

BAB III

METODE PENELITIAN

Penyusunan metode yang digunakan dalam penelitian bersifat aplikatif, yaitu perancangan dan realisasi alat untuk mendeteksi pola irama musik pengiring yang akan ditampilkan pada komputer sebagai panduan *programmer* tim KRSTI dalam membuat gerakan tarian robot, mengacu pada rumusan masalah. Diperlukan beberapa langkah dalam implementasi sistem antara lain penentuan spesifikasi alat, perancangan dan pembuatan alat, pengujian sistem serta pengambilan kesimpulan berdasarkan hasil pengujian untuk mendapatkan sistem yang sesuai dengan perancangan dan tujuan yang telah dibuat.

3.1 Penentuan Spesifikasi Alat

Sebelum melakukan perencanaan dan pembuatan alat, terlebih dahulu ditentukan spesifikasi alat yang akan dibuat. Spesifikasi alat ditetapkan sebagai acuan untuk mempermudah dalam perancangan dan pembuatan alat. Adapun spesifikasi alat yang akan direalisasikan adalah sebagai berikut:

1. Mikrokontroler yang digunakan adalah ATmega8.
2. Resolusi ADC yang dipakai untuk proses kuantisasi sebesar 10 bit.
3. Sumber musik yang diproses berasal dari *Bluetooth Audio Receiver* tipe TS-BT35A08.
4. Op-Amp yang digunakan pada rangkaian sistem adalah LM358.
5. *USB to TTL converter* yang digunakan bertipe CP2102.
6. Sumber tegangan menggunakan *Adaptor 5,4 V 1 A*.
7. Musik pengiring yang dipakai merupakan musik pengiring KRSTI 2016: Robot Penari Topeng Betawi

3.2 Perancangan dan Perealisasian Alat

Perancangan dan pembuatan alat dalam penelitian ini dibagi kedalam dua bagian yaitu perancangan *hardware* dan *software*. Perancangan *hardware* terbagi atas perancangan sistem dan perancangan elektrik. Perancangan *software* berupa perancangan algoritma dari sistem yang telah dirancang.

3.2.1 Perancangan dan Pembuatan Perangkat Keras (*Hardware*)

Secara garis besar perancangan dan pembuatan perangkat keras dibagi menjadi beberapa bagian yaitu perancangan diagram blok sistem keseluruhan dan perancangan elektrik sistem. Perancangan elektrik ditekankan pada perancangan rangkaian filter, rangkai-

an antarmuka mikrokontroler serta rangkaian *minimum system* ATmega8. Papan rangkaian tercetak (PCB) dirancang dengan menggunakan perangkat lunak EAGLE (*Easily Applicable Graphical Layout Editor*). Sementara untuk rangkaian *Bluetooth Audio Receiver* dan USB to TTL digunakan modul jadi.

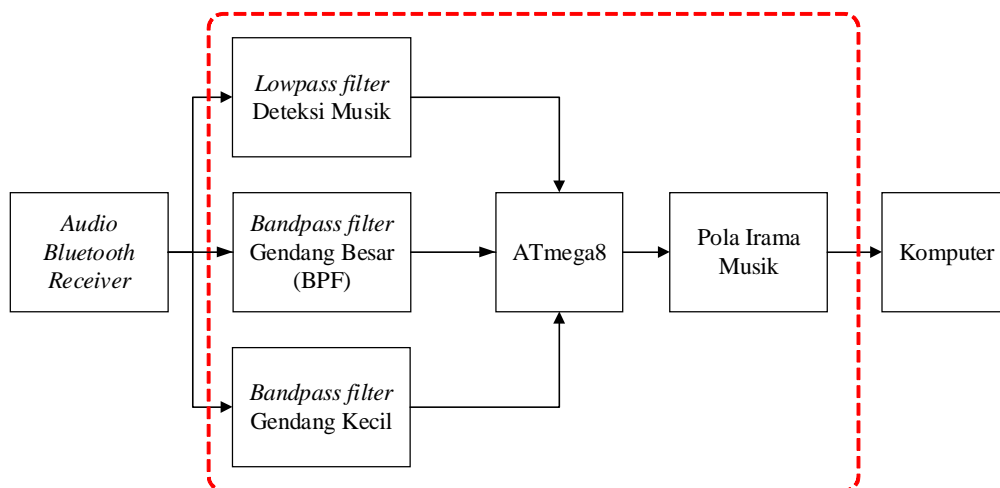
3.2.1.1 Perancangan Diagram Blok Sistem Keseluruhan

Perancangan digambarkan kedalam blok diagram sistem keseluruhan yang merupakan perancangan awal sistem secara umum. Perancangan ini merupakan gambaran umum dari sistem yang dibuat yang ditunjukkan pada Gambar 3.1.



Gambar 3.1 Diagram blok sistem keseluruhan

Secara keseluruhan, prinsip kerja sistem dimulai dengan *pairing* antar *bluetooth transmitter* yang terdapat pada komputer dan modul *bluetooth receiver* TS-BT35A08. Musik pengiring dikirim secara nirkabel menggunakan *audio bluetooth transmitter* dan diterima oleh *audio bluetooth receiver*. Suara musik yang diterima oleh *receiver* diproses kedalam sistem utama yang ditunjukkan pada Gambar 3.2. Data pemrosesan pada sistem utama dikirimkan ke komputer yang nantinya dijadikan sebagai acuan pembuatan gerakan tari.



Gambar 3.2 Diagram blok sistem utama

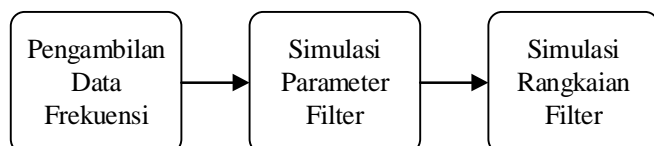
Sinyal suara musik yang diterima oleh *audio bluetooth receiver* diteruskan ke rangkaian *filter* melalui *audio jack* 3,5 mm yang tersedia pada modul TS-BT35A08. Pemrosesan dibagi menjadi dua bagian yaitu *lowpass filter* (LPF) dan *bandpass filter* (BPF). Rangkaian *filter* LPF digunakan sebagai pendeteksi ada tidaknya musik pengiring. Deteksi musik pengiring digunakan sebagai penanda sistem mulai aktif dan mulai melakukan perhitungan waktu.

Selain itu deteksi musik berguna untuk diterapkan langsung pada robot sebagai penentu robot harus menari saat musik dimainkan dan berhenti menari saat musik dihentikan.

Rangkaian *filter* BPF digunakan untuk menapis alat musik dominan pada musik pengiring. Pada penelitian ini dipilih gendang tipak karena pada musik pengiring KRSTI 2016 alat ini memiliki suara yang keras dan memiliki ritme yang bervariasi sebagai penanda mulainya bait tertentu. Proses pemfilteran gendang tipak dibagi menjadi dua yaitu gendang besar dan gendang kecil yang terdiri atas rangkaian *filter* BPF1 dan BPF2. Pembagian gendang besar dan kecil bertujuan untuk memperjelas tempo musik yang ditunjukkan dari ketukan gendang, agar mempermudah *programmer* dalam mengolah musik sehingga pola irama musik pengiring dapat diamati dengan mudah. Sinyal hasil *filter* diolah oleh mikrokontroler ATmega8 menjadi pola musik pengiring yang dikirimkan dan ditampilkan ke komputer sebagai panduan *programmer* dalam pembuatan gerakan tari robot.

3.2.1.2 Pengambilan Data Musik dan Simulasi

Pengambilan data musik diperlukan sebagai panduan untuk merancang sistem pendeteksi pola irama musik pengiring, sehingga *programmer* mampu mendesain gerakan tari yang sinkron dengan ritme musik pengiring. Penentuan dilakukan dengan menggunakan beberapa perangkat lunak yaitu *frequency analysis* dan *scientific filter tool* pada Adobe Audition untuk menentukan nilai frekuensi suara gendang tipak dan Tina-TI sebagai simulator. Proses pengambilan data musik ditunjukkan pada Gambar 3.3.



