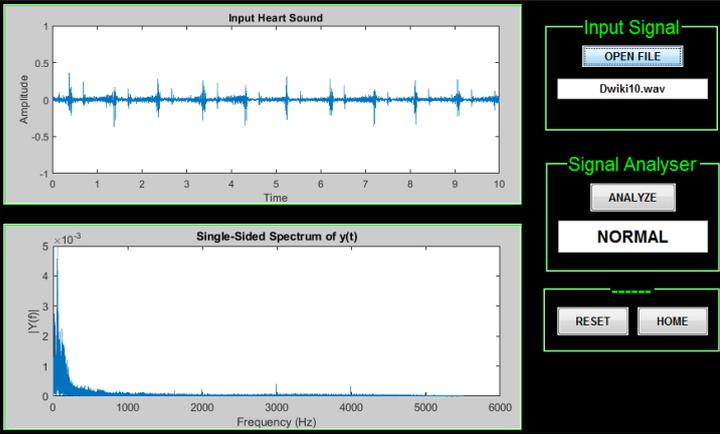
Pengujian unjuk kerja dilakukan untuk mengetahui kinerja seluruh bagian dari perancangan alat ini sebagai sebuah sistem secara utuh, hal yang perlu

diamati antara lain proses merekam suara jantung, filter suara jantung, ekstraksi ciri suara jantung dan JST. Pengujian ini bertujuan untuk mengetahui apakah alat sudah bekerja dengan baik atau masih terdapat kekurangan pada saat alat melakukan proses kerja.

a. Pengujian unjuk kerja alat

Tabel 20. Pengujian unjuk kerja alat

Fungsi	Tampilan GUI	Keterangan
Merekam Suara Jantung		Baik
Filter Suara Jantung		Baik

<p>Ekstraksi Ciri Suara Jantung</p>		<p>Baik</p>
<p>JST</p>		<p>Baik</p>

Tabel 20 menunjukkan hasil pengujian unjuk kerja alat secara keseluruhan dari alat mulai dari proses merekam suara dari *hardware*, melakukan proses *filtering* pada *software*, ekstraksi ciri suara jantung yang akan dianalisis, dan pengambilan keputusan berdasarkan JST yang telah dibuat. Pengujian ini menunjukkan bahwa seluruh system telah bekerja dengan baik.

## B. Pembahasan

Berdasarkan hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa, sistem cerdas yang telah dirancang, dapat bekerja dengan baik, sebagaimana fungsinya, walaupun masih terdapat *noise* suara pada perekaman dilakukan, berikut pembahasan dari pengujian yang telah dilakukan :

### 1. *Power supply*

*Power supply* pada perancangan sistem cerdas ini menggunakan modul regulator tegangan yaitu LM2596, regulator ini mempunyai keluaran arus maksimal sebesar 2 Ampere dengan suhu operasi maksimal mencapai 85°. Pengujian tegangan *power supply* ini menggunakan 2 buah indikator yaitu tanpa beban dan dengan beban. Hasil pengukuran tanpa beban dan dengan beban pada modul regulator tegangan LM2596 mempunyai hasil yang sama, bias

Dapat dilihat pada table 12 yang merupakan hasil pengujian tegangan power suplai pada rangkaian perekam suara, pada konsisi pertama power suplai diberikan input sebesar 5 V dan dilakukan stepdown menjadi 4.5 V sehingga keluaran tanpa beban menjadi 4.5 V, begitu juga ketika diberikan beban, keluaran tegangan tetap pada 4.5 V.

### 2. Rangkaian Filter (*low pass*) Perekam Suara

Rangkaian filter (*low pass*) ialah rangkaian untuk memfilter frekuensi suara yang dihasilkan oleh modul perekam suara pada saat dilakuan perekaman, *low pass* filter initerdiri dari resistor 4700 Ohm dan kapasitor 22 nf, rangkaian *low pass* filter ini mengambil nilai *cut off* sebesar 1500 Hz.

Nilai *cut off* sebesar 1500 Hz ini diambil menimbang dari batas akhir frekuensi suara jantung *murmur* adalah 1000 Hz, maka diambil jarak potong frekuensi sebesar 500 Hz dari batas akhir frekuensi suara jantung *murmur*, hal ini dilakukan untuk mengeliminasi *noise* yang dihasilkan lingkungan sekitar tanpa mengurangi detail-detail yang ada pada suara jantung yang terekam.

Berdasarkan pengujian pada table 13 menunjukkan rangkaian filter (*low pass*) bekerja dengan baik, ini ditandai dengan gambaran grafis gelombang suara sebelum difilter dan setelah difilter, dari gambaran tersebut menunjukkan bahwa rangkain filter pada perekam suara telah bekerja dengan baik.

### 3. Aplikasi

Pembahasan pada aplikasi ini menunjukkan kinerja dari *software* yang telah dibuat, pengujian ini fungsi dari masing-masing *push button* bekerja sesuai dengan fungsinya. Pengujian pada *software* ini dibagi 1 halaman home dan 5 panel, yang pertama yaitu panel *Input Sound*, yang kedua yaitu panel *Sound Filtering*, yang ketiga yaitu panel *Input Signal*, yang keempat yaitu panel *Signal Analyser*, dan yang kelima yaitu panel *Options*.

Berdasarkan tabel 14, hasil pengujian halaman Home melihat ketika *push button* “Mulai” ditekan maka yang sebelumnya berada pada halaman Home akan masuk ke halaman utama, pengujian ini menunjukkan fungsi dari *push button* “Mulai” telah bekerja sesuai dengan fungsinya.

Berdasarkan tabel 15, pengujian ini dilakukan untuk melihat kinerja *software* ketika sudah memasuki halaman utama, yaitu pada panel *Input Sound* yang terdiri dari *push button* “Record, PLAY, Save” apakah sudah bekerja sesuai dengan

fungainya. Hasil pada pengujian ini menunjukkan pada *push button* “*Record*” sudah dapat merekam suara dengan baik sesuai dengan waktu sampling yang ditentukan, begitu juga pada *push button* “*Save dan PLAY*” sudah dapat menyimpan hasil file rekaman dan memainkan suara hasil rekaman, hal ini menunjukkan bahwa semua *push button* pada panel *Input Sound* telah bekerja dengan baik.

Berdasarkan tabel 16, pengujian ini dilakukan untuk melihat kinerja *software* pada panel *Sound Filtering* yang terdiri dari *push button* “*Filter, PLAY, Save*” apakah sudah bekerja sesuai dengan fungsinya. Hasil pada pengujian ini menunjukkan pada *push button* “*Filter*” sudah dapat memfilter suara dari hasil rekaman dengan sesuai dengan fungsi filter yang ditentukan dan sekaligus melakukan ekstraksi ciri menggunakan FFT dengan baik, begitu juga pada *push button* “*PLAY dan Save*” sudah dapat memainkan dan menyimpan hasil suara rekaman yang telah difilter dan diekstraksi ciri, hal ini menunjukkan bahwa semua *push button* pada panel *Sound Filtering* telah bekerja dengan baik.

Berdasarkan tabel 17, pengujian ini dilakukan untuk melihat kinerja *software* pada panel *Input Signal* apakah sudah bekerja sesuai dengan fungsinya. Hasil pada pengujian ini menunjukkan ketika *push button* “*Open File*” ditekan sudah dapat membuka file yang tersimpan dalam PC dan menampilkannya ke dalam *software*, hal ini menunjukkan fungsi *push button* pada panel *Input Signal* telah bekerja dengan baik.

Berdasarkan tabel 18, pengujian ini dilakukan untuk melihat kinerja *software* pada panel *Signal Analyser* apakah sudah bekerja sesuai dengan fungsinya. Hasil pada pengujian ini menunjukkan ketika *push button* “*Analyze*” ditekan sudah dapat

mengkategorikan suara jantung *murmur* atau normal dan ditampilkan kedalam *software*, pengkategorian suara ini dapat dilakukan melalui rekaman langsung menggunakan *software* atau mengambil data yang sudah direkam sebelumnya yang tersimpan dialam PC, hal ini menunjukkan fungsi *push button* pada panel *Signal Analyser* telah bekerja dengan baik.

