

## PERANCANGAN DAN REALISASI FIR FILTER UNTUK INTER SATELLITE LINKS (ISL) PADA FREKUENSI 50-90 MHZ DENGAN MENGGUNAKAN FPGA

“DESIGN and REALIZATION of FIR FILTER FOR INTER SATELLITE LINKS (ISL) AT FREQUENCY 50-90 MHZ USING FPGA”

Lutfi Jamil Setiawan<sup>1</sup>, Dr. Heroe Wijanto, Ir., MT.<sup>2</sup>, Dr.Ir. Yuyu Wahyu, MT<sup>3</sup>

<sup>1,2,3</sup>Prodi S1 Teknik Telekomunikasi, Fakultas Teknik, Universitas Telkom

[lutfijamilsetiawan@gmail.com](mailto:lutfijamilsetiawan@gmail.com), [heroewijanto@gmail.com](mailto:heroewijanto@gmail.com), [yuyuwahyusr@yahoo.com](mailto:yuyuwahyusr@yahoo.com)

### ABSTRAK

Pada Tugas Akhir ini telah didesain filter FIR dengan metode *equiripple* yang nantinya akan melewati *bandwidth* sebesar 40 Mhz pada frekuensi 50-90 Mhz dan direalisasikan pada FPGA. Filter FIR ini diimplementasikan untuk digunakan pada ISL (*Inter Satellite Links*). Metode filter FIR dipilih dengan alasan komunikasi yang dilakukan pada ISL adalah komunikasi data. Pada komunikasi data yang dibutuhkan adalah respon fasa yang linear, sedangkan filter yang bisa menjamin respon fasanya linear adalah filter FIR. Metode *equiripple* dipilih karena metode ini adalah metode yang paling handal atau paling optimal karena melakukan pendekatan terhadap besarnya error antara respon frekuensi yang diinginkan dengan respon frekuensi sesungguhnya yang menyebar merata pada *passband* dan *stopband* sehingga dapat meminimalkan besarnya error. Perancangan filter FIR ini dilakukan dengan menggunakan program simulasi matlab R2012b. Perealisasian menggunakan FPGA dengan VHDL sebagai bahasa pemrogramannya dengan bantuan Xilinx ISE Design Suite 14.5. Pada perancangan filter FIR dibutuhkan orde sebesar 272. Dalam pengimplementasian filter FIR digital dalam FPGA GENESYS Xilinx Virtex-5 XC5VLX50T dibutuhkan resource FPGA yaitu : jumlah *slice register* 13 %, jumlah *slice LUT* 39 %, jumlah *bonded IOB* 3%, jumlah BUFG/BUFGCTRLs 9%, jumlah PLL\_ADV 16 %, dan Jumlah DSP48 93%.

**Kata Kunci :** *Filter FIR, equiripple, FPGA, VHDL*

### ABSTRACT

In this final project, the FIR Filter has been designed with *equiripple* method which will occur over a bandwidth of 40 MHz at a frequency of 50-90 MHz and realized on FPGA. This FIR filter is implemented to be used on the ISL (*Inter Satellite Links*). This filter method been selected because the communication that conducted on ISL is a data communication. Data communication required a linear phase response, whereas the filter that can guarantee a linear phase response is a FIR filter. *Equiripple* method is taken because this method is the most reliable and the most optimal due to approaching the *magnitude* of errors between the desired frequency responses to the actual frequency responses which spreads evenly in the *passband* and *stopband* so it can minimize the *magnitude* of errors. The design of this FIR filter is conducted with theoretical calculations and using R2012b matlab simulation program. In order to implement it, used FPGA with a VHDL as the programming language with a help of Xilinx ISE Design Suite 14.5. The design of FIR filter orde is 272. FIR filter digital implemented in FPGA GENESYS Xilinx Virtex-5 XC5VLX50T. resource FPGA : number of *slice register* 13%, number of *slice LUT* 39%, Number of *bonded IOB* 3%, number of BUFG/BUFGCTRLs 9%, number of PLL\_ADV 16 %, and number of DSP48 93%.

**Keywords:** *FIR filter, equiripple, FPGA, VHDL*

### 1. Pendahuluan

Satelit adalah suatu benda yang mengorbit mengelilingi bumi yang biasanya digunakan oleh manusia sebagai alat komunikasi, pemantau cuaca, dsb. Seiring perkembangan zaman satelit terus berkembang sehingga muncul teknologi *Inter satellite links* (ISL) yang dapat menghubungkan satelit secara langsung dengan satelit lain.

*Inter satellite links* (ISL) merupakan solusi dari permasalahan yang ada pada satelit LEO pemantau bumi. Satelit ini hanya bisa mengirim gambar ketika posisinya diatas stasiun bumi. Hal ini menyebabkan *delay* untuk mendapatkan informasi. Oleh sebab itu dibuatlah stasiun bumi dengan jumlah yang banyak agar dapat terus memantau satelit LEO. Namul hal ini membutuhkan biaya yang cukup besar, sehingga ISL yang dapat membuat satelit bisa berhubungan secara langsung dengan satelit lain dinilai lebih cepat dan efisien.