Gambar 3.3 Proses pengambilan data musik

Penentuan rentang frekuensi yang dihasilkan oleh gendang tipak didapat dengan mengamati respons *filter* pada waktu saat gendang tipak terdengar. Pemeriksaan adanya suara gendang tipak dilakukan dengan mendengar langsung musik pengiring dan/ atau mengamati dengan *spectral frequency display* pada Adobe Audition. Setelah didapatkan titik saat gendang tipak terdengar, respons diamati menggunakan *frequency analysis*. Untuk menampilkan hasil gendang besar dan gendang kecil bersamaan, pengamatan dilakukan bergantian dengan menyimpan data gendang besar menggunakan *tool hold* yang tersedia sebelum mengamati gendang kecil.

Hasil pengamatan *frequency analysis* dijadikan sebagai panduan awal dalam menentukan frekuensi *cutoff* dari *filter* yang akan dirancang. Adapun frekuensi yang dipilih merupakan *bandwidth* dengan atenuasi paling kecil (amplitudonya paling tinggi). Rentang

frekuensi yang didapat digunakan untuk melakukan simulasi *lowpass* dan *bandpass filter* menggunakan *scientific filter* pada Adobe Audition. Pada *scientific filter* dipilih mode *filter* (*lowpass*, *highpass*, *bandpass*, *notch filter*), jenis *filter* (*bessel*, *butterworth*, *chebysev*, *elliptic*), orde *filter* serta penguatan yang akan digunakan. Kemudian nilai frekuensi *cutoff* dimasukkan pada kolom *Cutoff* dengan memasukkan nilai sesuai hasil pengamatan sebelumnya, lalu *apply*. Hasil simulasi *filter* diamati sinyal keluarannya serta diuji langsung menggunakan media *speaker* atau *headphone*

Hasil *filter* musik pengiring menggunakan *scientific filter* dijadikan sebagai referensi dalam perancangan rangkaian. Perancangan rangkaian *filter* dilakukan dengan melakukan perhitungan berdasarkan parameter- parameter yang dimasukkan pada *scientific filter* sesuai topologi yang digunakan. Setelah didapatkan nilai- nilai komponen hasil dari perhitungan, dilakukan simulasi rangkaian menggunakan perangkat lunak Tina-TI. Sinyal keluaran dari rangkaian *filter* pada Tina-TI dibandingkan dengan sinyal keluaran hasil *scientific filter* untuk membuktikan bahwa rangkaian sudah sesuai atau belum.

3.2.1.3 Perancangan Rangkaian Filter

Beberapa parameter yang didapatkan pada simulasi digunakan dalam perancangan rangkaian *filter*. Rangkaian *filter* yang dirancang terdiri atas tiga buah *filter*, yaitu satu buah *lowpass filter* dan dua buah *bandpass filter*. Rangkaian *filter* yang pertama berupa *lowpass filter 2nd order*. Filter ini digunakan untuk mendeteksi keadaan ada tidaknya musik pengiring. Deteksi musik berguna sebagai penanda bahwa sistem pendeteksi musik pengiring mulai bekerja. Selain itu deteksi musik juga berguna saat perlombaan yaitu sebagai penanda robot harus menari ketika musik pengiring dimainkan atau harus berhenti menari ketika musik pengiring dihentikan. Penentuan frekuensi *cutoff* dilakukan dengan menandai respon musik pengiring pada detik tertentu secara acak menggunakan *frequency analysis* pada Adobe Audition dimana “frekuensi *cutoff* dari suatu penapis sebenarnya adalah suatu frekuensi pada saat penguatan tegangannya drop atau turun menjadi 0,707 atau $\frac{1}{\sqrt{2}}$ kali dari penguatan *passband*-nya” (Putra, 2002, p.29).

Berdasarkan hasil *scanning* acak, didapatkan bahwa rerata frekuensi tertinggi pada musik pengiring KRSTI 2016 sebesar 6000 sampai dengan 8000 Hz. Ditentukan frekuensi 7000 Hz sebagai frekuensi *cutoff* untuk penanda ada tidaknya musik pengiring. Rangkaian *filter* yang digunakan berupa *lowpass filter* orde 2 menggunakan topologi *Sallen- Key* penguatan unitas dengan menetapkan nilai komponen tertentu. Ditentukan nilai kapasitor C_1

dan C_2 menyesuaikan standar nilai kapasitor, menjadi $C_1 = C_2 = C = 10 \text{ nF}$. Nilai R_1 dan R_2 dapat ditentukan dengan menggunakan Persamaan (2-5) yaitu:

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C^2}}$$

$$7000 = \frac{1}{2 \times 3,1416 \sqrt{R_1 R_2 \times (10^{-8})^2}}$$

$$\sqrt{R_1 R_2} = \frac{1}{6,2832 \times 7000 \times 10^{-8}}$$

$$R_1 R_2 = \left(\frac{1}{4,3982 \times 10^{-4}} \right)^2$$

$$R_1 R_2 = \frac{1}{19,3444 \times 10^{-8}}$$

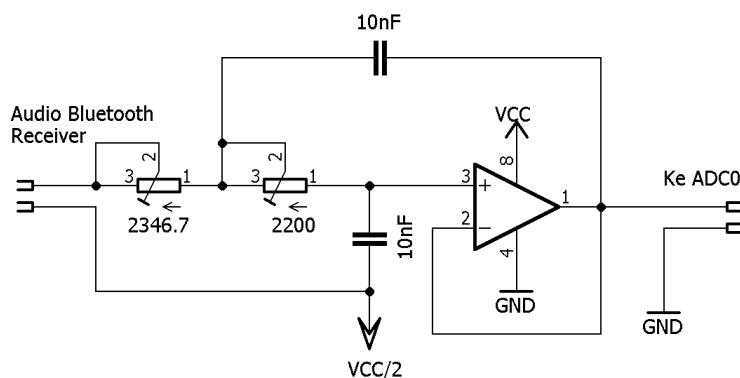
Dengan menentukan nilai resistansi pada resistor R_2 sebesar 2200 Ω , maka nilai R_1 dapat didapatkan:

$$R_1 = \frac{1}{19,3444 \times 2200 \times 10^{-8}}$$

$$R_1 = \frac{10^4}{4,2558}$$

$$R_1 = 2349,75 \Omega$$

Setelah seluruh komponen diketahui nilainya, selanjutnya dilakukan perancangan rangkaian *lowpass filter* ke dalam skematik. Nilai komponen yang sudah didapatkan yaitu $R_1 = 2349,75 \Omega$, $R_2 = 2200 \Omega$ dan $C_1 = C_2 = 10 \text{ nF}$. Rangkaian *lowpass filter* ditunjukkan pada Gambar 3.4.



Gambar 3.4 Rangkaian 2nd order lowpass filter topologi Sallen- Key

Rangkaian *filter* kedua dan ketiga merupakan rangkaian *bandpass filter* yang untuk menyaring suara gendang tipak: gendang besar dan gendang kecil. Penentuan spesifikasi *filter* dilakukan dengan mengamati respons ketika terdengar gendang tipak menggunakan

frequency analysis tool pada Adobe Audition. Dari hasil pengamatan, ditentukan spesifikasi *bandpass filter* sebagai berikut:

1) Desain *filter* gendang besar memiliki spesifikasi berikut ini:

- Frekuensi *cutoff* atas (f_H) = 140 Hz
- Frekuensi *cutoff* bawah (f_L) = 80 Hz
- Atenuasi minimum 20dB pada $f_2 = 250$ Hz
- Penguatan (A_0) = 1
- Topologi yang digunakan = *Multiple Feedback*

2) Desain *filter* gendang kecil memiliki spesifikasi:

- Frekuensi *cutoff* atas (f_H) = 350 Hz
- Frekuensi *cutoff* bawah (f_L) = 250 Hz
- Atenuasi minimum 20dB pada $f_2 = 500$ Hz
- Penguatan (A_0) = 1
- Topologi yang digunakan = *Multiple Feedback*

