# BAB I

# PENDAHULUAN

## 1.1 Latar Belakang

Dalam perkembangan awal elektronika, filter analog menjadi pilihan karena relatif murah dan mudah dalam perancanganya. Akan tetapi setelah ditemukan piranti digital dengan kemampuan komputasi yang cepat, implementasi filter digital sangat digemari dan sejak itu hingga kini telah banyak menggantikan peran filter analog. Sampai saat ini filter digital masih menjadi objek penelitian yang terus berkembang dan diminati banyak orang.

Filter digital adalah suatu prosedur matematika atau algoritma yang mengolah sinyal masukan digital dan menghasilkan isyarat keluaran digital yang memiliki sifat tertentu sesuai dengan tujuan filter. Filter digital dapat dibagi menjadi dua yaitu filter digital IIR (infinite impulse response) dan FIR (finite impulse response). Pembagian ini berdasarkan pada tanggapan impuls filter tersebut yaitu FIR memiliki tanggapan impuls yang panjangnya terbatas sedangkan IIR tidak terbatas.

Filter digital FIR didesain dengan teknik yang serupa dengan teknik yang digunakan pada perancangan filter analog. Pada setiap metode perancangan teknik digital FIR selalu diawali dengan perancangan filter analog terlebih dahulu dalam kawasan frekwensi analog. Kemudian dirancang filter digital FIR sebagai penandaan dari filter analog. Diharapkan karakteristik filter digital yang dibentuk akan sama atau mendekati sama dengan filter analog.

Dari uraian diatas, maka dalam judul tugas akhir ini penulis mengambil judul tentang "SIMULASI FILTER DIGITAL UNTUK FREKUENSI SUARA 20-20KHZ MENGGUNAKAN DSP DAN MATLAB V. 7"

## 1.2 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah :

- Memahami perbandingan antara jenis filter digital FIR.
- Mempelajari simulasi filter digital untuk frekuensi suara 20-20000Hz.
- Mengetahui hasil pengujian keluaran suara yang telah diredam noise-nya dengan menggunakan filter digital FIR.

## 1.3 Rumusan Masalah

Bagaimana merancang simulasi yang dapat menghasilkan keluaran filter digital FIR, sehingga dapat menampilkan sinyal keluaran digital dari analog, Khusus-nya untuk frekuensi suara 20-20000Hz.

## 1.4 Batasan Masalah

Ruang lingkup pembahasan masalah pada tugas akhir ini adalah sebagai berikut :

- Filter digital yang dirancang adalah FIR
- Jenis filter FIR yang dirancang adalah jenis type *hamming, Kaiser, Fir2, Blackman, Firls, Fircls1.*

- Sinyal masukan yang akan difilter adalah suara dengan frekuensi 20-20000Hz.
- Filter digital FIR di rancang menggunakan program matlab

**1.5 Metoda Penelitian**

Dalam Menyelesaikan Tugas Akhir ini penulis melakukan langkah-langkah :

- Mempelajari dan mengumpulkan data dari buku serta artikel-artikel yang berhubungan dengan Tugas Akhir ini.
- Melakukan Perancangan dan membuat simulasi filter digital FIR untuk suara 20-20000Hz
- Melakukan pengujian alat serta simulasi untuk menetukan kinerja dari tiap rangkaian serta melakukan pengukuran di laboratorium.
- Melakukan konsultasi dengan pembimbing untuk mengetahui dan mengatasi masalah serta hambatan yang timbul dalam pembuatan Tugas Akhir Ini.

**1.6 Sistematika Penulisan**

Sistematika penulisan laporan Tugas Akhir ini meliputi :

BAB I : PENDAHULUAN

Membahas latar belakang, tujuan, rumusan masalah, batasan masalah, ,metodologi penelitian dan sistematika penulisan.

BAB II : DASAR TEORI

Membahas teori dasar yang berhubungan dengan pembuatan tugas akhir ini.

BAB III : PERANCANGAN ALAT DAN SIMULASI

Membahas cara kerja rangkaian, langkah-langkah perancangan dan pembuatan perangkat ini.

BAB IV : PENGUJIAN ALAT

Berisi hasil pengukuran serta pengujian rangkaian dan analisa.

BAB V : PENUTUP

Berisi mengenai kesimpulan dari hasil pembuatan tugas akhir ini serta saran yang ditujukan untuk pengembangan alat lebih lanjut.

# BAB II

# DASAR TEORI

## 2.1.Filter Digital

Filter digital adalah suatu sistem yang berfungsi untuk menyaring frekuensi, serta memodifikasi sefekrum frekuensi disuatu sinyal sehingga diperoleh tujuan yang diinginkan.

Sefekrum frekuensi ini adalah suatu gerafik dimana dalam grafik itu menggambarkan bentuk-bentuk suatu sinyal yang merupakan kumpulan sinyal-sinyal sineusoida dengan amplitudo dan frekuensi yang berbeda-beda yang membentuk suatu sinyal.

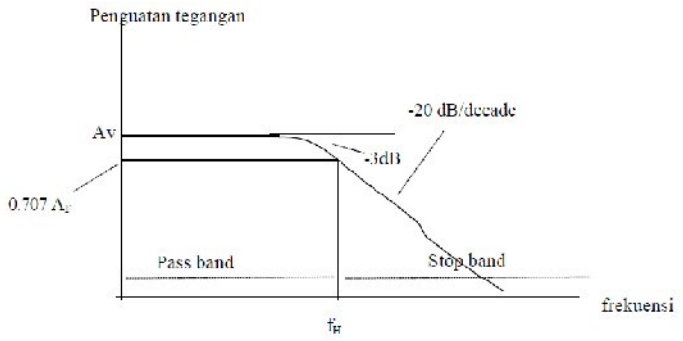
Ada beberapa macam klasifikasi filter yaitu :

1. Berdasarkan sinyal yang difilter yaitu, *Filter Analaog* dan *Filter Digital*
2. Berdasarkan resfon frekuensi yaitu, LPF (*low pass filter*), HPF (*high pass filter)*, BPF*(band pass filter)*, BSF*( band stop filter*)
3. Berdasarkan bentuk respon frekuensi yaitu, *Hamming, Kaiser, Firls, Blackman* dan *Fir2*
4. Berdasarkan respon implusnya yaitu, FIR *(funite impulse response)*

## 2.2.Jenis-Jenis Filter

*Filter Low Pass* adalah sebuah rangkaian yang tegangan keluarannya tetap dari dc naik sampai ke suatu frekuensi cut-off fc. Bersama naiknya frekuensi di atas fc, tegangan keluarannya diperlemah (turun).

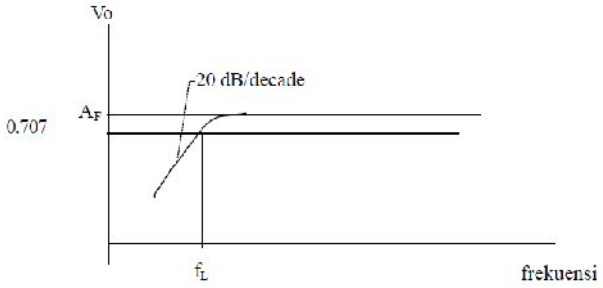
*Low Pass Filter* adalah jenis filter yang melewatkan frekuensi rendah serta meredam/menahan frekuensi tinggi. Bentuk respon LPF seperti ditunjukkan gambar di bawah ini.




Gambar 2.1 Gelombang *Respon LPF*

Frekuensi cutoff (fc) : disebut frekuensi 0.707, frekuensi 3-dB, frekuensi pojok, atau frekuensi putus.

*High Pass* Filter yang melewatkan frekuensi tinggi dan meredam frekuensi rendah



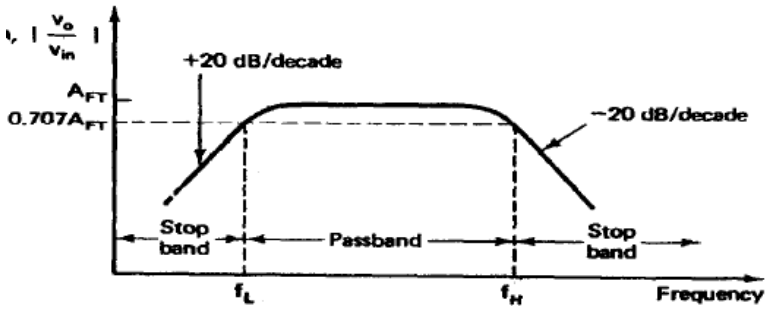

Gambar 2.2 Gelombang *High pass*

*Band Pass* Filter yang melewatkan suatu range frekuensi. Dalam perancangannya diperhitungkan nilai Q ( faktor mutu ) .

Q =faktormutu

fo=frekuensicutoff

B = lebar pita frekuensi




Gambar 2.3 Glombang Bend pass

**2.3. Filter Digital Analog**

*Filter Analog* memainkan peranan penting dalam sintesis subsistem Sinyal elektronik, menyediakan fungsi-fungsi seperti anti-aliasing dan kebisingan penyaringan untuk ADC, dan rekonstruksi pasca-penyaringan untuk DAC. Desain yang berbeda menentukan spesifikasi dan persyaratan penggunaan filter konfigurasi tertentu. Filter yang paling populer termasuk *Hamming, Blackman*, dan *Fir2*. Respon fase mereka bervariasi *nonlinearly* dengan frekuensi. Ciri-ciri ini sekarang tidak ada masalah untuk aplikasi berbasis amplitudo.

Seperti disebutkan sebelumnya, karena karakteristik mereka, *Bessel filter* domain waktu ideal untuk aplikasi, karena mereka tidak memiliki *distorsi* pada osiloskop / *analyzer* jenis pengukuran. Persamaan umum *filter Butterworth* yang telah dibahas sebelum-nya yang dapat ditulis kembali kedalam bentuk persamaan :

$$|Hn(J\omega)|^2 - \frac{1}{1+(\omega/\omega C)^{2N}} \quad \text{..... (1)}$$

Dimana parameter ωc yang mewakili frekuensi kritisfilter dan parameter N yang menyatakan tingkatan orde. Kemudian ditentukan nilai-nilai yang memenuhi kriteria untuk mendisain sebuah model *filter low-pass* dengan tingkat penguatan yang cukup dan zona *transitional band* yang tidak terlalu lebar seperti:

Selanjutnya nilai parameter δ1 dan δ2 akan digabungkan dengan nilai parameter frekuensi *pass-band* (ω1) nilai parameter ω1 diperoleh dari frekuensi *stop-band* (*ω2*) untuk mendapatkan tingkatan ordefilter yang diperlukan dengan

menggunakan persamaan :

$$N = \frac{10\left[\frac{10^{-k1/10}-1}{10^{-k2/10}-1}\right]}{2\,log\frac{\omega 1}{\omega 2}} \quad \text{..... (2)}$$

