

**SISTEM PENDETEKSI MUSIK PADA MODEL ROBOT PENARI  
DALAM STUDI KASUS KONTES ROBOT SENI INDONESIA**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan  
Memperoleh gelar Sarjana Teknik

**UNIVERSITAS BRAWIJAYA**



Disusun Oleh:

**ANDI SULISTIONO MISWANTO**  
**NIM 0510630014-63**

**KEMENTERIAN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2010**

**SISTEM PENDETEKSI MUSIK PADA MODEL ROBOT PENARI  
DALAM STUDI KASUS KONTES ROBOT SENI INDONESIA**

**SKRIPSI**

Diajukan untuk memenuhi sebagian persyaratan

Memperoleh gelar Sarjana Teknik



Disusun Oleh:

**ANDI SULISTIONO MISWANTO**

**NIM 0510630014**

Mengetahui dan menyetujui

Dosen pembimbing:

**Waru Djuriatno, ST., MT.**

**NIP. 19690725 199702 1 001**

**Adharul Muttaqin, ST., MT.**

**NIP. 19760121 200501 1 001**

**SISTEM PENDETEKSI MUSIK PADA MODEL ROBOT PENARI  
DALAM STUDI KASUS KONTES ROBOT SENI INDONESIA**

Disusun Oleh:

**ANDI SULISTIONO MISWANTO**

**NIM 0510630014**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 15 juli 2010

**DOSEN PENGUJI**

**Ir. M. Julius St., MS.**

**NIP. 19540720 198203 1 002**

**M. Rif'an, ST., MT.**

**NIP. 19710301 200012 1 001**

**Panca Mudjirahardjo, ST., MT.**

**NIP. 19700329 200012 1 001**

Mengetahui  
Ketua Jurusan Teknik Elektro

**Rudy Yuwono, ST., M.Sc.**  
**NIP. 19710615 199802 1 003**



## PENGANTAR

Alhamdulillah, segenap puji dan syukur penulis ucapkan kehadirat Allah SWT yang telah melimpahkan rahmat, hidayah, ridho, nikmat dan karunia-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan judul “**SISTEM PENDETEKSI MUSIK PADA MODEL ROBOT PENARI DALAM STUDI KASUS KONTES ROBOT SENI INDONESIA**”. Tidak lupa shalawat serta salam semoga senantiasa tercurahkan untuk Rasulullah Muhammad SAW beserta keluarga, sahabat, kerabat dan para pengikutnya sampai akhir jaman.

Dengan kesungguhan dan rasa rendah hati, penulis mengucapkan terima kasih kepada Bapak Waru Djuriatno, ST., MT dan Bapak Adharul Muttaqin, ST., MT sebagai dosen pembimbing atas saran, konsultasi, motivasi, kesabaran dan waktu yang telah dicurahkan dalam pengerjaan hingga penyelesaian skripsi ini. Tidak terkecuali, ucapan terima kasih penulis ucapkan kepada :

1. Kedua orang tua yang telah mendidik dan membesarkanku hingga saat ini, terima kasih atas doa dan kasih sayang yang telah diberikan, untuk adikku Tegar, terima kasih telah menginspirasi dan memotivasiku untuk selalu berjuang dan selalu pantang menyerah, serta untuk adikku Bobby terima kasih atas segala dukungan, bantuan, dan telah menjadi penyemangat bagiku
2. Bapak Rudy Yuwono, ST. M.Sc. selaku Ketua Jurusan Teknik Elektro Universitas Brawijaya
3. Bapak Muh. Aziz Muslim, ST.,MT., Ph.D. selaku Sekretaris Jurusan Teknik Elektro Universitas Brawijaya
4. Bapak Ir. M. Julius St., MS selaku Ketua Kelompok Dosen dan Keahlian Teknik Elektronika yang telah membantu dalam menyelesaikan proses penyelesaian skripsi ini mulai seminar proposal, hasil dan ujian yang dilakukan penulis
5. Bapak R. Arief Setyawan, ST., MT selaku dosen pembimbing akademik
6. Bapak, Ibu dosen serta segenap staf dan karyawan Jurusan Teknik Elektro baik secara langsung maupun tidak langsung yang telah banyak membantu dalam menyelesaikan skripsi ini
7. Sahabatku Domy terima kasih atas motivasi yang telah diberikan, dan Samsul terima kasih atas bantuannya

8. Rekan-rekan pengerjaan skripsi Rizal “Tero”, Abrar, Riezqy, Mas Arnes “Om”, Rangga, Nino, dan Emal yang selalu memberi semangat, bantuan dan saran dalam pengerjaan skripsi ini
9. Lana, Rosyam, Reza, Arif, dan Irfan sebagai tim robot divisi KRSI 2010 yang telah membantu dalam proses pengerjaan skripsi ini
10. Ridho, Tommy, dan Febri terima kasih telah bersedia mendengarkan keluh kesahku selama ini
11. Untuk sahabat- sahabatku di elektro 2005 Fajar “Gombes”, Iin, Taufiq, Tyas, Dwi, Prasetyo “Ebes”, Adi “Tuban”, Hendri “Sinx”, Angga, Fido, Eko, Adit, Aan, Syauqi, Agung, Agung “Kartolo”, Raisa, Siska “Emak”, Dina, Anggi, Shinta, Nufa, dan Ihda yang telah memberiku semangat dan dukungan dalam pengerjaan skripsi ini
12. Untuk sahabat-sahabatku Nera, Ayoen, Evi “Mama”, Purana “Tante”, Juwita Efna, Purei, Sari, Abied, Tri Nur, Lia, Dida, April, Ayik, Erni “Tenny”, Dicky “Papa”, Praba, Bayu “Pepe”, Bantar, Yhendris, Agung Dwi, Arif “Dana”, Masro’i, Harto, dan lainnya terima kasih atas doa dan dukungannya selama ini
13. Untuk mas Dhanie ”Tenyom” dan mas Amat terimakasih atas saran-saran yang telah diberikan selama ini
14. Rekan-rekan seperjuangan Tim Robot TEUB tahun 2008, 2009 dan 2010 terimakasih atas dukungannya
15. Teman-teman kos di kerto asri 94 A atas bantuannya dan dukungan moral dan materinya dalam pengerjaan skripsi ini
16. Semua pihak yang tidak dapat saya sebutkan satu persatu. Terima kasih, sehingga skripsi ini dapat terselesaikan dengan sebaik – baiknya

Penulis menyadari bahwa yang tersusun dalam skripsi ini masih banyak kekurangan karena keterbatasan materi dan pengetahuan yang dimiliki. Karena itu kritik dan saran sangat diharapkan demi kesempurnaan penyusunan skripsi ini. Akhir kata, penulis mengharapkan semoga yang terdapat dalam skripsi ini bermanfaat untuk kita semua, rekan-rekan mahasiswa khususnya dan bagi seluruh pembaca pada umumnya.

Malang, Juli 2010

Penulis



## ABSTRAK

**ANDI SULISTIONO MISWANTO, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya, Juli 2010, *Sistem Pendeteksi Musik pada Model Robot Penari dalam Studi Kasus Kontes Robot Seni Indonesia*, Dosen Pembimbing : Waru Djuriatno, ST., MT. dan Adharul Muttaqin, ST., MT.**

Pengembangan robot sebagai sarana hiburan semakin diminati oleh masyarakat. Kontes robot yang diselenggarakan oleh Direktorat Jenderal Pendidikan Tinggi merupakan salah satu upaya untuk menumbuh kembangkan kreatifitas dan minat dalam teknologi maju khususnya teknologi robotika. Kontes Robot Seni Indonesia (KRSI) merupakan salah satu ajang kontes robot tingkat nasional yang diselenggarakan tiap tahun. Kontes yang mengambil tema tarian tradisional Indonesia ini menghendaki robot harus mampu menari ketika musik diperdengarkan.

Karena robot hanya boleh menari ketika musik diperdengarkan, maka diperlukan suatu sistem yang dapat mendeteksi adanya suara musik. Sistem ini harus memiliki kemampuan untuk mengetahui apakah sinyal yang ditangkap oleh *microphone* benar-benar berasal dari suara musik atau tidak. Sinyal suara yang ditangkap kemudian difilter dan dikondisikan ke dalam level tegangan yang dapat diolah oleh mikrokontroler. Sinyal ini kemudian diubah ke dalam bentuk diskrit menggunakan ADC dalam mikrokontroler. Hasil konversi ADC kemudian diproses dalam persamaan filter digital, kemudian hasil perhitungan akan diolah untuk memutuskan apakah sinyal yang ditangkap berasal dari suara musik atau bukan.

Pengujian dilakukan dengan menempatkan robot pada arena lomba Kontes Robot Seni Indonesia. Hasil pengujian menunjukkan bahwa ketika sinyal yang ditangkap oleh *microphone* berasal dari suara musik, maka led indikator akan menyala. Sebaliknya, jika sinyal yang ditangkap berasal dari sinyal *noise* dalam hal ini sinyal *noise* berasal dari suara manusia, maka led indikator tidak menyala. Sistem hanya mampu mendeteksi sinyal yang berasal dari suara musik dengan *range* frekuensi yang berkisar antara 300 Hz sampai 2500 Hz.

Kata kunci: KRSI (Kontes Robot Seni Indonesia), *microphone*, ADC, sinyal, musik

## DAFTAR ISI

<b>PENGANTAR</b> .....	i
<b>ABSTRAK</b> .....	vi
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR GAMBAR</b> .....	x
<b>DAFTAR TABEL</b> .....	xii
<b>PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah.....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Sistematika Penulisan .....	3
<b>TINJAUAN PUSTAKA</b> .....	4
2.1 Kontes Robot Seni Indonesia.....	4
2.1.1 Spesifikasi Robot .....	4
2.1.2 Arena Pengujian.....	6
2.1.3 Microphone .....	6
2.2 Filter.....	8
2.2.1 Low Pass Filter .....	8
2.2.2 High Pass Filter.....	9
2.2.3 Band Pass Filter .....	10
2.2.4 Band Stop Filter .....	11
2.3 Sinyal .....	11
2.4 Noise .....	13
2.5 Sistem Waktu Diskrit.....	13
2.6 Filter Digital.....	15
2.7 Mikrokontroler AVR ATmega32.....	20
2.8 Analog To Digital Converter dalam Mikrokontroler ATmega32 .....	24
<b>METODE PENELITIAN</b> .....	28
3.1 Perencanaan dan Pembuatan Alat.....	28



3.2	Pengujian .....	28
3.2.1	Pengujian Microphone .....	29
3.2.2	Pengujian Filter dan Pengkondisi Sinyal .....	29
3.2.3	Pengujian Mikrokontroler dan ADC .....	29
3.2.4	Pengujian Mikrokontroler .....	29
PERENCANAAN DAN PEMBUATAN ALAT .....		30
4.1	Cara Kerja Alat .....	30
4.2	Perancangan Perangkat Keras .....	32
4.2.1	Rangkaian Catu Daya Sistem .....	32
4.2.2	Rangkaian Filter <i>Lowpass</i> .....	34
4.2.3	Rangkaian Pengkondisi Sinyal .....	35
4.2.4	Rangkaian Mikrokontroler .....	36
4.3	Perancangan Perangkat Lunak .....	38
4.3.1	Analog to Digital Converter .....	38
4.3.2	Algoritma Pendeteksi Suara Musik .....	39
PENGUJIAN DAN ANALISIS DATA .....		49
5.1	Pengujian Perangkat Keras .....	49
5.1.1	Pengujian Microphone .....	49
5.1.2	Pengujian Catu Daya Sistem .....	50
5.1.3	Pengujian Minimum Sistem Mikrokontroler .....	51
5.1.4	Pengujian Rangkaian Filter <i>Lowpass</i> .....	52
5.1.5	Pengujian Penguat Op-amp Non Pembalik .....	55
5.2	Pengujian Perangkat Lunak .....	57
5.2.1	Pengujian Frekuensi Sampling .....	57
5.2.2	Pengujian Output ADC .....	59
5.2.3	Pengujian Persamaan Filter IIR orde empat .....	61
5.2.4	Pengujian Algoritma Pendeteksian Sinyal .....	66
5.3	Pengujian Diagram Blok Sistem Keseluruhan .....	70



KESIMPULAN DAN SARAN .....	72
6.1    Kesimpulan .....	72
6.2    Saran .....	72
DAFTAR PUSTAKA .....	73



## DAFTAR GAMBAR

<i>Gambar 2.1 Simbol Microphone .....</i>	<i>6</i>
<i>Gambar 2.2 Penampang Dynamic Microphone .....</i>	<i>7</i>
<i>Gambar 2.3 Penampang Condenser Microphone .....</i>	<i>7</i>
<i>Gambar 2.4 Respons Low Pass Filter Ideal .....</i>	<i>8</i>
<i>Gambar 2.5 Rangkaian Low Pass Filter Non Inverting .....</i>	<i>9</i>
<i>Gambar 2.6 Respons High Pass Filter Ideal .....</i>	<i>9</i>
<i>Gambar 2.7 Rangkaian High Pass Filter Non Inverting .....</i>	<i>10</i>
<i>Gambar 2.8 Respons Band Pass Filter Ideal .....</i>	<i>10</i>
<i>Gambar 2.9 Respons Band Stop Filter Ideal .....</i>	<i>11</i>
<i>Gambar 2.10 Sinyal Sinusoida Analog .....</i>	<i>12</i>
<i>Gambar 2.11 Contoh Sinyal Diskrit .....</i>	<i>12</i>
<i>Gambar 2.12 Sinyal Pulsa yang Terkontaminasi Noise .....</i>	<i>13</i>
<i>Gambar 2.13 Diagram Blok ADC .....</i>	<i>14</i>
<i>Gambar 2.14 Perbedaan Proses Sampling dan Hold .....</i>	<i>14</i>
<i>Gambar 2.15 Sistem dalam kawasan Z .....</i>	<i>15</i>
<i>Gambar 2.16 Struktur FIR .....</i>	<i>16</i>
<i>Gambar 2.17 Struktur IIR .....</i>	<i>17</i>
<i>Gambar 2.18 Perbedaan respons empat tipe klasik filter IIR .....</i>	<i>18</i>
<i>Gambar 2.19 Diagram Blok ATmega32 .....</i>	<i>21</i>
<i>Gambar 2.20 Konfigurasi Pin ATmega32 .....</i>	<i>22</i>
<i>Gambar 2.21 Konfigurasi Bit ADMUX .....</i>	<i>24</i>
<i>Gambar 2.22 Konfigurasi Bit ADCSRA .....</i>	<i>25</i>
<i>Gambar 2.23 Konfigurasi Bit SFIOR .....</i>	<i>26</i>
<i>Gambar 4.1 Diagram Blok Sistem .....</i>	<i>30</i>
<i>Gambar 4.2 Rangkaian Catu Daya Sistem .....</i>	<i>33</i>
<i>Gambar 4.3 Rangkaian Filter Lowpass .....</i>	<i>34</i>
<i>Gambar 4.4 Rangkaian Penguat Non Inverting .....</i>	<i>36</i>
<i>Gambar 4.5 Minimum Sistem Mikrokontroler .....</i>	<i>37</i>
<i>Gambar 4.6 Diagram Alir Program Interupsi Timer0 .....</i>	<i>39</i>
<i>Gambar 4.7 Diagram Alir Program Interupsi ADC .....</i>	<i>39</i>
<i>Gambar 4.8 Diagram Alir Program Utama .....</i>	<i>41</i>
<i>Gambar 4.9 Realisasi Persamaan (4.18) dari Sistem Waktu Diskrit .....</i>	<i>43</i>
<i>Gambar 4.10 Diagram Alir Filter IIR Butterworth Orde 4 .....</i>	<i>43</i>
<i>Gambar 4.11 Realisasi Persamaan Filter IIR .....</i>	<i>46</i>
<i>Gambar 4.12 Realisasi Persamaan Filter IIR dalam Program .....</i>	<i>47</i>



<i>Gambar 5.1 Rangkaian Pengujian Microphone</i> .....	49
<i>Gambar 5.2 Sinyal Keluaran Microphone</i> .....	50
<i>Gambar 5.3 Hasil Pengujian Rangkaian Catu Daya Menggunakan Osiloskop</i> .....	51
<i>Gambar 5.4 Hasil Pengujian Rangkaian Catu Daya Menggunakan Multimeter</i> .....	51
<i>Gambar 5.5 Diagram Blok Pengujian Minimum Sistem Mikrokontroler</i> .....	52
<i>Gambar 5.6 Diagram Alir Program Pengujian Minimum Sistem Mikrokontroler</i> .....	52
<i>Gambar 5.7 Diagram Blok Pengujian Filter Lowpass</i> .....	53
<i>Gambar 5.8 Pengujian Rangkaian Filter Lowpass</i> .....	53
<i>Gambar 5.9 Grafik Respon Frekuensi Filter Lowpass</i> .....	54
<i>Gambar 5.10 Hasil Pengujian Rangkaian Filter Lowpass dengan Sinyal Masukan Frekuensi 300 Hz</i> .....	54
<i>Gambar 5.11 Hasil Pengujian Rangkaian Filter Lowpass dengan Sinyal Masukan Frekuensi 2400 Hz</i> .....	55
<i>Gambar 5.12 Diagram Blok Pengujian Penguat Op Amp Non Pembalik</i> .....	56
<i>Gambar 5.13 Hasil Pengujian Penguat Op Amp Non Pembalik dengan Sinyal Masukan Frekuensi 1 kHz Amplitudo 100 mVpp</i> .....	56
<i>Gambar 5.14 Hasil Pengujian Penguat Op Amp Non Pembalik dengan Sinyal Masukan Frekuensi 3 kHz Amplitudo 50 mVpp</i> .....	57
<i>Gambar 5.15 Diagram Blok Pengujian Frekuensi Sampling</i> .....	58
<i>Gambar 5.16 Diagram Alir Program Pengujian Frekuensi Sampling</i> .....	58
<i>Gambar 5.17 Hasil Pengujian Frekuensi Sampling</i> .....	59
<i>Gambar 5.18 Diagram Blog Pengujian Output ADC</i> .....	59
<i>Gambar 5.19 Bentuk Tampilan Keluaran Hyperterminal</i> .....	60
<i>Gambar 5.20 Diagram Blok Pengujian Persamaan Filter IIR Orde empat</i> .....	61
<i>Gambar 5.21 Diagram Alir Program Pengujian Persamaan Filter IIR Orde empat</i> ....	62
<i>Gambar 5.22 Grafik Respon Frekuensi Filter IIR Orde Empat</i> .....	66
<i>Gambar 5.23 Diagram Blok Pengujian Algoritma Pendeteksian Sinyal</i> .....	67
<i>Gambar 5.24 Diagram Alir Program Pengujian Algoritma Pendeteksian Sinyal</i> .....	68
<i>Gambar 5.25 Hasil Pengujian dengan Masukan Sinyal Sinusoida Frekuensi 1000 Hz</i>	68
<i>Gambar 5.26 Hasil Pengujian dengan Masukan Sinyal Sinusoida Frekuensi 2800 Hz</i>	69
<i>Gambar 5.27 Diagram Blok Pengujian Sistem Keseluruhan</i> .....	70



## DAFTAR TABEL

Tabel 2.1 <i>Pemilihan Tegangan Referensi ADC</i> .....	24
Tabel 2.2 <i>Pemilihan channel ADC</i> .....	25
Tabel 2.3 <i>Pengaturan Faktor Pembagi Clock Masukan ADC</i> .....	26
Tabel 2.4 <i>Pemilihan Picu ADC otomatis</i> .....	27
Tabel 5.1 <i>Hasil Pengujian Sistem Mikrokontroler</i> .....	52
Tabel 5.2 <i>Hasil Pengujian Rangkaian Filter Lowpass</i> .....	53
Tabel 5.3 <i>Hasil Pengujian Penguat Op Amp Non Pembalik</i> .....	56
Tabel 5.4 <i>Hasil Pengujian Output ADC</i> .....	60
Tabel 5.5 <i>Hasil Pengujian Persamaan Filter IIR Orde empat dengan Sinyal Masukan DC konstan</i> .....	63
Tabel 5.6 <i>Hasil Pengujian Persamaan Filter IIR Orde empat dengan Sinyal Masukan dengan frekuensi 1000 Hz</i> .....	64
Tabel 5.7 <i>Hasil Pengujian Persamaan Filter IIR Orde empat dengan Sinyal Masukan dengan frekuensi 4000 Hz</i> .....	64
Tabel 5.8 <i>Hasil Pengujian Persamaan Filter IIR Orde empat</i> .....	65
Tabel 5.9 <i>Hasil Pengujian Algoritma Pendeteksian Sinyal</i> .....	69
Tabel 5.10 <i>Hasil Pengujian Sistem Keseluruhan</i> .....	71



UNIVERSITAS BRAWIJAYA



## BAB I PENDAHULUAN

### 1.1 Latar Belakang

Perkembangan teknologi robotika akhir-akhir ini menunjukkan suatu peningkatan kualitas yang semakin pesat. Aplikasi teknologi semakin mempermudah pekerjaan manusia dengan adanya otomatisasi. Di sisi lain pengembangan robot sebagai sarana hiburan semakin diminati oleh masyarakat karena kemampuannya menirukan tingkah laku manusia sesuai dengan instruksi yang diberikan.

Kontes Robot Indonesia (KRI), Kontes Robot Cerdas Indonesia (KRCI), dan Kontes Robot Seni Indonesia (KRSI) adalah suatu ajang kontes robot yang diselenggarakan oleh Dirjen Dikti tiap tahunnya untuk menumbuhkembangkan potensi mahasiswa di bidang teknologi robotika. Dalam kontes ini dilombakan robot-robot dari berbagai universitas di Indonesia untuk menyelesaikan suatu misi tertentu. Dalam Kontes Robot Seni Indonesia dua robot diadu dalam suatu arena untuk menunjukkan kemampuannya menari mengikuti alunan suara musik. Kontes yang baru diselenggarakan pertama kali tahun 2009 ini, mengambil tema tarian tradisional Indonesia. Pertama kali diselenggarakan kontes ini mengambil tema robot penari jaipong pada tahun 2009, dan pada tahun 2010 mengambil tema robot penari pendet. Dimungkinkan pada tahun-tahun berikutnya, tema akan berubah-ubah.

Tujuan dari kontes robot ini adalah untuk menumbuh kembangkan kreatifitas dan minat mahasiswa dalam teknologi maju, khususnya teknologi robotika yang selain diperuntukkan bagi dunia industri juga diharapkan dapat membantu kegiatan manusia. Selain itu juga mengaplikasikan ilmu pengetahuan dan teknologi ke dalam dunia nyata dan meningkatkan kepekaan mahasiswa dalam pengembangan bidang teknologi robotika, dan membudayakan iklim kompetitif di lingkungan perguruan tinggi.

Dalam KRSI, robot harus mampu menari ketika musik mulai diperdengarkan dan berhenti ketika musik dimatikan. Hal ini membutuhkan suatu sistem yang dapat membedakan apakah sinyal yang diterima oleh sensor adalah sinyal yang berasal dari suara musik atau bukan. Karena pengolah data yang digunakan dalam sistem ini merupakan suatu komponen digital, maka sinyal yang diperoleh yang berupa sinyal analog harus dikonversi ke dalam bentuk digital menggunakan *Analog to Digital Converter*.



Pengolah data yang berupa mikrokontroler harus mampu membedakan antara sinyal informasi dan noise yang ditangkap oleh sensor. Sensor yang digunakan dalam sistem ini adalah *microphone*. Sinyal informasi yang diharapkan adalah sinyal yang berasal dari suara musik, sedangkan sinyal *noise* berasal dari suara manusia. Dalam arena lomba yang sesungguhnya, suara manusia yang dianggap *noise* ini adalah suara pembawa acara dan juri dalam lomba.

Sistem digital juga akan mempermudah dalam hal penyimpanan data, tanpa mengalami penurunan atau kehilangan keaslian sinyalnya. Sebagai konsekuensinya, sinyal-sinyal tersebut akan lebih mudah untuk dipindah-pindahkan ke sistem yang lain. Walaupun tidak dipungkiri akan banyak kesulitan dalam pembuatan algoritma program yang bagus, termasuk juga batasan-batasan komponen digitalnya, misalnya kecepatan pemrosesan datanya.

## 1.2 Rumusan Masalah

Berdasarkan masalah yang telah dijelaskan dalam latar belakang, dapat dibuat rumusan sebagai berikut:

- 1). Bagaimana membuat sistem pendeteksi suara musik menggunakan *microphone* sebagai sensor suara
- 2). Bagaimana merancang sistem antarmuka mikrokontroler dengan *microphone*
- 3). Bagaimana membuat sistem yang mampu mendeteksi musik menggunakan mikrokontroler AVR.

## 1.3 Batasan Masalah

Dalam perancangan untuk skripsi ini permasalahan dibatasi oleh hal-hal sebagai berikut :

- 1). Hanya membahas tentang bagaimana mendeteksi adanya suara musik
- 2). Filter analog yang digunakan adalah *Low Pass Filter* dan filter digital yang digunakan adalah *Band Pass Filter Butterworth* orde empat
- 3). Sensor yang digunakan adalah *microphone* dinamik
- 4). Mikrokontroler yang digunakan adalah ATmega32
- 5). Output sistem berupa logika satu atau nol yang menandakan ada atau tidak ada musik yang diperdengarkan.

## 1.4 Tujuan

Tujuan penelitian ini adalah merealisasikan suatu sistem pendeteksi musik pada robot penari sehingga mampu mendeteksi ada atau tidaknya suara musik.

## 1.5 Sistematika Penulisan

Penulisan skripsi ini terdiri dari enam bagian dengan sistematika sebagai berikut:

### **Bab I: Pendahuluan**

Berisi tentang uraian latar belakang, tujuan, batasan masalah, rumusan masalah, manfaat serta sistematika penulisan.

### **Bab II: Tinjauan Pustaka**

Membahas teori-teori yang mendukung dalam perencanaan dan pembuatan sistem.

### **Bab III: Metodologi Penelitian**

Membahas tentang metode-metode yang digunakan dalam penyusunan skripsi ini.

### **Bab IV: Perancangan**

Berisi perancangan dan perealisasiian sistem yang meliputi spesifikasi, perencanaan diagram blok, prinsip kerja dan realisasi sistem.

### **Bab V: Pengujian dan Analisis**

Membahas tentang proses pengujian dan analisis data yang diperoleh dari sistem yang telah dibuat.

### **Bab VI: Kesimpulan dan Saran**

Memuat kesimpulan dan saran-saran untuk pengembangan lebih lanjut dari sistem yang telah dibuat.



## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Kontes Robot Seni Indonesia

Dalam setiap sesi pertandingan, dua robot dari tim peserta akan diletakkan di atas panggung, masing-masing robot diletakkan dalam lingkaran putih berdiameter 2000mm, tebal garis putih lingkaran pembatas adalah sekitar 30mm. Lingkaran ini berfungsi sebagai batas gerak robot dalam melakukan gerakan tari mengikuti alunan musik yang disediakan oleh panitia. Tim Juri akan berkeliling untuk memberikan nilai untuk setiap robot tim peserta. Lagu instrumental (musik) yang digunakan akan ditentukan oleh panitia.

