

BAB II

LANDASAN TEORI

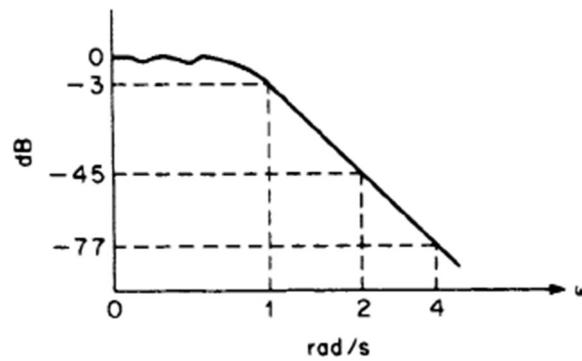
2.1 Fix Equalizer

Fix Equalizer adalah kumpulan dari beberapa *filter* yang berfungsi untuk menyamakan sinyal keluaran agar mendekati sinyal masukannya dengan frekuensi yang telah ditentukan. Pada *audio*, *equalizer* berfungsi untuk menyamakan antara suara yang masuk dengan suara yang keluar dari pengeras suara (*loud speaker*). Proses ekualisasi pada *audio* tergantung dari rasa seni dan respon telinga seseorang, maka dari itu *equalizer* dibuat agar setiap orang dapat mendengarkan musik sesuai dengan selernya masing – masing^[7]. Adapun manfaat dan kegunaan *equalizer* diantaranya :

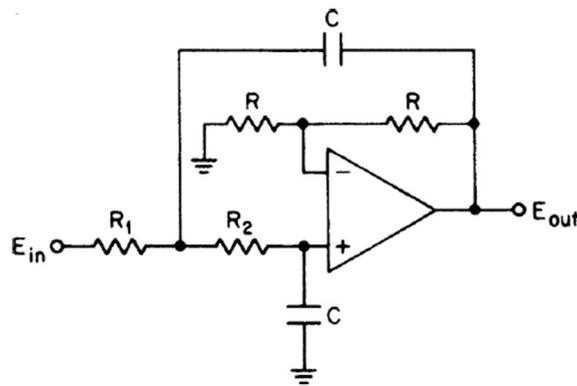
1. Memblok suara – suara yang tidak diperlukan
2. Meningkatkan kejernihan dan kejelasan suara dari instrumen
3. Menempatkan instrumen sehingga ada pada zona frekuensinya
4. Memisahkan tiap instrumen agar tidak bertabrakan dengan instrumen lain.

2.1.1 Filter Aktif Chebysev Low-Pass

Low pass filter adalah filter yang meloloskan sinyal dengan frekuensi yang lebih rendah daripada frekuensi *cut-off* yang telah ditentukan, dan meredam sinyal yang frekuensinya lebih tinggi daripada frekuensi *cut-off*. *Chebysev* merupakan jenis frekuensi *filter* yang memiliki ripple pada daerah *pass-band* dan curam pada frekuensi setelah *cut-off*, untuk mengurangi *ripple* bisa dengan menambahkan jumlah orde. **Gambar 2.1** adalah respon frekuensi dari *filter Low-pass Chebysev*, sedangkan **Gambar 2.2** adalah skema dasar *Rangkaian Filter Aktif Low-Pass*^[7].



Gambar 2. 1 Respon Filter *Low-Pass Chebysev*



Gambar 2. 2 Rangkaian Filter Aktif *Low-Pass*

Nilai – nilai komponen pada **Gambar 2.2** dapat di cari dengan beberapa tahapan kecuali nilai C dan R yang dapat kita tentukan sendiri. Pertama, dapat di cari nilai *frequency scaling factor (FSF)* atau perbandingan antara frekuensi yang diinginkan dan frekuensi yang telah diberikan menggunakan persamaan 2.1 [7].

$$FSF = 2\pi f_c \quad \dots\dots\dots (2.1)$$

Parameter – parameter *filter* dapat ditentukan besarnya berdasarkan tabel normalisasi *filter chebysev* seperti yang ditunjukkan **Tabel 2.1**. Setelah nilai normalisasi diketahui, maka dilakukan denormalisasi dengan tujuan agar didapatkan nilai komponen yang ada dipasaran. Untuk denormalisasi dapat dilakukan dengan cara berikut^[7],

$$\alpha' = \alpha \cdot \text{FSF} \quad \dots\dots\dots (2.2)$$

$$\beta' = \beta \cdot \text{FSF} \quad \dots\dots\dots (2.3)$$

untuk menentukan nilai komponen R1 dan R2, dimana ini berlaku untuk semua orde, jika orde diperbanyak maka rangkaian yang sama dipasang secara seri pada dari *output* rangkaian pertama ke *input* rangkaian selanjutnya^[7].

