

# Implementasi Filter FIR secara Real Time pada TMS 32C5402

Oleh:

Tri Budi Santoso, Hary Octavianto, Titon Dutono

E-mail: [tribudi@eepis-its.edu](mailto:tribudi@eepis-its.edu)

Laboratorium Sinyal, Politeknik Elektronika Negeri Surabaya  
Institut Teknologi Sepuluh Nopember Surabaya (ITS)

## Abstrak:

Telah dikembangkan modul praktikum Pengolah Sinyal Digital yang bersifat aplikatif praktis dengan memanfaatkan DSP Card TMS 32C5400. Satu implementasi Filter Digital dengan teknik perancangan FIR sengaja dipilih sebagai topik pada paper ini karena merupakan materi kunci dalam materi Pengolah Sinyal Digital.

Berawal dari perancangan FIR Filter menggunakan alat Bantu perangkat lunak Matlab kemudian koefisien-koefisien respon impulse yang telah diperoleh dimasukkan ke perangkat lunak DSP Starter Kit untuk dilakukan kompilasi. Hasil kompilasi berupa kode hexa selanjutnya di downloadkan ke DSP Card untuk melihat implementasi realnya pada sinyal yang sebenarnya.

Hasil yang diperoleh dari sistem ini menunjukkan karakteristik respon frekuensi yang mendekati dengan rumusan yang ada pada teori.

**Kata Kunci:** DSP Card, filter FIR, respon frekuensi

## 1. Pendahuluan

Satu permasalahan yang sering dijumpai dalam proses pembelajaran mata kuliah Pengolah Sinyal Digital adalah sulitnya mencapai keseimbangan antara penyampaian materi di kelas dan penyajian praktikum menggunakan perangkat keras DSP di laboratorium [1]. Pada penelitian sebelumnya telah dilakukan penyusunan modul berbasis Matlab dan DSP Card TMS32C5400, tetapi belum mampu menyajikan modul yang mendukung penyampaian teori di kelas [2].

Satu contoh model perancangan FIR Filter telah dibahas secara mendetail oleh Losada [3]. Disini disajikan tutorial yang menggambarkan aspek praktis beragam prosedur perancangan FIR filter dan diberi perbandingan kinerjanya satu dengan yang lain.

Satu hal yang baru dalam paper ini adalah bagaimana sebuah filter dirancang dengan menggunakan perangkat lunak Matlab, dicobakan dengan menggunakan TMS32C5400, dan hasilnya dibandingkan dengan perhitungan teoritis.

## 2. Konsep FIR Filter

### 2.1 Filter Ideal

Dalam pembicaraan filter ideal ada satu hal yang harus diikuti yaitu bahwa semua sinyal yang memiliki komponen frekuensi dibawah nilai cut-off harus dilewatkan dan semua komponen frekuensi diatas frekuensi cut-off harus direject.

Respon impulse pada filter ideal dapat dinyatakan sebagai:

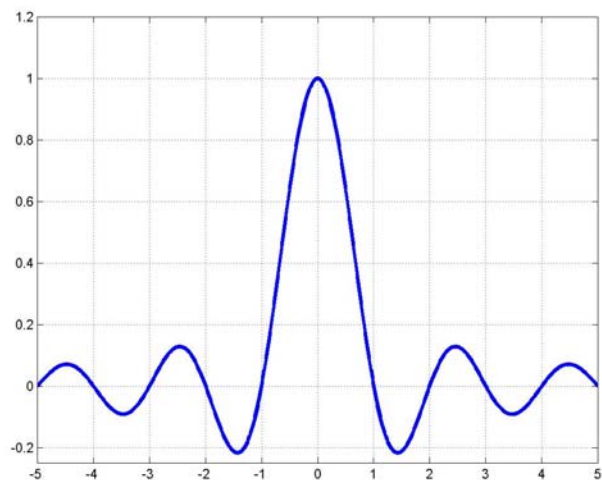
$$h_d[n] = \frac{\sin(\omega_c n)}{\pi n}, \quad -\infty < n < \infty \quad (1)$$

