

## EQUALIZER DIGITAL DENGAN PENGONTROL MENGGUNAKAN KOMPUTER

Adrian Suryadinata<sup>1)</sup>, Hartono Pranjoto, Ph.D<sup>2)</sup>

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Katolik Widya Mandala Surabaya, Jalan Kalijudan 37  
Surabaya

Email: [evermore.evermore@gmail.com](mailto:evermore.evermore@gmail.com)

### ABSTRAK

*Equalizer adalah suatu kumpulan elektronik filter yang berguna untuk mengontrol respon frekuensi pada suatu sound system. Melalui equalizer kita dapat mengubah karakter suara secara total atau sebagian. Equalizer terbagi menjadi 3 jenis, yaitu shelving equalizer, parametric equalizer, graphic equalizer. Pada graphic equalizer, pengaturan dilakukan di awal penginstalan sistem audio di suatu ruangan. Graphic equalizer berfungsi untuk mengkompensasi respon frekuensi dari speaker akibat peletakan speaker, memperbaiki masalah akibat acoustic ruangan, dan mereduksi feedback.*

*Untuk keperluan tugas akhir kali ini, jenis equalizer yang dibuat adalah graphic equalizer 2/3 oktaf. Adapun dalam pembuatan equalizer jenis filter yang digunakan adalah jenis filter IIR. Graphic equalizer akan dibuat pada DSP board dengan menggunakan modul OMAP-L137 EVM. Equalizer yang dibuat akan dioperasikan dengan menggunakan komputer. Pengontrol yang dibuat pada komputer akan dibuat dengan menggunakan bahasa pemrograman C#. Untuk menghubungkan antara interface yang ada pada komputer dengan board DSP, digunakan teknik komunikasi serial menggunakan RS-232 melalui konektor DB 9 yang ada pada komputer dan board DSP.*

*Hasil pengujian dari tugas akhir ini menunjukkan bahwa equalizer yang dibuat mampu mengubah karakter suara dari musik yang dimainkan dengan cara mengatur besar gain atau attenuasi yang diberikan pada lagu.*

**KataKunci:** *equalizer, acoustic, filter, frekuensi.*

### I. Pendahuluan

*Equalizer merupakan suatu alat yang penting dalam bidang audio. Melalui alat ini kita dapat mengubah karakteristik suara. Alat ini biasanya digunakan untuk mengontrol frekuensi respon dari soundsystem. Melalui equalizer, kita mampu melakukan perbaikan pada kualitas suara yang disebabkan oleh adanya keterbatasan dari peralatan audio.*

*Bila kita melihat dari sisi kualitas suara yang dihasilkan, equalizer analog memiliki beberapa kekurangan, antara lain equalizer analog juga lebih rentan terhadap noise, hal ini dikarenakan komponen elektronik yang digunakan kebanyakan adalah komponen analog, sehingga sering kali sinyal akan lebih mudah tercampur dengan noise yang berasal dari panas komponen, atau noise elektrostatis. Equalizer analog juga rentan terhadap gangguan distorsi, hal ini disebabkan karena komponen analog biasanya memiliki karakteristik yang nonlinear. Hal ini dapat menyebabkan terjadinya harmonic distortion.*

*Penggunaan slider sebagai pengontrol juga membawa permasalahan tersendiri bagi operator. Salah satu masalah yang sering ditemui adalah kesulitan dalam pengembalian parameter slider ke pengaturan awal setelah adanya perubahan. Sebagai contoh apabila posisi slider dari 2 dB*

*dipindah ke 3 dB, bila dikembalikan lagi ke posisi 2 dB, posisi slider sering tidak kembali tepat pada posisi awal.*

*Selain itu salah satu masalah yang sering terjadi pada penggunaan equalizer, adalah pada bagian pengontrol equalizer yang masih menggunakan pengontrol mekanik dengan menggunakan slider untuk memotong atau menguatkan sinyal. Masalah yang terjadi adalah, seringkali slider yang digunakan menjadi tidak responsive atau slider akan menjadi rusak karena alat sudah lama atau sering digunakan. Hal ini terjadi karena perangkat mekanik akan lebih cepat mengalami kerusakan akibat seringnya terkena gesekan, tekanan, atau pengaruh korosi. Masalah ini sering terjadi pada equalizer analog dan equalizer digital yang masih menggunakan pengontrolan analog (slider).*

*Selain itu dimensi dari equalizer yang menggunakan banyak slider akan sangat besar, terutama pada equalizer 2/3 oktaf, Karena jumlah slider yang mencapai 15 buah untuk satu sisi. Jumlah slider yang banyak disebabkan karena equalizer ini memiliki 2 frekuensi tengah di dalam setiap oktaf. Berdasarkan standar ISO perhitungan dimulai dari 1000 Hz, naik sampai 20 KHz dan turun sampai 20 Hz, maka di antara 20Hz – 20 KHz terdapat 10 oktaf. Bila dilakukan*

perhitungan dengan menggunakan standar ISO maka akan didapat 15 frekuensi tengah dari equalizer 2/3 oktaf.

Permasalahan yang ada pada penggunaan *equalizer* analog, dapat diselesaikan dengan mengubah rangkaian analog yang digunakan pada *equalizer* analog menjadi rangkaian digital. Sistem digital yang digunakan meliputi bagian pemrosesan sinyal sampai pada pengoperasian alat. Sinyal audio yang masuk diubah menjadi sinyal digital dengan menggunakan ADC (*analog to digital converter*). Sinyal digital ini diproses secara digital melalui persamaan filter untuk memberikan penguatan atau atenuasi pada frekuensi-frekuensi tertentu dan kemudian sinyal tersebut diubah lagi menjadi sinyal analog dengan menggunakan DAC (*digital to analog converter*). Penguatan atau atenuasi dilakukan sesuai dengan input data gain atau atenuasi yang diatur oleh operator melalui komputer.