Berdasarkan tabel 19, pengujian ini dilakukan untuk melihat kinerja *software* pada panel *Options* yang terdiri dari *push button* “*Reset* dan *Home*” apakah sudah bekerja dengan sesuai dengan fungsinya. Hasil pengujian ketika *push button* “*Reset*” ditekan sudah dapat membersihkan data-data yang diinputkan ke dalam *software* dan mengembalikan tampilan saat halaman utama dibuka. Hasil pengujian ketika *push button* “*Home*” ditekan sudah dapat mengembalikan *software* dari halaman utama ke halaman *Home*, sama seperti ketika *software* pertama kali dibuka, hal ini menunjukkan fungsi *push button* pada panel *Options* telah bekerja dengan baik.

#### 4. Jaringan Syaraf Tiruan (JST)

Pengujian ini dilakukan untuk melihat seberapa besar akurasi yang didapatkan, melihat dari jumlah *hidden layer* dan data pelatihan yang di inputkan. Hasil akurasi yang didapatkan bergantung pada kesesuaian jumlah *hidden layer* dan jumlah data yang akan dilatihkan.

Berdasarkan tabel 20 dapat dilihat bahwa, jumlah *hidden layer* tidak berbanding lurus dengan hasil akurasi yang akan didapatkan, sedangkan jumlah data yang dilatihkan akan sangat berpengaruh terhadap akurasi yang akan dihasilkan oleh JST, pada pengujian ini jumlah *hidden layer* 25 menghasilkan

akurasi paling baik yaitu 100% dengan tidak adanya data yg error dari data latih maupun data uji, hasil ini lebih baik dari jumlah hidden layer 15 atau 20.

## 5. Unjuk Kerja Alat

Pengujian ini dilakukan untuk melihat unjuk kerja sistem secara keseluruhan, pengujian ini berdasarkan fungsi dari masing-masing proses yang terdapat pada alat ini, yaitu terdiri dari fungsi perekaman suara jantung, fungsi filter suara jantung, fungsi ekstraksi ciri suara jantung, dan fungsi jaringan saraf tiruan sebagai pengambil keputusan.

Berdasarkan tabel 21 dapat dilihat bahwa fungsi perekaman suara jantung sudah dapat menampilkan hasil gelombang suara pada axes yang ada pada *software*. Fungsi filter juga dapat dilihat bahwa suara jantung yang dihasilkan oleh proses perekaman sudah dapat difilter, perbedaan ini dapat dilihat dari tampilan pada axes yang ada pada *software*. Pengujian fungsi ekstraksi ciri dapat dilihat melalui axes yang ada pada *software*, dimana tampilan menunjukkan ketika suara diambil dari file yang tersimpan pada PC langsung menunjukkan FFT atau ekstasi ciri dari suara jantung, hal ini menunjukkan bahwa fungsi sudah bekerja dengan baik. Fungsi JST merupakan fungsi pengambilan keputusan setelah dilakukan ekstraksi ciri, pada tabel 21 dapat dilihat bahwa fungsi JST sudah dapat mengkategorikan suara jantung, apakah suara itu masuk kategori normal atau masuk kategori *murmur*. Dari pengujian ini menunjukkan bahwa semua fungsi unjuk kerja alat telah bekerja dengan baik.

## BAB V

### KESIMPULAN DAN SARAN

#### A. Kesimpulan

Berdasarkan hasil perancangan, realisasi dan pengujian yang telah dilakukan terhadap Sistem Cerdas Deteksi Suara Untuk Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Syaraf Tiruan, maka dapat disimpulkan:

1. Perancangan *Hardware* Perekam suara

a. Perancangan Mekanik

Perancangan mekanik berupa desain untuk menaruh beberapa rangkaian komponen dalam alat, dengan menggunakan box akrilik yang sudah disesuaikan dengan kebutuhan, sehingga didalamnya dapat di tempatkan semua rangkaian yang ada di dalam alat perekam suara, seperti modul regulator tegangan *power supply*, dan rangkaian modul perekam suara.

b. Perancangan Elektronik

Perancangan elektronik merupakan perancangan rangkaian yang ada pada alat perekam suara ini, yaitu rangkaian perekam suara yang didalamnya terdapat rangkaian penguat dan rangkaian filter suara. Perancangan elektronik menggunakan *software* Proteus.

c. Perancangan Perangkat lunak

Perancangan perangkat lunak menggunakan Matlab R2018a dengan bahasa pemrograman yang digunakan ialah C/C++.

2. Merealisasikan rancangan *software* Sistem Cerdas Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Syaraf Tiruan.
  - a. Merealisasikan perancangan mekanik dengan cara melubangi box akrilik sesuai dengan desain yang telah dibuat, guna untuk memasang semua rangkaian.
  - b. Merealisasikan perancangan elektronik yang telah dirancang dan di desain menggunakan *software* Proteus ISIS dan ARES, kemudian di cetak menggunakan kertas *glossy*, lalu menggunakan metode *transfer toner* untuk membuat jalur PCB dapat menempel, dan terakhir dilarutkan dengan FeCL, yang nantinya akan melarutkan jalur yang tidak tertempel dengan tinta dari kertas PCB, setelah itu di bor sesuai dengan ukuran kaki komponen-komponen nya, untuk kemudian disolder komponennya, dan dilakukan pengujian sebagaimana telah disebutkan diatas.
  - c. Merealisasikan perancangan perangkat lunak menggunakan Matlab R2018a, untuk memberikan program kepada *software* pengklasifikasian penyakit jantung yang dapat membuat sistem berjalan dengan baik
3. Unjuk kerja Sistem Cerdas Deteksi Suara Untuk Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Saraf Tiruan.

Sistem sudah dapat berjalan dengan normal dan baik melihat dari akurasi JST mencapai 100%, dengan menggunakan *input* dari *user* dan dan alat perekam suara. Alat perekam sudah bekerja dengan baik dan dapat menunjukan hasil yang sesuai dengan apa yang diinputkan oleh *user* maupun alat perekam suara, secara keseluruhan alat sudah bekerja dengan baik.

## **B. Keterbatasan Alat**

Ada beberapa kendala saat melakukan pembuatan proyek akhir ini, sehingga ada hal-hal yang perlu diperlu diperhatikan, Sistem Cerdas Deteksi Suara Untuk Pengklasifikasian Penyakit Jantung Menggunakan Jaringan Syaraf Tiruan, memiliki beberapa keterbatasan dalam pembuatannya, antara lain yaitu:

1. Proses pengkategorian suara jantung saat ini belum bisa *real time*.
2. Stetoskop akuistik yang digunakan masih menggunakan harga yang murah sehingga kesensitifan dari membrane stetoskop rendah
3. Tidak adanya indikator secara *hardware* saat perekaman suara dimulai.
4. Alat perekam masih dapat merekam suara organ lain selain jantung.
5. Alat belum bisa berdiri sendiri, sehingga bergantung pada PC untuk sumber tegangan dan proses pengolahan data suara.
6. Ukuran *software* pengolahan suara jantung yang terhitung masih terlalu besar.
7. Kurangnya data latih dan data uji untuk kategori suara jantung *murmur*.

## **C. Saran**

Karena keterbatasan waktu, ilmu, dan dana pembuatan proyek akhir ini terdapat banyak kekurangan, sehingga diperlukan pengembangan lebih lanjut. Saran membangun dibutuhkan untuk menyempurnakan alat ini, antara lain sebagai berikut:

1. Membuat alat menjadi satu kesatuan menggunakan mini PC yang terpasang didalam alat serta display olahan data yang langsung dari satu alat tersebut.

