RANCANG BANGUN *DIGITAL AUDIO EFFECT* DENGAN MENGGUNAKAN DSP *STARTER KIT* TMS320C6713 BERBASISKAN MATLAB SIMULINK

SKRIPSI

Oleh

IWAN HERDIAN

04 04 03 0512

PROGRAM STUDI TEKNIK ELEKTRO

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

GENAP 2007/2008

RANCANG BANGUN *DIGITAL AUDIO EFFECT* DENGAN MENGGUNAKAN DSP *STARTER KIT* TMS320C6713 BERBASISKAN MATLAB SIMULINK

SKRIPSI

Oleh

IWAN HERDIAN

04 04 03 0512

SKRIPSI INI DIAJUKAN UNTUK MELENGKAPI SEBAGIAN PERSYARATAN MENJADI SARJANA TEKNIK

PROGRAM STUDI TEKNIK ELEKTRO

DEPARTEMEN TEKNIK ELEKTRO

FAKULTAS TEKNIK UNIVERSITAS INDONESIA

GENAP 2007/2008

PERNYATAAN KEASLIAN SKRIPSI

Saya menyatakan dengan sesungguhnya bahwa skripsi dengan judul:

RANCANG BANGUN *DIGITAL AUDIO EFFECT* DENGAN MENGGUNAKAN DSP *STARTER KIT* TMS320C6713 BERBASISKAN MATLAB SIMULINK

yang dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia, sejauh yang saya ketahui bukan merupakan tiruan atau duplikasi dari seminar yang sudah dipublikasikan dan atau pernah dipakai untuk mendapatkan gelar kesarjanaan di lingkungan Universitas Indonesia maupun di Perguruan Tinggi atau Instansi manapun, kecuali bagian yang sumber informasinya dicantumkan sebagaimana mestinya.

Depok, 19 Mei 2008

Iwan Herdian

NPM 04 04 03 0512

PENGESAHAN

Skripsi dengan judul:

RANCANG BANGUN *DIGITAL AUDIO EFFECT* DENGAN MENGGUNAKAN DSP *STARTER KIT* TMS320C6713 BERBASISKAN MATLAB SIMULINK

dibuat untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik pada program studi Teknik Elektro Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia dan disetujui untuk diajukan dalam sidang skripsi.

Depok, 19 Mei 2008

Dosen Pembimbing

Dr.Ir. Arman Djohan.M.Eng,

NIP. 131 476 472

UCAPAN TERIMA KASIH

Puji syukur kepada Tuhan Yesus Kristus atas anugerahNya penulisan skripsi ini dapat diselesaikan dengan baik. Penulis juga mengucapkan terima kasih kepada :

Dr.Ir. Arman Djohan. M.Eng

selaku dosen pembimbing yang telah bersedia meluangkan waktu untuk memberi pengarahan, diskusi dan bimbingan serta persetujuan sehingga skripsi ini dapat selesai dengan baik.

Iwan Herdian
NPM 04 04 03 0512
Departemen Teknik Elektro

Dosen Pembimbing
Dr.Ir. Arman Djohan.M.Eng,

RANCANG BANGUN
DIGITAL AUDIO EFFECT
DENGAN MENGGUNAKAN DSP *STARTER KIT* TMS320C6713
BERBASISKAN MATLAB SIMULINK

ABSTRAK

Saat ini aplikasi di bidang audio sudah banyak menggunakan teknologi DSP. Salah satu contoh adalah beralihnya berbagai jenis efek audio dari bentuk analog menjadi efek audio yang berbasis dijital dengan menggunakan prosesor DSP. Hal ini disebabkan karena efek audio yang dirancang dengan bantuan DSP memberikan kelebihan-kelebihan seperti efisiensi perancangan dan fleksibilitas.

Efek audio dijital dapat diterapkan secara *real time* menggunakan DSP *Starter Kit* TMS320C6713 dengan mengimplementasikan SIMULINK dalam pemodelannya. Pada skripsi ini dilakukan perancangan 3 jenis efek audio yaitu *Reverberation, Echo* dan *Chorus*. Perancangan dibuat untuk menerima masukan berupa suara *tone* 1 KHz dan sinyal acak dari ADC dengan frekuensi sampling audio sebesar 44100 Hz. Lalu dikondisikan menerima efek yang berbeda-beda. Setiap jenis efek memiliki model yang berbeda-beda. Hasil dari perancangan dianalisis dengan metode FFT baik sinyal *input* ataupun *output*. Dari hasil penelitian menunjukkan bahwa perancangan model efek audio dijital dapat diterapkan dengan DSK TMS320C6713 dimana pembuatan model berbasiskan SIMULINK®. Dari hasil pengolahan data sinyal *tone* 1 KHz didapatkan penundaan yang terjadi dalam efek *reverb* adalah 38 ms dan efek *echo* 160 ms. Untuk *chorus* terdapat satu suara campuran dengan pemberian satu LFO. Untuk sinyal acak didapatkan penundaan untuk *reverb* dan *echo* adalah sama seperti *tone*. Hal ini juga berlaku untuk efek *chorus* pada sinyal acak.

Kata Kunci : SIMULINK, MATLAB, DSK TMS320C6713, *Reverberation, Echo, Chorus.*

Iwan Herdian
NPM 04 04 03 0512
Electrical Engineering Department

Counsellor
Dr. Ir. Arman Djohan M.Eng

DEVELOPING

DIGITAL AUDIO EFFECT

ON DSP STARTER KIT TMS320C6713

BY USING MATLAB SIMULINK®

ABSTRACT

DSP technology is widely used in many audio applications. One of the examples is the changing of analog audio processing into digital audio processing using the DSP processor. The reason of this changing is that digital audio processing give many advantages compare to analog such as designing efficiency and flexibility.

A real time digital audio effect can be developed on DSP Starter Kit by using SIMULINK® in modelling. In this paper 3 type of audio effects are developed, which are reverberation, echo, and chorus. The design is developed to receive one KHz tone and random signal which are the audio sampling is 44100 Hz. Then each of them is given the different type of effects. The result is analyzed by using FFT. From research show that audio digital effect can be developed on DSP Starter Kit by using SIMULINK®. From the data, show that the delay from reverb effect for one KHz tone and random signal are 38 ms. The delay from echo effect for one KHz tone and random signal are 160 ms. For chorus effect, there is one addition signal because the model is only using one LFO.

Keywords : SIMULINK, MATLAB, DSK TMS320C6713, Reverberation, Echo, Chorus.

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I

PENDAHULUAN

1.1 LATAR BELAKANG

Aplikasi dari pemrosesan sinyal dijital saat ini sangat luas dan sudah merambah ke berbagai bidang dalam kehidupan manusia. Salah satu bidang aplikasinya adalah bidang audio [1]. Saat ini, sebagian besar aplikasi audio sudah memakai teknologi dijital. Prosesor DSP sudah menjadi satu bagian terpenting dari teknologi yang biasa disebut audio dijital. Pemakaian aplikasi dari audio dijital sudah bermacam-macam, bahkan mungkin secara tidak sadar sebagian besar sudah ada di sekitar manusia. Salah satu bentuk aplikasi audio dijital adalah penerapan berbagai efek audio, dimana penerapannya selama ini lebih banyak di dunia hiburan maupun industri musik. Pemakaian efek audio ini bertujuan untuk memperbaiki kualitas suatu sinyal audio dan menambah variasi suara dalam suatu aplikasi tertentu.

Pada awalnya efek audio ini dibuat dengan berbagai rangkaian analog. Biasanya, satu buah sistem rangkaian analog hanya bisa menghasilkan satu jenis efek tertentu saja. Jika ingin dibuat gabungan beberapa jenis efek, maka kompleksitas rangkaian kemungkinan akan bertambah. Sebaliknya, dengan adanya prosesor DSP pada efek audio dijital, maka pembuatan suatu aplikasi efek audio akan menjadi lebih praktis, yaitu cukup dengan menuliskan program yang diinginkan pada prosesor [2]. Selain itu pemodelan efek yang berbeda-beda tersebut dapat dilakukan dengan bantuan SIMULINK® [3].

Prosesor DSP yang ada saat ini bermacam-macam jenisnya, tergantung pada aplikasinya. Aplikasi audio yang akan dibuat disini diterapkan pada prosesor DSP C6713 yang diproduksi oleh *Texas Instruments*. Prosesor ini sudah tergabung ke dalam suatu sistem yang dipasang pada sebuah *board* yang dinamakan DSP *Starter Kit* TMS320C6713. DSK sebenarnya adalah suatu *platform* yang dikhususkan untuk tujuan pembelajaran. Walupun begitu, DSK ini sudah dapat melakukan berbagai aplikasi yang bersifat *real time*.

Aplikasi pada bidang audio adalah sesuatu yang menarik. Walaupun penerapannya banyak di dunia hiburan, tetapi di balik itu semua terdapat ilmu sains yang sebenarnya tidak sederhana. Hal yang menarik lainnya adalah penerapan efek

audio pada DSK TMS320C6713 dengan bantuan pemodelan SIMULINK® bukan sekedar menghasilkan simulasi, tetapi sudah merupakan aplikasi nyata yang bisa digunakan dalam kehidupan sehari-hari. Pada skripsi ini akan dirancang suatu aplikasi efek audio dengan bantuan SIMULINK® yang dapat bekerja secara *real time* untuk diimplementasikan pada DSK TMS320C6713. Hasil dari perancangan akan dianalisis untuk mengetahui performa dari tiap jenis efek.

1.2 PERUMUSAN MASALAH

Masalah yang ada adalah pemodelan bentuk efek audio dalam SIMULINK® yang terdapat dalam MATLAB. Lalu dari model tersebut, akan diaplikasikan pada DSP *Starter Kit* TMS320C6713 sehingga model yang sudah dibuat dapat berjalan dengan *real-time*.

1.3 TUJUAN PENELITIAN

Tujuan penelitian yang dilakukan adalah untuk merancang model efek audio dijital pada prosesor DSP *Starter Kit* TMS 320C6713 dengan bantuan SIMULINK®.

1.4 BATASAN MASALAH

Masalah yang dibahas dalam skripsi ini adalah:

1. Pemodelan efek audio dengan menggunakan SIMULINK® pada MATLAB
2. Hasil model akan dijalankan secara *real time* pada DSK TMS320C6713
3. Tiga jenis efek yang akan dievaluasi yaitu: *Reverberation, Echo,* dan *Chorus*
4. Suara *tone* 1 KHz dan sinyal acak akan menjadi masukan dari model yang akan dievaluasi

1.5 METODOLOGI PENELITIAN

Metodologi penelitian yang dipakai adalah dengan pemodelan simulasi dengan menggunakan SIMULINK® yang diaplikasikan dengan DSP *Starter Kit* lalu data hasil simulasi akan dibandingkan dengan hasil simulasi dan dianalisis.

1.6 SISTEMATIKA PENULISAN

Skripsi ini dibagi menjadi 5 bab, yaitu :

A. BAB 1 PENDAHULUAN

Menjelaskan mengenai latar belakang, perumusan masalah, tujuan, batasan masalah, metodologi penelitian serta sistematika penulisan skripsi.

B. BAB 2 TINJAUAN PUSTAKA

Memaparkan berbagai hal yang berhubungan dengan pengolahan sinyal dijital, SIMULINK®, DSK TMS320C6713 dan *digital audio effect*.

C. BAB 3 RANCANG BANGUN *DIGITAL AUDIO EFFECT* DENGAN SIMULINK®

Pada bab ini dijelaskan bagaimana merancang efek audio dengan menggunakan SIMULINK® serta DSK TMSC6713.

D. BAB 4 HASIL UJI COBA DAN ANALISIS HASIL SIMULASI

Pada bab ini dijelaskan bagaimana hasil rancangan yang telah diaplikasikan dengan DSP S*tarter Kit* beserta analisisnya.

E. BAB 5 KESIMPULAN

Berisi penutup yang berupa kesimpulan dari hasil dan analisis pada bab sebelumnya.

BAB II

HUBUNGAN ANTARA ALAT BANTU SIMULASI

2.1 PEMROSESAN SINYAL DIJITAL (PSD)

Pemrosesan sinyal dijital (PSD) adalah teknologi yang saat ini sudah mendasari hampir seluruh bidang dalam kehidupan manusia. PSD dibedakan dengan bidang sains yang lain karena keunikan data yang diolahnya, yaitu sinyal. Sinyal yang diolah pada umumnya berasal dari alam dalam bentuk analog [4]. Lalu sinyal ini diubah dalam bentuk dijital melalui proses *sampling*, kuantisasi, dan *coding*. Semua proses ini dilakukan oleh alat yang bernama ADC (*Analog to Digital Converter*). Setelah sinyal menjadi bentuk dijital, barulah diproses secara dijital oleh prosesor DSP. Hasil keluaran dari proses ini selanjutnya diubah menjadi analog kembali oleh DAC (*Digital to Analog Converter*). Hal ini penting dilakukan karena pada umumnya sinyal yang bisa kita lihat atau dengar adalah sinyal analog.

Pada penerapannya, pemrosesan sinyal banyak melibatkan bidang-bidang ilmu yang lain seperti teori telekomunikasi, analisis numerik, probabilitas/statistika, elektronik dijital, elektronika analog, dan sebagainya. Salah satu aplikasi yaitu pemrosesan audio seperti pada musik, *speech recognition*, *speech synthetis*, dan sebagainya.

Namun, dari sekian banyak aplikasi yang ada, pada prinsipnya pemrosesan sinyal dijital hanya memakai kurang lebih 5 operasi dasar, yaitu korelasi, konvolusi, transformasi, *Discrete Fourier Transform* (DFT) dan *Fast Fourier Transform* (FFT), dan *Filtering*. Dalam skripsi ini hanya akan dibahas mengenai DFT dan FFT serta *Filtering*.

2.1.1 DFT dan FFT

DFT termasuk salah satu jenis transformasi. Dengan menerapkan formula DFT, suatu sinyal dalam domain waktu dapat diubah ke dalam domain frekuensi. Pada domain frekuensi, sinyal dipresesntasikan dalam frekuensi (sumbu x) dan magnitude (sumbu y). Jadi dapat dilihat dengan jelas frekuensi kerja dan *power* suatu sinyal ataupun sistem. Formula DFT dapat dilihat di bawah ini [5].

$$X(k) = F_D[x(nT)] = \sum_{n=0}^{N-1} x(nT)e^{-jk\Omega nT} \quad (2.1)$$

FFT sebenarnya memiliki fungsi yang sama dengan DFT, hanya FFT menggunakan algoritma perhitungan yang lebih efisien dibandingkan dengan DFT. Hal ini diperlukan dalam aplikasi untuk meningkatkan efisiensi kerja suatu prosesor DSP.

2.1.2 Filter

Filter adalah sistem yang secara selektif dapat mengubah bentuk sinyal, menghilangkan sinyal tertentu, maupun berbagai karakteristik sinyal lainnya. Penggunaan filter ini banyak dan luas sekali. Sebagian besar aplikasi pemrosesan sinyal menggunakan filter. Pada PSD, filter yang didesain adalah filter dijital. Pada filter ini, penentu karakteristik dari filter adalah koefisien-koefisien filter. Secara umum, filter dijital dapat dibagi menjadi 2 jenis, yaitu :

a. *Finite Impulse Response* (FIR)

   FIR dipakai ketika jumlah koefisien tidak terlalu besar dan tidak diinginkan distorsi fasa. Formula FIR dapat dilihat di bawah ini [5].

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (2.5)$$

b. *Infinite Impulse Response* (IIR)

   IIR dipakai jika ingin didapatkan frekuensi *cutoff* yang tajam dan *throughput* yang tinggi. Formula IIR dapat dilihat di bawah ini [5].

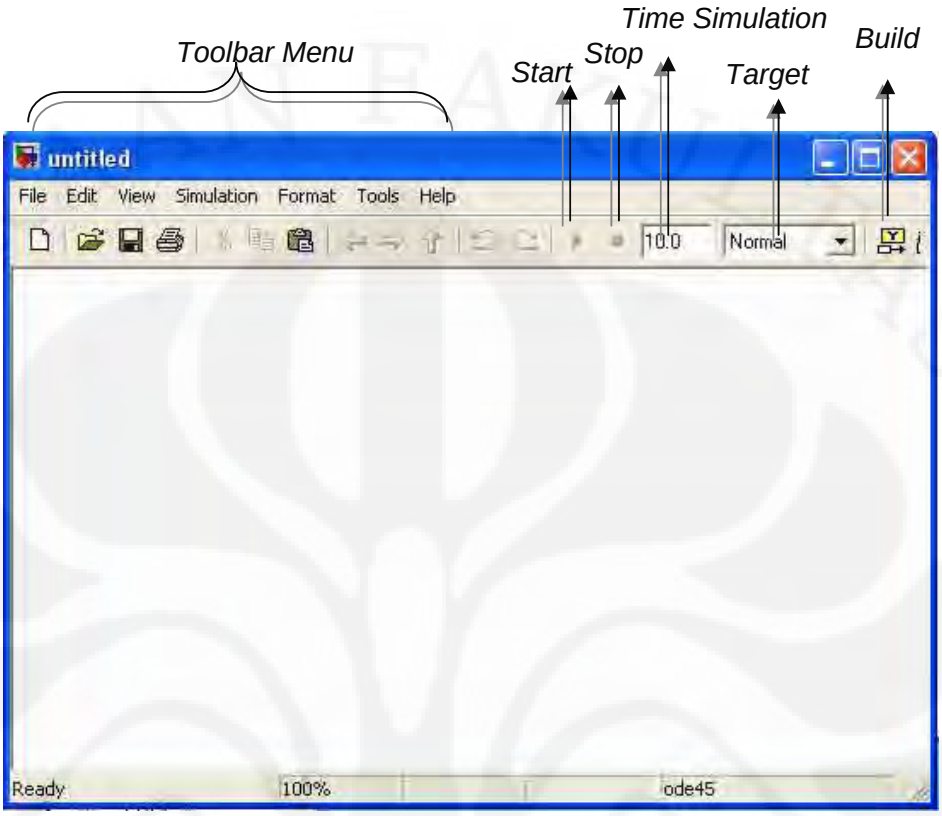
$$y(n) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (2.6)$$

Operasi dasar yang digunakan dalam pemrosesan sinyal hanya berupa perkalian dan penjumlahan sederhana saja. Namun kedua operasi yang dilakukan ini sangat banyak jumlahnya, sehingga untuk menerapkannya dalam aplikasi

diperlukan suatu prosesor yang sangat cepat dalam melakukan perhitungan matematis. Untuk itulah didesain suatu mikroprosesor yang bekerja khusus untuk memproses sinyal dijital yang disebut *Digital Signal Processor* (DSP).

2.2 SIMULINK®

SIMULINK® yang dikembangkan oleh *The MathWorks*, adalah sub-program dari MATLAB© yang digunakan untuk simulasi, pemodelan atau analisis sistem dinamis dalam multi domain tanpa harus menggunakan bahasa pemrograman tingkat tinggi seperti bahasa pemrograman C, meskipun demikian, SIMULINK® tetap dapat *disinkronisasikan* dengan bahasa pemrograman C. Bahkan setelah penambahan beberapa program yang bersifat *Real Time Workshop® (contoh: Code Composer Studio),* SIMULINK® mampu menghasilkan bahasa pemrograman C dari model yang telah dibuat. SIMULINK® menyediakan lingkungan bekerja dimana pengguna dapat memodelkan sistem secara fisik dan pengontrolannya sebagai blok diagram [6]. Simulink dapat dijalankan dari menu utama, yaitu dengan mengetik SIMULINK® atau dengan menekan *icon* yang menandakan SIMULINK®, setelah itu, *user* akan masuk ke menu utamanya.



Gambar 2.1 Menu SIMULINK®

Dalam menu utama ini diperlihatkan *library* SIMULINK®yang tersedia. Setiap model dikategorikan dalam berbagai kelas (*blockset)*, yang akan memudahkan pencarian. Tiap versi MATLAB, mempunyai *library* yang berbeda, dalam skripsi ini, MATLAB yang digunakan adalah versi R2007a. Tiap blok yang tersedia, dapat diketahui karakteristiknya dengan memasuki menu *help*. Jika kita ingin membuat suatu permodelan, maka kita harus membuat terlebih dahulu suatu *Workspace* yang baru (dapat diakses melalui *toolbar* yang tersedia). Lalu untuk menggunakan blok yang terdapat pada *library* kita cukup menggeser (*drag)* setiap blok yang dikehendaki menuju *workspace.*

Setelah kita selesai membuat suatu model, kita dapat menganalisisnya dengan mensimulasikan model tersebut, dengan cara memlih *icon –run-* pada *toolbar* yang tersedia. Pesan kesalahan akan muncul jika model tidak bisa disimulasikan. Saat mensimulasikan model, kita dapat memilih target simulasi, apakah hanya akan dijalankan pada komputer saja, atau dapat juga disimulasikan ke sebuah alat simulasi eksternal (contoh*:DSK board).*

2.2.1 *Targetting* DSK TMS320C6713 [7]

Pada MATLAB disediakan suatu alat bantu yaitu Target Support Package™ dimana memungkinkan pengguna untuk menggunakan *Real-Time®Workshop software* untuk menghasilkan program dengan *platform* bahasa C dimana merupakan hasil dari implementasi dari model yang telah dibuat di SIMULINK®. Dari hasil yang berupa bahasa C, pengguna dapat meng-*compile*, melakukan koneksi, dan menjalankan dengan *hardware* C6713 DSP *Starter Kit* (DSK).