Untuk dapat melakukan pemrosesan sinyal digital yang diinginkan, diperlukan suatu sistem mikrokontroler yang mampu untuk melakukan pemrosesan sinyal digital dengan cepat. Agar proses pengolahan sinyal dapat bekerja dengan maksimal dibutuhkan sistem mikrokontroler yang mampu melakukan proses penyamplingan dan pemrosesan sinyal pada bagian yang berbeda. Karena itu digunakanlah *board* OMAP-L137 EVM. *Board* ini dilengkapi dengan *codecaudio* sehingga proses pengambilan data ADC dapat dilakukan secara terpisah dengan prosesor utama sehingga proses pengolahan sinyal dapat bekerja dengan maksimal karena kerja dari prosesor utama di *board* ini terfokus pada pemrosesan sinyal dengan filter. Masalah yang muncul dalam pengerjaan alat adalah:

1. Bagaimana cara mengontrol *Analog to Digital converter* (ADC) dan *Digital to Analog Converter* (DAC) dari modul.
2. Melakukan pemrograman pada modul OMAP-L137 EVM.
3. Membuat *interface* pada komputer dan mengkomunikasikan antara *interface* dengan *board*.

Pembahasan masalah dalam program ini memiliki batasan-batasan sebagai berikut:

1. Menggunakan layar monitor sebagai *interface* operator dengan *Graphicequalizer*.
2. Jenis *equalizer* yang dibuat adalah *graphicequalizer* 2/3 oktaf.
3. Jenis *filter* yang digunakan adalah *filterInfiniteImpulseResponse* (IIR).

4. Tampilan alat menggunakan *slider* dan terdapat parameter dB dan frekuensi pada *slider*.

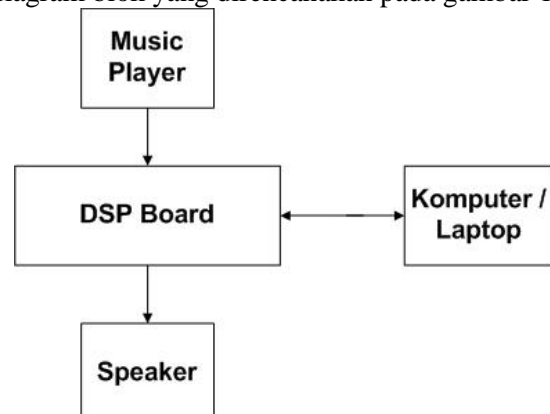
5. *Graphicequalizer* dibuat dengan menggunakan *board* OMAP-L137 EVM.

Tujuan yang hendak dicapai dalam pembuatan skripsi ini ialah membuat *equalizer* digital yang dikontrol dengan menggunakan komputer.

## II. METODE PENELITIAN

### II.1. Perancangan Alat Secara Garis Besar

Secara garis besar perancangan alat mengikuti diagram blok yang direncanakan pada gambar 1.



Gambar 1. Diagram blok alat secara garis besar

Pada alat yang akan dibuat, DSP board akan difungsikan sebagai *graphicequalizer*. DSP board yang digunakan adalah *board* OMAP-L137 EVM. *Board* ini akan menerima sinyal *input* yang berasal dari *music player*. Di dalam *Board* DSP, musik yang masuk akan disampling dengan frekuensi penyamplingan sebesar 48 KHz. Kemudian data musik akan dilewatkan serangkaian persamaan *filter* dengan tujuan agar dapat dilakukan proses pemotongan atau penguatan amplitudo sinyal pada pita frekuensi-frekuensi tertentu. Untuk mengontrol *equalizer* digunakan komputer / *laptop* sebagai *interface* antara operator dengan *equalizer*. Setelah data musik selesai dikontrol di dalam *board*, maka data dikeluarkan dan diubah menjadi sinyal analog lewat DAC yang ada di dalam *board* dan dimasukkan ke *speaker*.

*Equalizer* yang dibuat adalah jenis *graphicequalizer* 2/3 oktaf, dimana *equalizer* ini terdiri dari 15 band frekuensi. Untuk 2/3 oktaf *equalizer*, besar *Qfactor* ditentukan sebesar 2,146. Untuk pembuatan *filter* yang digunakan pada *equalizer*, akan digunakan jenis *filter* IIR. Untuk rancangan awal, *filter* akan dibuat dengan menggunakan persamaan *butterworth* atau *besel*. Hal ini dikarenakan kedua persamaan ini

memiliki pergeseran fase yang stabil dan *slopefilter* yang cukup untuk *equalizer*.

Dalam mendesain suatu *equalizer*, agar didapat *equalizer* yang baik, maka hal-hal yang perlu diperhatikan adalah *equalizer* yang dibuat sebaiknya tidak menghasilkan *noise* tambahan pada sinyal yang diproses, selain itu *equalizer* yang dibuat juga sebaiknya tidak membuat sinyal yang diproses menjadi mengalami distorsi. Selain itu *equalizer* juga harus dapat memotong dan menguatkan amplitudo dari sinyal yang diproses.

## II.2. Perancangan *Equalizer*

Pada pembuatan *equalizer* untuk keperluan tugas akhir ini, perancangan *equalizer* disesuaikan dengan standar dari *equalizer* yang beredar dipasaran, yaitu mengikuti standar IEC 61260/1260 : 1995, ISO 266 : 1997, ISO 3 : 1973, rumus perhitungannya dapat dilihat pada bagian tinjauan pustaka. Tabel 2 di bawah ini adalah tabel frekuensi tengah dan *bandwith* dari *equalizer*.