Sebelum menentukan nilai komponen, terlebih dahulu dicari nilai frekuensi tengah *filter* gendang besar menggunakan Persamaan (2-9):

$$f_o = \sqrt{f_L f_H}$$

$$f_o = \sqrt{80 \times 140} = 105,83 \text{ Hz}$$

Nilai f_o digunakan untuk menghitung Q_{BP} dan f_1 menggunakan persamaan (2-10) dan (2-12):

$$Q_{BP} = \frac{f_o}{BW}$$

$$Q_{BP} = \frac{105,83}{140 - 80} = 1,7638$$

$$f_1 f_2 = f_o^2$$

$$f_1 = \frac{105,83^2}{250} = 44,8 \text{ Hz}$$

Selanjutnya Persamaan (2-13) digunakan untuk mendapatkan faktor ketajaman *bandpass*:

$$A_s = \frac{\text{stopband bandwidth}}{\text{passband bandwidth}}$$

$$A_s = \frac{250 - 44,8 \text{ Hz}}{140 - 80 \text{ Hz}} = 3,42$$

Dengan Persamaan (2-9), (2-10), (2-12) dan (2-13), untuk spesifikasi gendang kecil didapatkan nilai- nilai sebagai berikut:

$$f_o = 295,8 \text{ Hz}$$

$$Q_{BP} = 2,958$$

$$f_1 = 175 \text{ Hz}$$

$$A_s = 3,25$$

Berdasarkan karakteristik atenuasi Gambar 2.11 didapatkan bahwa *Butterworth filter* orde kedua mampu meredam 20dB lebih dalam rasio 3,42: 1 untuk gendang besar dan 3,25:1 untuk gendang kecil. Sehingga kedua *filter* menggunakan *pole* ternormalisasi yang sama. Dari tabel 2.2 didapatkan bahwa *pole* ternormalisasi *Butterworth filter* orde dua adalah:

$$-7071 \pm j7071$$

Setelah didapatkan keempat parameter yaitu α , β , Q_{BP} dan f_o , selanjutnya dilakukan perhitungan menggunakan Persamaan (2-21) hingga (2-29) untuk menentukan nilai Q dan frekuensi resonansi dari gendang besar.

$$\begin{aligned} C &= \alpha^2 + \beta^2 \\ &= 0,7071^2 + 0,7071^2 = 1 \end{aligned}$$

$$\begin{aligned} D &= \frac{2\alpha}{Q_{BP}} \\ &= \frac{2 \times 0,7071}{1,7638} = 0,8018 \end{aligned}$$

$$\begin{aligned} E &= \frac{C}{Q_{BP}^2} + 4 \\ &= \frac{1}{1,7638^2} + 4 = 4,3214 \end{aligned}$$

$$\begin{aligned} G &= \sqrt{E^2 - 4D^2} \\ &= \sqrt{4,3214^2 - 4 \times 0,8018^2} = 4,0129 \end{aligned}$$

$$\begin{aligned} Q &= \sqrt{\frac{E + G}{2D^2}} \\ &= \sqrt{\frac{4,3214 + 4,0129}{2 \times 0,8018^2}} = 2,5461 \end{aligned}$$

$$\begin{aligned} M &= \frac{\alpha Q}{Q_{BP}} \\ &= \frac{0,7071 \times 2,5461}{1,7638} = 1,0207 \end{aligned}$$

$$W = M + \sqrt{M^2 - 1}$$

$$= 1,0207 + \sqrt{1,0207^2 - 1} = 1,2251$$

$$f_{ra} = \frac{f_o}{W}$$

$$= \frac{105,83\text{Hz}}{1,2551} = 86,38\text{Hz}$$

$$f_{rb} = Wf_o$$

$$= 1,2551 \times 105,83\text{Hz} = 129,65\text{Hz}$$

Setelah didapatkan Q , f_{ra} dan f_{rb} , nilai- nilai tersebut diimplementasikan kedalam rangkaian kaskade dua *bandpass filter* topologi *Multiple Feedback* dengan frekuensi resonansi f_{ra} pada *filter* bagian pertama dan frekuensi resonansi f_{rb} pada *filter* bagian kedua dengan nilai Q yang sama. Penguatan *bandpass filter* bagian pertama didapatkan dari Persamaan (2-31) yaitu:

$$A_r = A_0 \sqrt{1 + Q^2 \left(\frac{f_o}{f_r} - \frac{f_r}{f_o} \right)^2}$$

$$= 1 \times \sqrt{1 + 2,5461^2 \left(\frac{105,83}{86,38} - \frac{86,38}{105,83} \right)^2}$$

$$= 1,4435$$

Dengan menentukan nilai kapasitor C sebesar 220nF, maka R_2 , R_{1a} dan R_{1b} dapat ditentukan melalui Persamaan (2-15), (2-19) dan (2-20):

$$R_2 = \frac{Q}{\pi f_r C}$$

$$= \frac{2,5461}{3,1416 \times 86,38 \times 2,2 \times 10^{-7}}$$

$$= \frac{2,5461 \times 10^5}{5,9705}$$

$$= 42,664\text{k}\Omega$$

$$R_{1a} = \frac{R_2}{2A_r}$$

$$= \frac{42644,1}{2 \times 1,4435}$$

$$= 14,771\text{k}\Omega$$

$$\begin{aligned}
 R_{1b} &= \frac{R_2 / 2}{2Q^2 - A_r} \\
 &= \frac{42644,1 / 2}{2 \times 2,5461^2 - 1,4435} \\
 &= \frac{21322,05}{11,521} \\
 &= 1,851k\Omega
 \end{aligned}$$

Pada *filter* bagian kedua, dengan nilai frekuensi resonansi f_{rb} sebesar 129,65 Hz, didapatkan

$$A_r = 1,4435$$

Dengan menyamakan nilai kapasitor C dengan kapasitor pada filter bagian sebelumnya, yaitu sebesar 220nF, maka dengan Persamaan (2-15), (2-19) dan (2-20) didapatkan:

$$R_2 = 28,413k\Omega$$

$$R_{1a} = 9,842k\Omega$$

$$R_{1b} = 1,233k\Omega$$

Pada perhitungan gendang kecil, parameter yang didapat pada perhitungan sebelumnya adalah sebagai berikut:

$$=0,7071 \quad =0,7071 \quad Q_{BP}= 2,958 \quad f_o=295,8Hz$$

Keempat parameter gendang kecil tersebut, dengan langkah perhitungan yang sama dimasukkan ke dalam Persamaan (2-21) hingga (2-29) maka diperoleh:

$$C = 1 \quad M = 1,0072$$

$$D = 0,4781 \quad W = 1,1276$$

$$E = 4,1143 \quad f_{ra} = 262,33Hz$$

$$G = 4,0016 \quad f_{rb} = 333,55Hz$$

$$Q = 4,2135$$

Untuk setiap bagian *bandpass filter* gendang kecil didapatkan:

Bagian pertama: $f_r = 262,33Hz$

$$Q = 4,2135$$

$$A_r = 1,4244$$

Bagian kedua: $f_r = 333,55Hz$

$$Q = 4,2135$$

$$A_r = 1,4244$$

Dengan menyamakan seluruh nilai kapasitor C pada semua bagian dengan nilai 220nF , maka didapatkan keenam nilai resistor pada kedua bagian, dengan:

Bagian pertama: $R_2 = 23,239\text{ k}\Omega$

$$R_{1a} = 8,157\text{ k}\Omega$$

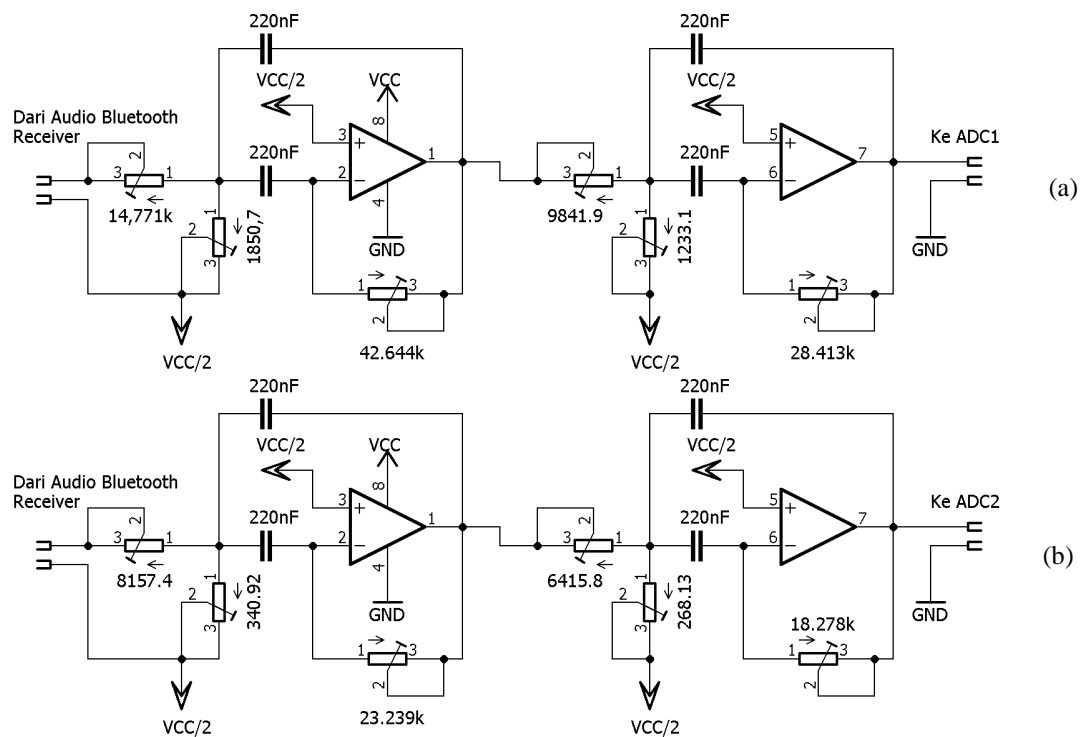
$$R_{1b} = 340,92\Omega$$

Bagian kedua: $R_2 = 18,278\text{ k}\Omega$

$$R_{1a} = 6,416\text{ k}\Omega$$

$$R_{1b} = 268,13\Omega$$

Setelah didapatkan seluruh nilai komponen, maka nilai- nilai tersebut dimasukkan ke dalam perancangan skematik rangkaian *bandpass filter*, seperti ditunjukkan pada Gambar 3.5.



Gambar 3.5 Rangkaian 2^{nd} order Butterworth bandpass filter topologi Multiple Feedback: (a) Filter gendang besar dan (b) Filter gendang kecil

3.2.1.4 Perancangan Antarmuka Mikrokontroler Utama

Mikrokontroler merupakan pengontrol utama sistem. Beberapa *pin* digunakan untuk menghubungkan mikrokontroler dengan perangkat- perangkat lain seperti rangkaian *filter*, LED dan komputer seperti ditunjukkan pada Gambar 3.6. Berikut merupakan *pin* yang digunakan dalam perancangan antarmuka mikrokontroler:

PC0 sebagai *input* ADC 1

PC1 sebagai *input* ADC 2

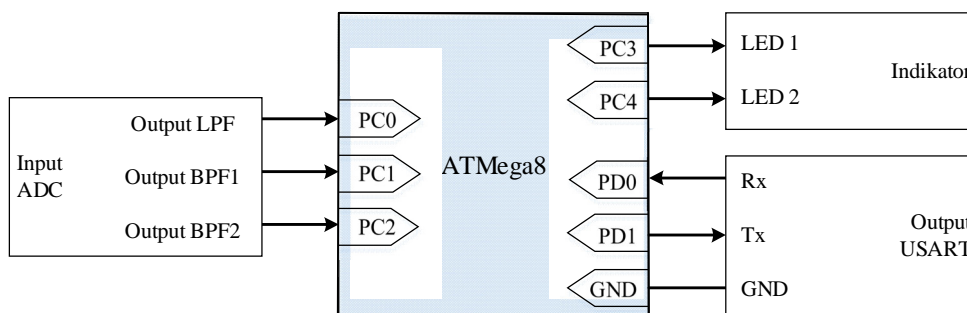
PC2 sebagai *input* ADC 3

PC3 sebagai indikator sistem aktif

PC4 sebagai indikator pola irama musik

PD0 sebagai Rx USART

PD1 sebagai Tx USART



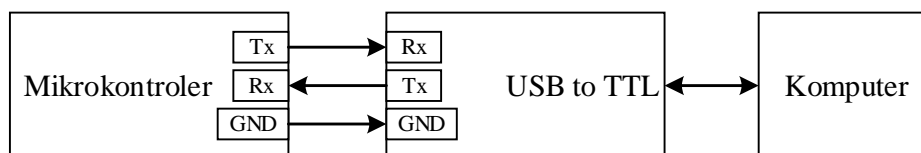
Gambar 3.6 Diagram blok antarmuka mikrokontroler utama

Mikrokontroler mengolah tiga *input* berupa sinyal analog yang berasal dari rangkaian *filter*, yaitu satu buah LPF dan dua buah BPF. Masing- masing *input* terhubung ke *pin* ADC untuk mengonversi sinyal analog *input* menjadi sinyal *digital*, dengan konfigurasi: rangkaian LPF terhubung ke PC0, rangkaian BPF pertama terhubung ke PC1 dan rangkaian BPF kedua terhubung ke PC2. Pembacaan sinyal analog pada *pin* PC0 sampai dengan PC2 menggunakan resolusi data 10 bit yang merupakan resolusi maksimal ADC yang terdapat pada ATmega8.

Hasil pemrosesan mikrokontroler dihubungkan pada tiga buah *pinoutput* yang memiliki fungsi berbeda. *Output* pertama yaitu PC3 yang terhubung dengan LED pertama yang berfungsi sebagai indikator apakah sistem sudah aktif atau belum. *Output* kedua yaitu PC4 yang terhubung dengan LED kedua sebagai penanda adanya pola irama musik pada waktu tertentu. *Output* yang ketiga merupakan data ketukan gendang serta data pola musik berupa data *serial* yang dikirim menggunakan USART. Komunikasi USART yang tersedia pada ATmega8 adalah PD0 sebagai *Receiver* Rx dan PD1 sebagai *Transmitter* Tx.

3.2.1.5 Perancangan USB to TTL

Data hasil pemrosesan sitem pada mikrokontroler utama dikirimkan ke komputer. Terdapat perbedaan *logic voltage level* antara USB pada komputer dengan TTL pada mikrokontroler. Untuk mengonversi *logic voltage level* dibutuhkan antarmuka berupa modul USB to TTL. Perangkat ini mampu menghubungkan komunikasi antara mikrokontroler dengan komputer dengan mengubah *logic level* dari TTL berupa Tx (*transmitter*) dan Rx (*receiver*) menjadi USB berupa D- dan D+, begitu pula sebaliknya. Diagram koneksi modul USB to TTL dapat dilihat pada Gambar 3.7.



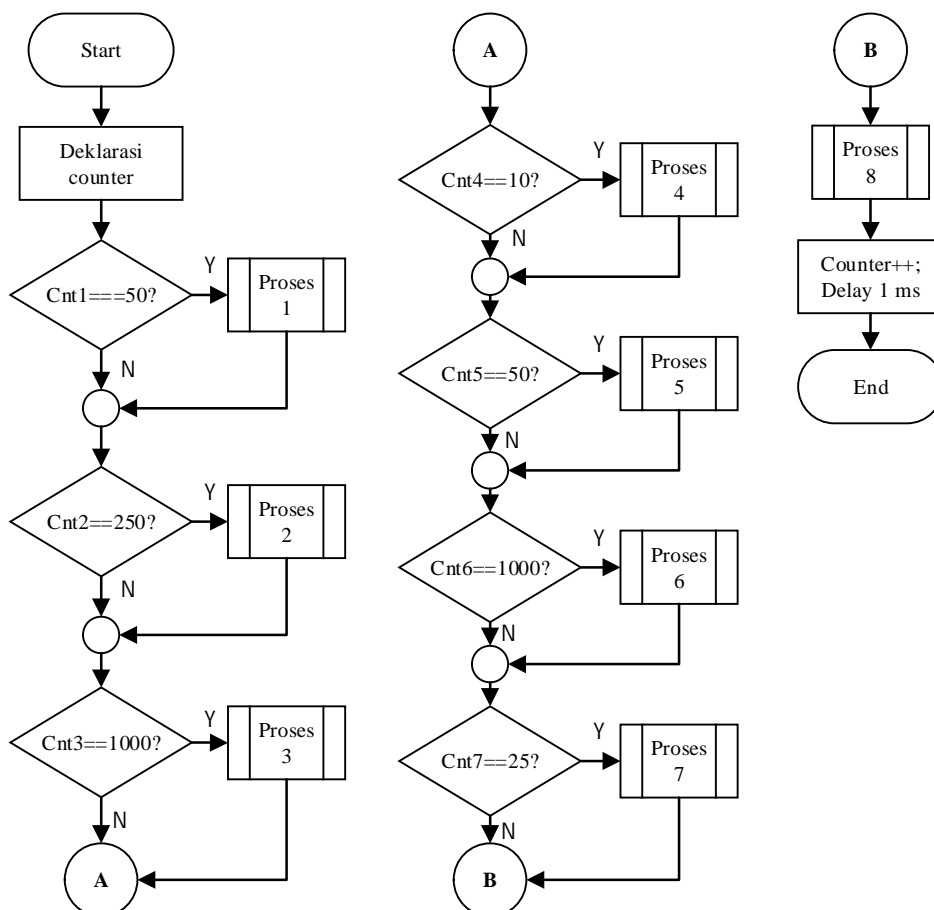
Gambar 3.7 Diagram koneksi modul USB to TTL antara mikrokontroler dengan komputer