Filter merupakan salah satu komponen yang digunakan dalam *inter satellite links* (ISL). Filter ini digunakan untuk melewati data atau sinyal yang diinginkan. Dengan cara melewati frekuensi sinyal digital yang berisi data yg dibutuhkan dan meredam frekuensi selain itu.

Pada Tugas Akhir ini dirancang sebuah filter FIR yang akan melewatkan frekuensi dalam rentang 50-90 Mhz. Dengan metode *equiripple*, optimal *chebyshev*, atau *minimax* metode agar hasil yang didapat sesuai dengan spesifikasi yang telah ditentukan

**2. Filter FIR**

Filter adalah suatu alat atau komponen pada sistem komunikasi yang digunakan untuk melewatkan frekuensi yang diinginkan dan meredam frekuensi yang tidak diinginkan [1]. *Band Pass Filter* adalah salah satu dari tipe filter yang memiliki sifat melewatkan frekuensi pada rentang daerah tertentu (*passband*), dan meredam frekuensi diatas dan dibawah daerah rentang frekuensi tersebut (*stopband*) [1].

Filter FIR adalah salah satu jenis tipe filter yang memiliki sifat *non rekursif* dimana keluaran saat ini tidak bergantung pada keluaran sebelumnya, sehingga mempunyai respon *impulse* yang terbatas [1]. Hal ini membuat keluaran filter dihitung sebagai nilai penjumlahan angka berhingga dari masa lalu, sekarang, dan mungkin masa depan nilai-nilai masukan filter. Yang dirumuskan dalam bentuk [1]:

$$y(n) = \sum_{k=0}^{M-1} x(n-k)h(k) = \sum_{k=0}^{M-1} h(k) \cdot x(n-k) \tag{I}$$

Filter FIR memiliki fungsi transfer sebagai berikut [2]:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M-1} h(k)z^{-k}}{1 + \sum_{k=1}^{M-1} a_k z^{-k}}$$

Bentuk respon Impuls filter ideal [3]:

$$\text{BPF} : h(n) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} e^{j\omega n} d\omega + \int_{\omega_s}^{\omega_p} e^{j\omega n} d\omega = \frac{1}{\pi} \text{sinc}(\omega_c n) - \frac{1}{\pi} \text{sinc}(\omega_s n) \tag{II}$$

**2.1 Equiripple Filter**

Pada dasarnya metode frekuensi *sampling* dan metode *windowing* adalah metode yang paling sederhana untuk mendesain filter FIR. Namun kedua metode itu juga memiliki kelemahan dalam kurangnya kehandalan untuk mengatur frekuensi kritis seperti  $\omega_p$  dan  $\omega_s$ .

*Equiripple* sendiri adalah suatu metode yang memberikan optimasi pada desain filter FIR, dengan melakukan pendekatan terhadap besarnya error antara respon frekuensi yang diinginkan dengan respon frekuensi sesungguhnya yang menyebar merata pada *passband* dan *stopband* sehingga dapat meminimalkan besarnya error. Pendekatan besarnya error dapat dirumuskan menjadi [2]:

$$E(\omega) = |Q(\omega)| [H(\omega) - D(\omega)] \tag{III}$$

Dimana fungsi *weighting* dan respon frekuensi yang diinginkan adalah [2]:

$$|Q(\omega)| = |W(\omega)| Q(\omega) \tag{IV}$$

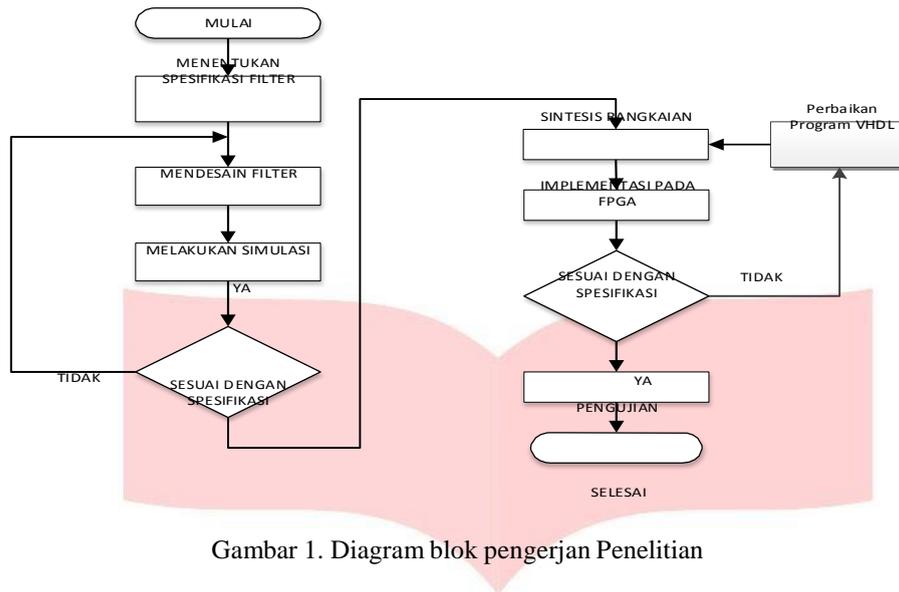
$$\hat{H}(\omega) = \frac{H_d(\omega)}{Q(\omega)} \tag{V}$$

