# IMPLEMENTASI ALGORITME *HIGH PASS FILTER* PADA FPGA MENGGUNAKAN PROSESOR NIOS II

*Kunnu Purwanto [1)], Agus Bejo [2)], Addin Suwastono [3)]*

*[1),2),3)] Departemen Teknik Elektro dan Teknologi Informasi Fakultas Teknik Universitas Gadjah Mada, Jln. Grafika No.2 Mlati, Sleman, Yogyakarta 55581*
*Email : kunnu_p19@yahoo.co.id*

**Abstrak -** *Image processing adalah teknik pengolahan citra yang aplikasinya banyak digunakan pada teknologi digital di segala bidang seperti militer, medis, pertanian, robotika, seni dan lainnya. Salah satu teknik image processing yang banyak digunakan adalah teknik penapisan atau filtering. High Pass Filter adalah salah satu contoh jenis filter yang digunakan untuk menapis sinyal dengan frekuensi tinggi. Algoritme High Pass Filter dapat diimplementasikan di berbagai platform perangkat keras baik komputer maupun embedded system. Implementasi High Pass Filter pada embedded system menghasilkan beban komputasi yang relatif berat sehingga membutuhkan waktu eksekusi yang lama karena karakteristik embedded system yang memiliki frekuensi clock rendah dan resource yang terbatas. Untuk mengatasi masalah tersebut maka dalam penelitian ini dirancang sebuah implementasi High Pass Filter dengan menggunakan FPGA dengan tujuan agar dapat meningkatkan komputasi dan mempercepat waktu eksekusi. Implementasi High Pass Filter ditulis dalam kode verilog kemudian disintesis menggunakan tool Quartus lalu diverifikasi pada FPGA Altera Cyclone IV EP4CE22F17C6N. Pengujian dilakukan dengan memverifikasi fungsi modul High Pass Filter dengan bantuan Nios II sebagai host prosesor. Hasil pengujian menunjukkan bahwa komputasi High Pass Filter pada FPGA dapat ditingkatkan menjadi 2.9 kali lebih cepat dibandingkan dengan ketika dijalankan diatas Nios II prosesor saja dengan kompensasi kenaikan jumlah Logic Element (LE) sebesar 22.4%.*

***Kata kunci****: high pass filter, FPGA, Nios II, image processing*

## 1. Pendahuluan

### 1.1. Latar Belakang

*Image processing* adalah teknik pengolahan citra yang aplikasinya banyak digunakan pada teknologi digital di segala bidang seperti militer, medis, pertanian, robotika, seni dan lainnya. *Image processing* merupakan teknik mengolah suatu *input* berupa gambar (*image*) dan mentransformasikan menjadi gambar lain agar keluarannya memiliki kualitas yang lebih baik dibandingkan kualitas gambar masukannya (*image input*).

Salah satu teknik *image processing* yang banyak digunakan adalah teknik penapisan atau *filtering*. Teknik *filtering* sangat bermanfaat diantaranya adalah untuk meningkatkan kualitas dan ketajaman, menghilangkan *noise* dan cacat pada suatu objek *image*.

*High Pass Filter* (HPF) adalah salah satu contoh jenis filter yang digunakan untuk menapis sinyal dengan frekuensi tinggi. Algoritme *High Pass Filter* dapat diimplementasikan di berbagai *platform* perangkat keras baik komputer maupun *embedded system*. Pengimplementasian *High Pass Filter* tidak mengalami kendala apabila dieksekusi di komputer. Namun Implementasi *High Pass Filter* pada *embedded system* menghasilkan beban komputasi yang relatif berat sehingga membutuhkan waktu eksekusi yang lama karena karakteristik *embedded system* yang memiliki frekuensi *clock* rendah dan *resource* yang terbatas. [1]

Untuk mengatasi masalah tersebut maka dalam penelitian ini dirancang sebuah implementasi *High Pass Filter* dengan menggunakan FPGA dengan tujuan agar dapat meningkatkan komputasi dan mempercepat waktu eksekusi. Penelitian ini berfokus pada implementasi algoritme *High Pass Filter* di FPGA dan membandingkan kecepatan eksekusinya dengan *embedded system* yang dalam hal ini digunakan prosesor Nios II dari Altera Cyclone IV.