dimana

$\omega_c$  = nilai frekuensi cut off

n = merupakan indek nilai sample waktu diskrit

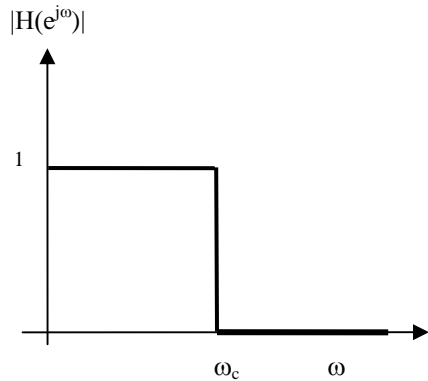
Selanjutnya dikenal sebagai filter infinite impulse response (IIR).



Gambar 1. Respon impulse filter ideal

Filter ini memiliki respon frekuensi seperti berikut:

$$H_{dLP}(e^{j\omega}) = \begin{cases} 1, & 0 \leq \omega \leq \omega_c \\ 0, & \omega_c \leq \omega \leq \pi \end{cases} \quad (2)$$



Gambar 2. Respon frekuensi LPF Ideal

Tetapi filter ideal semacam ini tidak mungkin untuk direalisasikan secara fisis, yang paling mungkin adalah menyusun suatu filter yang memiliki karakteristik mendekati filter ideal tersebut. Cara untuk memperoleh pendekatan adalah dengan memotong respon impulsenya mulai dari titik nol sampai dengan suatu nilai  $N$  tertentu, dengan demikian persamaan (1) diatas harus dimodifikasi menjadi:

$$h(n) = \begin{cases} h_d(n), & 0 \leq n \leq N-1 \\ 0, & n \text{ yang lain} \end{cases} \quad (3)$$

Respon ini merupakan bentuk respon impulse yang selanjutnya dikenal sebagai *finite impulse response* (FIR)

## 2.2 Implementasi Filter FIR

Untuk implementasi FIR filter dalam hardware, kita dapat mengikuti logika diagram blok seperti dalam Gambar 3, output  $\hat{x}[n]$  dapat diperkirakan sebagai:

$$\hat{x}[n] = h^T \cdot x_n \quad (4)$$

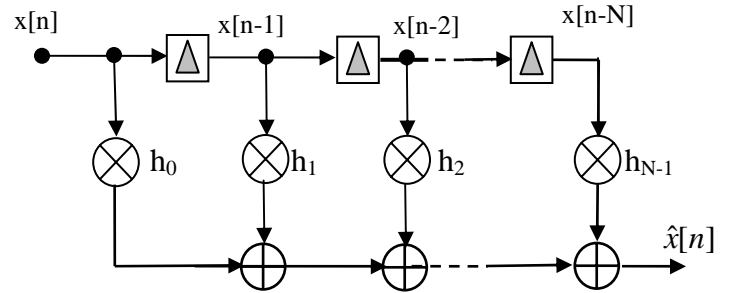
dimana  $h$  tersusun atas koefisien-koefisien filter dan  $x_n$  adalah sampel-sampel sinyal input yang ditahan dalam *tapped delay line* (TDL) pada waktu  $n$ .

Diketahui bahwa kedua parameter tersebut memiliki bentuk seperti berikut:

$$\mathbf{h}^T = [h_0, h_1, \dots, h_{N-1}] \quad (5)$$

$$\mathbf{x}^T = [x[n], x[n-1], \dots, x[n-N+1]] \quad (6)$$

Dalam kasus sederhana, dengan  $h_n=1/N$  untuk  $n = 0$  sampai  $N-1$  nilai, filter mengambil nilai rata-rata pada  $N$  nilai sample pada sinyal  $x[n]$ . Tipe ini selanjutnya dikenal sebagai suatu model operasi *averaging* (perataan) pada *low pass filter*.



