Implementasi Real Time Digital Audio *Equalizer* 4 *Band* menggunakan DSK TMS320C6713

Era Dwi Febrianti[1], Miftahul Huda[2]
Politeknik Elektronika Negeri Surabaya
Institut Teknologi Sepuluh Nopember(ITS) Surabaya
e-mail:rara.rianti@yahoo.com

Abstrak

Pada proyek akhir ini, dilakukan perancangan dan pengimplementasian secara real time sistem Digital Audio Equalizer sehingga dapat diubah frekuensi center-nya secara manual dengan menggunakan DSK (DSP Starter Kit) TMS320C6713 sebagai suatu perangkat untuk mengimplementasikan secara real time pada sistem Digital Audio Equalizer. Dalam proyek akhir ini digunakan pemrograman matlab untuk mendapatkan koefisien filter dan bahasa C sebagai bahasa pemrogramannya serta board DSK (DSP Starter Kit) TMS320C6713 telah di-download-kan program untuk proses implementasi sistem Digital Audio Equalizer.

Hasil dari proyek akhir ini, berupa sistem 4 band Equalizer yang berupa filter LPF dengan cut-off 5200Hz, filter BPF1 dengan cut-off antara 800Hz-7600Hz, filter BPF2 dengan cut-off antara 400Hz-7600Hz dan filter HPF 1600Hz cut-off yang diatur menggunakan software secara real time sehingga dapat menghasilkan suara bass, middle, trible yang sesuai dengan filter masing-masing sehingga dapat memberikan peningkatan kualitas (membatasi kebisingan dan distorsi, komponen frekuensi yang tidak diinginkan) dari output suara dan juga mengembangkan teknologi dibidang pengolahan sinyal.

Kata kunci: *Adaptif filter, Least Mean Square, TMS320C6713*

1. .Pendahuluan

Nurgiyatna, dkk[11], yang berjudul "Implementasi Filter Digital *Infinite Impulse Response* pada DSP TMS320C6711", pada penelitian ini,Filter IIR butterworth disimulasikan dengan Matlab dan diimplementasikan pada DSK TMS320C6713. Dengan orde filter, maka diperoleh data yaitu fc= 1030 Hz dan Δf=160Hz. Hasil pada simulasi dan implementasi filter menunjukkan perangkat keras yang digunakan untuk implementasi filter sudah sangat bagus dalam artian dapat menanggulangi kesalah kuantisasi ADC-DAC, kesalahan kuantisasi koefisien filter (panjang bit yang digunakan), dan kesalahan akibat pembulatan aritmatika.

Agfianto Eko Putra, Triasmono [13]. Yang berjudul "Pembuatan Ekualiser 10-*Band* Stereo Digital dengan Algoritma Penapis Lolos-pita Tanggap Impuls Tak-Hingga", ekualiser ini menggunakan algoritma Penapis Lolos-pita Tanggap Impuls Tak-hingga (IIR – *Infinite Impulse Response*). Dengan ekualiser tersebut dapat diubah penguatan pada tingkat frekuensi tertentu, sehingga dapat ditonjolkan suara bass, trebel maupun vokal dari suatu sinyal audio.

Proyek akhir ini merupakan gabungan dari 2 penelitian diatas, namun proyek akhir ini diaplikasikan menggunakan pemrograman bahasa C. Pembuatan sistem ini tidak hanya terbatas pada sinyal suara sebagai sinyal inputnya melainkan juga didapat dari gelombang sinus sebagai testing pointnya, serta didapat dari sinyal suara dengan frekensi antara 20Hz ~ 20kHz dan dari file audio yang berformat *mp3*. Dimana nantinya sistem ini tidak berbentuk simulasi melainkan diimplementasikan secara real time menggunakan *board* DSK (*DSP Starter Kit*) TMS320C6713.

Makalah proyek akhir inidisusun sebagai berikut:
Teori penunjang akan disajikan pada bab 2. Pada bab 3 akan disajikan perancangan sistem. Pada bab 4 akan disajikan pengujian dan analisis. Kesimpulan akan disajikan pada bab 5.

2.1 PERENCANAAN SISTEM DAN TEORI PENUNJANG

2.1.1 Infinite impulse response (IIR)

Filter *Infinite Impulse Response* (IIR) adalah salah satu tipe dari filter digital yang dipakai pada aplikasi *Digital Signal Processing* (DSP). Keuntungan filter IIR antara lain adalah membutuhkan koefesien yang lebih sedikit untuk respon frekuensi yang curam sehingga dapat mengurangi jumlah waktu komputasi. Fungsi transfer filter IIR dapat dilihat pada persamaan (1).