**Tabel 1. Daftar frekuensi tengah dan *bandwith* tiap frekuensi**

Freq tengah	Batas bawah	Batas atas
25	19.95	31.62
40	31.62	50.12
63	50.12	79.43
100	79.43	125.89
160	125.89	199.53
250	199.53	316.23
400	316.23	501.19
630	501.19	794.33
1000	794.33	1258.93
1600	1258.93	1995.26
2500	1995.26	3162.28
4000	3162.28	5011.87
6300	5011.87	7943.28
10000	7943.28	12589.25
16000	12589.25	19952.62

## II.3. Perancangan *filter*

*Filter* yang digunakan dalam perancangan *equalizer* ini adalah jenis *filter* IIR. Pada *bandpassfilter*, persamaan *zdomain* akan memiliki bentuk:

$$H_z = \frac{b_0 + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}$$

Untuk melakukan perhitungan koefisien dari *filter* digunakan rumus:

$$\omega_1 = 2 \cdot \pi \cdot f_{low}$$

$$\begin{aligned} \omega'_1 &= \tan \frac{\omega_1}{2 \cdot f_s} \\ \omega_2 &= 2 \cdot \pi \cdot f_{c_{high}} \\ \omega'_2 &= \tan \frac{\omega_2}{2 \cdot f_s} \\ \omega_0^2 &= \omega'_1 \cdot \omega'_2 \\ W &= \omega'_2 - \omega'_1 \\ a_0 &= 1 + \omega_0^2 + W \\ a_1 &= \frac{-2 + (\omega_0^2 \times 2)}{a_0} \\ a_2 &= \frac{1 - W + \omega_0^2}{a_0} \\ b_0 &= \frac{a_0}{W} \\ b_2 &= -\frac{W}{a_0} \end{aligned}$$

Dengan menggunakan rumus yang digunakan di atas, akan didapatkan nilai pada table 3 di bawah ini

**Tabel 2. Koefisien *bandpass filter***

Frek. tengah	b1	b2	a0	a2
25	-1.99846	0.99847	0.0007632	-0.00076
40	-1.99755	0.99758	0.0012094	-0.00120
63	-1.9961	0.99617	0.0019147	-0.00191
100	-1.99377	0.99394	0.0030316	-0.00303
160	-1.98998	0.99041	0.0047966	-0.00479
250	-1.98377	0.98484	0.0075802	-0.00758
400	-1.97339	0.97608	0.0119614	-0.01196
630	-1.95566	0.96235	0.0188271	-0.01882
1000	-1.92435	0.94096	0.0295195	-0.02951
1600	-1.86703	0.90798	0.0460109	-0.04601
2500	-1.7581	0.85782	0.0710897	-0.07108
4000	-1.54525	0.78309	0.1084565	-0.10845
6300	-1.1286	0.67469	0.1626542	-0.16265
10000	-0.35899	0.52229	0.238855	-0.2388
16000	0.786004	0.31314	0.3434306	-0.34343

Untuk mengaplikasikan persamaan *timedomain* dari *filter*, maka pada digunakan bentuk *direct form I structur*, sehingga persamaan *filter* menjadi:

$$y(n) = b_0 x(n) + b_2 x(n-2) - a_1 y(n-1) - a_2 y(n-2)$$

#### II.4. Perancangan program *equalizer*

DSP board akan diprogram untuk dapat melakukan pemrosesan terhadap sinyal yang didapat dari *music player*. Hal pertama yang dilakukan oleh program adalah melakukan inisialisasi dan *konfigurasi* terhadap board DSP.

Hal pertama yang dilakukan adalah inisialisasi yang bertujuan untuk melakukan pengaturan terhadap komponen-komponen yang akan digunakan dalam *board*. Berikutnya tahap yang dilakukan adalah konfigurasi terhadap AIC 3106. Pada bagian ini adalah tahap pengaturan ADC dan DAC. Hal-hal yang dipersiapkan pada bagian ini antara lain pengaturan frekuensi sampling ADC dan DAC menjadi 48 KHz. Untuk mengatur sampling frekuensi ( $F_s$ ) dari ADC dan DAC terlebih dahulu perlu dilakukan pembagian terhadap frekuensi *clock* sinyal. Pembagian ini dilakukan untuk mencari frekuensi *samplingreference* bagi *codec*. Rumus yang digunakan untuk mengatur frekuensi *sampling* adalah:

$$F_{S_{ref}} = \frac{24.576MHz}{(128 \times Q)}$$

Nilai  $Q$  dapat diubah pada register 3 page 0 pada bit 3-6. Dimana pada persamaan diatas nilai  $Q$  dibuat menjadi 4, maka akan didapat *samplingreference* sebesar 48 KHz. Berikutnya untuk membuat frekuensi sampling ADC dan DAC menjadi 48 KHz, kita perlu melakukan pengaturan terhadap koefisien pembagi yang ada pada register 2 page 0. Pada dasarnya bila register ini tidak di seting maka *codec* akan membuat frekuensi ADC dan DAC menjadi sama dengan frekuensi *samplingreference*. Dengan adanya register ini kita dapat membuat  $F_s$  dari ADC dan DAC menjadi sama atau berbeda atau kita dapat mengubah frekuensi sampling dengan lebih mudah tanpa harus menghitung nilai  $Q$ . Selain itu pengaturan jalur clock yang digunakan, pengaturan *modecontroller* menggunakan I2S, pengaturan *gain*, *digitalvolume*, dan jalur data ADC dan DAC, dan juga menyiapkan ADC dan DAC untuk siap bekerja.

Setelah konfigurasi pada *codecaudio* selesai berikutnya adalah tahap inisialisasi McASP. McASP adalah jalur data antara *codec* dengan prosesor OMAP L137. Pada bagian ini dilakukan pengaturan terhadap Tx dan Rx yang mana pengaturan tersebut meliputi pengaturan frekuensi *transfer* dari McASP. Setelah itu dilakukan tahap konfigurasi terhadap McASP inisialisasi *serial register*, *global register*,

*starting serial clock*, *inisialisasi state machine*, dan *frame sync*.

