

# IMPLEMENTASI *REAL TIME EFFECT* PADA GITAR BERBASIS WAKTU TUNDA / *DELAY* MENGUNAKAN DSK TMS320C6713

Boristan Siahaan<sup>1</sup>, Miftahul Huda<sup>2</sup>

<sup>1</sup>Mahasiswa Politeknik Elektronika Negeri Surabaya, Jurusan Teknik Telekomunikasi

<sup>2</sup>Politeknik Elektronika Negeri Surabaya Institut Teknologi Sepuluh Noverber

Kampus ITS, Surabaya 60111

email: [fransiscus\\_boristan@yahoo.com](mailto:fransiscus_boristan@yahoo.com)

## Abstrak

Efek audio digital juga banyak digunakan pada pengolahan sinyal audio digital untuk keperluan industri musik, salah satunya yaitu efek pada instrumen gitar elektrik. Banyak pemain gitar elektrik menggunakan beberapa bentuk efek untuk meningkatkan kualitas suara gitar, baik secara analog atau Digital Signal Processing (DSP).

Tugas akhir ini bertujuan sebagai penerapan dari mata kuliah yang di realisasikan dalam bentuk alat. Proyek ini bertujuan untuk mengembangkan teknologi dibidang pengolahan sinyal yang akan difokuskan pada implementasi efek delay antara gitar elektrik sebagai sumber dan TMS320C6713 sebagai device yang mengolah efek delay tersebut. Perancangan efek delay menggunakan DSP TMS320C6713 didasari dengan adanya waktu penundaan dari sinyal input, dalam hal ini sinyal audio sehingga tidak langsung dikeluarkan pada device output. Perangkat lunak yang digunakan adalah Code Composer Studio (CCS), yang merupakan perangkat lunak yang dibutuhkan untuk membuat program.

Kata kunci : *Efek gitar, DSK TMS320C6713, efek delay, real time*

## I. PENDAHULUAN

Audio merupakan sinyal yang menghasilkan suara yang mampu diindra oleh pendengaran kita. Perkembangan pengolahan audio tidak hanya fokus pada pengolahan secara analog, namun pengembangan yang mengarah pada pengolahan audio secara digital sudah mulai banyak yang mencoba.

Perancangan efek audio digital menggunakan DSP TMS320C6713 didasari dengan adanya waktu penundaan dari sinyal input, dalam hal ini sinyal audio sehingga tidak langsung dikeluarkan pada device output.

Pada proyek akhir kali ini, akan dibuat pengembangan dari proyek yang pernah dibuat dan diteliti sebelumnya, yaitu kearah pembuatan efek gitar digital yang terdiri dari efek gitar digital yang berbasis waktu tunda /time delay dengan menggunakan DSK TMS320C6713.

## II. TEORI PENUNJANG DAN PERENCANAAN SISTEM

### 2.1 Digitak Starter Kit (DSK)

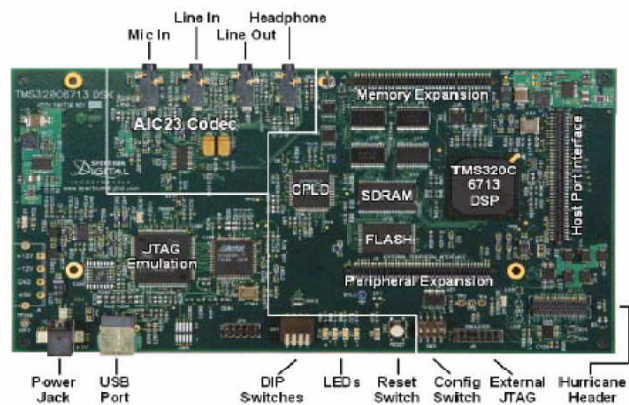
Pemroses sinyal digital seperti TMS 320C6713 adalah mikroprosesor dengan tipe khusus pada arsitekturnya dan sebuah instruksi yang cocok untuk pemrosesan sinyal. Pemrosesan sinyal digital digunakan untuk aplikasi pada range

yang lebar dan biasanya digunakan pada komunikasi kontrol hingga image processing.