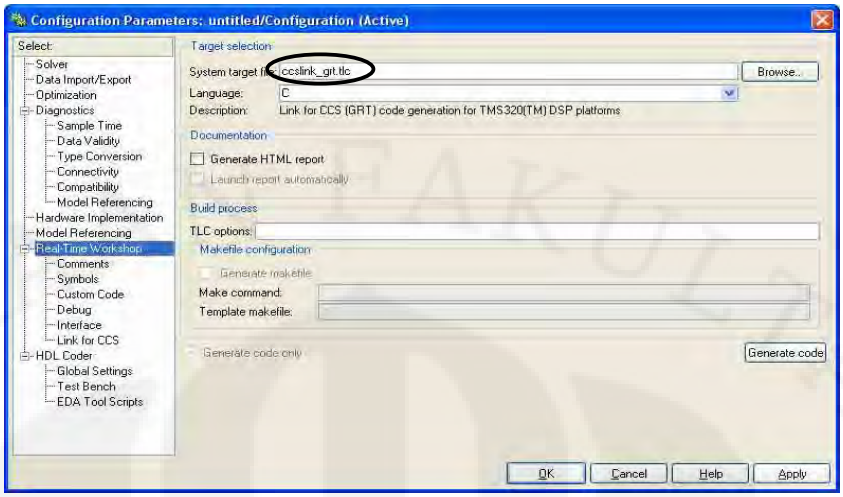
Dalam Matlab, kita dapat mengetikkan `c6713dsklib`. Hal ini akan membuka perpustakaan diagram blok yang dipakai dalam DSK C6713. Tabel 2.1 menunjukkan diagram blok dan penjelasannya yang dipakai dalam DSK C6713.

Tabel 2.1 Blok C6713 pada *Library* SIMULINK®

Nama Blok	Deskripsi Blok
C6713 DSK ADC	Mengkonfigurasi konverter analog ke dijital
C6713 DSK DAC	Mengkonfigurasi konverter dari dijital ke analog
C6713 DSK LED	Mengontrol status LED di C6713 DSK
C6713 DSK Reset	Me-*reset* prosesor di C6713 DSK

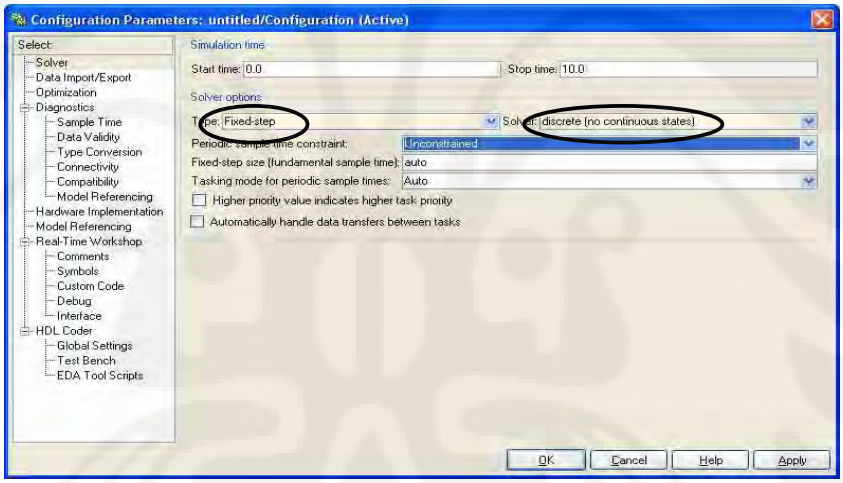
Setelah model yang kita buat sudah selesai, ada beberapa parameter konfigurasi yang harus diubah. Hal ini dapat kita lakukan dengan memilih ***Simulation → Configuration Parameters*.** Ada beberapa yang diubah yaitu:

1. Pada *Real-Time Workshop*, ubah *target selection* menjadi ccslink_grt.tlc seperti pada Gambar 2.2



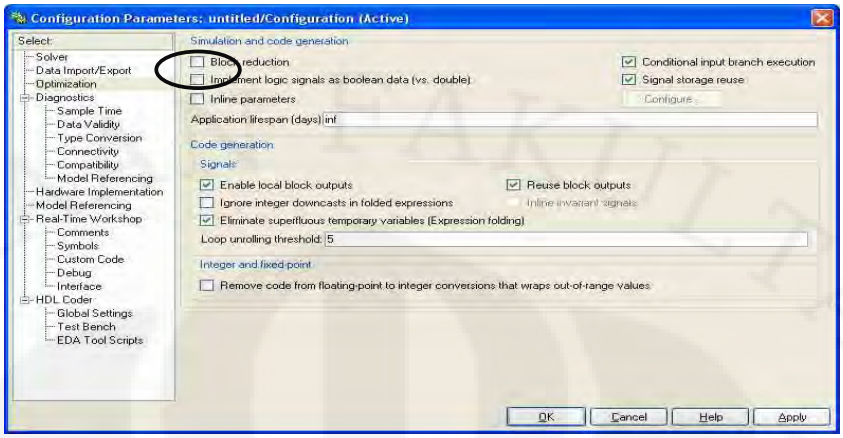
Gambar 2.2 Tab Menu *Real Time Workshop*

2. Pada *Solver*, ubah *solver options* dengan tipe *fixed-step* dan *solver* adalah *discrete* seperti pada Gambar 2.3



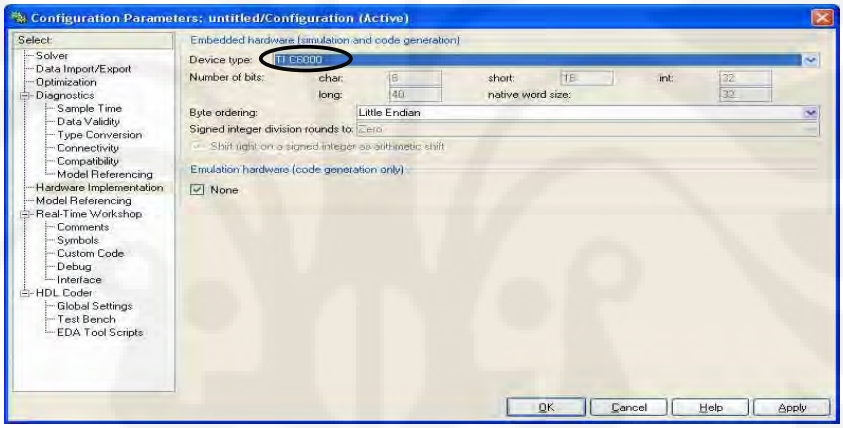
Gambar 2.3 Tab Menu *Solver*

3. Pada *Optimization*, *block reduction* dan *implement logic signal as boolean data* di-*uncheck* seperti pada Gambar 2.4



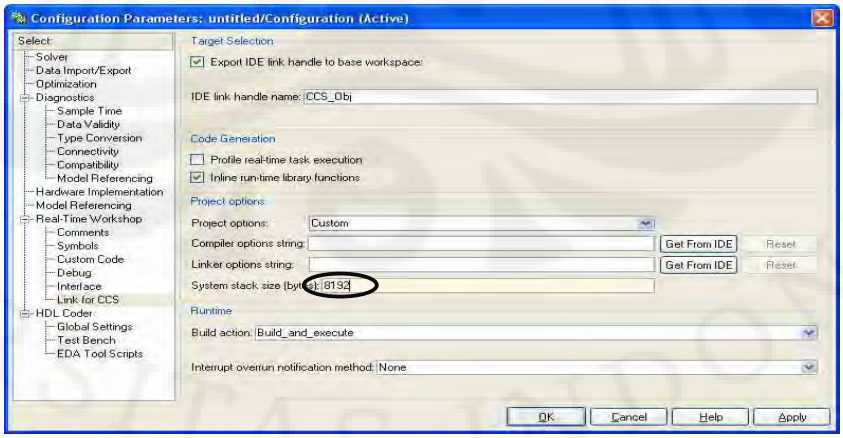
Gambar 2.4 Tab Menu *Optimization*

4. Pada *Hardware Implementation*, tipe alat yang digunakan adalah TI C6000 seperti pada Gambar 2.5



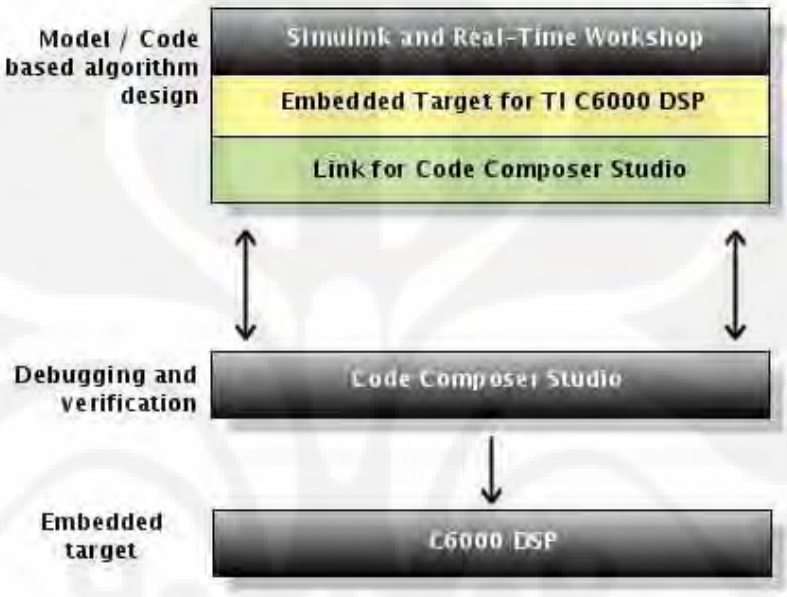
Gambar 2.5 Tab Menu *Hardware Implementation*

5. Pada *link for CCS*, *stack* yang disediakan sampai 8192 seperti pada Gambar 2.6.



Gambar 2.6 Tab Menu *Link for* CCS

Secara sederhana proses dalam *targetting* ini menggunakan SIMULINK® dan CCS. Untuk menghubungkan SIMULINK® dengan DSK dibutuhkan *Real Time Workshop, Embedded Target for TI C6000 DSP,* dan *Link for CCS*. Ketiga hal tersebut dapat ditemukan di SIMULINK® dan harus dilakukan pengaturan konfigurasi. Hubungan ketiga hal tersebut dapat dilihat pada Gambar 2.7.

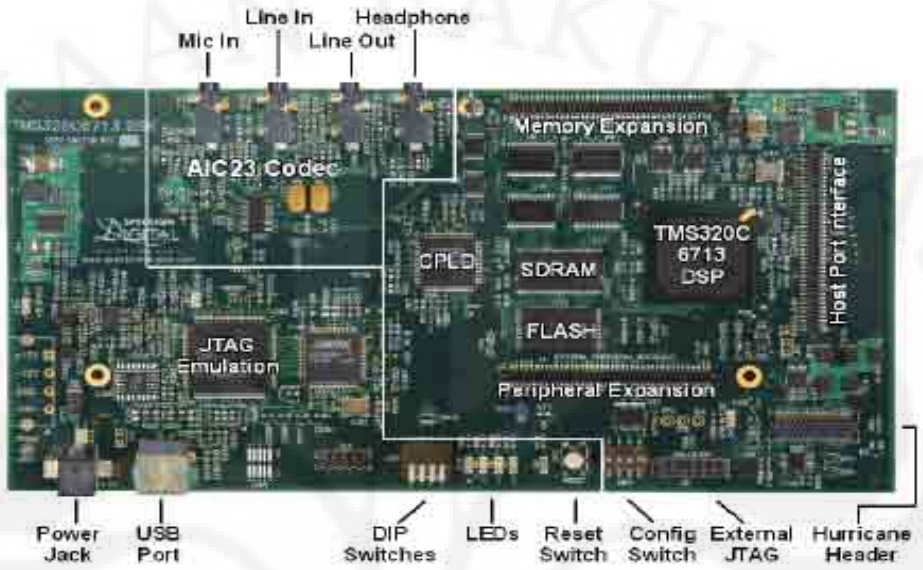


Gambar 2.7 Diagram Alir *Targetting to* C6000 DSP

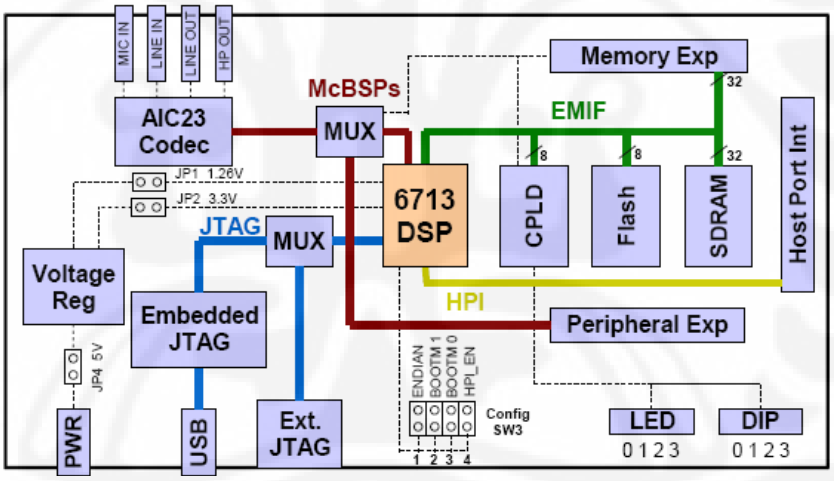
Dari Gambar 2.7 menunjukkan proses *debugging* dan *verification* dilakukan oleh *software* CCS. Penggunaan CCS memungkinkan untuk menghasilkan *code-code* yang akan digunakan dalam C6000 DSP sehingga tidak diperlukan lagi pembuatan program dengan manual karena sudah dilakukan oleh CCS.

2.3 DSK TMS320C6713

DSK TMS320C6713 adalah salah satu DSP tipe C6000 yang dapat bekerja pada *fixed-point* maupun *floating-point*. Tetapi, DSP ini masih berupa *starter kit*, yaitu suatu *platform* yang dapat mensimulasikan DSP C6713 yang sebenarnya. DSK ini lebih ditujukan untuk keperluan edukasi, penelitian, serta evaluasi. Namun, hasil dari aplikasi yang kita buat di DSK ini sangat mungkin untuk diterapkan pada DSP C6713 yang sebenarnya. Gambar 2.8 dan 2.9 memberikan gambaran fisik dan blok diagram dari DSK C6713.



Gambar 2.8 Bentuk fisik DSK TMS320C6713 [11]



Gambar 2.9 Blok diagram DSK TMS320C6713 [11]

Komponen-komponen utama dan pendukung dari DSK C6713 yaitu:

1. Prosesor TMS320C6713

   Merupakan prosesor dengan kecepatan *clock* 225 MHz yang mendukung operasi *fixed-point* dan *floating-point*. Kecepatan operasinya dapat mencapai 1350 juta operasi *floating-point* per detik (MFLOPS) dan 1800

juta instruksi per detik (MIPS). Selain itu, prosesor ini dapat melakukan 450 juta operasi *multiply-accumulate* per detik.

2. CPLD (*Complex Programmable Logic Device*)

   CPLD berisi register-register yang berfungsi untuk mengatur fitur-fitur yang ada pada board. Pada DSK C6713, terdapat 4 jenis register CPLD, yaitu:

   a. USER_REG Register

      Mengatur *switch* dan LED sesuai yang diinginkan *user*.

   b. DC_REG Register

      Memonitor dan mengontrol *daughter card*.

   c. VERSION Register

      Indikasi yang berhubungan dengan versi *board* dan CPLD.

   d. MISC Register

      Untuk mengatur fungsi lainnya pada *board*.

3. *Flash memory*

   DSK menggunakan memori *flash* yang berfungsi untuk *booting*. Dalam *flash* ini berisi sebuah program kecil yang disebut POST (*PowerOn Self Test*). Program ini berjalan saat DSK pertama kali dinyalakan. Program POST akan memeriksa fungsi-fungsi dasar *board* seperti koneksi USB, *audio codec*, LED, *switches*, dan sebagainya.

4. SDRAM

   Memori utama yang berfungsi sebagai tempat penyimpanan instruksi maupun data.

5. AIC23 *Codec*

   Berfungsi sebagai ADC maupun DAC bagi sinyal yang masuk ke *board*.

6. *Daughter card interface*

   Konektor-konektor tambahan yang berguna untuk mengembangkan aplikasi-aplikasi pada *board*. Terdapat 3 konektor, yaitu *memory expansion, peripheral expansion,* dan *Host Port Interface*.

7. LED dan *Switches*

   LED dan *switches* ini merupakan fitur yang dapat membantu dalam membangun aplikasi karena dapat deprogram sesuai keinginan *user*.

8. JTAG (Joint Test Action Group)

   Merupakan konektor yang dapat melakukan transfer data dengan kecepatan yang sangat tinggi. Hal ini akan berguna dalam aplikasi *real-time*.

2.4 EFEK AUDIO DIJITAL

Saat sekarang ini, jika berbicara tentang audio tentunya tidak dapat terlepas dari berbagai peralatannya yang berbasis dijital. Musik yang diproduksi sekarang pun sebagian sudah melalui proses pengolahan sinyal secara dijital. Dengan demikian, perkembangan prosesor DSP tentunya sangat mempengaruhi kemajuan di bidang audio.

PSD diterapkan dalam bidang audio dengan alasan-alasan tertentu. Misalnya saja untuk memanipulasi sinyal seperti menghilangkan *echo* atau *noise* tentunya lebih mudah dilakukan dalam bentuk dijital. Selain itu sinyal dijital lebih tahan terhadap *noise* bila dibandingkan dengan sinyal analog.

Pada dasarnya, prinsip kerja dari efek audio adalah memanipulasi sinyal *input* agar menjadi suatu sinyal *output* seperti yang kita inginkan. Dalam kehidupan sehari-hari, efek audio lebih banyak dipakai di bidang hiburan. Sebenarnya, penerapan efek audio sudah dapat dilakukan dengan rangkaian-rangkaian analog. Namun, dengan rangkaian analog ini untuk mendapatkan berbagai macam efek audio dalam satu perangkat, diperlukan rangkaian yang kompleks. Hal ini yang membedakannya dengan efek audio dijital. Dalam satu prosesor DSP, bisa diterapkan bermacam-macam efek audio yang diinginkan. Walaupun banyak efek audio yang ingin diterapkan, kompleksitas dari rangkaian DSP tidak akan berubah secara signifikan, hanya pemrogramannya yang akan bertambah banyak. Hal ini menjadi salah satu keunggulan DSP. Selain itu, keunggulan lainnya adalah kemudahan dan kecepatan dalam membangun aplikasi. Untuk membuat efek audio dalam bentuk analog, langkah yang harus dilakukan adalah melakukan perhitungan parameter-parameter yang diperlukan, lalu merangkainya dengan proses-proses tertentu, dan terakhir dilakukan *testing* pada alat untuk mengetahui apakah alat sudah sesuai dengan yang diinginkan. Namun, jika gagal (tidak sesuai dengan yang diinginkan) harus dilakukan perhitungan dan

pembuatan ulang. Dengan demikian untuk membuat suatu aplikasi, waktu yang dibutuhkan cukup lama.

Secara garis besar, efek audio diklasifikasikan menjadi 3 macam [13], yaitu:

a. *Drive*/Distorsi

Misalnya *overdrive, fuzz,* dan lain-lain

b. Modulation

Misalnya *chorus, flanger, phaser,* dan lain-lain

c. Ambience

Misalnya *delay, reverb, echo,* dan lain-lain

Setiap efek memiliki karakteristik yang berbeda-beda, sehingga konsep dasar efek sangat diperlukan.