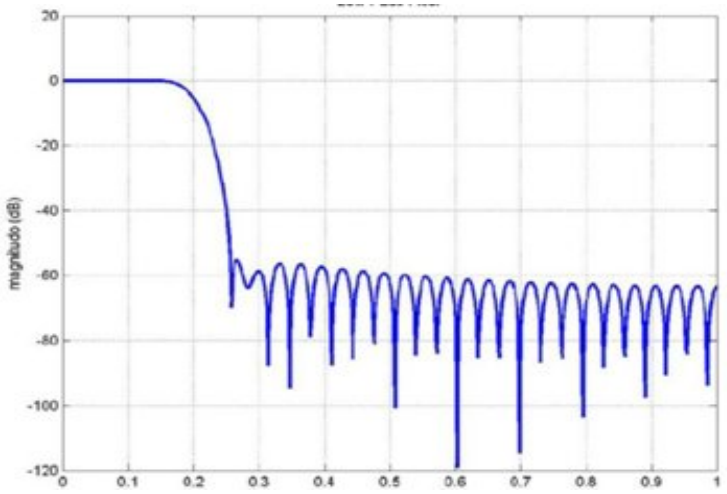
Setelah diperoleh tingkatan ordefilter yang diperlukan, selanjutnya akan dicari persamaan *transfer function* dari filter dengan menggunakan persamaan *Polinom Butterworth*. Untuk orde, diperoleh persamaan polinom bentuk normal :

$$Hn(s) = \frac{1}{\pi(s-sk)} \quad \text{..... (3)}$$

## 2.4. Filter Digital FIR

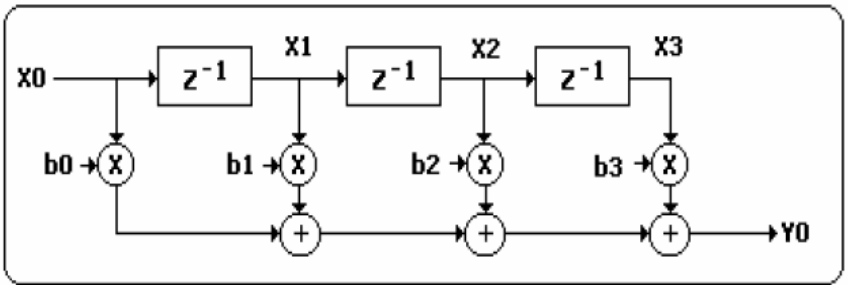
Filter digital *funite impulse response* (FIR) merupakan sistem *open loop* atau dikenal juga dengan sistem non-recursive. Pada sistem yang bersipat open loop/tanpa *feedback*, kesetabilan sistem tidak dapat di kendalikan, sehingga untuk memperoleh hasil respons yang lebih baik/stabil daerah kerja dari filter FIR harus dibatasi. Beberapa persyaratan yang harus diperhatikan antara lain :

- Panjang filter (N), pada filter digital FIR setara dengan orde.
- Daerah kerja redaman, pada daerah batas tertentu akan terjadi osilasi, seperti terlihat pada Gambar 6.1.
- Dipilih daerah kerja yang mempunyai *Phase delay linear.*




Gambar 2.4 Kurva magnitude response filter LPF FIR

Filter FIR adalah salah satu tipe dari filter digital yang dipakai pada aplikasi Digital Signal Processing (DSP). FIR kepanjangan dari Finite Impulse Response. Mengapa disebut respons impulsnya terbatas Karena tidak ada *feedback* didalam filter, jika anda memasukkan sebuah impulse (yaitu sebuah sinyal '1' diikuti dengan banyak sinyal '0'), sinyal nol akan keluar setelah sinyal 1 melewati semua delay line dengan koefisiennya. Keuntungan filter FIR antara lain adalah stabil dan memiliki phasa yang linier. Sedangkan kerugiannya adalah filter FIR terkadang membutuhkan lebih banyak memory dan perhitungan untuk mencapai karakteristik respon filter yang diberikan. Dan juga, respon tertentu tidak mudah dilaksanakan untuk diimplementasikan dengan filter FIR. Flow graph dari filter FIR ditunjukkan oleh Gambar 2.4.1




Gambar 2.5 Flow Graph Filter FIR

$$y[n] = \sum_{m=-\infty}^{\infty} h[m]x[n-m]$$

$x[n]$ → h[n] → y[n]

Untuk filter FIR: h[n] = {$b_0 b_1$ .....$b_q$ }

↑

n = 0

y[n] = $\sum_{m=0}^{q} b_m x[n-m]$

### 2.4.1. FIR Memiliki Respon Terbatas

Dalam kasus umum, respon impuls terbatas karena tidak ada umpan balik dalam FIR. Kurangnya jaminan umpan balik bahwa respon impuls akan terbatas. Oleh karena itu, "respon impulse yang terbatas istilah" hampir identik dengan "tidak ada umpan balik". Namun, jika umpan balik digunakan namun respon impuls terbatas, filter masih merupakan FIR. Contoh adalah filter rata-rata bergerak, di mana sampel sebelum Nth dikurangi makan kembalisetiap kali sampel baru yang masuk Filter ini memiliki respon impulse yang terbatas

meskipun menggunakan umpan balik: setelah N sampel dari impuls, output selalu akan menjadi nol.

### 2.4.2. Keuntungan dari Filter FIR

Dibandingkan dengan *filter IIR, filter FIR* menawarkan keuntungan sebagai berikut:

FIR dengan mudah dapat dirancang untuk menjadi "*fase linier*". Sederhananya, *linier-fase* penundaan filter sinyal input, tetapi tidak mengganggu fase.

FIR sederhana untuk diimplementasikan. Pada kebanyakan mikroprosesor DSP, perhitungan FIR dapat dilakukan dengan perulangan sebuah instruksi.

FIR cocok untuk aplikasi *multi-rate*. Dengan *multi-rate* "penipisan" (mengurangi laju sampling), "*interpolasi*" (meningkatkan tingkat sampling) atau keduanya. Penggunaan *filter FIR* memungkinkan beberapa perhitungan harus dihilangkan, sehingga memberikan efisiensi komputasi penting. Sebaliknya, jika *filter IIR* yang digunakan, out-put masing-masing harus dihitung secara individual, bahkan jika itu output yang akan dibuang (sehingga umpan balik akan dimasukkan ke dalam filter).

FIR memiliki sifat angka yang diinginkan. Dalam prakteknya, semua filter DSP harus dilaksanakan dengan menggunakan aritmatika presisi berhingga yaitu, sejumlah *bit*. Penggunaan aritmatika presisi berhingga dalam *filter IIR* dapat menyebabkan masalah yang signifikan akibat penggunaan umpan balik, tapi *filter*

*FIR* tanpa umpan balik biasanya dapat diimplementasikan dengan menggunakan bit yang lebih sedikit, dan desainer memiliki lebih sedikit masalah praktis untuk memecahkan berhubungan dengan non-ideal aritmatika.

FIR dapat diimplementasikan menggunakan aritmatika. Tidak seperti *filter IIR*, itu selalu mungkin untuk menerapkan sebuah *filter FIR* menggunakan koefisien dengan besarnya kurang dari 1,0. (Keuntungan keseluruhan dari *filter FIR* dapat disesuaikan pada output, jika diinginkan.) Ini merupakan pertimbangan penting ketika menggunakan *fixed-point* DSP, karena membuat implementasi lebih sederhana.

### 2.4.3.Kerugian dari Filter FIR

Dibandingkan dengan *filter IIR, filter FIR* terkadang memiliki kelemahan bahwa mereka memerlukan lebih banyak memori dan perhitungan untuk mencapai karakteristik respon filter yang diberikan.

## 2.5. MetodaFilter FIR

Terdapat dua buah metoda untuk mendisain sebuah filter digital FIR, yaitu metoda disain secara langsung (*directdesign*) dan metoda disain secara tidak langsung (*indirectdesign*). Proses *direct design* memerlukan perhitungan aproksimasi matematis dan membutuhkan pemakaian perhitungan diferensial yang rumit untuk setiap nilai parameter dari *transfer function* filter yang dikehendaki. Hal ini membuat persamaan matematis filter tersebut menjadi non-linier dan sukar dipecahkan. Untuk mendapatkan hasil perhitungan biasanya diperlukan

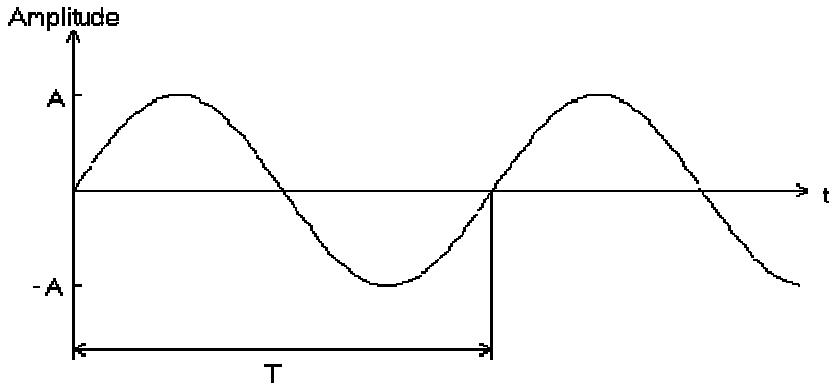
bantuan algoritma metode *numeric* dengan sebuah komputer.

Metoda disain secara tidak langsung relatip lebih sederhana dan lebih mudah dilakukan. Metoda ini terbagi dalam dua langkah utama, yaitu:

1. Mendisain secara matematis sebuah filter *prototype* berupa sebuah filter analog dengan spesifikasi yang diinginkan. Dari filter analog ini kemudian dicari persamaan *transfer function* analog-nya H(s).
2. *Transfer function* dari filter prototype kemudian ditransfor masikan kedalam bentuk diskritnya. Prosestransformasi ini dapat menggunakan beberapa metoda, seperti *impulse-invariant* dan *bilinear transformation*.

### 2.5.1. domain waktu dan domain frekuensi

Frekuensi domain adalah hanya cara lain untuk mewakili sinyal. Sebagai contoh, perhatikan sinusoid sederhana.




Gambar 2.6 Gelombang Sinusoida

Frekuensi menentukan amplitudo sumbu frekuensi mirip dengan cara frekuensi waktu ditentukan oleh waktu sumbu amplitudo. frekuensi diwakili ketika spektrum dari sinyal ditampilkan.

Perhatikan bahwa sinyal waktu dapat dianggap sebagai proyeksi jika sinusoid tersebut ke bidang waktu (waktu sumbu amplitudo). Sinusoid yang sebenarnya dapat dianggap sebagai sudah ada beberapa jarak di sepanjang sumbu frekuensi jauh dari frekuensi waktu. Jarak sepanjang sumbu frekuensi sinusoid itu, sama dengan kebalikan dari periode sinusoid itu.

Gelombang juga memiliki proyeksi ke bidang frekuensi. Jika Anda membayangkan diri Anda berdiri pada sumbu frekuensi, memandang ke arah sinusoid, Anda akan melihat sinusoid hanya sebagai sebuah garis. Garis ini akan memiliki ketinggian sama dengan amplitudo sinusoid. Jadi proyeksi sinusoid yang ke bidang frekuensi hanyalah sebuah garis sama dengan amplitudo sinusoid itu.

Kedua proyeksi berarti bahwa sinusoid muncul sebagai sinusoid dalam bidang waktu time sumbu amplitudo, dan sebagai garis pada bidang frekuensi frekuensi sumbu amplitudo naik dari sinusoid frekuensi untuk ketinggian sama dengan amplitudo sinusoid itu.

Perlu dicatat sangat hati-hati bahwa semua informasi tentang sinusoid ini frekuensi, amplitudo dan fase diwakili dalam proyeksi pesawat waktu, tetapi semua informasi fase hilang dalam proyeksi ke bidang frekuensi.Jika sinyal penuh

harus direkonstruksi dari representasi frekuensi maka grafik tambahan yang disebut *diagram fase* diperlukan. Diagram fase hanyalah sebuah grafik dari fase vs frekuensi, mirip dengan amplitudo vs frekuensi grafik diperoleh dari pesawat frekuensi.