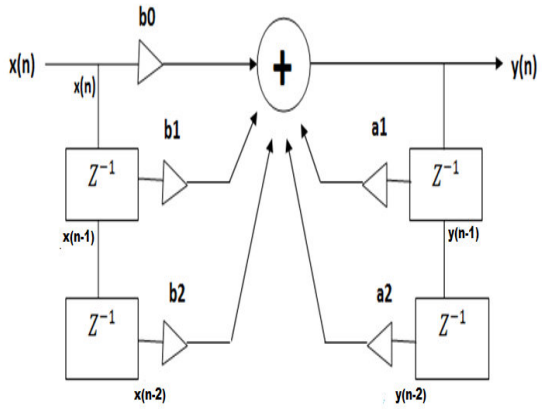
$$H(z) = \frac{bo+b_1z^{-1}+\cdots+b_Nz^{-m}}{1+a_1z^{-1}+\cdots+a_Nz^{-n}} \qquad (1)$$

dimana:
- H(z) merupakan fungsi transfer dari filter IIR
- $a_1, a_2 \ldots . a_N$ merupakan koefisien feed back dari filter IIR
- $b_0, b_1 \ldots . b_N$ merupakan koefisien feed forwad dari filter IIR

Proses pemfilteran pada sebuah sinyal akan mengikuti bentuk persamaan beda pada persamaan (2) berikut ini.

$$y[n] = \sum_{m=0}^{q} b_m\, x[n-m] - \sum_{m=1}^{p} a_m y[n-m] \quad (2)$$

Karena adanya proses feedback ini filter IIR juga dikenal sebagai recursive filter. Diagram blok untuk sebuah filter IIR dalam bentuk *direct form* I dapat digambarkan seperti gambar 1.



Gambar 1. Blok diagram filter IIR

Sebuah diagram blok dari filter IIR terlihat seperti berikut ini:
z - blok 1 adalah penundaan unit. Koefisien dan jumlah umpan balik / jalur umpan maju adalah tergantung dari implementasi. Karena adanya proses feedback ini filter IIR juga dikenal sebagai recursive filter

3. HASIL PENGUJIAN PROYEK AKHIR

3.1 *Install Code Composer Studio* (CCS) V3.1 dan Diagnostik DSK

Penginstalan *Code Composer Studio* (CCS) sangat penting, karena CCS membantu untuk membuat program. Sedangkan untuk pengecekan DSK dilakukan sebelum CCS diaktifkan. Sehingga kondisi DSK dapat diketahui.

3.2 Pembuatan Project Baru

1. Buatlah sebuah folder di direktori D untuk menempatkan semua project.
2. Buka (*Code Composer Studio*) CCS. Connect DSK dengan CCS dengan cara meilih *Debug* kemudian pilih *Connect*.
3. Buatlah project baru dengan cara buka *Project* kemudian pilih *New*. Maka akan tampil seperti gambar 2.

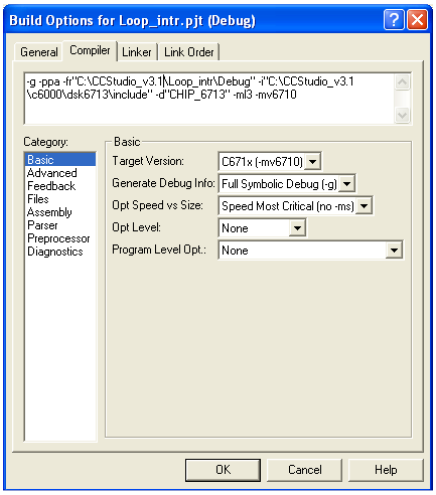
Gambar 2. New Project[4]