Data *output* dari mikrokontroler dikirimkan secara *serial* menggunakan *pin* USART yang tersedia pada mikrokontroler ATmega8 yaitu PD0 (*pinout* Rx) dan PD1 (*pinout* Tx). Koneksi *transmitter/ receiver* memiliki *pin* yang berbeda untuk komunikasi, yaitu Tx sebagai pengirim data dan Rx sebagai penerima data. Sehingga koneksi *pin* Tx mikrokontroler harus terhubung dengan *pin* Rx modul USB to TTL dan *pin* Rx mikrokontroler harus terhubung dengan *pin* Tx modul USB to TTL. Agar komunikasi dapat diteruskan ke komputer, dibutuhkan *pin ground* untuk menyamakan referensi tegangan.

3.2.2 Perancangan dan Pembuatan Perangkat Lunak (*Software*)

Perancangan perangkat lunak diawali dengan pembuatan diagram alir (*flowchart*) untuk setiap sub-fungsi. Masing- masing subfungsi menjelaskan proses mulai dari pengambilan data *input*, pengolahan data *input* hingga hasil pengolahan yang akan ditampilkan. Pembuatan diagram alir ini dijadikan sebagai panduan untuk mengimplementasikan algoritma kedalam kode program. Secara garis besar perancangan diagram alir dibagi atas empat bagian yaitu deteksi musik, deteksi pola irama musik pengiring, indikator musik dan pengiriman data musik.

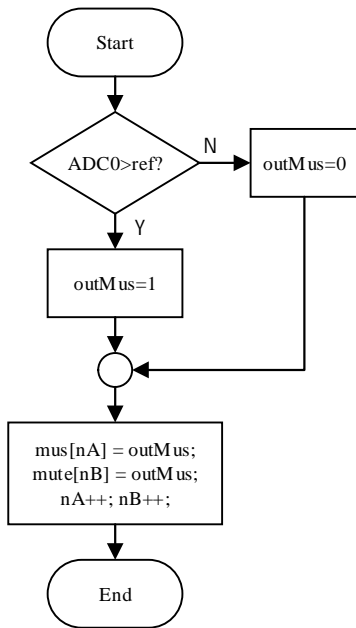
Fungsi utama menggabungkan seluruh subfungsi yang digunakan. Setiap proses diatur oleh *counter* yang memiliki *delay* berbeda disesuaikan dengan kegunaan masing- masing subfungsi. Proses 1 sampai dengan Proses 3 merupakan subfungsi yang digunakan untuk deteksi musik pengiring, dimana Proses 1 merupakan subfungsi untuk *sampling* data suara dari ADC0, Proses 2 merupakan subfungsi deteksi musik dari keadaan tidak ada suara dan Proses 3 merupakan subfungsi deteksi *mute* pada musik pengiring. Proses 4 sampai dengan Proses 6 merupakan subfungsi untuk deteksi pola irama musik pengiring, dengan Proses 4 sebagai subfungsi untuk *sampling* data gendang tipak, Proses 5 sebagai subfungsi deteksi gendang tipak dan Proses 6 sebagai subfungsi deteksi pola musik. Proses 7 merupakan subfungsi pengiriman data untuk mengirimkan data keluaran ke komputer serta Proses 8 merupakan subfungsi indikator sistem. Setiap proses dirancang secara terpisah bertujuan agar proses dapat dilakukan secara serempak, sehingga setiap subfungsi dapat mengambil data yang telah diproses oleh subfungsi lain secara langsung. Diagram alir fungsi utama ditampilkan pada Gambar 3.8.



Gambar 3.8 Diagram alir program utama

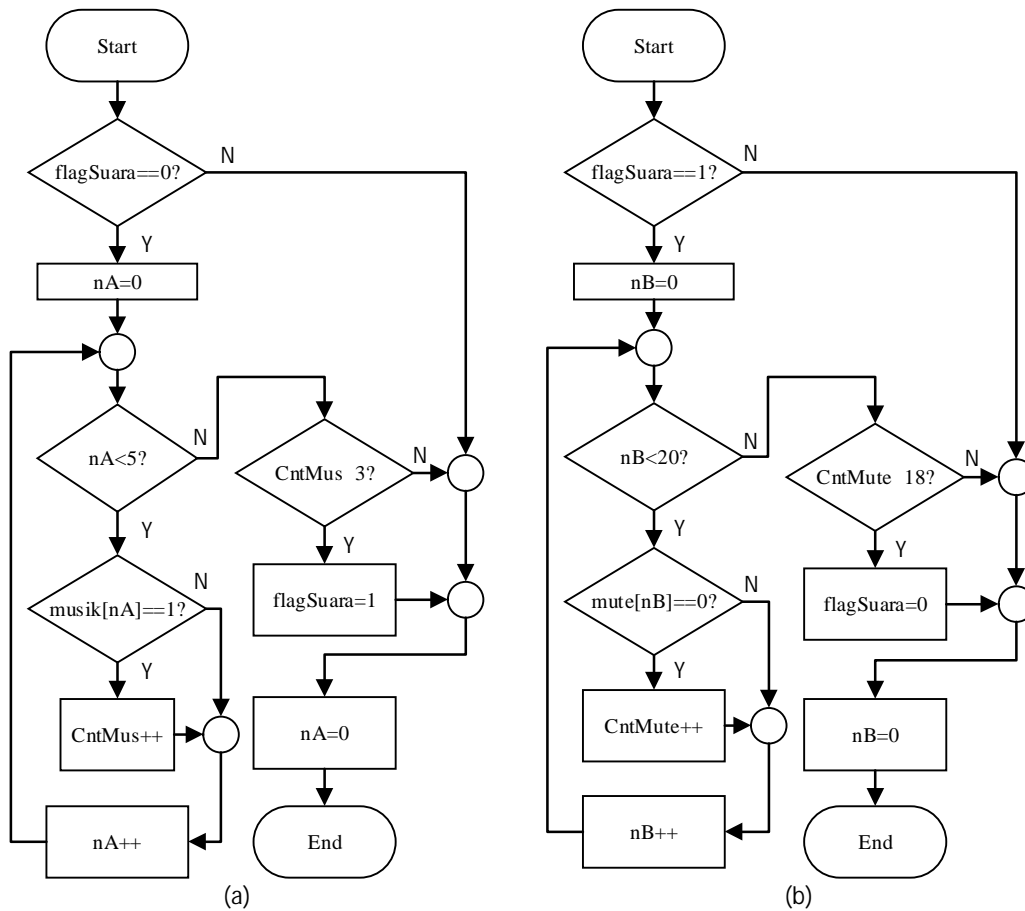
3.2.2.1 Deteksi Musik sebagai Awal Aktivasi Sistem

Sistem mulai membaca data ADC0 ketika musik pengiring dimainkan. Deteksi musik diawali dengan proses *sampling* nilai ADC0 pada subfungsi Proses 1 dengan periode pembacaan 50 ms. Nilai ADC0 dibandingkan dengan nilai ambang batas yang didapatkan dengan menguji nilai ADC saat tidak adanya musik pengiring, menghasilkan keluaran *digital*. Keluaran dimasukkan ke dalam *array* sebagai parameter penentuan deteksi musik pada Proses 2 dan Proses 3. Deteksi musik dan deteksi *mute* memiliki periode pemrosesan yang berbeda sehingga dibuat dua variabel *array* berbeda. Diagram alir proses *sampling* ADC musik pengiring ditunjukkan pada Gambar 3.9.



Gambar 3.9 Diagram alir proses *sampling* ADC musik pengiring

Deteksi musik dibagi menjadi dua bagian, yaitu subfungsi deteksi musik dengan kondisi musik dimainkan dari keadaan diam serta subfungsi deteksi *mute* dengan kondisi musik pengiring berhenti maupun *dimute*, yang ditunjukkan pada Gambar 3.10.