2. Membuat alat menjadi *real time* untuk proses pengolahan data suara jantung.
3. Menggunakan stetoskop yang lebih baik kualitasnya agar hasil perekaman menjadi lebih baik lagi.
4. Membuat indikator pada alat mulai dari indikator rekam hingga proses pengolahan selesai.
5. Menggabungkan alat perekam suara jantung dengan EKG untuk proses pengolahan data yang lebih tajam.
6. Pengkategorian suara dapat ditambah berdasarkan penyakit jantung yang diderita.
7. Alat ini sebaiknya diuji coba lagi dengan berbagai kondisi pengujian seperti:
  - a. Diuji terhadap rentang usia dari mulai anak kecil hingga orang dewasa.
  - b. Seberapa besar perbandingan unjuk kerja sistem cerdas ini jika dibandingkan dengan alat-alat serupa yang ada dipasaran.
8. Sebaiknya dibuat rangkaian tambahan pada *hardware* untuk filter *band pass*, dan notch filter.
9. Perlu ditambahkan *port headset* untuk langsung mendengar hasil perekaman suara jantung.
10. Perlunya dipikirkan lebih lanjut, sistem yang telah dirancang agar dapat diaplikasikan untuk kebutuhan dokter dalam mendiagnosa penyakit jantung pasien.

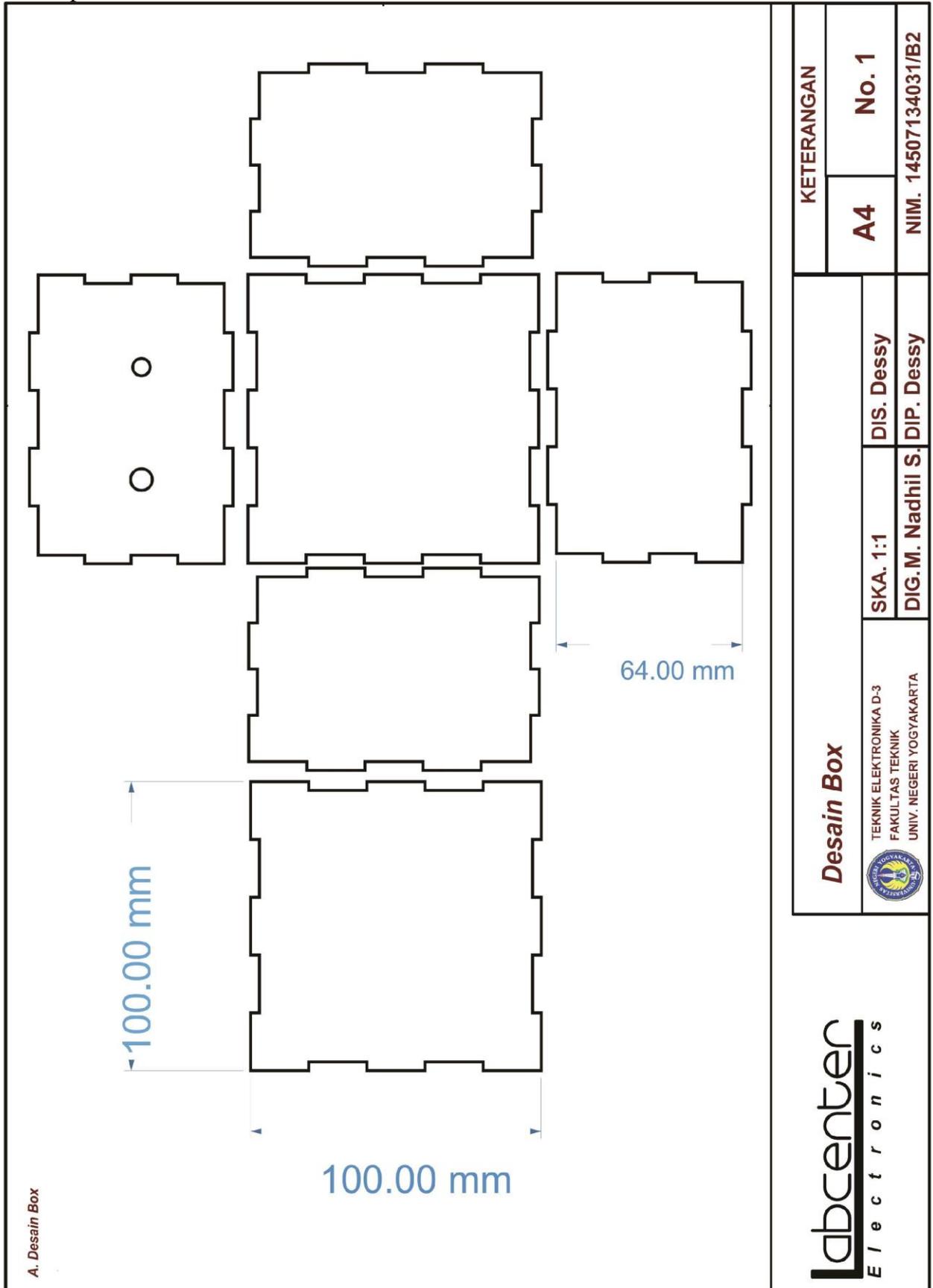
## DAFTAR PUSTAKA

- Adafruit. 2018. *MAX9814*.  
<https://www.adafruit.com/product/1713>. Diakses tanggal 26 Maret 2018.
- Addicore. 2018. *LM2596 Step-Down Adjustable DC-DC Switching Buck Converter*.<https://www.addicore.com/lm2596-module-p/ad281.htm>. Diakses tanggal 26 Maret 2018.
- Alodokter. 2018. *Membedakan Bunyi Jantung Normal dan Abnormal*.  
<https://www.alodokter.com/membedakan-bunyi-jantung-normal-dan-abnormal>. Diakses tanggal 15 Oktober 2018.
- Britannica. 2018. *Encyclopedia Britannica*.  
[www.Britannica.com](http://www.Britannica.com). Diakses tanggal 21 Desember 2018.
- Danmogot. 2018. *Jenis dan Fungsi Stetoskop*.  
<https://danmogot.com/blog/artikel-11552-jenis-dan-fungsi-stetoskop.html>.  
Diakses tanggal 20 Desember 2018.
- Dr. Raymond. 2007. *Stethographics*.  
[www.stethographics.com/main/physiology\\_hs\\_introduction.html](http://www.stethographics.com/main/physiology_hs_introduction.html).  
Diakses tanggal 25 Juli 2018.
- Ginanto Novika. 2012. *Backpropagation*.  
<https://novikaginanto.wordpress.com/2012/11/14/backpropagation/>. Diakses tanggal 19 November 2018.
- Kemendikbud. 2007. *Macam-macam Rangkaian Filter*.  
[https://medukasi.kemdikbud.go.id/medukasi/produkfiles/kontenonline/online2007/rangkaianfilter/macam\\_filter.htm](https://medukasi.kemdikbud.go.id/medukasi/produkfiles/kontenonline/online2007/rangkaianfilter/macam_filter.htm). Diakses tanggal 20 Desember 2018.
- Kidodi. 2012. *Pengertian Sistem Cerdas*.  
<https://kidodi.wordpress.com/2012/02/19/pengertian-sistem-cerdas/>. Diakses tanggal 8 November 2018.
- Medicalogy. 2017. *Jenis-Jenis Stetoskop*.  
<https://www.medicalogy.com/blog/jenis-jenis-stetoskop/>. Diakses tanggal 20 Desember 2018.