##### 2.1.1 Spesifikasi Robot

- Robot otomatis memiliki kemampuan melakukan gerakan menari mengikuti musik
- Robot yang dibuat memiliki kemampuan untuk mengikuti alunan musik yang dimainkan
- Robot dapat dirancang untuk dapat mendengarkan alunan musik melalui “telinga” atau sensor, atau pun mendengar melalui kabel atau kateter
- Robot memiliki bagian yang disebut sebagai sistem kaki, tangan, dan kepala. Jumlah derajat kebebasan masing-masing sistem ini tidak dibatasi
- Robot dapat dibuat beroda atau pun berkaki, tetapi harus memiliki lengan dan/atau jari untuk melakukan gerakan mengikuti alunan musik yang digunakan
- Robot tidak memecah diri menjadi beberapa robot dan bagian-bagian robot yang tidak dapat bergerak kecuali selendang atau bunga sebagai pelengkap tariannya
- Berat total seluruh robot tidak melebihi 50kg, dan batas tegangan catu daya tidak melebihi 24V dc
- Semua aktuator gerak berbasis elektromotor, tidak menggunakan sistem pneumatik maupun sistem hidrolis
- Gerakan tarian semua robot yang dibuat tidak melebihi area berdiameter 2000mm dilihat dari atas
- Pada saat start, ukuran seluruh robot yang digunakan tidak melebihi (L)1000mm × (W)1000mm × (H)1000mm



- Tinggi robot minimal 500mm



- Jika robot dirancang untuk mendengar musik melalui sensor (wireless), musik pengiring diputar melalui loudspeaker standar yang digunakan di arena
- Jika robot menggunakan sistem hubungan kabel, digunakan kabel output audio male stereo ukuran 3,5mm dengan spesifikasi standar AUDIO LINE OUT (impedansi output sekitar  $100\Omega$ , 300mVpp)
- Setiap robot dapat di-start hanya dengan satu tombol di badan robot diletakkan pada posisi yang mudah terjangkau.

### 2.1.2 Arena Pengujian

- Arena lomba terdiri dari dua buah lingkaran, satu lingkaran berwarna merah dan satu lingkaran berwarna biru diletakkan di atas bidang datar, masing-masing lingkaran dibatasi dengan lingkaran bergaris putih dengan diameter 2000mm
- Tebal garis putih pembatas adalah 30mm.

### 2.1.3 Microphone

*Microphone* adalah suatu transduser akustik ke elektrik atau suatu sensor yang mengubah suara ke dalam sinyal listrik. *Microphone* banyak digunakan dalam berbagai aplikasi, misalnya telepon dan *tape recorder*. Desain yang umum digunakan saat ini adalah menggunakan suatu membran tipis yang bergetar sebagai respon terhadap tekanan suara. Getaran ini kemudian diterjemahkan ke dalam sinyal listrik.

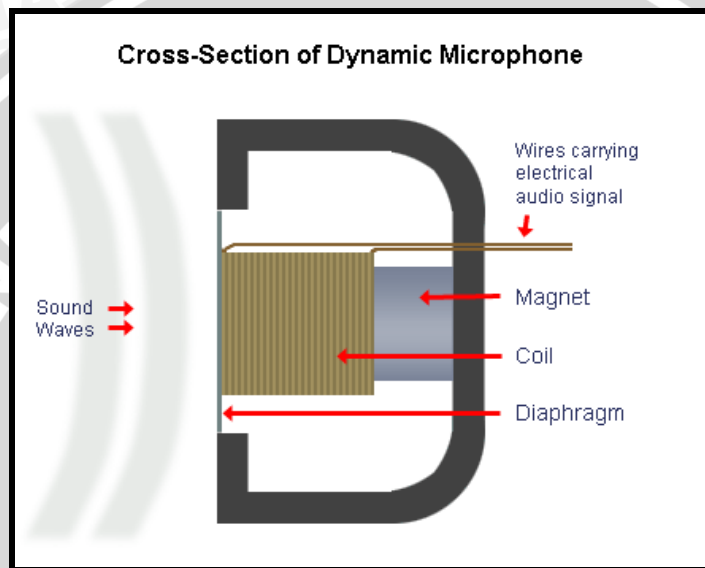


Gambar 2.1 Simbol *Microphone*

*Microphone* memiliki karakteristik elektrik yang disebut impedansi, dalam satuan ohm dan bergantung pada desainnya. Secara umum, besar impedansi ini telah ditentukan. Impedansi rendah bernilai kurang dari  $600\Omega$ , impedansi medium berkisar antara  $600\Omega$  sampai  $10k\Omega$ , dan impedansi tinggi bernilai lebih dari  $10k\Omega$ .

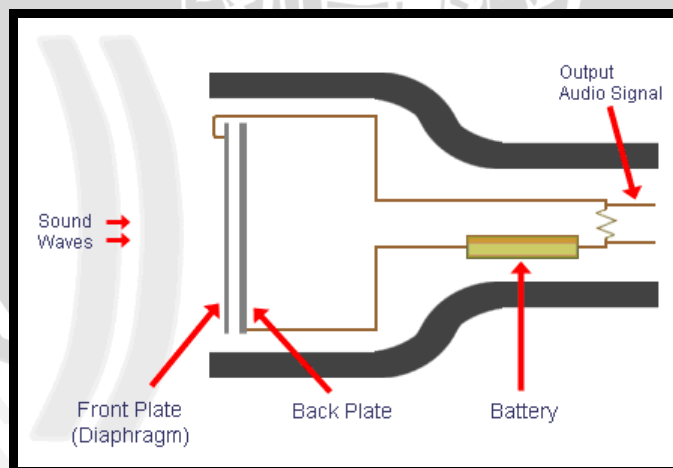
*Microphone* terdiri dari berbagai macam jenis. Antara lain *dynamic microphone*, *condenser microphone*, *ribbon microphone*, *carbon microphone*, *piezoelectric microphone* dan sebagainya. *Microphone* dinamik bekerja dengan menggunakan induksi elektromagnetik menggunakan kumparan dan membran sebagai media pengubah energi

suara menjadi energi listrik, dan mempunyai respon frekuensi antara 40Hz sampai 16kHz. Skema *microphone* dinamik dapat dilihat dalam Gambar 2.2. *Microphone* condenser menggunakan sebuah elektroda metal yang sangat tipis yang biasanya berupa lembar plastik yang dilapisi metal ditempatkan di depan elektroda lain yang disebut dengan *counter electrode* yang dibuat dari logam atau keramik yang dilapisi oleh logam. Kedua keeping ini berlaku sebagai kapasitor. *Microphone* jenis ini lebih sensitif dengan respon frekueansi antara 50Hz sampai 12kHz. Skema *microphone* condenser ditunjukkan dalam Gambar 2.3.



Gambar 2.2 Penampang Dynamic Microphone

Sumber: Anonim



Gambar 2.3 Penampang Condenser Microphone

Sumber: Anonim



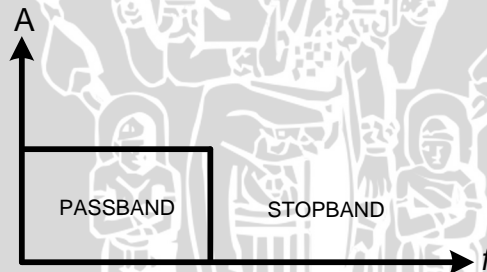
## 2.2 Filter

Filter adalah suatu rangkaian elektronik yang digunakan untuk melewatkan sinyal dengan rentang frekuensi tertentu dan meredam sinyal dalam rentang frekuensi yang lain. Filter terdiri atas beberapa jenis yaitu *Low Pass Filter*, *Band Pass Filter*, *High Pass Filter*, dan *Band Stop Filter*.

Jangkauan frekuensi yang dilalui filter dengan penguatan maksimum atau pelemahan minimum disebut *passband*. *Passband limit* biasanya diasumsikan sebagai frekuensi di mana penguatannya turun 3 dB ( $1/\sqrt{2}$  atau 0,707 dari penguatan tegangan maksimumnya). Frekuensi tersebut disebut dengan frekuensi *cutoff* atau frekuensi -3 dB.

### 2.2.1 Low Pass Filter

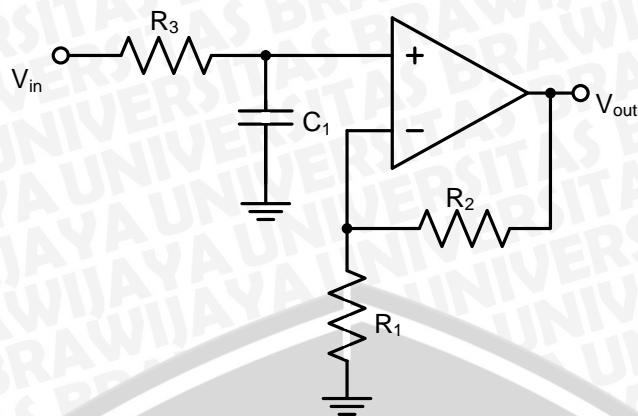
*Low pass filter* melewatkan semua sinyal dengan frekuensi mulai dari nol sampai dengan frekuensi *cutoff* dan meredam semua sinyal dengan frekuensi lebih besar daripada frekuensi *cutoff*. Frekuensi antara nol dan frekuensi *cutoff* disebut *pass band*, sedangkan frekuensi yang lebih dari frekuensi *cutoff* disebut *stop band*. Respon *low pass filter* ideal ditunjukkan dalam Gambar 2.4.



Gambar 2.4 Respons Low Pass Filter Ideal

Sumber: Malvino 2004: 240

Rangkaian *low pass filter* aktif orde satu *non inverting* ditunjukkan dalam Gambar 2.5.



Gambar 2.5 Rangkaian Low Pass Filter Non Inverting

Sumber: Malvino 2004: 263

Penguatan sinyal pada daerah *pass band* dirumuskan sebagai berikut:

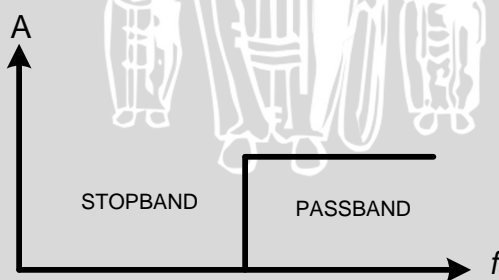
$$A_v = \frac{R_2}{R_1} + 1 \quad (2.1)$$

Frekuensi *cutoff* dapat dihitung dengan menggunakan rumus:

$$f_c = \frac{1}{2\pi R_3 C_1} \quad (2.2)$$

### 2.2.2 High Pass Filter

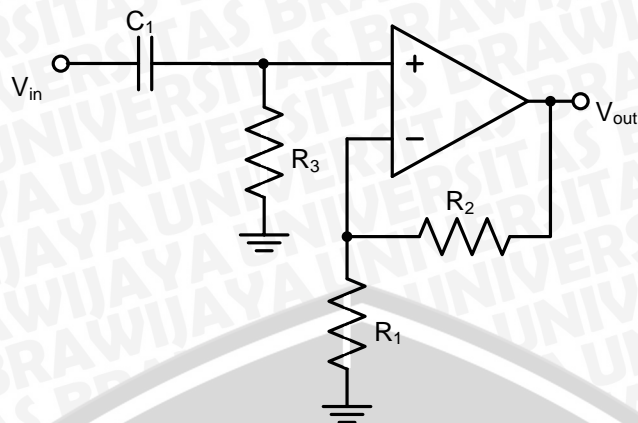
*High pass filter* meredam sinyal dengan frekuensi antara nol sampai frekuensi *cutoff* dan melewatkan sinyal dengan frekuensi lebih dari frekuensi *cutoff*. Respons *high pass filter* ideal ditunjukkan dalam Gambar 2.6.



Gambar 2.6 Respons High Pass Filter Ideal

Sumber: Malvino 2004: 241

Rangkaian high pass filter aktif orde satu non inverting ditunjukkan dalam Gambar 2.7.



Gambar 2.7 Rangkaian High Pass Filter Non Inverting

Sumber: Malvino 2004: 265

Penguatan sinyal pada daerah pass band dirumuskan sebagai berikut:

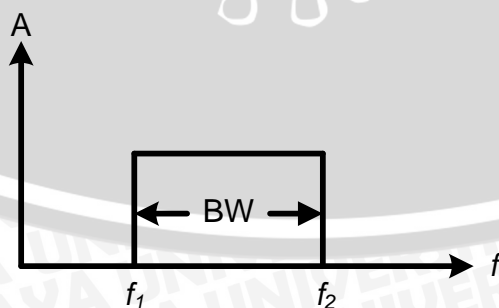
$$A_v = \frac{R_2}{R_1} + 1 \quad (2.3)$$

Frekuensi cutoff dirumuskan dengan:

$$f_c = \frac{1}{2\pi R_3 C_1} \quad (2.4)$$

### 2.2.3 Band Pass Filter

Band pass filter meredam semua sinyal berfrekuensi mulai dari nol sampai frekuensi *cutoff* pertama, melewatkan sinyal berfrekuensi antara frekuensi *cutoff* pertama dan frekuensi *cutoff* kedua, dan meredam sinyal dengan frekuensi lebih dari frekuensi *cutoff* kedua. Frekuensi *cutoff* pertama selalu lebih kecil daripada frekuensi *cutoff* kedua. Respon *band pass filter* ideal ditunjukkan dalam Gambar 2.8.



Gambar 2.8 Respon Band Pass Filter Ideal

Sumber: Malvino 2004: 241

Bandwidth dari suatu band pass filter dirumuskan dengan:

$$BW = f_2 - f_1 \quad (2.5)$$



Frekuensi tengah *band pass filter* dilambangkan dengan  $f_0$  dan dirumuskan dengan:

$$f_0 = \sqrt{f_1 f_2} \quad (2.6)$$

Faktor mutu, Q ditetapkan sebagai frekuensi tengah dibagi dengan *bandwidth*:

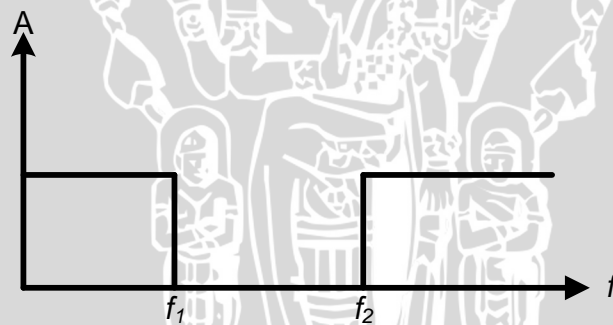
$$Q = \frac{f_0}{BW} \quad (2.7)$$

Ketika Q bernilai lebih dari 10, frekuensi tengah dapat ditentukan sebagai nilai rata-rata dari kedua frekuensi *cutoff*, sebagaimana rumus berikut:

$$f_0 = \frac{f_1 + f_2}{2} \quad (2.8)$$

### 2.2.4 Band Stop Filter

Band stop filter melewatkan sinyal dengan frekuensi antara nol dan frekuensi *cutoff* pertama, meredam sinyal berfrekuensi antara frekuensi *cutoff* pertama dan frekuensi *cutoff* kedua, dan melewatkan sinyal dengan frekuensi lebih dari frekuensi *cutoff* kedua. Respon band stop filter ideal ditunjukkan dalam Gambar 2.9.



Gambar 2.9 Respons Band Stop Filter Ideal

Sumber: Malvino 2004: 243

### 2.3 Sinyal

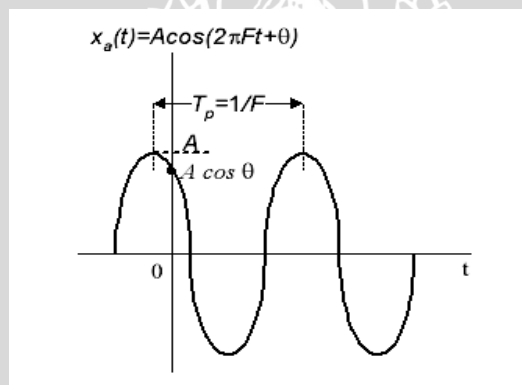
Sinyal adalah besaran fisis yang berubah menurut waktu, ruang, atau variabel bebas lainnya dan merupakan fungsi yang membawa informasi. Secara matematis, sinyal adalah fungsi dari satu atau lebih variabel independen. Proses ini dilakukan melalui pemodelan sinyal. Sinyal dalam kawasan waktu terdiri dari dua jenis, yaitu sinyal analog dan sinyal digital. Suatu sinyal analog  $s$  didefinisikan dengan suatu fungsi yang bernilai riil  $s(t)$  dari suatu variabel kontinyu  $t$  (disebut waktu), didefinisikan untuk semua waktu dalam interval  $-\infty < t < +\infty$ . Sedangkan suatu sinyal digital  $s$  didefinisikan

repository.ub.ac.id

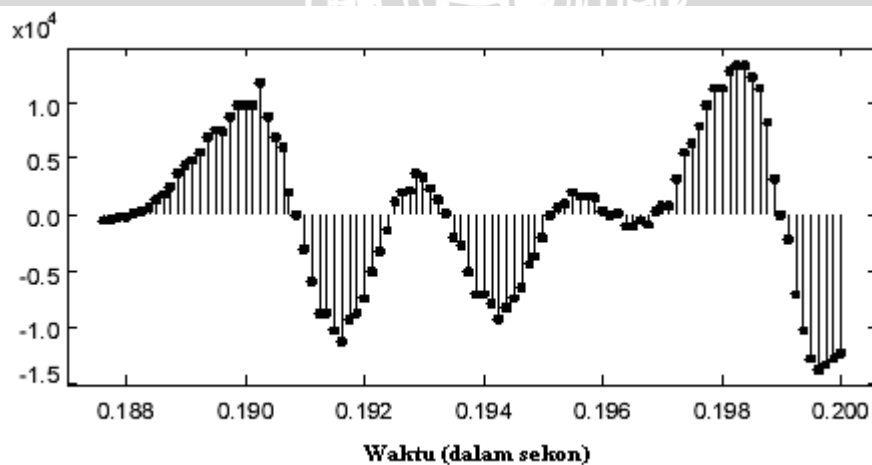
dengan urutan nilai-nilai diskrit terbatas  $s(n)$  dengan indeks tunggal  $n$  (disebut waktu diskrit), didefinisikan pada waktu-waktu tertentu  $n = -\infty, \dots, +\infty$ .

Sinyal analog harus bernilai riil dan tidak bernilai integer atau kompleks adalah karena sinyal pada dunia nyata, seperti kecepatan, tegangan, dan tekanan akustik adalah variabel kontinu yang sederhana. Sedangkan sinyal digital harus memiliki nilai-nilai representatif pada komputer digital dan tidak seperti kenyataan fisiknya. Kata diskrit berarti bahwa nilai yang sesungguhnya dikuantisasi menjadi nilai-nilai diskrit, misalnya integer.

Sebagai contoh sinyal waktu kontinu adalah sinyal sinusoida analog yang diperlihatkan dalam Gambar 2.10 dan sebagai contoh sinyal waktu diskrit diperlihatkan dalam Gambar 2.11.



Gambar 2.10 Sinyal Sinusoida Analog



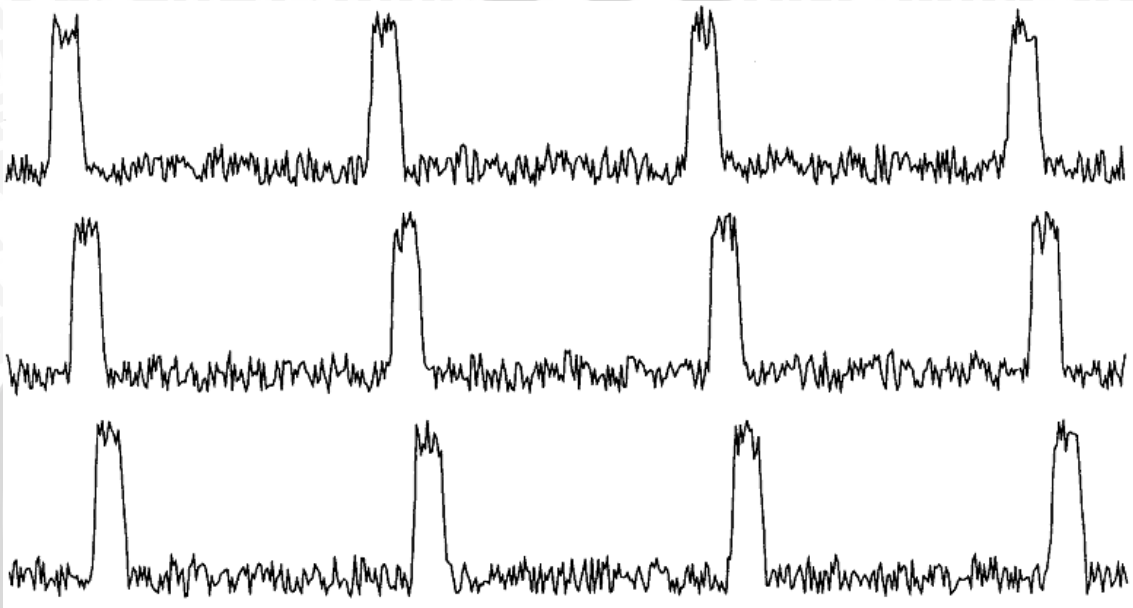
Gambar 2.11 Contoh Sinyal Diskrit

Sumber: Lee 2000:4

UNIVERSITAS BRAWIJAYA

## 2.4 Noise

Noise adalah suatu sinyal acak yang mengandung informasi yang tidak diharapkan. Sinyal acak ini terintegrasi dengan sinyal informasi yang diinginkan. Oleh karena itu noise harus diredam untuk memperoleh sinyal informasi yang diharapkan tersebut. Gambar 2.12 menunjukkan satu sinyal informasi dengan noise yang berbeda-beda, namun demikian bentuk pulsa yang dihasilkan masih tetap sama.



Gambar 2.12 Sinyal Pulsa yang Terkontaminasi Noise

Sumber: Stein 2000: 172

## 2.5 Sistem Waktu Diskrit

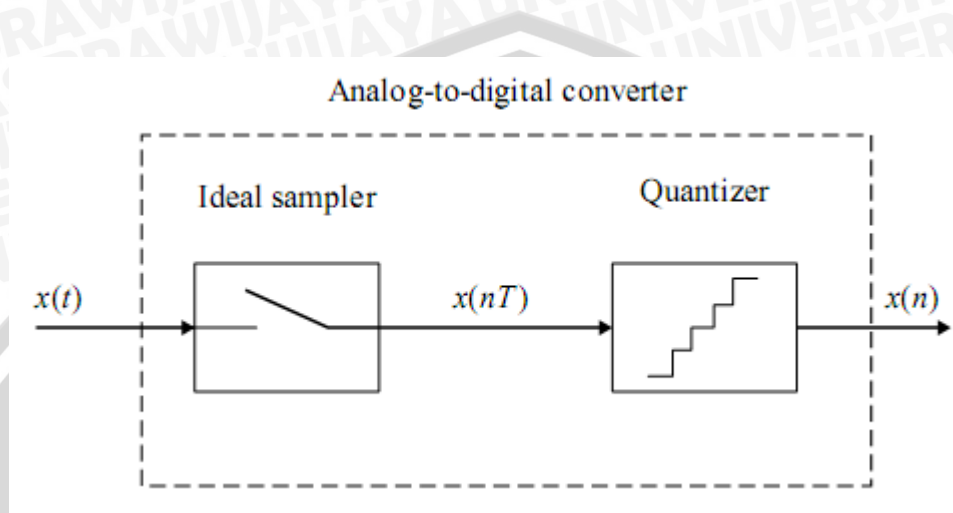
Sistem waktu diskrit melibatkan sinyal waktu diskrit pula. Untuk mendapatkan suatu sinyal digital dari sinyal analog, perlu diterapkan tiga tahapan sebagai berikut :

1. *Pencuplikan(Sampling)*, merupakan proses konversi suatu sinyal analog waktu kontinyu menjadi suatu sinyal diskret yang diperoleh dengan mengambil cuplikan sinyal waktu kontinyu pada saat waktu diskret. Jika misalnya  $x_a(t)$  adalah masukan terhadap pencuplik maka keluaran adalah  $x_a(nT) \equiv x(n)$  dengan  $T$  dinamakan selang pencuplikan. Sedangkan proses menjaga harga tegangan masukan pada saat proses konversi disebut *Hold*.
2. *Kuantisasi*, merupakan proses konversi sinyal yang bernilai-kontinyu waktu diskret menjadi sinyal bernilai diskret waktu diskret (digital). Nilai setiap cuplikan sinyal digambarkan dengan suatu nilai terpilih dari himpunan



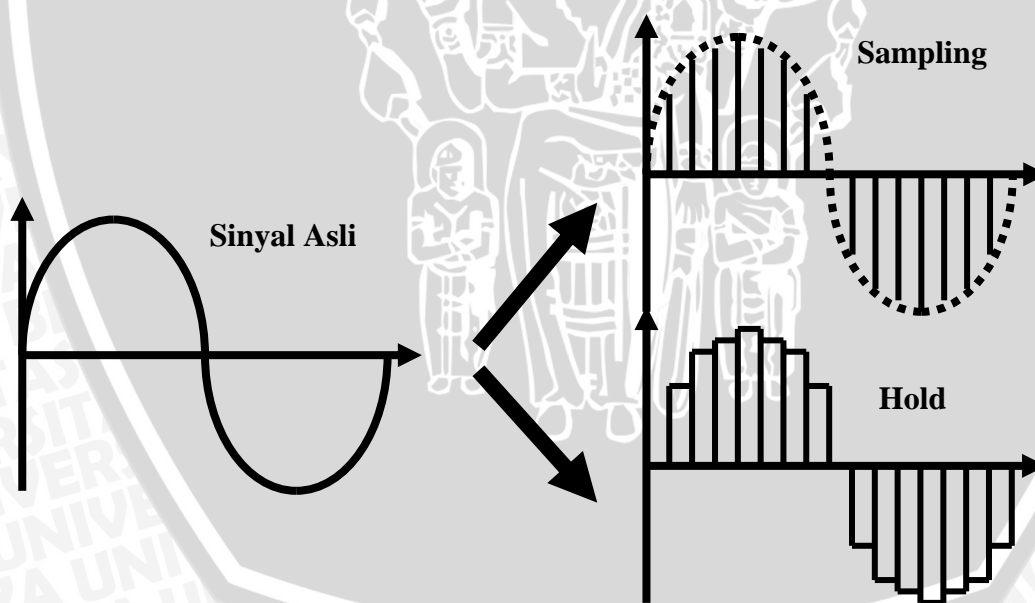
berhingga nilai-nilai yang mungkin. Selisih antara cuplikan  $x(n)$  yang tidak terkuantisasi dan keluaran  $x_q(n)$  yang terkuantisasi dinamakan Galat Kuantisasi (Quantization Error).

3. *Pengkodean*, dalam proses ini setiap nilai diskret  $x_q(n)$  digambarkan dengan suatu barisan biner.