**2.6. DSK TMS320C31**

DSKTMS320C31 adalah salah satu dari seri DSP Starter Kitkeluaran Texas Instrument. Board DSKTMS320C31 dapat dilihat pada Gambar2. Beserta skema umum DSK TMS320C31 pada perangkat DSK yang berbasi smikroprosessor TMS320C31-50 ini dapat digunakan untuk berbagai macam fungsi DSP, termasuk sebagai sebuah filter digital FIR.

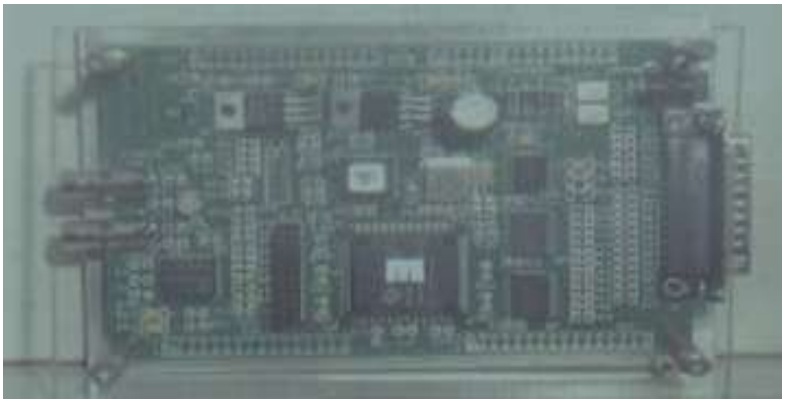
Filter FIR sebenarnya hanya set koefisien FIR. (Jika anda memasang sebuah "*impluse*" ke sebuah filter FIR yang terdiri dari sampel "1" diikuti oleh banyak sampel "0", output dari filter akan set koefisien, sebagai 1 bergerak sampel masa lalu masing-masing koefisien pada gilirannya untuk membentuk out put .

*Transisi Band* frekuensi antara tepi *pass band* dan *stop band* Sempit, keran lebih banyak diperlukan untuk melaksanakan filter. ( Sebuah hasil band transisi di filter "tajam" )

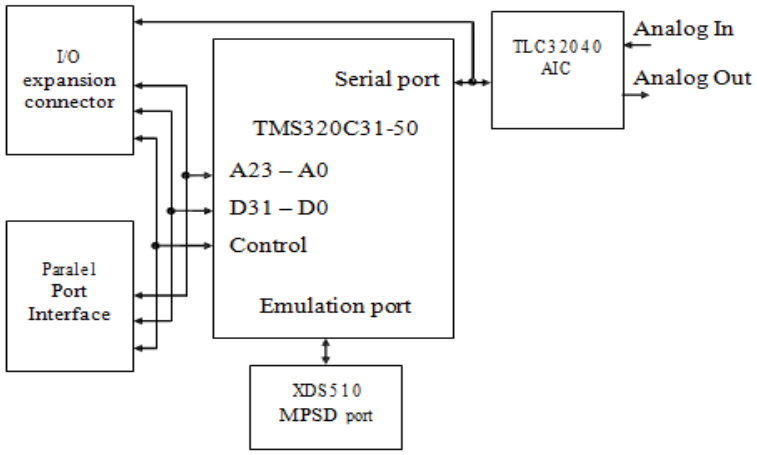
*Delay Line* Himpunan elemen memori yang menerapkan "Z ^ -1" unsur penundaan perhitungan FIR.

*Edaran Buffer* Penyangga khusus yang "melingkar" karena *incrementing* pada akhirnya menyebabkan *Edaran Buffer,* atau karena *decrementing* dari awal

menyebabkan *Edaran Buffer*. *Edaran Buffer* sering disediakan oleh DSP mikroprosesor untuk menerapkan "gerakan" dari sampel melalui FIR delay-line tanpa harus harfiah memindahkan data dalam memori. Ketika sampel baru ditambahkan ke *buffer,* maka secara otomatis menggantikan yang terlama.



Gambar 2.7 DSKTMS320C31




Gamabr 2.8 Sekema umun DSK TMS320C31

Komponen-komponen penting yang terdapat dalam gambar diatas adalah:

1. Mikroprosessor *floating-point* TMS320C31-50, prosessor ini dilengkapi

dengan memori kerja sebesar 16MW.

2. *Chip* converter DAC/ADC AIC TLC32040, yang berguna untuk menerima input analog dari sumber (Signal Generator) kemudian merubahnya kedalam bentuk digital 16 bit, dan begitupula sebaliknya agar hasil filtering dari DSK dapat ditampilkan dalam instrument pengukuran analog (*Oscilloscope*).

*Board*DSK ini terhubung dengan sebuah mikrocomputer melalui jalur *parallel port*. Mikrocomputer berfungsi sebagai host yang menyediakan proses *assembling* dan *debugging* yang diperlukan dalam mengolah *source code* yang berisi perintah untuk dijalankan oleh mikroprosessor.

## 2.7. MATLAB 7

Matlab 7 merupakan software program aplikasi yang digunakan untuk komputasi teknik. Nama Matlab merupakan singkatan dari matrix laboratory. Matlab mampun mengintegrasikan komputasi, visualisasi dan pemograman untuk dapat digunakan secara mudah.Penggunaan Matlab diantaranya adalah pada:

1. Matematika dan Komputans
2. Pengembangan algoritma
3. Pemodelan simulasidan prottying
4. Analisa eksplorasi danvisualisasi data

5. Pengolahan grafik untuk sains dan teknik

Pada tugasakhir ini Matlab 7 digunakan untuk proses pengolahan data, yakni proses yang berkaitan dengan analisa dan visualisasi data.

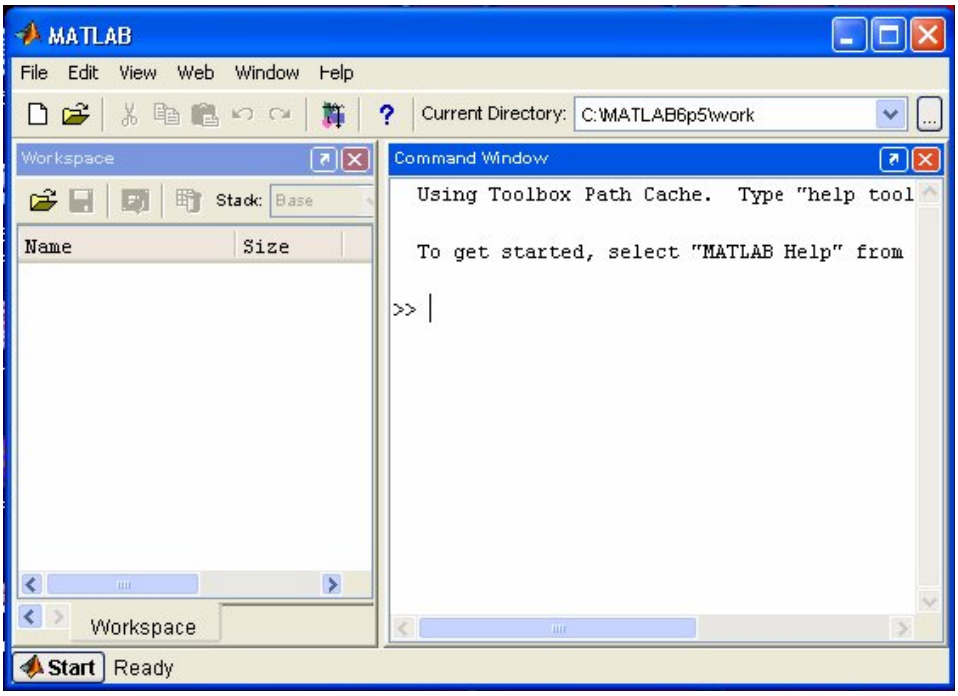
### 2.7.1. LingkupMatlab

Ada beberapa tools yang disediakan oleh Matlab 6.5 diantaranya sebagai berikut:

- *Command Window*, yang berfungsi untuk tempat memasukkan dan menjalankan variabel (fungsi) dari Matlab dan M File.
- *Command History*, yang berfungsi menampilkan fungsi-fungsi yang telah dikerjakan pada *command window*.
- *Launch Pad*, yang berfungsi untuk akses tools, demo, dan dokumentasi semua produk Math Works.
- *Help Browser*, yang berfungsi untuk menampilkan dan mencari dokumentasi yang ada pada Matlab.
- *Current Directory Browser*, yang berfungsi menampilkan file-file Matlab dan file yang terkait serta mengerjakan operasi file seperti membuka dan mencari isi file.

- *Workspace Browser*, yang memuat variabel-variabel yang dibuat dan yang disimpan dalam memori saat penggunaan Matlab.

- *Editor / Debugger*, yang berfungsi untuk membuat dan memeriksa M File. Beberapa tools ini merupakan tools yang secara umum digunakan pada Matlab, Namun sebenarnya selain itu ada banyak tools tambahan lainnya pada Matlab.

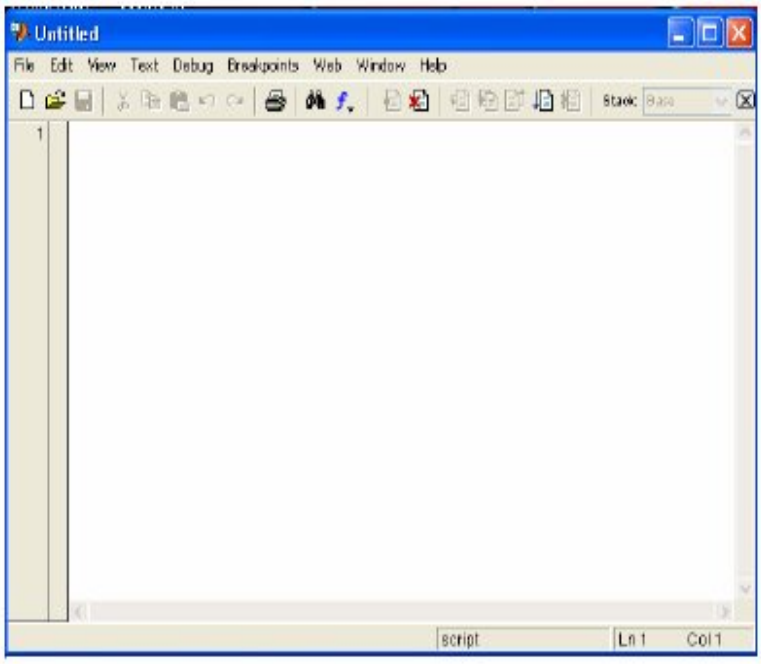



Gambar 2.8 Tampilan Program Matlab

### 2.7.2. M File Editor

M File merupakan file teks yang memuat variabel- variabel dan fungsi yang ada pada Matlab. M File berupakan nama file script dalam Matlab yang disimpan dengan ekstensi ‘m’. M File memudahkan dalam penulisan (pembuatan)

program dalam Matlab. Dimana fungsi-fungsi yang ada pada M File tersebut dapat mengakses semua variable Matlab dan menjadi bagian dari ruang kerja Matlab.




Gambar 2.10 M File

# BAB III

# PERANCANGAN

Perancangan dan pembuatan simulasi merupakan hal yang sangat penting dalam suatu proyek laporan akhir. Pada tahap awal dilakukan perancangan filter digital FIR, untuk itu diperlukan data-data yang memuat spesifikasi filter digital tersebut, dan kemudian dilanjutkan dengan perbandingan ke dalam simulasi.