- Purnama Agus. 2013. *Jaringan Saraf Tiruan*.  
<http://elektronika-dasar.web.id/jaringan-saraf-tiruan-neural-network/#>.  
Diakses tanggal 7 Oktober 2018.
- Rich, Elaine dan Knight, Kevin. 1991. *Artificial Intelligence*. New York: MacGraw-Hill.
- Riskerdas. 2013. *Riset Kesehatan Dasar*. Jakarta: Kementerian Kesehatan RI
- Simon H.A, Langley P, Bradshaw G, dan Zytkow J. 1987. *Scientific Discovery : computational explorations of the creative processes*. Massachusetts : MIT press.
- Tespenku .2018. *Low Pass Filter LPF Filter Pasif RC*.  
<http://www.tespenku.com/2018/01/low-pass-filter-lpf-filter-pasif-rc.html>.  
Diakses tanggal 20 Desember 2018.
- Winston dan Prendergast. 1984. *The "AI Business" began to be lucrative*. Massachusetts : MIT press.
- V Rajaraman. 2014. "*John MC Carthy – Father of Artificial Intelligence*".  
[www.ias.ac.in/article/fulltext/resp /019/03/0198-0207](http://www.ias.ac.in/article/fulltext/resp /019/03/0198-0207). Diakses tanggal 15 Desember 2018

# LAMPIRAN

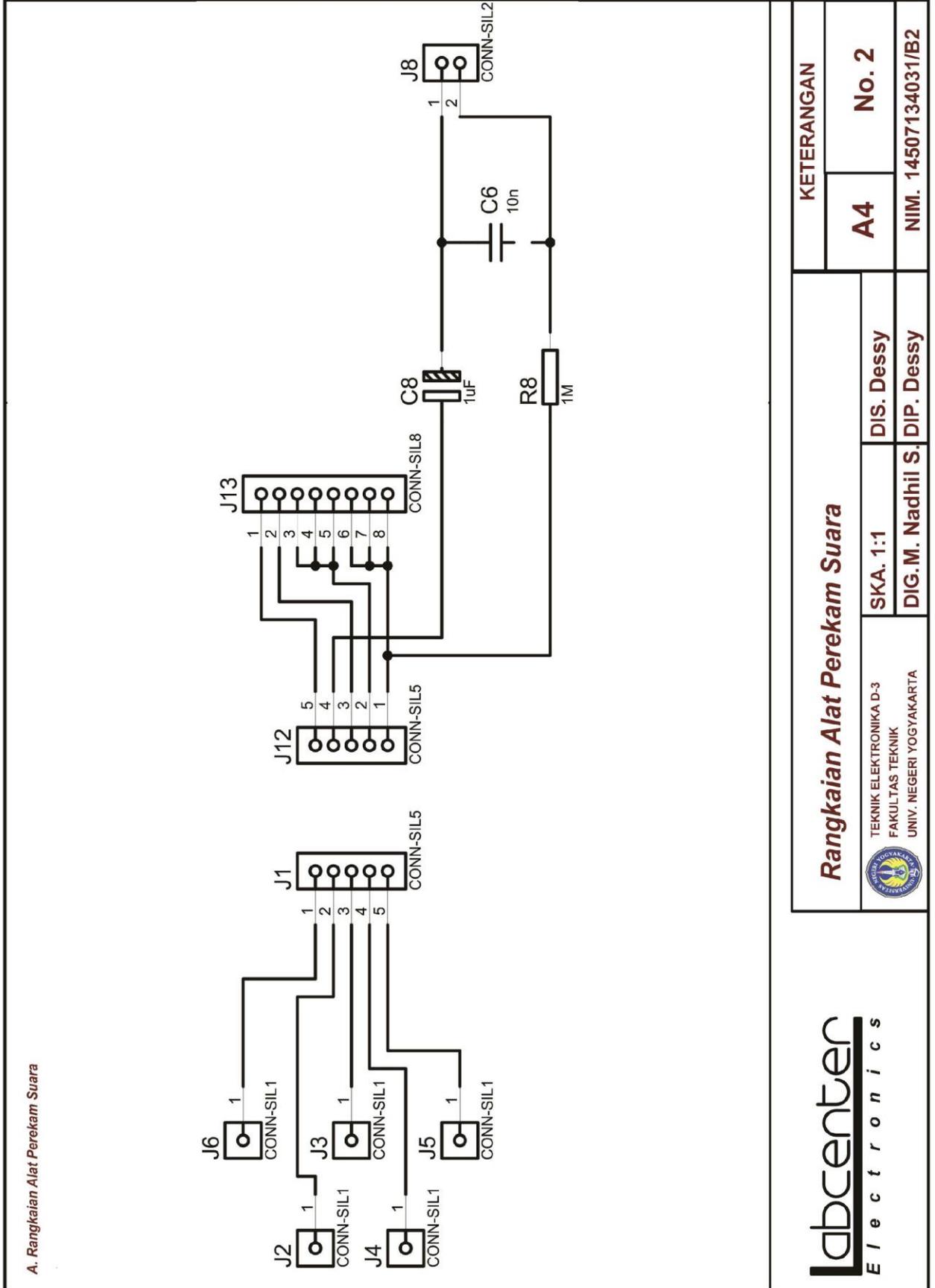
Lampiran 1. Desain *Box Panel*



<b>Desain Box</b>	<b>KETERANGAN</b>	
	<b>A4</b>	<b>No. 1</b>
 TEKNIK ELEKTRONIKA D-3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA	SKA. 1:1	DIS. Dessy
	DIG. M. Nadhil S	DIP. Dessy
NIM. 14507134031/B2		