### 1.2. Landasan Teori

#### 1.2.1. *Image Processing*

*Image Processing* adalah suatu sistem dimana proses dilakukan dengan masukan ( *input* ) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*). Pada awalnya pengolahan

citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision*. [2]

### 1.2.2. Penapisan (*Filtering*)

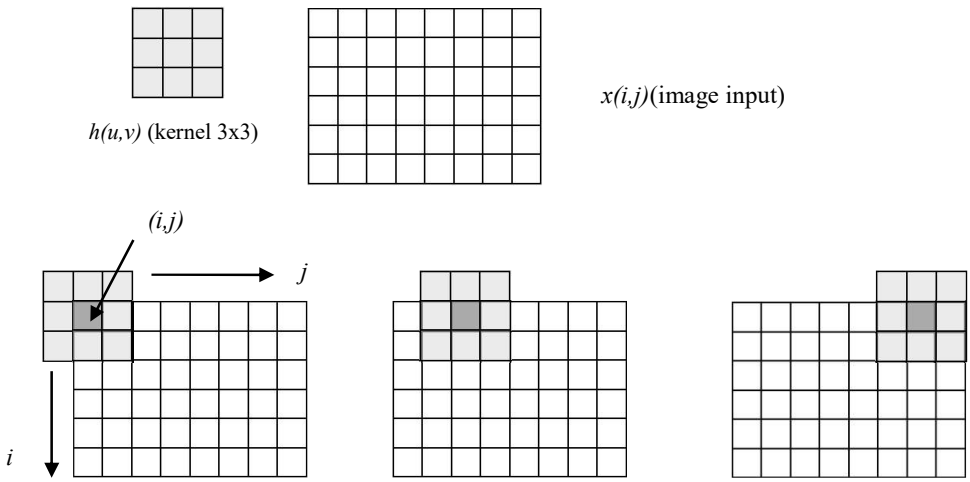
Penapisan atau *filtering* merupakan salah satu teknik dalam *image processing* untuk memperbaiki kualitas citra. Penapisan yang banyak digunakan adalah teknik *spatial filtering.* Operasi *spatial filtering* ini memodifikasi *pixel* berdasarkan dari *pixel* yang di pilih dan *pixel* tetangganya. Operasi ini membutuhkan kernel yaitu berupa matrik bujur sangkar seperti 3x3, 5x5 atau yang lebih besar dari itu. *Spatial fitering* dapat didefinisikan sebagai :

$$y(i,j) = \sum_{v}\sum_{u} h(u,v).x(i-u, j-v) \tag{1}$$

Operasi pada persamaan 1 dapat dipandang sebagai konvolusi antara *image input* x(i,j) dengan kernel tapis h(u,v). [3]

$$y(i,j) = h(u,v) * x(i,j) \tag{2}$$

Proses konvolusi




Gambar 1. Proses Konvolusi

Hasil konvolusi antara *image input* dengan kernel diletakkan bagian tengah dari kernel.

Jenis filter yang termasuk dalam kategori *spatial filtering* yaitu:

*1) Low Pass Filter*

*Low Pass Filter* (LPF) adalah suatu bentuk filter yang mengambil data citra pada frekwensi rendah dan membuang pada frekwensi tinggi. LPF digunakan untuk melakukan proses efek *blur* atau *smooth* dan reduksi noise. Ciri-ciri kernel dari LPF adalah semua nilainya positif dan jumlah dari semua nilainya sama dengan satu. [3]

$$\sum_{uv} H(u,v) = 1 \tag{3}$$

Contoh kernel LPF ukuran 3x3

$$h = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4}$$

2) *High Pass Filter* (*Sharpening*)