Paket DSK sangat bertenaga dengan hardware dan software yang membantu untuk pemrosesan sinyal secara real time. DSK memiliki ukuran 5 x 8 in meliputi C6713 floating point digital signal processor dan 32 bit stereo codec TLV 320AIC23. DSK meliputi 16 MB (MegaByte) pada Synchronous Dynamic Random (SDRAM) dan 256 KB (KiloBytes) pada flash memory. Terdapat empat konektor pada DSK untuk input dan output yaitu MIC IN untuk input dari microphone, LINE IN untuk input dari function generator, LINE OUT untuk output, dan HEADPHONE untuk output pada headphone.

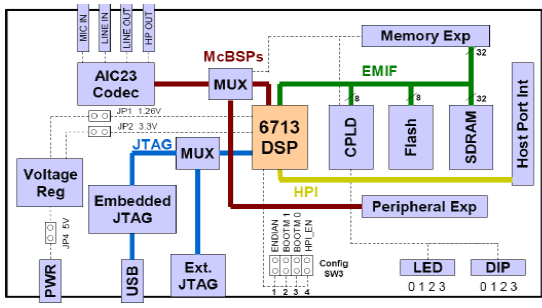
### 2.2 TMS 320C6713

TMS 320C6713 pada DSK adalah floating point processor yang didasari pada arsitektur VLIW (Very-Long-Instruction-Word). Memori internal meliputi 2 level penyimpanan daya, yang terdiri dari 4 KB pada level penyimpanan program (L1P), 4 KB pada level 1 penyimpanan data (L1D), dan 256 pada level 2 pembagian memori antara program dan data. Selain itu, juga mempunyai sebuah interface untuk memerintah synchronous dan memori (SDRAM dan SBSRAM) dan asynchronous memori (SPRAM dan EPROM).



Gambar 1. DSK

Selain itu, terdapat 2 McBSP, 2 timer, sebuah host interface dan 32 bit EMIF. Semua itu membutuhkan 32 bit program address bus, 256 bit program data bus. Untuk mengakomodasi 32 bit instruksi, 2 data address bus sebanyak 32 bit dan 2 data bus sebanyak 64 bit sehingga total memori adalah 4 GB yang meliputi 4 eksternal memori yaitu CE0, CE1, CE2, CE3.



Gambar 2. Blok Diagram pada TMS 320C6713

**2.3 Code Composer Studio (CCS)**

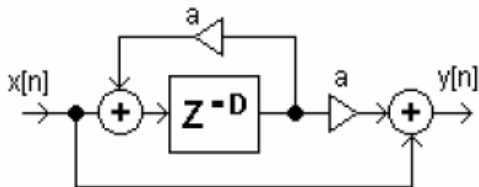
CCS (Code Composer Studio) adalah penyedia sebuah Integrated Development Enviroment (IDE) untuk menggabungkan peralatan pada software. Peralatan tersebut antara lain C compiler, sebuah assembler, dan sebuah linker.

Di dalam Code Composer Studio (CCS) terdapat beberapa file pendukung antara lain adalah

1. C6713dskinit.c: mengandung fungsi – fungsi untuk inialisasi ke DSK, codec, serial port, serta untuk input dan output
2. C6713dskinit.h: adalah sebuah file header dengan fungsi prototypes. Biasanya digunakan untuk pemilihan input seperti memilih input dari microphone sebagai pengganti dari line input, input gain, dan sebagainya
3. C6713dsk.cmd: adalah sebuah sample linker command file. File ini dapat dirubah saat menggunakan external memori.
4. Vectors\_intr.asm: adalah sebuah vector file yang telah dimodifikasi yang terdapat di dalam CCS untuk mengatur interrupt. Terdapat 12 interrupt dimana INT4 hingga INT 15 dapat digunakan. Dan yang biasa menggunakan vector ini adalah INT 11.
5. Vectors\_poll.asm: file vector untuk program yang menggunakan polling.
6. rts6700.lib, dsk6713bsl.lib, csl6713.lib: run-time, board, dan chip yang mendukung file library.