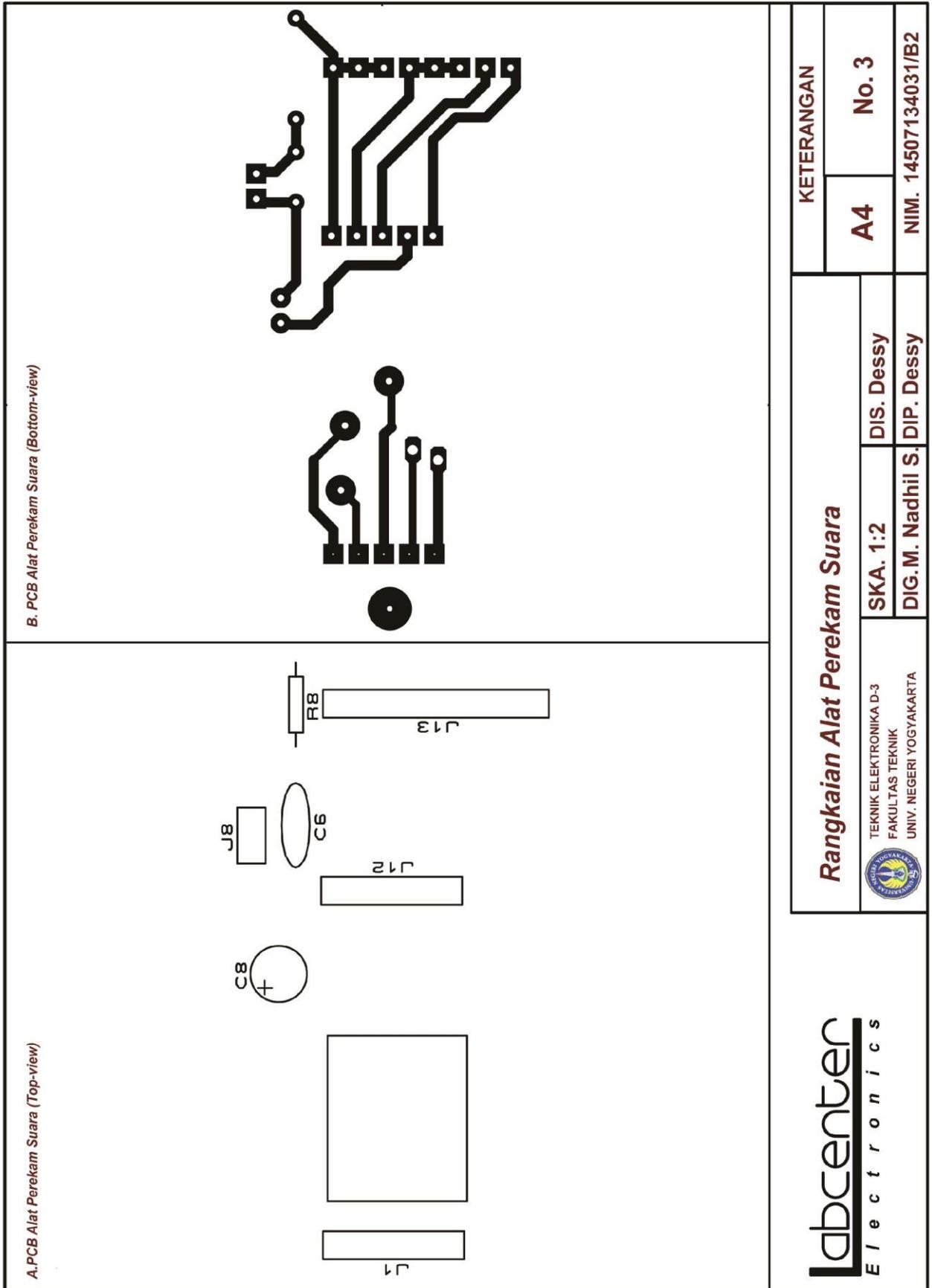
**labcenter**  
Electronics

Lampiran 2. Rangkaian Alat Perekam Suara



<b>Rangkaian Alat Perekam Suara</b>		<b>KETERANGAN</b>	
		<b>A4</b>	<b>No. 2</b>
TEKNIK ELEKTRONIKA D-3 FAKULTAS TEKNIK UNIV. NEGERI YOGYAKARTA		SKA. 1:1	DIS. Dessy
		DIG. M. Nadhil S.	DIP. Dessy
		NIM. 14507134031/B2	

Lampiran 3. Rangkaian Alat Perekam Suara



#### Lampiran 4. Daftar Komponen, Alat dan Harga

##### A. Rangkaian Alat Perekam Suara

No.	Nama Komponen	Nilai	Banyak	Harga	Total
1.	Kapasitor	15nF	1	Rp. 100	Rp. 100,-
2.	Resistor	4K7	1	Rp. 100,-	Rp. 100,-
3.	Stereo <i>Female</i> connector		1	Rp. 2.000,-	Rp. 2.000,-
4.	Stereo <i>Jack</i>		1	Rp. 2.500,-	Rp. 2.500,-
5.	Pin Header (Male)	1x40	1	Rp. 1.500,-	Rp. 1.500,-
6.	Pin Header (Female)	1x40	1	Rp. 2.000,-	Rp. 2.000,-
				TOTAL	Rp. 8.200,-

##### B. Alat-Alat

No.	Nama Barang	Banyak	Harga	Total
1.	Stetoskop Akuistik	1	Rp. 65.000,-	Rp. 65.000,-
2.	Kabel audio	1	Rp. 5.000,-	Rp. 5.000,-
3.	Box alat (Akrilik)	1	Rp. 13.000,-	Rp. 13.000,-
4.	Kabel USB	1	Rp. 10.000,-	Rp. 10.000,-
			TOTAL	Rp. 93.000,-

##### C. Modul-modul

No.	Nama Modul	Banyak	Harga	Total
1.	MAX9814	1	Rp. 35.000,-	Rp. 35.000,-
2.	Step-Down LM2596	1	Rp. 17.000,-	Rp. 17.000,-
			TOTAL	Rp. 52.000,-

## Lampiran 5. Source Code

```
function varargout = TANadhil(varargin)
% TANADHIL MATLAB code for TANadhil.fig
%     TANADHIL, by itself, creates a new TANADHIL or raises the
existing
%     singleton*.
%
%     H = TANADHIL returns the handle to a new TANADHIL or the
handle to
%     the existing singleton*.
%
%     TANADHIL('CALLBACK',hObject,eventData,handles,...) calls
the local
%     function named CALLBACK in TANADHIL.M with the given input
arguments.
%
%     TANADHIL('Property','Value',...) creates a new TANADHIL or
raises the
%     existing singleton*. Starting from the left, property
value pairs are
%     applied to the GUI before TANadhil_OpeningFcn gets called.
An
%     unrecognized property name or invalid value makes property
application
%     stop. All inputs are passed to TANadhil_OpeningFcn via
varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows
only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help TANadhil

% Last Modified by GUIDE v2.5 09-Oct-2018 00:20:23

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @TANadhil_OpeningFcn, ...
                  'gui_OutputFcn',  @TANadhil_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
% End initialization code - DO NOT EDIT

% --- Executes just before TAnadhil is made visible.
function TAnadhil_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to TAnadhil (see VARARGIN)

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

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes TAnadhil wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = TAnadhil_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

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

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uigetfile('*.wav', 'Select a wave
file', 'MultiSelect', 'on');
[y, Fs] = audioread(fullfile(pathname, filename));

handles.y = y;
handles.Fs = Fs;
guidata(hObject, handles);

%---Fungsi ngatur plot waktu

```

```

y = y(:,1);
dt = 1/Fs;
t = 0:dt:(length(y)*dt)-dt;
L = length(y);

axes(handles.axes1)
plot(t,y)
xlabel('Time')
ylabel('Amplitude')
title('Input Heart Sound')
set(handles.edit4, 'String', filename)
set(gca, 'YLim', [-1 1])

axes(handles.axes2)
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y = fft(y,NFFT)/L;
f = Fs/2*linspace(0,1,NFFT/2+1);

x = 2*abs(Y(1:NFFT/2+1));
handles.x = x;
guidata(hObject,handles);

% Plot single-sided amplitude spectrum.
plot(f,x)
title('Single-Sided Spectrum of y(t)')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
x = handles.x;

load net8
output = round(sim(net,x));

if output == [1;0]
    set(handles.edit2, 'String', 'NORMAL')
elseif output == [0;1]
    set(handles.edit2, 'String', 'MURMUR')
end

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

```

```

% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

Fs = 11025;
recObj = audiorecorder(Fs, 16, 1);
set(handles.edit6, 'String', 'start recording..')
recordblocking(recObj, 10);
set(handles.edit6, 'String', 'end of recording')
y = getaudiodata(recObj, 'single');

handles.y = y;
handles.Fs = Fs;
guidata(hObject, handles);

%---Fungsi ngatur plot waktu
y = y(:,1);
dt = 1/Fs;
t = 0:dt:(length(y)*dt)-dt;
L = length(y);

axes(handles.axes1)
plot(t,y)
xlabel('Time')
ylabel('Amplitude')
title('Input Heart Sound')
set(gca, 'YLim', [-2 2])
% filename = 'cobasaveaudio1.wav';
% audiowrite (filename,y,11025);
% set(handles.edit6, 'String', filename)
% clear recObj Fs

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton4 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)
y = handles.y;
Fs = handles.Fs;
y = y(:,1);
dt = 1/Fs;
t = 0:dt:(length(y)*dt)-dt;
L = length(y);

% pemilihan fungsi berpengaruh
dalam pemrosesan suara
% [z,p,k] = cheblap(10,3); % Chebyshev type 1 filter
orde 10 dengan 3db ripple
% [z,p,k] = cheb2ap(5,3);
[z,p,k] = besslap(10); % Bessell type filter