*High Pass Filter* (HPF) adalah suatu bentuk filter yang mengambil data citra pada frekwensi tinggi dan membuang pada frekwensi rendah. HPF digunakan untuk melakukan proses deteksi tepi, sehingga filter ini sering disebut juga filter *sharpening* (penajaman). Ciri-ciri kernel dari HPF adalah nilai-nilainya terdiri dari nilai positif, nol dan negatif, dan jumlah dari semua nilainya sama dengan nol. [3]

$$\sum_{uv} H(u,v) = 0 \tag{5}$$

Contoh kernel HPF ukuran 3x3

$$h = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \tag{6}$$

3) *High Pass Filter* (*Sharpening*) tanpa kernel penajaman.
*High Pass Filter* jenis ini dilakukan untuk memperoleh *image output* yang tajam dengan mengurangi hasil filter LPF dengan *image input*. [3]

$$High\ Pass\ Filter = image\ input - Low\ Pass\ Filter \tag{7}$$

**1.3. Metodologi.**

**1.3.1. Alat dan Bahan**

Alat dan bahan yang digunakan dalam perancangan sistem ini meliputi diantaranya:

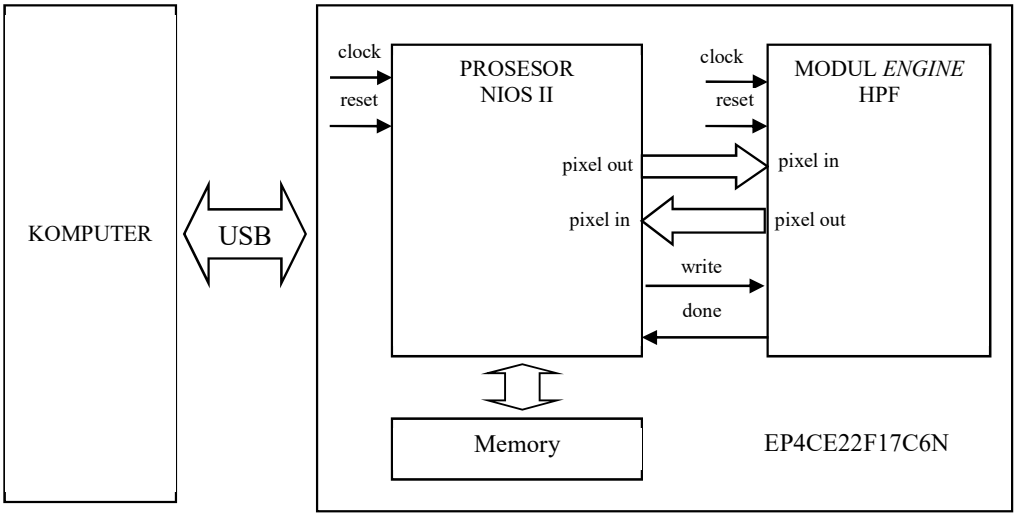
1) Komputer
2) FPGA Altera Cyclone IV EP4CE22F17C6N
3) *Software* Quartus II include Nios II EDS 13.0

**1.3.2 Langkah Perancangan**

1) Mendesain *hardware* Nios II dan Modul *engine* HPF pada FPGA dalam kode *verilog* dengan menggunakan software Quartus II
2) Menverifikasi desain *hardware* dengan menggunakan FPGA Altera Cyclone IV EP4CE22F17C6N
3) Merancang program HPF dengan bahasa C pada *hardware* prosesor Nios II dengan *tool* Nios II EDS

**1.3.3. Perancangan *Hardware* FPGA.**

Perancangan sistem *hardware* pada FPGA dapat dilihat pada Gambar 2. Terdiri dari 2 blok utama yaitu komputer dan FPGA Altera Cyclone IV EP4CE22F17C6N yang didalamnya terdapat prosesor Nios II, memori dan modul *engine* HPF yang dirancang.