Gambar 2.13 Diagram Blok ADC

Sumber: Wiley 2006: 4



Gambar 2.14 Perbedaan Proses Sampling dan Hold

Sumber: Anonim

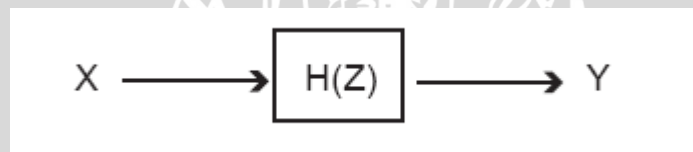
## 2.6 Filter Digital

Filter digital dibagi menjadi dua jenis, yaitu *Finite Impulse Response* filter (filter FIR) dan *Infinite Impulse Response* filter (filter IIR). FIR memiliki respon impuls  $h(n)$  berharga tidak nol hanya pada jumlah tertentu dari suatu sampling sinyal, sedangkan IIR respon impulsnya memiliki jumlah tak terhingga dari sample berharga tidak nol. Output filter FIR hanya dipengaruhi oleh input filter dan dihitung menggunakan algoritma yang tidak berulang (*feed forward*). Output filter IIR berdasarkan input dan umpan balik dari output, dan outputnya ini dihitung menggunakan algoritma yang berulang.

Persamaan kedua filter ini secara umum adalah:

$$\sum_{m=0}^M a_m \cdot y[n - m] = \sum_{k=0}^N b_k \cdot x[n - k] \quad (2.14)$$

Bentuk transformasi Z-nya dari persamaan 2.14 adalah :



Gambar 2.15 Sistem dalam kawasan Z

Sumber: Atmel Application Note AVR223 : 2

$$\sum_{m=0}^M a_m \cdot y[n - m] = \sum_{k=0}^N b_k \cdot x[n - k] \xleftrightarrow{Z} Y(z) \cdot \sum_{m=0}^M a_m \cdot z^{-m} = X(z) \cdot \sum_{k=0}^N b_k \cdot z^{-k} \quad (2.15)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k \cdot z^{-k}}{\sum_{m=0}^M a_m \cdot z^{-m}} \quad (2.16)$$

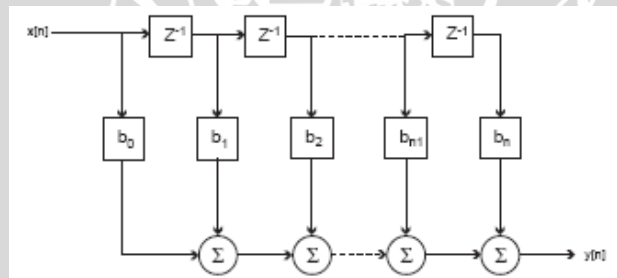
Koefisien filter yang berkaitan dengan bagian *feedback* dan *feedforward* dituliskan sebagai  $a_m$  dan  $b_k$ .

Karena FIR keluaran filter hanya tergantung pada input saja, maka nilai  $M=0$  dan  $N!=0$ , sehingga persamaan umum filter FIR adalah:

$$y[n] = \sum_{k=0}^N b_k \cdot x[n - k] \quad (2.17)$$

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k \cdot z^{-k}}{1} = \sum_{k=0}^N b_k \cdot z^{-k} \quad (2.18)$$

FIR filter akan selalu stabil mengingat filter ini bersifat BIBO (*Bounded Input Bounded Output*). FIR bisa dipakai baik sebagai HPF (*High Pass Filter*) dan LPF (*Low Pass Filter*), tergantung pada pemilihan koefisien yang tepat untuk mendapatkan respon frekuensi yang sesuai. FIR baik untuk menghilangkan frekuensi sinyal yang tidak diinginkan. FIR juga sering dipakai untuk menghaluskan sinyal dan meminimalisasi fluktuasi sinyal terhadap noise. Struktur filter FIR ditunjukkan dalam Gambar 2.16.



Gambar 2.16 Struktur FIR

Sumber: Atmel Application Note AVR223 :4

Sedangkan pada IIR keluaran filter selain dipengaruhi oleh masukannya, juga dipengaruhi oleh umpan balik (*feedback*) dari keluaran filter. IIR lebih efisien daripada FIR, namun jenis filter ini bersifat kurang stabil. IIR sering dipakai untuk aplikasi yang membutuhkan *single frequency* dan pita sempit. Persamaan filter ini adalah:

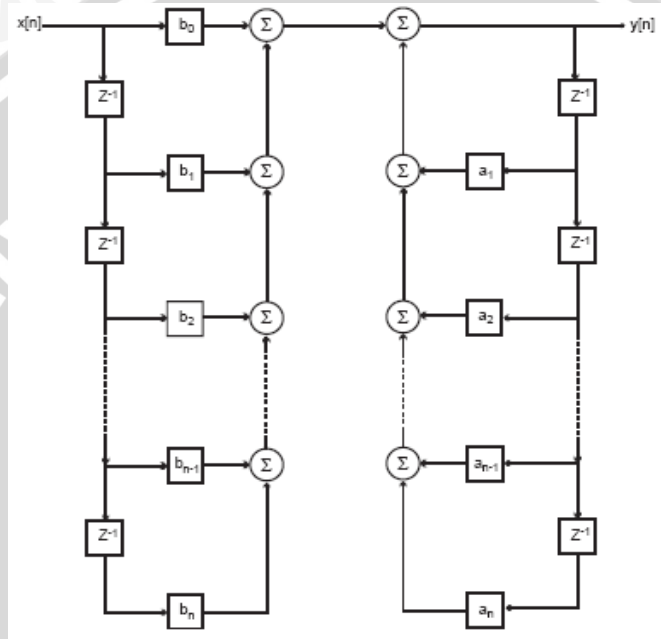
$$y[n] = \sum_{k=0}^N b_k \cdot x[n - k] + \sum_{m=1}^M -a_m \cdot y[n - m] \quad (2.19)$$



$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k \cdot z^k}{1 + \sum_{m=1}^M a_m \cdot z^m} \quad ; \text{for } a_0 = 1$$

(2.20)

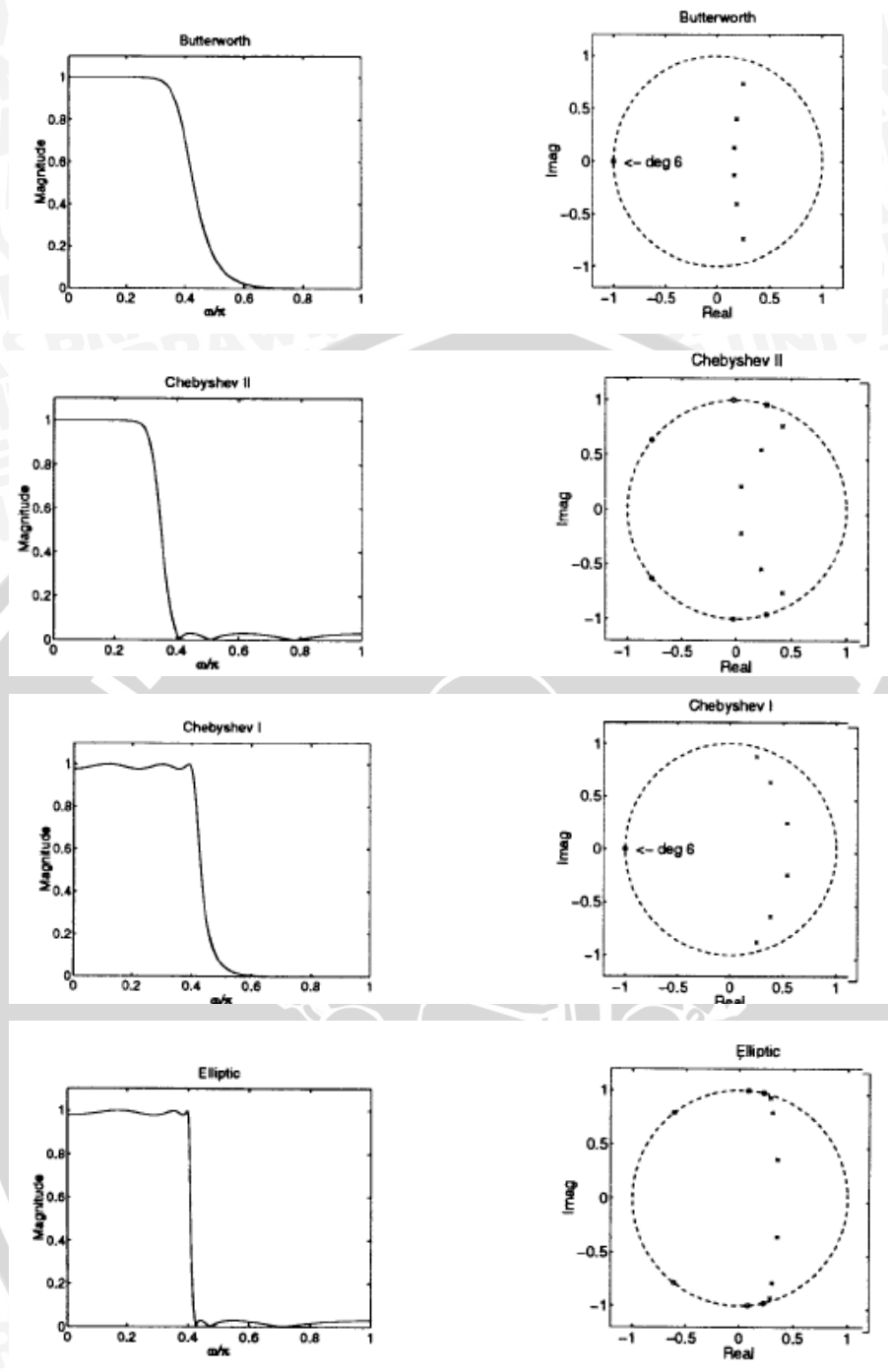
Struktur filter IIR ditunjukkan dalam Gambar 2.17.



Gambar 2.17 Struktur IIR

Sumber: Atmel Application Note AVR223 : 6

Filter IIR dibagi menjadi empat jenis tipe, yaitu Butterworth, Chebyshev I, Chebyshev II, dan Elliptic. Perbandingan respon keempat filter ini ditunjukkan dalam Gambar 2.18.



Gambar 2.18 Perbedaan respons empat tipe klasik filter IIR

Sumber: Anonim

Karakteristik filter IIR adalah sebagai berikut :

1. Respon impuls ( $h(n)$ ) memiliki sample berharga tidak nol sebanyak tak terhingga (*infinite length*).
2. Frekuensi respon  $H(\omega)$  merupakan fungsi rasional, yaitu merupakan rasio perbandingan dua polinomial  $e^{j\omega}$  berorde tertentu, yaitu :

$$H(\omega) = \frac{B(\omega)}{A(\omega)} = e^{-j\omega N_0} \frac{\sum_{k=0}^M b_k e^{-j\omega k}}{\sum_{k=0}^N a_k e^{-j\omega k}}$$

yang mana  $N_0$  adalah konstanta bilangan bulat. Pangkat dari IIR tergantung pada pangkat  $N$  pada penyebutnya.  $M$  pada pembilang akan berharga lebih kecil dari  $N$ . Bilangan  $N$  juga akan mempengaruhi jumlah dari output sample sebelumnya. IIR juga disebut *feedback system*. Koefisien filter  $b_n$  dan  $a_n$  merupakan parameter yang harus dicari sesuai dengan spesifikasi sistem.

3. Desain filter digital IIR merupakan metode untuk menemukan persamaan rasional  $H(\omega)$  yang mendekati spesifikasi desain yang diinginkan. Dalam domain frekuensi, hal ini bisa dilakukan dengan menghitung koefisien optimal dari  $b_n$  dan  $a_n$  dalam persamaan rasional tersebut. Orde filter  $N$  biasanya sudah tetap, tetapi dapat dipertimbangkan untuk lebih dioptimalkan.
4. Pada bagian sebelumnya disebutkan bahwa fungsi transfer filter dinotasikan dalam  $H(z)$  yang merupakan transformasi  $z$  dari  $h(n)$  yang bisa dipakai untuk mengetahui kestabilan sistem. Dalam konteks filter LSI kestabilan akan menyebabkan kondisi *Bounded Input Bounded Output* artinya perubahan kecil pada masukan juga akan menyebabkan perubahan pada output. Sistem yang stabil dan kausal ditandai dengan pole yang berada pada daerah unit circle (Unit Circle terletak di  $|z|=1$  pada bidang  $z$ ). Pole merupakan akar dari bagian pembilang pada fungsi transfer  $H(z)$ .
5. Filter kausal fasa linier tidak mungkin diperoleh karena filter IIR kausal yang dihasilkan akan selalu bersifat fasa non linier. Filter dengan *forward backward* dapat diterapkan untuk mendapatkan pendekatan terhadap respon fasa nol.
6. Karena impulse responnya bersifat tidak terhingga, konvolusi tidak dapat digunakan sebagai metode untuk filter IIR. Akan lebih baik jika filter IIR menggunakan metode persamaan beda dengan umpan balik (*feedback difference equation*).
7. Karakteristik noise filter IIR merupakan pertimbangan utama saat akan menerapkan jenis filter ini, terutama dalam *fixed point arithmetic*.
8. Pada filter IIR, spesifikasi filter yang diinginkan bisa diperoleh dengan menggunakan orde filter yang kecil, sehingga parameter-parameter filter akan lebih sedikit yang akan dihitung dan disimpan, yang artinya akan lebih



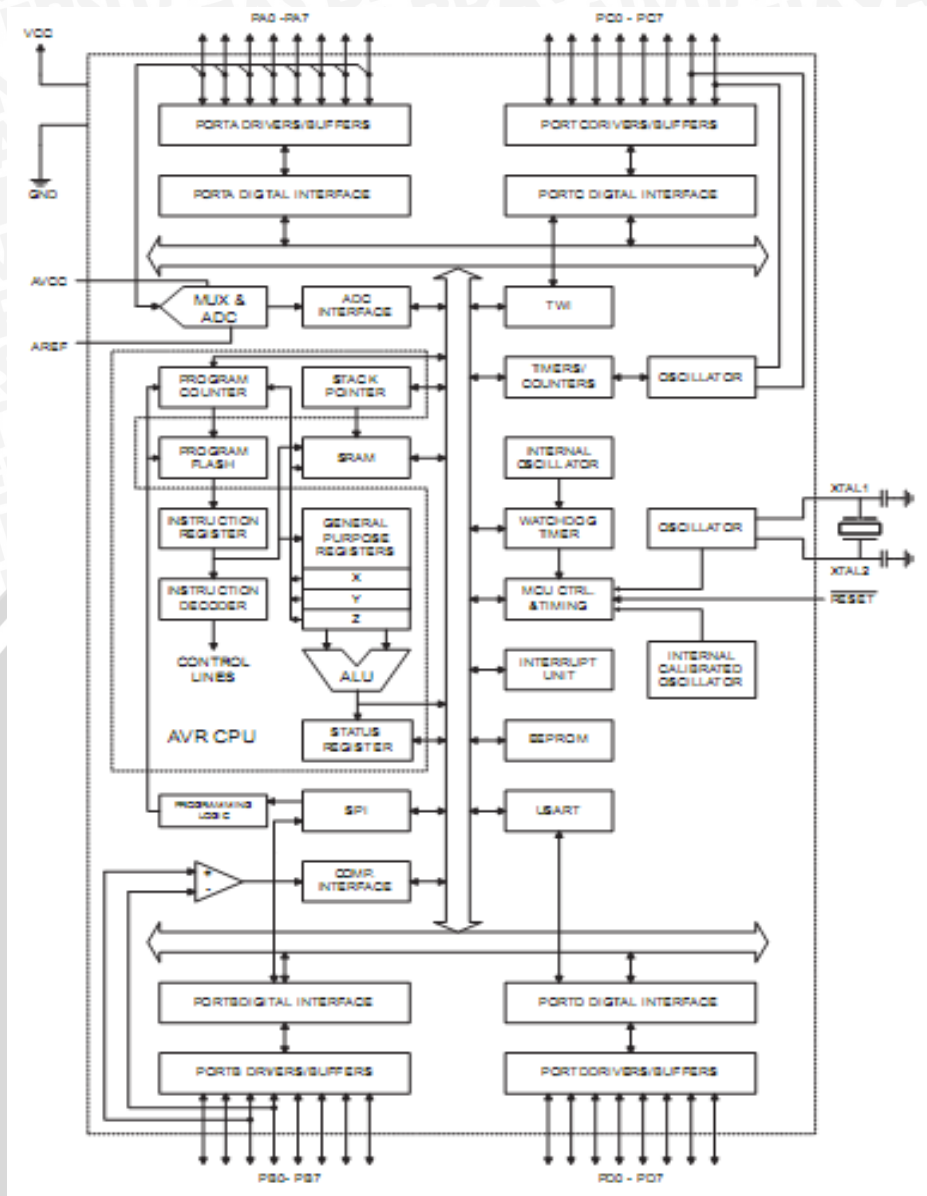
mengurangi kesulitan dalam desain. Respon phase dari filter IIR tidak pernah linier sehingga diperlukan menggunakan *all pass filter* untuk mengkompensasi *delay* yang terjadi, yang kemudian akan meningkatkan orde filter serta kompleksitasnya.

9. IIR biasanya dirancang dengan menggunakan rumusan bentuk tertutup yang erat hubungannya dengan jenis filter klasik. Jika filter FIR menggunakan rumusan dengan metode pendekatan, maka rumusan pada IIR sudah pasti karena melalui penurunan sifat-sifat matematis pada prototype filter klasik. Rumusan ini sangat berguna untuk mendapatkan orde dari filter IIR untuk memenuhi spesifikasi yang diperlukan.

### 2.7 Mikrokontroler AVR ATmega32

ATmega32 adalah salah satu mikrokontroler AVR buatan ATMEL keluarga ATmega yang memiliki 32K byte memori program Flash, 1024 byte EEPROM (*Electrical Erasable Programmable Read Only Memory*), 2K byte SRAM (*Static Random Access Memory*), 32 jalur I/O, dan 32 register serba guna. Mikrokontroler ini memiliki beberapa fitur periferil yaitu dua buah *Timer/Counter* 8 bit, satu buah *Timer/Counter* 16 bit, 4 channel PWM, 8 channel ADC 10 bit, antarmuka serial 2 kabel berorientasi byte, serial USART, antarmuka serial SPI *Master/Slave*, *watchdog timer*, dan *analog comparator*.

Mikrokontroler ATmega32 merupakan mikrokomputer 8 bit yang terintegrasi dalam satu keping IC (*single chip microcomputer*) dan menggunakan arsitektur RISC (*Reduced Instruction Set Computing*) 8 bit. Kombinasi antara arsitektur RISC, memori *flash* internal dan 32 buah register menyebabkan ukuran kode program, kinerja, dan konsumsi daya yang optimal. Sebagian besar instruksi AVR dieksekusi dalam satu siklus *clock*. Kelebihan lainnya, arsitektur AVR dirancang untuk bekerja secara efisien menggunakan bahasa pemrograman tingkat tinggi C. Diagram blok mikrokontroler ATmega32 ditunjukkan dalam Gambar 2.19.



Gambar 2.19 Diagram Blok ATmega32

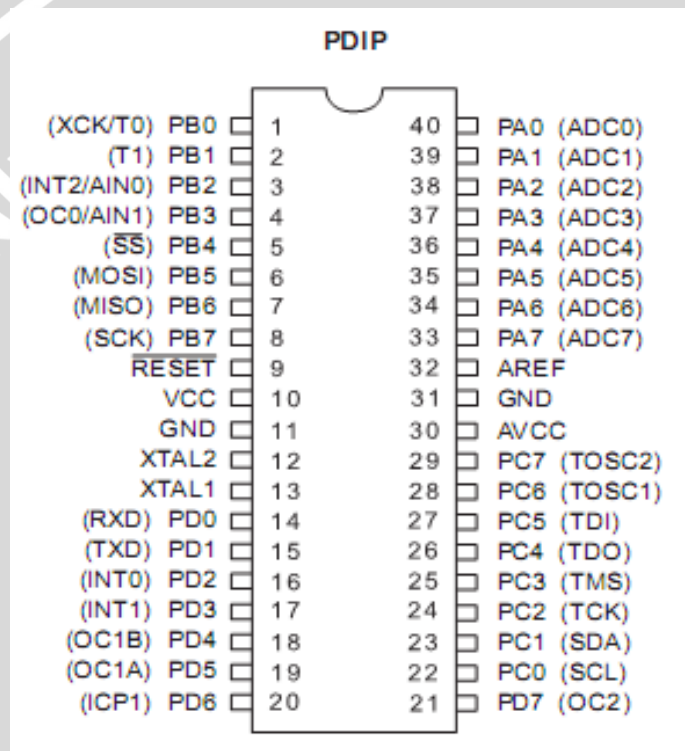
Sumber: Datasheet ATmega32

Berdasarkan diagram blok tersebut dapat diketahui bahwa ATmega32 memiliki bagian-bagian sebagai berikut:

1. Saluran I/O sebanyak 32 buah, yaitu Port A, Port B, Port C, dan Port D
2. ADC 10 bit sebanyak 8 saluran
3. Tiga buah Timer/Counter dengan kemampuan perbandingan
4. CPU yang terdiri dari 32 buah register serba guna
5. Watchdog Timer dengan osilator internal
6. SRAM sebesar 2K byte
7. Memori Flash sebesar 32K byte dengan kemampuan *Read While Write*

8. Unit interupsi internal dan eksternal
9. Port antarmuka SPI
10. EEPROM sebesar 1024 byte yang dapat diprogram saat beroperasi
11. Antarmuka komparator analog
12. Port USART untuk komunikasi serial

Masing-masing kaki mikrokontroler ATmega32 memiliki fungsi tersendiri. Dengan mengetahui fungsi masing-masing kaki atau pin mikrokontroler ini, perancangan aplikasi menggunakan ATmega32 akan semakin mudah. Mikrokontroler ini memiliki 40 kaki, susunan masing-masing kaki dapat dilihat dalam Gambar 2.20.



Gambar 2.20 Konfigurasi Pin ATmega32

Sumber: Datasheet ATmega32

Fungsi kaki-kaki ATmega32 adalah sebagai berikut:

- VCC, merupakan saluran untuk tegangan masukan catu daya digital sebesar 5volt DC
- Ground, merupakan saluran Ground
- Port A (PA7...PA0), Port A merupakan input analog untuk ADC, akan tetapi jika ADC tidak digunakan maka Port A hanya merupakan 8 bit saluran I/O dua arah
- Port B (PB7...PB0), Port B adalah 8 bit saluran I/O dua arah dengan resistor pull up internal. Selain itu, masing-masing bit Port A memiliki fungsi tertentu. Port B0 juga



berfungsi sebagai T0 (*timer/counter0 external counter input*) dan XCK (*USART external clock input/output*). Port B1 memiliki fungsi lain sebagai T1 (*timer/counter1 external counter input*). Port B2 berfungsi sebagai AIN0 (*Analog comparator positive input*) dan sebagai INT2 (*External interrupt 2 input*). Port B3 juga berfungsi sebagai AIN1 (*Analog comparator negative input*) dan OC0 (*Timer/counter0 output compare match output*). Port B4 berfungsi sebagai SS (*SPI slave select input*). Port B5 juga merupakan pin MOSI (*SPI bus Master Output/Slave Input*). Port B6 merupakan pin MISO (*SPI bus Master Input/Slave Output*). Dan Port B7 berfungsi sebagai SCK (*SPI bus serial clock*)

- Port C (PC7...PC0), Port C adalah 8 bit saluran I/O dua arah dengan resistor pull up internal. Masing-masing bit Port C juga memiliki fungsi tertentu. Port C0 memiliki fungsi sebagai SCL (*Two-Wire serial bus clock line*). Port C1 berfungsi sebagai SDA (*Two-Wire serial bus data input/output line*). Port C2 berfungsi sebagai TCK. Port C3 berfungsi sebagai TMS. Fungsi lain Port C4 adalah sebagai TDO. Port C5 juga berfungsi sebagai TDI. Port C6 juga berfungsi sebagai TOSC1 (*Timer oscillator pin1*). Dan Port C7 berfungsi sebagai TOSC2 (*Timer oscillator pin2*)
- Port D (PD7...PD0), Port D adalah 8 bit saluran I/O dua arah dengan resistor pull up internal. Masing-masing bit Port D juga memiliki fungsi tertentu. Port D0 memiliki fungsi lain sebagai RXD (*USART input pin*). Port D1 juga berfungsi sebagai TXD (*USART output pin*). Port D2 berfungsi sebagai INT0 (*External interrupt 0 input*). Port D3 sebagai INT1 (*External interrupt 1 input*). Port D4 berfungsi sebagai OC1B (*Timer/counter1 output compare B match output*). Port D5 berfungsi sebagai OC1A (*Timer/counter1 output compare A match output*). Port D6 berfungsi sebagai ICP1 (*Timer/counter1 input capture pin*). Dan Port D7 juga berfungsi sebagai OC2 (*Timer/counter2 output compare match output*)
- RESET, merupakan pin input RESET. Pemberian logika low pada pin ini lebih lama daripada panjang pulsa minimum akan membangkitkan reset pada mikrokontroler. Lebar pulsa minimum untuk pin reset ini adalah 1,5µs
- XTAL1, merupakan input untuk *inverting Oscillator amplifier* dan input untuk rangkaian yang mengoperasikan *internal clock*
- XTAL2, merupakan output dari *inverting Oscillator amplifier*
- AVCC, merupakan sumber catu tegangan untuk Port A dan ADC. AVCC harus dihubungkan secara eksternal pada VCC walaupun ADC tidak digunakan. Jika ADC digunakan, harus dihubungkan pada VCC melalui *low pass filter*

- AREF, merupakan pin referensi analog untuk ADC.

## 2.8 Analog To Digital Converter dalam Mikrokontroler ATmega32

Mikrokontroler ATmega32 memiliki fitur ADC yang dihubungkan dengan 8 buah channel multiplexer analog yang semuanya terdapat pada pin-pin dari portA. Resolusi yang dapat digunakan adalah 10 bit atau 8 bit. Tegangan referensi ADC yang merupakan nilai maksimum dapat diambil dari 3 buah sumber antara lain dari pin AREF, pin AVcc, dan tegangan referensi internal 2,56 V.

Register-register yang digunakan dalam ADC antara lain:

- ADC Multiplexer Selection Register (ADMUX)

Register ini digunakan untuk mengatur tegangan referensi dan untuk memilih channel ADC yang digunakan. Konfigurasi bit-bit register ini ditunjukkan dalam Gambar 2.21.

Bit	7	6	5	4	3	2	1	0	
	<b>REFS1 REFS0 ADLAR MUX4 MUX3 MUX2 MUX1 MUX0</b>								<b>ADMUX</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.21 Konfigurasi Bit ADMUX

- Bit 7 dan 6 (REFS1 dan REFS0): digunakan sebagai bit-bit pemilihan tegangan referensi. Pengaturan bit-bit ini ditunjukkan dalam Tabel 2.1.