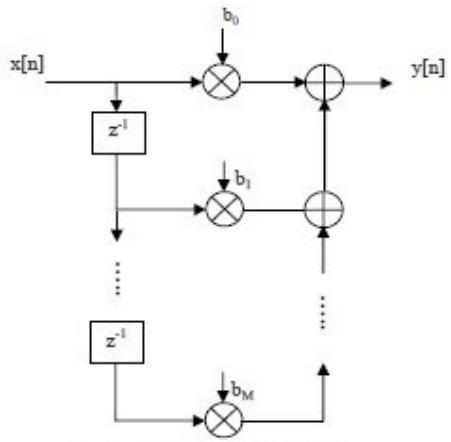
## 3.1. Tujuan Perancangan Simulasi

Perancangan simulasi merupakan suatu system perancangan yang dilakukan dalam membuat suatu filter digital yang akan dijalankan dan dianalisa dalam bentuk sinyal. Dalam perancangan simulasi ini terdapat beberapa prosedur yang digunakan untuk mempermudah proses pembuatan simulasi. Prosedur perancangan simulasi ini terdiri dari pembuatan simulasi dari berbagai jenis filter digital FIR dengan freuensi sampling yang sama dengan orde yang sama. Dimana frekuensi cut-off adalah setengah dari frekuensi sampling.

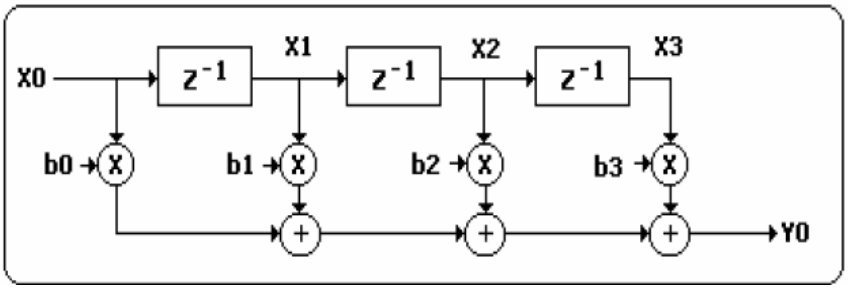
## 3.2. Perancangan Simulasi Filter Digital FIR pada Matlab

Pada gambar blok diagram dibawah Filter FIR adalah salah satu tipe dari filter digital yang digunakan pada aplikasi DSP. Karena filter FIR ini terbatas jadi tidak ada *feedback* didalam filter-nya, jadi jika ada masukan sebuah impulse (

yaitu sebuah sinyal ‘1’ diikuti dengan banyak sinyal ‘0’) maka sinyal nol akan keluar setelah sinyal 1 melewati semua delay line dengan koefisiennya.

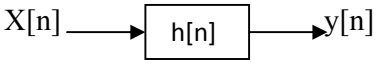



Gambar 3.1 Diagram Blok FIR




Gambar 3.2 Flow grafik FIR

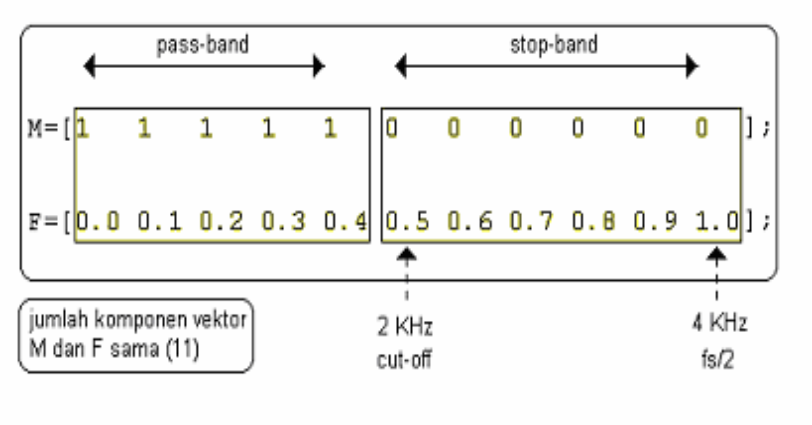
Diagram blok diatas menyatakan hubungan input dan outputnya bisa dirancang secara sederhana sebagai.



### 3.2.1 Menghitung Nilai Koefisien Menggunakan Matlab

Untuk mengitung nilai koefosien dari response filtern FIR terlebih dahulu yaitu. Menghitung nilai cut-off pada vektor F-nya dimana setengah frekuensi sampling ( 8KHz : 2) = 4 KHz → pada vektor F nilai 1.0 harus sama dengan 4KHz .

Untuk frekuensi cut-off 2KHz dimana (2KHz:4KHz) = 0.5→pada fektor 0.5 untuk menentukan nilai vektor M bisa dilihat pada gambar dibawah ini.




Gambar 3.3 Menentukan nilai vektor M terhadap vektor

Pada gambar diatas terlihat bahwa diharapkan pada daerah setelah cut-off 2KHz terjadi stop-band. Kemudian menghitung koefisien filter pada Matlab digunakan perintah seperti.

*b=fir2 (orde,F,M,blackman (orde+1));*

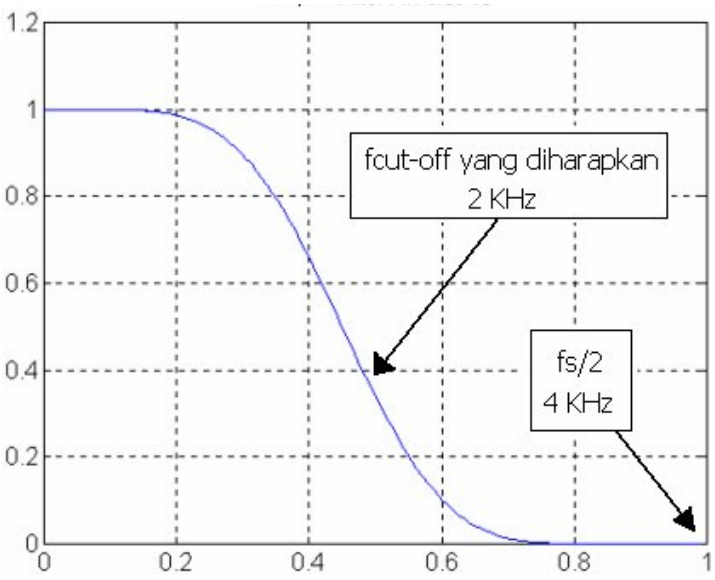
*untuk orde 16 maka :*

*b=fir2 (16,F,M,blackman (17));*

untuk melihat hasil perancangan plot respon filternya maka kita lihat rancangan dibawah yang menunjukan listing filter low-pass FIR.

```
M=[  1   1   1   1   1   0   0   0   0   0   0  ] ;
F =[  0  .1  .2  .3  .4  .5  .6  .7  .8  .9  1. ] ;
b = fir2 ( 16, F, M, blackman ( 17 ) ) ;
[h,w] = freqz ( b,1,128 ) ;
Plot (w / pi,abs ( h ) ) ;
```

Gambar 3.4 Listing filter low-pass FIR pada Matlab




Gambar 3.5 Plot respon frekuensi filter FIR orde 16

### 3.2.2 Mengkonversi Nilai Koefisien

Nilai b dibawah ini adalah hasil rancangan matlab diatas, untuk mendapatkan nilai koefisienya mengetikan b pada *Matlab command Window* maka akan tampil seperti nilai- nilai dibawah ini.

```
>> b ( enter )

b =

Coloms 1 through 7

0.0000  -0. 0002  0.0025  0.0069  -0.0151  -0.0504   0.0380

Columns 8 throgh 14

0.1938  0.4492    0.2938    0.0380   -0.0504   -0.0151   0.0069

Columns 15 through 17

0.0025   -0.0002   0.0000
```

Nilai – nilai diatas masih dalam berbentuk nilai pecahan, sedangkan TMS320C31 yang digunakan mempunyai tipe fixed-point (integer) sehingga harus dikonversi dengan mengalikan nilai koefisien yang didapat dengan Matlab tadi dengan bilangan $2^{15}$ (32768). Kemudian hasil perkalian yang sudah

dibulatkan tadi dapat digunakan didalam program FIR seperti nilai – nilai yang sudah dikalikan dibawah ini.

```
>> c = b*32768
C = 1.0e-004*
Columns 1 through 7
0.0000  -0.0008   0.0082   0.0227   -0.0496   -0.1652    0.1246
Columns 8 through 14
0.9627   1.4720   0.9627   0.1246   -0.1652   -0.0496   0.0227
Columns 15 through 17
0.0082   -0.0008    0.0000
```

Maka setelah nilai- nilai diatas dibulatkan kita dapat mengetahui nilai koefisienya yaitu.

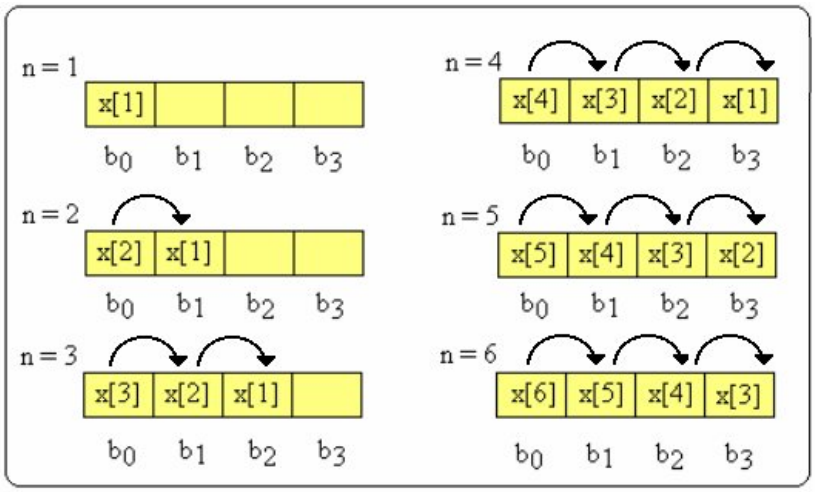
| Nilai koefisien filter FIR |
|---|
| 0.015062 |
| 0.209623 |
| 0.209623 |
| 0.015062 |

Rancangan dibawah adalah prinsip kerja dan rancangan filter FIR di dalam Matlab.

```
m=[1 0.8 0 0 0 0 ]; % vektor m
f=[0.0 0.25 0.4 0.6 0.7 1.0]; % vektor f
n=16; fs2=8; % orde 16, ½ frek.sampling=8KHz
b=fir2(n,f,m); % hitung koefesien filter
% tulis kedalam file fir.txt
fid = fopen('fir.txt','w'); % buka file
fprintf(fid,strcat('Koefesien filter FIR orde-
',int2str(n),'\n'));
fprintf(fid,strcat('Generated on-',datestr(now),'\n\n'));
for i=1:n+1 % simpan dalam file
```

### 3.3. Membuat Perancangan Program Filter FIR pada DSP

Seperti diagram flow diatas pada gambar 1, maka langkah selanjutnya akan seperti dibawah ini.