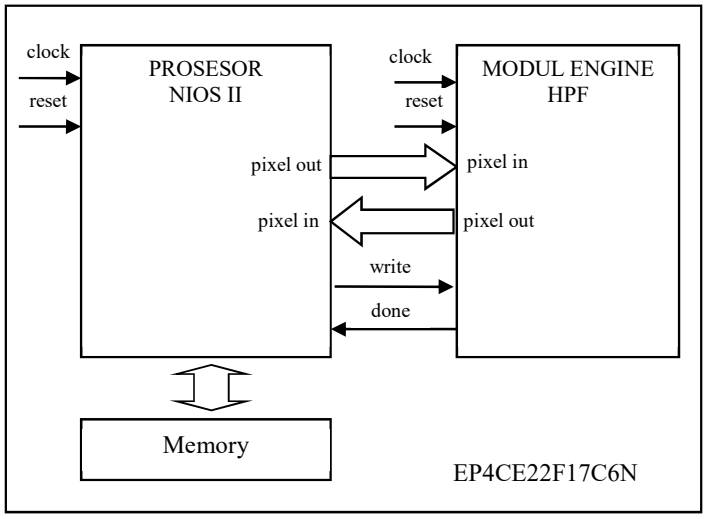



Gambar 2. Perancangan Sistem

Penjelasan secara umum blok pada Gambar 2, komputer berfungsi sebagai *programmer* untuk mendesain *hardware* FPGA dengan *software* Quartus II dan memverifikasi desain tersebut kedalam FPGA. Kemudian komputer akan berfungsi sebagai penerima pada saat desain *hardware* yang diverifikasi pada FPGA sukses dan prosesor Nios II telah diisi program pengiriman data serial ke komputer. Prosesor Nios II bekerja sebagai *host* prosesor yang bertugas untuk mengirimkan data *image* berupa *pixel* 8 bit yang tersimpan dalam *memory* ke modul *engine* HPF, modul HPF memproses *pixel input* tersebut dengan algoritma HPF yang telah ditanamkan pada FPGA dan mengirimkan kembali hasil filter ke prosesor Nios II, kemudian data hasil filter dikirimkan kembali ke komputer. Penjelasan rinci perancangan blok FPGA adalah sebagai berikut :

1) Prosesor Nios II dan Modul *Engine* HPF

Pada FPGA ada 3 blok *hardware* yang dirancang yaitu Prosesor Nios II bertugas sebagai kontrol utama, Modul HPF sebagai modul yang melakukan proses filter dan *memory* sebagai penyimpan data.




Gambar 3. Nios dan Modul *engine* HPF

Pada Gambar 3 ditunjukkan desain *hardware* berupa pin *input* dan pin *output* pada prosesor Nios II maupun modul *engine* HPF. Pin *clock* digunakan untuk masukan *clock* sebagai pemicu, pada FPGA ini besarnya *clock* dihasilkan dari kristal 50 Mhz, pin *reset* digunakan sebagai isyarat untuk *system reset* apabila berlogika 0 (*active low*). Pin *write* digunakan sebagai isyarat untuk mengaktifkan modul *engine* HPF jika berlogika 1, dan *done* digunakan sebagai isyarat proses filter telah selesai apabila berlogika 1. Port *pixel out* pada prosesor Nios II digunakan sebagai jalur pengiriman data *pixel image input* dengan lebar data data 8 bit dan data ini diterima oleh modul *engine* HPF melalui port *pixel in* dengan lebar data yang sama. *Port pixel in* pada prosesor digunakan sebagai jalur masukan dengan lebar data 8 bit yang dikirim dari *port pixel out* modul *engine* HPF, data yang dikirim oleh modul *engine* HPF tersebut merupakan data *pixel* hasil filter yang telah dikerjakan oleh modul.

*2) Memory*

*Memory* FPGA berfungsi sebagai penyimpan data *image input* yang akan difilter dan data *image ouput* yang telah difilter, *memory* ini mampu menyimpan *image* dengan ukuran 100x100 *pixel*.

### 1.3.4. Perancangan Program *High Pass Filter* (HPF) di Prosesor Nios II.

Perancangan program HPF diprosesor Nios II ditulis dalam kode bahasa C dan menggunakan *tool* Nios II EDS. Program ini dibagi menjadi dua bagian, yaitu program HPF di Nios II dan program HPF di Nios II untuk *engine* HPF. Alur program HPF di prosesor Nios II dimulai dengan membaca *image input* kemudian melakukan proses filter. Pada penelitian ini algoritme HPF yang digunakan adalah HPF tanpa menggunakan kernel *sharpening* (penajaman), tetapi menggunakan kernel LPF yang ukurannya adalah 16x16, sehingga algoritma program menggunakan formula dalam persamaan 7. Kemudian sebelum dilakukan operasi HPF, terlebih dahulu prosesor Nios II memerintahkan untuk menghitung waktu eksekusi menggunakan *IP interval timer* yang ditanam di FPGA. Setelah itu dilakukan iterasi HPF bekerja seperti biasa. Setelah iterasi selesai dan *pixel* hasil