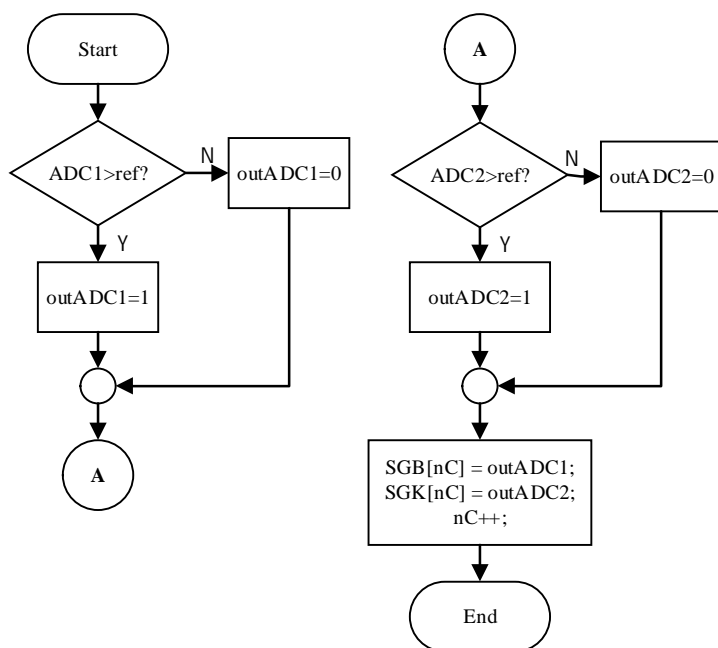


Gambar 3.10 Diagram alir deteksi musik: (a) deteksi suara & (b) deteksi *mute*

Data hasil *sampling* yang masuk ke dalam variabel *array* dipanggil ke dalam subfungsi deteksi menggunakan iterasi *for*. Iterasi berfungsi menghitung jumlah keluaran berlogika tinggi selama periode 250 ms pada subfungsi deteksi suara dan menghitung jumlah keluaran logika rendah selama periode 1000 ms pada subfungsi deteksi *mute*. Batas minimum jumlah keluaran logika menjadi penentu *flag* aktif atau tidak. Variabel *flagSuara* merupakan penentu yang digunakan beberapa subfungsi untuk melakukan proses atau melewatinya.

3.2.2.2 Deteksi *Gendang Tipak* sebagai Parameter Penentuan Pola Irama Musik

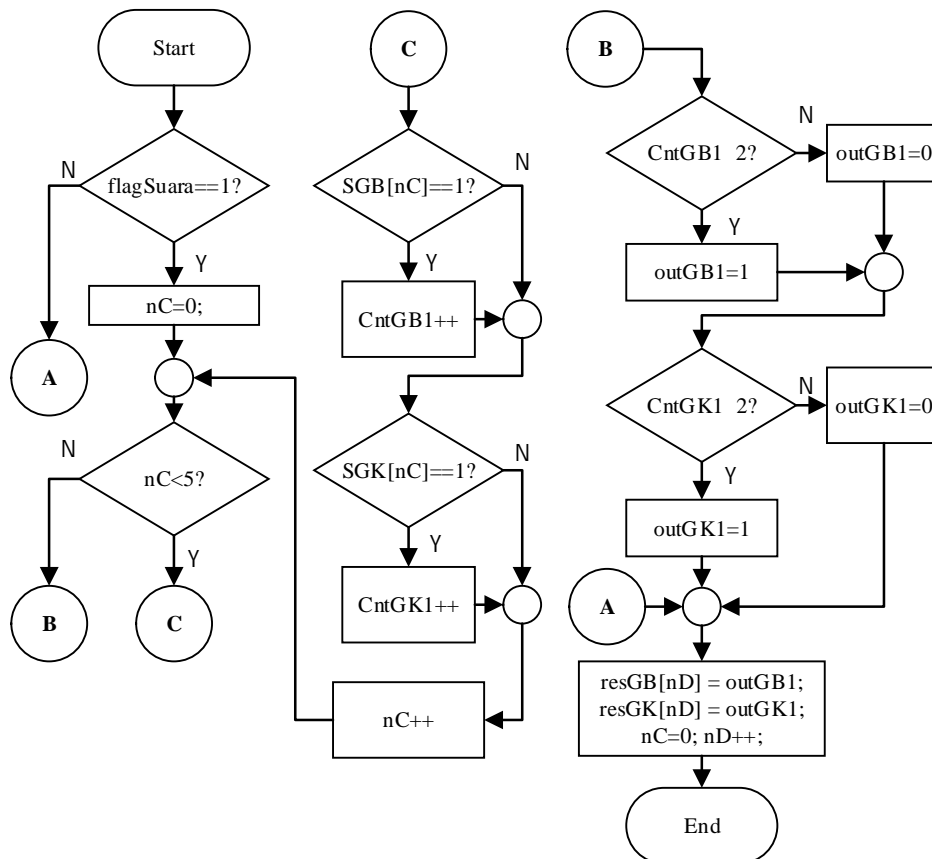
Deteksi pola musik diawali dengan *sampling* suara gendang tipak dengan periode 10ms. Proses *sampling* tidak dipengaruhi oleh *flagSuara* sehingga proses tetap aktif selama sistem menyala. Subfungsi *sampling* ADC gendang tipak memproses data dari ADC1 yaitu keluaran *filter* gendang besar dan ADC2 keluaran *filter* gendang kecil. Masing-masing *input* dibandingkan dengan nilai referensi ADC untuk menentukan keluaran ADC. Jika nilai ADC lebih dari ambang batas ADC maka keluaran berlogika tinggi. Hasil *sampling* kedua gendang dimasukkan ke dalam variabel *array* yang akan diproses pada subfungsi deteksi ketukan gendang. Proses *sampling* gendang tipak ditunjukkan dengan diagram alir pada Gambar 3.11.



Gambar 3.11 Diagram alir proses *sampling* gendang tipak

Subfungsi deteksi gendang tipak ditentukan oleh kondisi musik pengiring. Proses akan berjalan bila kondisi *flag* aktif yaitu saat musik pengiring diputar. Proses menggunakan dua *counter* untuk menghitung keluaran ADC1 dan ADC2 yang akan dijadikan sebagai penentu adanya ketukan gendang tipak. Variabel *counter* dimasukkan ke dalam iterasi *for* yang membaca keluaran variabel *array* pada subfungsi *sampling* gendang tipak. *Counter* akan

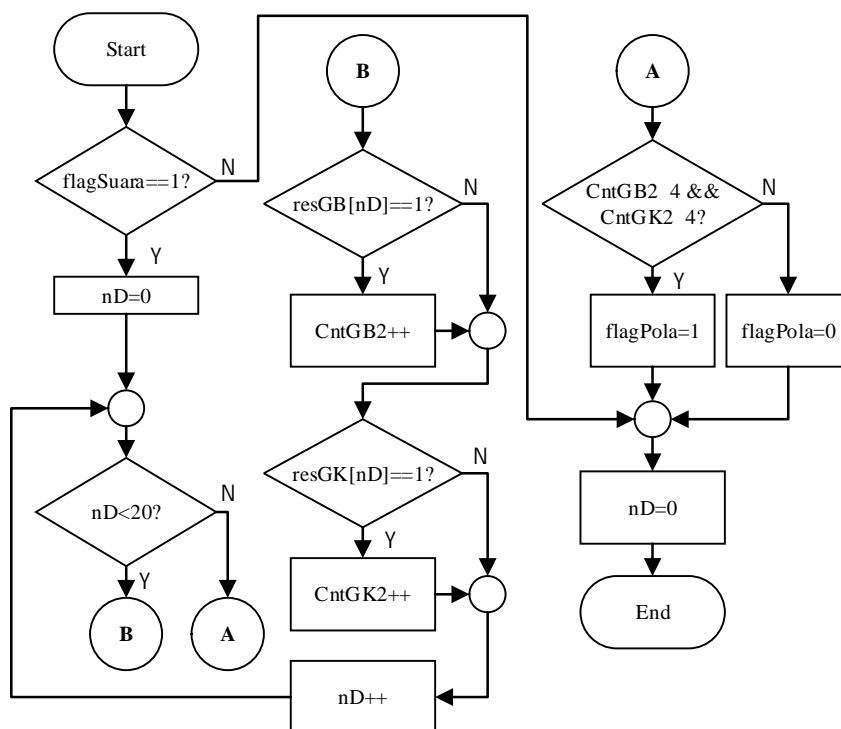
menambahkan satu jika keluaran berlogika tinggi. Jumlah akhir *counter* dibandingkan dengan batas minimum penentu adanya gendang. Hasil perbandingan inilah yang menjadi keluaran subfungsi deteksi gendang tipak, dengan kondisi: ada gendang jika keluaran berlogika tinggi. Ketukan gendang dimasukkan ke dalam variabel *array* untuk proses penentuan pola irama musik. Proses deteksi gendang tipak berulang setiap periode 50 ms selama *flag* aktif. Adapun diagram alir deteksi gendang tipak ditunjukkan pada Gambar 3.12.