Gambar 3.6 Langkah program filter FIR dengan menggunakan orde 4

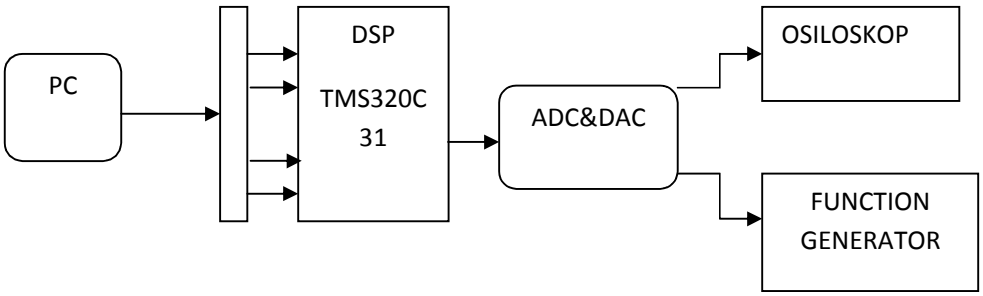
Setiap data yang masuk melewati ( x[n] ) ditampung pada sebuah buffer, kemudian tiap elemen buffer dikalikan dengan koefisien dan dijumlahkan

menghasilkan sebuah output, untuk data selanjutnya isi buffer digeser dan diperlakukan sama seperti pada keadaan pertama. Dibawah ini cara membuat buffer dan melakukan pergeseran buffer dalam bahasa C.

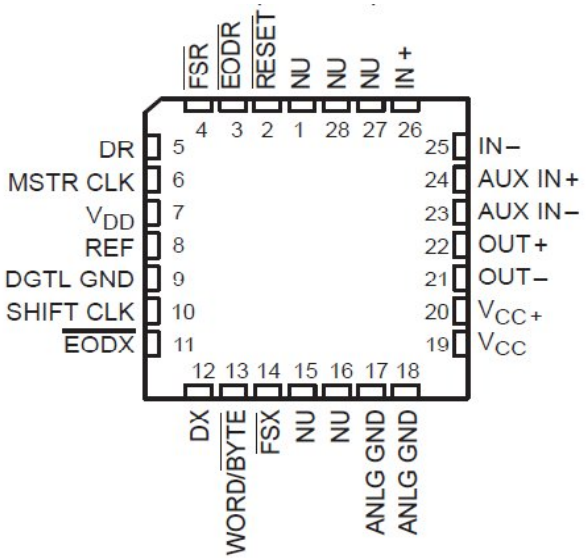
```
Int buffer [ 4 ] ;
. . .
. . .
for  ( k = 3 ;  k > 0 ;  k -- )
{
Buffer [ k ] = buffer  [ k – 1 ] ;
}
```

### 3.4 Perancangan Perangkat Keras (hardware)

Perancangan perangkat keras ini merupakan perencanaan yang dilakukan untuk mengetahui sinyal keluaran dari DSP melalui osiloskop yang diberi masukan dari *function generator*.sehingga masukannya *function generator* akan sama dengan masukan dari PC berupa sinyal sinus dengan keluaran sinyal kotak.




Gambar 3.7 Blok Diagram Perancangan Pengukuran Alat



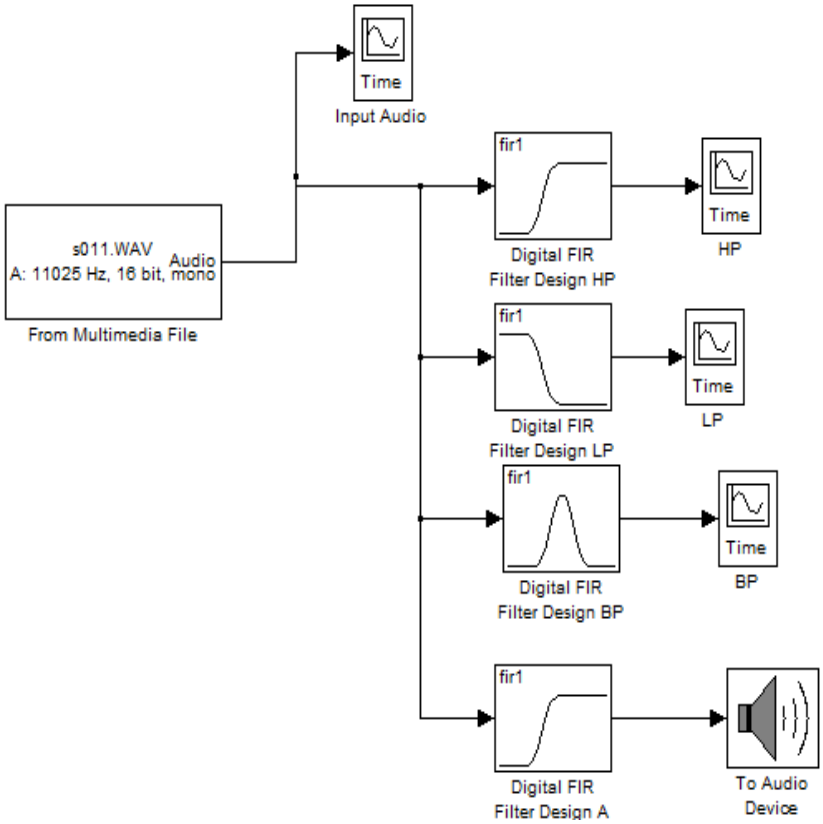

Gambar 3.8 Skema Rangkaian ADC/DAC TLC32040CFN

Maka dari skema diatas kita dapat persamaan rumus ADC :

Resolusi = $\frac{V_{in}}{2^n}$ .....(1)

### 3.5 Perancangan Simulasi dengan Menggunakan MP3

Perancangan simulasi ini merupakan perancangan dimana MP3 dapat dimasukan kedalam simulasi yang akan dibuat. Sehingga dapat kita lihat sinyal frekuensi-nya sesuai type filter digital yang kita rancang. Dimana simulasi ini menggunakan pemodelan. Dimana dibawah ini Perancangan simulasinya:




Gambar 3.9 Skema Rangkaian Simulasi dengan

Mengaplikasikan MP3

# BAB IV

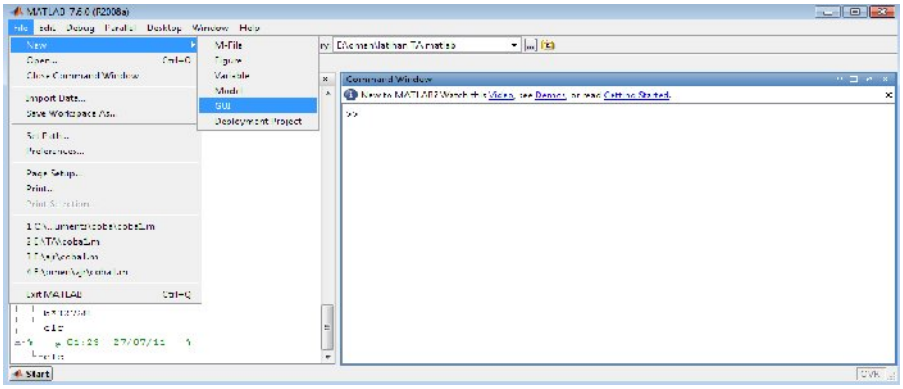
# PENGUKURAN DAN ANALISA

## 4.1. Tujuan Penggukuran dan Analisa Simulasi

Tujuan pengukuran dan analisa simulasi ini adalah untuk membuktikan dan membandingkan hasil percobaan dengan hasil perancangan, sehingga dapat mengetahui bentuk dan frekuensi gelombang dan perbedaan sinyal keluaran tiap jenis filter FIR yang berbeda menggunakan simulasi.

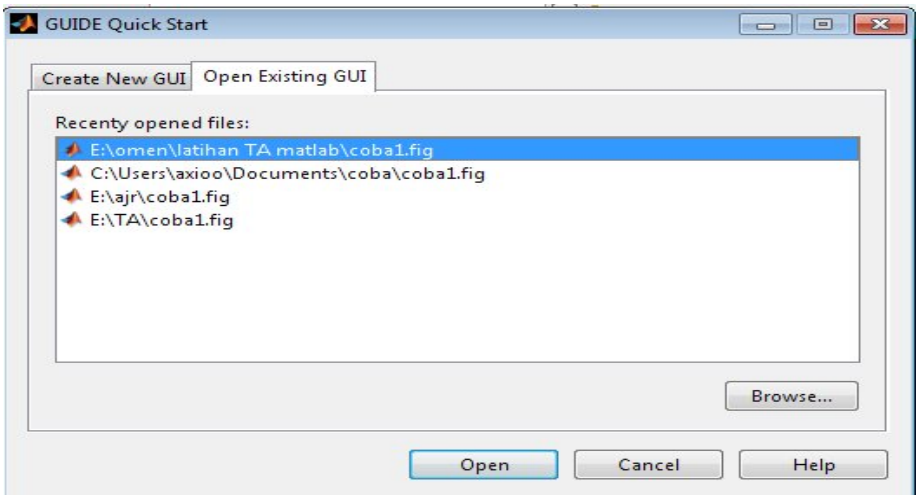
## 4.2. Proses Pengambilan Data Simulasi Filter FIR dalam Matlab

Sebelum melakukan simulasi dan pengambilan data, tahap pertama masuk dulu keprogram Matlab untuk menjalankan simulasi yang akan dijalankan, klik program matlab setelah muncul pilih file dan pilih model Gui untuk melakukan pengujian.



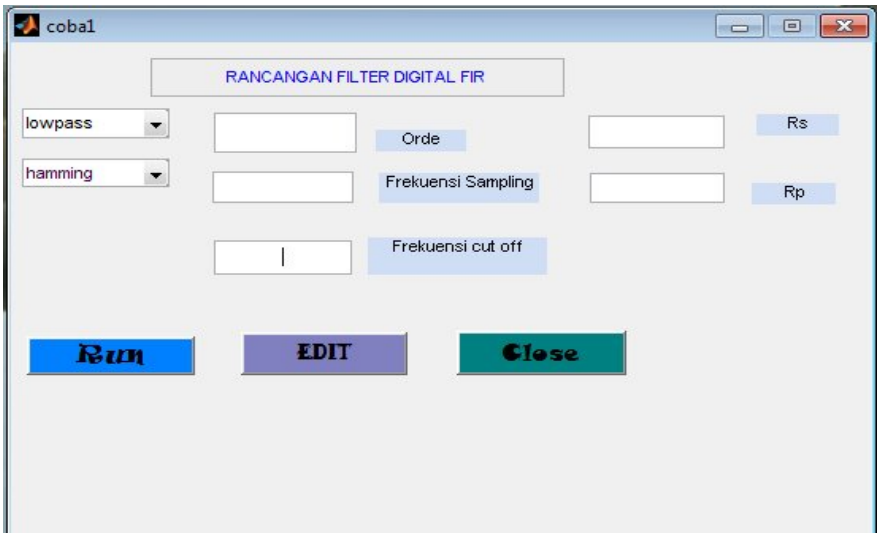

Gambar 4.1 Tampilan Program Matlab Awal Sebelum Proses simulasi

Setelah muncul seperti tampilan diatas maka *doble klik* menu *Gui* dan memilih *Open Existing GUI* untuk mulai menjalankan proses simulasi filter digital FIR, maka akan muncul menu seperti gambar dibawah ini.