Tahap ketiga adalah tahap inisialisasi UART yang digunakan untuk melakukan komunikasi antara PC (sebagai *interface*) dengan *board* OMAP.

Tahap yang dilakukan berikutnya adalah tahap deklarasi *variable* dimana setiap *variable* yang akan digunakan dalam perhitungan *filter* dan pengiriman data dideklarasikan secara lokal. Setelah tahap deklarasi selesai maka selanjutnya adalah tahap pengambilan data penguatan atau pelemahan sinyal dari interface PC melalui jalur komunikasi serial, yang mana data ini berguna untuk menentukan gain atau atenuasi dari frekuensi.

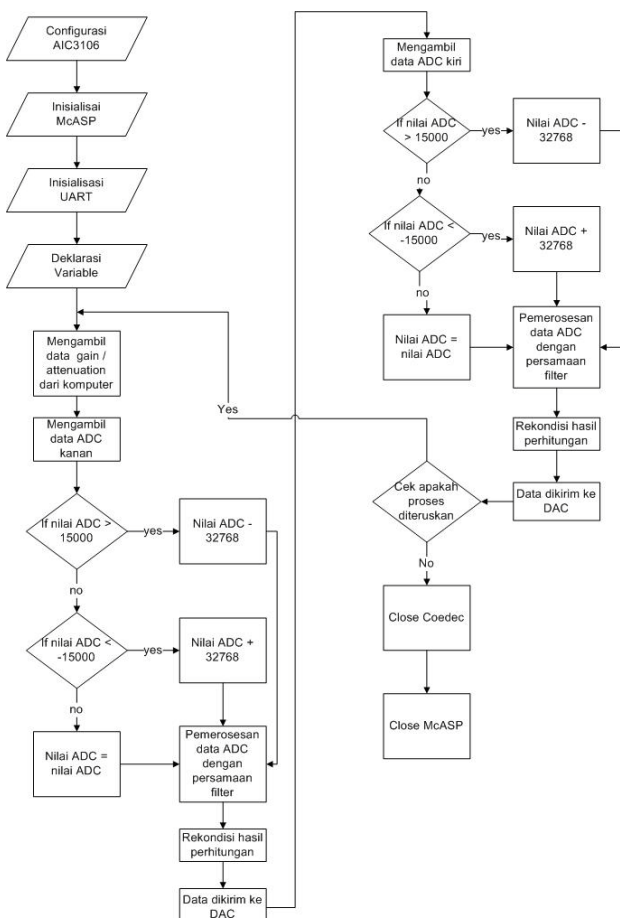
Setelah itu mulailah bagian pengambilan data ADC dan pemrosesan data. Data dari ADC kanan dan kiri dikirimkan secara bergantian oleh McASP. Setelah data didapat, data harus direkondisi terlebih dahulu, proses rekondisi meliputi pergeseran bit sebanyak 16 bit didalam register 30 bit. Hal ini perlu dilakukan karena data yang dikirim dari McASP adalah data 32 bit, yang mana 16 awal (MSB – bit ke 17) adalah data yang sebenarnya, dan 16 bit berikutnya (bit ke 16 – LSB) adalah data yang akan dikirim oleh McASP berikutnya. Setelah proses rekondisi selesai, tahap selanjutnya adalah proses pengecekan *error* dari data, yang mana bila data ADC mencapai nilai  $> 15.000$ , maka data akan dikurangkan dengan 32768, sedangkan bila data memiliki nilai  $< -15.000$ , maka data ADC akan ditambah dengan nilai 32768. Batas nilai ADC sebesar 15.000 diambil setelah dilakukan percobaan terhadap nilai maksimal dari *output* DAC sebelum terjadi *clippingsinyal*.

Bila data ADC sudah benar, maka data akan diproses dengan menggunakan persamaan filter. Pada bagian perhitungan digunakan 15 persamaan filter yang bekerja secara *parallel*. Pada tahap pemfilteran, sinyal akan mengalami penguatan atau atenuasi sesuai dengan data yang didapat dari *interface*. Setelah itu tiap hasil perhitungan dijumlahkan dan kemudian hasil tersebut direkondisi kembali menjadi format yang dapat dikenali oleh DAC. Setelah proses rekondisi selesai, data dikirimkan kembali ke DAC melalui McASP untuk diubah ke bentuk sinyal.

Setelah data ADC kanan dikirim ke DAC, maka tahap berikutnya adalah tahap mengambil data ADC kiri, setelah data dari ADC kiri didapat, data akan diproses dengan langkah yang sama seperti data dari ADC kanan sampai

akhirnya data akan dikirim ke DAC untuk diubah menjadi sinyal analog.

Setelah data kiri selesai dikirim ke DAC maka akan dilakukan pengecekan apakah proses pengambilan data diteruskan atau tidak. Apabila proses diteruskan, maka program *looping* kembali ke proses pengambilan data *gain* dan ADC. Apabila proses dihentikan, maka akan dilakukan proses mematikan *codec* dan *McASP*. Hal ini perlu dilakukan agar modul *McASP* tidak menjadi *error*. Gambar 2 adalah gambar diagram blok alur jalannya program secara keseluruhan.



Gambar 2. Diagram blok alur jalannya program

## II.5. Perancangan program *interface*

Untuk dapat melakukan pengendalian terhadap *equalizer* pada *board* DSP, yaitu untuk melakukan penguatan atau pelemahan terhadap sinyal atau lagu yang dimainkan, diperlukan suatu *interface* dimana operator dapat mengoperasikan *equalizer* secara mudah.