2.4.1 *Drive*/Distorsi [12]

Distorsi pada prinsipnya adalah pengubahan atau rusaknya bentuk asli dari sebuah sinyal informasi baik secara sengaja ataupun tidak. Biasanya efek ini sering digunakan dalam musik. Pada dunia audio, efek seperti distorsi ini diinginkan karena akan menambah variasi bunyi yang akan membuat musik menjadi lebih menarik.

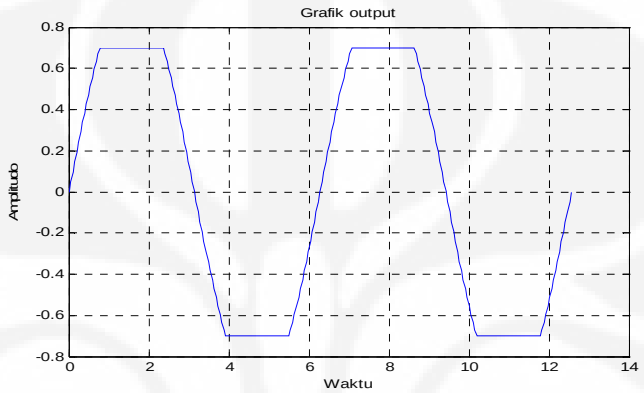
Seperti yang telah disebutkan sebelumnya, terdapat beberapa jenis efek distorsi, seperti *overdrive, fuzz,* dan lain-lain. Prinsip dari semua efek distorsi sebenarnya sama, yaitu menimbulkan distorsi harmonik dari sebuah sinyal. Distorsi harmonik itu sendiri berarti munculnya komponen-komponen frekuensi lain selain frekuensi fundamental. Perbedaan antara jenis-jenis efek tersebut terletak pada metodenya dalam membuat distorsi harmonik.

2.4.1.1 *Overdrive*

Efek ini dalam aplikasinya menggunakan *vacuum tube,* atau dengan menggunakan simulated tube modeling techniques. Bagian teratas dari gelombang bunyi yang terjadi dikompres, hal ini akan memberikan sinyal yang terdistorsi lebih halus daripada efek distorsi yang biasa.

2.4.1.2 *Fuzz*

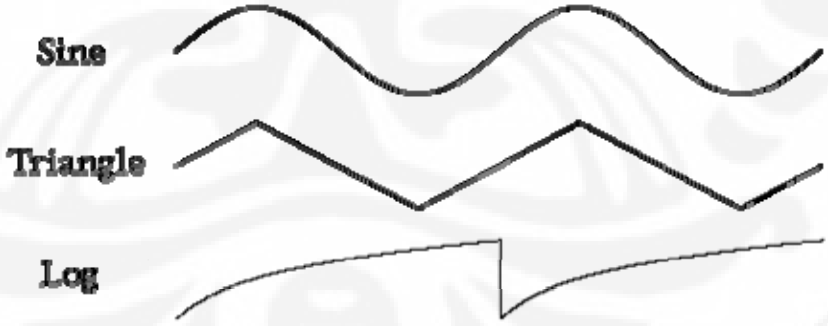
Efek *fuzz* diciptakan pada tahun 1960 dimana bunyi dari efek bunyi yang mengalami *overdrive* dikombinasi oleh *torn speaker cones*. Metode yang digunakan dalam efek ini adalah metode *clipping*. Prinsip kerjanya cukup sederhana, yaitu dengan menentukan suatu nilai batas pada *output*. Jika *output* melebihi nilai batas ini, maka secara otomatis sinyal *output* akan terpotong.



Gambar 2.10 *Output* yang mengalami *clipping*[12]

2.4.2 *Modulation*

Modulasi yang diterapkan pada musik umumnya menggunakkan *Low Frequency Oscillator* (LFO). LFO selanjutnya akan digunakan untuk mengontrol nilai delay yang berubah-ubah menurut waktu[13].



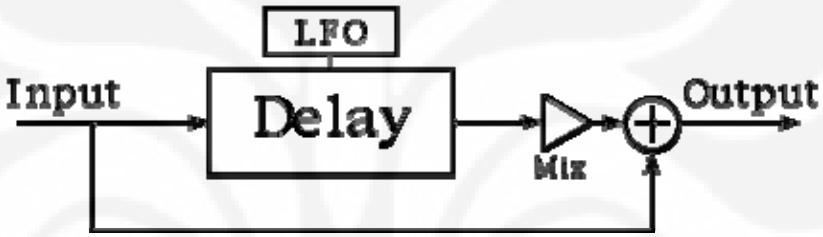
Gambar 2.11 Jenis LFO yang biasa dipakai

2.4.2.1 *Chorus*

Efek *chorus* dapat divisualisasikan seperti beberapa orang memainkan banyak instrumen musik dalam bentuk unisono. Pada *chorus*, nada yang

dihasilkan dapat berdeviasi bergantung pada penundaan yang diberikan. Penundaan terjadi secara perlahan dan pembentukan tunda ini dapat diwakilkan dengan variabel baris tunda yang berulang. Penundaan ini mengakibatkan perubahan yang terjadi berulang-ulang seiring dengan berjalannya waktu.

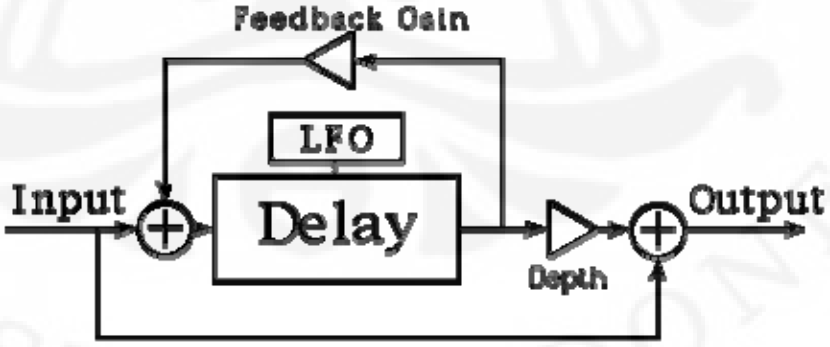
Efek *chorus* berbeda dengan efek *flang* dari segi waktu tunda yang terjadi. Waktu tunda yang terjadi pada efek *chorus* lebih panjang daripada efek *flang*. Biasanya waktu tunda untuk efek *chorus* diantara 20-30 ms sedangkan efek *flang* adalah 1-10 ms [14]. Perbedaan selanjutnya adalah pada diagram blok efek *chorus* tidak terdapat komponen umpan balik / *feedback*. Blok diagram untuk efek *chorus* diwakilkan oleh Gambar 2.12.



Gambar 2.12 *Chorus* dengan komponen LFO[14]

2.4.2.2 *Flanger*

Efek *flang* menghasilkan suara dengan mencampurkan suara asli dengan suara yang sudah mengalami penundaan dimana panjang penundaan yang terjadi selalu berubah. Hal ini sama dengan efek *chorus* namun pada efek *flang* diberi komponen tunda. Gambar 2.13 menunjukkan blok diagram untuk *flang*.

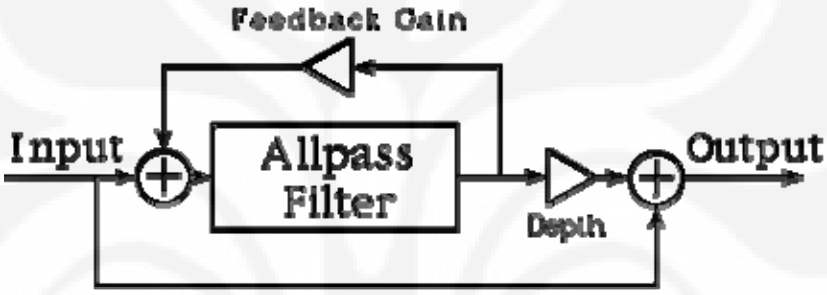


Gambar 2.13 Blok diagram efek *flang*[15]

2.4.2.3 *Phaser*

Efek *phaser* atau lebih dikenal dengan *phase shifter* bekerja dengan membuat satu atau lebih *notch* dalam domain frekuensi. *Notch* ini dapat dibuat dengan melakukan filterisasi sinyal dan kemudian mencampurnya dengan sinyal asli. Filter ini dapat disesuaikan dengan jumlah notch yang ingin dibuat bahkan letak *notch* itu berada. Biasanya filter yang digunakan adalah *allpass filter*.

Dengan menggunakan *allpass filter*, respon dari frekuensi yang terjadi dapat didistorsi karena *allpass filter* tidak memiliki karakteristik fase yang linear. Gambar 2.14 menunjukkan diagram blok dari efek *phaser*.



Gambar 2.14 Diagram blok efek *phaser* [16]

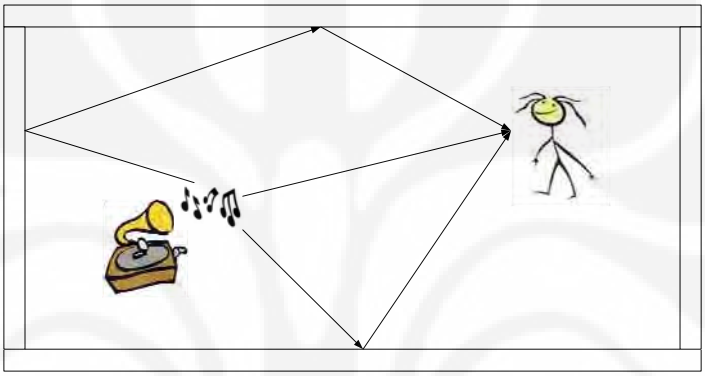
2.4.3 *Ambience*

Efek yang termasuk dalam golongan *ambience* biasa disebut juga *delay based audio effect. Delay based audio effect* adalah jenis efek audio yang prinsip kerjanya berdasarkan keterlambatan waktu. Secara garis besar, pada efek ini terdapat 2 komponen, yaitu bunyi asli dan bunyi tiruan yang mengikuti bunyi asli dalam selang waktu tertentu. Sama seperti pada efek lainnya, efek *delay* dibedakan menjadi beberapa jenis, yaitu *echo*, *reverb*, dan sebagainya. Prinsip kerjanya sama, perbedaannya hanya terletak pada besarnya waktu keterlambatan antara bunyi asli dengan bunyi tiruan.

2.4.3.1 *Reverberation*

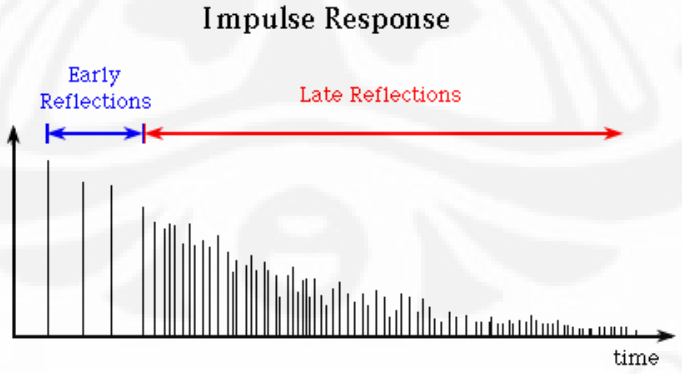
Reverberation atau r*everb* merupakan hasil dari banyak pantulan suara yang terjadi di dalam suatu ruangan yang berasal dari beberapa sumber suara misalnya *speaker* dari alat musik. Memang terdapat jalur yang langsung ke telinga kita, namun hal itu tidak berarti hanya satu-satunya jalur. Gelombang suara bisa

saja memakai jalur yang panjang dengan memantulkan gelombang suaranya terlebih dahulu ke tembok atau langit-langit sebelum sampai ke telinga kita. Hasil pantulan ini akan menyebabkan adanya suara yang sampai ke telinga kita namun terdengar lebih terlambat dan lebih lemah yang disebabkan penyerapan oleh tembok atau langit-langit. Mekanisme tersebut dapat dilihat pada Gambar 2.15 di bawah ini



Gambar 2.15 Mekanisme *Reverberation*

Gambar 2.15 menunjukkan mekanisme dan pengaruh efek *reverb*. Setiap pantulan dari sumber bunyi akan menjadi suatu sumber bunyi baru. Dari sini, tampak bahwa efek *reverb* terdiri dari kumpulan *echo* dengan keterlambatan waktu yang sangat kecil. Namun, pada kenyataan tidak demikian. Hal ini dapat dilihat dari *impulse response*-nya pada Gambar 2.16 berikut ini.



Gambar 2.16 *Impulse Response* pada *Reverberation*

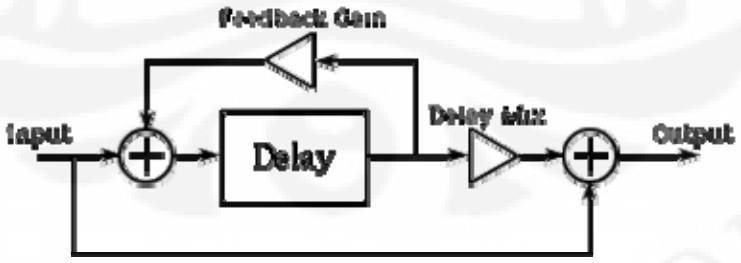
Pada efek *reverb*, keterlambatan waktu antar tiap bunyi pantulan bernilai tetap hanya pada selang waktu tertentu saja, selanjutnya besarnya keterlambatan akan berubah terhadap waktu. Pada Gambar 2.16 dapat dilihat bahwa, terdapat 2

komponen utama pada efek ini, yaitu *early reflections* dan *late reflections*. *Early reflections* adalah pantulan-pantulan yang jaraknya masih teratur. Sedangkan *late reflections* adalah bunyi pantulan sesudah *early reflections* yang bersifat acak dan lebih sulit untuk diukur. Kedua komponen ini akan sangat bergantung pada bentuk dan ukuran dari ruangan. Semakin besar bentuk ruangan, maka jarak tiap bunyi pantulan akan semakin jauh sehingga keterlambatan waktunya akan lebih besar. Pada *reverb* terdapat 2 parameter yang penting [17], yaitu :

- Predelay, merupakan selang waktu sebelum munculnya bunyi pantulan yang pertama kali.
- Reverb decay, adalah waktu terjadinya *reverb* sejak *input* berhenti.

Efek yang berdasarkan keterlambatan waktu ini biasanya muncul secara alami dalam kehidupan manusia sehari-hari dan sulit untuk dihindari. Pada bidang telekomunikasi, adanya efek keterlambatan ini biasanya mempunyai dampak negatif. Tetapi lain halnya dengan bidang audio. Misalnya pada konser-konser musik yang diadakan di dalam ruangan, adanya efek ini tidak mengganggu, melainkan akan menambah kualitas musik yang didengar. Permasalahan yang terjadi adalah efek ini tidak bisa selalu didapatkan dalam porsi yang sama disebabkan oleh bentuk ruangan yang berbeda-beda. Efek *reverb* yang ada ketika musik diputar dalam sebuah mobil tentu akan berbeda dengan efek *reverb* yang muncul ketika musik di putar di gedung yang besar. Untuk itu, agar efek ini bisa dikendalikan diperlukan suatu alat yang bisa memunculkan efek yang sama seperti efek *reverb* yang alami.

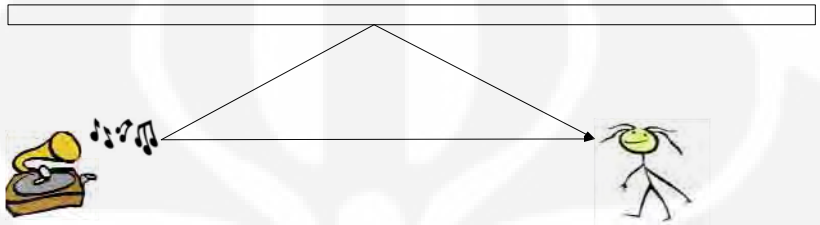
Blok diagram efek *reverb* digambarkan pada Gambar 2.17



Gambar 2.17 Blok diagram efek *reverb*

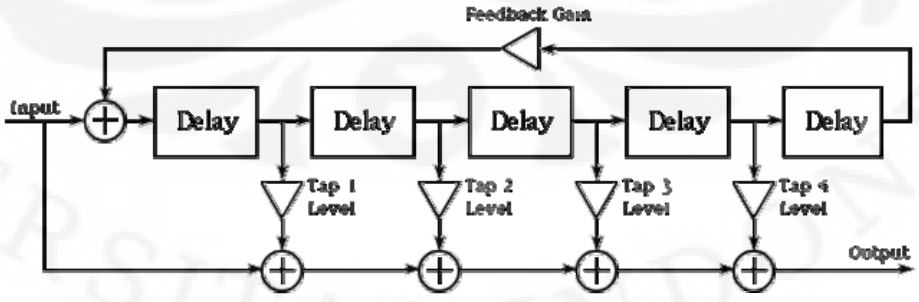
2.4.3.2 *Echo*

Echo adalah pantulan bunyi yang datang ke pendengar pada selang waktu tertentu setelah bunyi asli. Pada umumnya, *echo* memiliki keterlambatan waktu yang relatif besar, yaitu lebih dari satu detik. Sehingga, pada efek *echo* bunyi asli dan bunyi tiruan terpisah dengan jelas dan masih dapat dibedakan oleh kuping manusia. Mekanisme dari *echo* dapat dilihat pada Gambar 2.18 berikut.



Gambar 2.18.Mekanisme terjadinya *echo*

Seperti terlihat pada Gambar 2.18, suara berjalan dari sumber menuju pendengar dengan 2 cara. Pertama, suara berjalan melalui jalur langsung dari sumber menuju pendengar. Kedua, suara menuju ke dinding lalu dipantulkan ke pendengar [11]. Jika dilihat dari segi waktu, jalur kedua ini tentu akan memakan waktu lebih lama daripada jalur yang pertama, sehingga pendengar akan mendengar 2 bunyi pada waktu yang berbeda. Sedangkan dari segi kekuatan, sinyal yang menempuh jalur kedua akan mengalami atenuasi yang disebabkan oleh pemantulan. Gambar 2.19 menunjukkan blok diagram efek *echo*.



Gambar 2.19 Blok diagram efek *echo* dengan 4-*tap*

BAB III

RANCANG BANGUN *DIGITAL AUDIO EFFECT* DENGAN SIMULINK

Pada DSK TMS320C6713, aplikasi dapat dibuat dalam bentuk algoritma bahasa C ataupun assembler. Jadi proses yang harus dilakukan pertama kali adalah *coding*. Setelah selesai, algoritma ini dapat langsung diterapkan ke DSK dan dilihat bagaimana performanya.

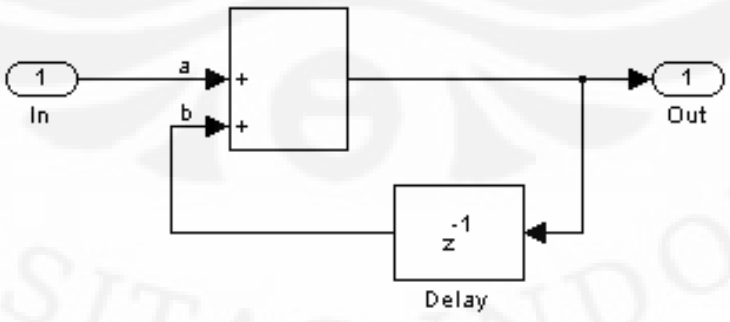
Selain dengan algoritma bahasa C dimana menggunakan *software* CCS, perancangan efek audio dijital dapat dilakukan dengan pemodelan menggunakan SIMULINK®. Penggunaan SIMULINK® memudahkan dalam membentuk gambaran dari algoritma masing-masing efek. Masing-masing efek memiliki blok diagram yang berbeda-beda.

3.1 *DELAY BASED EFFECT*

Dalam perancangan ini, efek yang berdasarkan *delay* dibagi dua yaitu *reverberation* dan *echo*.

3.1.1 *Reverberation*

Konsep dalam menghasilkan efek *reverb* adalah dengan mencampurkan suara asli dengan suara yang sudah mengalami penundaan/*delay*. Hal tersebut dapat digambarkan dalam diagram blok pada Gambar 3.1.