**2.4 Efek Echo**

Pada efek echo, sinyal masukkan dijumlahkan dengan sinyal tertunda yang telah dilemahkan, kemudian dijumlahkan dengan sinyal masukan untuk dimasukkan kembali ke dalam blok delay (Z-D). Efek chorus terjadi jika sinyal asli dicampur dengan salinan yang tertunda (lebih dari 50 ms) dari sinyal asli.



Gambar 3 Blok Diagram Efek Echo

Secara matematis efek echo diperoleh dari persamaan:

$$y[n] = x[n] + ax[n-k] \dots\dots(1)$$

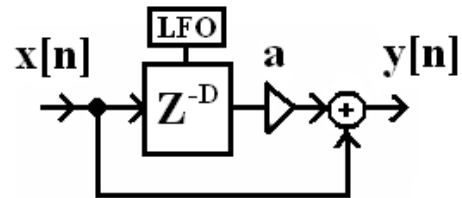
Dimana  $x[n]$  adalah sinyal masukan,  $a$  adalah attenuator atau factor pelemah,  $x[n-k]$  adalah sinyal tertunda atau Z-D, dan  $y[n]$  adalah sinyal keluaran.

Program :

```
int efek_echo(int input)
{
    if (pointer < size){
        pointer++;
    } else {
        pointer=0;
    }
    output = buffer[pointer]*0.5+input*0.5;
    buffer[pointer]=buffer[pointer]*0.5 + input;
    return(output);
}
```

**2.5 Efek Chorus**

Efek chorus mengambil input audio dari satu instrumen dan membuatnya terdengar seperti kelipatan instrumen yang sama sedang dimainkan pada waktu yang sama. Efek chorus terjadi jika sinyal asli dicampur dengan salinan yang tertunda (antara 10-25 ms) dari sinyal asli.



Gambar 4 Blok Diagram Efek Chorus

$$Y[n] = x[n] + ax[n-\Delta[n]] \dots\dots(2)$$

Dimana :

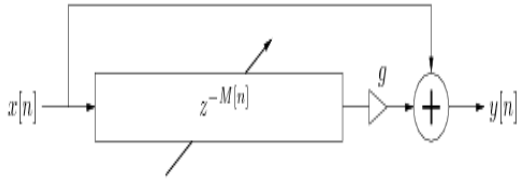
- $\Delta[n]$  = sebuah fungsi diskrit dari delay periodik.
- $a$  = attenuator (diuji coba  $a = 0.5$ )
- $x[n]$  = sinyal masukan dari gitar
- $y[n]$  = sinyal keluaran dari gitar

Program :

```
int efek_chorus(int input)
{
    LFO();
    if (pointer < size){
        pointer++;
    } else {
        pointer=0;
    }
    output = buffer[pointer]*0.5+input;
    buffer[pointer]=input + buffer[pointer]*0.5;
    return(output);
}
```

**2.6 Efek Flanger**

Struktur efek ini dasarnya temannya chorus. Salah satu perbedaan adalah menjumlah delay yang digunakan. Waktu delay pada chorus lebih besar dari flanger, umumnya diantara 20 ms dan 30 ms. (waktu flanger antara 1 ms sampai 10 ms). Flanger juga dibedakan dengan chorus tidak menghasilkan umpan balik.



**Gambar 5** Blok Diagram Efek Flanger

$$Y[n] = ax[n] + ax[n-\Delta[n]] \dots\dots(4)$$

Dimana :

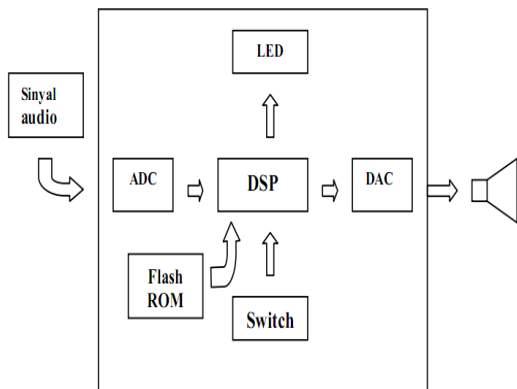
- $\Delta[n]$  = sebuah fungsi diskrit dari delay periodik.
- a = attenuator (diuji coba a = 0.5)
- x[n]= sinyal masukan dari gitar
- y[n]= sinyal keluaran dari gitar