4. Tambahkan file-file pendukung seperti:
- C6713dskinit.c
- Vector_intr.asm
- C6713dsk.cmd
- Dsk6713_aic23.h
- File.c

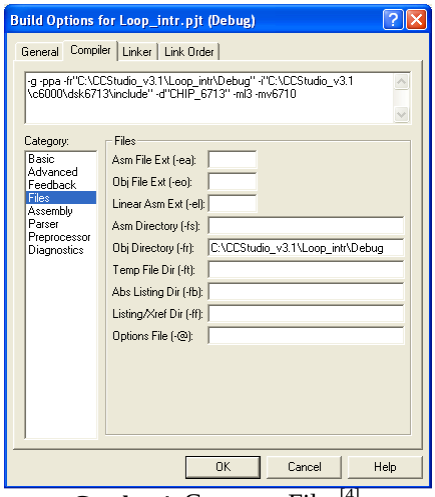
Kelima file-file diatas berada didalam folder C:\CCStudio_v3.1\loop_intr dan pilih All Files. Setelah menemukan seluruhnya, maka klik dua kali atau klik open. Selain kelima diatas, tambahkan pula

- rts6700.lib,file tersebut berada didalam folder C:\CCStudio_v3.1\c6000\cgtools\lib
- dsk6713bsl.lib,file tersebut berada didalam folder C:\CCStudio_v3.1\c6000\dsk6713\lib
- csl6713.lib,file tersebut berada didaam folder C:\CCStudio_v3.1\c6000\bios\lib

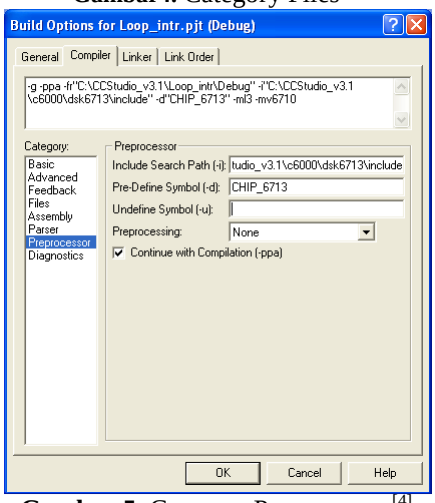
5. Setelah semua file diatas telah dimasukkan kedalam project, maka lakukan langkah-langkah ketergantungan dengan cara pilih **Project** kemudian **Scan All Files Dependencies**. Maka secara otomatis akan terhubung file-file pendukung yang diperlukan pada proyek yang telah dibuat.
6. Kemudian atur BUILD OPTION seperti gambar 3 untuk pengaturan Compiler Category Basic. Untuk Compiler Category Files seperti gambar 4 dan Category Preprocessor, set seperti gambar 5.



Gambar 3 Category Basic[4]

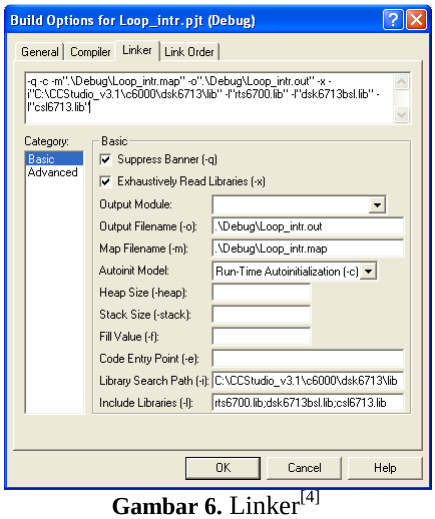


Gambar4. Category Files[4]



Gambar 5. Category Preprocessor[4]

7. Untuk Linker, set seperti gambar 6



Gambar 6. Linker[4]

8. **Rebuild All** program hingga tidak ada error yang terdeteksi. Dengan cara pilih Project pada toolbar lalu Rebuild All.
9. Load program ke DSK dengan cara buka File→Load Program. Pilih file yang berekstensi .out.
10. Jalankan program dengan pilih Debug kemudian Run.
11. Dengarkan hasilnya melalui speaker. Selain itu sinyal input dan output dapat diamati melalui oscilloscope.