Gambar 4.2 Tampilan Program *GUI* Untuk Menjalankan Proses Simulasi

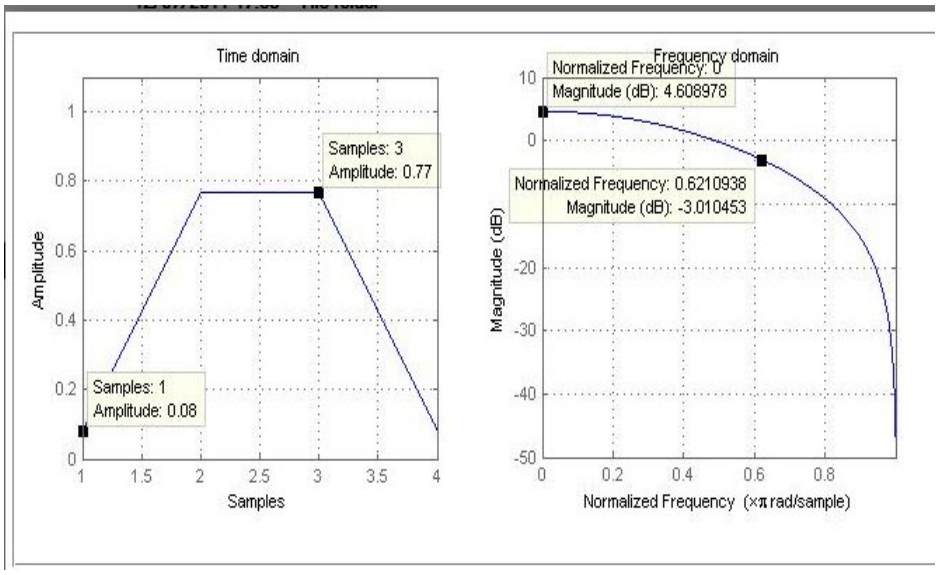
Setelah melakukan pemilihan data seperti tampak pada gambar diatas maka klik *Open* untuk masuk ke simulasi yang akan dijalankan dan dianalisa, maka akan muncul suatu Oprasi Matlab filter FIR dengan menggunakan *GUI*. Seperti tampak pada gambar dibawah ini.




Gambar 4.3 Tampilan Program *GUI* Filter FIR Untuk Menjalankan Proses Simulasi

### 4.3. Percobaan Simulasi jenis-jenis Filter FIR dan Analisa

Simulasi filter FIR tahap pertama ini dimuali dari filter low pass dengan menggunakan type *hamming, Firls, Kaiser, Fir2, Blackman, Fircls1,* . Dmana menggunakan orde 3, frekuensi samplingnya dimulai dari 2000hz dan frekuensi cut-off 1000hz dibawah frekuensi sampling, dengan nilai Rs 5 dan Rp 10.



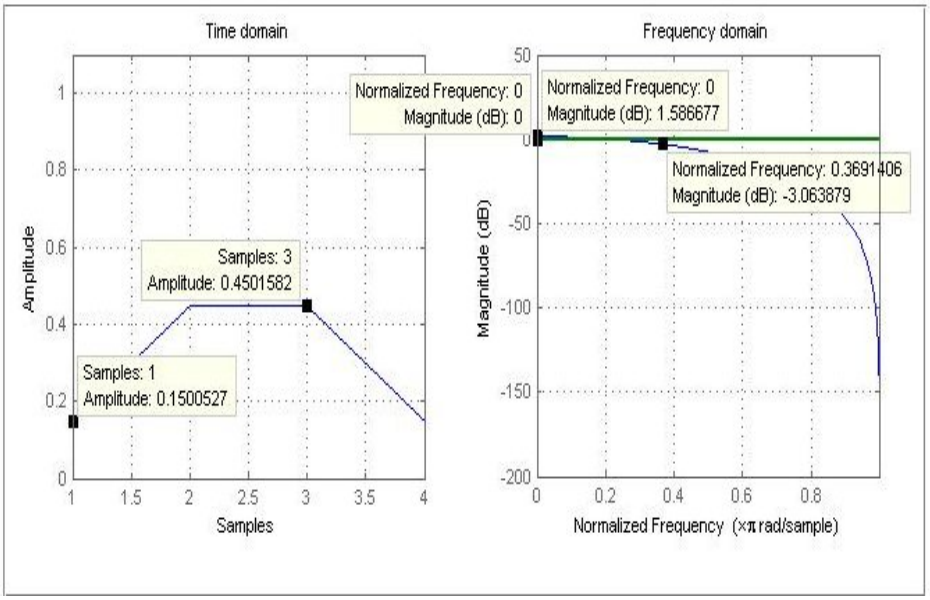

Gambar 4.3 Percobaan Simulasi FIR Hamming orde 3

Dari gambar percobaan di atas maka dapat mengetahui nilai b (koefisien filternya). Dimana tabel dibawah ini nilai koefisien dari setiap samples bentuk gelombang time domain dari samples 1-4.

| | | | | |
|---|---|---|---|---|
| Orde 3 | 0.0800 | 0.7700 | 0.7700 | 0.0800 |

Pada percobaan simulasi jenis *Hamming* orde 3, terlihat digambar amplitudo *time domain*, bahwa titik kecuramannya sama dan titik puncak *Amplitude*-nya bernilai 0,77. Apabila nilai sampling semakin besar maka titik kecuraman bentuk gelombangnya akan semakin curam. Dan nilai masukan dari samples 1 akan sama dengan nilai akhir samples 4 dengan nilai 0,08.

Sedangkan untuk *Magnitude dB, frekuensi domainnya* adalah menunjukan *respons filter dari time domain*. Sehingga bentuk gelombangnya akan terjadi pembatasan dan berhenti dinilai *Magnitude dB* -3,010453 dengan *frekuensi normaliszed*-nya bernilai 0,6, karena dinilai -3 dB akan terjadi pembatasan dan penurunan setengah daya maksimum. Sedangkan nilai-nilai selanjutnya kebawah dianggap nilai redamannya, karena respons frekuensinya berawal dari frekuensi diatas -3dB.




Gambar 4.4 Percobaan simulasi FIR Firls orde 3

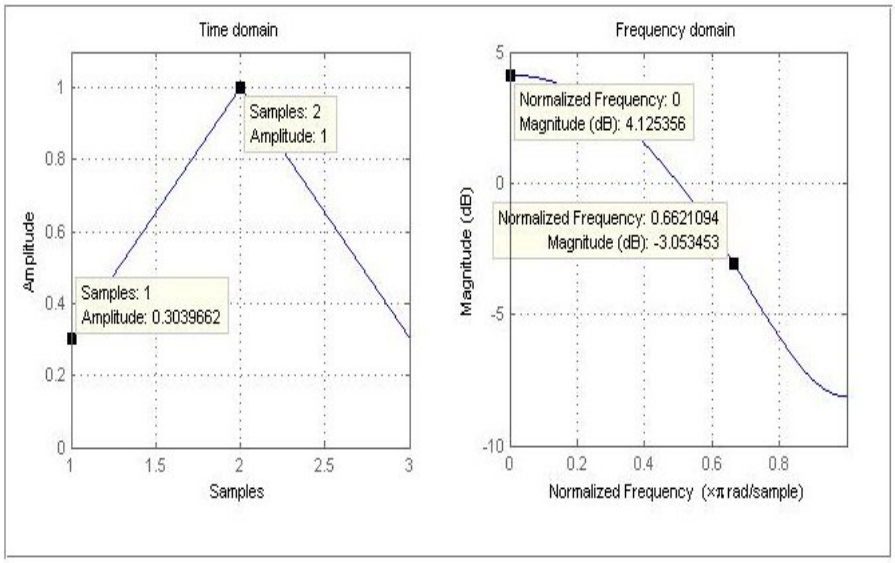
Dari gambar percobaan di atas maka dapat mengetahui nilai b (koefisien filternya). Dimana tabel dibawah ini nilai koefisien dari setiap samples bentuk gelombang time domain dari samples 1-4.

| | | | | |
|---|---|---|---|---|
| Orde 3 | 0.1501 | 0.4502 | 0.4502 | 0.1501 |

Pada percobaan simulasi jenis *Firls* orde 3, terlihat digambar amplitudo *time domain*, bahwa titik kecuramannya berbeda bentuk gelombang tetapi nilai titik kecuramanya sama dan titik puncak *Amplitude*-nya bernilai 0,4501582. Apabila nilai sampling semakin besar maka titik kecuraman bentuk gelombangnya akan semakin curam dan berubah-rubah sesuai tingkatan orde yang dimasukan. Dan nilai masukan dari samples 1 akan sama dengan nilai akhir samples 4 dengan nilai 0,1500527. Ini berbeda dengan jenis *Hamming.*

Sedangkan untuk *Magnitude dB, frekuensi domainnya* adalah menunjukan *respons filter* dari *time domain*. Sehingga bentuk gelombangnya akan terjadi pembatasan dan berhenti dinilai *Magnitude dB* -3,063 dengan *frekuensi normaliszed*-nya bernilai 0,369, karena dinilai -3 dB akan terjadi pembatasan dan penurunan setengah daya maksimum. Sedangkan nilai-nilai selanjutnya kebawah dianggap nilai redamannya, karena respons frekuensinya berawal dari frekuensi diatas -3dB.

Dari percobaan jenis *Firls* ini ketinggian suatu gelombang dibatasi oleh batasan frekuensi maxsimum yang berwarna hijau.sehingga ketinggian gelmbang tidak akan melebihi batas maxsimum tersebut.




Gambar 4.5 Percobaan simulasi FIR Kaiser orde 3

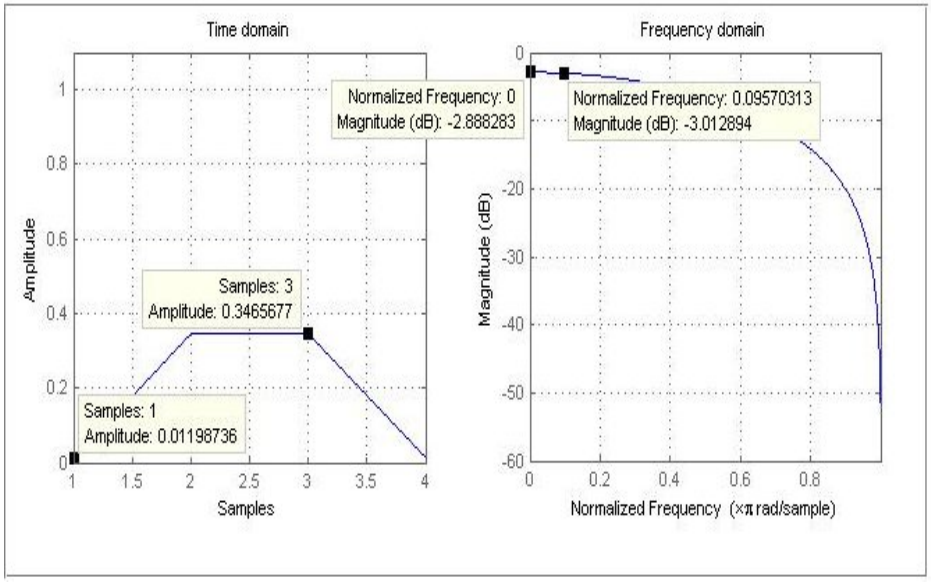
Dari gambar percobaan di atas maka dapat mengetahui nilai b (koefisien filternya). Dimana tabel dibawah ini nilai koefisien dari setiap samples bentuk gelombang time domain dari samples 1-3.

| | | | |
|---|---|---|---|
| Orde 3 | 0.3040 | 1.0000 | 0.3040 |

Pada percobaan simulasi jenis *Kaiser* orde 3, terlihat digambar amplitudo *time domain*, bahwa titik kecuramannya berbeda bentuk gelombang tetapi nilai titik kecuramanya sama dan titik puncak *Amplitude*-nya bernilai 1. Apabila nilai sampling semakin besar maka titik kecuraman bentuk gelombangnya akan semakin curam dan berubah-rubah sesuai tingkatan orde yang kita masukan. Dan

nilai masukan dari samples 1 akan sama dengan nilai akhir samples 3 dengan nilai 0,3039662. Ini sama seperti jenis *Hamming*.