Tabel 2.1 Pemilihan Tegangan Referensi ADC

Sumber: Datasheet ATmega32

REFS1	REFS0	Sumber Tegangan Referensi
0	0	pin AREF dengan Vref internal dimatikan
0	1	pin AVCC dengan kapasitor eksternal pada pin AREF
1	0	tidak digunakan
1	1	tegangan referensi internal 2,56V

- Bit 5 (ADLAR): digunakan untuk mengatur hasil konversi ADC yang tersimpan dalam register data ADC (ADCL dan ADCH). Penulisan 1 pada bit ini akan mengakibatkan 8bit MSB data hasil konversi ADC disimpan dalam register ADCH dan 2bit LSBnya disimpan dalam register ADCL pada bit 7 dan bit 6. Sedangkan penulisan 0 pada bit ini mengakibatkan 2bit MSB data hasil konversi



disimpan dalam register ADCH pada bit 1 dan bit 0 dan 8bit LSBnya disimpan dalam register ADCL

- Bit 4...0 (MUX4...0): digunakan untuk mengatur pemilihan channel yang digunakan dalam ADC. Pengaturan bit-bit ini ditunjukkan dalam Tabel 2.2

Tabel 2.2 *Pemilihan channel ADC*

Sumber: Datasheet ATmega32

MUX4...0	channel ADC
00000	ADC0
00001	ADC1
00010	ADC2
00011	ADC3
00100	ADC4
00101	ADC5
00110	ADC6
00111	ADC7

➤ **ADC Control and Status Register A (ADCSRA)**

Register ini digunakan untuk mengaktifkan ADC, mengisyaratkan untuk memulai konversi, mengaktifkan program interupsi ADC, dan untuk mengatur clock masukan dari ADC. Konfigurasi bit-bit register ini ditunjukkan dalam Gambar 2.22.

Bit	7	6	5	4	3	2	1	0	
	<b>ADEN</b>	<b>ADSC</b>	<b>ADATE</b>	<b>ADIF</b>	<b>ADIE</b>	<b>ADPS2</b>	<b>ADPS1</b>	<b>ADPS0</b>	<b>ADCSRA</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Gambar 2.22 *Konfigurasi Bit ADCSRA*

- Bit 7 (ADEN): jika diisi dengan nilai 1 maka ADC telah diaktifkan dan jika diisi 0 maka ADC dimatikan
- Bit 6 (ADSC): digunakan untuk mengisyaratkan konversi ADC dimulai dengan menuliskan 1 pada bit ini
- Bit 5 (ADATE): penulisan 1 pada bit ini mengaktifkan picu otomatis ADC yang sumber pemicuannya dipilih oleh SFIOR
- Bit 4 (ADIF): bit ini akan bernilai 1 jika konversi telah usai dan isi register data ADC telah diperbaharui





- Bit 3 (ADIE): bit ini digunakan untuk mengaktifkan program interupsi ADC, jika bit ini bernilai 1 dan konversi ADC telah usai maka program interupsi ADC akan dipanggil
- Bit 2..0 (ADPS2..0): digunakan untuk mengatur clock masukan ADC dengan mengatur faktor pembagi. Pengaturan bit-bit ini dapat dilihat dalam Tabel 2.3.

Tabel 2.3 Pengaturan Faktor Pembagi Clock Masukan ADC

Sumber: Datasheet ATmega32

ADPS2	ADPS1	ADPS0	faktor pembagi
0	0	0	2
0	0	1	2
0	1	0	4
0	1	1	8
1	0	0	16
1	0	1	32
1	1	0	64
1	1	1	128

➤ ADC Data Register

Register ini digunakan untuk menyimpan data hasil konversi ADC. Terdiri dari dua register, yaitu ADCH dan ADCL. ADCH digunakan untuk menyimpan data MSB, sedangkan ADCL untuk menyimpan data LSBnya. Penggunaan register ini diatur oleh bit ADLAR dalam register ADMUX.

➤ Special FunctionIO Register (SFIOR)

Register ini digunakan untuk memilih sumber pemicuan otomatis pada ADC jika bit ADATE dalam register ADCSRA diisi dengan 1. Pemilihan sumber pemicuan ini menggunakan kombinasi bit 7, 6, dan 5, sedangkan bit lainnya tidak digunakan. Konfigurasi bit-bit register ini ditunjukkan dalam Gambar 2.22 dan pemilihan sumber pemicuan ditunjukkan dalam Tabel 2.4.

Bit	7	6	5	4	3	2	1	0	
	ADTS2	ADTS1	ADTS0	-	ACME	PUD	PSR2	PSR10	SFIOR
Read/Write	R/W	R/W	R/W	R	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

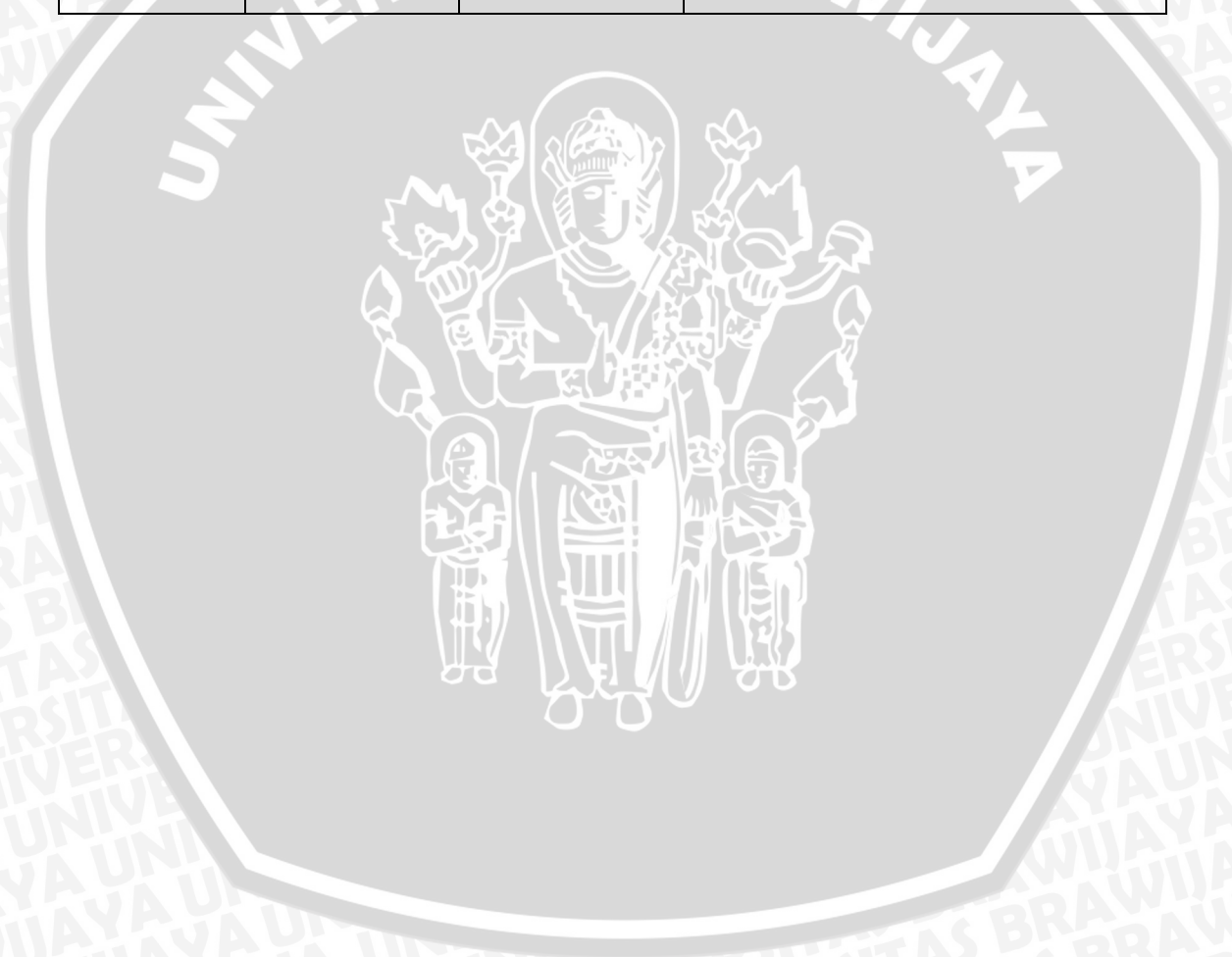
Gambar 2.23 Konfigurasi Bit SFIOR



Tabel 2.4 *Pemilihan Picu ADC otomatis*

Sumber: Datasheet ATmega32

ADTS2	ADTS1	ADTS0	sumber picu
0	0	0	Free Running Mode
0	0	1	Analog Comparator
0	1	0	External Interrupt Request 0
0	1	1	Timer/Counter0 Compare Match
1	0	0	Timer/Counter0 Overflow
1	0	1	Timer/Counter1 Compare Match B
1	1	0	Timer/Counter1 Overflow
1	1	1	Timer/Counter1 Capture Event



## BAB III

### METODE PENELITIAN

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiannya agar dapat menampilkan unjuk kerja sesuai yang direncanakan dengan mengacu pada rumusan masalah. Pemilihan komponen dilakukan berdasarkan perencanaan dan disesuaikan dengan komponen yang mudah dijumpai di pasaran. Untuk merealisasikan alat ini, langkah-langkah yang diambil adalah sebagai berikut:

#### 3.1 Perencanaan dan Pembuatan Alat

Sebelum melakukan perencanaan dan perealisasiannya, maka ditentukan spesifikasi alat yang akan dibuat. Spesifikasi yang akan direalisasikan adalah sebagai berikut:

- 1) Mikrokontroler yang digunakan sebagai pengolah data adalah ATmega32
- 2) Algoritma filter digital yang digunakan adalah filter IIR *Butterworth* orde empat
- 3) Frekuensi sampling ADC yang digunakan adalah 20 kHz
- 4) Filter *Lowpass* yang digunakan memiliki frekuensi cutoff 4 kHz
- 5) *Microphone* yang digunakan adalah *dynamic microphone*
- 6) Rangkaian pengkondisi sinyal yang digunakan adalah rangkaian penguat op amp non pembalik.

#### 3.2 Pengujian

Untuk mengetahui kemampuan kerja rangkaian serta kesesuaian dengan perencanaan, maka dilakukan pengujian rangkaian. Pengujian rangkaian dilakukan tiap blok. Selanjutnya dilakukan penggabungan semua blok dan dilakukan pengujian secara keseluruhan. Macam-macam pengujian yang dilakukan adalah sebagai berikut :

- 1) Pembuatan perangkat keras sistem dengan menggunakan komponen elektronika yang telah direncanakan
- 2) Pengujian keseluruhan sistem

Pengujian keseluruhan sistem dilakukan dengan menggabungkan blok perangkat keras dan mengoperasikan sistem kemudian dapat diketahui apakah alat ini bekerja sesuai dengan spesifikasi yang diharapkan.



Pengujian dilakukan pada masing-masing blok, baru kemudian dilakukan pengujian keseluruhan sistem. Secara garis besar pengujian perblok adalah sebagai berikut:

### 3.2.1 Pengujian Microphone

Tujuan pengujian ini adalah untuk mengetahui apakah *microphone* dapat menerima sinyal suara yang ditangkap dengan baik. Pengujian dilakukan dengan memberikan sinyal suara ke *microphone* dan menampilkan sinyal keluaran *microphone* pada osiloskop. Dari pengujian akan diamati apakah *microphone* dapat bekerja dengan baik.

### 3.2.2 Pengujian Filter dan Pengkondisi Sinyal

Pengujian filter dan pengkondisi sinyal bertujuan untuk mengetahui apakah filter dan pengkondisi sinyal bekerja sesuai dengan perancangan. Pengujian filter dilakukan dengan memberikan sinyal dengan variasi frekuensi tertentu pada input filter dan menampilkan hasil keluarannya pada osiloskop. Pengujian pengkondisi sinyal dilakukan dengan memberikan sinyal dengan frekuensi tertentu pada input pengkondisi sinyal dan menampilkan keluarannya pada osiloskop. Dari hasil pengujian akan diamati apakah filter dan pengkondisi sinyal telah sesuai dengan perancangan.

### 3.2.3 Pengujian Mikrokontroler dan ADC

Pengujian ADC dilakukan untuk mengetahui apakah ADC dapat mengkonversi sinyal analog menjadi sinyal diskrit. Pengujian dilakukan dengan menghubungkan *ADC channel* pada mikrokontroler dengan sinyal input analog, kemudian diamati hasil konversinya apakah sudah sesuai dengan teori yang ada.

### 3.2.4 Pengujian Mikrokontroler

Pengujian mikrokontroler dilakukan untuk mengetahui apakah mikrokontroler dapat berfungsi dengan baik atau tidak. Pengujian dilakukan dengan memberikan program sederhana pada mikrokontroler. Melalui minimum sistem akan diamati apakah mikrokontroler bekerja sebagaimana mestinya sesuai dengan program yang dituliskan.

## BAB IV

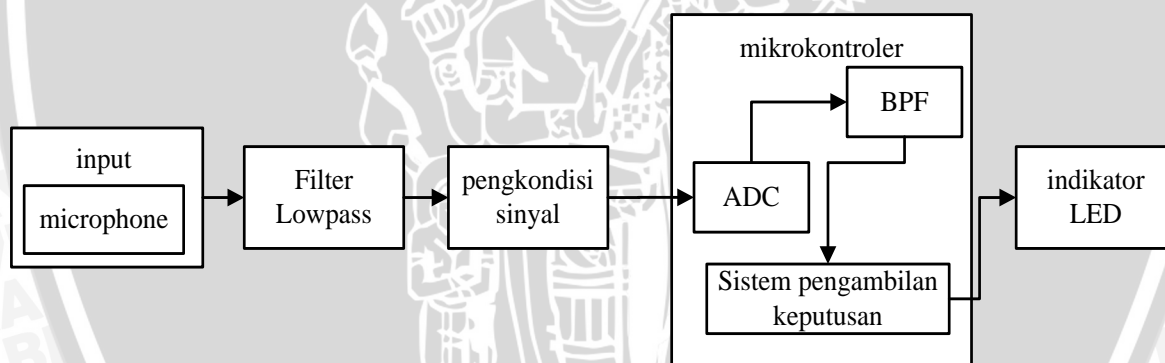
### PERENCANAAN DAN PEMBUATAN ALAT

Perencanaan alat dilakukan secara bertahap yaitu dilakukan blok demi blok. Hal ini dilakukan untuk mempermudah penganalisaan di setiap bloknya maupun secara keseluruhan. Perancangan alat ini terdiri atas cara kerja alat yang memuat blok diagram sistem, perancangan perangkat keras, simulasi Matlab yang bertujuan untuk merancang filter digital yang digunakan, dan perancangan perangkat lunak.

#### 4.1 Cara Kerja Alat

Perancangan alat secara garis besar digambarkan dengan diagram blok dalam Gambar 4.1. Alat terdiri atas beberapa bagian, yaitu input, filter analog *lowpass*, pengkondisi sinyal, mikrokontroler ATmega32, dan output.

Pada sistem ini menggunakan *microphone* sebagai sensor. Sensor ini digunakan untuk menangkap sinyal suara yang berasal dari musik yang diperdengarkan.



Gambar 4.1 Diagram Blok Sistem

Filter analog yang digunakan dalam perancangan ini adalah filter *lowpass*. Filter ini digunakan untuk memisahkan sinyal informasi dan sinyal noise yang ditangkap oleh *microphone* agar sinyal yang ditangkap adalah benar-benar sinyal yang diinginkan. Selain itu, pemilihan filter *lowpass* bertujuan untuk membatasi frekuensi maksimum sinyal masukan yang akan diproses oleh ADC. Hal ini bertujuan agar frekuensi sinyal yang masuk tidak berharga lebih dari setengah frekuensi sampling. Setelah sinyal masukan dilewatkan, sinyal ini akan dikondisikan oleh rangkaian pengkondisi sinyal untuk menyesuaikan dengan karakteristik yang dimiliki mikrokontroler. Di samping itu,

rangkaian pengkondisi sinyal juga terdiri atas rangkaian penguat yang berfungsi untuk menguatkan sinyal. Hal ini dilakukan agar sinyal dapat dibaca oleh mikrokontroler.





Proses selanjutnya terjadi dalam mikrokontroler, yang merupakan proses utama dalam sistem ini. Sebelum diproses oleh filter digital, sinyal akan diubah terlebih dahulu menjadi sinyal digital. Proses ini dilakukan oleh ADC yang merupakan fasilitas dalam mikrokontroler. Secara matematis sinyal sebelum diproses ADC merupakan persamaan sinyal waktu kontinyu  $V = x(t)$ . Setelah diproses oleh ADC, sinyal tersebut ditetapkan sebagai titik-titik sekuensial yang mewakili sinyal analog masukan dan akan disimpan di memori mikrokontroler. Titik-titik tersebut kemudian dinotasikan dengan  $x_0, x_1, x_2, x_3, \dots, x_n$ .

Proses selanjutnya adalah sinyal akan diproses oleh BPF (*Band Pass Filter*). Filter yang digunakan dalam perancangan ini adalah filter IIR (*Infinite Impulse Response*) *Butterworth* orde empat yang dirancang dengan menggunakan Matlab. Penerapan orde filter yang relatif rendah ini bertujuan untuk kemudahan dalam pengimplementasian dalam sistem digital *real-time* yang menuntut kecepatan proses. Filter ini digunakan untuk memisahkan sinyal yang berasal dari suara musik dengan *noise* yang berasal dari suara manusia. Selanjutnya jika sinyal telah diolah oleh BPF, mikrokontroler akan menerapkan satu sistem keputusan yang memastikan bahwa sinyal ini adalah benar-benar sinyal informasi yang diinginkan. Jika ini sesuai, mikrokontroler akan memberikan indikator berupa cahaya LED.

## 4.2 Perancangan Perangkat Keras

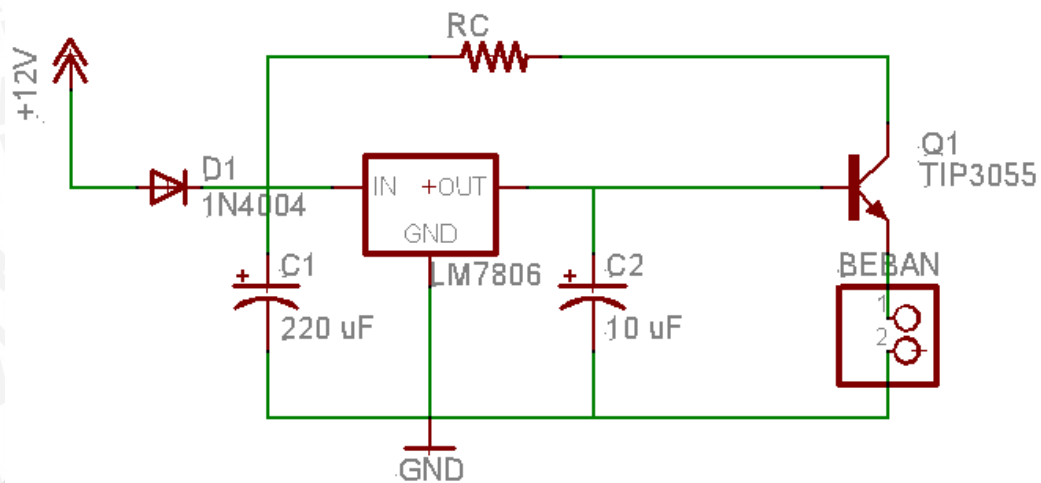
Perancangan perangkat keras terdiri dari beberapa bagian, yaitu rangkaian catu daya sistem, rangkaian *microphone*, rangkaian filter *lowpass*, rangkaian pengkondisi sinyal, dan rangkaian mikrokontroler.

### 4.2.1 Rangkaian Catu Daya Sistem

Sistem pendeteksi musik ini membutuhkan catu daya sebesar 5 volt pada sistem elektrik dan mikrokontrolernya. Catu daya sebesar 5 volt ini sudah memenuhi tegangan kerja mikrokontroler ATmega32 yang berkisar antara 4,5 volt sampai 5,5 volt berdasarkan datasheet.

Rangkaian catu daya ini menggunakan *Fixed Output Regulator* sesuai pada datasheet LM78XX. Regulator yang digunakan adalah jenis LM7806 yang memiliki tegangan keluaran minimal 5,7 volt dan maksimal 6,3 volt berdasarkan datasheetnya dengan tegangan masukan antara 8 volt sampai 21 volt dan arus keluaran regulator sebesar antara 5 mA sampai 1 A. Karena tegangan catu yang dibutuhkan adalah 5 volt

dan tegangan keluaran regulator adalah 6 volt, maka ditambahkan transistor TIP3055 pada rangkaian catu daya ini. Skema rangkaian catu daya sistem ditunjukkan dalam Gambar 4.2.



Gambar 4.2 Rangkaian Catu Daya Sistem

Rangkaian catu daya sistem dirancang menggunakan tegangan sumber sebesar 12V. Berdasarkan datasheet transistor TIP3055 diketahui bahwa  $V_{BE}$  sebesar 0,7V,  $V_{CE\ sat}$  maksimal sebesar 1,1V. Nilai tegangan beban dapat diperoleh menggunakan persamaan loop I berikut ini:

$$V_{BB} = V_{BE} + V_{Beban} \quad (4.1)$$

$$6V = 0,7V + V_{Beban}$$

$$V_{Beban} = 6V - 0,7V$$

$$V_{Beban} = 5,3V$$

Tegangan 5,3V ini masih dapat diaplikasikan pada mikrokontroler berdasarkan datasheet ATmega32.

Besar arus beban dalam perancangan diharapkan sebesar 0,5 A. Penentuan nilai  $R_C$  dapat dihitung menggunakan persamaan loop II berikut ini:

$$V_{CC} = I_C \cdot R_C + V_{CE\ sat\ maks} + V_{Beban} \quad (4.2)$$

$$12V = 0,5\ A \cdot R_C + 1,1V + 5,3V$$

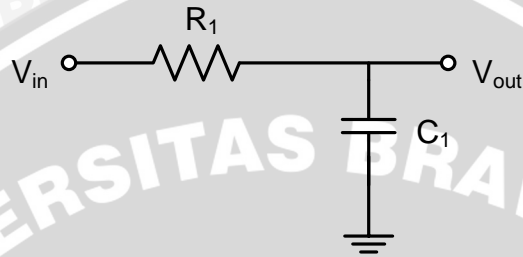
$$5,6V = R_C \cdot 0,5\ A$$

$$R_C = \frac{5,6V}{0,5\ A}$$

$$R_C = 11,2\ \Omega$$

#### 4.2.2 Rangkaian Filter *Lowpass*

Frekuensi sinyal informasi tertinggi yang diterima oleh *microphone* adalah sebesar 2,5 kHz. Oleh karena itu, filter *lowpass* dirancang memiliki frekuensi *cutoff* sebesar 4 kHz. Ini untuk memastikan bahwa sinyal dengan frekuensi tertinggi yang diinginkan masih berada di daerah *passband* dan sinyal yang memiliki frekuensi mendekati nilai setengah dari frekuensi sampling ADC benar-benar berada di daerah *stopband*. Rangkaian filter *lowpass* ditunjukkan dalam Gambar 4.3.



Gambar 4.3 Rangkaian Filter *Lowpass*

Nilai  $V_{out}$  dapat diperoleh dengan menggunakan persamaan sebagai berikut

$$V_{out} = \frac{\frac{1}{sC_1}}{\frac{1}{sC_1} + R_1} \cdot V_{in} \quad (4.3)$$

$$V_{out} = \frac{1}{1 + sR_1C_1} \cdot V_{in} \quad (4.4)$$

$$V_{out} = \frac{1}{1 + j\omega R_1C_1} \cdot V_{in} \quad (4.5)$$

Fungsi alih rangkaian filter *lowpass* dapat diperoleh dengan menggunakan persamaan

$$H(j\omega) = \frac{V_{out}}{V_{in}} \quad (4.6)$$

$$H(j\omega) = \frac{1}{1 + j\omega R_1C_1} \quad (4.7)$$

Besar nilai fungsi alih ini merupakan penurunan penguatan filter dalam respons frekuensi filter *lowpass* itu sendiri. Nilai tersebut dapat diperoleh menggunakan persamaan sebagai berikut



$$|H(j\omega)| = \frac{1}{\sqrt{1+(\omega R_1 C_1)^2}} \quad (4.8)$$

Frekuensi *cutoff* adalah frekuensi di mana penguatan filter turun sebesar  $1/\sqrt{2}$ .

Maka nilai frekuensi *cutoff* dapat dihitung menggunakan Persamaan 4.9.

$$|H(j\omega_c)| = \frac{1}{\sqrt{2}} \quad (4.9)$$

$$\frac{1}{\sqrt{2}} = \frac{1}{\sqrt{1+(\omega_c R_1 C_1)^2}} \quad (4.10)$$

$$\frac{1}{\sqrt{2}} = \frac{1}{\sqrt{1+(\omega_c R_1 C_1)^2}} \quad (4.11)$$

$$\omega_c R_1 C_1 = 1 \quad (4.12)$$

$$2\pi f_c R_1 C_1 = 1 \quad (4.13)$$

Sehingga nilai frekuensi *cutoff* filter *lowpass* dapat diketahui dengan menggunakan Persamaan 4.14.

$$f_c = \frac{1}{2\pi R_1 C_1} \quad (4.14)$$

maka nilai-nilai variabel dalam rangkaian filter *lowpass* dengan frekuensi *cutoff* sebesar 4000 Hz adalah sebagai berikut:

$$4000 \text{ Hz} = \frac{1}{2\pi R_1 C_1}$$

$$R_1 C_1 = \frac{1}{2\pi 4000}$$

$$R_1 C_1 = 3,97887 \cdot 10^{-5}$$

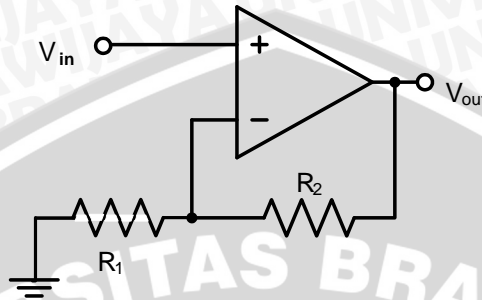
jika  $C_1 = 10 \text{ nF}$  maka  $R_1 = 3978,874 \ \Omega$ .

### 4.2.3 Rangkaian Pengkondisi Sinyal

Untuk mendapatkan sinyal yang besarnya sesuai dengan yang diinginkan maka dibutuhkan rangkaian pengkondisi sinyal. Rangkaian pengkondisi sinyal ini merupakan penguat opamp. Rangkaian penguat ini digunakan untuk menguatkan sinyal yang

ditangkap oleh *microphone* yang relatif kecil, sehingga dapat diolah oleh mikrokontroler.

Rangkaian penguat yang digunakan adalah penguat operasional non pembalik (*non inverting*). Rangkaian penguat ini dapat dilihat dalam Gambar 4.4.