output filter didapat, maka proses perhitungan waktu eksekusi berhenti. *Pixel* output hasil filter dan lamanya waktu eksekusi dapat dilihat pada layar komputer.
Pada program HPF di prosesor Nios II untuk *engine* HPF, prosesor Nios II berfungsi sebagai pengirim *pixel image input* dan penerima data *pixel output* hasil filter dari *engine* HPF. Pertama-tama prosesor Nios II akan mengirimkan data *image* berupa *pixel* 8 bit ke modul *engine* HPF sebanyak 16 byte *pixel*. Kemudian proses perhitungan waktu eksekusi baru dimulai pada saat isyarat *write* diaktifkan untuk mulai menjalankan operasi HPF. Setelah itu Nios II akan membaca pin *done* terus menerus sampai pin *done* bernilai 1. Prosesor Nios II akan mengambil data hasil filter *engine* HPF dan menyimpannya dalam *memory* serta menonaktifkan pin *write*. Proses iterasi ini akan berulang sampai semua *pixel input* dikirimkan oleh prosesor Nios II ke *engine* HPF. Apabila semua *pixel input* telah selesai dikirimkan maka operasi HPF di *engine* sudah selesai dan saatnya perhitungan waktu eksekusi dihentikan.

## 2. Pembahasan

Setelah dilakukan pengujian simulasi fungsional terhadap *engine* HPF yang telah dirancang, menunjukkan bahwa *engine* HPF mampu memfilter data dengan benar. Selain itu dilakukan pula pengujian algoritme HPF menggunakan *embedded system* prosesor Nios II. Berdasarkan data pada Tabel 1, *engine* HPF mampu mempercepat waktu komputasi 2.9 kali lebih cepat dibandingkan apabila hanya menggunakan prosesor Nios II. Sedangkan untuk total *logic element* yang diperlukan, *engine* HPF menambah jumlah *logic element* sebesar 587 elemen atau 22,4% dari jumlah total *logic element* yang digunakan Nios II. Dari peningkatan kecepatan dan penambahan jumlah *logic element* maka dapat diketahui rasionya yang disebut *energy efficiency* yaitu sebesar 12.9.

Tabel 1. Hasil Perbandingan Nios II dengan Nios II + *engine* HPF

| Parameter | Nios II | Nios II + *engine* HPF | Peningkatan | Efisiensi |
|---|---|---|---|---|
| Panjang siklus | 6369000 siklus | 2191430 siklus | Meningkat 2.9 kali (290%) | 12.9 |
| Waktu Eksekusi | 127380 us | 43828 us | | |
| Total *logic element* | 2561 | 3148 | Meningkat 587 elemen (22.4 %) | |

## 3. Simpulan

Modul *engine* HPF yang dirancang telah berjalan dengan baik. Hasil pengujian menunjukkan bahwa komputasi *High Pass Filter* pada FPGA dapat ditingkatkan menjadi 2.9 kali lebih cepat dibandingkan dengan ketika dijalankan diatas prosesor Nios II saja dengan kompensasi kenaikan jumlah *Logic Element* (LE) sebesar 22.4%.

## Daftar Pustaka

[1]. K. Babu Ravi Teja, Abhilash S. Warrier, Akshay S. Belvadi, Dhiraj R. Gawhane, *Design and Implementation of Neighborhood Processing Operations on FPGA using Verilog HDL,* IOSR Journal of VLSI and Signal Processing (IOSR-JVSP) Volume 4, Issue 1, Ver. II (Jan. 2014), PP 75-80 e-ISSN: 2319 – 4200, p-ISSN No. : 2319 – 4197
[2]. Schalkoff, R. J., 1989, *Digital Image Processing and Computer Vision*, John Willey & Sons, Inc.
[3]. R. Gonzalez and R. Woods *Digital Image Processing*, 3rd edition, Prentice Hall, 2008