Sedangkan untuk *Magnitude dB, frekuensi domainnya* adalah menunjukan *respons filter* dari *time domain*. Sehingga bentuk gelombangnya akan terjadi pembatasan dan berhenti dinilai *Magnitude dB* -3,053 dengan *frekuensi normaliszed*-nya bernilai 0,662. Sedangkan nilai-nilai selanjutnya kebawah dianggap nilai redaman, karena respons frekuensinya berawal dari frekuensi diatas -3dB.



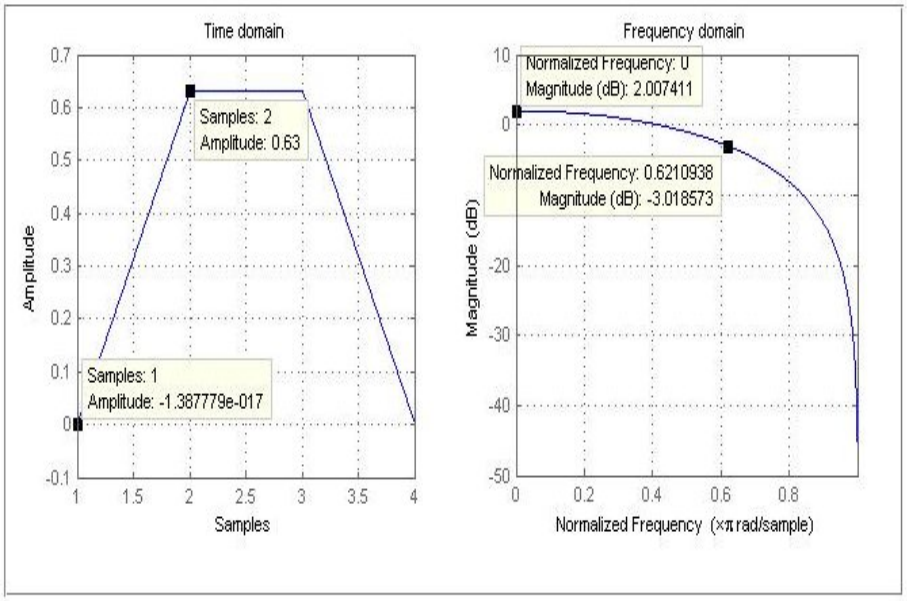

Gambar 4.6 Percobaan simulasi FIR Fir2 orde 3

Dari gambar percobaan di atas maka dapat mengetahui nilai b (koefisien filternya). Dimana tabel dibawah ini nilai koefisien dari setiap samples bentuk gelombang time domain dari samples 1-4.

| | | | | |
|---|---|---|---|---|
| Orde 3 | 0.0120 | 0.3466 | 0.3466 | 0.0120 |

Pada percobaan simulasi jenis *Fir2* orde 3, terlihat digambar amplitudo *time domain*, bahwa titik kecuramannya berbeda bentuk gelombang tetapi nilai titik kecuramanya sama dan titik puncak *Amplitude*-nya bernilai0,346. Apabila nilai sampling semakin besar maka titik kecuraman bentuk gelombangnya akan semakin curam dan berubah-rubah sesuai tingkatan orde yang dimasukan. Dan nilai masukan dari samples 1 akan sama dengan nilai akhir samples 4 dengan nilai 0,011. Jenis *Fir2* ini lebih miirip seperti jenis *kaiser.*

Sedangkan untuk *Magnitude dB, frekuensi domainnya* adalah menunjukan *respons filter* dari *time domain*. Sehingga bentuk gelombangnya akan terjadi pembatasan dan berhenti dinilai *Magnitude dB* -3,012 dengan *frekuensi normaliszed*-nya bernilai 0,095, karena dinilai -3 dB akan terjadi pembatasan dan penurunan setengah daya maksimum. Sedangkan nilai-nilai selanjutnya kebawah dianggap nilai redaman, karena respons frekuensinya dibatasi maka berawal dari frekuensi diatas -3dB.



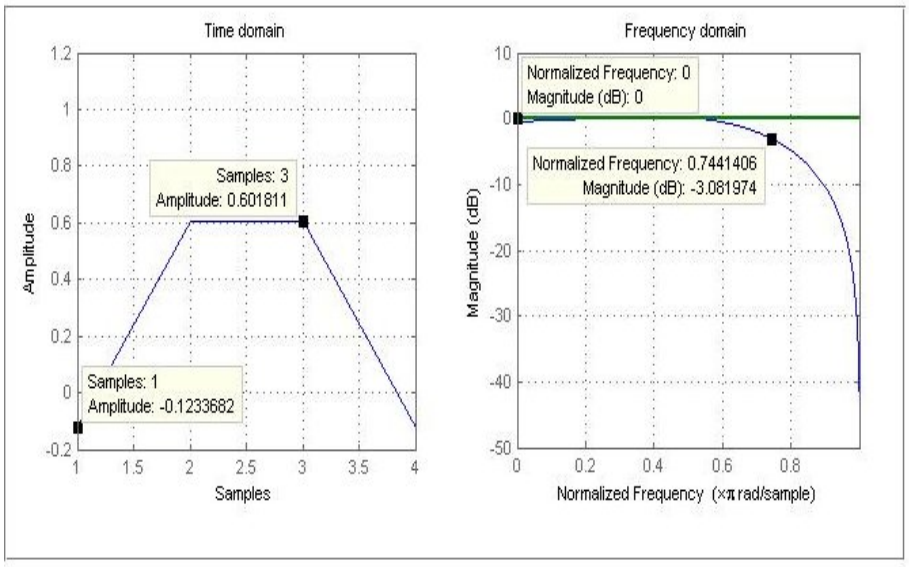

Gambar 4.7 Percobaan simulasi FIR Blackman orde 3

Dari gambar percobaan di atas maka dapat mengetahui nilai b (koefisien filternya). Dimana tabel dibawah ini nilai koefisien dari setiap samples bentuk gelombang time domain dari samples 1-4.

| | | | | |
|---|---|---|---|---|
| Orde 3 | -0.0000 | 0.6300 | 0.6300 | -0.0000 |

Pada percobaan simulasi jenis *Blackman* orde 3, terlihat digambar amplitudo *time domain*, bahwa titik kecuramannya berbeda bentuk gelombang tetapi nilai titik kecuramanya sama dan titik puncak *Amplitude*-nya bernilai 0.63. Apabila nilai sampling semakin besar maka titik kecuraman bentuk gelombangnya akan semakin curam. Dan nilai masukan dari samples 1 akan sama dengan nilai akhir samples 4 brnilai -1,387779e-017.

Sedangkan untuk *Magnitude dB, frekuensi domainnya* adalah menunjukan *respons filter* dari *time domain*. Sehingga bentuk gelombangnya akan terjadi pembatasan dan berhenti dinilai *Magnitude dB* -3,018 dengan *frekuensi normaliszed*-nya bernilai 0,621, karena dinilai -3 dB akan terjadi pembatasan dan penurunan setengah daya maksimum. Sedangkan nilai-nilai selanjutnya kebawah dianggap nilai redaman, karena respons frekuensinya berawal dari frekuensi diatas -3dB.




Gambar 4.8 Percobaan simulasi FIR Fircls1 orde 3

Dari gambar percobaan di atas maka dapat mengetahui nilai b (koefisien filternya). Dimana tabel dibawah ini nilai koefisien dari setiap samples bentuk gelombang time domain dari samples 1-4.

| | | | | |
|---|---|---|---|---|
| Orde 3 | -0.1234 | 0.6018 | 0.6018 | -0.1234 |

Pada percobaan simulasi jenis *Fircls1* orde yang digunakan mulai dari orde 3, karena fircls1 ini berbeda dengan jenis lain, frekuensi samplingnya harus diatas 5000hz dan cut-off-nya harus diantara 1000-2000hz, karena jika kurang atau lebih maka sinyal yang kita inginkan tidak keluar. terlihat digambar amplitudo *time domain*, bahwa titik kecuramannya berbeda bentuk gelombang tetapi nilai titik kecuramanya sama antara kiri dan kanan, dan titik puncak *Amplitude*-nya bernilai 0,601. Apabila nilai sampling semakin besar maka titik kecuraman bentuk gelombangnya akan semakin curam dan berubah-rubah sesuai tingkatan orde yang kita masukan. Dan nilai masukan awal dari samples 1 akan sama dengan nilai akhir samples 4 dengan nilai -0,123.

Sedangkan untuk *Magnitude dB, frekuensi domainnya* adalah menunjukan *respons filter* dari *time domain*. Sehingga bentuk gelombangnya akan terjadi pembatasan dan berhenti dinilai *Magnitude dB* -3,069, dengan *frekuensi normaliszed*-nya bernilai 0,744, karena dinilai -3 dB akan terjadi pembatasan dan penurunan setengah daya maksimum. Sedangkan nilai-nilai selanjutnya kebawah dianggap nilai redaman, karena respons frekuensinya berawal dari frekuensi diatas -3dB.

Dari percobaan jenis *Fircls1* ini ketinggian suatu gelombang dibatasi oleh batasan maxsimum yang berwarna hijau.sehingga ketinggian gelmbang tidak akan melebihi batas maxsimum tersebut. Jenis Fircls1 ini sama persis seperti jenis Firls yang membedakannya adalah nilai samping dan cut-off-nya. Fircls1 mesti lebih besar dari 4000hz nilai samplingny sedangkan cut-off-nya harus berada diantara 1000-2000hz. Tetapi untuk jenis Firls tidak terpengaruh.

**4.3. Analisa Perhitungan**

Menentukan Orde

Mencari dan menentukan nilai $d_1$ dan $d_2$ untuk mendapatkan orde yang kita inginkan.

$20 \log (1 - d_1) = -0,2$

$\text{Log } (1-d_1) = \frac{-0,2}{20}$

$\text{Log } (1-d_1) = -0,01$

$1-d_1 = 10^{-0.01} \rightarrow$ Antilog

$1-d_1 = 0,977$ $d_1 = 1-0,977$

$= 0,023$

$20 \log (1 - d_2) = -25$

$\text{Log } (1-d_2) = \frac{-25}{20}$

$\text{Log } (1-d_2) = -1,25$

$1-d_2 = 10^{-1,25}$

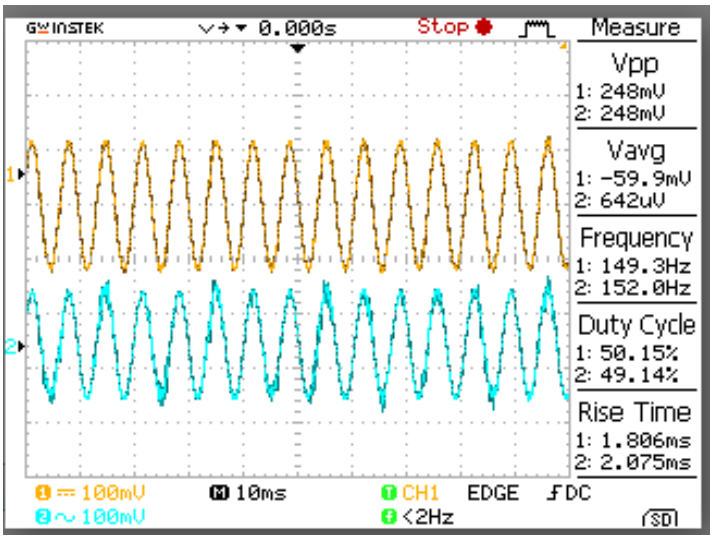
$d_2 = 0,056$

$$N = 1/2\log 10 \frac{\frac{\delta_1(2-\delta_1).\delta_2^2}{(1-\delta_1)^2(1-\delta_2^2}}{log10\frac{Wp}{Ws}}$$