Gambar 4.4 Rangkaian Penguat Non Inverting

Besar penguatan  $A_v$  dari rangkaian tersebut dapat diperoleh dengan menggunakan persamaan berikut:

$$\frac{V_{out} - V_-}{R_2} = \frac{V_- - 0}{R_1} \quad (4.15)$$

yang mana  $V_- = V_+ = V_{in}$  sehingga

$$\frac{V_{out} - V_{in}}{R_2} = \frac{V_{in}}{R_1}$$

$$V_{out} = V_{in} + \frac{R_2}{R_1} V_{in}$$

$$V_{out} = \left(1 + \frac{R_2}{R_1}\right) V_{in}$$

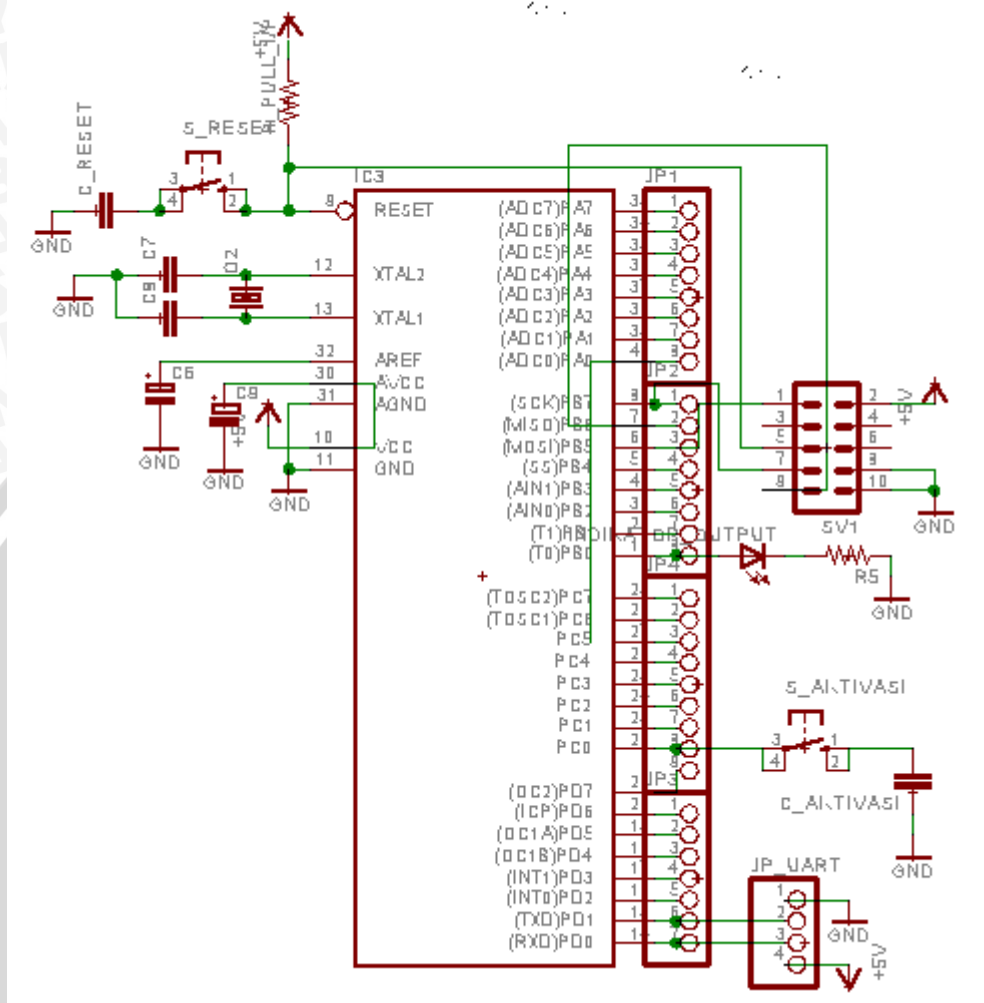
maka diperoleh penguatannya adalah sebesar  $A_v = 1 + \frac{R_2}{R_1}$

Untuk mendapatkan rangkaian penguat yang memiliki penguatan sebesar 100 kali maka nilai  $R_1$  dan  $R_2$  yang harus ditentukan. Jika  $R_2$  yang digunakan bernilai 100 k $\Omega$ , maka nilai  $R_1$  adalah 1 k $\Omega$ . Dalam realisasinya, rangkaian penguat ini menggunakan  $R_2$  sebesar 100 k $\Omega$  dan  $R_1$  digantikan dengan resistor variabel agar penguatan dapat dengan mudah diatur dan disesuaikan.

#### 4.2.4 Rangkaian Mikrokontroler

Pada perancangan sistem pendeteksi suara musik ini digunakan mikrokontroler ATmega32 sebagai pengolah utama dalam menentukan apakah sinyal masukan yang

ditangkap oleh *microphone* adalah sinyal yang berasal dari suara musik atau tidak. Konfigurasi pin mikrokontroler ATmega32 ditunjukkan dalam Gambar 4.5.



Gambar 4.5 Minimum Sistem Mikrokontroler

Mikrokontroler ATmega32 memiliki kaki sebanyak 40 pin, 32 diantaranya dapat diprogram menjadi masukan atau keluaran. Berikut ini adalah pin-pin yang digunakan dalam perancangan ini:

PA0: digunakan sebagai masukan ADC (pin ADC0)

AVCC: digunakan sebagai tegangan referensi ADC

PB0: digunakan sebagai port output sistem yang terhubung dengan indikator led

PC0: digunakan sebagai input tombol push button

PB5: digunakan sebagai antarmuka dengan *downloader* (MOSI-SPI)

PB6: digunakan sebagai antarmuka dengan *downloader* (MISO-SPI)

PB7: digunakan sebagai antarmuka dengan *downloader* (SCK-SPI)

RESET: digunakan sebagai antarmuka dengan *downloader* dan sebagai pin reset

PD0: digunakan untuk komunikasi UART (pin RXD)



PD1: digunakan untuk komunikasi UART (pin TXD)

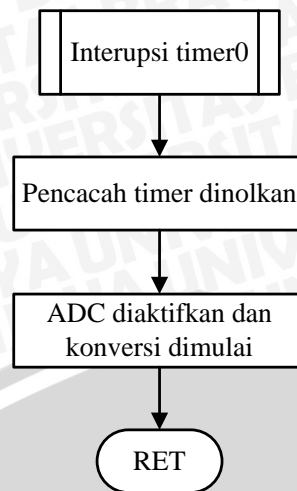
### 4.3 Perancangan Perangkat Lunak

#### 4.3.1 Analog to Digital Converter

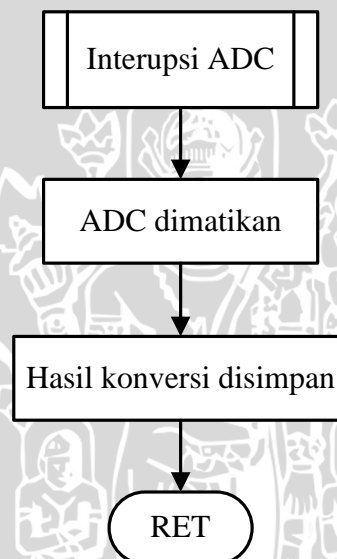
Untuk mengubah sinyal analog yang ditangkap oleh bagian input sistem menjadi sinyal digital yang dapat diolah oleh mikrokontroler, maka digunakan fitur ADC yang ada dalam mikrokontroler ATmega32. Kaki ADC yang dapat dipilih berjumlah delapan channel, dan channel yang digunakan dalam sistem ini adalah kaki ADC0. Spesifikasi yang diperlukan dalam ADC antara lain adalah resolusi dan tegangan referensi. Resolusi yang digunakan adalah 10 bit dengan tegangan referensi tegangan  $V_{CC}$ . Hal ini menyebabkan perubahan sinyal analog terkecil yang masih mampu terdeteksi oleh ADC adalah sebesar 4,88 mV. Nilai ini diperoleh berdasarkan rumus hasil konversi ADC sebagai berikut:

$$out = \frac{V_{in} \cdot (2^{10} - 1)}{V_{ref}} \quad (4.16)$$

Frekuensi sampling ADC yang digunakan sebesar 20 kHz. Pengaturan frekuensi sampling menggunakan program interupsi timer0 yang dieksekusi tiap satu periode sampling yaitu 50  $\mu$ s. Program interupsi ini mengaktifkan dan memulai konversi ADC. Penyimpanan hasil konversi ADC dilakukan dalam program interupsi ADC yang dieksekusi ketika proses konversi telah usai. Hal ini dilakukan supaya penyimpanan data hasil konversi benar-benar dilakukan setelah konversi benar-benar telah selesai. Dalam program interupsi, ADC dimatikan terlebih dahulu baru kemudian data hasil konversi disimpan agar ADC tidak melakukan konversi lagi sebelum periode sampling. Diagram alir program interupsi timer0 dan program interupsi ADC ditunjukkan dalam Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Diagram Alir Program Interupsi Timer0



Gambar 4.7 Diagram Alir Program Interupsi ADC

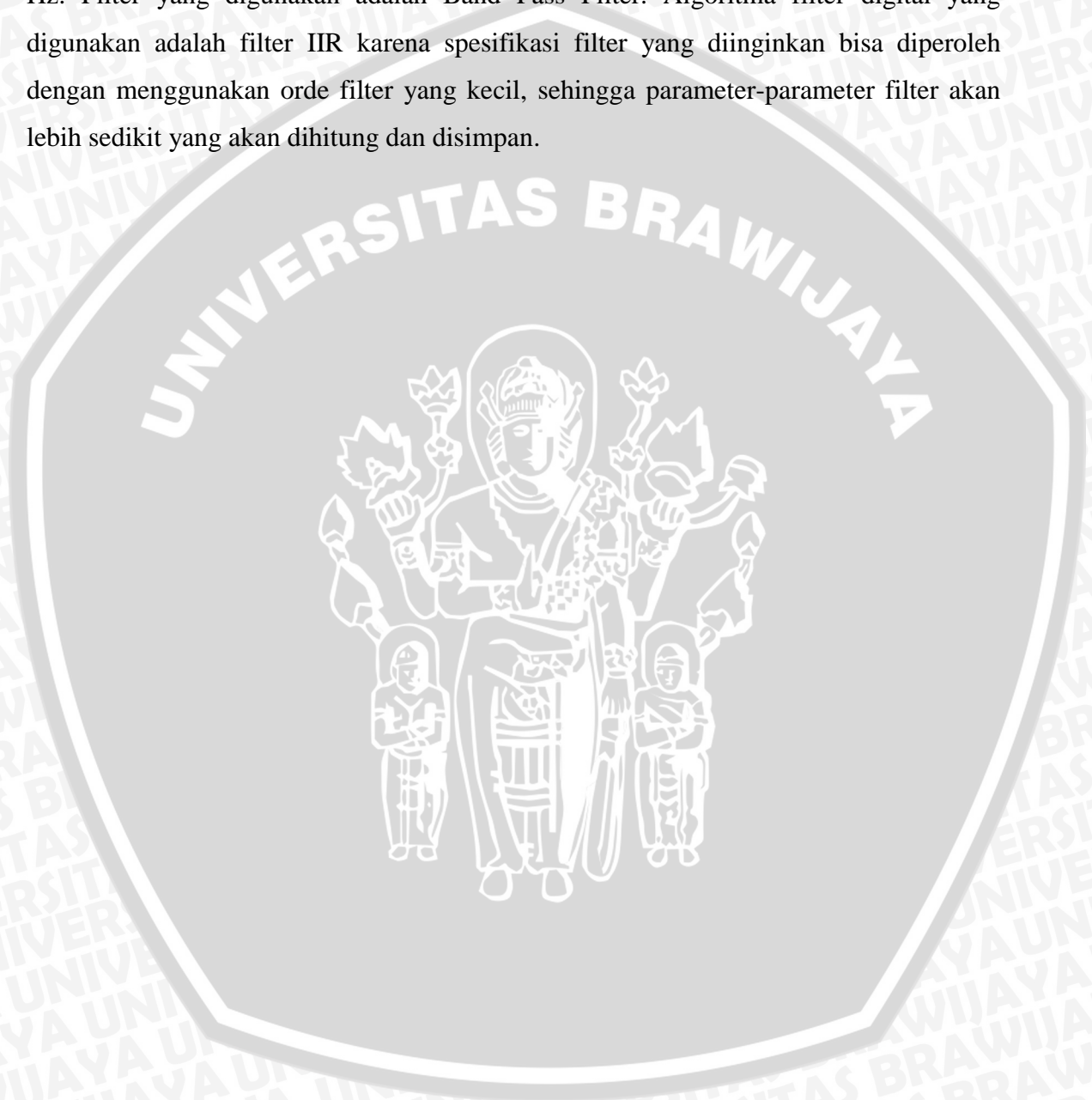
#### 4.3.2 Algoritma Pendeteksi Suara Musik

Berdasarkan Gambar 4.1, dapat diketahui bahwa proses pengolahan data secara digital yang terjadi dalam mikrokontroler ATmega32 antara lain adalah ADC, BPF, dan pengambilan keputusan. Setelah sinyal masukan diolah oleh ADC menjadi sinyal diskrit, maka data hasil konversinya akan diolah berdasarkan algoritma filter IIR butterworth orde empat. Kemudian diterapkan suatu sistem pengambilan keputusan. Diagram alir program utama ditunjukkan dalam Gambar 4.8.

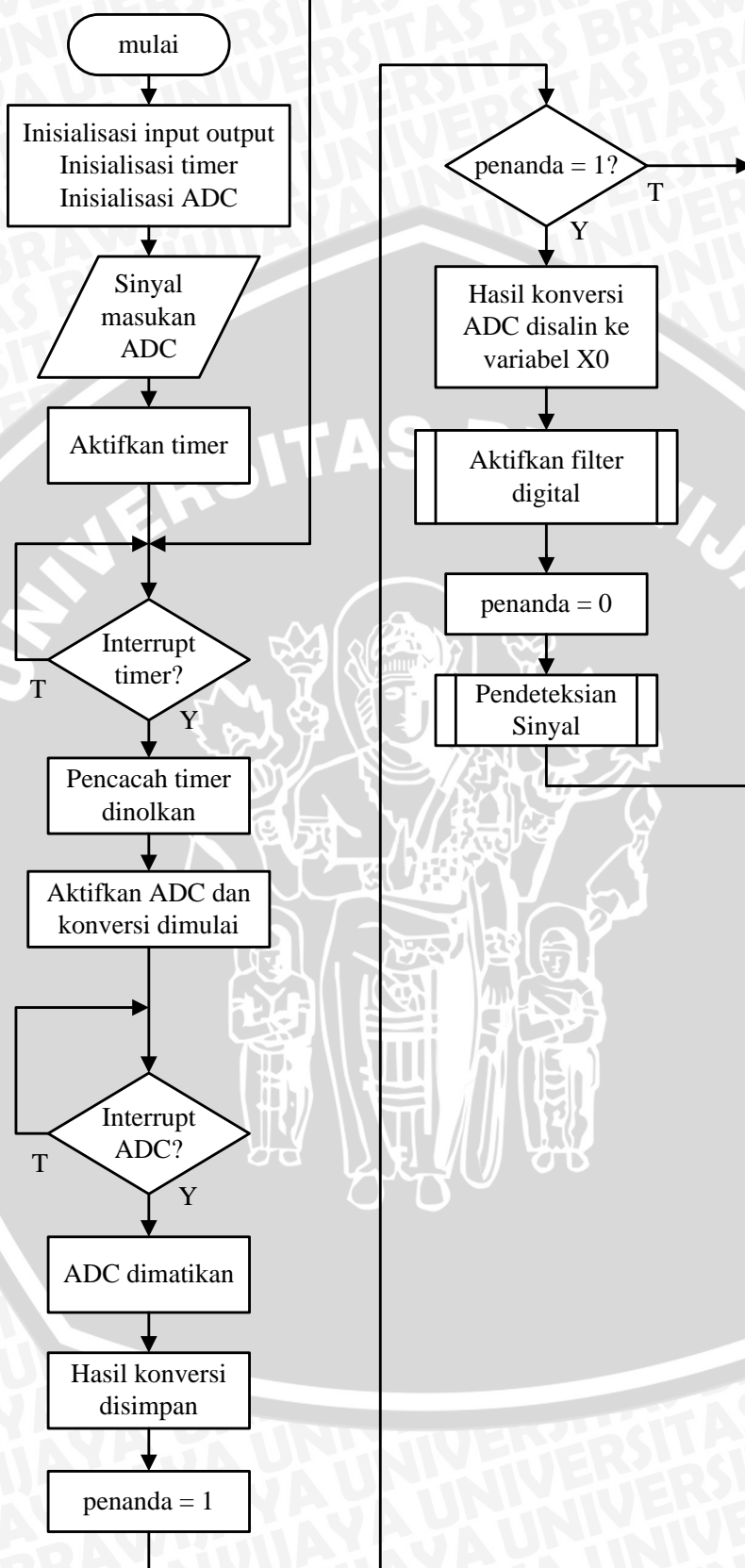
Sinyal yang diharapkan berupa sinyal yang berasal dari suara musik yang diperdengarkan. Sinyal ini memiliki *range* frekuensi antara 300 Hz sampai 2500 Hz. Sinyal ini terinterferensi oleh *noise* yang berasal dari suara manusia. Secara teoritis,

suara manusia memiliki range frekuensi berkisar antara 100 Hz sampai 4000 Hz. Akan tetapi frekuensi suara yang lazim dihasilkan manusia adalah antara 300 Hz sampai 3000 Hz.

Setelah sinyal masukan diolah oleh ADC menjadi sinyal diskrit, langkah selanjutnya adalah meredam sinyal dengan frekuensi di bawah 300 Hz dan di atas 2500 Hz. Filter yang digunakan adalah Band Pass Filter. Algoritma filter digital yang digunakan adalah filter IIR karena spesifikasi filter yang diinginkan bisa diperoleh dengan menggunakan orde filter yang kecil, sehingga parameter-parameter filter akan lebih sedikit yang akan dihitung dan disimpan.







Gambar 4.8 Diagram Alir Program Utama

Filter IIR dibagi menjadi empat tipe, yaitu *Butterworth*, *Chebyshev I*, *Chebyshev II*, dan *Elliptic*. Filter *Butterworth* memiliki respon frekuensi dengan daerah transisi (*transition band*) relatif lebih landai, namun tidak ada *ripple* pada daerah *passband* maupun daerah *stopband*-nya. Respon frekuensi filter *Chebyshev* memiliki daerah transisi yang lebih curam, akan tetapi filter *Chebyshev I* memiliki *ripple* pada daerah *passband* dan *Chebyshev II* memiliki *ripple* pada daerah *stopband*. Sedangkan respon frekuensi filter *Elliptic* juga memiliki daerah transisi yang relatif curam, namun memiliki *ripple* pada daerah *passband* maupun *stopband*-nya. Dalam sistem pendeteksi suara ini, tidak menghendaki adanya *ripple* pada respon frekuensi filter digital yang digunakan. Oleh karena itu algoritma filter digital yang digunakan adalah filter IIR *Butterworth*.

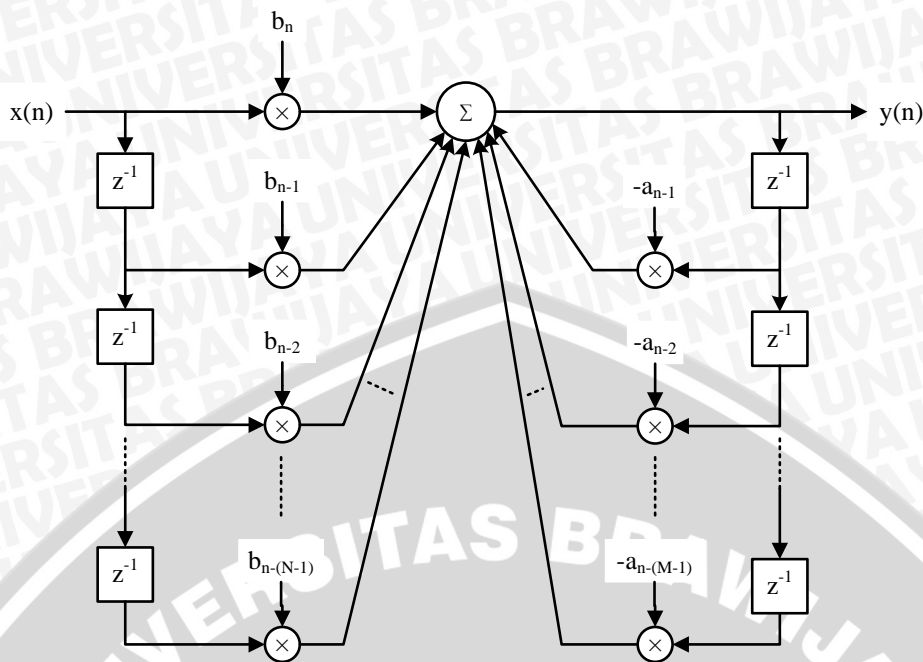
Dengan menggunakan algoritma filter IIR *Butterworth*, dirancang suatu filter *bandpass* dengan frekuensi cutoff  $f_{C1}$  sebesar 300 Hz dan frekuensi cutoff  $f_{C2}$  sebesar 2500 Hz karena sinyal informasi yang berasal dari suara musik memiliki frekuensi pada *range* tersebut. Persamaan matematis yang digunakan dalam algoritma ini adalah sebagai berikut:

$$y[n] = \sum_{k=0}^N b_k \cdot x[n - k] + \sum_{m=1}^M -a_m \cdot y[n - m] \quad (4.17)$$

dalam bentuk diskrit Persamaan 4.17 adalah sebagai berikut:

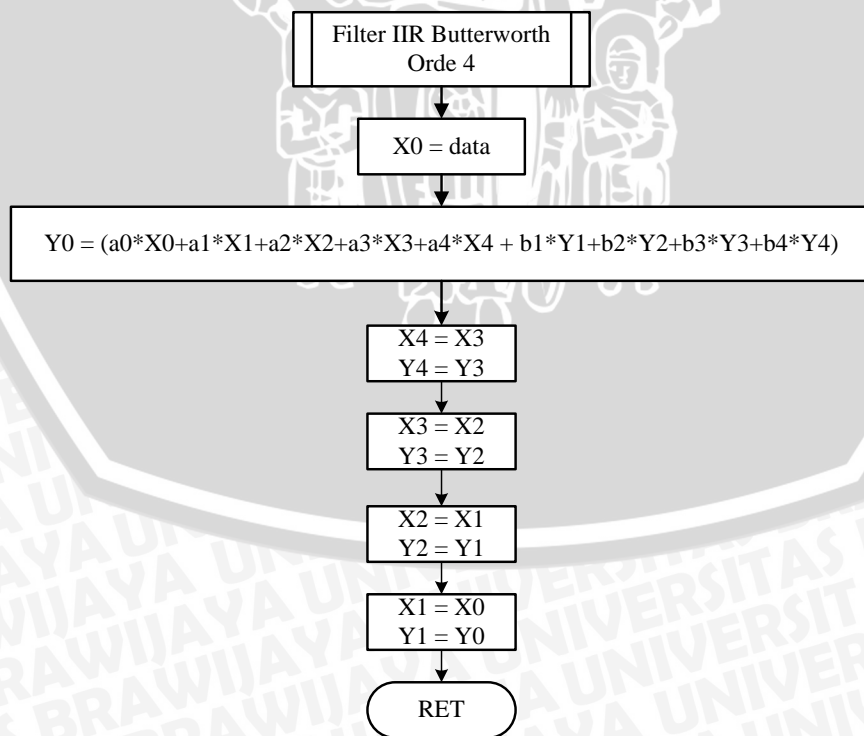
$$Y[z] \left( 1 + \sum_{m=1}^M a_m \cdot z^{-m} \right) = X[z] \sum_{k=0}^N b_k \cdot z^{-k} \quad (4.18)$$

realisasi dari sistem waktu diskrit sesuai dengan Persamaan 4.18 ditunjukkan dalam Gambar 4.9.



Gambar 4.9 Realisasi Persamaan (4.18) dari Sistem Waktu Diskrit

Orde filter digital IIR *butterworth* yang digunakan adalah empat. Pemilihan nilai orde filter ini adalah karena nilai tersebut relatif kecil sehingga mudah untuk diimplementasikan dalam sistem digital *real-time* yang menuntut kecepatan proses. Diagram alir algoritma filter IIR *butterworth* orde empat ditunjukkan dalam Gambar 4.10.



Gambar 4.10 Diagram Alir Filter IIR Butterworth Orde 4



Berdasarkan Persamaan 4.18 maka dapat diperoleh fungsi alih filter adalah sebagai berikut:

$$\frac{Y[z]}{X[z]} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + b_3 \cdot z^{-3} + b_4 \cdot z^{-4}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3} + a_4 \cdot z^{-4}}$$
$$H[z] = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + b_3 \cdot z^{-3} + b_4 \cdot z^{-4}}{a_0 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + a_3 \cdot z^{-3} + a_4 \cdot z^{-4}} \quad (4.19)$$

Koefisien a dan b diperoleh dengan menggunakan simulasi Matlab. Hal ini dilakukan untuk mempermudah perolehan nilai koefisien filter IIR *Butterworth*. Simulasi ini dilakukan dengan menggunakan sintaks:

$$[b,a] = \text{butter}(N, [f_1 f_2] / 0.5 \cdot f_s)$$

yang mana a dan b adalah koefisien pada fungsi transfer  $H(z)$ ,  $f_1$  dan  $f_2$  adalah frekuensi cutoff yang diinginkan pada filter bandpass, N menyatakan setengah dari orde filter, sedangkan  $f_s$  adalah frekuensi sampling yang digunakan.