Gambar 3. blok diagram FIR filter

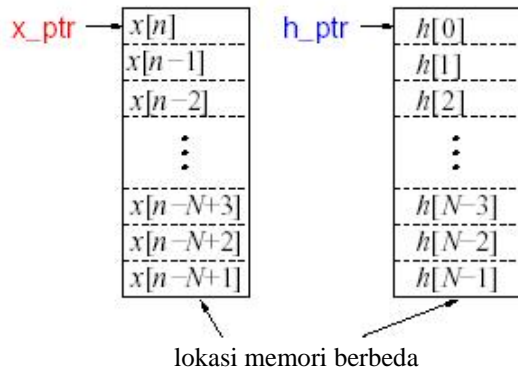
## 2.3. Operasi FIR Filter dalam DSP Card

Operasi yang dilakukan dalam persamaan (5) memerlukan  $N$  multiply-accumulates (MACs), atau lebih mudah diterjemahkan sebagai operasi perkalian dan penjumlahan sebanyak  $N$ . Perlu diketahui bahwa spesifikasi TMS32C5402 yang akan digunakan dalam penelitian ini memiliki kemampuan  $< 300$  juta MAC/detik.

Untuk implementasi suatu filter FIR, dua array pada *tapped delay line* (TDL) sample sebanyak  $x_n$  dan koefisien sebanyak  $h$  diperlukan seperti pada Gambar 2 diatas. Dengan perkalian yang mengacu pada elemen-elemen array dan jumlahan hasil kalinya, satu sample pada  $\hat{x}[n]$  akan dapat dihitung. Array ini harus diberi address sebagai indeks, atau dengan menggunakan pointer seperti ditunjukkan pada Gambar 4.

DSP Card yang akan digunakan ini memiliki sampling rate sebesar 16 kHz. Permasalahan yang akan kita telaah selanjutnya adalah bagaimana mewujudkan suatu filter FIR yang dapat memenuhi kriteria real time. Dengan menggunakan bermacam teknik dimulai dari jumlah taping yang bervariasi dan algoritma yang

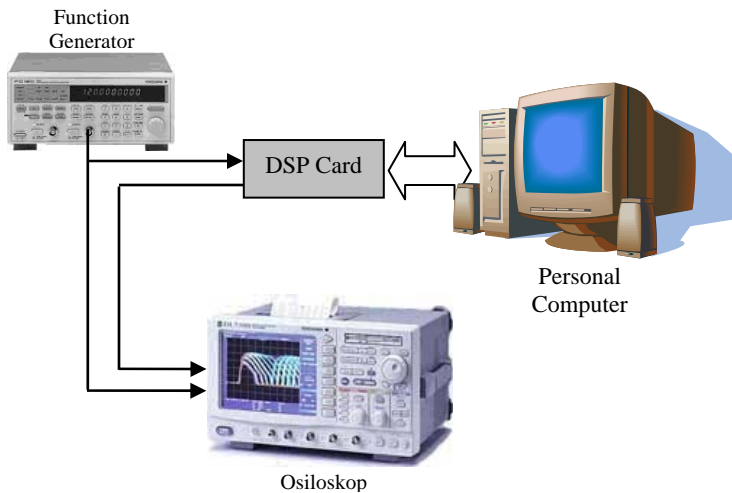
digunakan maka selanjutnya akan diuji kelayakan real time untuk FIR.



Gambar (4) Array pada memory holding tap delay line (TDL) bernilai  $x_n$  dan koefisien  $h$

### 3. Rancangan Sistem

Untuk merealisasikan sebuah FIR filter secara real time diperlukan TMS 32C5402 yang didukung oleh personal computer yang didalamnya sudah terinstall perangkat lunak code composer studio (CCS) untuk mendrive DSP Card tsb. Sebagai sumber sinyal digunakan function generator yang mampu bekerja pada frekuensi suara (300 ~ 4000 Hz), dalam hal ini kita gunakan yang mampu membangkitkan sinyal sinus dan persegi, dengan frekuensi kerja dari DC sampai 2 MHz. Untuk menguji hasilnya kita gunakan sebuah osiloskop yang dapat dilengkapi dengan fasilitas storage system. Lebih jelasnya bisa dilihat seperti pada Gambar 5.