Dari fungsi error  $E(\omega)$ , masalah dari metode pendekatan *chebyshev* (*mini-max*) adalah untuk mencari parameter filter  $\alpha_k$  yang meminimalkan maksimal error dari *band* frekuensi yg diinginkan atau dapat dirumuskan dengan [2]:

$$\min_{\alpha_k} [\max_{\omega \in \Omega_p} |E(\omega)|] = \min_{\alpha_k} [\max_{\omega \in \Omega_p} \{ |Q(\omega)| [H(\omega) - \sum_{k=0}^{M-1} \alpha_k \cos \omega k] | \}] \tag{VI}$$

**3. Desain dan Realisasi Sistem**

Berikut adalah diagram blok dari perancangan filter FIR *equiripple* :



Gambar 1. Diagram blok pengerjaan Penelitian

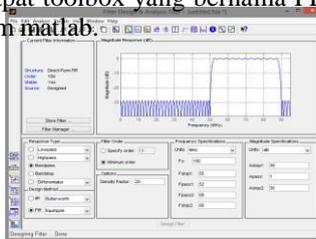
3.1 Perancangan Filter equiripple

3.1.1 Perancangan dengan Bantuan Matlab

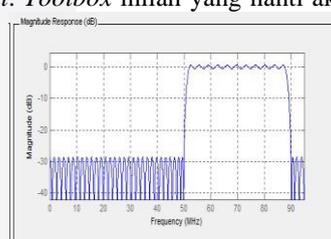
Dalam perancangan filter FIR dengan bantuan matlab kita harus mendefinisikan spesifikasi filter terlebih dahulu.:

- Frekuensi cutoff 1 : 52 MHz
- Frekuensi Stopband 1 : 50 MHz
- Frekuensi cutoff 2 : 88 MHz
- Frekuensi Stopband 2 : 90 MHz
- Ripple : 0,1 dB
- Redaman Stopband : 60 dB
- Frekuensi sampling : 200 Mhz

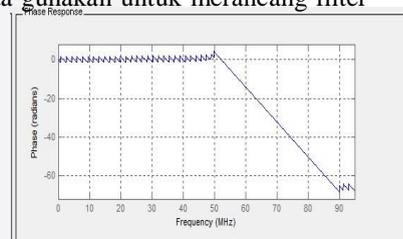
Setelah semuanya telah ditentukan, kita bisa masuk ke perancangan dengan bantuan matlab. Pada matlab terdapat toolbox yang bernama *FDA tool*. *Toolbox* inilah yang nanti akan kita gunakan untuk merancang filter dalam matlab.



Gambar 2 FDA Toolbox



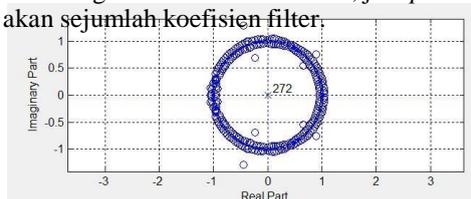
Gambar 3 respon magnitude



Gambar 4 Respon Fasa

*FDA tool* nantinya bisa menunjukkan respon *magnitude*, respon fasa lalu letak *pole* dan *zero*. Pembuktian bahwa respon fasa dari filter yang dirancang linear, dan linearnya terdapat pada respon filter adalah misi utama yang dilakukan dengan *FDA tool*

Letak *pole* dan *zero* untuk membuktikan kestabilan dari filter ini. Filter Fir tidak memiliki *zero*, karena tidak terdapat koefisien *feedback* didalamnya. Filter FIR memiliki koefisien *feed forward*. Inilah kenapa filter FIR bisa stabil. Apabila *pole* terdapat dalam lingkaran maka filter stabil, jika *pole* terdapat diluar lingkaran maka filter tidak stabil. Jumlah *pole* nantinya akan sejumlah koefisien filter.