Filter *Bandpass* yang dirancang memiliki frekuensi cutoff  $f_1$  sebesar 300 Hz dan  $f_2$  sebesar 2500 Hz. Frekuensi sampling  $f_s$  yang digunakan adalah 20 kHz, dan orde filter N yang digunakan adalah orde 4. Dari hasil simulasi Matlab ini diperoleh nilai koefisien a dan b filter sebagai berikut:

$$a_0 = 1$$

$$a_1 = -2,9449$$

$$a_2 = 3,3196$$

$$a_3 = -1,7498$$

$$a_4 = 0,3786$$

$$b_0 = 0,0791$$

$$b_1 = 0$$

$$b_2 = -0,1582$$

$$b_3 = 0$$

$$b_4 = 0,0791$$

sehingga fungsi alih filter  $H(z)$  menjadi

$$H(z) = \frac{0,0791 - 0,1582z^{-2} + 0,0791z^{-4}}{1 - 2,9449z^{-1} + 3,3196z^{-2} - 1,7498z^{-3} + 0,3786z^{-4}}$$

jika fungsi alih filter diubah ke dalam persamaan dengan  $Y(z)$  sebagai variabel keluaran dan  $X(z)$  sebagai variabel masukan, maka diperoleh persamaan sebagai berikut:

$$\frac{Y(z)}{X(z)} = \frac{0,0791 - 0,1582z^{-2} + 0,0791z^{-4}}{1 - 2,9449z^{-1} + 3,3196z^{-2} - 1,7498z^{-3} + 0,3786z^{-4}}$$

$$Y(z)(1 - 2,9449z^{-1} + 3,3196z^{-2} - 1,7498z^{-3} + 0,3786z^{-4}) = X(z)(0,0791 - 0,1582z^{-2} + 0,0791z^{-4})$$

$$Y(z) - 2,9449z^{-1}Y(z) + 3,3196z^{-2}Y(z) - 1,7498z^{-3}Y(z) + 0,3786z^{-4}Y(z) = 0,0791X(z) - 0,1582z^{-2}X(z) + 0,0791z^{-4}X(z)$$

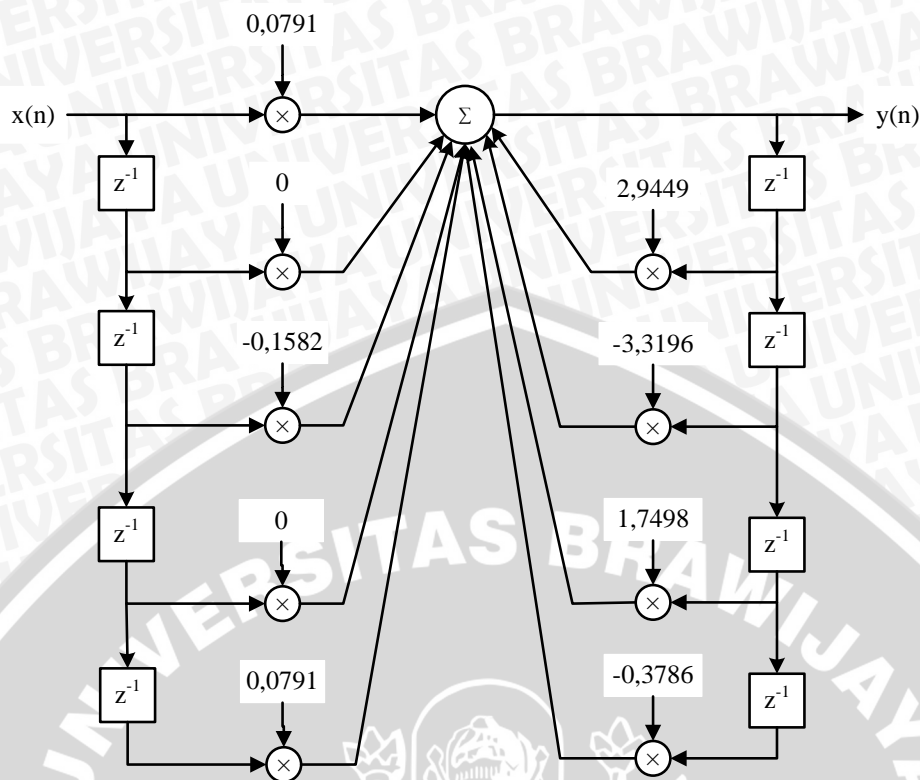
Bentuk invers transformasi Z untuk mendapatkan fungsi  $y(n)$ :

$$y(n) - 2,9449y(n-1) + 3,3196y(n-2) - 1,7498y(n-3) + 0,3786y(n-4) = 0,0791x(n) - 0,1582x(n-2) + 0,0791x(n-4)$$

sehingga persamaan untuk fungsi  $y(n)$  adalah sebagai berikut:

$$y(n) = 0,0791x(n) - 0,1582x(n-2) + 0,0791x(n-4) + 2,9449y(n-1) - 3,3196y(n-2) + 1,7498y(n-3) - 0,3786y(n-4)$$

Realisasi untuk persamaan filter ini ditunjukkan dalam Gambar 4.11.



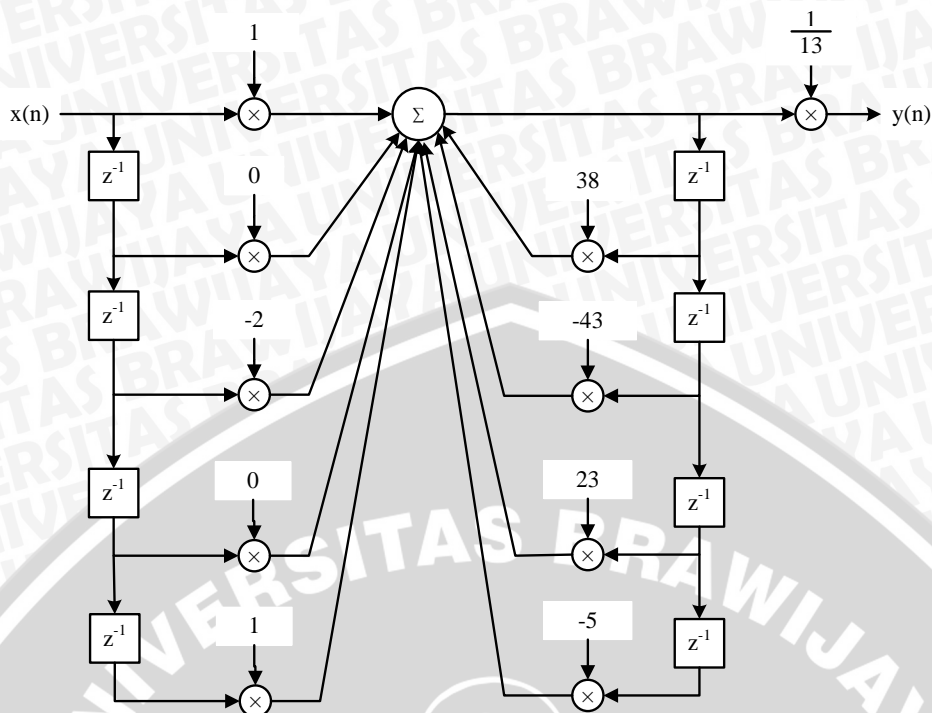
Gambar 4.11 Realisasi Persamaan Filter IIR

Karena persamaan filter IIR memuat nilai-nilai yang berupa bilangan pecahan, maka persamaan ini didekati dengan nilai bilangan bulat. Hal ini dilakukan untuk mempercepat proses perhitungan yang dilakukan oleh mikrokontroler, dan sekaligus disesuaikan dengan kemampuan mikrokontroler yang dirancang untuk melakukan operasi aritmatik bilangan bulat. Sehingga persamaan filter menjadi:

$$y(n) = \frac{x(n) - 2 \cdot x(n - 2) + x(n - 4) + 38 \cdot y(n - 1) - 43 \cdot y(n - 2) + 23 \cdot y(n - 3) - 5 \cdot y(n - 4)}{13}$$

Realisasi persamaan yang digunakan dalam program mikrokontroler ditunjukkan dalam Gambar 4.12.





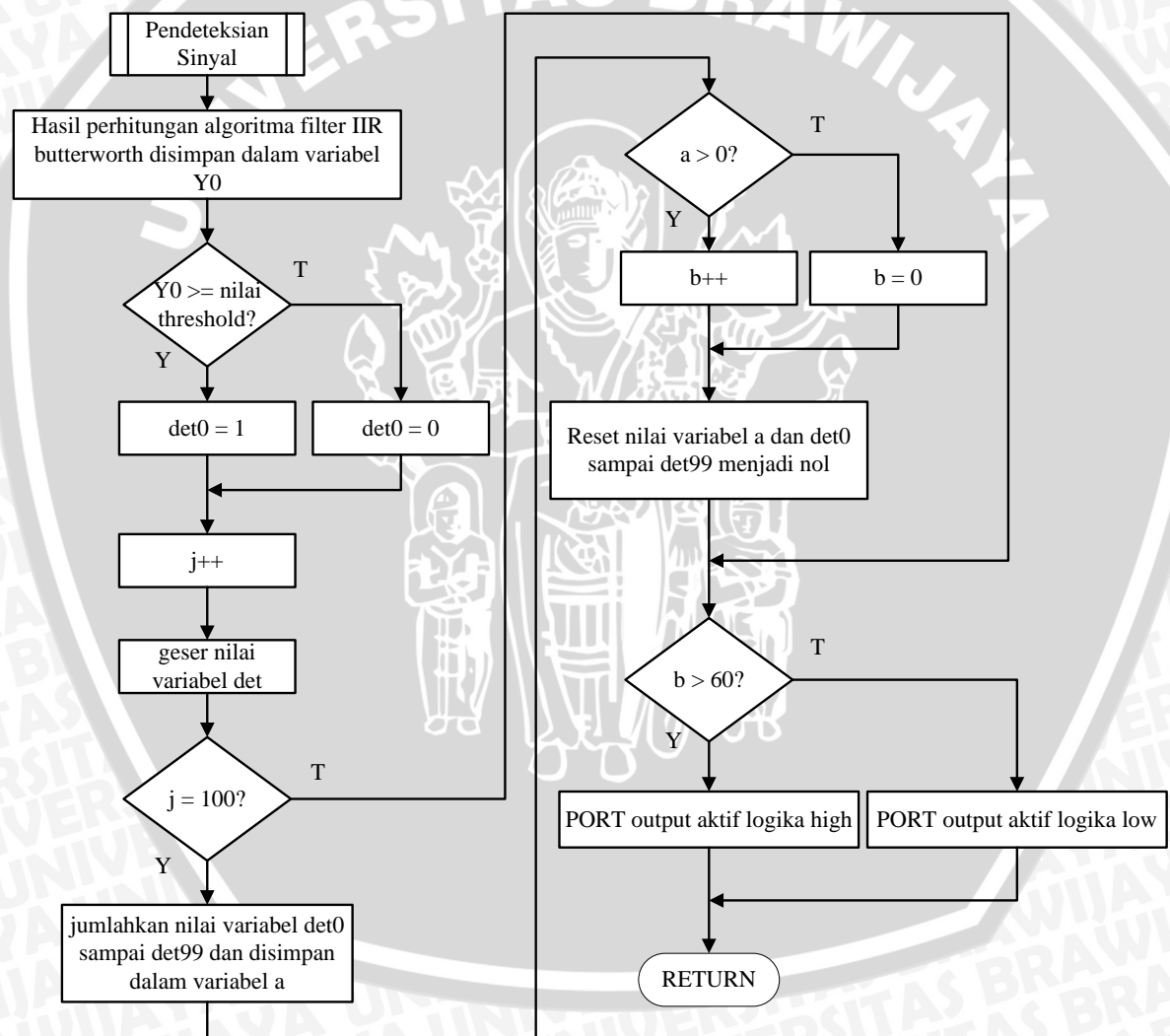
Gambar 4.12 Realisasi Persamaan Filter IIR dalam Program

Setelah diolah berdasarkan algoritma filter IIR *Butterworth*, maka diterapkan suatu sistem pengambilan keputusan. Sistem pengambilan keputusan ini bekerja dengan membandingkan hasil perhitungan algoritma filter IIR *Butterworth* dengan suatu nilai *threshold* tertentu. Hasil perhitungan algoritma filter IIR *Butterworth* dari sinyal dengan frekuensi di luar daerah *passband* akan bernilai lebih kecil daripada nilai *threshold* tersebut. Perbandingan ini dilakukan pada selang waktu tertentu, yaitu tiap 100 data hasil perhitungan. Ketika pada selang tersebut terdapat nilai yang melebihi nilai *threshold*, maka dianggap terdeteksi sinyal, akan tetapi belum diputuskan bahwa sinyal tersebut berasal dari suara musik. Untuk membedakan apakah sinyal tersebut berasal dari suara musik atau suara manusia, maka dilakukan perhitungan berapa kali terdeteksi sinyal. Hal ini dilakukan untuk mengetahui kontinuitas sinyal, karena sinyal yang berasal dari suara manusia hanya muncul sesaat pada selang-selang waktu yang relatif pendek sehingga sinyalnya terputus-putus, sedangkan sinyal yang berasal dari musik muncul secara terus-menerus.

Ketika terdeteksi sinyal terus-menerus lebih dari 60 kali yang artinya terjadi pada selang 6000 data sampling hasil perhitungan algoritma filter digital, maka diputuskan bahwa sinyal tersebut berasal dari suara musik. Dan jika hanya terdeteksi sinyal kurang dari 60 kali, maka sinyal yang masuk bukan berasal dari suara musik. Penentuan nilai 60 ini adalah berdasarkan karakteristik sinyal yang berasal dari suara

manusia yang hanya muncul pada selang-selang waktu yang relatif pendek berdasarkan suku kata yang diucapkan. Ketika sinyal yang ditangkap berasal dari suara manusia, sistem akan mendeteksi adanya sinyal berkisar sebanyak 40 kali, kemudian tidak terdeteksi pada selang berkisar 20 kali, baru kemudian terdeteksi kembali. Hal ini menunjukkan bahwa sinyal yang berasal dari suara manusia tidaklah kontinyu, akan tetapi terputus-putus.

Sistem akan memberikan logika high pada port output jika telah benar-benar diputuskan bahwa sinyal berasal dari suara musik. Sebaliknya, jika tidak terdeteksi sinyal yang berasal dari suara musik, maka sistem akan memberikan logika low pada port outputnya. Diagram alir sistem pendeteksi sinyal ditunjukkan dalam Gambar 4.13.



Gambar 4.13 Diagram Alir Pendeteksian Sinyal

## BAB V

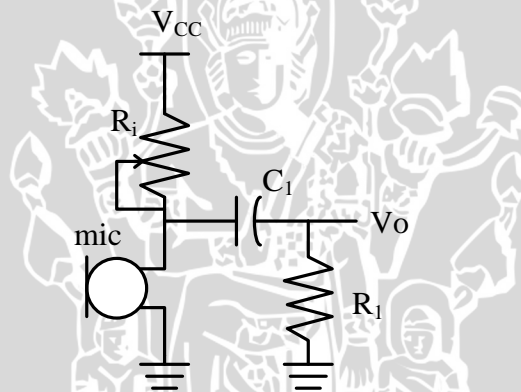
## PENGUJIAN DAN ANALISIS DATA

Pengujian alat ini bertujuan untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perencanaan. Pengujian ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk mempermudah analisis apabila alat tidak bekerja sesuai dengan perencanaan.

### 5.1 Pengujian Perangkat Keras

#### 5.1.1 Pengujian Microphone

Pengujian *microphone* ini bertujuan untuk mengetahui bahwa *microphone* dapat berfungsi dengan baik atau tidak. Pengujian dilakukan dengan menghubungkan keluaran rangkaian *microphone* dengan osiloskop TEKTRONIX TDS-1012B. *Microphone* diberi masukan berupa sinyal suara musik dan dilihat hasil keluarannya. Rangkaian pengujian *microphone* dapat dilihat dalam Gambar 5.1. Hasil pengujian ditunjukkan dalam Gambar 5.2.

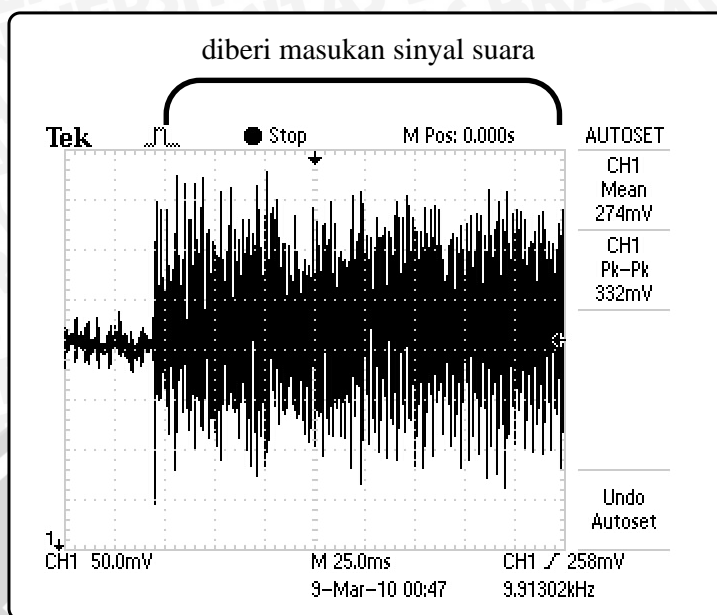


Gambar 5.1 Rangkaian Pengujian Microphone

Untuk mengatur tegangan *microphone* digunakan resistor variabel  $R_i$  yang dapat diubah-ubah. Dalam pengujian ini resistor variabel yang digunakan bernilai maksimum 50 k $\Omega$ . Kapasitor  $C_1$  yang digunakan bernilai 10nF dan resistor  $R_1$  bernilai 50k $\Omega$ . Nilai-nilai tersebut diambil dengan tujuan membuat suatu rangkaian *high pass filter* dengan nilai frekuensi cutoff sebesar 300 Hz. Terminal keluaran  $V_o$  dihubungkan dengan CH1 pada osiloskop agar dapat dilihat hasil keluarannya.

Berdasarkan hasil pengujian dapat diketahui bahwa *microphone* berfungsi dengan baik dan dapat menerima sinyal suara yang diperdengarkan.

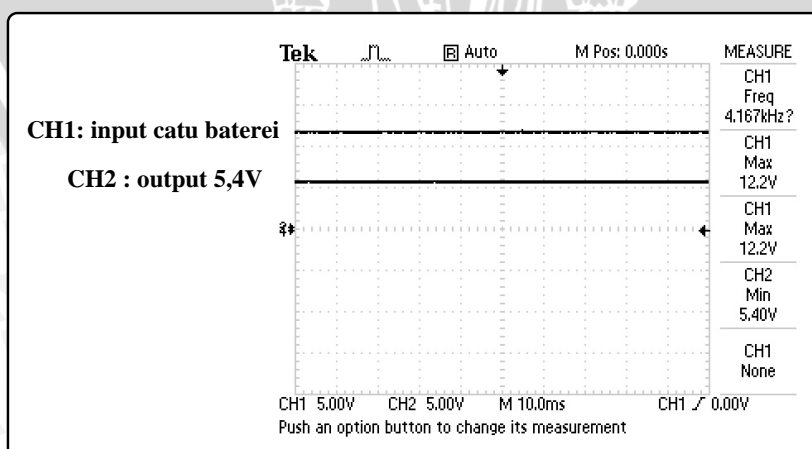




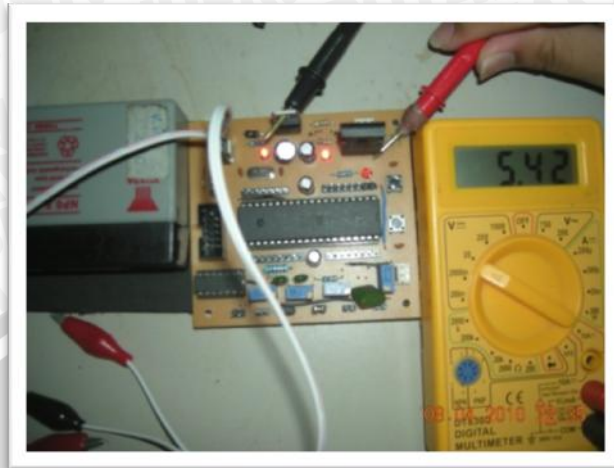
Gambar 5.2 Sinyal Keluaran Microphone

### 5.1.2 Pengujian Catu Daya Sistem

Pengujian rangkaian catu daya sistem bertujuan untuk mengetahui kesesuaian keluaran rangkaian catu daya. Pengujian dilakukan dengan menghubungkan masukan rangkaian catu daya dengan sumber DC konstan dan keluarannya diukur menggunakan multimeter yang difungsikan sebagai voltmeter untuk mengetahui besar tegangan keluarannya, sebagai perbandingan keluaran dan masukan rangkaian catu daya juga dihubungkan dengan osiloskop TEKTRONIX TDS-1012B. CH1 osiloskop dihubungkan dengan baterai 12V, dan CH2 dihubungkan dengan keluaran catu daya. Hasil pengujian ditunjukkan dalam Gambar 5.3 dan Gambar 5.4.



Gambar 5.3 Hasil Pengujian Rangkaian Catu Daya Menggunakan Osiloskop

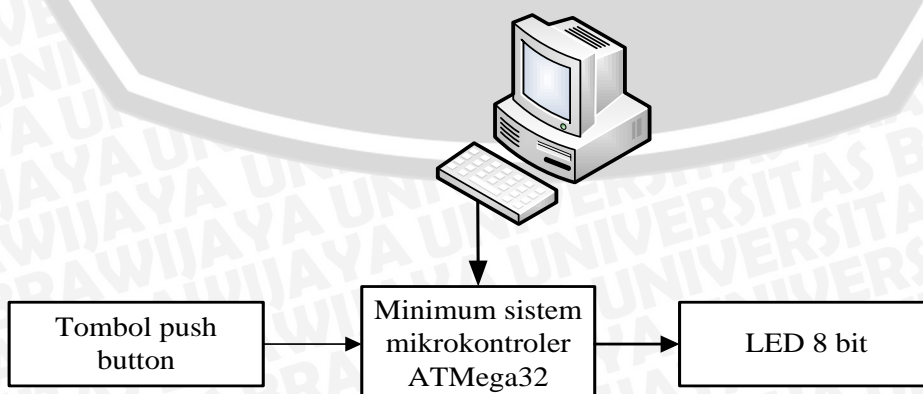


Gambar 5.4 Hasil Pengujian Rangkaian Catu Daya Menggunakan Multimeter

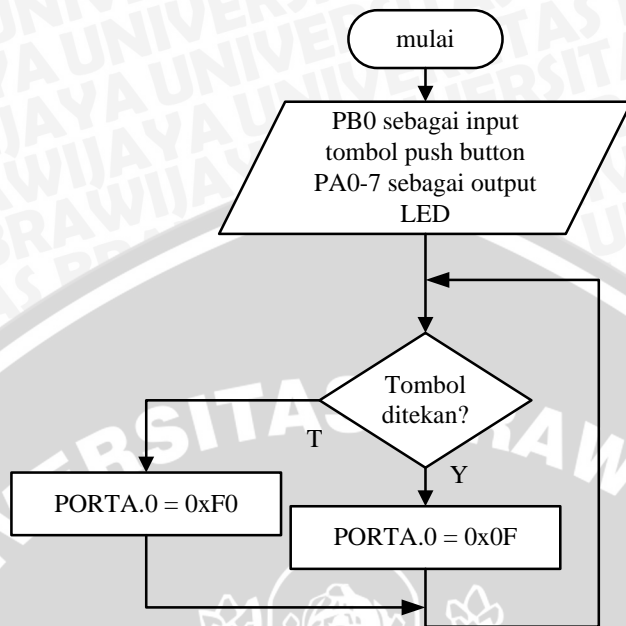
Dari hasil pengujian rangkaian catu daya tampak bahwa nilai tegangan keluaran rangkaian ini adalah 5,4V. Nilai ini masih berada di dalam *range* tegangan kerja mikrokontroler ATmega32 sehingga dapat diaplikasikan di dalam sistem.

### 5.1.3 Pengujian Minimum Sistem Mikrokontroler

Pengujian minimum sistem mikrokontroler bertujuan untuk mengetahui apakah mikrokontroler dapat berfungsi dengan baik atau tidak. Untuk pengujian ini, diagram blok yang digunakan ditunjukkan dalam Gambar 5.5. Masukan mikrokontroler berupa satu tombol push button dan keluarannya dihubungkan dengan LED yang mewakili keluaran 8 bit. Mikrokontroler diisi dengan program sederhana, yaitu jika tombol push button ditekan maka mikrokontroler akan mengeluarkan 0FH pada porta sebagai outputnya, akan tetapi jika tombol tidak ditekan akan mengeluarkan F0H. Diagram alir program pengujian ini ditunjukkan dalam Gambar 5.6.



Gambar 5.5 Diagram Blok Pengujian Minimum Sistem Mikrokontroler



Gambar 5.6 Diagram Alir Program Pengujian Minimum Sistem Mikrokontroler

Tabel 5.1 Hasil Pengujian Sistem Mikrokontroler

tombol push button	keluaran pada LED							
	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
ditekan	0	0	0	0	1	1	1	1
tidak ditekan	1	1	1	1	0	0	0	0

Berdasarkan tabel pengujian dapat diketahui bahwa ketika tombol push button ditekan maka output mikrokontroler mengeluarkan logika 0FH dan ketika tidak ditekan mengeluarkan logika F0H sesuai dengan isi program yang dituliskan dalam mikrokontroler. Ini menunjukkan bahwa minimum sistem mikrokontroler ATmega32 berfungsi dengan baik.

#### 5.1.4 Pengujian Rangkaian Filter Lowpass

Pengujian rangkaian filter lowpass ini bertujuan untuk mengetahui kondisi filter agar sesuai dengan spesifikasi yang diinginkan. Untuk melakukan pengujian ini menggunakan diagram blok seperti yang ditunjukkan dalam Gambar 5.7. Function generator digunakan sebagai masukan untuk menghasilkan sinyal berfrekuensi tertentu. Masukan dihubungkan pada CH1 dari osiloskop TEKTRONIX TDS-1012B dan keluaran filter dihubungkan pada CH2-nya. Pengujian dilakukan dengan memberikan





sinyal pada masukan filter lowpass menggunakan function generator dengan frekuensi yang diinginkan dengan tegangan peak to peak ( $V_{pp}$ ) sebesar 5V. Kemudian dilihat sinyal keluaran filter dan dicatat hasilnya. Hasil pengujian dapat dilihat dalam Tabel 5.2.