```

```

% [n,d] = zp2tf(z,p,k);           % mengubah zero-pole-gain ke
bentuk fungsi transfer
[A,B,C,D] = zp2ss(z,p,k);       % mengubah zero-pole-gain ke
bentuk state-space
u1 = 2*pi*20;                    % U1 high pass 20Hz
u2 = 2*pi*1000;                 % U2 low pass 1000Hz
Bw = u2-u1;
Wo = sqrt(u1*u2);
% [n1,d1] = lp2bp(n,d,Wo,Bw);
[At,Bt,Ct,Dt] = lp2bp(A,B,C,D,Wo,Bw); % Fungsi Band-pass filter
[n1,d1] = ss2tf(At,Bt,Ct,Dt);
sys = tf(n1,d1);
x = lsim(sys,y,t);

% Proses untuk membuat plot spektrum frekuensi (FFT)
NFFT = 2^nextpow2(L); % Next power of 2 from length of y
Y1 = fft(y,NFFT)/L;
f1 = Fs/2*linspace(0,1,NFFT/2+1);

% Plot single-sided amplitude spectrum.
axes(handles.axes1)
plot(f1,2*abs(Y1(1:NFFT/2+1)))
title('Single-Sided Spectrum of y(t) before filtering')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

Y2 = fft(x,NFFT)/L;
f2 = Fs/2*linspace(0,1,NFFT/2+1);
z = 2*abs(Y2(1:NFFT/2+1));

% Plot single-sided amplitude spectrum.
axes(handles.axes2)
plot(f2,2*abs(Y2(1:NFFT/2+1)))
title('Single-Sided Spectrum of y(t)after filtering')
xlabel('Frequency (Hz)')
ylabel('|Y(f)|')

handles.x = z;
handles.y1 = x;
handles.Fs = Fs;
guidata(hObject,handles);

% filename = 'nadhil2filter.wav';
% audiowrite (filename,x,Fs);

function edit2_Callback(hObject, eventdata, handles)
% hObject handle to edit2 (see GCBO)
% eventdata reserved - to be defined in a future version of
MATLAB
% handles structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of
edit2 as a double

% --- Executes during object creation, after setting all
properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of
edit4 as a double

% --- Executes during object creation, after setting all
properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

function edit6_Callback(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of
edit6 as a double

% --- Executes during object creation, after setting all
properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject      handle to edit6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton5 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
y = handles.y;
Fs = handles.Fs;

sound(y,Fs);

% --- Executes on button press in pushbutton6.
function pushbutton6_Callback(hObject, eventdata, handles)
% hObject      handle to pushbutton6 (see GCBO)
% eventdata    reserved - to be defined in a future version of
MATLAB
% handles      structure with handles and user data (see GUIDATA)
[filename, pathname] = uiputfile('*.wav');

if ~isequal(filename,0)
    Fs = handles.Fs;
    y = handles.y;
    audiowrite(fullfile(pathname,filename),y,Fs)
else
    return

```

```

end

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

cla(handles.axes1 , 'reset');
cla(handles.axes2 , 'reset');

set(handles.edit6, 'String', '')
set(handles.edit2, 'String', '')
set(handles.edit4, 'String', '')

handles.y = [];
handles.Fs = [];
handles.x = [];
handles.y1 = [];
guidata(hObject, handles);
clear

function edit7_Callback(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of edit7 as text
%        str2double(get(hObject, 'String')) returns contents of
edit7 as a double

% --- Executes during object creation, after setting all
properties.
function edit7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit7 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUiControlBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

```

```

% --- Executes on button press in pushbutton8.
function pushbutton8_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton8 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
y1 = handles.y1;
Fs = handles.Fs;

sound(y1,Fs);

% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton9 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
[filename, pathname] = uiputfile('*.wav');

if ~isequal(filename,0)
    Fs = handles.Fs;
    y1 = handles.y1;
    audiowrite(fullfile(pathname,filename),y1,Fs)
else
    return
end

% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton10 (see GCBO)
% eventdata  reserved - to be defined in a future version of
MATLAB
% handles    structure with handles and user data (see GUIDATA)
cl = questdlg('Are You Sure Back To HOME
?', 'HOME', 'YES', 'NO', 'NO');
switch cl
    case 'YES'
        close();
        clear all;
        return;
    case 'NO'
        quit cancel;
end

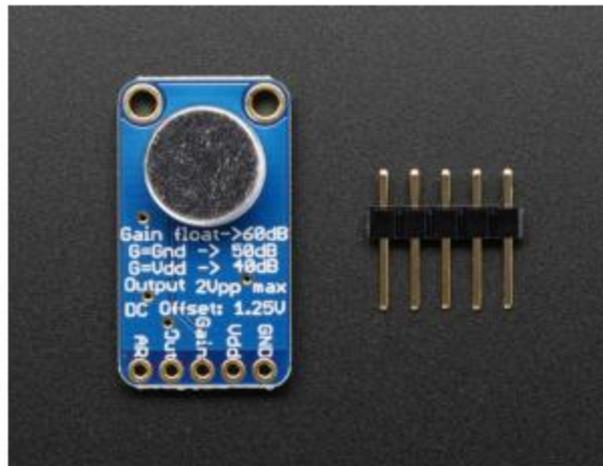
```

Lampiran 6. Datasheet MAX9814



**Adafruit AGC Electret Microphone Amplifier - MAX9814**

Created by lady ada.

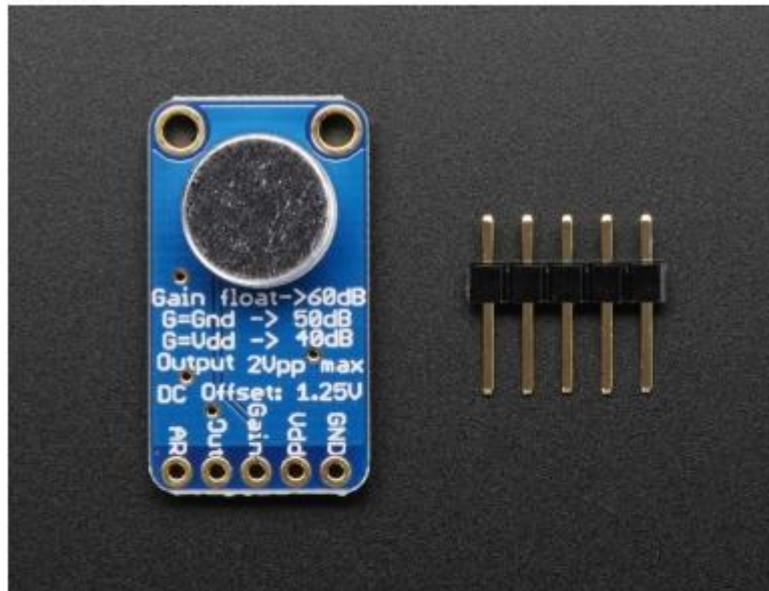


Last updated on 2018-08-22 03:39:53 PM UTC

## Guide Contents

Guide Contents	2
Overview	3
Assembly	5
Prepare the header strip:	5
Add the breakout board:	6
And Solder!	6
Wiring and Test	7
Wiring	7
Test!	8
Do more!	8
Downloads	10
Datasheets & Files	10
Schematics	10
PCB Dimensional Diagram	10
3D models	10

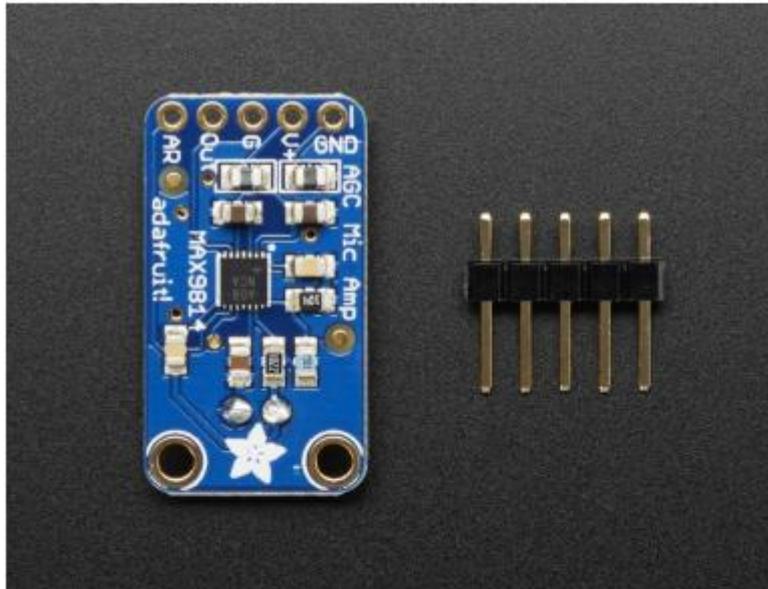
## Overview



This fancy microphone amplifier module is a step above the rest, with built in automatic gain control. The AGC in the amplifier means that nearby 'loud' sounds will be quieted so they don't overwhelm & 'clip' the amplifier, and even quiet, far-away sounds will be amplified. This amplifier is great for when you want to record or detect audio in a setting where levels change and you don't want to have to tweak the amplifier gain all the time.