Gambar 5. Pole dan Zero

4. Analisis dan Pengujian Hasil Simulasi

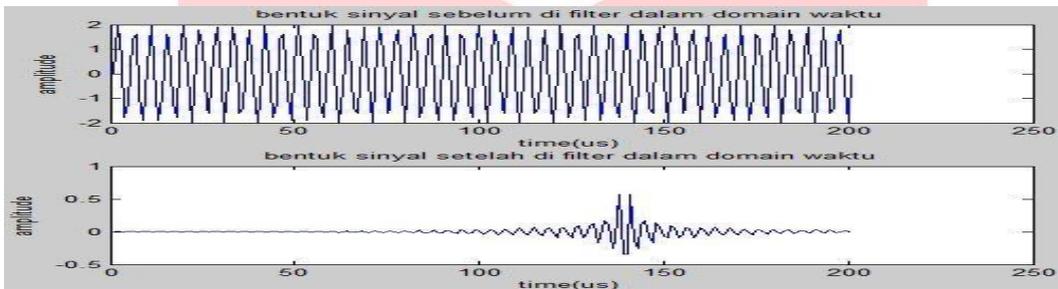
4.1 Analisis dan Pengujian Filter Menggunakan Matlab

Pada pengujian menggunakan matlab akan dilakukan simulasi dengan skala 1 : 1.000.000. Proses pensimulasiannya pun terangkum dalam gambar berikut ini :



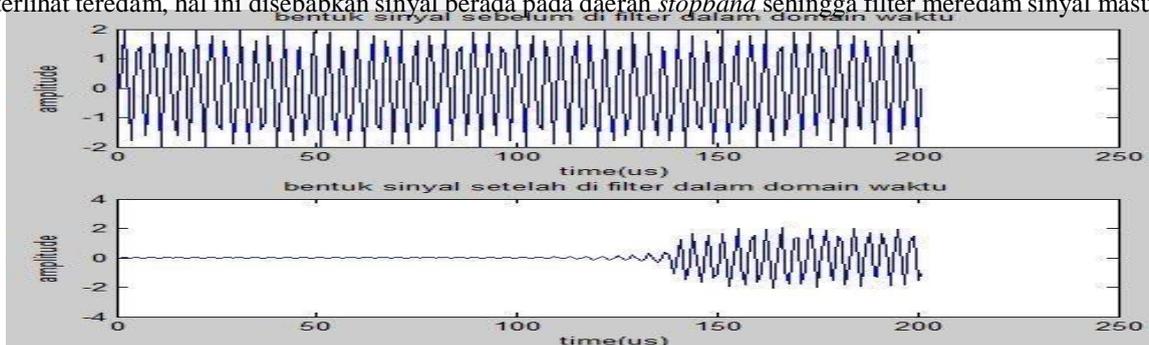
**Gambar 6 Skenario Pengujian Dengan Simulasi Matlab**

Pada simulasi pengujian filter FIR dalam matlab ini akan dilewatkan sinyal sinusoidal dengan frekuensi 45 Mhz, 50 Mhz, 60 Mhz, 65 Mhz,70 Mhz, 75 Mhz, 80 Mhz,85 Mhz, 90 Mhz, dan 95 Mhz. Sinyal-sinyal sinusoidal dengan frekuensi yang disebutkan diatas kemudian akan dilewatkan melewati filter FIR digital yang telah diskalakan 1 : 1.000.000. Kemudian dilihat hasilnya dengan memplot hasil sinyal sinusoidal keluaran filter pada matlab agar dapat bisa dilihat.



**Gambar 7 Sinyal Sinusoidal dengan frekuensi 45 Mhz**

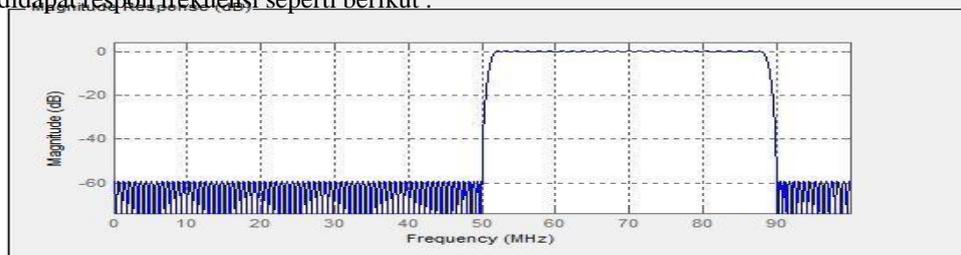
Gambar 7 merupakan gambaran sinyal masuk dan sinyal keluar pada frekuensi 45 Mhz. Sinya keluar terlihat teredam, hal ini disebabkan sinyal berada pada daerah *stopband* sehingga filter meredam sinyal masuk.