Program :

```
int efek_flanger(int input)
{
    LFO();
    if (pointer < size){
        pointer++;
    } else {
        pointer=0;
    }
    output = buffer[pointer]*0.5 + input*0.3;
    buffer[pointer]=input*0.5;
    buffer[pointer]*0.5;
    return(output);
}
```

**2.7 Perencanaan Sistem**

Pada proyek akhir ini, dibuat efek gitar berbasis delay yaitu echo, chorus dan flanger. Untuk meringkas semua operasi sinyal tersebut, maka digunakan dip switch untuk memilih efek yang ingin digunakan. Blok diagram yang diterapkan pada proyek akhir ini seperti gambar berikut:



**Gambar 6.** Blok Diagram Sistem

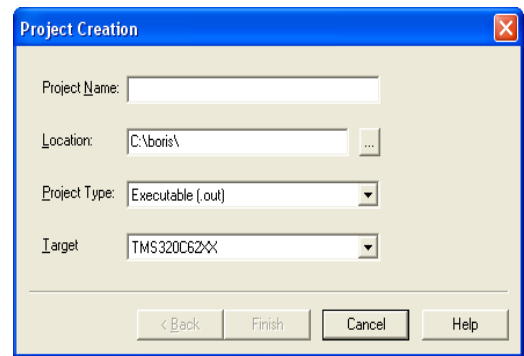
**III. HASIL PENGUJIAN PROYEK AKHIR**

**3.1 Install Code Composer Studio V3 dan Diagnostik DSK.**

Penginstalan Code Composer Studio 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 memilih Debug kemudian pilih Connect
3. Buatlah project baru dengan cara buka Project kemudian pilih New. Maka akan tampil seperti gambar 7



**Gambar 7.** New Project

4. Tambahkan semua file- file pendukung seperti

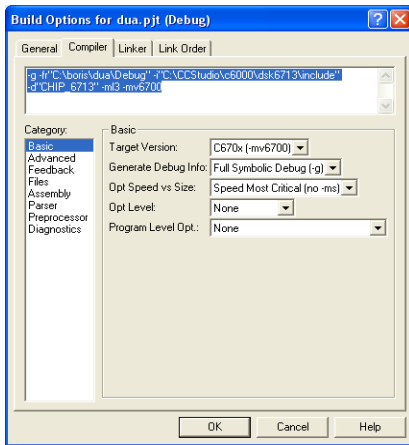
- C6713dskinit.c
- Vector\_intr.asm
- C6713dsk.cmd
- dsk6713\_aic23.h
- file.c

Kelima file-file di atas berada di dalam folder C:\CCStudio\MyProjects dan pilih All Files. Setelah menemukan seluruhnya, maka klik dua kali atau klik open. Selain kelima file di atas, tambahkan pula

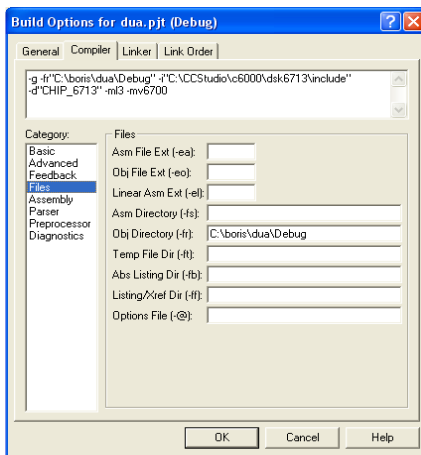
- rts6700.lib,file tersebut berada di dalam folder C:\CCStudio\c6000\cgtools\lib
- dsk6713bsl.lib,file tersebut berada di dalam folder C:\CCStudio\c6000\dsk6713\lib
- csl6713.lib,file tersebut berada di dalam folder C:\CCStudio\c6000\bios\lib

5. Setelah semua file di atas telah dimasukkan ke dalam project,maka lakukan langkah-langkah ketergantungan dengan cara pilih **Project** kemudian **Scan All Files Dependenciess**. Maka secara otomatis akan terhubung file-file pendukung yang diperlukan pada proyek yang telah dibuat.