Berikut ini adalah penjelasan mengenai fungsi-fungsi dari tiap bagian pada *interface*:

### 1. Tab *Chart-Status*

komputer pada *controlpanel* atau dengan menyesuaikan *port* dari program. Setelah itu

Adalah bagian untuk men-switch tampilan yang terlihat pada bagian atas *interface*.

### 2. Tombol “*Save*”

Untuk menyimpan parameter yang telah dibuat.

### 3. Tombol “*Load*”

Untuk menampilkan kembali *user* setting yang telah dibuat pada bagian *slider*.

### 4. Tombol “*FlatEQ*”

Untuk mengembalikan seluruh parameter *slider* ke titik 0 dB.

### 5. Tombol “*JoinEQ*”

Untuk melakukan pengontrolan *slider* kanan dan kiri secara bersamaan.

### 6. Tombol “*FlatEQ*”

Untuk mengembalikan seluruh parameter *slider* ke titik 0 dB.

### 7. Tombol “*JoinEQ*”

Untuk melakukan pengontrolan *slider* kanan dan kiri secara bersamaan.

### 8. Tombol “*BypassEQ*”

Untuk menonaktifkan parameter *gain* atau *attenuation* dari *equalizer* tanpa mengubah nilai *gain* atau *attenuation* yang telah diseting.

### 9. Tombol “*Duplicate*”

Untuk menduplikasi parameter yang dari salah satu sisi (sisi kiri atau kanan). Untuk melakukan duplikasi, operator harus memilih sisi mana yang akan diduplikat dengan memilih salah satu radio button yang ada di samping tombol “*duplicate*”.

### 10. RadioButton “*left to right / right to left*”

Melakukan pemilihan terhadap sisi yang akan diduplikasi.

### 11. *SliderLeft&right*

*Slider* berfungsi untuk mengatur besar *gain* atau *attenuasi* yang diberikan pada filter yang terdapat pada DSP Board

### 12. *Textbox*

Merupakan besar *gain / attenuasi* yang diatur pada tiap *slider*. Tampilan angka yang terbaca pada text box adalah nilai *gain* atau *attenuasi* dalam dB.

### 13. *Slider “Gain”*

*Slider* yang berfungsi untuk mengatur nilai *gain* yang diberikan pada keseluruhan filter.

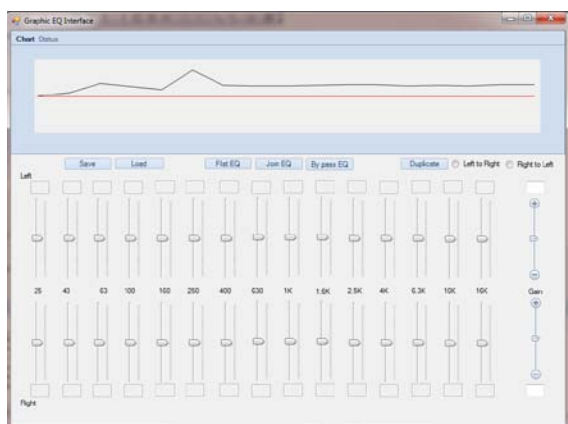
*Interface* dari *equalizer* dibuat dengan menggunakan bahasa C#. Terlebih dahulu komunikasi serial perlu di seting pada *port* 1. Bila *port* dari komunikasi serial tidak berada di *port* 1, maka dapat dilakukan penggantian *port* program akan menunggu sampai operator melakukan perubahan atau pengaturan pada

equalizer. Apabila *slider* pada program diubah, maka alamat *slider* yang berubah akan disimpan pada *variable* “*slider*”, dan nilai dari *slider* akan disimpan dalam *variable* R atau L. Setelah itu alamat dari *slider* dan nilai dari *slider* akan dikirim dengan menggunakan komunikasi serial ke dalam OMAP. Data akan terus dikirimkan sampai komputer menerima balasan dari OMAP.

Apabila operator menekan tombol “*Save*” maka nilai dari setiap *slider* akan disimpan dalam *variable* di dalam *applicationsetting*. Apabila operator menekan tombol “*Load*” maka nilai dari *slider* akan berubah sesuai nilai *variable* pada *applicationsetting*.

Bila operator menekan tombol “*joinEQ*”, maka program *interface* akan membuat *sliderleft* dan *right* memiliki nilai yang sama saat digerakan. Apabila operator menekan tombol “*flat EQ*”, maka program *interface* akan mengirimkan perintah untuk membuat semua nilai *gain* dan *attenuation* dari *filter* menjadi 1 dan membuat semua nilai dari *slider*.

Tombol “*BypassEQ*” berguna untuk membuat *equalizer* bekerja dengan penguatan atau *attenuasi* sebesar 0dB tanpa mengubah nilai penguatan yang telah diatur, sehingga saat mode *bypass* di nonaktifkan, maka *equalizer* akan kembali bekerja sesuai penguatan yang telah diatur. Tombol *duplicate* berfungsi untuk menyamakan nilai dari salah satu sisi *slider* (kanan atau kiri) ke *slider* di sisi yang lain. Untuk menjalankan fitur ini, terlebih dahulu operator harus memilih sisi mana yang akan diduplikat dengan memilih salah satu *radiobutton* yang ada di sebelah tombol *duplicate*.



Gambar3. Tampilan *interface equalizer*

### III. HASIL PENELITIAN DAN PEMBAHASAN

Pengukuran dan pengujian alat dilakukan untuk mengetahui kinerja alat. Pengukuran dan

pengujian dilakukan dengan beberapa cara, antara lain:

- Pengukuran amplitudo gelombang setelah dilewatkan filter dengan menggunakan oscilloscope.
- Pengukuran penguatan atau pelemahan sinyal.
- Pengujian alat dengan menggunakan lagu.