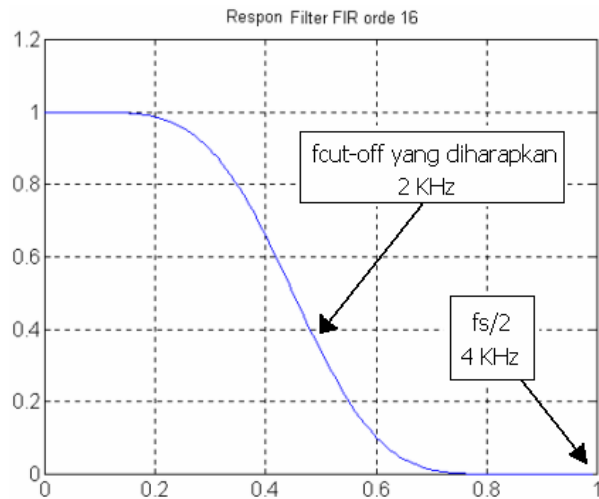


Gambar 5. Gambaran sistem Filter FIR

Proses diawali dengan perancangan dengan perangkat lunak untuk menentukan nilai-nilai koefisien filter. Dalam kasus ini diambil untuk memilih koefisien pada low pass filter. Dari hasil perancangan diperoleh nilai-nilai koefisien filter seperti dibawah.

-0.0002 0.0025 0.0069 -0.0151 -0.0504 0.0380  
 0.2938 0.4492 0.2938 0.0380 -0.0504 -0.0151  
 0.0069 0.0025 -0.0002

Sedangkan respon frekuensi yang dihasilkan seperti pada Gambar 6.



Gambar 6. Respon frekuensi Filter LPF hasil perancangan

Nilai koefisien filter sebanyak 16 titik ini selanjutnya dikopikan secara manual ke CCS yang sudah disiapkan. Proses selanjutnya adalah kompilasi program CCS dan dilanjutkan dengan download nilai hexa hasil kompilasi ke DSP Card. Pada langkah ini program sudah siap dijalankan dan DSP Card siap untuk mengolah sinyal input.

Untuk menguji apakah sistem yang dirancang telah mampu bekerja dengan benar, masukkan sinyal sinus dari *function generator* ke DSP Card, dan amati hasil keluaran dari DSP Card. Bandingkan dengan sinyal asli yang berasal dari *function generator*. Selanjutnya nilai frekuensi sinyal input dinaikkan sedikit demi sedikit dari 0 Hz sampai dengan 6KHz. Apabila anda melihat perubahan nilai level sinyal output sebagai pengaruh kenaikan frekuensi sinyal input, hal ini menunjukkan sistem LPF yang anda rancang telah menunjukkan kinerja yang benar.

#### 4. Karakteristik FIR Filter Hasil Perancangan

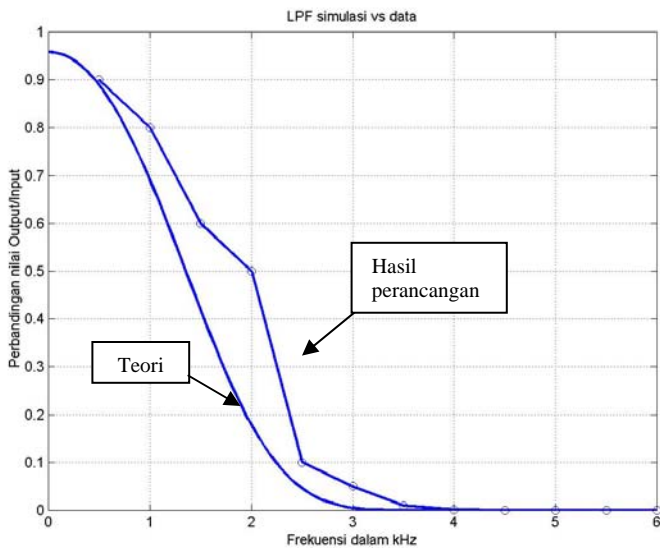
Sesuai dengan perancangan FIR untuk *low pass filter* (LPF), analisa yang disajikan disini juga untuk kasus LPF. Hasil pengujian terhadap sinyal input dan output menunjukkan kinerja seperti yang ditunjukkan pada Tabel 1 berikut ini.