4. Konsep IIR Filter

4.1 Filter IIR

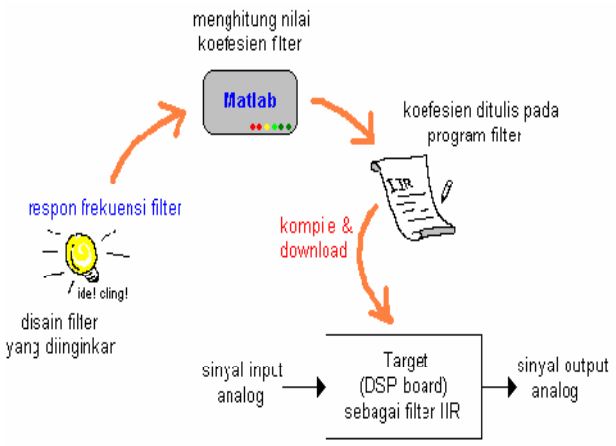
Filter IIR dari (*Infinite Impulse Response*) adalah salah satu tipe dari filter digital yang dipakai pada aplikasi *Digital Signal Processing*(DSP). Keuntungan filter IIR antara lain adalah membutuhkan koefesien yang lebih sedikit untuk respon frekuensi yang curam sehingga dapat mengurangi jumlah waktu komputasi.
Fungsi transfer filter IIR adalah

$$H(z) = \frac{b_0 + b_1 z^{-1} + ... + b_N z^{-M}}{1 + a_1 z^{-1} + ... + a_N z^{-N}} \qquad (3)$$

dimana:
- H(z) merupakan fungsi transfer dari filter IIR
- a1, a2,..., aN merupakan koefisien feed back dari filter IIR
- b0, b1,...bN merupakan koefisien feed forwad dari filter IIR

4.2.Implementasi Filter IIR pada TMS320C5402

Untuk mengimplementasikan IIR filter secara real time pada sebuah DSP Card yang dalam hal ini kita gunakan sebuah produk dari Texas Instrument, TMS32C5402 yang dilengkapi dengan Code Composer Studio (CCS) versi 2, kita dapat mengikuti langkah seperti Gambar 2.



Gambar 7. Ilustrasi alur implementasi filter IIR[13]

Langkah pertama adalah bagaimana merancang filter dengan menggunakan perangkat Lunak Matlab. Berikut contoh disain program Matlab untuk menghasilkan koefesien filter IIR low-pass dengan frekuensi cut-off 8KHz pada frekuensi sampling sebesar 16KHz. Nilai koefesien yang dihasilkan disimpan dalam file teks. Pada langkah ini akan diperoleh respon frekuensi dan koefisienfilter seperti yang diilustrasikan pada Gambar 2 tersebut.

Sebagai ilustrasi adalah perancangan low pass filter dengan spesifikasi berikut ini:
- Frekuensi sampling filter = 16KHz
- Setengah frekuensi sampling filter = 8KHz
- Frekuensi cut-off filter = 800Hz

Nilai W1 sebagai *cut-off* pada kondisi ternormalisasi terhadap setengah dari frekuensi sampling adalah:

wc=2*Fc/Fs;

Program Matlab untuk mendapatkan koefesien filter low-pass IIR orde 4 adalah:

[B,A]=butter(4,wc)

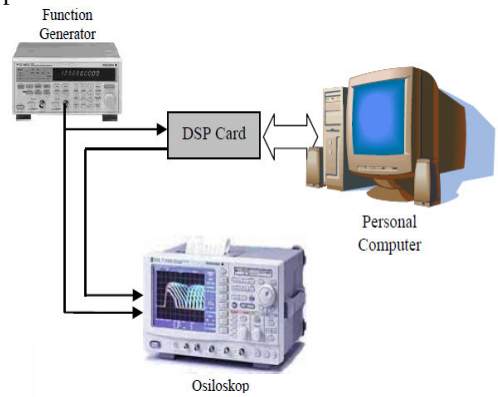
Program Matlab untuk mendapatkan koefesien filter *band*pass IIR orde 2 adalah:

[B,A]=butter(2,wc)

Program Matlab untuk mendapatkan koefesien filter highpass IIR orde 4 adalah:

[B,A]=butter(4,wc)

Proses berikutnya adalah mengcopy koefisien filter hasil perancangan ke sebuah program CCS 3.1 yang merupakan perangkat lunak pendukung DSP Card TMS 320C6713. Program iir filter dikompila si ulang ulang pada CCS 3.1 dan hasil kompilasi berupa kode hexa di downloadkan ke TMS320C6713. Sebagai langkah awal pengujian filter IIR hasil perancangan digunakan function generator yang mampu bekerja pada frekuensi suara (300 ~ 4000 Hz), dalam hal ini kita gunakan yang mampu membangkitkan sinyal dengan frekuensi kerja dari DC sampai 2 MHz. Untuk menguji hasilnya kita gunakan sebuah osiloskop yang dapat dilengkapi dengan fasilitas storage system. Lebih jelasnya bisa dilihat seperti padaGambar 4.