The chip at the heart of this amp is the [MAX9814](https://adafru.it/d9r) (<https://adafru.it/d9r>), and has a few options you can configure with the breakout. The default 'max gain' is 60dB, but can be set to 40dB or 50dB by jumpering the **Gain** pin to VCC or ground. You can also change the Attack/Release ratio, from the default 1:4000 to 1:2000 or 1:500. The output from the amp is about 2Vpp max on a 1.25V DC bias, so it can be easily used with any Analog/Digital converter that is up to 3.3V input. If you want to pipe it into a Line Input, just use a 1uF blocking capacitor in series.

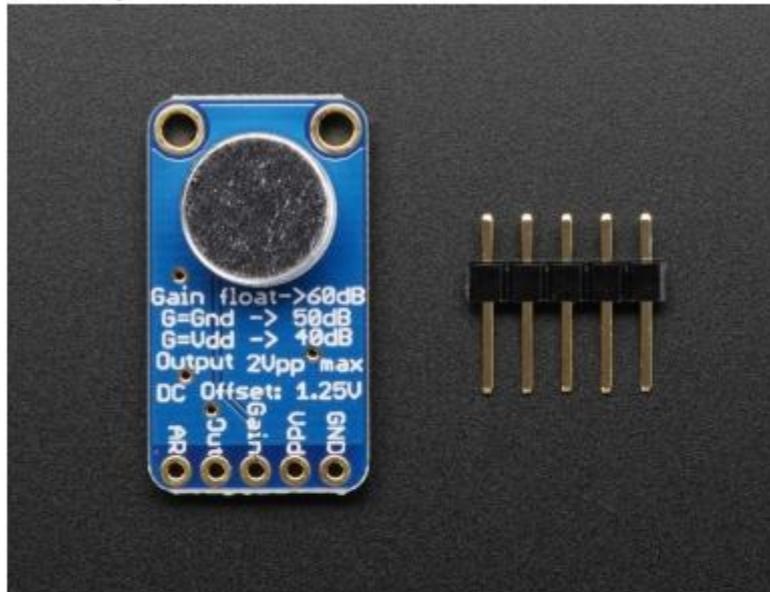
Each order comes with one assembled and tested board, with electret mic pre-soldered on, and a small piece of header.



Specifications:

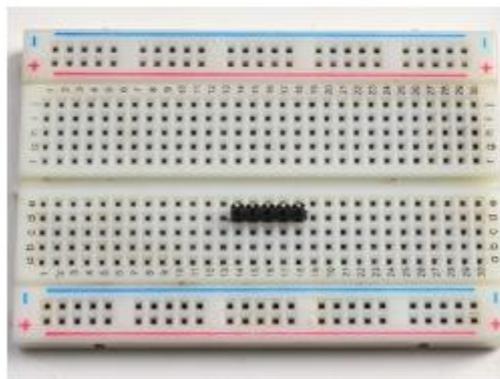
- Supply Voltage: 2.7V-5.5V @ 3mA current
- Output: 2V<sub>pp</sub> on 1.25V bias
- Frequency Response: 20Hz - 20 KHz
- Programmable Attack and Release Ratio
- Automatic gain, selectable max from 40dB, 50dB or 60dB
- Low Input-Referred Noise Density of  $30\text{nV}/\sqrt{\text{Hz}}$
- Low THD: 0.04% (typ)

## Assembly



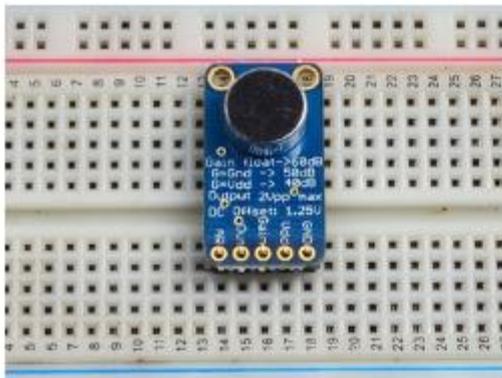
Assembly is really easy, you can use straight or 'right-angle' style headers to attach to the PCB. We'll be using the plain straight headers included

The board comes with all surface-mount components pre-soldered. The included header strip can be soldered on for convenient use on a breadboard or with 0.1" connectors. You can also skip this step and solder on wires.



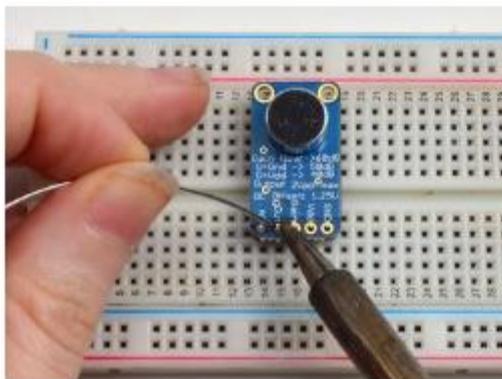
### Prepare the header strip:

Cut the strip to length if necessary. It will be easier to solder if you insert it into a breadboard - **long pins down**.



#### Add the breakout board:

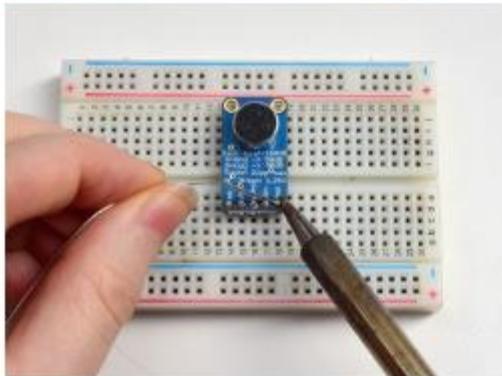
Place the breakout board over the pins so that the short pins poke through the breakout pads



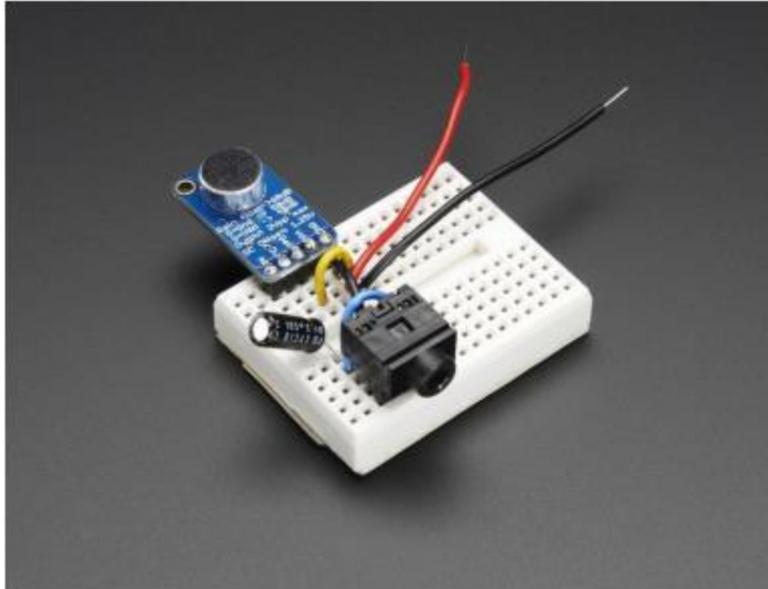
#### And Solder!