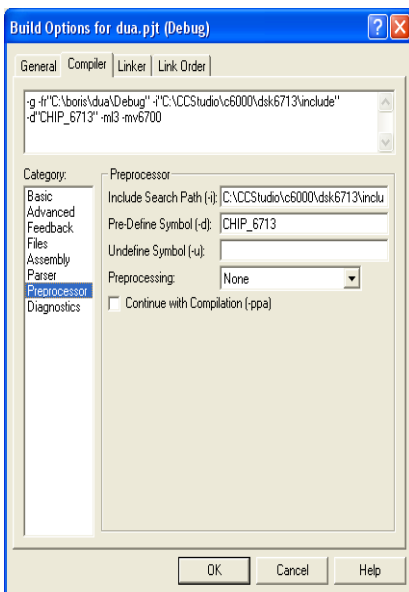
6. Kemudian atur BUILD OPTION seperti gambar 8 untuk pengaturan Compiler Category Basic. Untuk Compler Category Files seperti gambar 9 dan category preprocessor, set seperti gambar 10.



Gambar 8. Category Basic

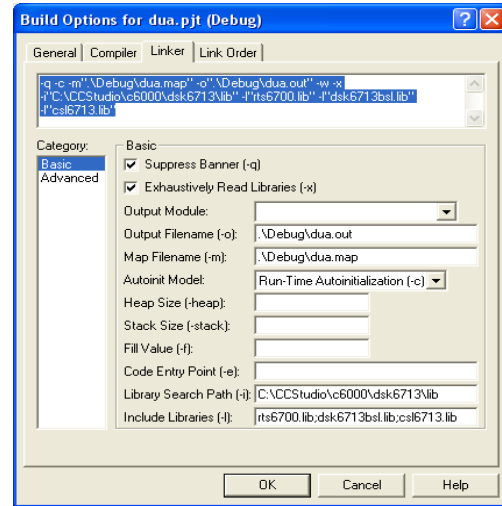


Gambar 9. Category Files



Gambar 10. Category Processor

7. Untuk Linker, set seperti gambar 11



Gambar 11. Linker

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 dan lihat sinyal output melalui oscilloscope.
12. Untuk membangkitkan sinyal kotak maka rubahlah bentuk sinyal pada function generator

### 3.3 Analisa Hasil Pengujian

#### 3.3.1 Penggabungan Program Efek Gitar Berbasis Delay

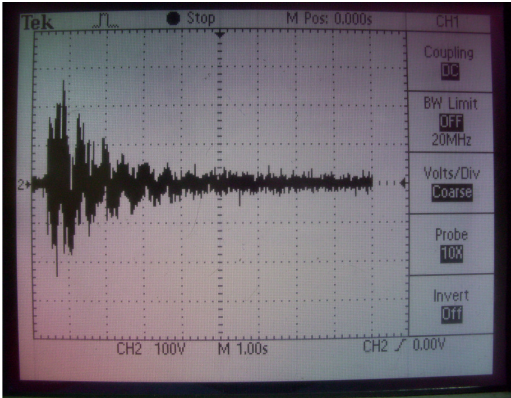
Penggabungan efek gitar berbasis delay pada proyek akhir ini ditujukan agar pada saat menjalankan masing-masing efek tidak perlu menghentikan program, sehingga saat hendak mengganti efek satu ke efek yang lain kita hanya melakukan perubahan posisi dari swich pada DSK saja.

#### 3.3.2 Pengujian Pemanggilan Efek Menggunakan Dip Switch

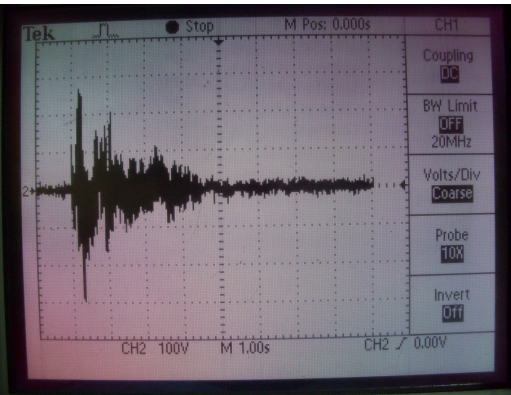
Pengujian hasil yang didapat adalah dengan membandingkan sinyal output yang dihasilkan oleh komparator MATLAB dan sinyal output yang dihasilkan oleh C6713 DSK yang ditampilkan oleh oscilloscope. Pada penguatan sinyal, yang akan dikuatkan adalah besarnya amplitude.