**Gambar 8 Sinyal Sinusoidal sebelum di Filter pada frekuensi 55**

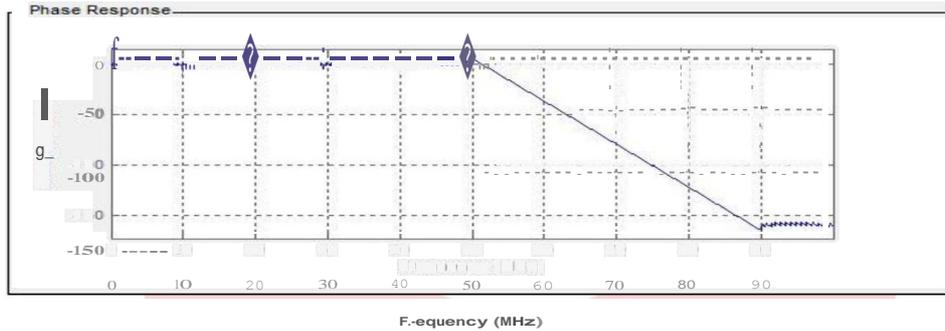
Gambar 8 merupakan gambar sinyal masuk dan sinyal keluar pada frekuensi 55 Mhz. Karena sinyal terdapat pada daerah *passband* maka sinyal akan dilewatkan. Namun terlihat terdapat *delay* sebesar 130 us sebelum sinyal bisa dibaca oleh siste. Hal ini disebabkan oleh element *delay* yang terdapat pada filter.

Setelah melakukan beberapa percobaan melewati sinyal sinusoidal pada filter digital yang telah dirancang didapat respon frekuensi seperti berikut :



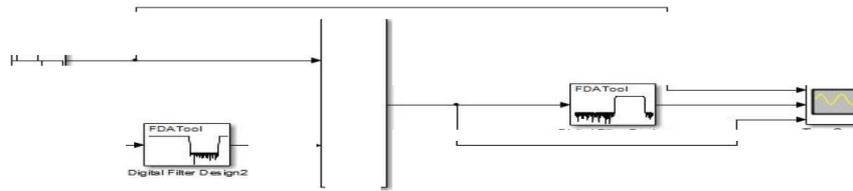
**Gambar 9 Respon Magnitude FIR Filter**

Sedangkan respon fasanya :



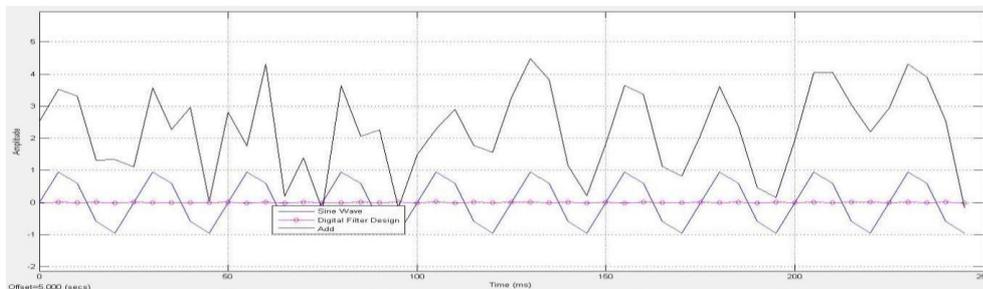
Gambar 9 Respon Fasa FIR Filter

4.1.2 Analisis pengujian simulasi filter pad Simulink



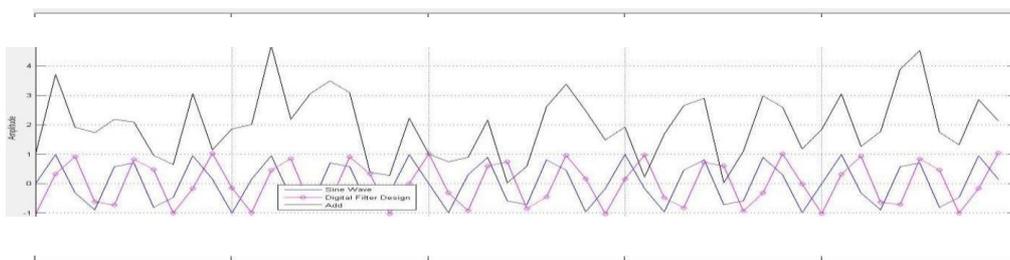
Gambar 12 Blok Rancangan di Simulink

Pada gambar blok diagram 12 dilakukan uji coba dengan dua masukan sinyal sinusoidal. Sinyal yang berasal dari DSP *sine wave* merupakan sinyal informasi, sedangkan sinyal yang berasal dari *random source* adalah sinyal *noise*. Digital filter design2 merupakan *Band Stop Filter* yang berfungsi untuk memfilter sinyal yang berasal dari *random source*. Sehingga sinyal yang dilewatkan merupakan sinyal *noise* yang memiliki frekuensi untuk tidak dilewatkan. *Add* berfungsi sebagai modulator untuk menggabungkan kedua sinyal informasi dan sinyal *noise*. Setelah itu sinyal keluaran *add* dimasukan ke filter FIR yang telah dirancang. *Time scope* pada diagram ini berfungsi untuk melihat 3 hal. Yaitu sinyal informassi, sinyal keluaran *adder* sebelum difilter dan sinyal keluaran filter.