**Tabel 1.** Hasil pengukuran pada LPF

No	V <sub>in</sub> = 50 mVolt		V <sub>out</sub> /V <sub>in</sub>
	Freq (kHz)	V <sub>out</sub> (mVolt)	
1	1	45	0,9
2	1,5	40	0,8
3	2	30	0,6
4	2,5	25	0,5
5	3	5	0,1
6	3,5	2,5	0,05
7	4	0,5	0,01
8	4,5	0,05	0,001
9	5	0,015	0,0003
10	5,5	0,0005	0,0001

Dari data yang diperoleh menunjukkan kenaikan frekuensi menyebabkan sinyal output mengalami penurunan level. Pada nilai frekuensi rendah, pelemahan sinyal relatif kecil, tetapi pada saat melewati nilai 2 kHz pelemahan sinyal input mulai terasa. Pada nilai frekuensi semakin tinggi,  $f \geq 3,5$  kHz menunjukkan pelemahan sinyal yang sangat significant.

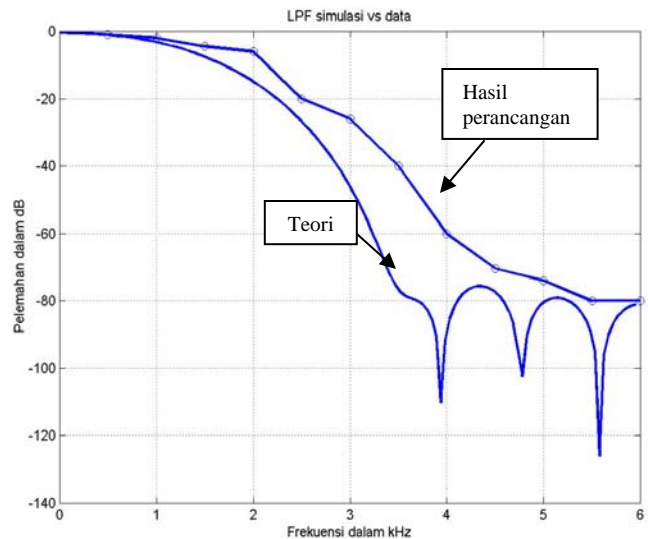
Dengan memanfaatkan data yang diperoleh pada Tabel 1, dapat diperoleh tampilan grafik perbandingan magnitudo sinyal input dan sinyal output seperti pada Gambar 7 berikut ini.



**Gambar 7.** Perbandingan magnitudo sinyal input-output hasil percobaan dengan teori

Gambar 7 memberikan ilustrasi perbandingan nilai magnitudo sinyal output dengan sinyal input ditunjukkan dengan garis dengan pointer. Sementara perbandingan menurut perhitungan teori ditunjukkan dengan garis biasa. Dari grafik tersebut menunjukkan dalam bentuk perbandingan magnitudo sinyal output-input dari hasil percobaan sudah mendekati hasil perhitungan teori.

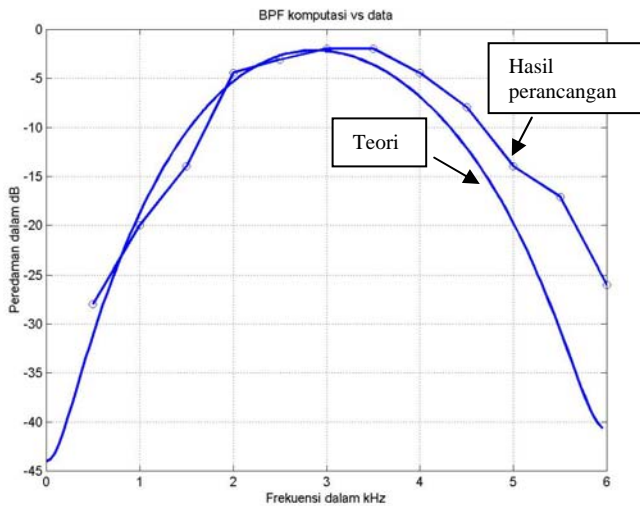
Untuk lebih jelasnya perbandingan antara perhitungan teori dengan hasil percobaan low pass filter menggunakan sistem FIR yang dirancang dengan TMS diberikan dalam bentuk skala logaritmik, atau yang juga dikenal sebagai respon frekuensi filter seperti pada Gambar 8 berikut ini.