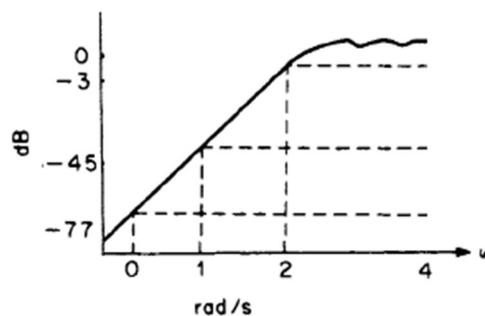
$$R_1 = \frac{1}{2\alpha' C} \quad \dots\dots\dots (2.4)$$

$$R_2 = \frac{2\alpha'}{C(\alpha'^2 + \beta'^2)} \quad \dots\dots\dots (2.5)$$

Tabel 2.1 dicantumkan karena untuk memudahkan dalam melakukan perancangan. Dengan melihat nilai pada tabel maka denormalisasi dapat dilakukan dengan mudah^[7].

2.1.2 Filter Aktif Chebyshev High-Pass

High-pass filter adalah filter yang meloloskan sinyal dengan frekuensi diatas frekuensi *cut-off* yang telah ditentukan, dan meredam sinyal yang frekuensinya dibawah frekuensi *cut-off*. Dalam *filter chebyshev*, *high-pass filter* merupakan transformasi dari *low-pass filter*, sehingga rangkaiannya pun tidak berbeda. Hanya saja menukar posisi R menjadi C, dan sebaliknya. **Gambar 2.3** adalah *Respon frekuensi Filter Aktif High-Pass*^[7].



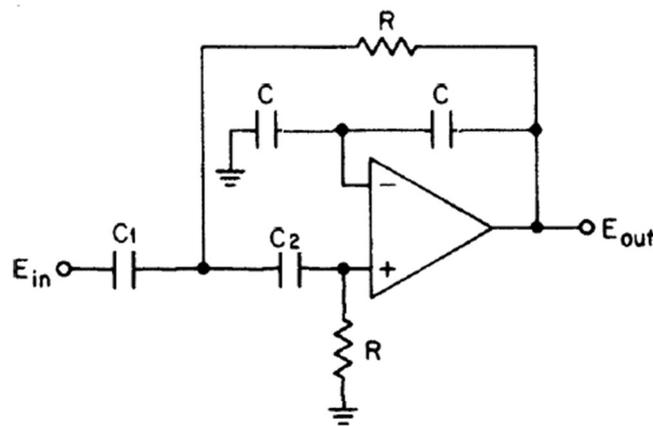
Gambar 2. 3 Respon Filter *High-Pass Chebyshev*

untuk mencari nilai komponen pada *filter* jenis ini maka digunakan rumus sebagai berikut^[7]:

$$C_{hp} = \frac{1}{R_{Lp}} \quad \dots\dots\dots (2.6)$$

$$R_{hp} = \frac{1}{C_{Lp}} \quad \dots\dots\dots (2.7)$$

dari **Gambar 2.2** maka di transformasi menjadi, seperti **Gambar 2.4** berikut^[7],



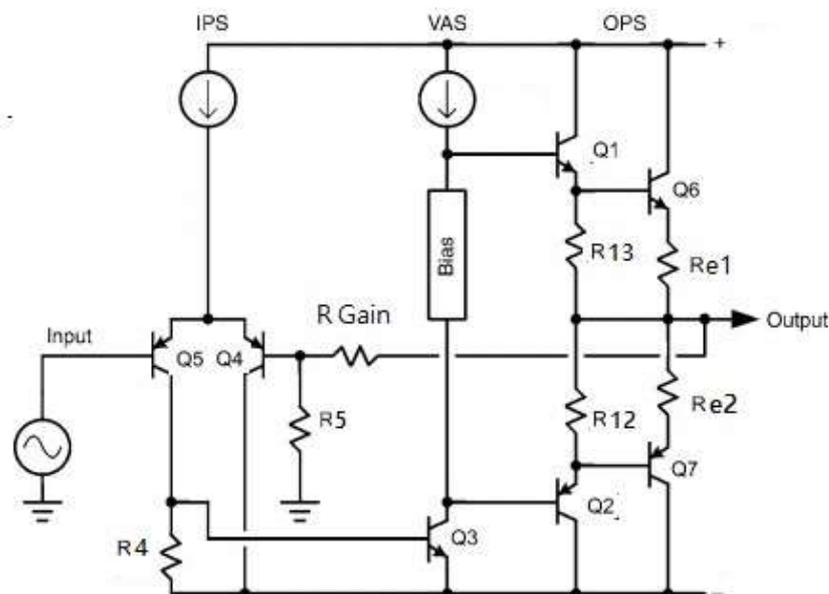
Gambar 2. 4 Transformasi dari LPF ke HPF