Gambar 13 hasil Simulink sinyal frekuensi 45 Mhz

Pada gambar 13 DSP *sine wave* mengeluarkan sinyal sinusoidal berfrekuensi sebesar 45 Mhz dengan amplitudo sama dengan 1. *Random source* akan mengeluarkan sinyal tak tentu dengan amplitudo dari 0-4 dengan frekuensi diantara 0-50 Mhz dan > 90 Mhz. Terlihat bahwa pada grafik garis berwarna hitam hasil penggabungan sinyal sinusoidal yang akan difilter. Garis grafik berwarna ungu adalah sinyal sinusoidal yang akan dilewatkan filter, terlihat filter tidak melewatkan atau meredam sinyal sinusoidal. Hal ini sesuai karena sinyal frekuensi sebelum 52 Mhz akan diredam.



**Gambar 14 hasil Simulink sinyal frekuensi 55Mhz**

Pada gambar 14 DSP *sine wave* mengeluarkan sinyal sinusoidal berfrekuensi sebesar 55 Mhz dengan amplitudo sama dengan 1. *Random source* akan mengeluarkan sinyal tak tentu dengan amplitudo dari 0-4 dengan frekuensi diantara 0-50 Mhz dan > 90 Mhz. Terlihat bahwa pada grafik garis berwarna hitam hasil penggabungan sinyal sinusoidal yang akan difilter. Garis grafik berwarna ungu adalah sinyal sinusoidal yang akan dilewatkan filter, terlihat filter melewatkan sebuah sinyal sinusoidal yang memiliki amplitudo yang sama dengan sinyal informasi namun memiliki bentuk yang sedikit berbeda. Hal ini karena terjadi *delay* sebesar 130 us pada filter FIR yang berasal dari komponen pembentuk filter FIR.

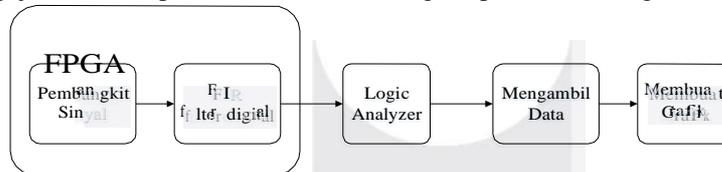
**Tabel 1 Resource yang digunakan FPGA untuk implementasi**

Device Utilization Summary (estimated values)			[-]
Logic Utilization	Used	Available	Utilization
Number of Slice Registers	3860	28800	13%
Number of Slice LUTs	11382	28800	39%
Number of fully used LUT-FF pairs	691	14551	4%
Number of bonded IOBs	19	480	3%
Number of Block RAM/FIFO	1	60	1%
Number of BUFG/BUFGCTRLs	2	32	6%
Number of DSP48Es	45	48	93%
Number of PLL_ADVs	1	6	16%
Average Fanout of Non-Clock Nets	2.73		

Dari hasil sintesis dan implementasi tersebut dapat disimpulkan bahwa filter digital yang telah dirancang dapat diimplementasikan pada perangkat FPGA dengan cukup baik karena tidak ada *resource* FPGA yang dipakai hingga 100%.

**4.2 Pengujian hasil implementasi filter**

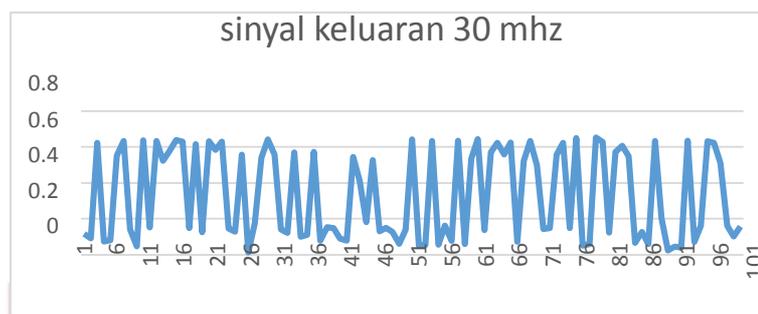
Pada pengujian hasil implementasi filter FIR digital pada FPGA digunakan skenario seperti gambar dibawah ini :