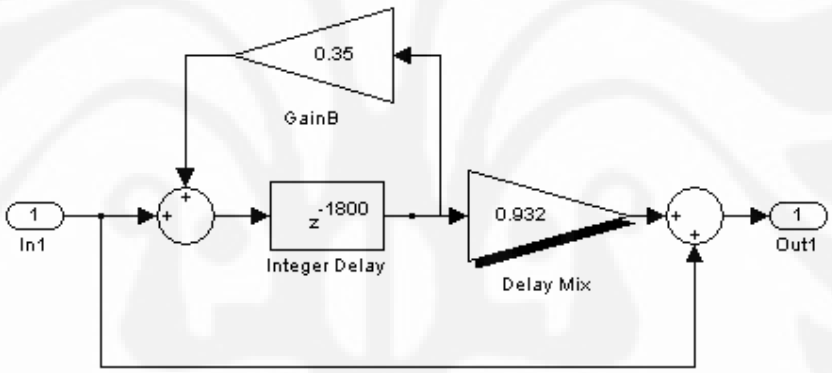
Gambar 3.1 Blok Dasar *Reverberation*

Diagram pada Gambar 3.1 menunjukkan diagram efek *reverb* yang masih sederhana dalam arti belum diberikan *gain*/penguatan. Diagram tersebut dapat dibuat dalam persamaan matematis sebagai berikut[18]:

$$out(z) = a * in(z) + b * out(z-nT) \qquad (3.1)$$

Dimana persamaan 3.1 dalam domain Z dan T menunjukkan waktu cuplik. Jadi keluaran sekarang bergantung dari keluaran sebelumnya yang sudah mengalami penundaan/*delay*.

Dalam menggambarkan aplikasi efek *reverb* ini, kita dapat menggunakan SIMULINK®. Dalam aplikasi yang dirancang, komponen *gain* harus ditambahkan untuk memperjelas efek yang terjadi. Bila tidak ada *gain* maka yang terjadi adalah efek yang diproduksi tidak terlalu berpengaruh. Berikut ini diagram blok yang mewakili efek *reverb*:

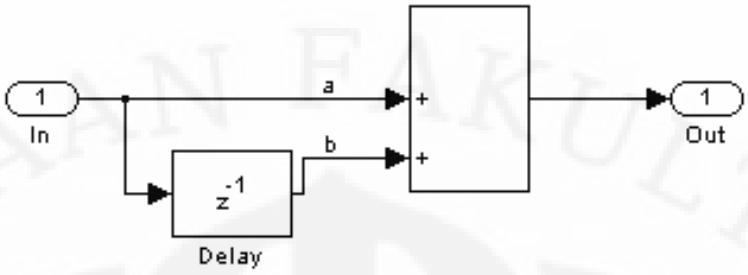


Gambar 3.2 Blok Diagram Implementasi *Reverberation* [17]

Pada Gambar 3.2 terdapat dua blok *gain* dimana `GainB` menunjukkan penguatan dari masukan yang sudah mengalami *delay* lalu dilanjutkan dengan `Delay Mix` yang merupakan tahap selanjutnya dari penguatan yang terjadi. Hal ini dilakukan agar keluaran, ketika sudah digabung dengan masukan sinyal asli, dapat menghasilkan efek yang diharapkan.

3.1.2 *Echo*

Efek *Echo* memiliki blok dasar seperti pada Gambar 3.3.



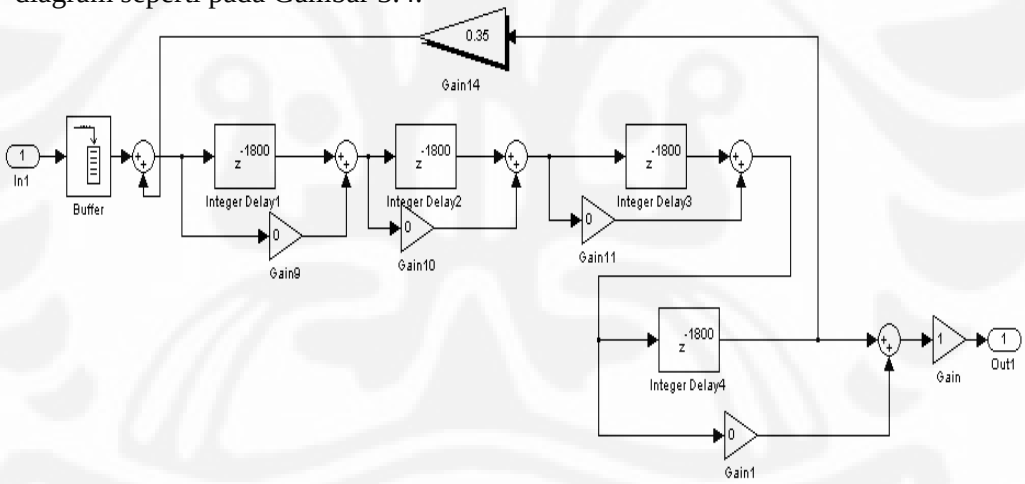
Gambar 3.3 Blok Dasar *Echo*

Konsep dasar efek *echo* yaitu mencampurkan suara asli dengan suara masukan yang sudah mengalami penundaan/*delay*. Pada Gambar 3.3 dapat dibuat dalam bentuk matematis, yaitu sebagai berikut [18]:

```
out(z) = a * in(z) + b * in(z-nT)
```
(3.2)

Pada persamaan 3.1 maupun 3.2 konstanta a dan b menunjukkan nilai *gain*.

Dalam menggambarkan aplikasi efek *echo* ini, kita dapat menggunakan SIMULINK®. Berikut ini merupakan efek *echo* yang digambarkan dengan blok diagram seperti pada Gambar 3.4.



Gambar 3.4 Blok Diagram Implementasi *Echo*[19]

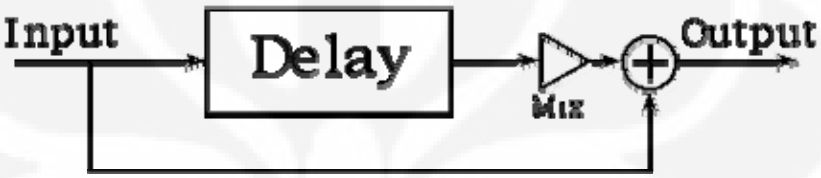
3.2 *CHORUS*

Chorus merupakan efek yang menghasilkan suara terdengar seperti campuran beberapa suara instrumen yang dimainkan bersama. Dalam implementasinya mirip dengan *echo*. Masukan dicampur dengan masukan hasil delay untuk menghasilkan keluaran, bedanya dengan *echo* adalah *sampling rate* berubah secara perlahan.

Chorus dalam waktu yang sebenarnya (*real time*) sangat mudah dalam mengontrolnya. Input yang sudah di-*sampling* dalam *rate* yang sudah tetap (*fixed*) lalu dilakukan interpolasi maka akan menghasilkan output *chorus.*

Efek *Chorus* dapat ditingkatkan dengan menambahkan ukuran *delay* yang berbeda-beda dimana hal ini akan meningkatkan jumlah "suara" yang bercampur untuk menghasilkan efek *chorus*.

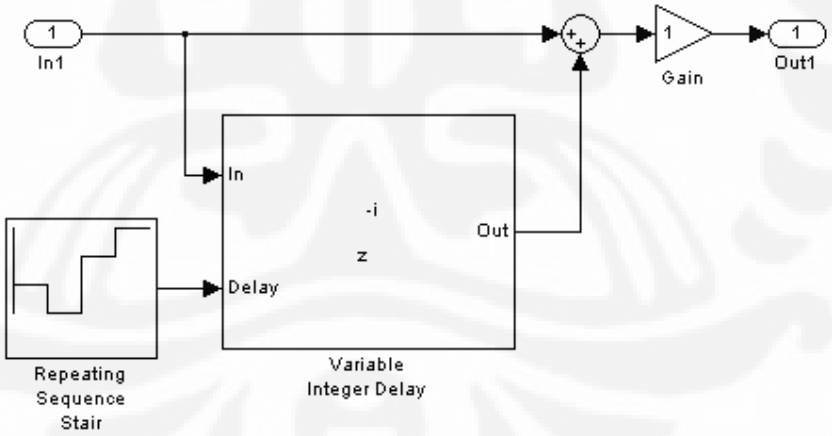
Secara sederhana efek *chorus* dapat diwakilkan oleh blok diagram pada Gambar 3.5



Gambar 3.5 Blok diagram *chorus* secara sederhana[14]

Chorus memiliki komponen LFO *(Low Frequency Oscillator)* dimana hal ini nantinya akan mengontrol penundaan yang akan terjadi seiring dengan perubahan waktu yang terjadi. Hal ini dapat diwakilkan oleh Gambar 2.12.

Bila kita menggambarkan blok diagram dengan menggunakan SIMULINK®, maka akan didapatkan seperti pada Gambar 3.6.



Gambar 3.6 Blok Diagram Implementasi *Chorus*

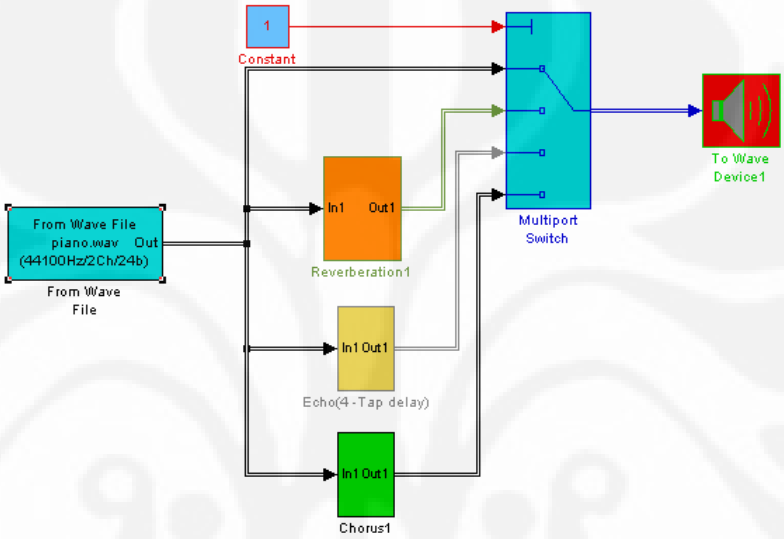
3.3 PERANCANGAN SISTEM KESELURUHAN

Perancangan ini dilakukan agar pengguna dapat memilih efek musik yang diinginkan. Karena itu dibutuhkan suatu simulasi awal untuk menguji sistem terlebih dahulu sebelum diimplementasikan dengan DSK *Board*. Untuk itu kita dapat menggunakan blok *Wave Device* dimana memungkinkan untuk mendengar

terlebih dahulu rancangan yang sudah dibicarakan sebelumnya. Setelah itu, *Wave Device* akan digantikan dengan blok DAC yang terdapar dalam *library* C6000 dimana blok ini akan diimplementasikan ke dalam DSK *Board*. Untuk suara masukan, digunakan *tone* 1 KHz dan sinyal yang tidak periodik.

3.3.1 Rancangan Simulasi Tanpa Alat

Gambar 3.7 menunjukkan rancangan untuk simulasi internal



Gambar 3.7 Blok Diagram Sistem Keseluruhan Simulasi Internal

Gambar 3.7 merupakan blok yang menggambarkan simulasi internal dimana belum menyertakan komponen DSK *board* dan masih merupakan gabungan sementara. Hal ini dilakukan untuk melihat hasil simulasi apakah dapat berjalan dengan baik. Beberapa blok yang penting akan dijelaskan pada bagian berikutnya.

3.3.1.1 *Multiport Switch*

Blok ini merupakan bagian yang penting, dimana pengguna akan dapat memilih efek yang akan terjadi. Input pada blok ini dapat diatur, dalam hal ini terdapat empat masukan. Lalu sinyal kontrol diatur oleh blok `constant`, dimana nilai yang tersedia merupakan nilai *integer* yang merupakan penunjuk urutan untuk setiap urutan efek. Dalam hal ini untuk efek *reverb* adalah 2, efek *echo* adalah 3, dan efek *chorus* adalah 4. Sedangkan 1 untuk input yang masih asli.

3.3.1.2 *Constant*

Blok ini digunakan untuk menunjukkan nilai konstan. Pada Gambar 3.7 digunakan untuk mengontrol alur efek yang diinginkan.

3.3.1.3 *From Wave File*

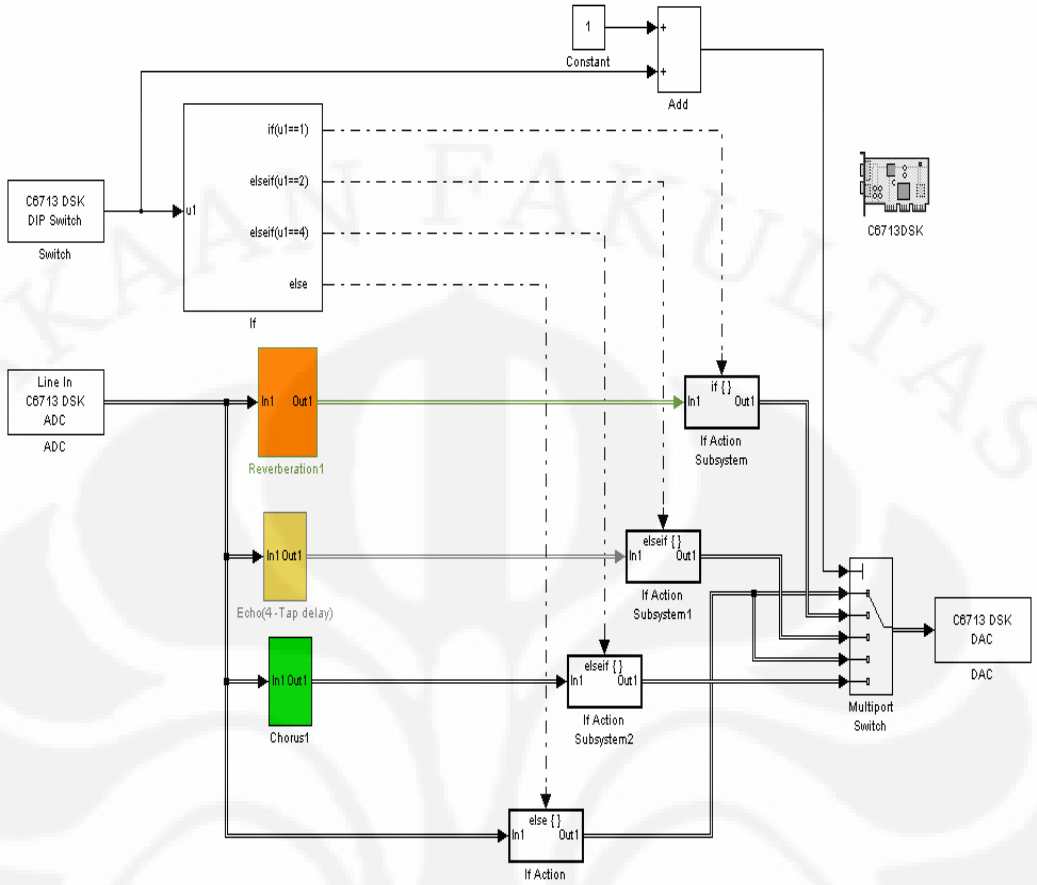
Blok ini berfungsi untuk membaca data yang berupa musik dengan *extension* wav. Blok ini digunakan sementara untuk mensimulasikan dengan SIMULINK® dan belum diaplikasikan dengan alat**.** Bila akan disimulasikan dengan alat DSK, maka blok ini akan diganti dengan blok ADC.

3.3.1.4 *To Wave Device*

Blok ini berfungsi sebagai pengganti speaker pada simulasi. Blok ini digunakan sementara untuk mensimulasikan dengan SIMULINK® dan belum diaplikasikan dengan alat DSK. Bila akan disimulasikan dengan alat DSK, maka blok ini akan diganti dengan blok DAC.

3.3.2 Rancangan Simulasi Dengan Alat DSK TMS320C6713

Pada gambar 3.8 menunjukkan blok diagram system secara keseluruhan. Sistem tersebut akan diaplikasikan dengan alat DSK TMS320C6713. Penggunaan alat tersebut akan memungkinkan system yang dirancang berjalan dengan *real-time*.



Gambar 3.8 Blok Diagram Simulasi Dengan DSP *Board*

Berikut ini beberapa penjelasan blok yang dipakai dalam model Gambar 3.8

3.3.2.1 *Line In* C6713 DSK ADC

Blok ini merupakan blok yang berfungsi untuk mengubah sinyal analog menjadi sinyal dijital. Penggunaan blok ini mengaktifkan penggunaan *audio coder-decoder(codec) module* pada C6713 DSK. *Codec* pada C6713 DSK biasa disebut AIC23 *codec*. Blok ini bekerja dengan cara *frame-based* dimana menempatkan sementara (*buffering*) data *input* ke dalam beberapa *frame* dengan nilai *sample per frame* yang sudah ditentukan. Dalam Simulink, bila input berupa data *monoaural* maka akan dibentuk menjadi vektor kolom dengan sebesar N-element. Bila input berupa data *stereo*, maka akan dibentuk matrix Nx2 dengan sebesar N-data dan dua menunjukkan *channel stereo*. Pada blok ini terdapat beberapa hal yang penting antara lain:

1. Penggunaan *stereo* atau *monoaural* sebagai input. Bila input adalah *stereo* maka kotak *stereo* harus dicek.

2. Penggunaan mic sebagai input harus diberi tambahan 20 dB.
3. *Sampling rate* adalah 44,1 KHz dengan *samples per frame* adalah sebesar 256. Hal ini menunjukkan besar nilai *frame rate* yaitu 44100/256 = 172,265625.

3.3.2.2 *Line Out* C6713 DSK DAC

Blok ini berguna untuk mengubah sinyal dijital menjadi sinyal analog kembali. Blok ini mengkonfigurasi AIC23 *codec* yang dimiliki oleh DSK C6713 untuk mengirim sinyal yang sudah mengalami pengubahan ke *jack output* DSK C6713 yang dihubungkan dengan *speaker*.

3.3.2.3 C6713 DIP *Switch*

Blok ini merupakan penggambaran dari alat yang sesungguhnya yang memiliki *dual inline pin* (DIP). DIP sangat bermanfaat dalam simulasi-simulasi yang berhubungan dengan pemrosesan sinyal. DIP terdiri dari empat yaitu `Switch0`, `Switch1`, `Switch2`, dan `Switch3`. `Switch0` merupakan LSB dan `Switch3` merupakan MSB. Hal ini sangat berpengaruh dalam besar *output* dari DIP yang bernilai integer yaitu dari 0-15. Angka tersebut diperoleh dari kombinasi keempatnya yaitu $2^4 = 16$. Sehingga terdapat 16 nilai yaitu dari 0-15. Berikut ini Tabel 3.1 yang menunjukkan beberapa kombinasi dari DIP.

Tabel 3.1 Penggunaan *Switch* DIP pada DSK

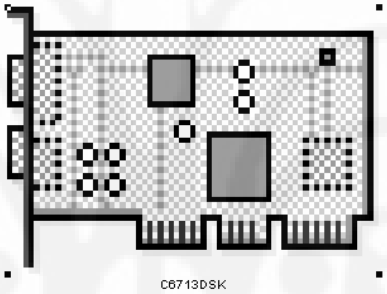
Switch 0 (LSB)	**Switch 1**	**Switch 2**	**Switch 3 (MSB)**	***Boolean Output***	***Integer Output***
Kosong	Kosong	Kosong	Kosong	0 0 0 0	0
Cek	Kosong	Kosong	Kosong	0 0 0 1	1
Kosong	Cek	Kosong	Kosong	0 0 1 0	2
Cek	Cek	Kosong	Kosong	0 0 1 1	3
Kosong	Kosong	Cek	Kosong	0 1 0 0	4

3.3.2.4 *Add*

Blok ini berfungsi untuk menjumlah dua sinyal masukan atau lebih. Sinyal masukan bisa berupa konstan ataupun kontinu. Dalam blok pada Gambar 3.8, berfungsi untuk menambahkan nilai *integer* yang keluar dari blok DIP SWITCH. Nilai yang ditambahkan ini akan menyesuaikan dengan nilai *integer* pada *Multiport Switch* dimana nilai harus sesuai dengan *port* yang dilewati. Bila tidak maka sistem tidak akan berjalan dengan semestinya.

3.3.2.5 *Target* C6713 DSK

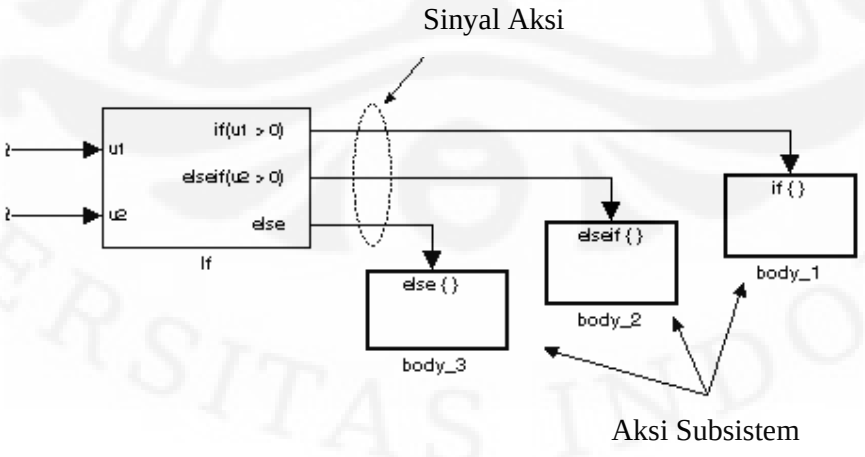
Blok ini merupakan blok penting karena SIMULINK® dapat menjalankan simulasi dengan alat melalui blok ini. Gambar 3.9 menunjukkan gambar *hardware* yang digunakan untuk menjalankan simulasi secara *real-time*.