#### III.1. Pengukuran dengan menggunakan *oscilloscope*

Bagian ini berisi hasil pengukuran gelombang sinus pada frekuensi tertentu yang telah dilewatkan/diproses dengan menggunakan persamaan *filter*. Tujuan dari pengukuran *filter* dengan cara memberikan masukan gelombang sinus adalah untuk mengetahui kinerja dari persamaan *filter* yang digunakan pada *equalizer*. Adapun hasil yang didapat dan yang ditampilkan dibawah ini merupakan hasil dari perancangan *filter* yang dibuat hingga *equalizer* dapat bekerja dengan baik. Tabel 3 berikut ini adalah tabel frekuensi puncak dari *filter*. Tabel 4 adalah hasil dari pengukuran *cutoff* batas bawah dari *filter* dan tabel 5 adalah hasil dari pengukuran *cutoff* batas atas dari *filter*.

Tabel 3. Frekuensi puncak *filter*

Perancangan	Realisasi	Error %
25 Hz	25 Hz	0 %
40 Hz	40 Hz	0 %
63 Hz	63 Hz	0 %
100 Hz	100 Hz	0 %
160 Hz	160 Hz	0 %
250 Hz	250 Hz	0 %
400 Hz	400 Hz	0 %
630 Hz	630 Hz	0 %
1000 Hz	1000 Hz	0 %
1600 Hz	1600 Hz	0 %
2500 Hz	2500 Hz	0 %
4000 Hz	4000 Hz	0 %
6300 Hz	6465 Hz	2,6 %
10000 Hz	11169 Hz	11,69 %
16000 Hz	16024 Hz	0.15 %

Tabel 4. Frekuensi *cutoff* batas bawah *filter*

Perancangan	Realisasi	Error %
19.9 Hz	17.32 Hz	13%
31.62 Hz	26.84 Hz	15%
50.12 Hz	38.27 Hz	24%
79.43 Hz	64.9 Hz	18%
125.89 Hz	96.76 Hz	23%
199.53 Hz	138.19 Hz	31%
316.23 Hz	256.72 Hz	19%
501.19 Hz	371.3 Hz	26%
794.33 Hz	600.58 Hz	24%
1258.93 Hz	938.9 Hz	25%

1995.26 Hz	1177.1 Hz	41%
3162.28 Hz	2341.3 Hz	26%
5011.87 Hz	3192.4 Hz	36%
7943.28 Hz	5404 Hz	32%
12589.25 Hz	11027 Hz	12%

**Tabel5.Frekuensi *cutoff* batas atas *filter***

Perancangan	Realisasi	Error %
31.6 Hz	38.08 Hz	21%
50.1 Hz	60 Hz	20%
79.43 Hz	117.8 Hz	48%
125.89 Hz	183 Hz	45%
199.53 Hz	272.8 Hz	37%
316.23 Hz	374.46 Hz	18%
501.19 Hz	708.35 Hz	41%
794.33 Hz	1085 Hz	37%
1258.93 Hz	1683 Hz	34%
1995.26 Hz	2509 Hz	26%
3162.28 Hz	3380.3 Hz	7%
5011.87 Hz	6702.8 Hz	34%
7943.28 Hz	12083 Hz	52%
12589.25 Hz	17330 Hz	11,69 %
19952.62 Hz	19968.2 Hz	0.15 %

Berdasarkan hasil pengukuran respon frekuensi persamaan *filter* dengan menggunakan excel dan pengukuran *filter* dengan memberikan input gelombang sinus pada persamaan, didapati bahwa terdapat ketidak samaan *cutoff* frequency antara *cutofffilter* yang dituliskan dalam perancangan alat di bab 3 dengan *cutofffilter* yang didapat dari hasil pengukuran. Ketidak samaan *cutoff* ini bukanlah merupakan suatu kesalahan perhitungan atau kesalahan yang dibuat secara tidak sengaja, ketidak samaan ini dilakukan dengan tujuan membuat *equalizer* dapat bekerja dengan lebih baik.

Pergeseran *cutoff* yang dilakukan didasari karena adanya perbedaan fase antara sinyal yang dihasilkan oleh tiap *filter*. Perbedaan fase ini menyebabkan respon frekuensi dari *equalizer* menjadi tidak *flat* terutama pada perbatasan diantara *cutoff 2 filter*. Apabila kita melakukan penjumlahan sinyal yang memiliki frekuensi pada batas *cutoff* dari dua *filter*, maka hasil penjumlahan kedua sinyal seringkali hanya menghasilkan amplitudo setengah dari sinyal yang seharusnya dihasilkan apabila kedua sinyal memiliki *fase* yang sama. Bahkan pada beberapa bagian penjumlahan antara kedua sinyal menyebabkan terjadinya pelemahan *amplitudo* sinyal yang dikarenakan perbedanan fase antara

kedua hasil perhitungan *filter* mendekati 180 derajat.

Oleh karena itu untuk membuat respon frekuensi dari *equalizer* menjadi mendekati *flat*, maka dilakukan pelebaran *cutoff* frekuensi dari *filter* dengan tujuan mengkompensasi efek dari ketidak samaan fase sehingga penjumlahan antara kedua *filter* dapat menghasilkan amplitudo sinyal yang mendekati nilai yang seharusnya.