Gambar 3.12 Diagram alir deteksi gendang tipak

3.2.2.3 Deteksi Pola Irama Musik Pengiring

Subfungsi deteksi pola irama musik merupakan subfungsi yang mengolah ketukan gendang tipak yang telah diproses pada subfungsi deteksi gendang tipak menjadi data *digital* penanda pola musik pengiring. Proses deteksi pola irama musik dipengaruhi oleh *flag* sebagai penentu sistem aktif dan melakukan proses. Selama sistem aktif (*flag* aktif), proses berulang setiap 1000 ms dengan masukan berupa variabel *array* ketukan gendang besar dan gendang kecil. Ketukan gendang dijadikan sebagai acuan keadaan ada/ tidak ada pola musik. Adapun diagram alir deteksi pola irama musik ditunjukkan pada Gambar 3.13.

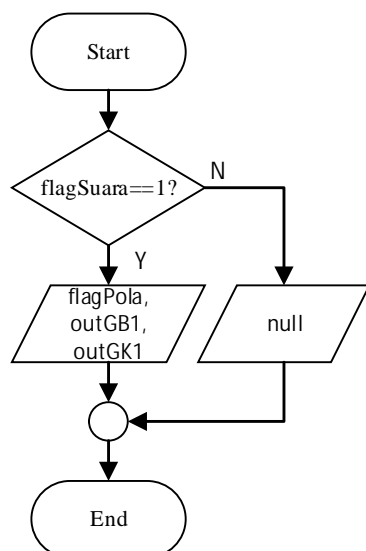


Gambar 3.13 Diagram alir deteksi pola irama musik

Sistem menggunakan tiga buah *counter* dimana satu *counter* merupakan variabel global yang digunakan untuk iterasi *for* dan dua *counter* gendang merupakan variabel lokal untuk menghitung jumlah ketukan gendang selama 1000 ms atau sebanyak 20 data. Pembacaan gendang besar dan gendang kecil memiliki proses yang sama dan berada dalam satu iterasi. Setiap iterasi, keluaran *array* dibaca dan jika ada gendang maka *counter* akan menambahkan satu. Setelah proses iterasi selesai, maka kedua nilai *counter* akan dibandingkan dengan nilai batas untuk menentukan hasil pola musik. Sistem membaca ada pola irama musik jika *counter* gendang besar dan *counter* gendang kecil sama-sama memiliki ketukan lebih dari sama dengan empat. Nilai *counter* dikembalikan ke nol untuk setiap perulangan proses atau setiap 1000 ms.

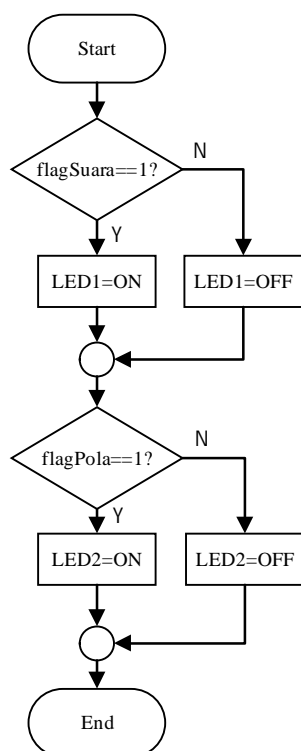
3.2.2.4 Pengiriman Data Musik dan Indikator Sistem

Data hasil deteksi gendang dan deteksi pola musik merupakan data yang akan dikirimkan ke komputer. Namun masing-masing subfungsi memiliki periode pemrosesan yang berbeda, menyebabkan pengiriman data tidak bisa dilakukan secara terpisah pada masing-masing subfungsi. Subfungsi pengiriman data mengirimkan data berupa pulsa *digital* dari *flagPola* yang merupakan keluaran subfungsi deteksi pola irama musik serta *outGB1* dan *outGB2* yang merupakan ketukan gendang besar dan gendang kecil. Data dikirimkan setiap 25ms. Masing-masing data akan ditampilkan pada komputer berupa pulsa *digital*. Diagram alir pengiriman data ditunjukkan pada Gambar 3.14.



Gambar 3.14 Diagram alir pengiriman data

Indikator dibuat sebagai penanda sistem secara *realtime* berupa nyala LED. Subfungsi indikator dibuat terpisah karena penetapan kondisi tidak berdasarkan waktu, namun berdasarkan hasil keluaran deteksi suara dan deteksi pola. Digunakan dua buah LED sebagai penentu dua kondisi yaitu kondisi ada suara serta kondisi ada pola musik. Kedua LED akan menyala jika memenuhi masing- masing kondisi, yaitu LED1 akan menyala jika flagSuara aktif dan LED2 akan menyala jika flagPola aktif. Diagram alir indikator sistem ditunjukkan pada Gambar 3.15.



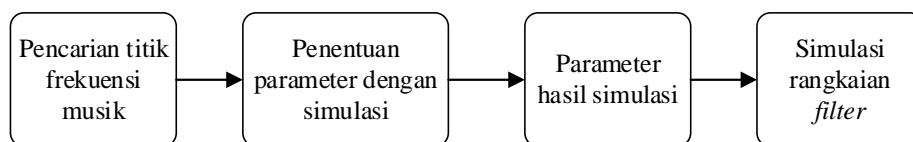
Gambar 3.15 Diagram alir indikator sistem

3.3 Pengujian Alat

Untuk menganalisis kinerja alat apakah sudah sesuai dengan perancangan diperlukan serangkaian pengujian sistem. Pengujian dilakukan secara terpisah setiap bagian pada blok diagram perancangan *hardware* untuk memeriksa apakah masing- masing subsistem sudah berfungsi atau tidak. Selanjutnya dilakukan pengujian keseluruhan untuk mengetahui kinerja sistem apakah sudah sesuai dengan tujuan yang dibuat. Pengujian yang dilakukan meliputi:

3.3.1 Analisis Hasil Pengambilan Data Musik dan Simulasi

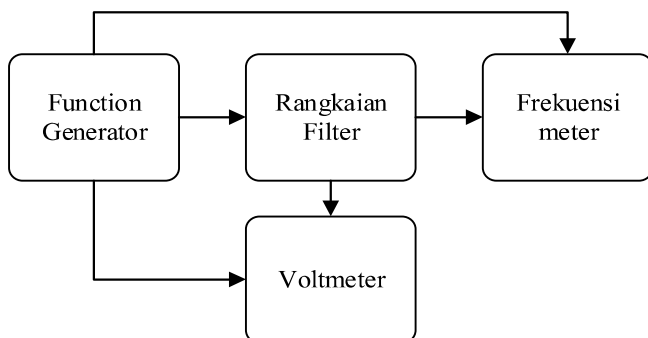
Data musik yang diambil berupa gendang tipak didapatkan dengan memeriksa respons frekuensi pada waktu dimana terdengar suara gendang besar dan waktu terdengar gendang kecil menggunakan *frequency analysis tool*. Respons yang didapatkan dijadikan sebagai patokan awal perancangan *filter*. Selanjutnya dilakukan pengujian dengan membuat simulasi *filter* pada *scientific filter tool* dengan ambang batas frekuensi sesuai dengan data yang ditampilkan *frequency analysis tool*. Simulasi *filter* dilakukan dengan melakukan pemilihan jenis *filter* yang digunakan, menentukan respons *filter* serta menentukan orde *filter*. Hasil dari simulasi dijadikan sebagai patokan dalam perancangan rangkaian *filter*. Sebelum diimplementasikan ke dalam rangkaian sesungguhnya, dilakukan simulasi rangkaian menggunakan *software* Tina-TI untuk membandingkan respon *filter* perancangan dengan simulasi *scientific filter*. Skema pengambilan data musik ditunjukkan pada Gambar 3.16.