Gambar 3.9 Blok Target C6713 DSK

3.3.2.6 *If* dan *If Action Subsystem*

Kedua blok ini saling berpasangan karena keduanya saling melengkapi dari segi fungsinya dalam memodelkan fungsi *If*. Kedua blok ini dapat dilihat pada Gambar 3.10 sebagai berikut:



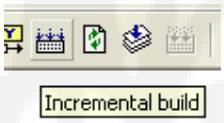
Gambar 3.10 Penggunaan blok *If* dan *If Action Subsystem*

Gambar 3.10 menunjukkan masukan ada 2 dan masing-masing diberikan kondisi. Bila kondisi yang terjadi terpenuhi maka akan melanjutkan ke aksi subsistem. Begitu juga dengan rancangan sesuai dengan gambar 3.8, dimana terdapat 4 aksi subsistem. Masing-masing subsistem menggambarkan sistem yang akan terjadi pada alat C6713 DSK yaitu:

1. Bila DIP *Switch* bernilai satu maka efek *reverb* akan dijalankan.
2. Bila DIP *Switch* bernilai dua maka efek *echo* akan dijalankan.
3. Bila DIP *Switch* bernilai empat maka efek *chorus* akan dijalankan.
4. Selain daripada itu maka akan menghasilkan bunyi asli.

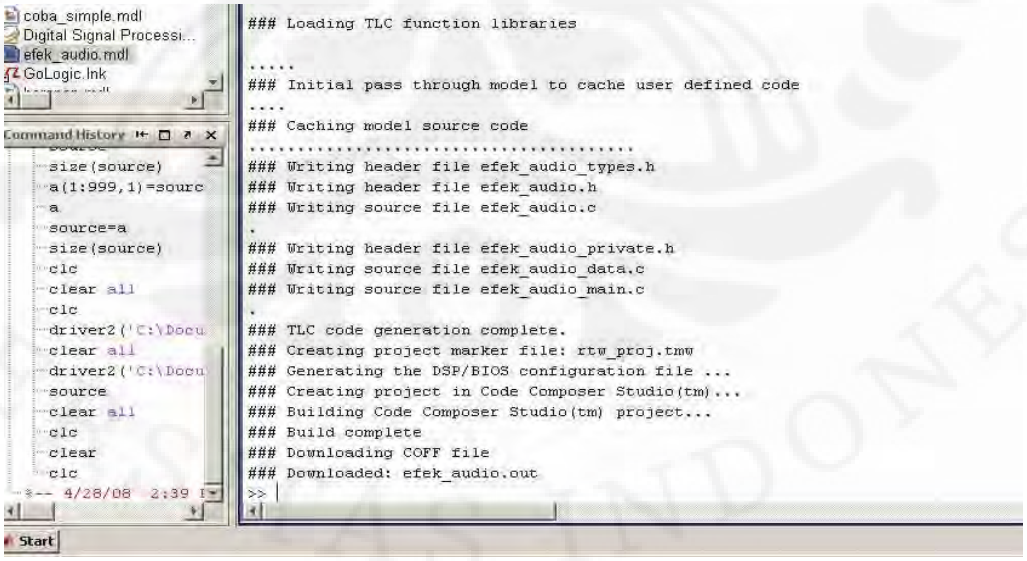
3.3.3 Prosedur *Targetting* C6713 DSK

Sesuai dengan penjelasan pada subbab 2.2.1, maka hasil perancangan dapat diterapkan pada alat. Gambar 3.11 menunjukkan tombol yang harus ditekan ketika kita akan mengaplikasikannya ke alat. Hal tersebut dapat dilakukan dengan menekan tombol Ctrl+B.



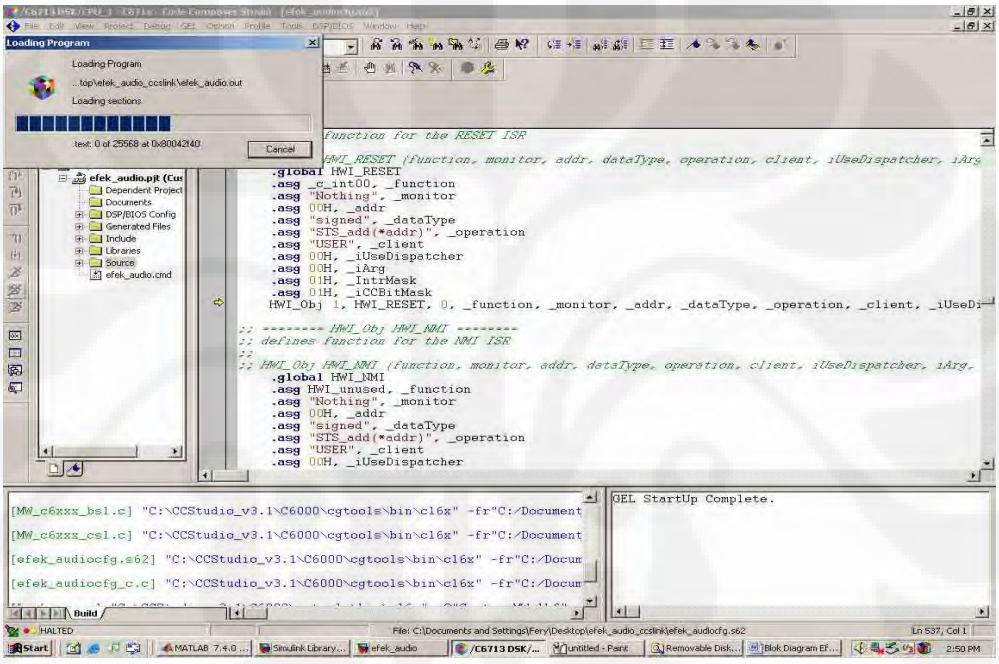
Gambar 3.11 Tombol *Incremental Build*

Setelah itu akan ditampilkan proses yang terjadi seperti pada Gambar 3.12 seperti di bawah ini.



Gambar 3.12 Proses *Incremental Build*

Incremental Build yang dilakukan oleh MATLAB© akan memanggil program CCS 3.1 yang sebelumnya sudah terinisialisasi di dalam komputer. Proses selanjutnya dilakukan oleh *software* CCS 3.1 untuk menghasilkan kode dengan bahasa C. Gambar 3.13 menunjukkan proses *build* yang selanjutnya akan ditangani oleh CCS dimana akan menghasilkan *file* dengan ekstensi out yang merupakan hasil dari rancangan yang dibuat di MATLAB©. *File* inilah yang akan digunakan untuk diaplikasikan ke *hardware*.



Gambar 3.13 Proses *Load Program*

BAB IV
HASIL UJI COBA DAN ANALISIS
HASIL SIMULASI

Suara yang dijadikan *sample* dalam simulasi ini ada dua. Suara tersebut adalah *tone* 1 KHz dan *sample* selanjutnya adalah suara yang memiliki grafik yang acak dan tidak periodik. Semua suara *sample* dibuat dengan program editor audio *Cool Edit Pro* 2.0. Semua suara *sample* memiliki *audio sample rate* 44100 Hz dan ukuran ketelitian *sample* adalah 32 bit. Pemilihan ini berdasarkan besarnya *sample rate* standar pada *file* audio dan ukuran ketelitian dari *sample* audio.

Dalam melakukan uji coba, akan dilakukan perbandingan antara hasil simulasi tanpa alat (dengan SIMULINK®) dengan hasil simulasi dengan alat DSK C6713. Simulasi tanpa alat dibagi menjadi dua dengan empat kondisi yang berbeda berdasarkan efek audio yang akan dievaluasi.

4.1 HASIL UJI COBA DAN ANALISIS HASIL SIMULASI SIMULINK®

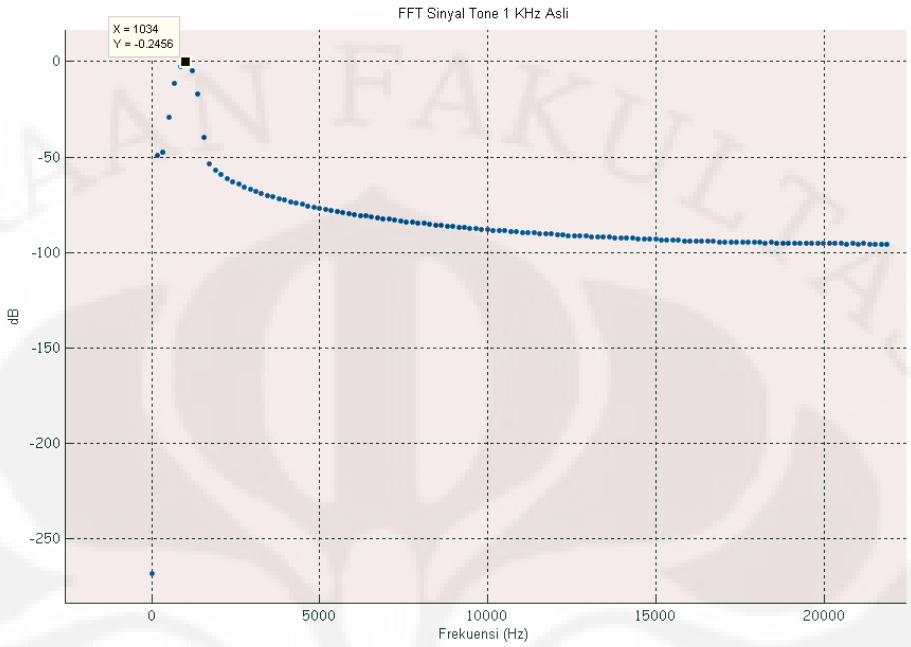
Hasil uji coba akan dibagi menjadi dua yaitu simulasi dengan SIMULINK® dan simulasi dengan menggunakan DSK C6713. Hasil uji coba dengan SIMULINK® dibagi menjadi dua dengan empat perlakuan yang berbeda. Lalu akan disertai analisis untuk masing-masing hasil yang ada.

4.1.1 Hasil Uji Coba *Tone* 1 KHz Dengan SIMULINK®

Untuk suara *tone* 1 KHz yang periodik akan diberi empat perlakuan yang berbeda yaitu suara asli, suara hasil efek *reverb*, suara hasil efek *echo*, dan suara hasil efek *chorus*.

4.1.1.1 Suara *tone* 1 KHz

Untuk suara *tone* 1 KHz akan menghasilkan gelombang dengan T = 0.001 detik. Untuk analisis *tone* 1 KHz maka diperlukan grafik dari FFT gelombang suaranya. Gambar 4.1 menunjukkan hasil FFT dari gelombang suara 1 KHz. Pengolahan untuk gambar dilakukan di MATLAB.

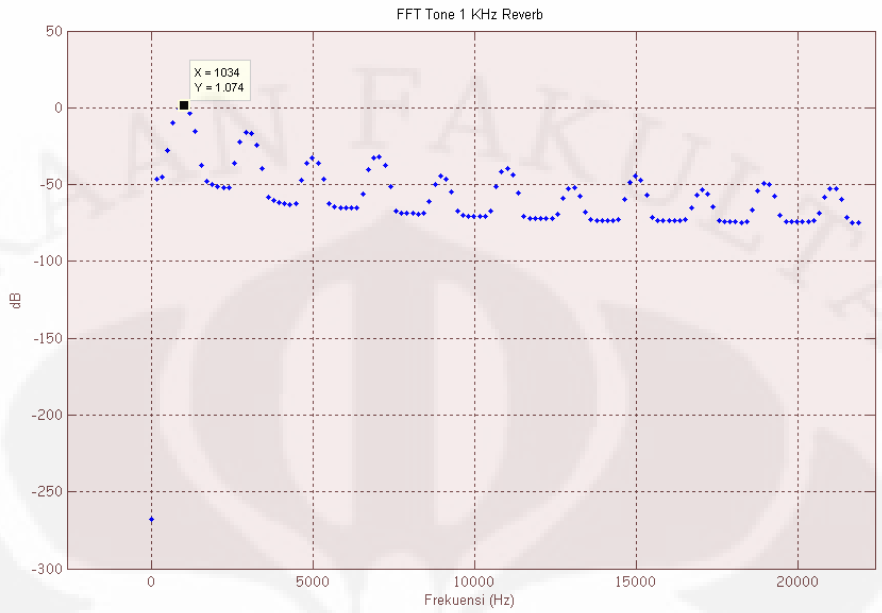


Gambar 4.1 FFT Gelombang 1 KHz

Pada Gambar 4.1 menunjukkan frekuensi fundamental adalah 1034 Hz dimana mendekati nilai 1 KHz dan intensitas suara 0 dB. Suara yang ada merupakan suara yang belum diberikan efek audio.

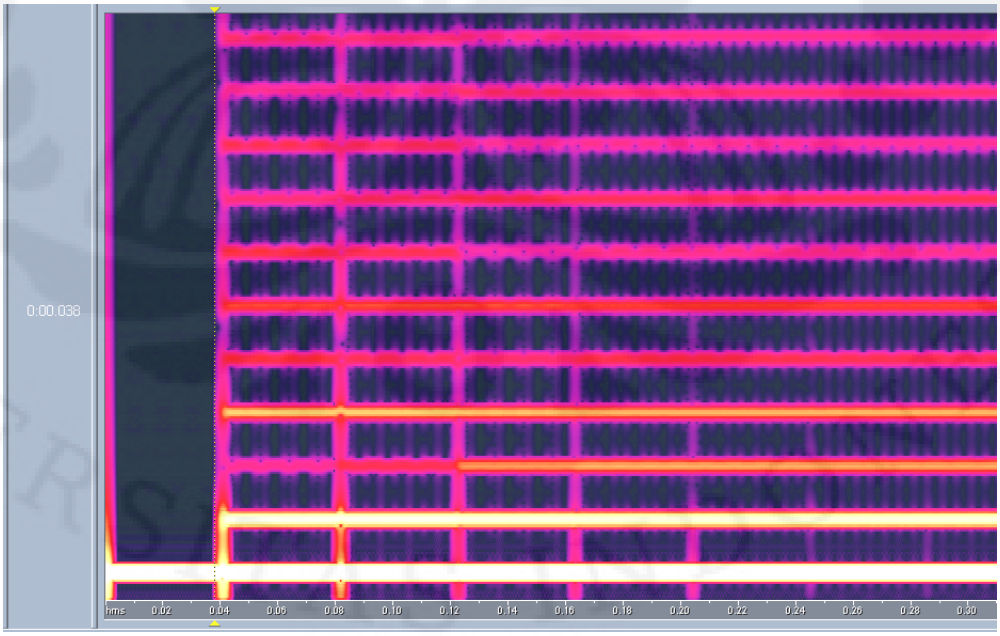
4.1.1.2 Suara r*everb tone* 1 KHz

Gambar 4.2 menunjukkan hasil FFT dari gelombang 1 KHz setelah diberi efek *reverb*. Bila dibandingkan dengan Gambar 4.1 maka terdapat frekuensi harmonik yang terjadi. Frekuensi harmonik terjadi pada frekuensi ganjil seperti pada frekuensi 1 KHz, 3 KHz, 5KHz, dst.



Gambar 4.2 FFT Gelombang 1 KHz setelah diberi efek *reverb*

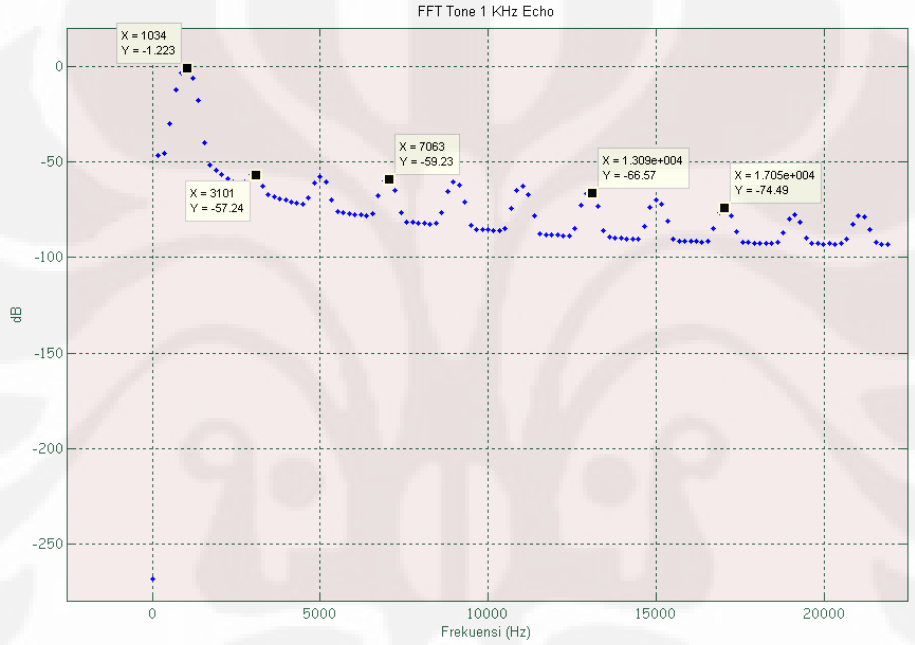
Dalam domain waktu, hasil efek *reverb* lebih dapat dievaluasi. Gambar 4.3 menunjukkan sinyal keluaran dalam domain waktu. Dalam efek *reverb* terdapat dua hal yang bisa didapatkan yaitu nilai *predelay* dan *reverb* decay. Nilai *predelay* sebesar 38 ms dan nilai *reverb decay* sebesar 286 ms. Nilai ini merupakan nilai hasil simulasi berdasarkan visualisasi oleh program *Cool Edit Pro* 2.0.



Gambar 4.3 Sinyal hasil keluaran dari efek *reverb* dalam domain waktu

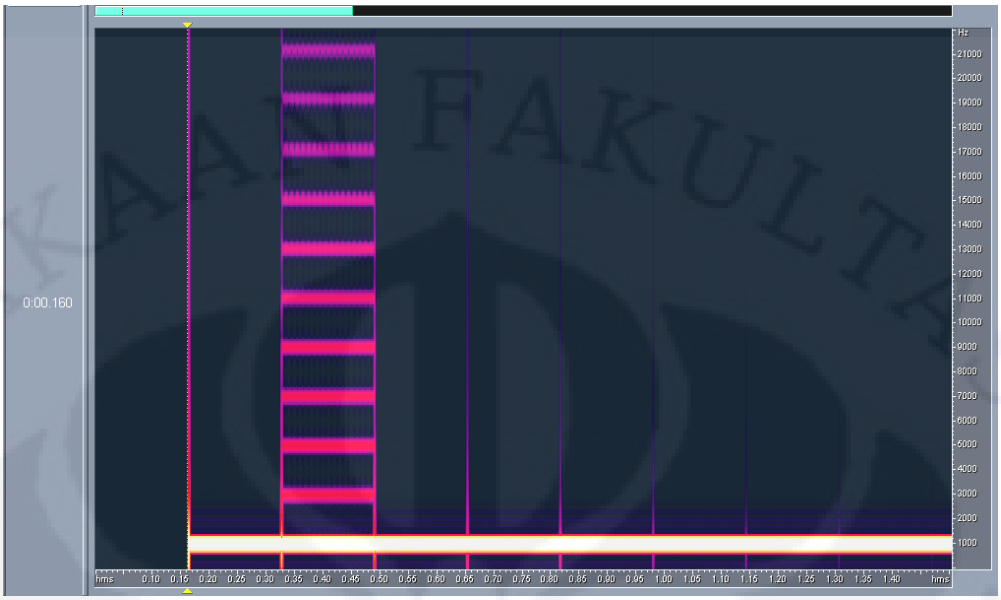
4.1.1.3 Suara *echo tone* 1 KHz

Gambar 4.4 menunjukkan FFT dari output sinyal *tone* 1 KHz yang sudah mengalami efek *echo*. Bila dibandingkan dengan Gambar 4.1, maka terdapat pelemahan sinyal yang cukup signifikan. Pengurangan yang terjadi akibat nilai masukan yang terjadi diberikan *delay*. Pada Gambar 4.4 juga terdapat frekuensi harmonik yang terjadi pada frekuensi ganjil. Kondisi ini serupa dengan hasil keluaran *reverb*.