Tabel 2. 1 Tabel Normalisasi *Filter Chebyshev*

Order n	Real Part $-\alpha$	Imaginary Part $\pm j\beta$
2	0.6743	0.7075
3	0.4233 0.8467	0.8663
4	0.6762 0.2801	0.3828 0.9241
5	0.5120 0.1956 0.6328	0.5879 0.9512
6	0.5335 0.3906 0.1430	0.2588 0.7072 0.9660
7	0.4393 0.3040 0.1085 0.4876	0.4339 0.7819 0.9750
8	0.4268 0.3618 0.2418 0.08490	0.1951 0.5556 0.8315 0.9808
9	0.3686 0.3005 0.1961 0.06812 0.3923	0.3420 0.6428 0.8661 0.9848

2.2 Penguat Daya

Penguat daya adalah gabungan dari penguat arus dan tegangan yang berfungsi untuk meningkatkan daya keluaran ke beban sesuai kebutuhan, tanpa mengurangi data sinyal yang diterima dari masukan yang daya dan resistansinya kecil. Selain itu ada parameter seperti THD, SNR, dan intensitas kebisingan suara yang akan diukur untuk mengetahui kualitas dari penguat daya. **Gambar 2.5** merupakan skematik keseluruhan umum dari pembuatan penguat daya. ^{[1][4]}



Gambar 2.5 Rangkaian dasar *PoAmp*

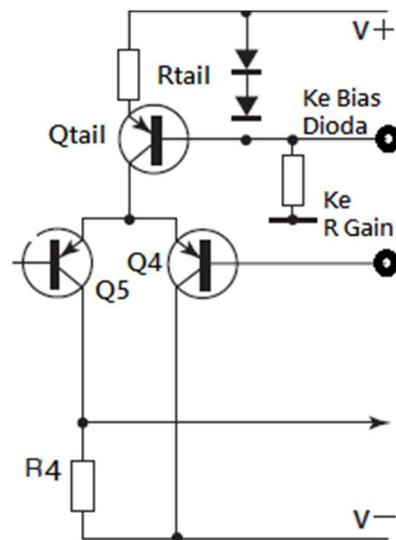
2.2.1 Bagian – Bagian *PoAmp* Kelas AB

Dalam perancangannya *PoAmp* tidak berdiri sendiri. Untuk membuat *PoAmp* terdiri dari penguat diferensial, VAS, dan penguat akhir.

2.1.1.1 Penguat Diferensial

Penguat diferensial merupakan penguat dengan dua buah transistor yang emitemnya dihubungkan menjadi satu. Penguat diferensial yang digunakan dalam *PoAmp OCL* yaitu beroperasi pada (*Common-mode Operation -CM*), dimana cara

kerjanya hanya salah satu *input* yang digunakan dan input lainnya di *feedback* ke input pertama. **Gambar 2.6** merupakan gambar dari rangkaian penguat differensial^[4].



Gambar 2.6 Rangkaian Penguat Differensial

Pada **Gambar 2.6** Transistor *Q tail* merupakan *Tail Constant Current* untuk mendapatkan nilai arus pada *tail* dapat dihitung dengan menentukan frekuensi maksimal untuk menentukan nilai $\omega = 2\pi f$. Dimana nilai ω merupakan kecepatan sudut dalam satuan Radial/detik. Maka untuk mencari arus *tail* (arus yang mengalir ke kaki emiter penguat diferensial) dapat digunakan rumus sebagai berikut^[4].

$$i_{tail} = \omega \cdot C_{dom} \cdot V_{pk} \quad \dots\dots\dots (2.8)$$

Dengan menetapkan arus tail yang mengalir ke emiter penguat diferensial = 2 mA. Maka nilai R_E dapat dicari dengan menggunakan rumus $R_E = \frac{VD}{I_{Etot}}$, dimana nilai $VD = 0,7v$ dengan syarat karakteristik transistor dan dioda harus sama. Penetapan arus bias dengan 2 buah dioda dan 1 transistor ini dinamakan *constat bias diode* yang berguna untuk menstabilkan arus yang mengalir ke penguat diferensial^[3]. Sehingga nilai $r'_e (= 25 \text{ mV} / (I_{Etot}/2))$. Dari parameter – parameter

tersebut untuk menghitung penguatan diferensial dapat menggunakan rumus sebagai berikut^[6];

$$AD = -\frac{R2//(\beta.r'_e)}{2(r'_e+RE)} \dots\dots\dots (2.9)$$

tegangan output,

$$v_{out} = Vin.AD \dots\dots\dots (2.10)$$

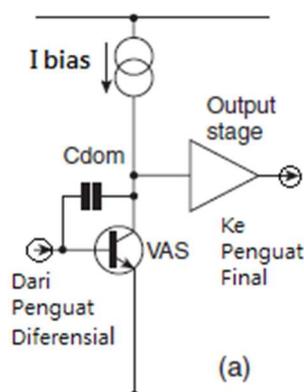
jika terdapat perbedaan antara nilai v_{o1} dan v_{o2} yang disebabkan nilai RL yang berbeda, maka nilai penguatan akan = 0.

serta impedansi input,

$$Z_i = 2\beta RE \dots\dots\dots (2.11)$$

2.1.1.2 Tahap Penguatan Tegangan (VAS)

VAS (*Voltage Amplification Stage*) atau tahap penguat tegangan adalah transistor yang berfungsi sebagai transistor power yang bekerja untuk menarik sinyal bias ke tegangan negatif, sedangkan yang menarik sinyal bias ke tegangan positif adalah nilai resistansi yang dipasang secara seri. Karena arus yang mengalir pada masing – masing berbeda maka diperlukan *driver* sebelum masuk ke penguat *final*. **Gambar 2.7** merupakan rangkaian dasar dari VAS^[4].