Gambar 3.16 Skema pengambilan data musik

3.3.2 Pengujian Rangkaian Filter

Pengujian rangkaian filter dilakukan untuk menguji nilai penguatan $H(s)$ secara praktik dengan $H(s)$ secara teori, baik dalam penguatan tegangan maupun penguatan desibel. Skema pengujian diperlihatkan pada Gambar 3.17.

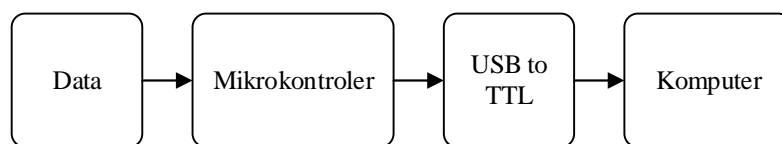


Gambar 3.17 Skema pengujian rangkaian filter

Rangkaian *filter* yang diuji terdiri atas sebuah *lowpass filter* dan dua buah *bandpass filter*. Pengujian rangkaian *filter* dilakukan dengan memberikan *input* tegangan sinusoida 1 V_{pp} atau 0,7071 V_{rms} dengan beragam frekuensi antara 0- 24 kHz menggunakan *function generator* untuk ketiga rangkaian *filter*. Hasil keluaran yang diamati pada frekuensi meter adalah frekuensi *input*, sedangkan voltmeter digunakan untuk mengukur tegangan *input* pada awal pengukuran dan mengukur tegangan *output* setiap perubahan frekuensi *input*. Ketiga parameter yang didapatkan dalam pengujian digunakan untuk mencari nilai penguatan dalam tegangan serta penguatan dalam desibel. Pengujian dianggap berhasil jika besar penguatan secara praktik mendekati nilai perhitungan secara teori.

3.3.3 Pengujian USB to TTL

Pengujian USB to TTL dilakukan untuk memeriksa kesesuaian data yang dikirim oleh mikrokontroler dengan data yang diterima oleh komputer. Skema pengujian ditunjukkan pada Gambar 3.18.

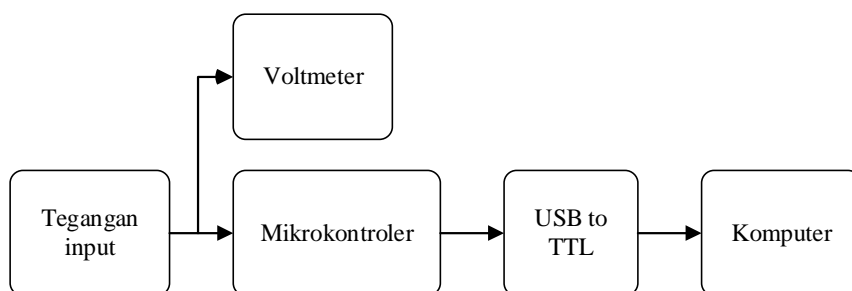


Gambar 3.18 Skema pengujian USB to TTL

Pengujian dilakukan dengan mengirimkan data berupa karakter yang dimasukkan kedalam program pada mikrokontroler ke komputer melalui USB to TTL. Pengujian dianggap berhasil jika seluruh data *output* yang ditampilkan pada komputer sama dengan data *input* yang dikirimkan oleh mikrokontroler.

3.3.4 Pengujian ADC Mikrokontroler

Data *input* mikrokontroler merupakan data analog keluaran *filter* yang dikonversi menjadi data *digital* menggunakan ADC. Pengujian ADC mikrokontroler diperlukan untuk menguji nilai digital hasil konversi tegangan analog yang diolah oleh mikrokontroler. Pengujian dilakukan dengan memberikan tegangan *input* analog pada *pin* ADC mikrokontroler, kemudian mengamati nilai ADC keluaran berupa data digital. Pengujian ADC mikrokontroler ditunjukkan pada Gambar 3.19.



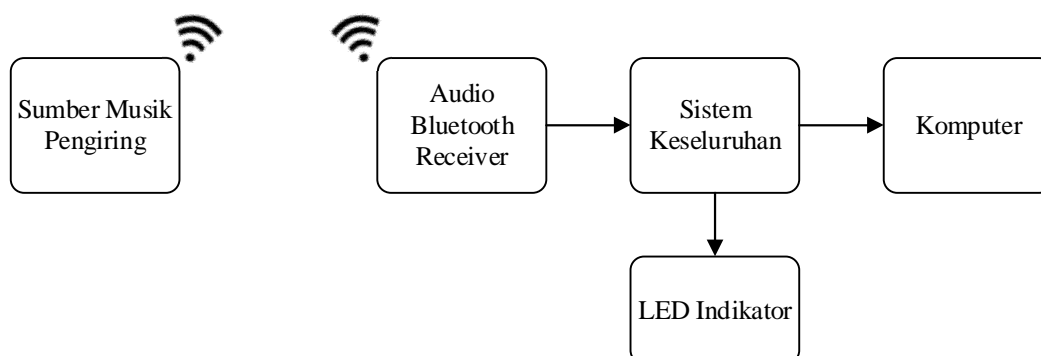
Gambar 3.19 Skema pengujian ADC mikrokontroler

Sumber tegangan *input* DC diubah- ubah dari 0 V sampai 5,42 V menggunakan resistor variabel sebagai rangkaian pembagi tegangan dengan perubahan setiap 0,25 V yang diukur menggunakan voltmeter. Data keluaran ADC mikrokontroler dibandingkan dengan perhitungan secara teori untuk melihat apakah *error* pada ADC mikrokontroler besar atau kecil. Pengujian dianggap berhasil jika nilai konversi ADC mikrokontroler sama atau mendekati nilai perhitungan secara teori.

Pengujian ADC dilanjutkan dengan menguji nilai ADC musik pengiring keluaran rangkaian *filter*. Musik pengiring digunakan dalam pengujian ADC musik untuk mengamati sinyal suara hasil *filter* gendang besar dan gendang kecil. Data ADC yang dikirimkan ke komputer disalin dan *plot* ke dalam grafik. Grafik digunakan untuk membandingkan antara data pengujian ADC musik dengan grafik musik hasil simulasi *filter* Tina TI. Ketukan gendang yang bergantian dan teratur dijadikan sebagai patokan dalam penentuan adanya pola musik pengiring.

3.3.5 Pengujian Performa Sistem Keseluruhan

Pengujian keseluruhan sistem dilakukan untuk mengetahui apakah setiap blok yang telah diuji dapat bekerja sama optimalnya ketika dirangkai menjadi satu sistem yang utuh sesuai dengan perencanaan sebelumnya. Penggabungan *hardware* dan *software* dilakukan untuk memeriksa sistem. Adapun skema pengujian unjuk kerja sistem keseluruhan ditunjukkan pada Gambar 3.20.



Gambar 3.20 Skema pengujian performa sistem keseluruhan

Prosedur pengujian dilakukan dengan menggunakan musik pengiring KRSTI 2016. Sumber musik dihubungkan ke *audio bluetooth transmitter* yang tersedia pada komputer dan diterima oleh *audio bluetooth receiver*. *Bluetooth receiver* dihubungkan ke sistem keseluruhan, dan sistem dihubungkan ke komputer menggunakan *USB to TTL*. Terdapat indikator bahwa sistem mulai melakukan pemrosesan berupa LED sistem *on* serta LED deteksi pola yang akan menyala setiap ada pola musik yang dihasilkan. Hasil pengolahan mikrokontroler ditampilkan pada komputer. Data yang diterima oleh komputer tidak secara *realtime* melainkan disimpan terlebih dahulu agar dapat ditampilkan secara keseluruhan. Komputer akan menampilkan ketukan gendang besar dan gendang kecil serta pola irama yang dihasilkan. Sistem dianggap berhasil apabila pola keluaran sistem sesuai dengan kondisi pola yang telah ditentukan pada perancangan *software* serta indikator LED mampu menunjukkan keadaan ada/ tidak adanya musik pengiring serta ada/ tidak adanya pola irama musik pengiring.