### 3.3.3 Pengujian Efek Echo

Untuk melakukan pengukuran terhadap waktu eksekusi tiap efek gitar digital maka digunakan alat ukur sinyal yaitu oscilloscope. Dengan menggunakan alat ini dapat dilihat bentuk sinyal waktunya, sehingga dapat dihitung waktu eksekusi dari masing-masing efek gitar digital.



(a)



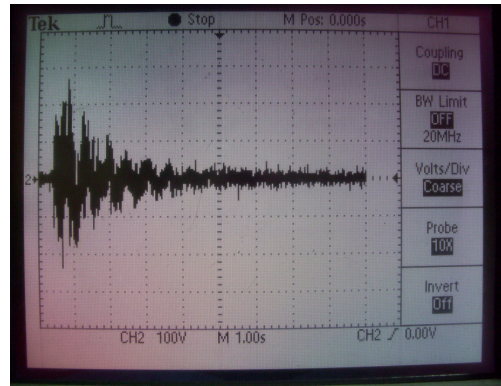
(b)

**Gambar 12** (a) Sinyal input (b) Sinyal efek echo hasil pengujian dengan nilai buffer 8000 dan attenuator = 0,5 s

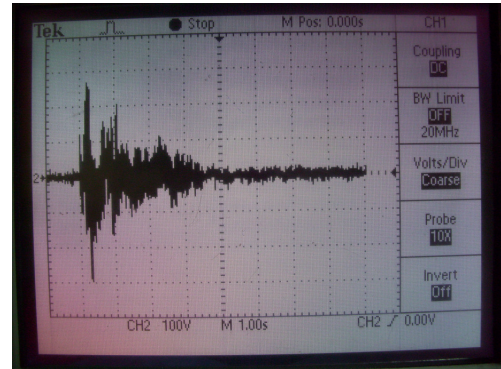
Suara yang dihasilkan efek echo adalah perulangan suara yang cukup lama hingga suara melemah, surara menyerupai gema.

### 3.4.2.2 Pengujian Efek Chorus

Dan pada umumnya beberapa periodik bentuk gelombang yang di gunakan pada efek ini adalah gelombang sinus dan gelombang segitiga. Selama proses pengujian dan pengambilan data pada masing- masing efek gitar digital, input berupa sinyal tunggal yang berasal dari petikan dawai no-4 yang bernada dasar D untuk kemudian diolah menggunakan program efek flanger yang dihasilkan mengeluarkan bentuk sinyal output seperti berikut.



(a)



(b)

**Gambar 13** (a) Sinyal input (b) Sinyal efek chorus hasil pengujian dengan nilai buffer 480 dan attenuator = 0,5 s

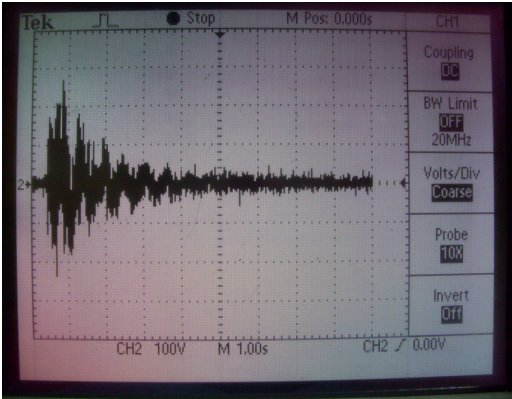
Pada teorinya efek chorus dapat terjadi dengan menjumlahkan sebuah echo sinyal masukan dan kemudian merubah nilai delay echo antara maksimum dan minimum. Efek chorus dapat terjadi dengan menjumlahkan sebuah echo sinyal masukan dan kemudian merubah nilai delay echo antara maksimum dan minimum. Waktu delay pada chorus lebih besar dari flanger, umumnya diantara 20 ms dan 30 ms. (waktu flanger antara 1 ms sampai 10 ms)

Suara yang dihasilkan menjadi tebal dan seperti terdengar banyak suara yang dihasilkan.