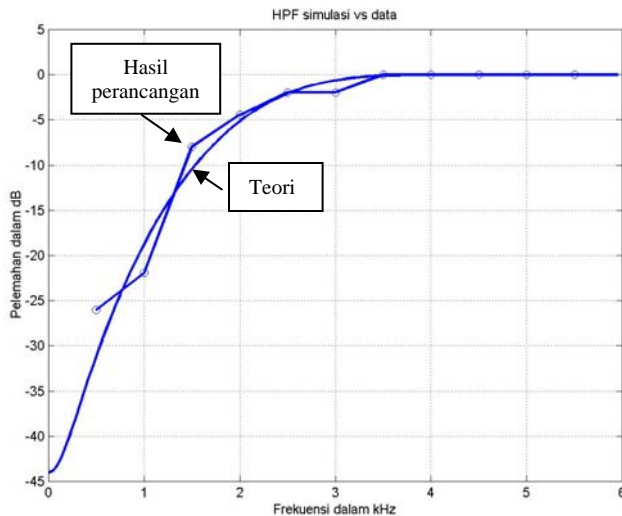
**Gambar 8.** Perbandingan respon frekuensi hasil percobaan dengan teori

Dari Gambar 8 menunjukkan hasil percobaan menggunakan sistem yang dirancang memiliki penyimpangan dibandingkan dengan karakteristik respon frekuensi menurut perhitungan teori. Tetapi nilai perbedaan yang terjadi tidak terlalu besar, sehingga sistem FIR untuk low pass filter yang dirancang menggunakan TMS32C5402 masih relatif menunjukkan kinerja yang cukup bagus dan cukup mewakili.

Untuk kasus FIR lain yang telah dilakukan pada *band pass filter* (BPF) dan *high pass filter* (HPF) ditunjukkan pada Gambar 9 dan Gambar 10.



**Gambar 9.** Perbandingan respon frekuensi hasil perancangan dan teori, pada sistem BPF



**Gambar 10.** Perbandingan respon frekuensi hasil perancangan dan teori, pada sistem HPF

## 5. Kesimpulan

Dari hasil yang diperoleh dalam pengujian sistem diatas menunjukkan sistem FIR untuk low pass filter (LPF), band pass filter (BPF), dan high pass filter (HPF) cukup mewakili sistem ideal dan telah menunjukkan kinerja yang bagus.

Tetapi satu hal yang perlu ditingkatkan disini adalah perlunya untuk melakukan pengujian lebih banyak lagi dengan beragam sinyal input yang berbeda, seperti sinyal suara dan sinyal audio yang lainnya seperti musik. Untuk mendapatkan hasil yang lebih baik, juga diperlukan untuk melakukan

pengujian dengan berbagai jumlah tapping yang bervariasi, sehingga diperoleh nilai yang optimal untuk perancangan sebuah filter FIR.

## Referensi:

- [1.] Thompson Michel, Burleigh Jojan, “*Matlab/DSK GUI for FIR Filter Design Using a TMS320C3X DSK*”, Baylor University, Texas, USA 2000.
- [2.] Tri Budi Santoso, Hary Oktavianto, “*Pengembangan Materi Praktikum Pengolah Sinyal Digital berbasis Matlab dan TMS 32C5400 DSK*”, Proceedings SNTE 2003, UGM, 25 September 2003.
- [3.] Ricardo A Losada, “*Practical FIR Filter Design in Matlab*”, The Math Works, Inc., 3 Apple Hill Dr., USA, March 2003.

## Bio Data Penulis



### Tri Budi Santoso

Pada tahun 1994 menyelesaikan program S1 di Jurusan Teknik Fisika-Program Studi Energi dan Teknik Kondisi Lingkungan. Pada tahun 1999 menyelesaikan program S2 di Teknik Elektro-Telekomunikasi. Aktifitas riset yang sedang digelutinya adalah *Speech Processing*.

### Hary Octavianto

Menyelesaikan pendidikan S1 Jurusan Teknik Elektro, program Studi Elektronika pada tahun 2001. Bergabung dengan Politeknik Elektronika Negeri Surabaya, sejak tahun 2001. Saat ini bekerja sebagai staf pengajar di Jurusan Elektronika dan riset yang digelutinya adalah *Embedded System for Digital Signal Processing*.