Gambar 8. Gambaran sistem pengujian Filter II [13]

Untuk menguji apakah sistem yang dirancang telah mampu bekerja dengan benar, masukkan sinyal sinus dari *function generator* ke DSP Card, dan amati hasil keluaran dari DSP Card. *Band*ingkan dengan sinyal asli yang berasal dari func*tion generator*. Selanjutnya nilai frekuensi sinyal input dinaikkan sedikit demi sedikit dari 100Hz sampai dengan 8KHz. Apabila terdapat perubahan nilai level sinyal output sebagai pengaruh kenaikan frekuensi sinyal input, hal inimenunjukkan sistem LPF, BPF maupun HPF yang telah menunjukkan kinerja yang benar.

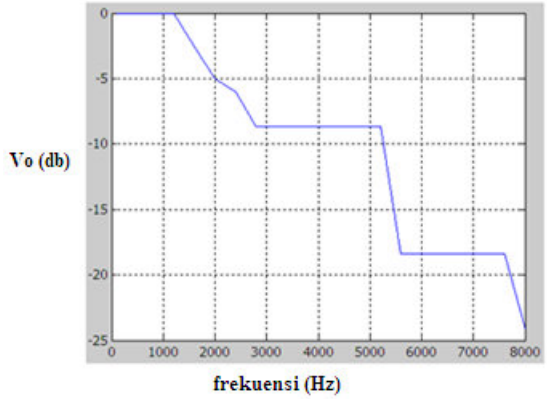
4.3. Karakteristik IIR Filter Hasil Perancangan

Sesuai dengan perancangan IIR untuk *low pass filter* (LPF), analisa yang disajikan disini juga untuk kasus LPF,BPF1, BPF2, dan HPF. Hasil pengujian terhadap sinyal input dan output menunjukkan kinerja seperti yang

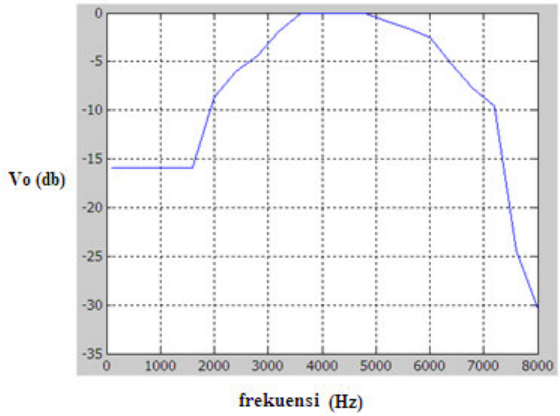
ditunjukkan pada Tabel 1 berikut ini.

No	Freq(Hz)	Vin=1 Vpp			
		LPF	BPF1	BPF2	HPF
1	100	98.06	-20	-24.43	-40
2	400	98.06	-13.97	-13.97	-40
3	800	98.06	-13.97	-6.37	-40
4	1200	98.06	-13.97	0	-40
5	1600	95.56	-13.97	0	-40
6	2000	93.06	113,06	0	-33.97
7	2400	92.04	115.56	0	-30.45
8	2800	89.54	118.06	0	-27.95
9	3200	89.54	120	0	-21.93
10	4000	89.54	121.58	0	-17.07
11	4400	89.54	121.58	0	-13.97
12	4800	87.95	121.58	0	-9.89
13	5200	86.02	121.58	0	-7.95
14	5600	86.02	121.21	2.27	-7.13
15	6000	0	120	1.93	-1.11
16	6400	0	119.08	-6.02	-1.11
17	6800	0	116.25	-10.45	-1.11
18	7200	0	113.97	-27.95	-1.11
19	7600	0	112.04	-27.95	-1.11
20	8000	0	106.25	0	-1.11

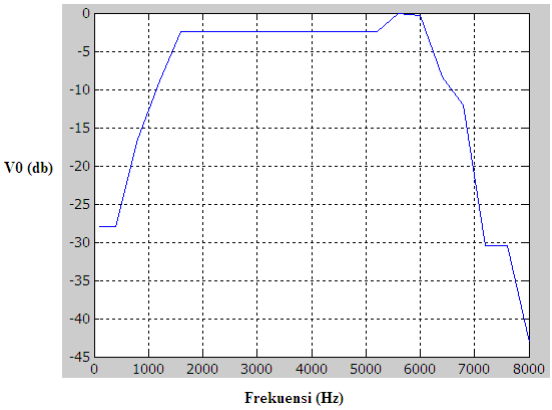
Dari Tabel 1 dapat digambarkan seperti berikut :