Gambar 5.7 Diagram Blok Pengujian Filter Lowpass



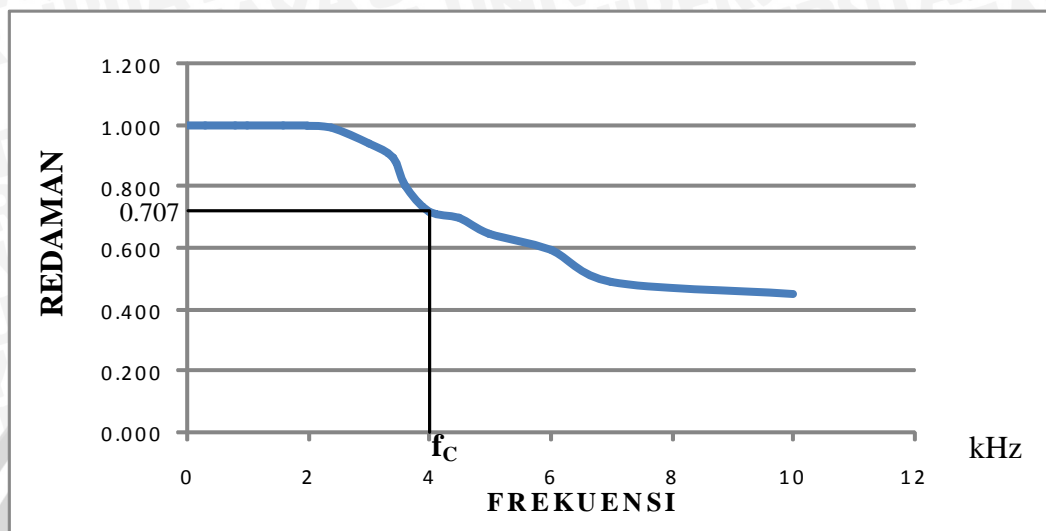
Gambar 5.8 Pengujian Rangkaian Filter Lowpass

Tabel 5.2 Hasil Pengujian Rangkaian Filter Lowpass

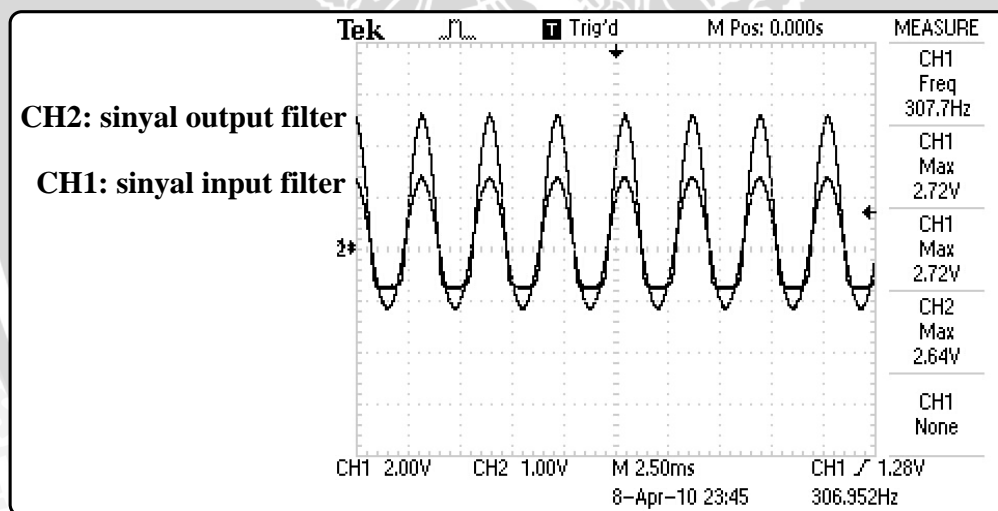
frekuensi (Hz)	$V_{in}$ p-p (volt)	$V_{out}$ p-p (volt)	$ H(j\omega) $
0	5 $V_{DC}$	5 $V_{DC}$	1
300	5	5	1
800	5	5	1
1000	5	5	1
1600	5	5	1
2000	5	5	1
2400	5	4,96	0,992
3000	5	4,72	0,944
3400	5	4,48	0,896
3600	5	4,02	0,804
4000	5	3,60	0,72
4500	5	3,48	0,696
5000	5	3,24	0,648
6000	5	2,96	0,592

7000	5	2,44	0,488
10000	5	2,24	0,448

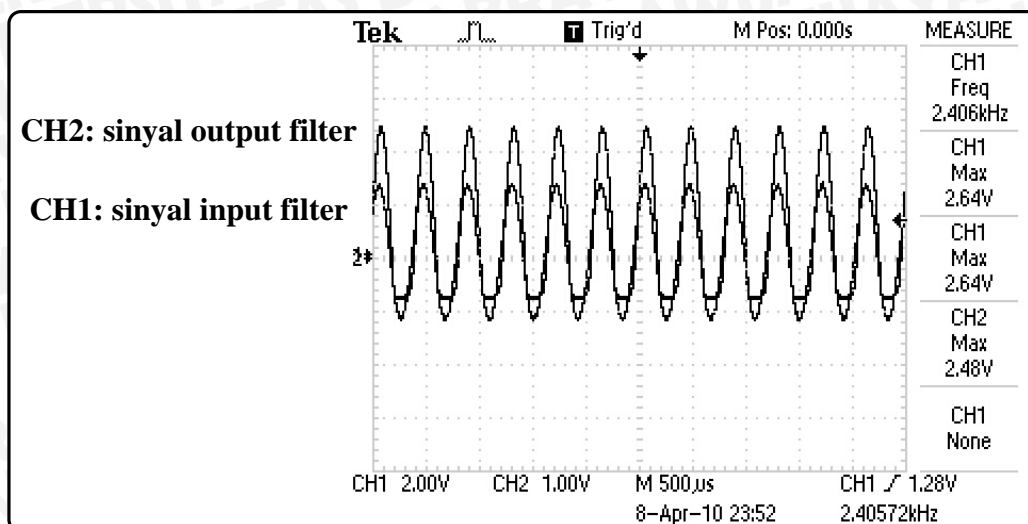
Berdasarkan hasil pengujian dalam Tabel 5.2 maka diperoleh grafik respon frekuensi yang ditunjukkan dalam Gambar 5.9.



Gambar 5.9 Grafik Respon Frekuensi Filter Lowpass



Gambar 5.10 Hasil Pengujian Rangkaian Filter Lowpass dengan Sinyal Masukan Frekuensi 300 Hz



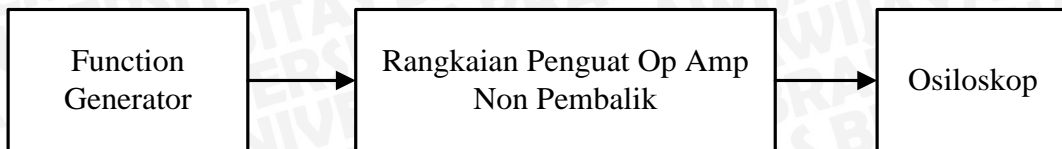
Gambar 5.11 Hasil Pengujian Rangkaian Filter Lowpass dengan Sinyal Masukan Frekuensi 2400 Hz

Berdasarkan hasil pengujian rangkaian filter lowpass sudah bekerja sesuai dengan perancangan. Frekuensi cutoff dirancang sebesar 4 kHz, dan dari hasil pengujian diperoleh frekuensi cutoff sebesar 4,12 kHz. Nilai ini masih mendekati spesifikasi yang diinginkan. Ketika masukan berupa sinyal dengan frekuensi kurang dari frekuensi cutoff, sinyal keluaran filter sama besar atau mendekati nilai sinyal masukan. Ketika masukan berupa sinyal dengan frekuensi lebih besar daripada frekuensi cutoff, maka sinyal keluaran mulai diredam. Peredaman sinyal pada frekuensi yang masih mendekati nilai cutoff tidaklah terlalu besar, dikarenakan penggunaan orde filter yang relatif kecil.

### 5.1.5 Pengujian Penguat Op-amp Non Pembalik

Pengujian penguat op amp non pembalik bertujuan untuk mengetahui keluaran penguat telah sesuai dengan perancangan atau tidak. Hal ini perlu dilakukan untuk memastikan bahwa sinyal telah dikuatkan pada level tegangan yang diinginkan agar dapat diproses oleh ADC pada mikrokontroler. Pengujian dilakukan dengan merangkai peralatan sesuai dengan diagram blok yang ditunjukkan dalam Gambar 5.12. Function generator digunakan sebagai masukan untuk menghasilkan sinyal dengan frekuensi tertentu dengan tegangan peak to peak ( $V_{pp}$ ) yang diinginkan. Osiloskop TEKTRONIX TDS-1012B digunakan untuk menampilkan sinyal masukan penguat pada CH1 dan sinyal keluaran penguat pada CH2. Pengujian dilakukan dengan memberikan sinyal masukan pada penguat dengan beberapa frekuensi tertentu dan mengubah-ubah nilai tegangan peak to peaknya dan kemudian diamati sinyal keluaran yang dihasilkan. Hasil pengujian penguat op amp non pembalik dapat dilihat dalam Tabel 5.3.

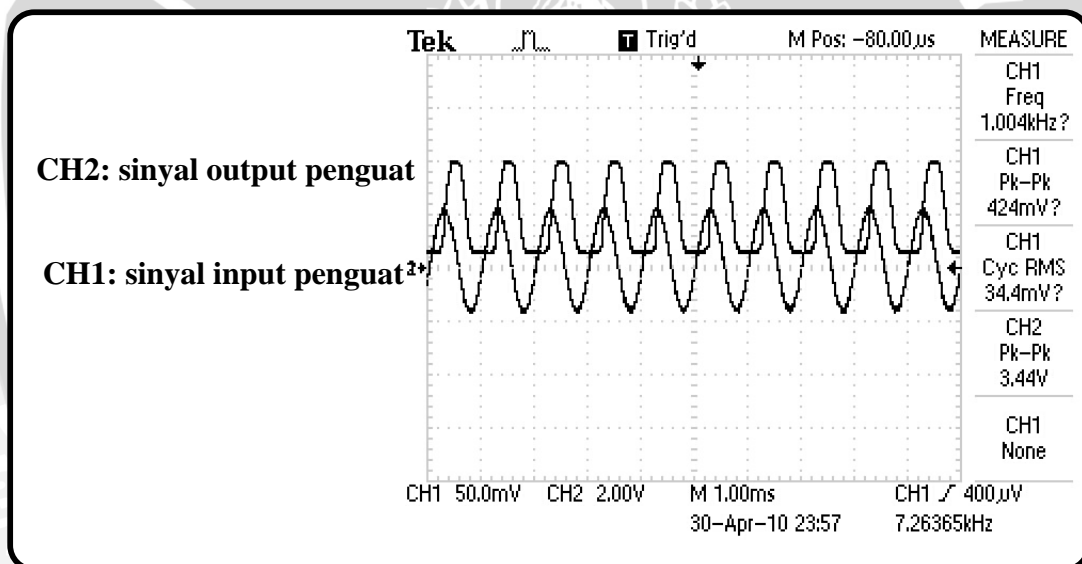




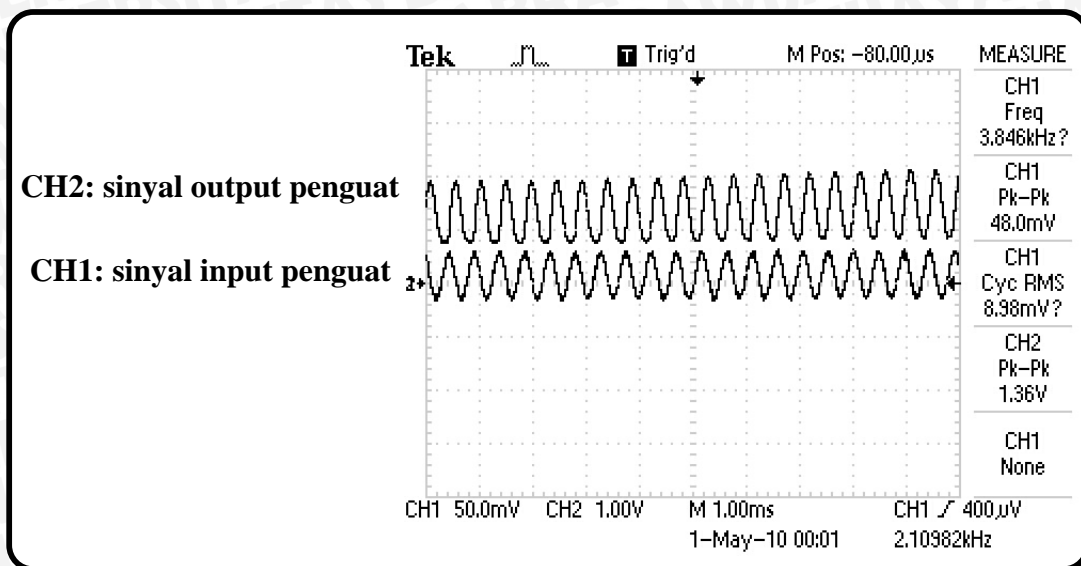
Gambar 5.12 Diagram Blok Pengujian Penguat Op Amp Non Pembalik

Tabel 5.3 Hasil Pengujian Penguat Op Amp Non Pembalik

frekuensi (Hz)	Vin p-p (milivolt)	Vout p-p (volt)
1000	50	2,12
1000	100	3,44
2000	50	1,56
2000	100	2,72
3000	50	1,36
3000	200	2,48



Gambar 5.13 Hasil Pengujian Penguat Op Amp Non Pembalik dengan Sinyal Masukan Frekuensi 1 kHz Amplitudo 100 mVpp



Gambar 5.14 Hasil Pengujian Penguat Op Amp Non Pembalik dengan Sinyal Masukan Frekuensi 3 kHz Amplitudo 50 mVpp

Berdasarkan hasil pengujian, rangkaian penguat op amp non pembalik telah bekerja dengan baik dan dapat menguatkan sinyal ke level tegangan yang diharapkan yaitu ke level tegangan yang dapat dibaca oleh mikrokontroler. Pada siklus negatif (pada saat sinyal masukan bernilai negatif), sinyal keluaran penguat akan dipotong pada ground. Hal ini karena penggunaan ground sebagai tegangan saturasi negatifnya. Dan amplitudo maksimal sinyal keluaran adalah 4 volt, karena penggunaan  $V_{CC}$  sebagai tegangan saturasi positif pada rangkaian penguat. Ketika sinyal keluaran bernilai lebih dari tegangan saturasi, sinyal tersebut akan dipotong pada nilai 4 volt.

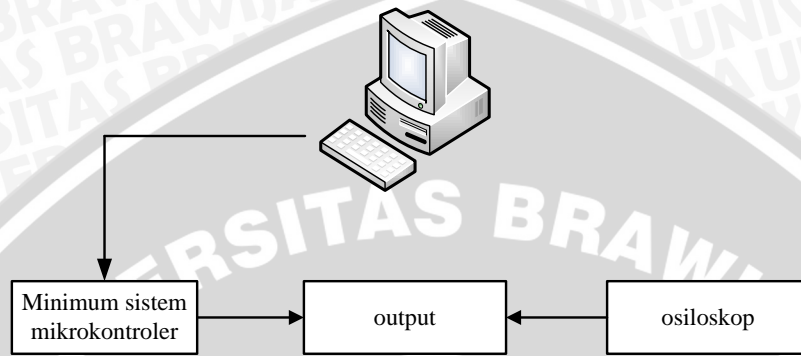
## 5.2 Pengujian Perangkat Lunak

### 5.2.1 Pengujian Frekuensi Sampling

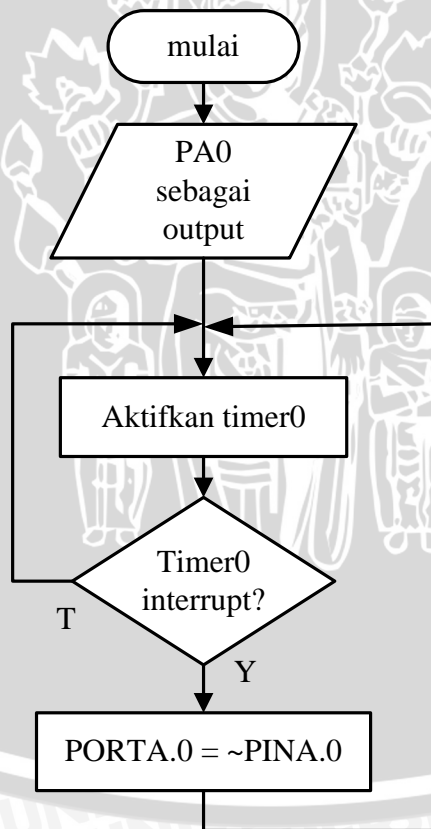
Pengujian ini bertujuan untuk mengetahui besar frekuensi sampling yang diterapkan agar sesuai dengan perhitungan yang diharapkan. Blok diagram pengujian frekuensi sampling ditunjukkan dalam Gambar 5.15. Untuk mengatur frekuensi sampling digunakan fitur interupsi timer0 pada mikrokontroler. Diagram alir pengujian frekuensi sampling ditunjukkan dalam Gambar 5.16. Keluaran program ditampilkan dalam osiloskop untuk melihat periode perubahan logika pada pin output mikrokontroler.

Timer0 yang digunakan diatur supaya memanggil program interupsi tiap  $50 \mu s$ , sehingga frekuensi pemanggilan interupsi adalah  $1/(50 \cdot 10^{-6}) = 20kHz$ . Pin output yang digunakan dalam pengujian akan berubah kondisi logikanya setiap kali terjadi interupsi

timer0 ini. Hal ini menyebabkan pin output memiliki frekuensi sebesar setengah dari frekuensi 20kHz, karena pin output memiliki periode dua kali periode timer0. Dari hasil pengujian yang dapat dilihat dalam Gambar 5.17 diperoleh periode pin output sebesar 102  $\mu$ s sehingga frekuensinya sebesar 9,803 kHz sehingga nilai frekuensi samplingnya adalah sebesar 19,606 kHz. Nilai ini mendekati nilai 20 kHz sehingga dapat dikatakan bahwa timer0 telah berfungsi dengan baik sesuai dengan perancangan.

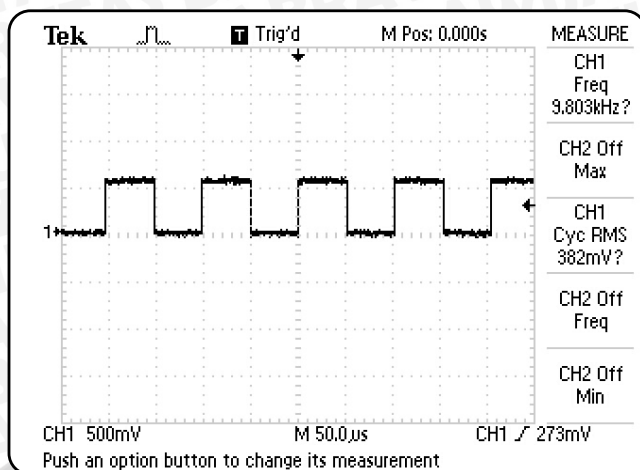


Gambar 5.15 Diagram Blok Pengujian Frekuensi Sampling



Gambar 5.16 Diagram Alir Program Pengujian Frekuensi Sampling





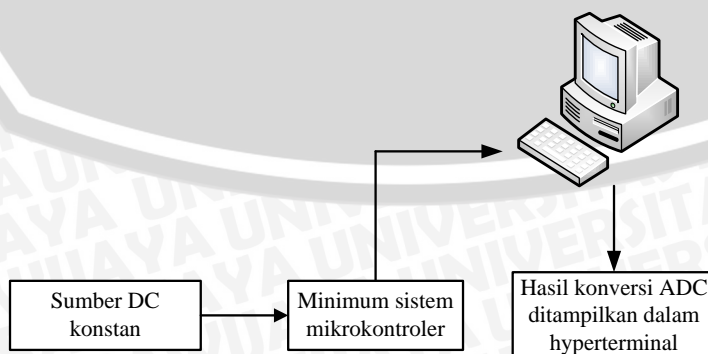
Gambar 5.17 Hasil Pengujian Frekuensi Sampling

### 5.2.2 Pengujian Output ADC

Pengujian ini bertujuan untuk mengetahui apakah kinerja ADC pada mikrokontroler sesuai dengan spesifikasi yang tertera dalam datasheet atau tidak. Peralatan yang digunakan dalam pengujian ini antara lain adalah minimum sistem mikrokontroler, dan komputer. Salah satu channel ADC pada mikrokontroler dihubungkan dengan suatu sumber DC konstan dan hasil konversinya ditampilkan pada hyperterminal. Susunan peralatan dalam pengujian ini ditunjukkan dalam Gambar 5.18. Hasil konversi ADC akan memenuhi persamaan:

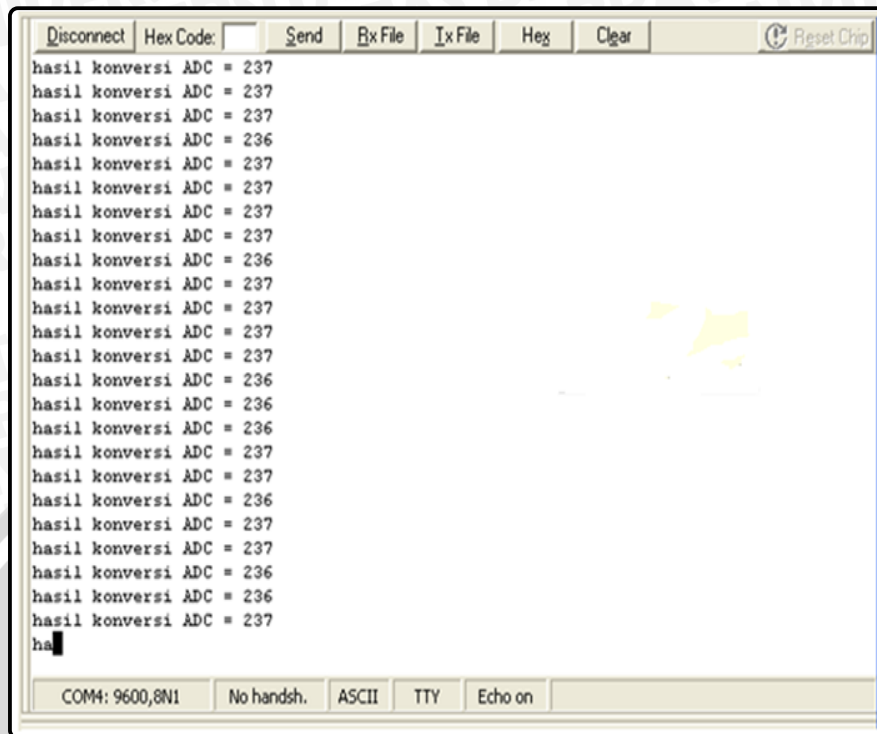
$$out = \frac{V_{in} \cdot (2^{resolusi} - 1)}{V_{reff}}$$

yang mana tegangan referensi  $V_{reff}$  adalah sama dengan tegangan maksimum yang digunakan. Dalam pengujian ini digunakan  $V_{cc}$  sebagai tegangan referensi yang nilainya adalah 5,4V. Hasil pengujian ADC mikrokontroler ditunjukkan dalam Tabel 5.4.



Gambar 5.18 Diagram Blog Pengujian Output ADC

Bentuk tampilan *hyperterminal* pada salah satu level tegangan:



Gambar 5.19 Bentuk Tampilan Keluaran Hyperterminal

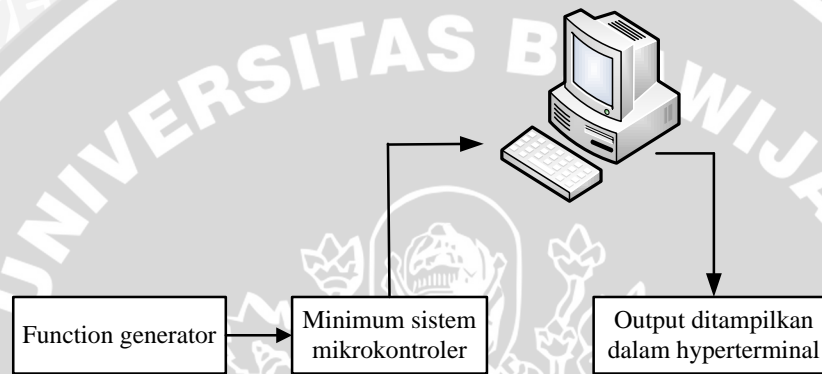
Tabel 5.4 Hasil Pengujian Output ADC

tegangan input analog	output ADC secara teori		output ADC pada hyperterminal	
	8 bit	10 bit	8 bit	10 bit
0V	0	0	0	0
0,5V	23	94	24	96
1V	47	189	48	190
1,5 V	70	284	70	287
2 V	94	378	95	380
2,5 V	118	473	118	476
3 V	141	568	140	569
3,5 V	165	663	165	665
4 V	188	757	188	760
4,5 V	212	852	211	851
5 V	236	947	237	946

Berdasarkan hasil pengujian terdapat nilai error dari konversi ADC. Akan tetapi hal ini masih sesuai dengan datasheet bahwa error yang terjadi adalah sebesar +/- 3 LSB.

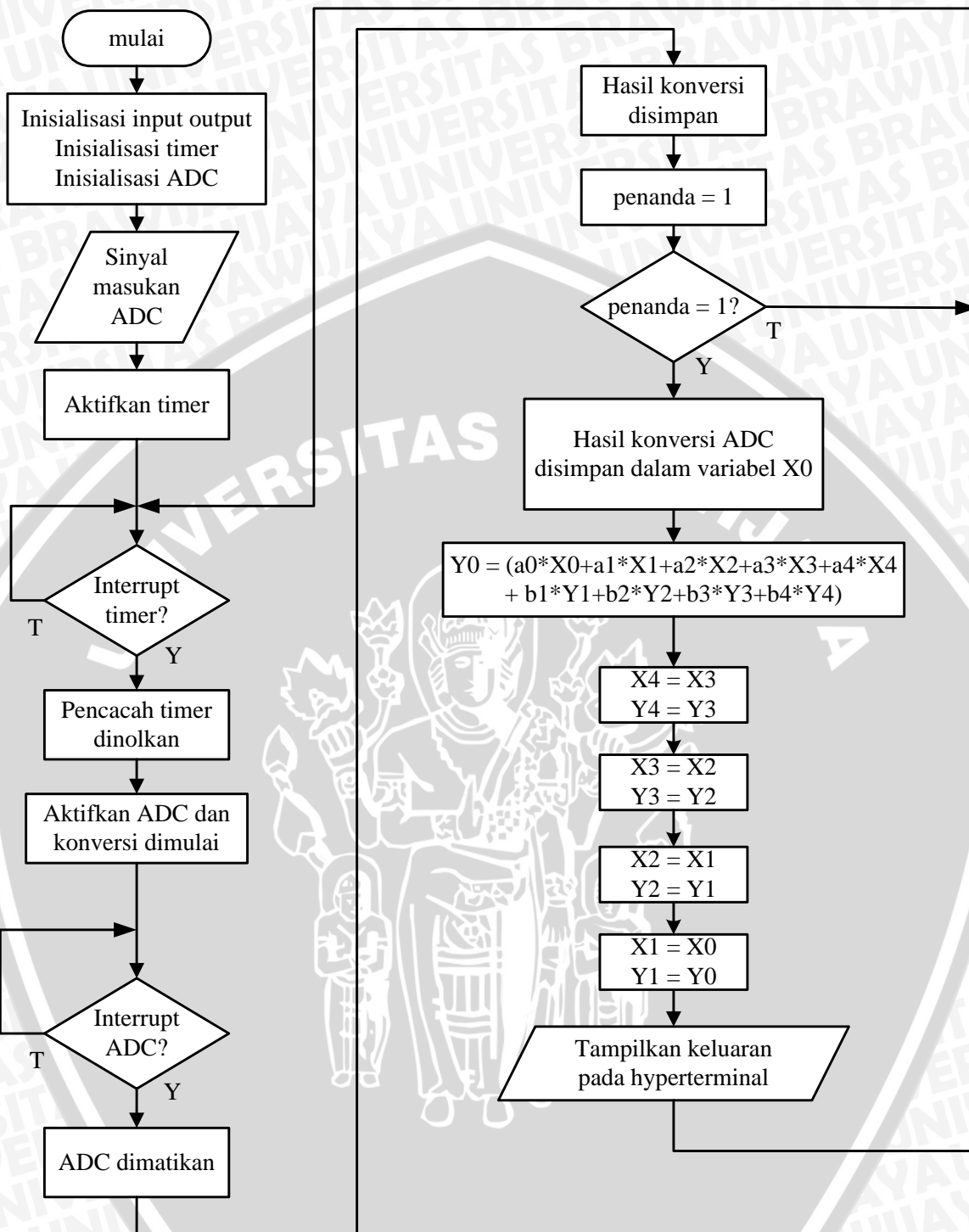
### 5.2.3 Pengujian Persamaan Filter IIR orde empat

Pengujian persamaan filter IIR orde empat ini bertujuan untuk mengetahui kinerja filter digital ini telah menghasilkan keluaran yang sudah benar atau tidak. Untuk melakukan pengujian ini maka perlengkapan disusun sesuai dengan diagram blok pengujian sebagaimana ditunjukkan dalam Gambar 5.20. Salah satu kanal ADC diberi sinyal masukan yang berasal dari *function generator*. Keluaran filter IIR orde empat yang dirancang diamati dengan menggunakan program *hyperterminal*. Diagram alir program pengujian persamaan filter IIR orde empat ini ditunjukkan dalam Gambar 5.21. Hasil pengujian dapat dilihat dalam Tabel 5.5 sampai Tabel 5.7.