Gambar 2. 7 Rangkaian VAS

Pada **Gambar 2.7** sebelum masuk ke basis ada nilai resistansi yang disebut R_B maka arus yang mengalir disebut sebagai arus panjar ($I_{bias} = 10 \text{ mA}$), sehingga dapat dicari menggunakan rumus biasing dioda sebagai berikut^[6].

$$I_{vas} = \frac{V_{cc} - V_{Dtot}}{R} \quad \dots\dots\dots (2.12)$$

Jika *output* dipasang secara darlington maka penguatannya dapat dicari dengan rumus sebagai berikut,

$$G = \frac{\beta_1 \cdot \beta_2}{R_e} \quad \dots\dots\dots (2.13)$$

dari nilai i_e dapat dicari tegangan *output* yaitu,

$$v_{out} = v_{in} \cdot G \quad \dots\dots\dots (2.14)$$

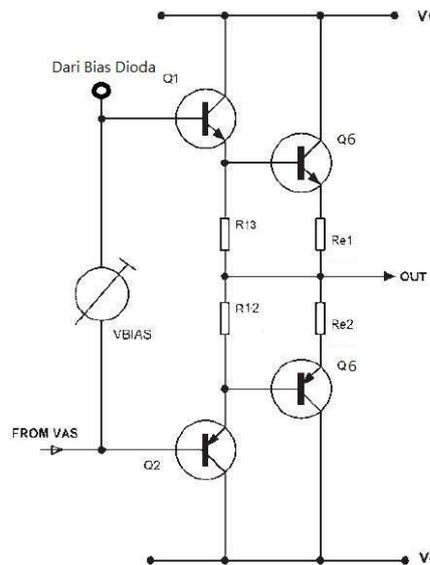
dapat juga ditambahkan *miller capacitor*, untuk menstabilkan loop umpan balik yang dipasang secara paralel dengan kaki basis-kolektor pada VAS, dengan menggunakan rumus,

$$C_{dom} = \frac{0,159}{f \cdot R_{ltp} \cdot A_{cl}} \quad \dots\dots\dots (2.15)$$

2.1.1.3 Penguat Akhir kelas AB

Penguat Akhir Kelas AB adalah salah satu dari jenis penguat daya gabungan dari penguat daya kelas A dan kelas B untuk menghilangkan cacat silang dengan efisiensi $\pm 60\%$ tetapi tidak menghilangkan fidelitas sinyal *output* nya. Dinamakan *cascade* karena penguatan kelas AB yang bertingkat sebanyak 2x, sehingga menghasilkan arus yang tinggi. Untuk itu penguat akhir jenis ini banyak digunakan pada pembuatan *PoAmp* di dalam sistem audio. **Gambar 2.8** merupakan rangkaian penguat akhir kelas AB *darlington complementary symetri output* atau kaki emiter

pada penguat driver dipasang ke kaki basis dari penguat *final* sehingga penguatannya adalah hasil kali dari beta transistor^{[6][4]}.



Gambar 2. 8 Penguat Kelas Ab *Darlington Complementary Symetri Output*

Dari **Gambar 2.8** arus yang mengalir pada R1 dan R2 adalah sebesar 20 mA dan total arus bisa didapatkan dengan mengalikan β atau $\beta = \beta_1 \cdot \beta_2$ dari ke 2 transistor yang dipasang secara darlington^[4]. Untuk nilai Re1 dan Re2 berfungsi sebagai stabilitas bias yang disebut dengan RE terpasang langsung ke *speaker output* sehingga penguatannya adalah sebagai berikut^[5].