Be sure to solder all 5 pins for reliable electrical contact.

*(For tips on soldering, be sure to check out our [Guide to Excellent Soldering](https://adafruit.it/aTkj) (<https://adafruit.it/aTkj>).*



## Wiring and Test



The microphone amp is thankfully quite simple to get started. No microcontroller or programming required. We suggest wiring it up directly with a battery pack and headphones to start while you get a hang of how the AGC acts and responds.

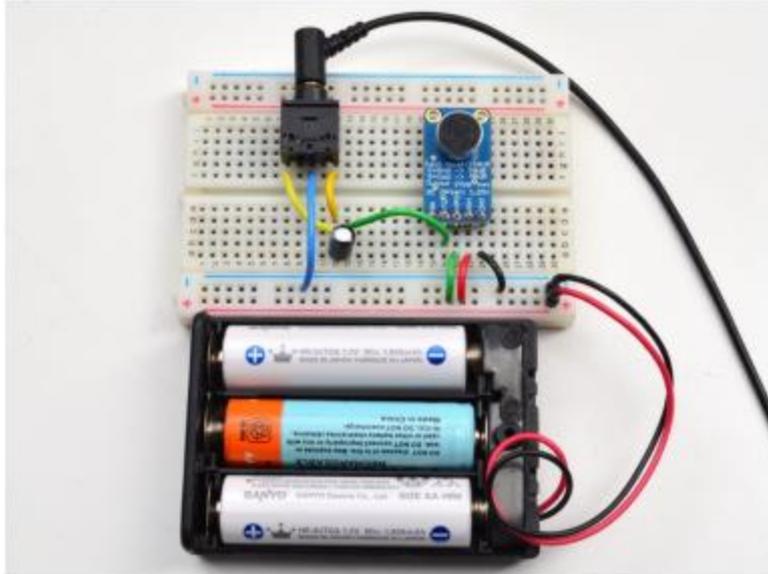
You'll need a [3 AA/AAA battery pack](http://adafruit.it/727) (or some external source of 3-5VDC) - we suggest batteries since they are very electrically-quiet compared to USB or wall power adapters.

(For rechargeable NiMH cells, you can use a 4xAA/AAA pack for a total of 4.8v)

You'll also need a [headphone jack, this breadboard-friendly one](http://adafruit.it/1699) and a 1 to 100uF electrolytic capacitor. This is just to protect your headphones from the DC bias voltage. Just about any capacitor will work.

Find the 'cheapest' headphones you have, or use earbuds, they won't load down the microphone amp as much! This is not a good time to use your huge Sennheiser cans. Don't connect to speakers or you will get really horrible feedback effects

## Wiring



Connect the amplifier, battery pack and headphone jack as shown above.

For the Microphone amp:

- **GND** connects to the battery pack ground - black wire
- **VDD** connects to the battery pack positive - red wire
- Connect the **Gain** pin to VDD for now - green wire
- Connect the **OUT** pin to a 1uF-100uF capacitor (really any value will do). If the capacitor is polarized, connect the OUT pin to the positive side - green wire

For the headphone jack

- Connect the **Left** and **Right** pins together (the microphone is mono out) to the negative side of the capacitor - yellow wires.
- Connect the center ground pin to the battery ground - blue wire

## Test!

Now try listening to sounds on the headphones, you should notice a strange 'effect' where you can hear people from further away than your hearing is used to! Try setting the **Gain** pin to the **GND** pin to get 10dB more gain. You can also remove the Gain wire, to set the gain to 60dB but you may find the gain is way too high, and it sounds 'too noisy', so we suggest sticking to 40 or 50dB for most purposes.

You can also try adjusting the A/R (attack/response) ratio by connecting the A/R pin to VDD or GND but we found that the default no-wire works pretty well for most purposes.

## Do more!

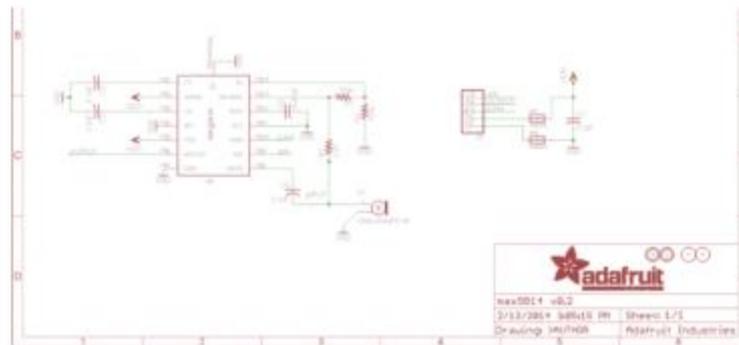
Now that you have the amplifier tested, you can use this microphone amp just like our non-AGC board.

## Downloads

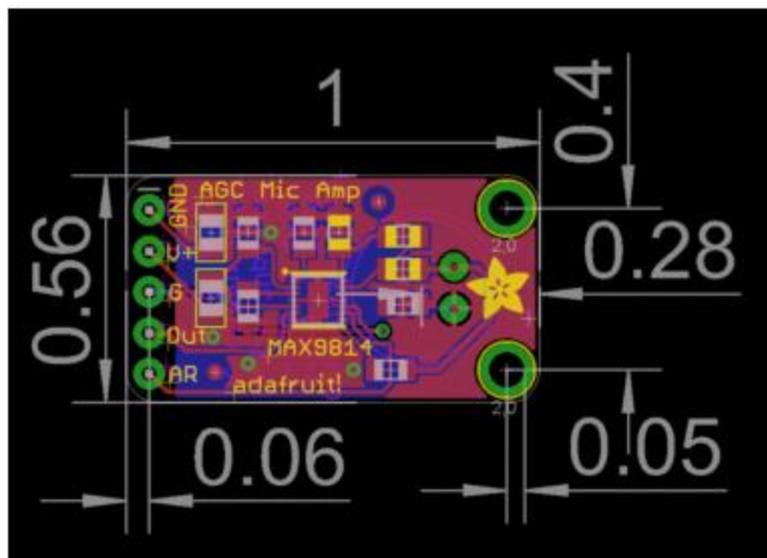
## Datasheets & Files

- [Datasheet for the MAX9814 \(https://adafru.it/d9o\)](https://adafru.it/d9o)
- [Datasheet for the electret mic capsule \(https://adafru.it/aW6\)](https://adafru.it/aW6)
- [Fritzing object in Adafruit Fritzing library \(https://adafru.it/aP3\)](https://adafru.it/aP3)
- [EagleCAD PCB files in GitHub \(https://adafru.it/rpb\)](https://adafru.it/rpb)

## Schematics



## PCB Dimensional Diagram



## 3D models

## LM2596 DC-DC Adjustable PSU Module



LM2596 DC to DC step down regulator, adjustable +1.23 to 35vdc output, 2A. Ideal for battery operated projects requiring a regulated powersupply.

### Specifications

Regulator Type:	Step Down ( Non Isolated input to Output )
Input Voltage:	+4 to 40vdc
Output Voltage:	+1.23 to 35vdc
Output Current:	2A rated, ( 3A maximum with heatsink )
Efficiency:	Up to 92% ( when output voltage is set high )
Switching Frequency:	150kHz
Dropout Voltage:	2vdc minimum
Protection:	Short circuit current limiting
Load Regulation:	+/- 0.5%
Voltage Regulation:	+/- 2.5%
Temperature:	-40 to +85 deg C ( output power less than 10Watts )
Board Size:	43.6mm L x 21mm W x 14mm H
Data Sheet:	<a href="#">National LM2596</a>