Gambar 5.20 Diagram Blok Pengujian Persamaan Filter IIR Orde empat





Gambar 5.21 Diagram Alir Program Pengujian Persamaan Filter IIR Orde empat

Tabel 5.5 Hasil Pengujian Persamaan Filter IIR Orde empat dengan Sinyal Masukan DC konstan

variabel	kondisi waktu diskrit							
	n-7	n-6	n-5	n-4	n-3	n-2	n-1	n
x0	1017	1017	1017	1016	1017	1017	1016	1016
x1	1016	1017	1017	1017	1016	1017	1017	1016
x2	1017	1016	1017	1017	1017	1016	1017	1017
x3	1016	1017	1016	1017	1017	1017	1016	1017
x4	1016	1016	1017	1016	1017	1017	1017	1016
y0	3	3	2	0	-2	-3	-3	-2
y1	2	3	3	2	0	-2	-3	-3
y2	0	2	3	3	2	0	-2	-3
y3	-2	0	2	3	3	2	0	-2
y4	-3	-2	0	2	3	3	2	0

Kondisi sekarang dinotasikan dengan  $n$ , kondisi sebelum  $n$  dinotasikan dengan  $n-1$ , dan kondisi sebelum  $n-1$  dinotasikan dengan  $n-2$ , dan seterusnya. Data sampling ADC disimpan dalam variabel `data_adc` yang kemudian disalin dalam variabel `x0` sebagai masukan persamaan filter IIR orde dua. Variabel `x1` dan `x2` digunakan untuk menyimpan pergeseran nilai dari `x0` pada kondisi sebelum sekarang ( $n-1$ ) dan kondisi sebelum  $n-1$  ( $n-2$ ) dan seterusnya. Hasil perhitungan filter untuk kondisi sekarang disimpan dalam variabel `y0`, kondisi sebelumnya disimpan dalam variabel `y1`, dan kondisi sebelumnya lagi disimpan dalam variabel `y2` dan seterusnya.

Pengujian persamaan filter IIR orde empat ini dilakukan tiga kali dengan masukan sinyal yang berbeda-beda. Pada pengujian pertama, masukan yang diberikan pada kanal ADC berupa sumber DC konstan. Hasil pengujiannya ditunjukkan dalam Tabel 5.5. Pengujian kedua menggunakan masukan berupa sinyal sinusoida dengan frekuensi sebesar 1000 Hz yang merupakan sinyal dengan frekuensi yang berada pada daerah *passband* dari filter. Hasil pengujian dapat dilihat dalam Tabel 5.6. Pengujian ketiga menggunakan masukan berupa sinyal sinusoida dengan frekuensi sebesar 4000 Hz yang merupakan sinyal dengan frekuensi yang berada pada daerah *stopband* dari filter. Hasil pengujian dapat dilihat dalam Tabel 5.7.

Tabel 5.6 Hasil Pengujian Persamaan Filter IIR Orde empat dengan Sinyal Masukan dengan frekuensi 1000 Hz

variabel	kondisi waktu diskrit							
	n-7	n-6	n-5	n-4	n-3	n-2	n-1	n
x0	579	711	561	222	129	129	129	151
x1	151	579	711	561	222	129	129	129
x2	129	151	579	711	561	222	129	129
x3	129	129	151	579	711	561	222	129
x4	129	129	129	151	579	711	561	222
y0	-157	6	192	262	169	-4	-147	-209
y1	-206	-157	6	192	262	169	-4	-147
y2	-147	-206	-157	6	192	262	169	-4
y3	-8	-147	-206	-157	6	192	262	169
y4	162	-8	-147	-206	-157	6	192	262

Tabel 5.7 Hasil Pengujian Persamaan Filter IIR Orde empat dengan Sinyal Masukan dengan frekuensi 4000 Hz

variabel	kondisi waktu diskrit							
	n-7	n-6	n-5	n-4	n-3	n-2	n-1	n
x0	569	129	760	129	150	757	129	748
x1	271	569	129	760	129	150	757	128
x2	129	271	569	129	760	129	150	757
x3	760	129	271	569	129	760	129	150
x4	129	760	129	271	569	129	760	129
y0	-26	-8	20	55	28	-13	4	34
y1	-15	-26	-8	20	55	28	-13	4
y2	-9	-15	-26	-8	20	55	28	-13
y3	-19	-9	-15	-26	-8	20	55	28
y4	32	-19	-9	-15	-26	-8	20	55

Bentuk persamaan filter IIR orde empat yang digunakan adalah:

$$y(n) = 0,0791.x(n) - 0,1582.x(n-2) + 0,0791.x(n-4) + 2,9449.y(n-1) - 3,3196.y(n-2) + 1,7498.y(n-3) - 0,3786.y(n-4)$$



sedangkan bentuk realisasi dalam program dinyatakan dengan:

$$y_0 = (x_0 - 2x_1 + x_2 + 38y_1 - 43y_2 + 23y_3 - 5y_4) / 13$$

berdasarkan pengujian dengan menggunakan sumber DC konstan sebagai sinyal masukan, pada kondisi waktu diskrit  $n$ , nilai variabel-variabelnya adalah sebagai berikut:

$$x_0 = 1016, x_1 = 1016, x_2 = 1017, x_3 = 1017, x_4 = 1016$$

$$y_1 = -3, y_2 = -3, y_3 = -2, y_4 = 0$$

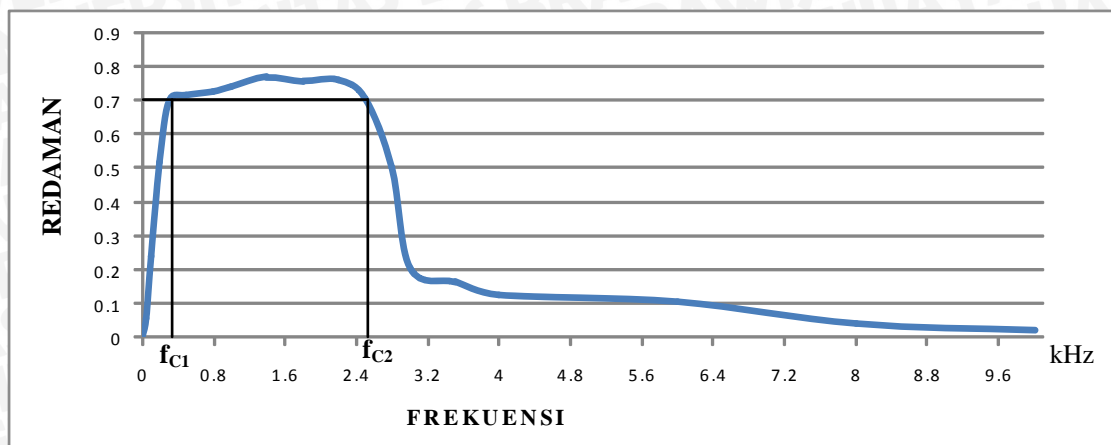
nilai  $y_0$  berdasarkan persamaan filter IIR orde empat yang sesungguhnya adalah -2,53, sedangkan berdasarkan persamaan pada realisasi program,  $y_0$  bernilai -2. Nilai ini tidak bernilai pecahan karena tipe data variabelnya adalah signed integer, namun nilai ini masih mendekati nilai yang seharusnya.

Pengujian dilakukan untuk mengetahui respon frekuensi dari persamaan filter IIR orde empat yang telah dirancang. Hasil pengujian ini ditunjukkan dalam Tabel 5.8.

Tabel 5.8 Hasil Pengujian Persamaan Filter IIR Orde empat

frekuensi(Hz)	ADC sinyal input		output filter		amplitudo		redaman
	min	max	min	max	sinyal input	output	
0	1017 (sinyal DC)		-3	3	1017	6	0,0059
50	129	760	-7	29	631	36	0,057
100	129	760	-64	87	631	151	0,239
200	129	760	-98	230	631	328	0,519
300	129	760	-196	246	631	442	0,700
500	129	760	-200	251	631	451	0,715
800	129	760	-201	258	631	459	0,727
1000	129	760	-206	262	631	468	0,742
1400	129	760	-282	286	631	568	0,769
1800	129	760	-280	271	631	551	0,758
2200	129	760	-216	264	631	480	0,721
2500	129	760	-199	247	631	446	0,707
2800	129	760	-89	228	631	317	0,502
3000	129	760	-69	60	631	129	0,204
3500	129	760	-48	56	631	104	0,165
4000	129	760	-26	55	631	81	0,128
6000	129	760	-20	47	631	67	0,106
8000	129	760	-8	18	631	26	0,041
10000	129	760	-6	9	631	15	0,024

Besar nilai redaman diperoleh dengan cara membandingkan nilai amplitudo keluaran persamaan filter IIR orde empat dengan nilai masukannya. Berdasarkan Tabel 5.8 maka diperoleh grafik respon frekuensi filter yang dirancang. Grafik respon filter IIR orde empat ditunjukkan dalam Gambar 5.22.



Gambar 5.22 Grafik Respon Frekuensi Filter IIR Orde Empat

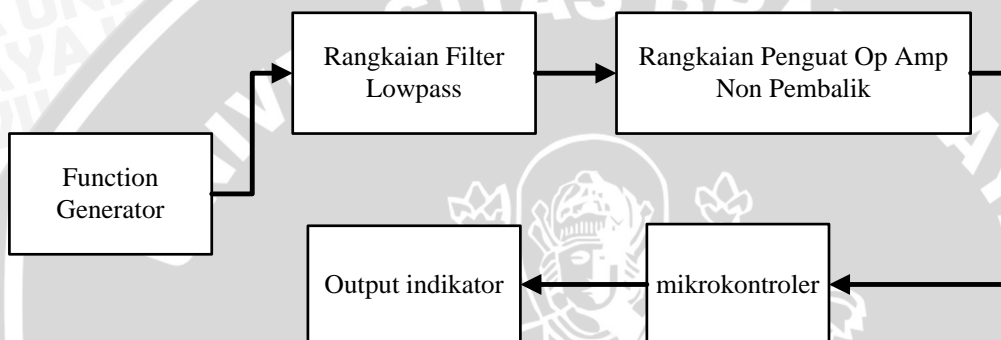
Berdasarkan grafik respon frekuensi yang diperoleh, persamaan filter IIR orde empat sudah bekerja sesuai dengan perancangan. Frekuensi cutoff bawah dirancang sebesar 300 Hz dan frekuensi cutoff atas sebesar 2500 Hz. Berdasarkan hasil pengujian diperoleh frekuensi cutoff bawah berada pada nilai 320 Hz dan frekuensi cutoff atas berada pada nilai 2500 Hz. Nilai ini masih mendekati nilai spesifikasi yang diinginkan. Ketika masukan berupa sinyal dengan frekuensi di daerah bandstop, keluaran filter mengalami peredaman. Ketika masukan berupa sinyal dengan frekuensi yang berada dalam daerah bandpass maka keluaran filter mendekati besar masukannya. Peredaman sinyal pada frekuensi yang masih mendekati nilai cutoff tidaklah terlalu besar, dikarenakan penggunaan orde filter yang relatif kecil. Sehingga dapat dikatakan bahwa hasil keluaran filter telah sesuai dengan yang diinginkan.

#### 5.2.4 Pengujian Algoritma Pendeteksian Sinyal

Pengujian ini bertujuan untuk mengetahui keberhasilan sistem mendeteksi sinyal. Diagram blok pengujian algoritma pendeteksian sinyal dapat dilihat dalam Gambar 5.23, sedangkan diagram alir program pengujiannya dapat dilihat dalam Gambar 5.24. Pendeteksian ini dilakukan setiap seratus kali sampling. Setiap sampling data hasil perhitungan algoritma filter IIR orde empat akan dibandingkan dengan satu nilai *threshold* tertentu. Jika nilai hasil keluaran filter IIR pada saat tertentu melebihi nilai *threshold*, maka variabel det sebagai variabel penanda akan berisi nilai satu, sebaliknya jika kurang dari nilai *threshold* variabel det akan berisi nilai nol. Variabel det berupa array, sehingga untuk menjamin proses ini terjadi secara kontinyu, nilai variabel det akan digeser sesuai dengan kondisi waktu samplingnya. Untuk elemen ke-0 akan

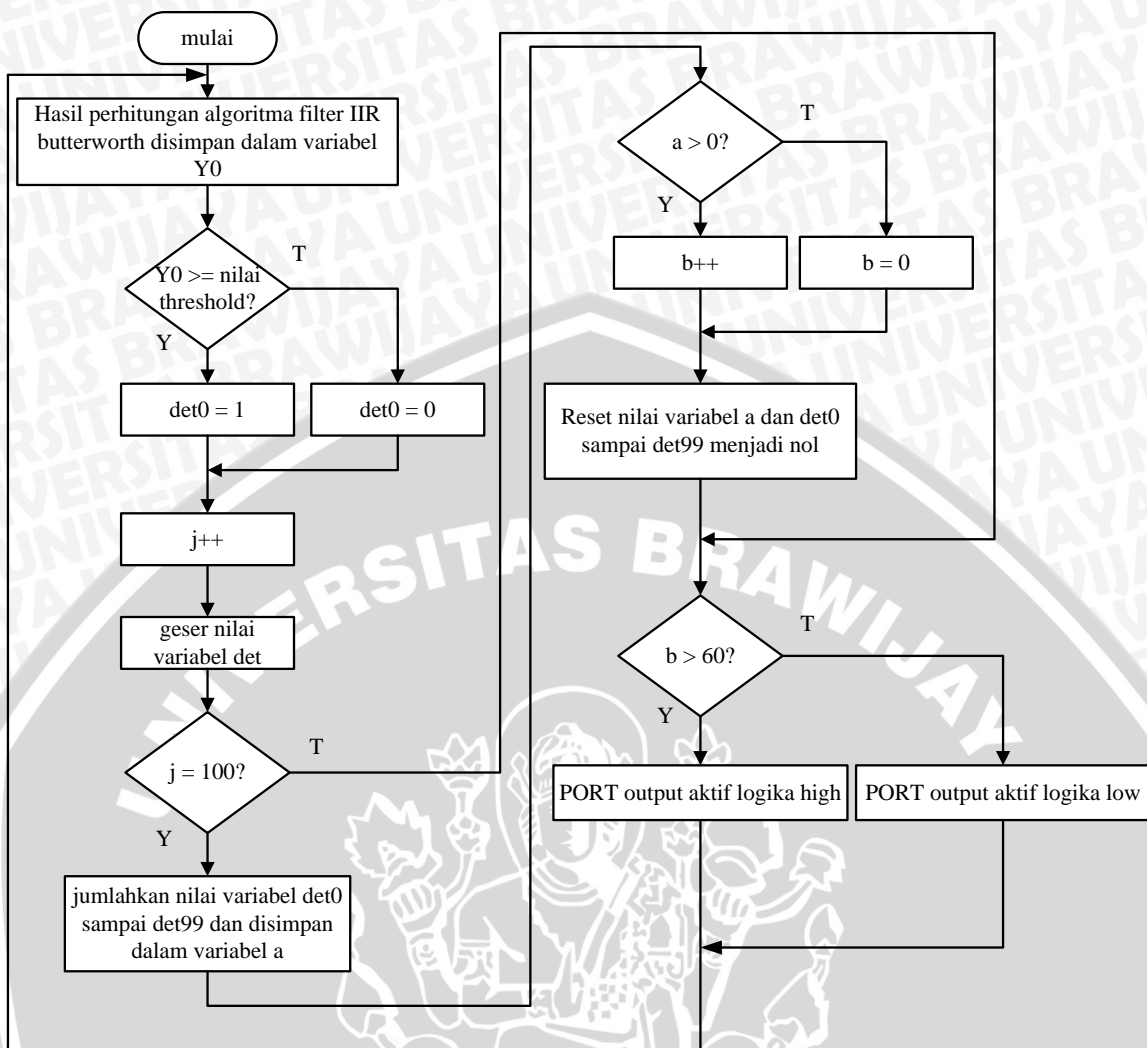
mewakili kondisi sekarang ( $n$ ), elemen ke-1 akan mewakili kondisi sebelum sekarang ( $n-1$ ), dan seterusnya.

Nilai tiap elemen dari variabel det dijumlahkan ketika jumlah data yang dibandingkan dengan nilai *threshold* telah mencapai seratus data. Jika hasil penjumlahan variabel det bernilai lebih dari nol maka pada range seratus data sampling tersebut terdeteksi adanya sinyal. Kemudian dengan menggunakan variabel pencacah akan dihitung berapa kali sinyal terdeteksi. Jika sinyal terdeteksi lebih dari enam puluh kali secara terus-menerus, maka port output indikator akan bernilai satu, namun jika sinyal hanya terdeteksi secara sesaat dan tidak terus-menerus maka port output indikator akan bernilai nol.



Gambar 5.23 Diagram Blok Pengujian Algoritma Pendeteksian Sinyal





Gambar 5.24 Diagram Alir Program Pengujian Algoritma Pendeteksian Sinyal



Gambar 5.25 Hasil Pengujian dengan Masukan Sinyal Sinusoida Frekuensi 1000 Hz



Gambar 5.26 Hasil Pengujian dengan Masukan Sinyal Sinusoida Frekuensi 2800 Hz

Tabel 5.9 Hasil Pengujian Algoritma Pendeteksian Sinyal

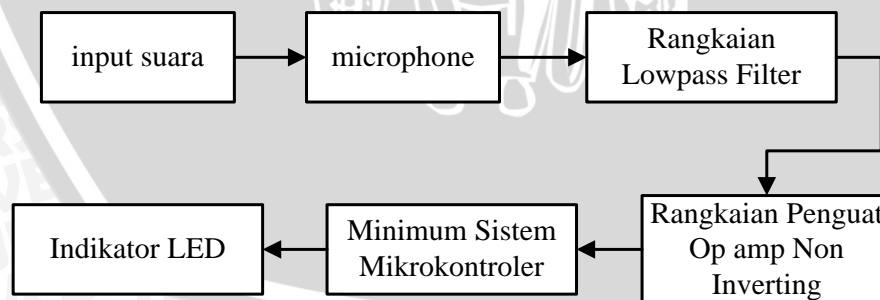
sumber sinyal masukan	frekuensi sinyal(Hz)	keluaran led indikator	keterangan
DC konstan	0	mati	tidak terdeteksi sebagai sinyal
sinusoida	100	mati	tidak terdeteksi sebagai sinyal
sinusoida	200	nyala	terdeteksi sebagai sinyal
sinusoida	1000	nyala	terdeteksi sebagai sinyal
sinusoida	2000	nyala	terdeteksi sebagai sinyal
sinusoida	2800	mati	tidak terdeteksi sebagai sinyal
sinusoida	4000	mati	tidak terdeteksi sebagai sinyal

Berdasarkan hasil pengujian, sistem telah dapat mendeteksi adanya sinyal. Filter IIR yang digunakan dalam pengujian adalah filter *bandpass* dengan frekuensi cutoff bawah 300 Hz dan frekuensi cutoff atas 2500 Hz. Hasil pengujian menunjukkan bahwa pada sinyal masukan dengan frekuensi 200 Hz masih terdeteksi sebagai sinyal, hal ini dikarenakan oleh penggunaan orde filter yang relatif kecil. Sedangkan, pada sinyal masukan berfrekuensi lebih dari 2800 Hz tidak terdeteksi sebagai sinyal. Hal ini

menunjukkan bahwa jika sinyal yang diberikan memiliki frekuensi yang berada dalam daerah *passband* dari filter, maka sinyal tersebut akan terdeteksi. Sebaliknya, jika sinyal yang diberikan memiliki frekuensi yang berada dalam daerah *stopband*, maka sinyal tersebut tidak terdeteksi. Dapat dikatakan bahwa sistem telah bekerja sesuai dengan yang diinginkan.

### 5.3 Pengujian Diagram Blok Sistem Keseluruhan

Pengujian ini bertujuan untuk mengetahui keberhasilan sistem dalam mendeteksi adanya sinyal, dalam hal ini adalah sinyal yang berasal dari suara musik. Untuk melakukan pengujian ini, maka disusun peralatan sesuai dengan diagram blok pengujian yang ditunjukkan dalam Gambar 5.27. *Microphone* digunakan sebagai sensor suara yang menangkap sinyal suara dari input. Rangkaian lowpass filter digunakan untuk membatasi frekuensi maksimum sinyal masukan yang akan diproses oleh ADC dalam mikrokontroler. Rangkaian penguat op amp non inverting digunakan untuk mengubah level tegangan sinyal yang ditangkap oleh *microphone* ke level tegangan yang dapat diolah oleh mikrokontroler. Mikrokontroler yang digunakan adalah jenis ATmega32, sebagai sistem pengolah data. Di dalamnya terdapat ADC yang mengubah sinyal masukan analog menjadi sinyal diskrit, filter IIR *Butterworth* orde empat sebagai bandpass filter yang digunakan untuk memisahkan sinyal informasi yang berasal dari musik dan sinyal noise, dan terdapat sistem pendeteksi sinyal yang memutuskan sinyal yang ditangkap benar-benar berasal dari musik atau tidak. Indikator LED digunakan sebagai tanda bahwa sinyal masukan berupa sinyal informasi yang diharapkan atau bukan.



Gambar 5.27 Diagram Blok Pengujian Sistem Keseluruhan



Tabel 5.10 Hasil Pengujian Sistem Keseluruhan

kondisi <i>microphone</i>	masukan	LED indikator	keterangan
diberi lapisan berbentuk corong	musik	nyala	terdeteksi sebagai sinyal
	suara manusia	kedip	masih ada yang terdeteksi sebagai sinyal
	tanpa masukan	mati	tidak terdeteksi sinyal
diberi lapisan tipis di bagian ujung <i>microphone</i>	musik	nyala	terdeteksi sebagai sinyal
	suara manusia	mati	tidak terdeteksi sebagai sinyal
	tanpa masukan	mati	tidak terdeteksi sinyal
tidak diberi lapisan	musik	kedip	masih ada yang tidak terdeteksi sebagai sinyal
	suara manusia	mati	tidak terdeteksi sebagai sinyal
	tanpa masukan	mati	tidak terdeteksi sinyal

Pengujian dilakukan tiga kali dengan kondisi yang berbeda. Kondisi pengujian pertama adalah *microphone* diberi lapisan yang berbentuk corong. Ketika musik diperdengarkan indikator led menyala, menandakan bahwa terdeteksi adanya sinyal. Namun ketika masukan berasal dari suara manusia, kadang-kadang masih terdeteksi sebagai sinyal sehingga indikator led berkedip. Kondisi pengujian kedua adalah *microphone* diberi lapisan tipis hanya pada ujungnya saja. Ketika musik diperdengarkan, indikator led menyala sebagai tanda terdeteksi sinyal, dan ketika masukan hanya berupa suara manusia indikator led mati menandakan tidak terdeteksi sinyal. Pengujian ketiga dilakukan tanpa memberi lapisan apa pun pada *microphone*. Ketika musik diperdengarkan, kadang-kadang tidak terdeteksi sebagai sinyal, ditandai dengan indikator led yang berkedip. Sedangkan ketika masukan hanya berupa suara manusia led indikator mati. Dari ketiga pengujian jika sistem tidak diberi masukan apa pun, maka led indikator mati menandakan bahwa tidak ada sinyal yang terdeteksi. Sehingga dapat disimpulkan bahwa sistem dapat mendeteksi adanya sinyal yang berasal dari suara musik ketika musik diperdengarkan, dan sistem tidak mendeteksi adanya sinyal dari suara musik ketika musik dimatikan.

## BAB VI

### KESIMPULAN DAN SARAN

#### 6.1 Kesimpulan

Berdasarkan hasil pengujian tiap blok dan pengujian sistem secara keseluruhan yang telah dilakukan dalam Bab V, dapat disimpulkan beberapa hal sebagai berikut:

1. Sistem yang telah dirancang dan dibuat dapat mendeteksi adanya suara musik dan dapat diaplikasikan dalam Kontes Robot Seni Indonesia
2. Filter IIR *Butterworth* orde empat dengan frekuensi ADC sampling 20 kHz dapat diimplementasikan dalam sistem ini. Pemilihan frekuensi sampling haruslah tepat sehingga sistem tidak akan *overlapping* dalam mengolah data
3. Respon filter ideal tidak akan pernah dapat dipenuhi, yang dapat dilakukan adalah pemilihan spesifikasi yang mendekati kondisi ideal
4. *Microphone* dapat digunakan sebagai sensor suara dalam sistem pendeteksi musik dan dapat diaplikasikan dalam Kontes Robot Seni Indonesia
5. Mikrokontroler ATmega32 dapat digunakan sebagai pengolah data sekaligus sebagai pengambil keputusan dalam sistem pendeteksi musik ini.

#### 6.2 Saran

Meskipun alat ini telah bekerja dengan baik sesuai dengan spesifikasi yang telah ditentukan, namun ada beberapa hal yang perlu dikembangkan agar diperoleh sistem yang lebih baik, antara lain:

1. Disarankan menggunakan mikrokontroler dengan tingkat frekuensi kerja yang lebih tinggi dan memori yang cukup besar untuk mengoptimalkan pengolahan sistem digital
2. Pemilihan sensor suara perlu diperhatikan untuk memperoleh respon yang lebih baik dikarenakan banyaknya kekurangan yang terdapat pada *microphone* dinamik



## DAFTAR PUSTAKA

Atmel. 2006. ATmega32/ATmega32L, 8-bit AVR Microcontroller with 32 kbytes in System Programmable Flash. [http://atmel.com/dyn/resources/prod\\_documents/doc2502.pdf](http://atmel.com/dyn/resources/prod_documents/doc2502.pdf), diakses tanggal: 10 Agustus 2009.

Atmel. AVR223: *Digital Filters with AVR*, <http://www.atmel.com>, diakses tanggal: 20 Oktober 2009.

DIKTI. 2008. Panduan Kontes Robot Seni Indonesia 2009. Jakarta: DIKTI

DIKTI. 2009. Panduan Kontes Robot Seni Indonesia 2010. Jakarta: DIKTI

Kuo, Shen M. 2006. *Real Time Digital Signal Processing*. West Sussex: John Wiley

Lee Edward A. and Varaiya Pravin, 2000. *Structur and Intrepretation signal and sistem*.California: EEC Science.

Malvino, Albert Paul. 1992. Prinsip-prinsip Elektronik. Edisi Kedua, Terjemahan Hanapi Gunawan. Jakarta : Erlangga.

Malvino, Albert Paul. 2004. Prinsip-prinsip Elektronika. Edisi Kedua, Terjemahan Joko Santoso. Jakarta: Salemba Teknika.

Shenoi, B.A. 2006. *Introduction to Digital Signal Processing and Filter Design*. New Jersey: John Wiley and Sons, Inc.

Stein, Jonathan.2000. *Digital Signal Processing: A Computer Science Perspective*. New Jersey: John Wiley and Sons, Inc.

Wardhana, L. 2006. Belajar Sendiri Mikrokontroler AVR Seri ATMega8535 Simulasi, Hardware dan Aplikasi. Yogyakarta: Andi.