$$G = \frac{R_L}{R_L + Re1} \dots\dots\dots (2.16)$$

Parameter – parameter untuk menentukan tegangan dan arus yang digunakan untuk membuat *power amplifier* adalah sebagai berikut^[5]. Untuk mencari tegangan input *peak* maka,

$$V_i(p) = V_L(p) = \sqrt{2} V_i(rms) \dots\dots\dots (2.17)$$

daya output adalah,

$$P_o(ac) = \frac{V_L^2(p)}{2R_L} \dots\dots\dots (2.18)$$

arus beban puncak *peak* dapat di cari dengan,

$$I_L(p) = \frac{V_L(p)}{R_L} \dots\dots\dots (2.19)$$

daya input yang diberikan adalah,

$$P_i(dc) = V_L I_L \dots\dots\dots (2.20)$$

maka daya disipasi dapat dihitung dengan,

$$P_Q = \frac{P_i - P_o}{2} \dots\dots\dots (2.21)$$

sehingga efisiensi didapatkan dengan,

$$\% \eta = \frac{P_o}{P_i} \times 100\% \dots\dots\dots (2.22)$$

maka untuk mengetahui nilai arus yang mengalir pada transistor driver dapat

menggunakan rumus sebagai berikut;

$$I_{dr} = \frac{I_L(p)}{\beta} \dots\dots\dots (2.23)$$

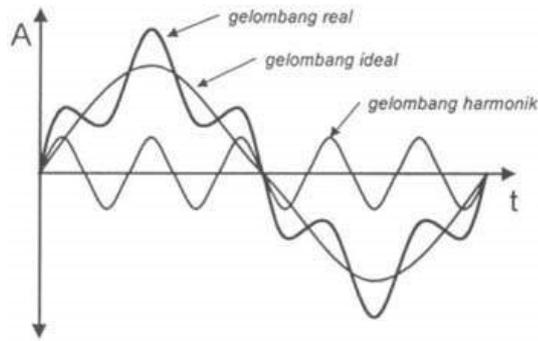
Jika dalam pengukuran maka daya *output* dapat dihitung menggunakan persamaan sebagai berikut.

$$P_{Out} = V_{(p)}.I_{out} \dots\dots\dots (2.24)$$

2.2.2 THD

THD (*Total Harmonic Distortion*) adalah komponen yang sangat penting dalam sebuah system *audio*. THD sendiri merupakan ukuran jumlah harmonik komponen yang ada didalam sinyal dalam satuan persen, atau perbandingan antara seluruh jumlah harmonik komponen dengan frekuensi dasarnya. Dimana nilai THD yang baik tidak boleh lebih dari 5% karena akan merusak komponen saat ia bekerja.

Gambar 2.9 adalah gambar yang menjelaskan bagaimana distorsi harmonisa terjadi. [5]



Gambar 2.9 Gelombang Fundamental Harmonisa

Gambar tersebut menjelaskan adanya 2 gelombang berbeda, yaitu gelombang *real* yang datang secara acak dikalikan dengan gelombang ideal yang maksudnya untuk mempermudah dalam pembacaan gelombang. Perhitungan distorsi harmonisa pada sistem *audio* sebenarnya dihitung setiap frekuensi yang masuk. Selain menggunakan alat, dapat dihitung sesuai dengan kepekaan dan selera telinga manusia itu sendiri. Untuk perhitungan matematisnya dapat digunakan rumus sebagai berikut^{[3] [5][20]}.

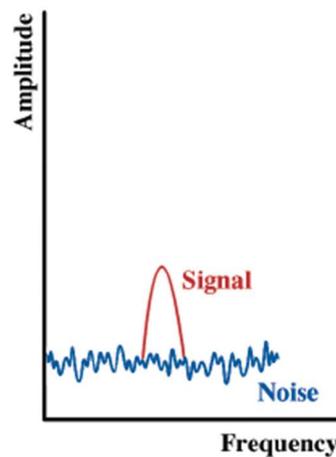
$$\%THD = 100 \times 10^{\frac{Total\ Distorsi}{10}} \quad \dots\dots\dots (2.25)$$

Untuk mendapatkan nilai Total Distorsi dengan menggunakan persamaan $Total\ Distorsi = 10\text{Log}(10^{\frac{A2}{10}} + 10^{\frac{A3}{10}} + 10^{\frac{A4}{10}} + \dots + 10^{\frac{An}{10}})$, mengetahui baik atau buruknya suatu THD dapat dilihat dari seberapa kecil persentase yang dihasilkan dari perhitungan diatas. Jika tegangan yang diketahui dalam bentuk V_{pp} maka ubah kedalam bentuk V_{rms} dengan menggunakan $\frac{1}{2\sqrt{2}} \times V_{pp}$.

2.2.3 SNR

SNR (*Signal to Noise Ratio*) adalah perbandingan antara *strength* sinyal yang dihasilkan dengan *noise* atau sinyal pengganggu. Dimana sinyal *noise* merupakan sinyal yang tidak diinginkan, meskipun tidak bisa dihilangkan tetapi bisa di

minimalisir. Sumber dari sinyal *noise* dihasilkan dari piranti dalam komponen elektronik itu sendiri, maupun energi dari luar komponen elektronik itu sendiri seperti dari catudaya, kondisi lingkungan, dll. **Gambar 2.10** adalah gambar yang menunjukkan antara sinyal asli dan sinyal *noise*^[4].