**Gambar 15 skenario pengujian FIR filter pada FPGA**

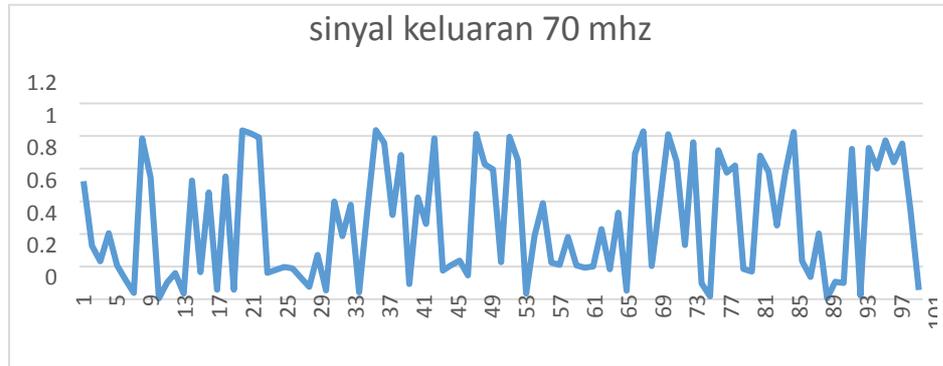
Kemampuan FPGA untuk membangkitkan sinyal dimanfaatkan untuk pengujian hasil implementasi filter FIR pada kali ini. Keluaran filter FIR digital, dapat dilihat menggunakan *logic analyzer*. Lalu setelah data dilihat pada *logic analyzer*, datanya akan dikumpulkan kedalam tabel. Kemudian data dalam tabel itu kemudian dikonversi menjadi decimal. Sehingga dapat digambarkan hasil keluaran filter digital. Kemudian hasil keluaran filter digital dikumpulkan dan akan dibuat grafik respon *magnitude* dari data-data tersebut.

Data yang diambil dari *logic analyzer* hanya sebesar 100 kali sampling saja. Hal ini dirasa sudah cukup menggambarkan hasil keluaran sinyal dari filter digital. Dari pengujian tersebut didapatkan gelombang keluaran menjadi sebagai berikut :



**Gambar 16 Sinyal Hasil Keluaran Filter dengan frekuensi 30 MHz**

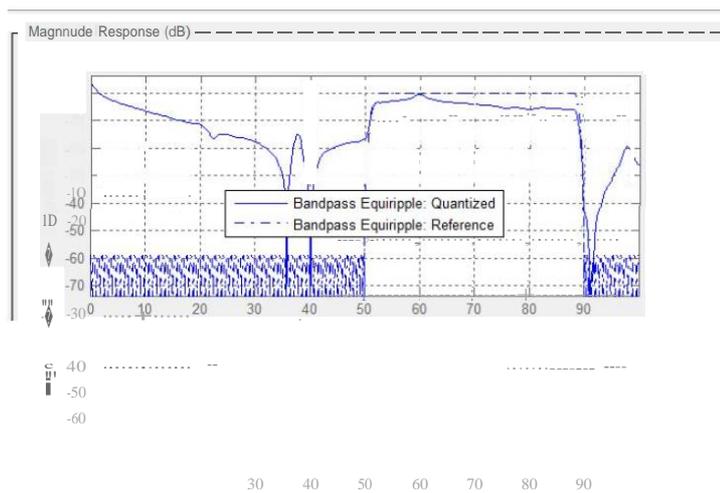
Pada gambar 16 bisa dilihat bahwa sinyal hasil keluaran filter dengan frekuensi 30 Mhz atau yang dibawah *band pass* memiliki bentuk sinyal yang teredam dan tidak membentuk gelombang sinusoidal hal ini disebabkan oleh sinyal pada frekuensi tersebut diredam oleh sistem sehingga membuat bentuk sinyal menjadi rusak.



**Gambar 17 Sinyal Hasil Keluaran Filter dengan frekuensi 70 MHz**

Pada gambar 17 bisa dilihat bahwa sinyal hasil keluaran filter dengan frekuensi 70 MHz membentuk sebuah gelombang sinusoidal. Hal ini terjadi karena pada frekuensi bandpaas sinyal diloloskan oleh sistem yang membuat bentuk sinyal terlihat menyerupai gelombang sinusoidal.

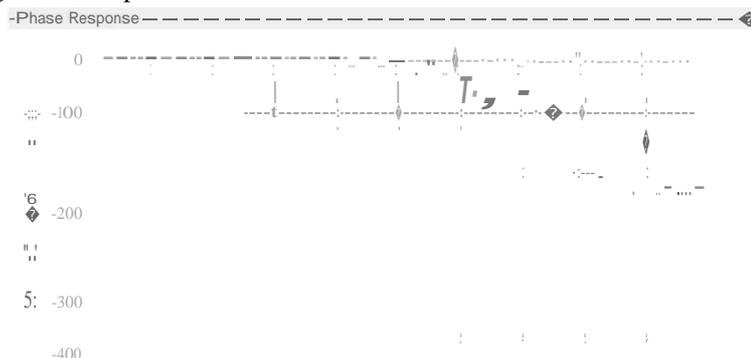
Dari beberapa hasil keluaran sinyal FIR filter digital yang diberasal dari sinyal dengan frekuensi 30 Mhz, 40 Mhz, 50 Mhz , 60 Mhz,70 Mhz,80 Mhz,90 Mhz, dan 100 Mhz didapatkan respon frekuensi sebagai berikut :