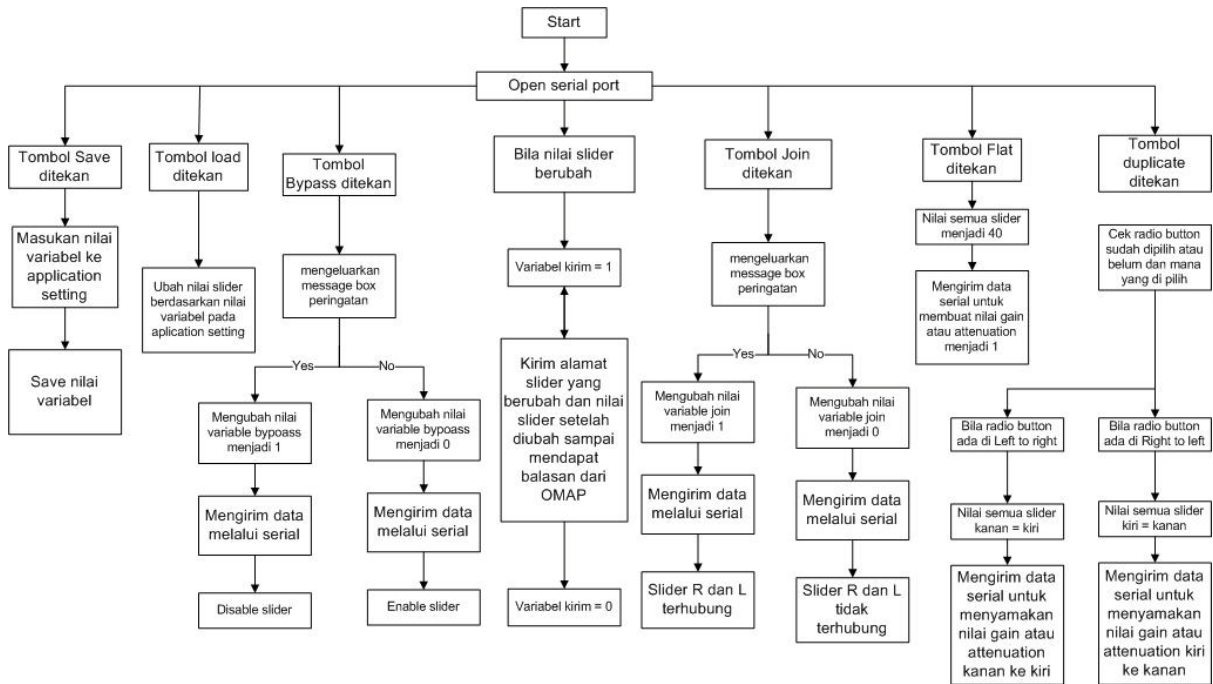
### III.2. Pengukuran Penguatan *filter*

Operator dapat melakukan penguatan dan pelemahan dari sinyal dengan cara mengubah posisi *slider* yang ada pada tampilan *interface* di komputer. Berikut ini adalah hasil pengukuran sinyal setelah mengalami penguatan, gambar 5 adalah gambar sinyal saat mengalami penguatan 0 dB dimana amplitudo sinyal adalah 400 mV, gambar 6 adalah gambar sinyal setelah mengalami penguatan 3 dB, dan gambar 7 adalah gambar sinyal setelah mengalami penguatan sebesar 6 dB.

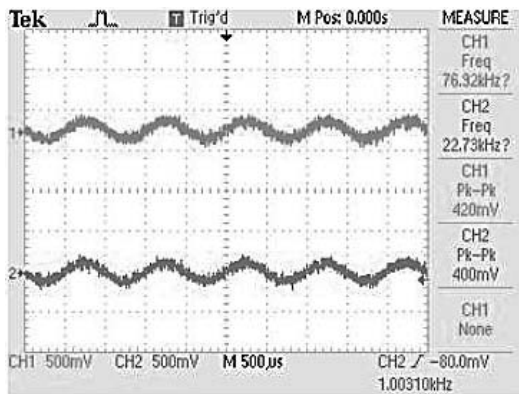
Untuk hasil pengukuran sinyal setelah mengalami pelemahan dapat dilihat pada gambar 8 adalah gambar sinyal saat mengalami pelemahan 0 dB dimana amplitudo sinyal adalah 1500 mV *peak to peak*, gambar 9 adalah gambar sinyal setelah mengalami pelemahan -3 dB, dan gambar 10 adalah gambar sinyal setelah mengalami pelemahan sebesar -6 dB.

### III.3. Pengujian dengan menggunakan lagu

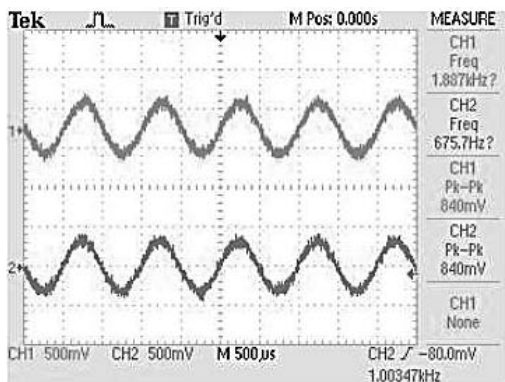
Kegunaan utama dari sebuah *equalizer* adalah untuk melakukan pemrosesan sinyal *audio*, oleh karena itu perlu dilakukan pengujian terhadap *equalizer* dengan memberikan *input* berupa musik. Saat musik dimainkan dengan menggunakan *equalizer*, dan dilakukan pengujian dengan cara memberikan tambahan *gain* pada frekuensi tertentu, maka suara yang terdengar dari *speaker* mengalami perubahan pada beberapa bagian tertentu. Saat dilakukan penguatan (*boosting*) sinyal pada fekuensi 100 Hz, maka suara *bass* dan *beat* dari lagu terdengar mengalami peningkatan. Begitu juga saat dilakukan pelemahan (*cutting*) pada frekuensi 100 Hz maka suara *bass* atau *beat* dari musik mengalami pelemahan. Begitu juga saat dilakukan penguatan pada frekuensi 10 KHz atau 16 KHz, suara *cymbal* dari lagu menjadi lebih tajam.