Gambar 2. 10 Respon Frekuensi Sinyal Asli dan *Noise*

Gambar diatas menjelaskan bahwa spektrum yang berwarna biru adalah sinyal gangguan atau derau yang terbawa bersama dengan sinyal informasi yang berwarna merah. Sinyal derau tidak boleh lebih besar amplitudanya dari sinyal informasi karena jika hal tersebut terjadi, maka akan terjadi peredaman pada sinyal informasi atau sinyal informasi tidak terkirim secara utuh. Untuk menghitung SNR dapat menggunakan rumus $SNR = \frac{S}{N}$ ^[4].

Jika diubah kedalam satuan *decibel* (dB), maka

$$SNR_{dB} = 20 \log \left(\frac{V_{out_{rms}}}{V_{noise_{rms}}} \right) dB \quad \dots\dots\dots (2.26)$$

Kualitas sinyal yang dihasilkan akan semakin baik jika nilai SNR semakin besar, karena menandakan nilai pembaginya kecil yang menandakan bahwa *noise* nya kecil.

2.2.4 Tingkat Kebisingan Suara

Kebisingan adalah suara yang tidak diinginkan dan dapat mengganggu kenyamanan dan kesehatan dilingkungan sekitar satuan dai intensitas *level* kebisingan yaitu decibel (dB). *Level* intensitas cahaya bisa didapatkan dengan rumus sebagai berikut^{[9][10]}.

$$L_I(dB) = 10 \log \left(\frac{I}{I_0} \right) dB \quad \dots\dots\dots (2.27)$$

Dimana,

L_I = Level Intensitas Suara (dB)

I = Instensitas Suara (W/m^2)

I_0 = Intensitas suara referensi = $10^{-12} W/m^2$

Untuk mencari I dapat diperoleh dengan rumus sebagai berikut.

$$I = \frac{P}{A} \quad \dots\dots\dots (2.28)$$

Dimana,

P = Daya yang dihasilkan dari sumber suara (Watt)

A = Luas daerah cakupan suara (m^2)

Berikut adalah tabel nilai batas ambang kebisingan yang masih bisa diterima oleh telinga manusia^[10].

Tabel 2. 2 Tingkat Kebisingan Suara

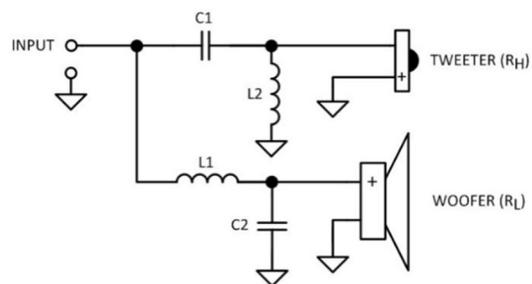
190 dB	Senjata berat, 10 m di belakang senjata
180 dB	Pistol mainan ditembakkan dekat telinga
170 dB	Menampar di telinga, cracker api meledak di bahu, ledakan senjata kecil pada jarak 50 cm
160 dB	Bor palu pada tabung kuningan atau pelat baja pada jarak 1 m, penyebaran airbag sangat dekat pada jarak 30 cm
150 dB	Bor palu dengan pandai besi pada jarak 5 m
130 dB	Tepukan tangan keras pada jarak 1 m
120 dB	Bersiullah pada jarak 1 m, lakukan uji jet pada jarak 15 m

	Ambang batas rasa sakit ditelinga
115 dB	Suara pesawat take-off pada jarak 10 m
110 dB	Sirine *) pada jarak 10 m, suara diskotik, dekat dengan pengeras suara di konser rock, biola dekat dengan telinga seorang musisi orkestra
105 dB	Gergaji rantai pada jarak 1 m, membenturkan pintu mobil pada jarak 1 m, mobil balap pada jarak 40 m, music dengan speaker
100 dB	Mendengar music dengan head phones, mesin bor pada jarak 10m
95 dB	Suara tangisa, gergaji pada jarak 1 m
90 dB	Mesing penggiling pada jarak 1 m
	Selama 40 jam per minggu, kerusakan pendengaran bisa terjadi
85 dB	rantai gergaji 2-langkah pada jarak 10 m, penyiraman WC pada jarak 1 m
80 dB	Kebisingan lalu lintas yang sangat keras dari lori yang melaju pada jarak 7,5 m, lalu lintas tinggi di jalan tol dengan jarak 25 m
75 dB	Melewati mobil pada jarak 7,5 m, mesin penghancur kayu tanpa pembungkaman pada jarak 10 m
70 dB	Jalan raya di siang hari, hair dryer pada jarak 1 m dekat telinga
65 dB	Detakan ketika sakit jantung
60 dB	Mesin pemotong rumput bising pada jarak 10 m
55 dB	Volume rendah radio atau TV pada jarak 1 m, suara vacuum cleaner pada jarak
50 dB	Kulkas pada jarak 1 m, kiacaau burung di luar pada jarak 15 m
45 dB	Suara normal; berbicara, atau mendengar radio yang terhalang
40 d	Distraksi ketika sedang konsentrasi
35 dB	Kipas ruangan yang sangat tenang dengan kecepatan rendah pada jarak 1 m
25 dB	Suara nafas pada jarak 1 m
0 dB	Ambang batas pendengaran