Gambar 4.4 FFT Gelombang 1 KHz setelah diberi efek e*cho*

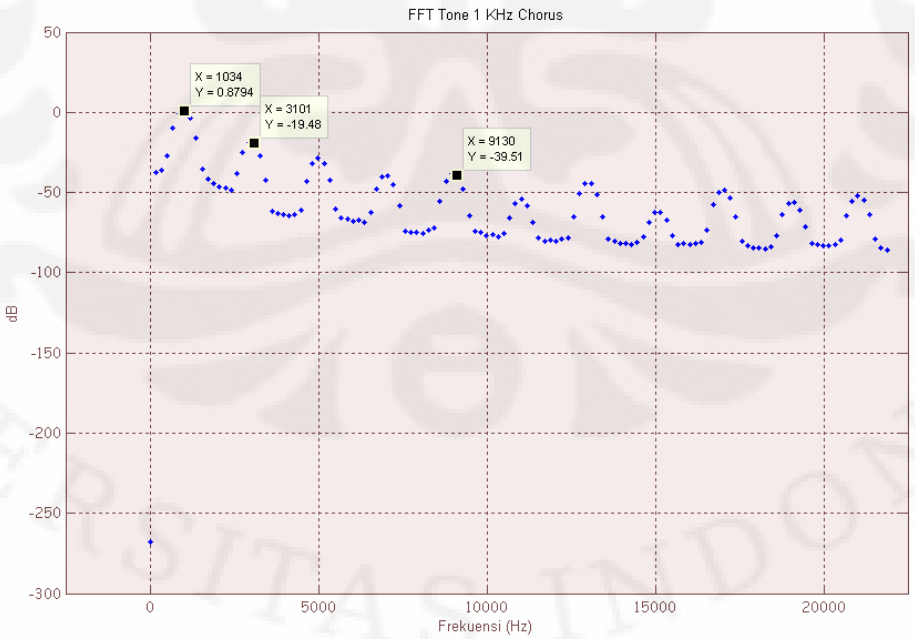
Visualisasi grafik hasil keluaran efek *echo* akan lebih mudah dilihat dalam domain waktu sesuai dengan Gambar 4.5. Sinyal baru masuk pada detik ke 0,16. Hal ini membuktikan bila masukan diberi penundaan/*delay*.



Gambar 4.5 Sinyal hasil keluaran dari efek *echo* dalam domain waktu

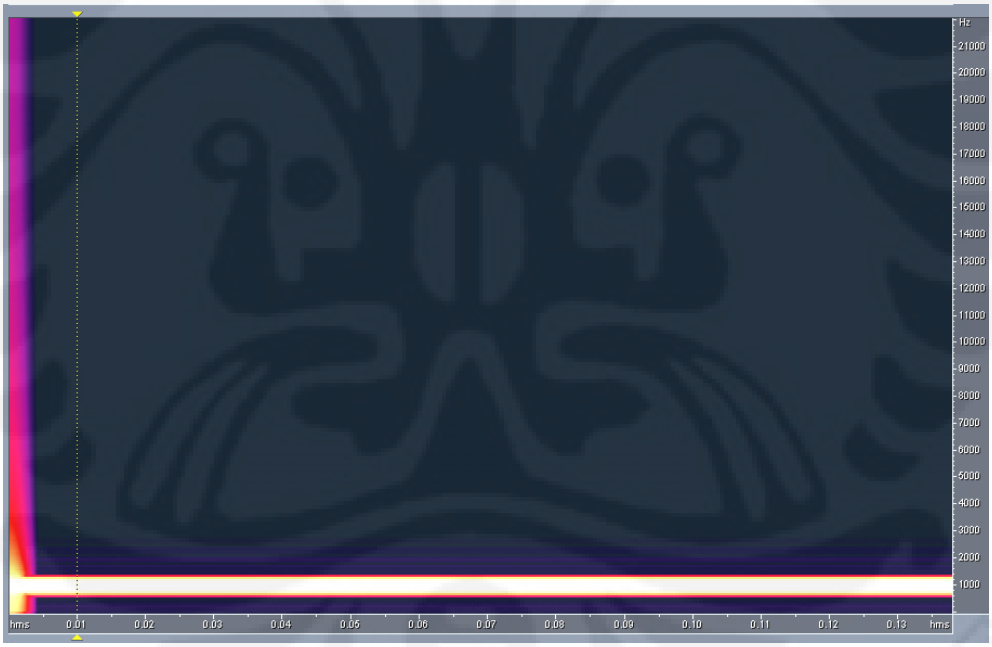
4.1.1.4 Suara c*horus tone* 1 KHz

Hasil suara 1 KHz yang diberikan efek *chorus* dapat dilihat pada Gambar 4.6. Bila dibandingkan dengan Gambar 4.1, maka terdapat peningkatan intensitas suara untuk frekuensi 1 KHz sebesar 1,125 dB sehingga menjadi 0,8794 dB. Hal ini membuktikan efek *chorus* menghasilkan suatu efek tambahan pada suatu gelombang bunyi. Hal inilah yang menyebabkan kenaikan intensitas suara.

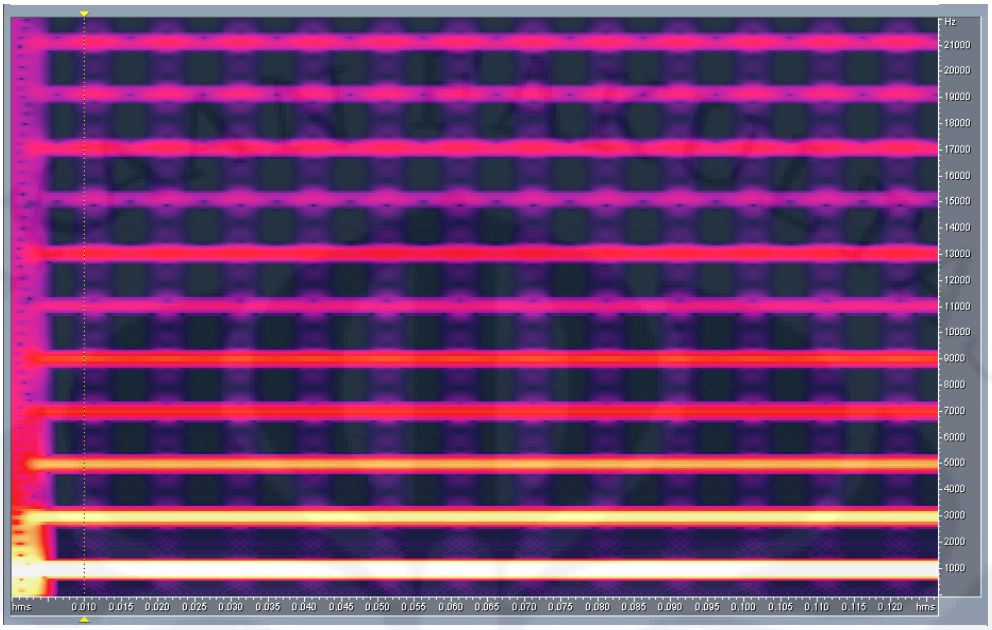


Gambar 4.6 FFT Gelombang 1 KHz setelah diberi efek *chorus*

Gambar 4.6 menunjukkan grafik dalam domain frekuensi. Agar lebih mudah untuk melihat perubahan yang terjadi maka analisa dalam domain waktu. Gambar 4.7 menunjukkan grafik sinyal *tone* dalam domain waktu. Sedangkan Gambar 4.8 menunjukkan grafik sinyal *tone* setelah diberikan efek *chorus*. Kedua gambar tersebut sangat jelas mempelihatkan perbandingan antara sinyal asli dengan sinyal yang telah diberi efek *chorus*. Sinyal asli tidak mengalami penundaan yang bersifat terus-menerus, sehingga tidak terlihat efek penundaan pada detik ke 0,01. Sedangkan pada Gambar 4.8, hasil dari efek *chorus* menghasilkan suatu penundaan yang terus menerus terjadi dimulai dari detik ke 0,01. Hal ini menunjukkan suatu campuran suara yang terjadi karena adanya suatu penundaan. Hal ini membuktikan bahwa suara *chorus* dihasilkan dengan memberikan penundaan pada masukan terus-menerus berdasarkan waktu. Dimana satu efek penundaan berarti satu campuran suara.



Gambar 4.7 Grafik sinyal *tone* 1 KHz dalam domain waktu



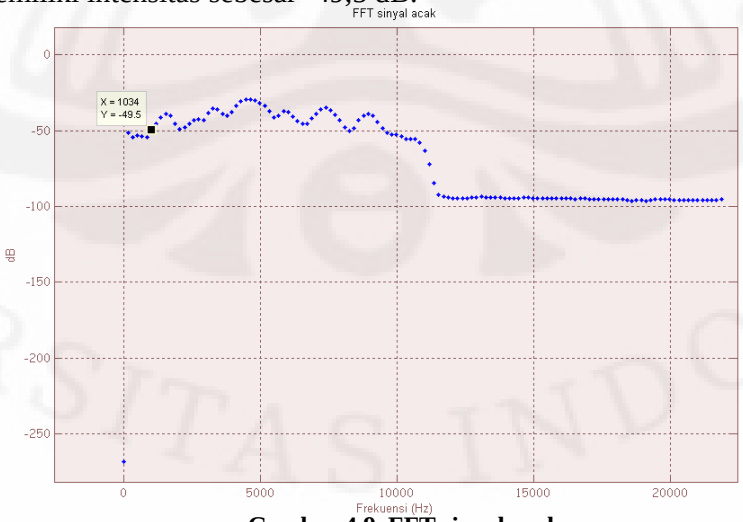
Gambar 4.8 Grafik sinyal *tone* setelah diberi efek *chorus* dalam domain waktu

4.1.2 Hasil Uji Coba Sinyal Acak Dengan SIMULINK®

Sinyal acak yang diuji merupakan sinyal yang memiliki rentang nilai frekuensi yang acak dan tidak periodik. Sinyal acak akan diberi perlakuan yang berbeda-beda tergantung dari efek yang diberikan.

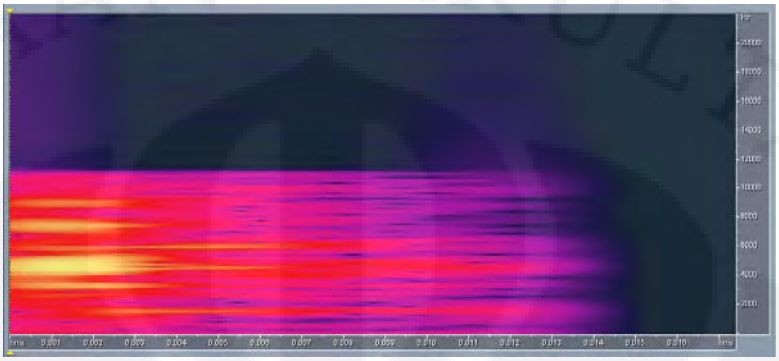
4.1.2.1 Sinyal acak

Gambar 4.9 menunjukkan grafik FFT dari sinyal acak. Pada frekuensi 1 KHz memiliki intensitas sebesar -49,5 dB.



Gambar 4.9 FFT sinyal acak

Gambar 4.10 menunjukkan grafik sinyal acak dalam domain waktu. Analisis dalam domain waktu akan lebih mudah terlihat perbandingan antara efek yang digunakan.

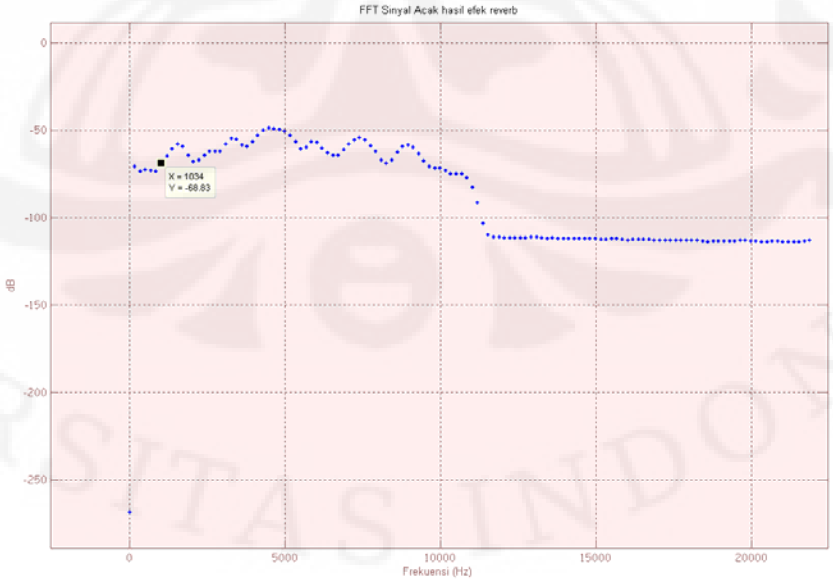


Gambar 4.10 Sinyal acak dalam domain waktu

Bila diperhatikan, sinyal acak tersebut memiliki durasi sepanjang 0,015 detik.

4.1.2.2 Sinyal acak *reverb*

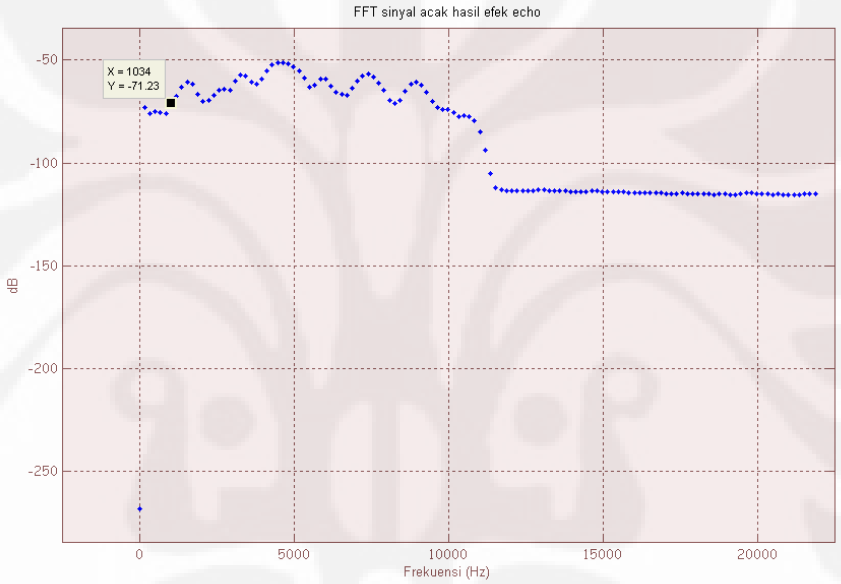
Gambar 4.11 menunjukkan perbedaan dengan sinyal acak asli. Pada frekuensi 1 KHz terdapat penurunan intensitas suara sebesar 19,33 dB menjadi sebesar -68,83 dB. Sehingga untuk *reverb* terdapat pelemahan intensitas suara yang disebabkan oleh penundaan.



Gambar 4.11 FFT sinyal acak hasil efek *reverb*

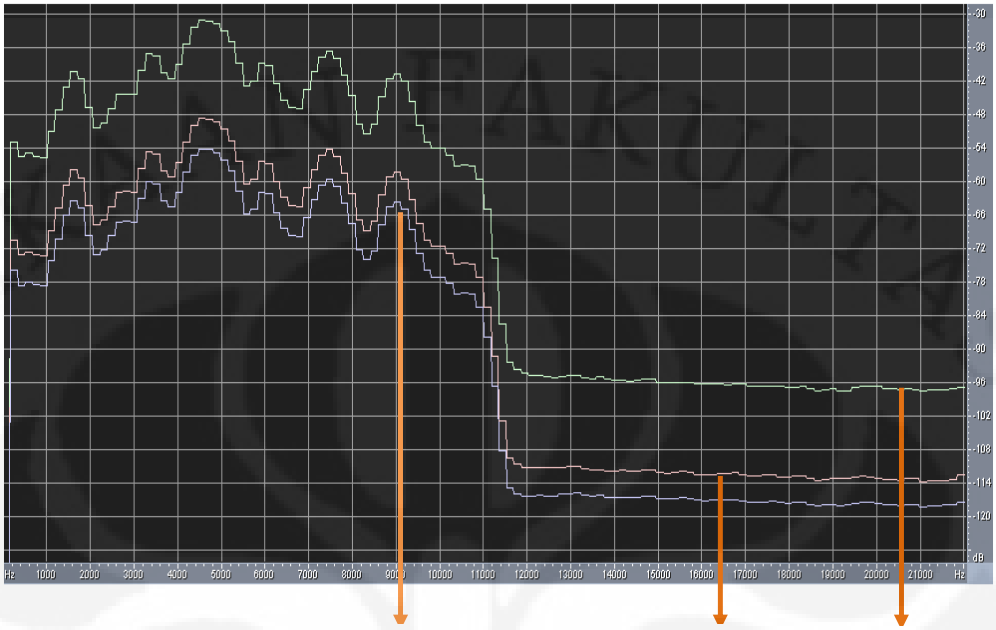
4.1.2.3 Sinyal acak *echo*

Pada sinyal acak *echo*, sinyal asli akan diberikan perlakuan model efek *echo* yang sama seperti pada subbab 4.1.1.3. Gambar 4.12 menunjukkan FFT sinyal acak dari hasil efek *echo* dalam domain frekuensi. Pada frekuensi 1 KHz terdapat penurunan intensitas yang sangat signifikan menjadi sebesar -71,23 dB. Hal ini menunjukkan bahwa sinyal masukan mengalami penundaan dahulu dan akan dicampur dengan sinyal asli yang nantinya akan menjadi keluaran. Bila dibandingkan dengan sinyal acak hasil efek *reverb*, nilai intensitas berkurang banyak pada hasil efek *echo*.



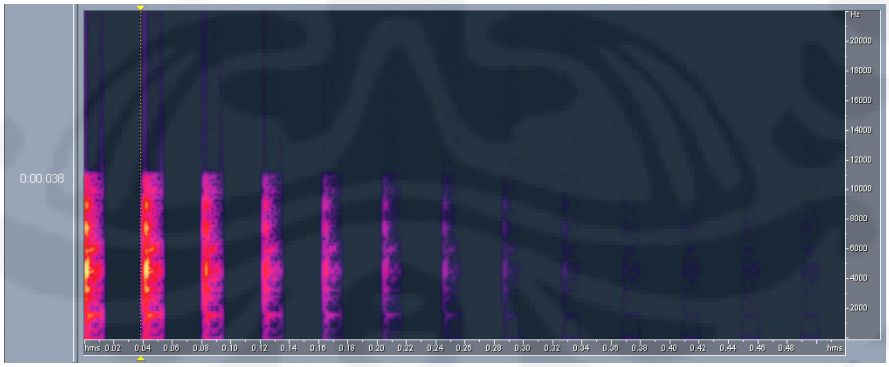
Gambar 4.12 FFT sinyal acak hasil efek *echo*

Gambar 4.13 menunjukkan perbandingan kedua efek yang masih dalam klasifikasi *delay based effect*. Sinyal yang berwarna hijau menunjukkan sinyal acak yang asli. Sinyal yang berwarna merah menunjukkan sinyal acak hasil efek *reverb*. Sinyal yang berwarna ungu menunjukkan sinyal acak hasil efek *echo*. Terlihat bahwa terjadi penurunan intensitas suara, penurunan intensitas suara ini dalam praktis kehidupan sehari-hari bias ditangani dengan *amplifier*.



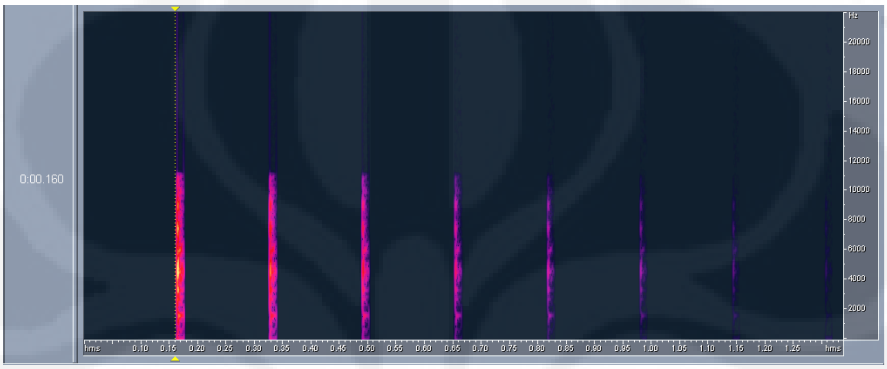
Gambar 4.13 Perbandingan efek *reverb* dengan *echo* dalam domain frekuensi

Pada domain waktu, hasil efek *reverb* dan efek *echo* akan lebih mudah dilihat perbandingannya. Gambar 4.14 menunjukkan hasil sinyal acak yang telah diberikan efek *reverb*. Nilai *predelay* sebesar 380 ms dan *reverb decay* 286 ms. Nilai ini sama dengan analisis subbab 4.1.1.2. Hal ini dikarenakan model yang diberikan kepada kedua suara adalah sama.