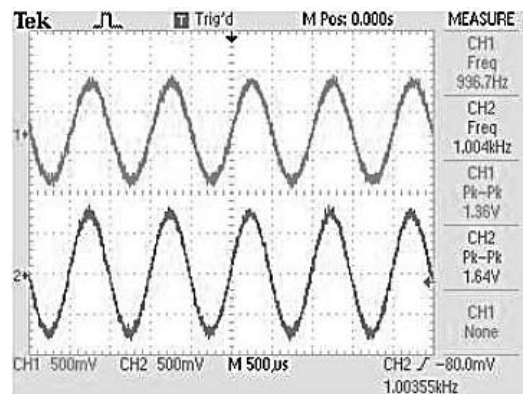
Gambar 4. Diagram blok pemrograman interface



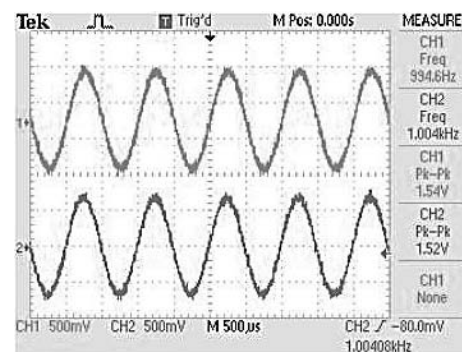
Gambar 5. Penguatan 0dB



Gambar 6. Penguatan 3dB

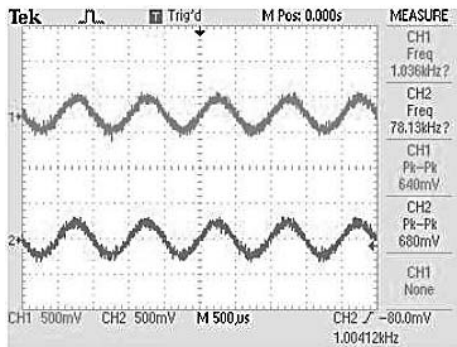


Gambar 7. Penguatan 6dB

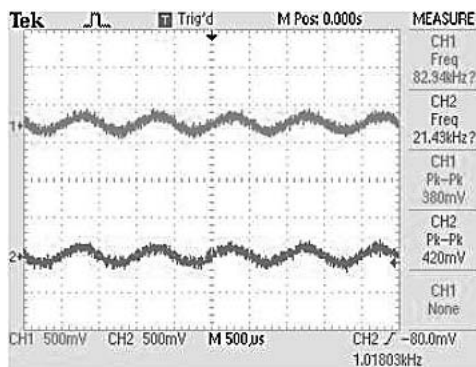


Gambar 8. Pelemahan 0dB





Gambar 9. Pelemahan -3dB



Gambar 10. Pelemahan -6dB

## Kesimpulan

Berdasarkan hasil perancangan, pembuatan, dan pengukuran, maka dapat diambil kesimpulan:

Dengan menggunakan beberapa *bandpassfilter* yang memiliki frekuensi *cutoff* yang dirancang dan diatur dengan benar, kita dapat membuat suatu *equalizer* yang dapat bekerja dengan baik.

Berdasarkan hasil pengujian dengan menggunakan lagu, maka dapat disimpulkan bahwa *equalizer* yang telah dibuat dapat bekerja dengan baik untuk mengubah karakteristik suara dari lagu.

Program *interface* yang dibuat untuk mengontrol *equalizer* dapat menjadi sarana bagi operator untuk mengatur penguatan dan pelemahan dari *equalizer* dan program *interface* yang dibuat dapat bekerja dengan baik untuk mengontrol kinerja *equalizer*.

## Daftar pustaka

- [1] *Board OMAP-L137*  
-, 2009, OMAP-L137 Evaluation module Technical refrence, Spectrum Digital INCORPORATED, Spectrum digital
- [2] IC AIC3106  
-, 2008, TLV320AIC3106 datasheet, Texas instrument
- [3] OMAP-L137 hardware setup

[http://processors.wiki.ti.com/index.php/OMAP\\_L137\\_EVM\\_Hardware\\_Setup](http://processors.wiki.ti.com/index.php/OMAP_L137_EVM_Hardware_Setup) (diakses tanggal 1 Maret 2013)

- [4] About C language  
<http://curiosity.discovery.com/question/c-programming> (diakses tanggal 18 September 2013)
- [5] -, 2006, TMS320C64x+ DSP Little-Endian DSP Library Programmer's Reference, Texas Instruments
- [6] Rulph Chassaing, 2002, DSP Applications Using C and the TMS320C6x DSK, Wiley-Interscience
- [7] Rulph Chassaing, 2005, Digital\_Signal\_Processing and Applications using C6713 and C6416 DSK, Wiley-Interscience
- [8] Sen M Kuo and Bob H Lee, 2001, RealTime Digital Signal Processing Implementation, Application and Experiments With TMS320C55x, John wiley & Sons, Ltd
- [9] -, 2010, TMS320C674x/OMAP-L1x Processor Universal Asynchronous Receiver/Transmitter(UART) User's guide, Texas Instruments
- [10] -, 2011, Multichannel Audio Serial Port (McASP) chapter 22, Texas Instruments
- [11] -, 2000, TMS320C6000 Code Composer Studio Tutorial, Texas Instruments
- [12] Library bahasa pemrograman C#: [http://msdn.microsoft.com/en-us/library/aa288436\(v=vs.71\).aspx](http://msdn.microsoft.com/en-us/library/aa288436(v=vs.71).aspx) (diakses tanggal 4 November 2013)
- [13] Professor Colin H Hansen, Fundamentals of acoustics, Department of Mechanical Engineering, University of Adelaide, South Australia  
([www.who.int/occupational\\_health/publications/noise1.pdf](http://www.who.int/occupational_health/publications/noise1.pdf))
- [14] Anonim, 1997, Low-Voltage 24-Bit Fet Bus-Exchange Switch datasheet, Texas Instruments