2.2 Jaringan Crossover Pasif

Crossover Pasif adalah kumpulan dari *flter* pasif yang dirangkai sedemikian rupa untuk memproteksi dan membagi *range* frekuensi. Pada sebuah sistem *audio crossover* pasif dibagi sesuai jalur frekuensi yang akan diloloskan, misalnya 2 way (*Bass* dan *Treble*), 3 way (*Bass*, *Middle*, dan *Treble*). Diantara kedua jalur frekuensi tersebut yang paling umum digunakan adalah *crossover* 3 way, karena nada yang diperoleh akan semakin lengkap. Untuk itu akan dibahas mengenai *crossover* 3 way . Cara kerja *crossover* yaitu, jika yang diinginkan nada *bass* maka yang akan

diloloskan hanya frekuensi rendah saja, sedangkan frekuensi tengah dan tinggi akan diabaikan. Sedangkan jika yang diinginkan adalah nada *middle* maka frekuensi rendah dan tinggi akan diabaikan. Lalu jika yang diinginkan adalah nada *treble* maka frekuensi rendah dan tengah akan diabaikan. Gambar rangkaian untuk *crossover* pasif 2 way dapat dilihat dari gambar 2.11^[8].



Gambar 2. 11 Rangkaian *Crossover 2 Way*

Pada **Gambar 2.11** digunakan *filter* pasif untuk membuat rangkaian *crossover* dikarenakan tegangan *speaker* sudah di *supply* dari penguat daya, sehingga hanya membutuhkan proteksi dan pembagian frekuensi yang masuk ke masing – masing *speaker* saja. Untuk *speaker tweeter* digunakan *filter highpass* dan untuk *speaker midbass* digunakan *filter lowpass*.

2.4 Speaker

Speaker atau *loadspeaker* adalah sebuah transduser yang berfungsi untuk mengubah sinyal listrik ke bentuk getaran suara^[4]. Berdasarkan frekuensi kerjanya *speaker* terbagi kedalam beberapa jenis, diantaranya sebagai berikut:

a) *Mid Bass / Mid Woofer*

Merupakan jenis *speaker* yang mampu menghasilkan frekuensi suara 80 Hz – 350 Hz bisa dikatakan nada *bass* menengah.

b) *Tweeter*

Merupakan jenis *speaker* yang mampu menghasilkan frekuensi suara 5 KHz – 20 KHz atau nada – nada tinggi.

Kualitas *speaker* tidak hanya bergantung pada *speaker* itu sendiri melainkan ada beberapa aspek yang harus diperhatikan pada saat merancang suatu box sistem *speaker*, berikut adalah beberapa aspek yang harus diperhatikan pada saat membuat *box speaker*.

- a. Daya dan impedansi *speaker* tidak lebih kecil dari daya *amplifier*
- b. Bahan *box* yang digunakan harus bahan yang kedap suara
- c. Sesuaikan dengan penggunaan, seperti dimana dan untuk apa.

Dalam sebuah sistem audio *speaker* adalah komponen yang sangat penting digunakan. *Speaker* sendiri memiliki daya (P) dan impedansi (Z). Untuk menghitung daya *speaker* harus memerhatikan impedansi total *speaker*. Untuk memperoleh impedansi total *speaker* yang sesuai dengan impedansi pada *amplifier*, maka perhitungan dan penyambungannya dapat dilakukan dengan 3 cara, yaitu sambungan seri, paralel, dan seri-paralel. Perhitungan impedansi sama saja seperti pada saat menghitung resistor. Tetapi untuk daya yang dihasilkan, bagaimanapun sambungannya tetap akan bertambah sesuai dengan daya yang dimiliki oleh *speaker* tersebut.

2.6 Wireless

Wireless adalah teknologi yang dapat menghubungkan antara satu perangkat dengan perangkat yang lainnya tanpa media kabel, media yang digunakan pada teknologi *wireless* yaitu media udara. Seperti RF atau IR, menurut IEEE standar kode dari *wireless* adalah 802.xxx fungsi dari standar kode tersebut antara lain

untuk menghindari interferensi antar frekuensi pada saat berkomunikasi. Berikut adalah tabel yang menjelaskan spesifikasi wireless. **Tabel 2.3** merupakan spesifikasi dari alokasi frekuensi *wireless*^[11].