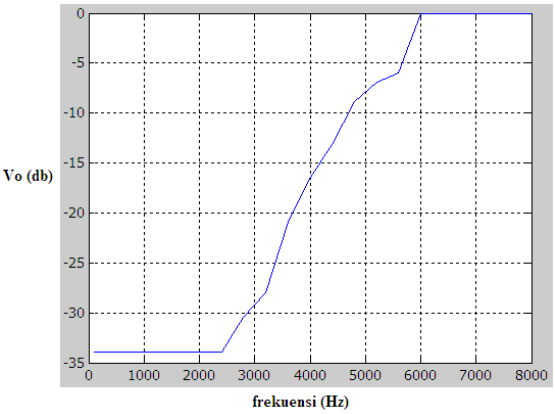
Gambar 9. Hasil pengujian filter LPF



Gambar10. Hasil pengujian filter BPF1



Gambar11. Hasil pengujian filter BPF2



Gambar12. HPF dengan cut-off 2400Hz dan 4800Hz

4.4 Aplikasi pada Gain slider

Untuk lebih yakin dengan sistem yang telah dirancang.Maka, Hasil dari aplikasi ini dianalisis dengan *speaker* untuk mendengarkan hasil dari sistem Audio Digital *Equalizer* dengan file audio berformat *mp3*. Pengujian ini bertujuan untuk mengamati suara dari file audio apabila dilakukan perubahan pada gain slidernya.

Pada saat pengujian sistem Audio Digital *Equalizer* ini, dapat dianalisis bahwa pada saat gain slider untuk LPF diaktifkan maka suara menjadi suara bass. Pada saat gain slider untuk BPF1 diaktifkan maka suara menjadi suara middle,begitu pula dengan BPF2. Pada saat gain slider untuk HPF diaktifkan maka suara menjadi suara trible. Dalam artian disini suara akan berubah sesuai dengan saat gain slider diaktifkan oleh masing-masing filter.

Referensi

[1]http://c6000.spectrumdigital.com/dsk6713/V2/docs/dsk6713_TechRef.pdf.“DSK (DSP Starter Kit) TMS 320C6713.diakses : 1 November 2010,”

[2]Tri Budi Santoso,et all. “Sinyal dan Sistem”, PENS-ITS : Surabaya.2007.

[3]http://www.teworks.comdiakses : 28 November 2010, “Komponen DSPLab”,diakses :1 November 2010

[4]Tri Budi Santoso, Miftahul Huda. “Pengolahan Sinyal Digital (Berbasis TMS 320C6713)”, Buku Petujuk Praktikum,PENS-ITS : Surabaya.2007.

[5]http://kursusaudio.wordpress.com/tag/*Equalizer*/ “*Equalizer*”. diakses : 1 November 2010

[6]http://student.eepis-its.edu/~ty2n/modul%20pengolahan%20sinyal/ps6_iir.pdf “Preset *Equalizer* menggunakan filter IIR” diakses : 2 November 2010

[7]http://dewey.petra.ac.id/dgt_res_detail.php?mode=extended&knokat=1691, “Parametric *Equalizer* 3 channel center frekuensi” .Diakses : 2 Desember 2010

[8] http://www.scribd.com/doc/26975534/Graphic-New, “Digital Graphic *Equalizer*” ,January 2009. Diakses : 30 Desember 2010

[9]http://en.wikipedia.org/wiki/Infinite_impulse_response, “Infinite impuls response (IIR)”. Diakses : 30 Desember 2010

[10]http://www.teworks.com. “Komponen DSPLab” diakses : 31 Desember 2010

[11]Nurgiyatna, et all.“Implementasi Filter Digital Infinite Impulse Response pada DSP TMS320C6711”.Surakarta. 2002

[12]http://www.minidsp.com/images/documents/Product%20Brief-4way%20Xover%20 plug-in.pdf “Audio flow chart diagram”. Diakses : 31 Desember 2010

[13]Tri Budi Santoso, Hary Octavianto, Titon Dutono.”Implementasi Filter IIR secara Real

Time pada TMS 32C5402".PENS-ITS.Surabaya.