$$= 1/2\log \frac{0{,}028\,(2-0{,}028)0{,}056^2}{(1-0{,}056)}$$

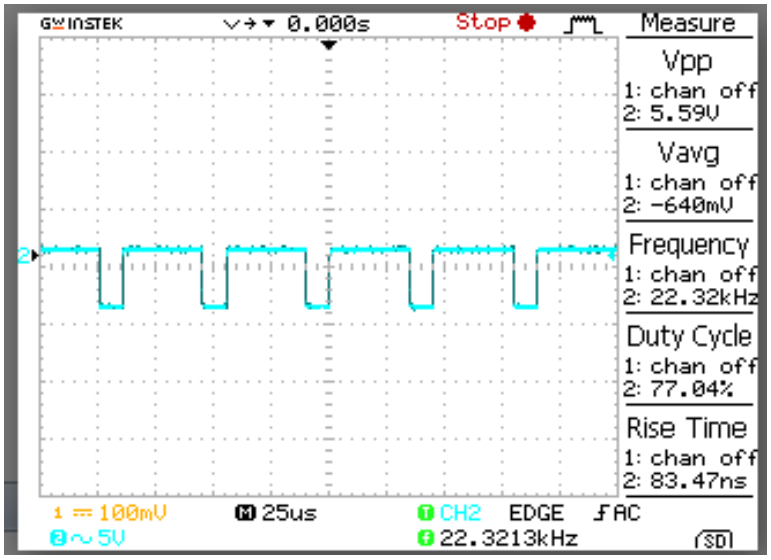
$$= 3$$

### 4.4. Pengukuran Alat dan Analisa




Gambar 4.9 Keluaran DAC

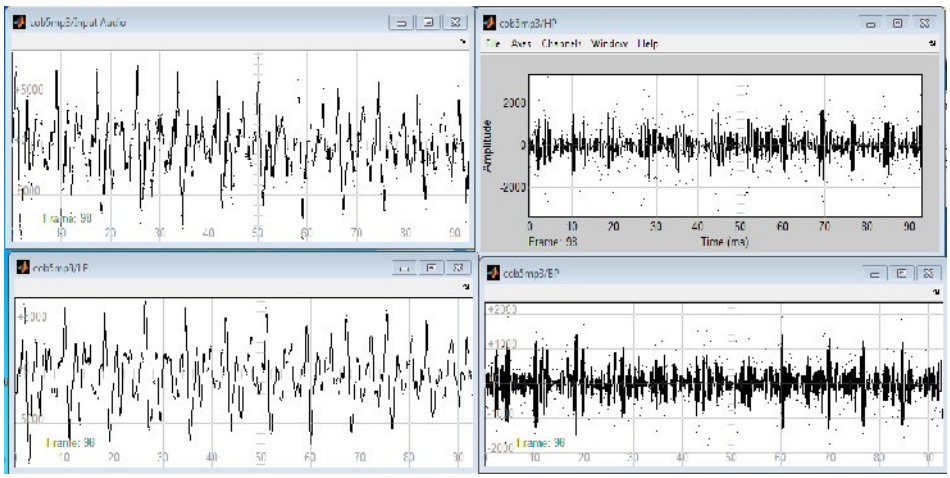
Dapat diamati dari gambar 4.8 diatas, masukan dari PC dengan masukan yang berasal dari *function generator* itu sama. Dimana kedua input keluaranya sama dengan nilai amplitudo dan fase-nya.



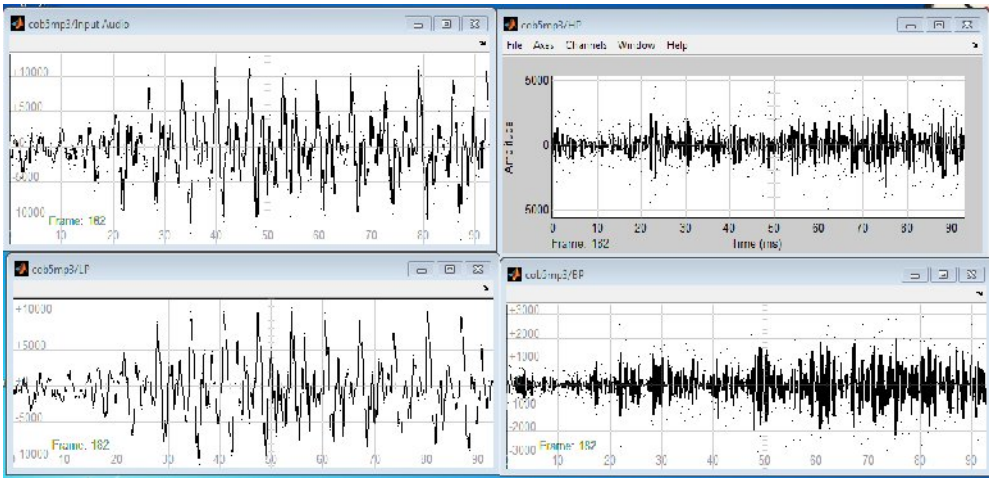

Gambar 4.10 Keluaran ADC

Dari gambar 4.9 dapat diamati bahwa sinyal keluaran dari ADC berbentuk sinyal kotak .

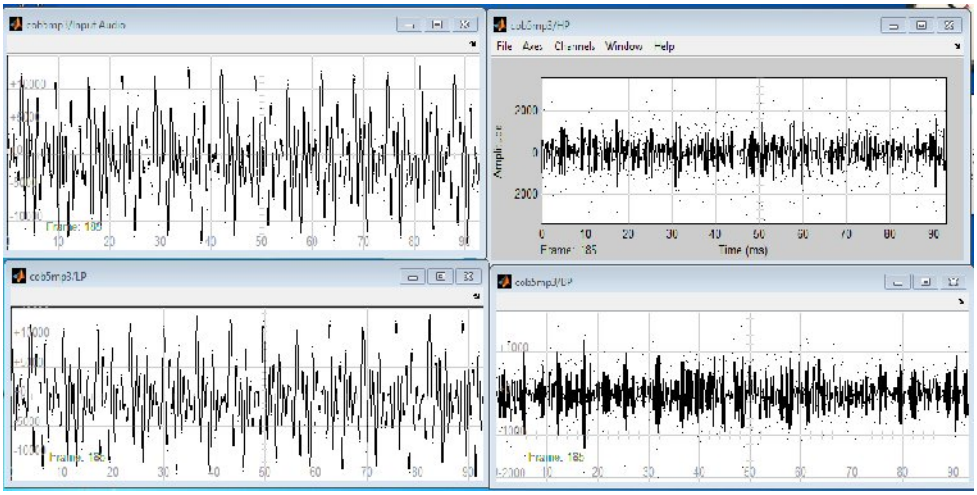
## 4.5. Pengujian Simulasi dan Analisa dengan MP3



Gambar 4.11 Keluaran Simulasi *Low pass*



Gambar 4.12 Keluaran Simulasi *High Pass*



Gambar 4.13 Keluaran Simulasi *Band Pass*

Dapat diamati dari gambar 4.5.1-4.5.3 keluaran diatas untuk kualitas fidelitas suaranya dari jenis filter digitalnya adalah:

Untuk *Low pass*, kualitas fidelitas suaranya lebih bagus dan tidak terjadi pelemahan suara. Sedangkan untuk *High pass* dan *Band pass* ter jadi pelemahan suara dengan fidelitas suaranya kurang jelas meskipun nilai frekuensinya sama.

Dikarenakan dari 3 jenis filter diatas memiliki cara kerja yang berbeda untuk setiap inputan yang kita masukan.

# BAB V

# KESIMPULAN DAN SARAN

## 5.1. Kesimpulan

Dari analisa simulasi dan pembahasan yang telah dilakukan maka dapat diambil beberapa kesimpulan antara lain :

1. Berdasarkan analisa pengukuran untuk simulasi filter FIR terlihat bahwa perbandingan antara jenis filter *Amplitudo time domain* berbeda dalam bentuk gelombang, dengan nilai titik kecuraman yang berbeda.
2. Pada simulasi jenis-jenis filter digital FIR semakin besar nilai *frekuensi sampling*, maka titik kecuramaan suatu gelombang dan bentuknya akan berubah dan menyempit sesuai tingkatan ordenya. Sedangkan di *Magnitude frekuensi domainnya* menunjukan respon frekuensi dari *domain waktu* sehingga akan terjadi pembatasan di nilai -3dB. Kaena di -3dB adalah saat dimana pembatasan filter FIR bekerjanya. Dan menunjukan daya maksimum filter turun ½ daya maksimumnya. Dari beberapa jenis simulasi filter digital FIR terlihat bahwa setiap jenis filter memiliki perbedaan bentuk *response*-nya sesuai jenis filter digital itu sendiri.

3. Dari percobaan alat dengan mengukur ADC dan DAC-nya ternyata ADC memiliki keluaran sinyal kotak, sedangkan DAC sama keluaranya dari amplitudo dan frekuensinya.

4. Berdasarkan simulasi menggunakan pemodelan ternyata keluaran yang didapat adalah sinyal domain waktu, sedangkan untuk sinyal frekuensinya tdak dapat ditampilkan secara fisualisali karena berada di sinyal domain waktu. Dengan keluaran kualitas fidelitas suaranya berbeda stiap filter digitalnya.

## 5.2.Saran

Adapun saran-saran yang penulis ajukan adalah sebagai berikut :

1. Simulasi yang digunakan dengan matlab 7 sangat rentang terhadap gangguan karena seringnya mengolah dan membuat data hususnya data yang dibuat menggunakan GUI/Pemodelan menjadikan program simulasi mengalami error, oleh karena itu untuk menghindarinya, saat mau di *save* data *faile* harus berbeda nama penyimpananya.

2. Untuk pengembanagn lebih lanjut penulis menyarankan agar saat pengukuran alat pada DSP program yang ada didalamnya harus benar-benar lengkap agar dapat dibandingkan dengan simulasinya.

# DAFTAR PUSTAKA


Prijono, Agus dan Marvin Ch. Wijaya. *Pengolahan Citra Digital Menggunakan Matlab Image Processing Toolbox*. Cetakan Pertama. 2007. Bandung: INFORMATIKA.

Hall, Prentice dan Tolmasci. *Advance Communication System*. 2008

Millman dan Halkiits. Elektronika Terpadu, *Rangkaian dan System Analog dan Digital*. Edisi Pertama. 2007. Jakarta: Erlangga.

Modul Praktikum Telekomunikasi II. 2006 Bandung : UNIKOM.

Modul Praktikum Telekomunikasi III. 2006 Bandung : UNIKOM.

http://id.wikipedia.org/wiki/PSD . waktu akses : 15-01-2011. 16:25 WIB.

http://en.wikipedia.org/wiki/Filter_Digital_FIR-shift_keying. Waktu akses: 5-02-2011. 23:00 WIB.

http://en.wikipedia.org/wiki/Frequency-shift_keying. Waktu akses: 5-02-2011. 23:15 WIB.

http://en.wikipedia.org/wiki/Phase-shift_keying#FilterDigitalFir .Waktu akses:

5-02-2011. 23:30 WIB.

http://en.wikipedia.org/wiki/Phase-shift_keying#.28DSP.29. Waktu akses: 5-02-2011. 23:30 WIB.

http://www.diacritech.com/samples/technical/electronics.pdf. Waktu akses: 28-05-2011. 16:30 WIB.

http://www.mathworks.com/products/matlab/. Waktu akses: 25-06-2011. 23:16 WIB.