Gambar 4.14 Hasil efek *reverb* dalam domain waktu

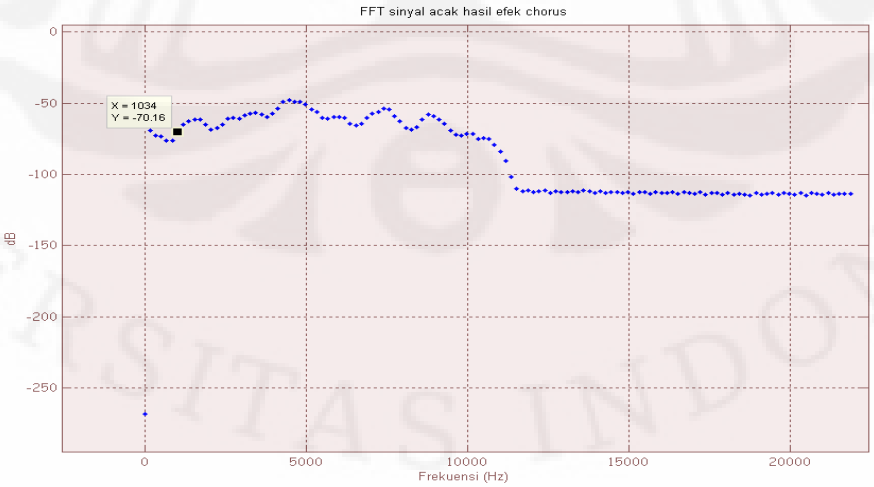
Gambar 4.15 menunjukkan hasil sinyal acak yang telah diberikan efek *echo*. Bila diperhatikan maka hasil efek *echo* memberikan *delay* terlebih dahulu pada masukan lalu dicampur dengan sinyal aslinya. Penundaan yang terjadi sebesar 0,16 s. Nilai ini lebih besar dibandingkan dengan nilai penundaan yang terjadi pada efek *reverb*. Hal ini membuktikan bahwa nilai penundaan yang terjadi pada efek *reverb* lebih kecil daripada nilai penundaan yang terjadi pada efek *echo*.



Gambar 4.15 Hasil efek *echo* dalam domain waktu

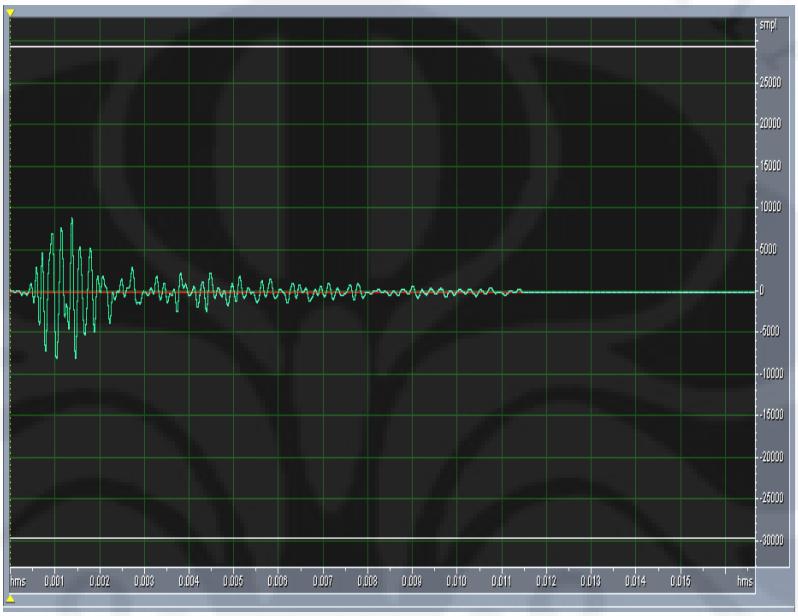
4.1.2.4 Sinyal acak *chorus*

Gambar 4.16 menunjukkan hasil dari efek *chorus* dalam domain frekuensi. Bila dilihat sepintas, grafik tersebut memiliki bentuk yang mirip dengan dua efek sebelumnya dan memiliki nilai intensitas suara yang berkurang juga namun tidak terlalu signifikan bila dibandingkan dengan hasil efek *echo*. Hal pelemahan intensitas suara disebabkan oleh karena adanya komponen penundaan/*delay*. Seperti pada bagian sebelumnya, hal ini dapat diatasi dengan *amplifier*.

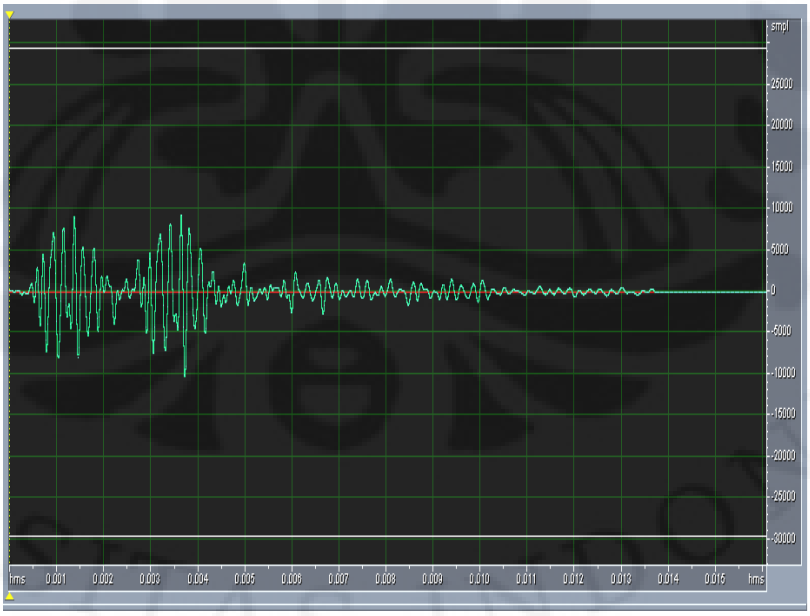


Gambar 4.16 FFT Sinyal acak hasil efek *chorus*

Perbedaan sinyal yang telah diberi efek *chorus* dengan sinyal asli akan lebih mudah dilihat dengan memperhatikan gelombang yang terjadi. Gambar 4.17 menunjukkan sinyal asli yang belum diberi efek apapun. Sedangkan Gambar 4.18 menunjukkan sinyal asli yang telah diberikan efek *chorus*.



Gambar 4.17 Gelombang sinyal asli



Gambar 4.18 Gelombang sinyal asli setelah diberi efek *chorus*

Gambar 4.18 sangat jelas menunjukkan hasil dari pemberian efek *chorus* pada sinyal asli. Pada model *chorus* yang diberikan, hanya terdapat satu bagian LFO (*Low Frequency Oscillator*) sehingga hanya terdapat satu tambahan suara yang baru. Sesuai dengan teori dimana pemberian efek *chorus* akan menghasilkan suatu campuran suara yang lain dimana jumlah campuran suara tersebut dikendalikan oleh LFO. Bila terdapat dua LFO maka terdapat dua tambahan suara.

4.2 PERBANDINGAN GRAFIK SIMULASI DENGAN HASIL PADA DSK C6713

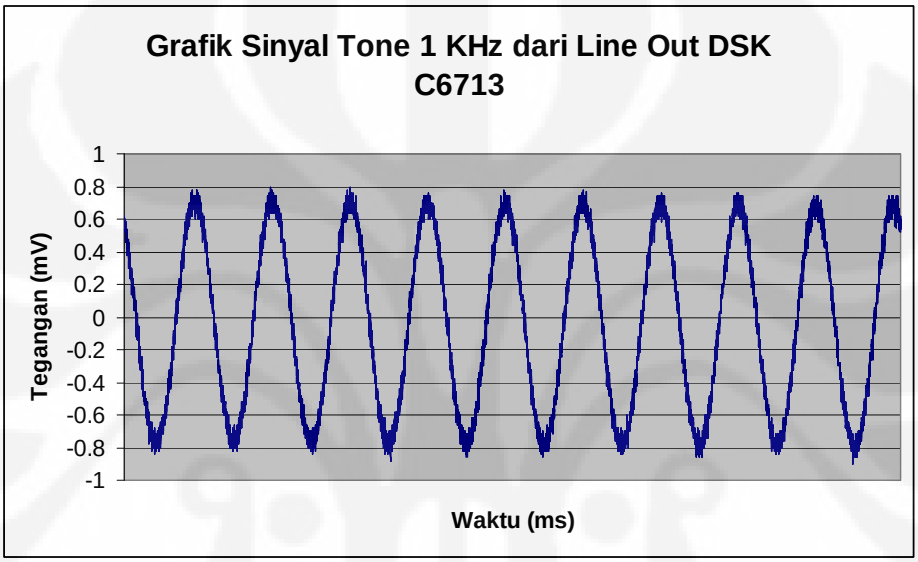
Hasil uji coba dengan DSK C6713 dibagi menjadi dua dengan empat perlakuan yang berbeda dan masing-masing akan disertai dengan analisis. Pengambilan data dilakukan dengan mengirimkan sinyal analog dikirimkan dari komputer dilewatkan ke ADC dari DSK C6713, lalu diberi perlakuan berdasarkan DIP SWITCH yang ditekan. Bila menekan DIP SWITCH 0 maka akan menghasilkan efek *reverb*, bila menekan DIP SWITCH 1 maka akan menghasilkan efek *echo* dan bila menekan DIP SWITCH 2 maka akan menghasilkan efek *chorus*. Bila tidak menekan apa-apa maka akan mengeluarkan bunyi aslinya. Suara yang akan dievaluasi adalah suara *tone* 1 KHz dan sinyal acak. Grafik sinyal diamati dengan menggunakan alat TEKTRONIX *Digital Phosphor Oscilloscope*. Penggunaan alat tersebut memungkinkan untuk mengambil data sebanyak 10000 dengan waktu yang singkat.

4.2.1 Hasil Uji Coba *Tone* 1 KHz

Pada bagian ini akan memperlihatkan hasil dari grafik tiap-tiap sinyal yang sudah mengalami perlakuan yang berbeda. Pada *tone* 1 KHz tidak akan terlalu terlihat signifikan perubahan grafiknya. Hal ini dikarenakan sinyal tersebut konstan dan tidak acak. Namun sinyal acak akan lebih memperlihatkan perbedaan yang cukup siginifikan.

4.2.1.1 Sinyal asli 1 KHz

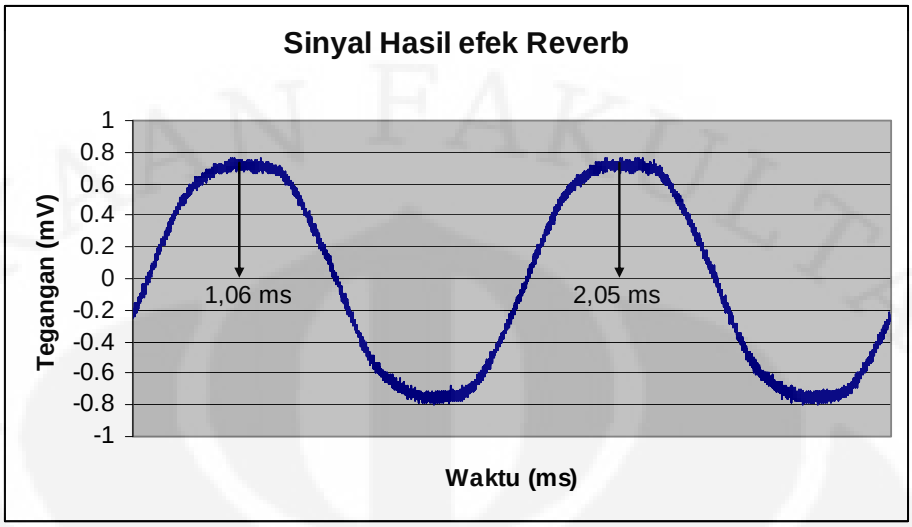
Pada Gambar 4.19 terlihat sinyal yang ditampilkan tidak sehalus seperti pada simulasi menggunakan komputer dan belum dilewati DSK C6713. Pada saat pengolahan didapati bahwa pada frekuensi 1 KHz memiliki intensitas suara sebesar -5,7375 dB. Pengolahan data untuk 30 data pertama terdapat dalam Lampiran.



Gambar 4.19 Grafik sinyal *tone* 1 KHz dari DSK

4.2.1.2 Sinyal hasil efek *reverb*

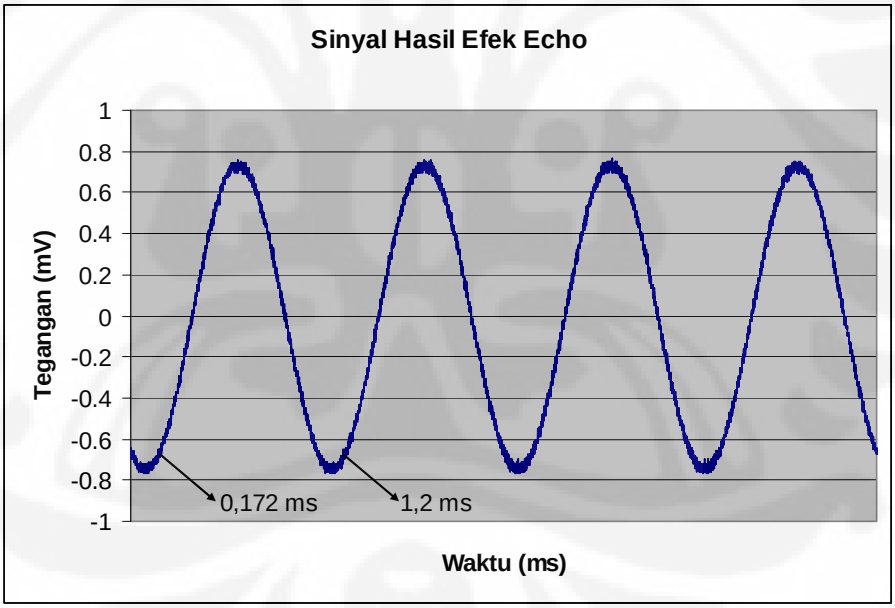
Hasil sinyal yang telah dikenai efek *reverb* dapat dilihat pada Gambar 4.20. Bila diperhatikan kondisi serupa dengan Gambar 4.19 dimana terdapat *noise*. Sehingga hasil tidak semulus sesuai dengan simulasi oleh MATLAB. Pada saat pengolahan data, didapati nilai intensitas suara untuk frekuensi 1 KHz adalah sebesar -5,01288 dB. Hal ini menunjukan kenaikan nilai intensitas suara tetapi tidak terlalu besar. Hal ini bisa disebabkan oleh nilai keluaran yang diberikan penundaan diberikan penguatan sebesar 0,35. Hal ini bisa dilihat dari model pada Gambar 3.2. Untuk frekuensi tidak mengalami perubahan, dalam perhitungan didapati waktu untuk satu gelombang sebesar 0,9964 ms.



Gambar 4.20 Grafik sinyal *tone* 1 KHz hasil efek *reverb*

4.2.1.3 Sinyal hasil efek *echo*

Hasil sinyal yang telah dikenai efek *echo* dapat dilihat pada Gambar 4.21.

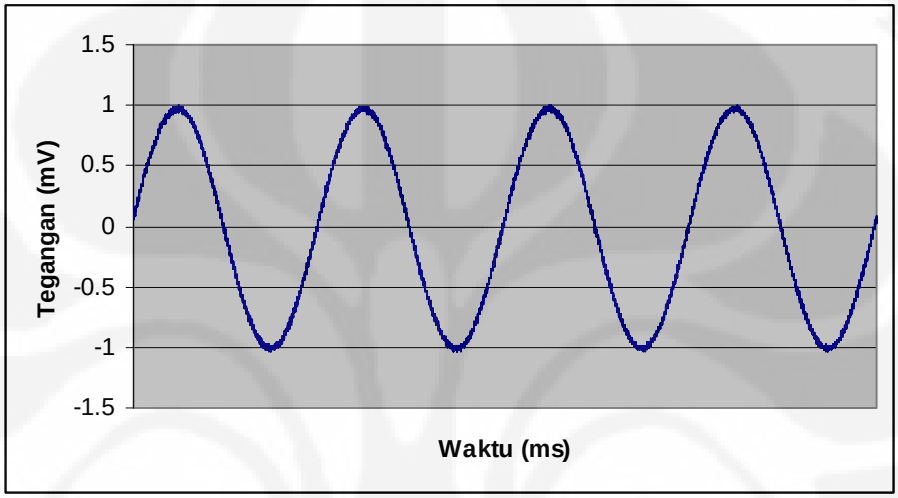


Gambar 4.21 Grafik sinyal setelah diberi efek *echo*

Bila diperhatikan, Gambar 4.21 dan 4.20 tidak jauh berbeda. Untuk waktu selama satu periode yang terjadi tidak jauh berubah menjadi 1,028 ms.

4.2.1.4 Sinyal hasil efek *chorus*

Gambar 4.22 menunjukkan sinyal hasil efek *chorus* dimana mengalami peningkatan tegangan. Hal ini disebabkan karena bercampurnya suara hasil penundaan. Namun karena memiliki rentang waktu periode yang seragam, maka hasilnya hanya berefek pada amplitudo yang meningkat. Selisih sesudah dengan sebelum diberi efek disebut dengan parameter *sweep depth*.



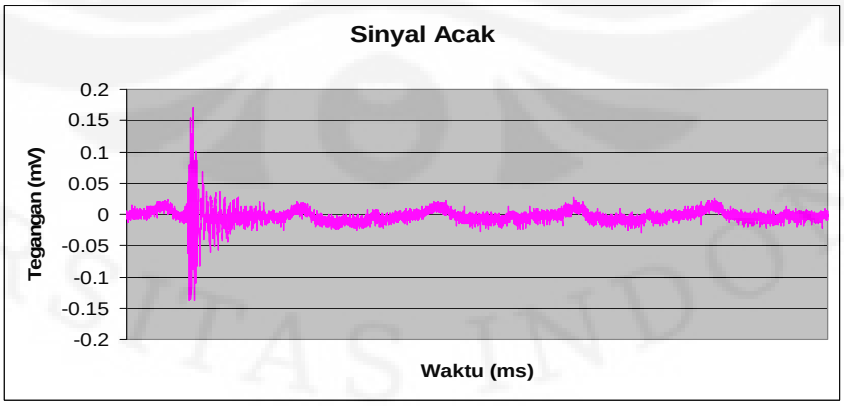
Gambar 4.22 Grafik sinyal hasil efek *chorus*

4.2.2 Hasil Uji Coba Sinyal Acak

Hasil sinyal acak digambarkan dengan grafik waktu terhadap tegangan. Penggambaran dalam domain waktu lebih mewakili hasil efek yang terjadi.

4.2.2.1 Sinyal acak asli

Gambar 4.23 menunjukkan grafik sinyal acak asli.

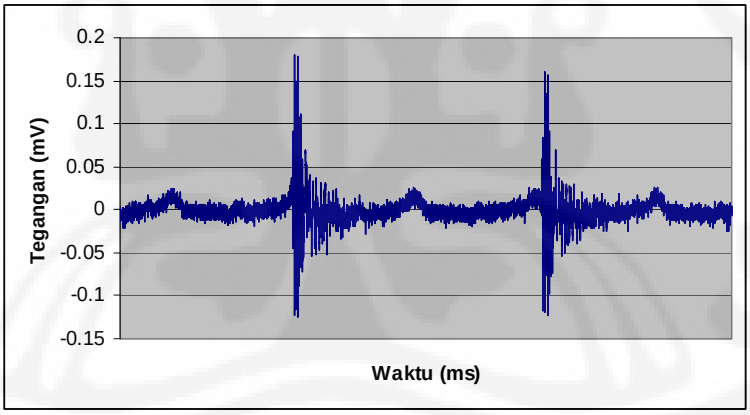


Gambar 4.23 Sinyal acak asli

Sinyal acak tersebut tidak terlihat semulus pada simulasi tanpa alat. Hal itu dikarenakan adanya *noise* pada saat transmisi. Selain sinyal acak yang asli yang digambarkan, tertangkap juga *noise* yang memang terdapat dalam DSK C6713. *noise* tersebut dapat mengurangi performa dari model yang sudah dirancang. Biasanya hal tersebut berpengaruh pada intensitas suara, kejadian yang sering terjadi adalah pengurangan intensitas suara. Namun hal ini ada solusinya yaitu dengan memberikan penguatan dengan menggunakan alat *amplifier*.