Tabel 2.3 Spesifikasi Wireless

802.11 Wireless Standards					
IEEE Standard	802.11a	802.11b	802.11g	802.11n	802.11ac
Tahun diadopsi	1999	1999	2003	2009	2014
Frekuensi	5 Ghz	2.4 Ghz	2.4 Ghz	2.4/5 Ghz	5Ghz
Kecepatan Data Maks	54 Mbps	11 Mbps	54 Mbps	600 Mbps	1 Gbps
Jenis Jangkauan Indoors	100 ft	100 ft	125 ft	225 ft	90 ft
Jenis Jangkauan Outdoors	400 ft	450 ft	450 ft	825 ft	1.000 ft

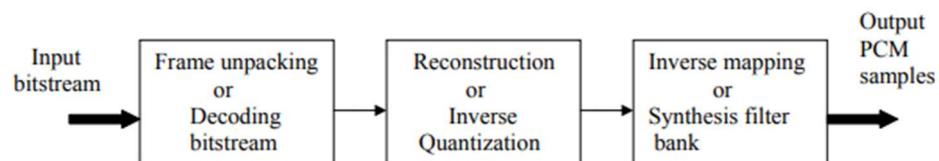
2.7 Android

Android adalah pengembangan dari sistem operasi Linux yang digunakan pada telepon genggam (*smartphone* atau *tablet*). Sistem operasi ini bersifat *open source* atau bebas dimodifikasi *source code*-nya sesuai selera pengembang. Sistem operasi android dapat bekerja pada perangkat itu sendiri, maupun menjadi jembatan agar dapat berkomunikasi dengan perangkat lain seperti dengan laptop, mikrokontroler, bahkan *smartphone*. Tentunya agar dapat berkomunikasi sistem operasi android tidak langsung dapat bekerja, melainkan harus di program dengan kemampuan pengembangnya. ^{[12][13]}

2.8 Decode

Decode adalah proses pemecahan data dari bentuk utuh ke bentuk pecahan berupa byte data. Sehingga data bisa dibaca, diproses, dan diolah. Decode

dibutuhkan ketika data yang sudah menjadi satu paket tidak dapat diotak – atik isinya sehingga sulit ketika akan mengirimkan file atau data tersebut. Maka jika dilakukan proses decode, isi dari data tersebut dapat diketahui sehingga kita bisa bebas mengakses isi daripada data tersebut sesuai kebutuhan^[16]. **Gambar 2.12** menunjukkan proses dari decode sebuah file^[17].



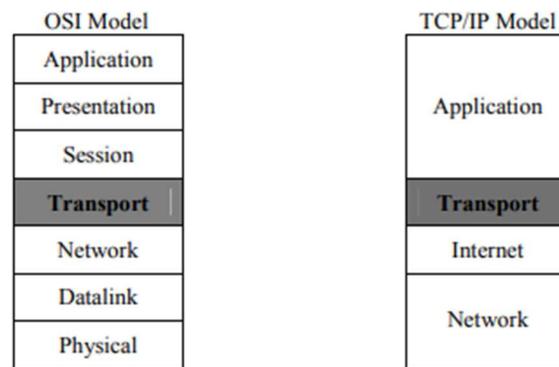
Gambar 2.12 Decode File

2.9 UDP

UDP atau *User Datagram Protocol* merupakan salah satu lapisan pada TCP/IP yang dibuat setelah protocol TCP. UDP dibuat untuk mengatasi masalah pada TCP dimana data yang dikirim harus dikoreksi dan berurutan sehingga menyebabkan delay yang cukup panjang. Sedangkan pada UDP data yang dikirim berupa seluruh data acak yang tidak tersusun^[17]. Karakteristik – karakteristik dari UDP adalah sebagai berikut :

- a. Tanpa koneksi : sehingga pesan dapat dikirim tanpa harus memiliki koneksi internet
- b. Tidak andal : karena data yang dikirim tanpa diurutkan terlebih dahulu
- c. Menyediakan data 16 bit : sehingga cocok digunakan untuk media *streaming*

Pada OSI Layer UDP terletak pada lapisan ke 4 yaitu transport layer dan pada lapisan ke 3 pada TCP/IP model. Dimana berfungsi untuk mengirim data – data dari satu media ke media lain. Pada **Gambar 2.13** ditunjukkan letak trasport layer^[17].



Gambar 2. 13 Lapisan OSI Model dan TCP/IP Model

2.10 PCM

PCM atau Pulse Code Modulation adalah sampel *audio digital* mentah. PCM adalah hasil ekstraksi dari format mp3, aac, dan format audio lain. Proses pengextraksian dari format audio ke PCM itu disebut decode yang hasilnya berupa data biner. Setelah di ekstraksi maka PCM harus diubah ke bentuk sinyal analog, agar dapat diputar melalui *speaker*. Untuk proses pengiriman pada protocol PCM dikirim menggunakan UDP untuk menjadikannya sinyal analog. Maka dipenerima harus disediakan pembaca sinyal PCM dalam bentuk UDP yang dapat menerima data PCM dan mengeluarkannya berupa suara pada *speaker*. Sehingga pada proses ini diperlukan filter analog untuk memastikan *noise* yang diterima dapat diredam^[19].