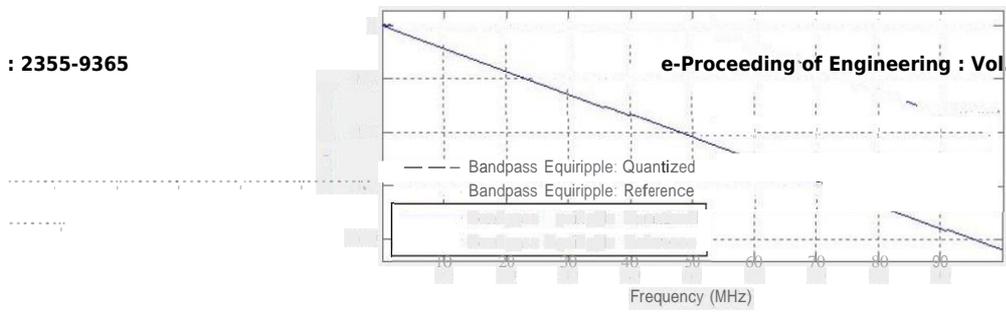


**Gambar 18 Respon Magnitude Filter FIR**

Pada gambar 18 terdapat dua garis. Garis putus-putus merupakan hasil simulasi dan garis tebal merupakan hasil implementasi. Dari gambar dapat disimpulkan bahwa respon frekuensi hasil implementasi filter hasilnya sudah cukup baik hal ini bisa dinyatakan karena respon frekuensi yang diinginkan hampir menyerupai dengan spesifikasi awal meskipun *bandwidth* filter melebar 10 Mhz menjadi 40-90 Mhz

Dibandingkan dengan *bandwidth* hasil simulasi, *bandwidth* hasil implementasi melebar menjadi 40-90 Mhz. Hal ini terjadi disebabkan oleh beberapa faktor, yaitu kecilnya frekuensi *sampling* yang digunakan sehingga mengakibatkan orde yang tidak cukup untuk membuat *bandwidth* filter dari 50-90 Mhz





**Gambar 19 Respon Fasa Filter FIR**

Pada gambar 19 terdapat dua garis. Garis putus-putus merupakan hasil simulasi dan garis tebal merupakan hasil implementasi. Dapat dilihat pada hasil simulasi dan implementasi bahwa respon fasa sistem cukup linear

pada rentang frekuensi yang diinginkan pada 50-90 Mhz, hal ini membuktikan bahwa filter FIR memang mampu membuat respon fasa linear dan cocok untuk komunikasi data

## 5 Kesimpulan

Setelah melakukan simulasi dan pengujian pada filter FIR digital yang telah diimplementasikan pada FPGA GENESYS Xilinx Virtex-5 XC5VLX50T. Didapatkan kesimpulan bahwa :

1. Dari hasil simulasi pada matlab dan simulink menunjukkan bahwa filter FIR digital yang didesain dengan metode *equiripple* dapat dengan baik di implementasikan pada frekuensi ISL yaitu pada frekuensi 50 Mhz - 90 Mhz dengan *stopband* sebesar 60 db.
2. Dalam pengimplementasian filter FIR digital dalam FPGA GENESYS Xilinx Virtex-5 XC5VLX50T. Dibutuhkan *resource* FPGA yaitu : jumlah *slice register* 13%, jumlah *slice LUT* 39%, jumlah *bonded IOB* 3%, jumlah *BUFG/BUFGCTRLs* 9%, jumlah *PLL\_ADV* 16%, dan Jumlah *DSP* 48 93%. Dari hal ini dapat disimpulkan filter FIR dapat diimplementasikan di FPGA GENESYS Xilinx Virtex-5 XC5VLX50T dengan baik
3. Pada saat mengimplementasikan filter FIR digital pada FPGA GENESYS Xilinx Virtex-5 XC5VLX50T. Respon *magnitudenya* berubah dan daerah *bandwidth* melebar menjadi 40-90 Mhz. Hal ini disebabkan oleh terbatasnya frekuensi *sampling* yang bisa diimplementasikan pada FPGA.
4. Respon fasa yang dimiliki sistem cukup linear sehingga filter FIR memang terbukti memiliki respon fasa yang linear dan bisa dikategorikan baik untuk melewati komunikasi data.

## Daftar Pustaka

- [1] A. m. Stuart lawson, Wave Digital Filter, 1990.
- [2] A. D. W. F. E. Puns kaya, Particle filtering methods for Digital Communications, in The Telecom Handbook., 2006.
- [3] I. p. s. digital, MODUL PRAKTIKUM PENGOLAHAN SINYAL DIGITAL, Institut Teknologi Telkom, 2013.
- [4] R. A. Losada, Digital Filter with Matlab, the mathwork inc, 2008.
- [5] R. G. Lyons, Understanding Digital Signal Processing, 2011.
- [6] M. A. Elbani, Desain dan implementasi FIR filter equiripple pada frekuensi 1-30 Mhz dengan menggunakan FPGA., 2012: Institut Teknologi Telkom, bandung.