4.2.2.2 Sinyal acak hasil efek *reverb*

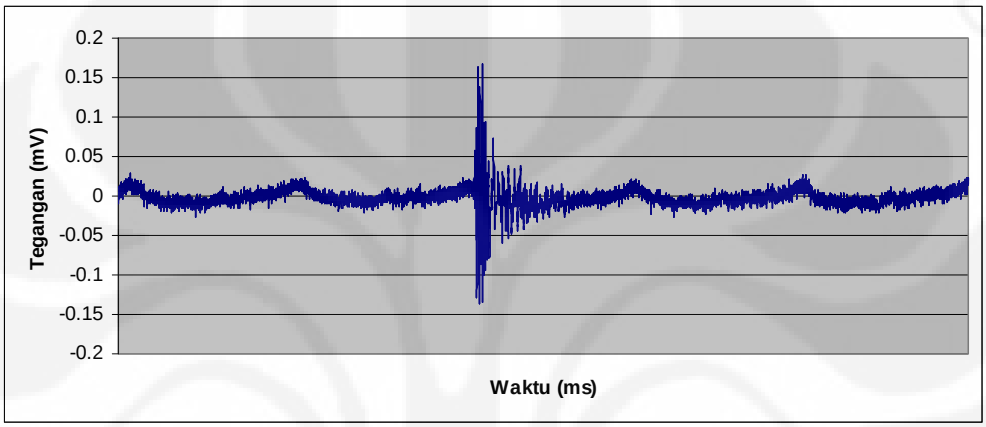
Untuk Gambar 4.24 menunjukkan adanya dua sinyal yang terjadi. Sinyal pertama adalah sinyal asli sedangkan sinyal yang kedua adalah sinyal hasil efek *reverb* dimana mengalami penundaan. Sinyal yang kedua lebih lemah tetapi memiliki bentuk gelombang yang sama dengan sinyal aslinya. Gelombang ini muncul bersamaan pada suatu waktu yang tidak terlalu berjauhan. Tampilan ini hanya akan membandingkan gelombang yang terjadi, tidak menghitung parameter yang ada karena adanya komponen *noise*.



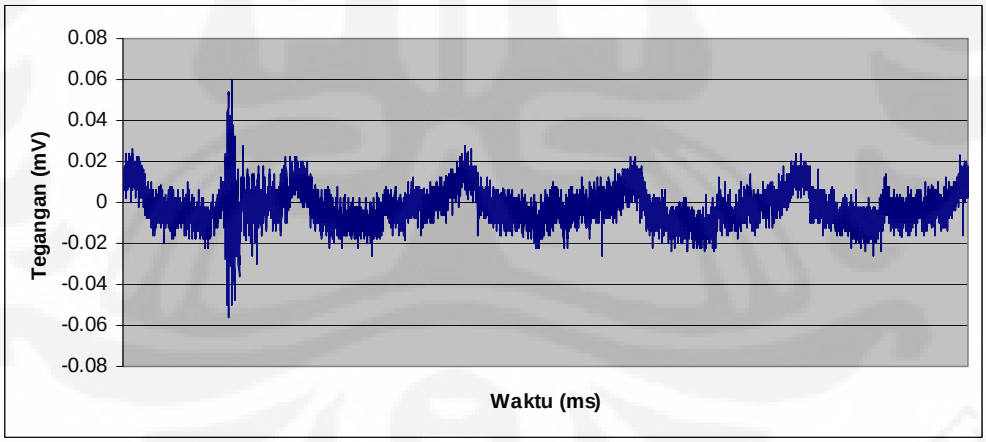
Gambar 4.24 Sinyal acak hasil efek *reverb*

4.2.2.3 Sinyal acak hasil efek *echo*

Untuk Gambar 4.25 menunjukkan sinyal acak asli yang terjadi akibat efek *echo*. Hanya terdapat satu gelombang namun gelombang tersebut akan melemah sehingga lama-lama akan menuju nilai nol seperti pada Gambar 4.26. Hal ini sesuai dengan karakteristik efek *echo* dimana bunyi yang terjadi akan mengalami pemantulan.



Gambar 4.25 Sinyal acak hasil efek *echo* di awal simulasi

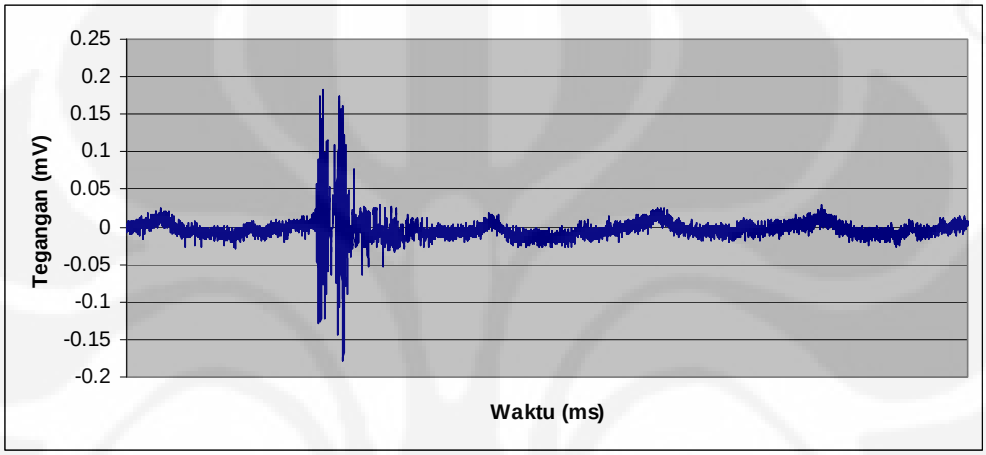


Gambar 4.26 Sinyal acak hasil efek *echo* di akhir simulasi

Besarnya waktu adalah lebih besar dibandingkan dengan efek *reverb*. Hal itu dapat dibuktikan dengan grafik pada Gambar 4.24 dimana pada waktu yang kurang lebih sama terdapat dua gelombang.

4.2.2.4 Sinyal acak hasil efek *chorus*

Pada Gambar 4.27 menunjukkan adanya dua sinyal yang bersamaan muncul. Hal ini menunjukkan adanya suara yang bercampur dengan suara aslinya. Hal ini sama dengan hasil pada Gambar 4.18. Kesulitan dalam mencari nilai yang ada dikarenakan sinyal yang ada sudah mengalami penurunan kualitas akibat *noise*.



Gambar 4.27 Sinyal acak hasil efek *chorus*

BAB V
KESIMPULAN

Dari hasil analisis yang dilakukan maka dapat disimpulkan sebagai berikut:

1. Dari hasil penelitian menunjukkan bahwa perancangan model efek audio dijital dapat diterapkan dengan DSK TMS320C6713 dimana pembuatan model berbasiskan SIMULINK®.
2. Dari hasil pengolahan data sinyal *tone* 1 KHz didapatkan penundaan yang terjadi dalam efek *reverb* adalah 38 ms dan efek *echo* 160 ms. Untuk *chorus* terdapat satu suara campuran dengan pemberian satu LFO.
3. Untuk sinyal acak didapatkan penundaan untuk *reverb* dan *echo* adalah sama seperti *tone*. Hal ini juga berlaku untuk efek *chorus* pada sinyal acak.
4. Terdapat *noise* ketika simulasi dilakukan dengan DSK C6713. Hal ini terjadi pada saat transmisi dilakukan.

DAFTAR ACUAN

[1] Ifeachor, E.C. dan Jervis, B.K. (2001), *Digital Signal Processing*, Prentice Hall Second Edition.

[2] Blonstein, S. dan Katorgi, M. (2004), *eXpressDSP for DUMMIES*, Wiley Publishing, Inc.

[3] Oboril, David. (2000). *Modelling Digital Musical Effects For Signal Processors, Based On Real Effects Manifestation Analysis.* Proceeding COST G-6 DAFX-00. 7-9 Desember 2000

[4] Mahendra, A. (2004). *Petunjuk Penggunaan DSP TMS320C50*.

[5] Gunawan, D. (2007). *Lecture Note Digital Signal Processing*, Fakultas Teknik Universitas Indonesia.

[6] SIMULINK Help

[7] Target Support Package™ TC6 3 User's Guide Hal 2-12 – 2-27. 26 Februari 2008

[8] Hartono, Rio Harlan. (2006). *Analisis Performa Perancangan Real Time Digital Audio Effect Menggunakan DSP Starter Kit TMS320C6713.* Skripsi. Program Sarjana Fakultas Teknik UI, Depok hal 17-18.

[9] Hussain, A. (2005), *DSP Selection Guide*, Texas Instrument, Inc.

[10] "_____". (2003). *Code Composer Studio Help*. Texas Instrument.

[11] Frescura, F (2003). *TMS320C6713 DSK Technical Reference.* Spectrum Digital, Inc.

[12] Hartono, Rio Harlan. (2006). *Analisis Performa Perancangan Real Time Digital Audio Effect Menggunakan DSP Starter Kit TMS320C6713.* Skripsi. Program Sarjana Fakultas Teknik UI, Depok hal 25-27

[13] Harmony Central Effects Explained Modulation. (12 Maret 2008) http://www.harmony-central.com/Effects/Articles/Modulation/

[14] Harmony Central Effects Explained Chorus. (12 Maret 2008) http://www.harmony-central.com/Effects/Articles/Chorus/

[15] Harmony Central Effects Explained Flang. (12 Maret 2008) http://www.harmony-central.com/Effects/Articles/Flanging/

[16] Harmony Central Effects Explained Phaser. (12 Maret 2008) http://www.harmony-central.com/Effects/Articles/Phase Shifter/

[17] Lehman, S. *Reverberation*. (12 Maret 2008). http://www.harmony-central.com/Effects/Articles/Reverb/

[18] Currington, M. (2008), *Audio Effects Frequently Asked Question*.

[19] Harmony Central Effects Explained Delay. (12 Maret 2008) http://www.harmony-central.com/Effects/Articles/delay/

DAFTAR PUSTAKA

Chassaing, Rulph. "*Digital Signal Processing and Apllications with the C6713 and C6416 DSK"*, John Willey &Sons.inc, 2005.

Hartono, Rio Harlan.," Analisis Performa Perancangan *Real Time Digital Audio Effect* Menggunakan DSP *Starter Kit* TMS320C6713." Skripsi. Program Sarjana Fakultas Teknik UI, Depok, 2006.

Keller, Jason dan Joel Koepke." *An Audio Amplifier System with Digital Delay-Based Effects*." 2004

MATLAB Link for Code Composer Studio Development Tools User's Guide.pdf (C) COPYRIGHT 2002 by The MathWorks, Inc

MATLAB Link for Code Composer Studio Development Tools Release Note.pdf (C) COPYRIGHT 2002 by The MathWorks, Inc

MATLAB Target Support Package™ TC6 3 User's Guide

LAMPIRAN

40 DATA PERTAMA PERHITUNGAN

FFT SINYAL TONE 1 KHz

No.	Frekuensi (Hz)	Intensitas (dB)
1	0.00E+00	-26.5531
2	5.00E+01	-27.3125
3	1.00E+02	-28.0688
4	1.50E+02	-32.075
5	2.00E+02	-36.0781
6	2.50E+02	-42.4875
7	3.00E+02	-48.8938
8	3.50E+02	-48.0563
9	4.00E+02	-47.2188
10	4.50E+02	-47.7031
11	5.00E+02	-48.1875
12	5.50E+02	-48.4188
13	6.00E+02	-48.6469
14	6.50E+02	-43.3969
15	7.00E+02	-38.1438
16	7.50E+02	-29.1875
17	8.00E+02	-20.2313
18	8.50E+02	-14.5563
19	9.00E+02	-8.88125
20	9.50E+02	-7.30938
21	1.00E+03	-5.7375
22	1.05E+03	-7.30938
23	1.10E+03	-8.88125
24	1.15E+03	-14.5344
25	1.20E+03	-20.1875
26	1.25E+03	-29.1656
27	1.30E+03	-38.1438
28	1.35E+03	-44.1031
29	1.40E+03	-50.0594
30	1.45E+03	-50.0594
31	1.50E+03	-50.0594
32	1.55E+03	-51.1156
33	1.60E+03	-52.1688
34	1.65E+03	-50.1781
35	1.70E+03	-48.1875
36	1.75E+03	-48.4188
37	1.80E+03	-48.6469
38	1.85E+03	-49.3531
39	1.90E+03	-50.0594
40	1.95E+03	-51.1156

FFT SINYAL TONE REVERB

No.	Frekuensi (Hz)	Intensitas (dB)
1	0.00E+00	-29.3813
2	5.00E+01	-30.65
3	1.00E+02	-31.9156
4	1.50E+02	-35.875
5	2.00E+02	-39.8313
6	2.50E+02	-44.2406
7	3.00E+02	-48.6469
8	3.50E+02	-47.2656
9	4.00E+02	-45.8844
10	4.50E+02	-47.9719
11	5.00E+02	-50.0594
12	5.50E+02	-48.3188
13	6.00E+02	-46.575
14	6.50E+02	-42.7344
15	7.00E+02	-38.8938
16	7.50E+02	-29.0844
17	8.00E+02	-19.2719
18	8.50E+02	-13.775
19	9.00E+02	-8.275
20	9.50E+02	-6.65
21	1.00E+03	-5.02188
22	1.05E+03	-6.65938
23	1.10E+03	-8.29688
24	1.15E+03	-13.8125
25	1.20E+03	-19.325
26	1.25E+03	-29.4563
27	1.30E+03	-39.5844
28	1.35E+03	-44.2406
29	1.40E+03	-48.8938
30	1.45E+03	-50.5313
31	1.50E+03	-52.1688
32	1.55E+03	-51.1156
33	1.60E+03	-50.0594
34	1.65E+03	-49.3531
35	1.70E+03	-48.6469
36	1.75E+03	-49.3531
37	1.80E+03	-50.0594
38	1.85E+03	-51.1156
39	1.90E+03	-52.1688
40	1.95E+03	-49.3719

FFT SINYAL TONE ECHO

No.	Frekuensi (Hz)	Intensitas (dB)
1	0.00E+00	-41.6438
2	1.00E+02	-39.5844
3	3.00E+02	-37.525
4	4.00E+02	-28.6125
5	5.00E+02	-19.6969
6	6.00E+02	-14.3469
7	8.00E+02	-8.99688
8	9.00E+02	-7.325
9	1.00E+03	-5.65
10	1.10E+03	-7.325
11	1.30E+03	-8.99688
12	1.40E+03	-14.4125
13	1.50E+03	-19.825
14	1.60E+03	-31.1656
15	1.80E+03	-42.5063
16	1.90E+03	-49.8625
17	2.00E+03	-57.2188
18	2.10E+03	-54.6938
19	2.30E+03	-52.1688
20	2.40E+03	-54.125
21	2.50E+03	-56.0813
22	2.60E+03	-58.1563
23	2.80E+03	-60.2313
24	2.90E+03	-56.2
25	3.00E+03	-52.1688
26	3.10E+03	-58.1906
27	3.30E+03	-64.2094
28	3.40E+03	-60.1469
29	3.50E+03	-56.0813
30	3.60E+03	-56.0813
31	3.80E+03	-56.0813
32	3.90E+03	-58.1563
33	4.00E+03	-60.2313
34	4.10E+03	-58.1563
35	4.30E+03	-56.0813
36	4.40E+03	-56.0813
37	4.50E+03	-56.0813
38	4.60E+03	-58.1563
39	4.80E+03	-60.2313
40	4.90E+03	-56.7375

FFT SINYAL TONE CHORUS

No.	Frekuensi (Hz)	Intensitas (dB)
1	0.00E+00	-31.0969
2	5.00E+01	-32.6313
3	1.00E+02	-34.1656
4	1.50E+02	-37.25
5	2.00E+02	-40.3344
6	2.50E+02	-42.1469
7	3.00E+02	-43.9563
8	3.50E+02	-49.5688
9	4.00E+02	-55.1781
10	4.50E+02	-48.675
11	5.00E+02	-42.1688
12	5.50E+02	-51.6844
13	6.00E+02	-61.1969
14	6.50E+02	-48.5531
15	7.00E+02	-35.9094
16	7.50E+02	-27.7688
17	8.00E+02	-19.625
18	8.50E+02	-14.0125
19	9.00E+02	-8.4
20	9.50E+02	-6.8125
21	1.00E+03	-5.225
22	1.05E+03	-6.8125
23	1.10E+03	-8.39687
24	1.15E+03	-14.0438
25	1.20E+03	-19.6875
26	1.25E+03	-28.8531
27	1.30E+03	-38.0188
28	1.35E+03	-45.0938
29	1.40E+03	-52.1688
30	1.45E+03	-51.1156
31	1.50E+03	-50.0594
32	1.55E+03	-49.3531
33	1.60E+03	-48.6469
34	1.65E+03	-49.3531
35	1.70E+03	-50.0594
36	1.75E+03	-50.0594
37	1.80E+03	-50.0594
38	1.85E+03	-48.3188
39	1.90E+03	-46.575
40	1.95E+03	-50.8781

FFT SINYAL ACAK REVERB

No.	Frekuensi (Hz)	Intensitas (dB)
1	0.00E+00	-58.9188
2	5.00E+00	-61.7938
3	1.00E+01	-64.6656
4	1.50E+01	-60.9438
5	2.00E+01	-57.2188
6	2.50E+01	-60.5875
7	3.00E+01	-63.9563
8	3.50E+01	-58.4156
9	4.00E+01	-52.8719
10	4.50E+01	-49.35
11	5.00E+01	-45.825
12	5.50E+01	-48.8875
13	6.00E+01	-51.95
14	6.50E+01	-55.2406
15	7.00E+01	-58.5281
16	7.50E+01	-69.8625
17	8.00E+01	-81.1969
18	8.50E+01	-67.5313
19	9.00E+01	-53.8656
20	9.50E+01	-53.8375
21	1.00E+02	-53.8094
22	1.05E+02	-53.5719
23	1.10E+02	-53.3313
24	1.15E+02	-58.7719
25	1.20E+02	-64.2094
26	1.25E+02	-63.8875
27	1.30E+02	-63.5656
28	1.35E+02	-63.4563
29	1.40E+02	-63.3438
30	1.45E+02	-59.2563
31	1.50E+02	-55.1656
32	1.55E+02	-58.0563
33	1.60E+02	-60.9469
34	1.65E+02	-64.5688
35	1.70E+02	-68.1875
36	1.75E+02	-65.7656
37	1.80E+02	-63.3438
38	1.85E+02	-61.8594
39	1.90E+02	-60.3719
40	1.95E+02	-60.6594

FFT SINYAL ACAK CHORUS

No.	Frekuensi (Hz)	Intensitas (dB)
1	0.00E+00	-45.9313
2	1.00E+01	-45.2375
3	3.00E+01	-44.5438
4	4.00E+01	-45.6969
5	5.00E+01	-46.8469
6	6.00E+01	-46.1125
7	8.00E+01	-45.375
8	9.00E+01	-46.8594
9	1.00E+02	-48.3438
10	1.10E+02	-48.7188
11	1.30E+02	-49.0938
12	1.40E+02	-52.1313
13	1.50E+02	-55.1656
14	1.60E+02	-54.0406
15	1.80E+02	-52.9125
16	1.90E+02	-54.8844
17	2.00E+02	-56.8531
18	2.10E+02	-56.5406
19	2.30E+02	-56.2281
20	2.40E+02	-62.2094
21	2.50E+02	-68.1875
22	2.60E+02	-63.4188
23	2.80E+02	-58.6469
24	2.90E+02	-60.1531
25	3.00E+02	-61.6563
26	3.10E+02	-61.1063
27	3.30E+02	-60.5531
28	3.40E+02	-63.8875
29	3.50E+02	-67.2188
30	3.60E+02	-72.7031
31	3.80E+02	-78.1875
32	3.90E+02	-70.5313
33	4.00E+02	-62.8719
34	4.10E+02	-69.025
35	4.30E+02	-75.1781
36	4.40E+02	-67.925
37	4.50E+02	-60.6688
38	4.60E+02	-65.9344
39	4.80E+02	-71.1969
40	4.90E+02	-67.9313