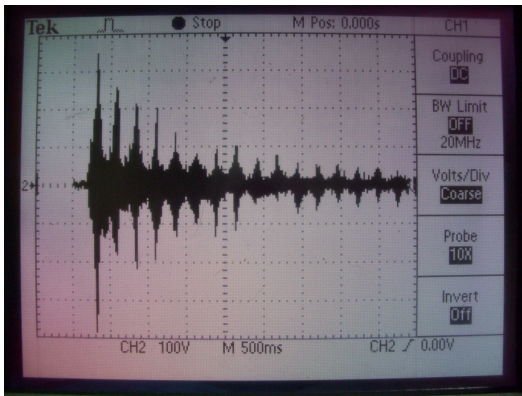
### 3.4.2.3 Pengujian Efek Flanger

Pada efek flanger, sinyal masukan dijumlahkan dengan sinyal tertunda yang telah dilemahkan kemudian sinyal tertunda yang telah dilemahkan dijumlahkan dengan sinyal masukan untuk dimasukkan kembali ke blok delay ( Z-D ) dan LFO berfungsi untuk variasi waktu tunda.

Selama proses pengujian dan pengambilan data pada masing- masing efek gitar digital, input berupa sinyal tunggal yang berasal dari petikan dawai no-4 yang bernada dasar D untuk kemudian diolah menggunakan program efek flanger yang dihasilkan mengeluarkan bentuk sinyal output seperti berikut.



(a)



(b)

**Gambar 14** (a) Sinyal input (b) Sinyal efek flanger hasil pengujian dengan nilai buffer 160 dan attenuator = 0,5 s

Suara yang dihasilkan adalah suara input yang disertai suara tambahan yang bergerak naik turun.

#### IV. KESIMPULAN

Dari hasil pengujian dan analisa pada bab sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut :

1. Dari efek delay yang telah dibuat kemudian diukur waktu eksekusinya, maka ternyata semuanya menunjukkan sistem efek gitar digital yang real time karena ukuran buffer maksimal 8000 dan frekuensi sampling 16KHz.
2. Efek Chorus dapat dikontrol dengan mengubah frekuensi, amplitudo dan periodik. Dengan frekuensi 33.3 Hz, amplitudo 480 dan periodik 30 ms, maka dapat dihasilkan efek chorus yang sesuai dengan yang diharapkan.
3. Pada efek flanger pembuatannya hampir sama dengan efek chorus, yang membedakannya terletak pada feedback. Dengan frekuensi 100 Hz, amplitudo 160 dan periodik 10 ms, maka dapat dihasilkan efek flanger yang sesuai dengan karakteristiknya.
4. Jumlah buffer mempengaruhi waktu delay yang dihasilkan semakin besar jumlah buffer yang digunakan maka waktu delay yang akan dihasilkan akan semakin besar/ lama.
5. LFO (Low Frequency Osillator) berfungsi untuk variasi waktu tunda.

6. Setelah dilakukan pengujian terhadap sinyal tiap efek gitar digital dengan function generator dan oscilloscope, ternyata hasil keluaran (output) dari masing-masing efek delay echo, flanger dan chorus menggunakan gitar listrik menghasilkan output sinyal yang bervariasi.

#### V. DAFTAR PUSTAKA

- [1] ChassaigRulph. "Digital Signal Processing and Application with the C6713 and C6416 DSK"
- [2] Tri Budi Santoso. 2009. "Modul Ajar Operasi Dasar Sinyal"
- [3] Amir, 2003 "Rancang Bangun Efek Audio Digital menggunakan DSP TMS320C5402 (Perangkat Lunak)", Proyek Akhir Politeknik Elektronika Negeri Surabaya.
- [4] Datasheet DAFX.pdf
- [5] DSP Development Systems.2003. "TMS320C6713 DSK Techocal Reference".