

# Implementasi Penampil Citra Dengan Menggunakan *Picoblaze* FPGA

Debyo Saptono<sup>1</sup>, Reza Aditya Firdaus<sup>2</sup>, Atit Pertiwi<sup>3</sup>

<sup>1</sup>Fakultas Teknik-Jurusan Teknik Elektro, Universitas Gunadarma, Depok 16424  
E-mail : [debyo@staff.gunadarma.ac.id](mailto:debyo@staff.gunadarma.ac.id)

<sup>2</sup>Fakultas Teknik -Jurusan Teknik Informatika, Universitas Gunadarma, Depok 16424  
E-mail : [reza\\_adiya\\_f@student.gunadarma.ac.id](mailto:reza_adiya_f@student.gunadarma.ac.id)

<sup>3</sup>Fakultas Ilmu Komputer dan Teknologi Informasi, Universitas Gunadarma, Depok 16424  
E-mail : [atit@staff.gunadarma.ac.id](mailto:atit@staff.gunadarma.ac.id)

## ABSTRAK

*Picoblaze* merupakan sebuah prosesor yang memiliki 3 buah core yang tertanam di dalam satu chip. Prosesor *picoblaze* dapat diimplementasikan dalam sistem yang besar dan mempunyai fleksibilitas yang tinggi dalam desain berbasis FPGA. Perancangan *Picoblaze* yang dapat mengerjakan pengolahan citra diperlukan untuk dapat mempermudah proses penampilan citra sehingga setiap proses secara otomatis tanpa harus melakukan perubahan terhadap keseluruhan program yang digunakan. Citra tersebut dapat disimpan di dalam blok memori FPGA dengan menggunakan IPcore generator yang kemudian akan diproses dengan menggunakan *picoblaze* serta menanamkan program tersebut pada prosesor KCPSM3 yang kemudian akan ditampilkan pada monitor. *Picoblaze* merupakan mikrokontroler 8-bit yang didesain khusus untuk diimplementasikan pada FPGA. Hasil pengujian pada sistem FPGA Spartan 3E berjalan dengan baik dengan kebutuhan slice kurang dari 10 %.

**Kata kunci :** Perancangan, *Picoblaze*, KCPSM3, IPCore, FPGA

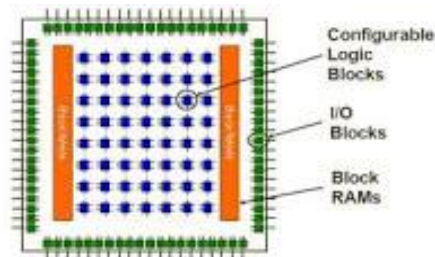
## 1. PENDAHULUAN

Pada pengolahan citra (*image processing*) diperlukan sebuah proses penampilan gambar yang disimpan didalam RAM sehingga dapat dimanipulasi sebelum ditampikan keluarannya pada layar monitor. Untuk dapat menghasilkan fleksibilitas yang tinggi pada pengolahan citra diperlukan sebuah prosesor yang dapat melakukan proses tersebut secara otomatis. Prosesor tersebut adalah KCPSM3, yang dapat diimplementasikan pada FPGA keluarga Spartan3. Dengan mengaplikasikan program menggunakan processor tersebut dapat menghasilkan fleksibilitas yang tinggi sehingga apabila dilakukan modifikasi program tidak perlu diubah seluruhnya[5][7]. Program pada KCPSM3 ditulis dalam bahasa assembler yang memiliki extension *.psm*.

## 2. METODOLOGI

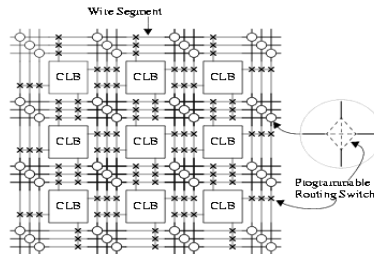
### 2.1 Field Programmable Gate Array (FPGA)

Field Programmable merupakan sebuah IC digital yang bersifat dapat dirancang dan diprogram sesuai dengan keinginan dan kebutuhan user/pemakai tanpa melalui tahap “burn” di laboratorium atau di “hardwire” oleh pabrik piranti. Sedangkan Gate Array artinya bahwa FPGA ini terdiri atas gerbang-gerbang digital dimana interkoneksi masing-masing gerbang tersebut dapat dikonfigurasi antara satu sama lainnya. FPGA merupakan sebuah IC digital yang bersifat programmable[2][7]. FPGA terdiri dari beberapa bagian diantaranya adalah Configure Logic Blocks (CLB), I/O Blocks, dan Programmable Interconnect.



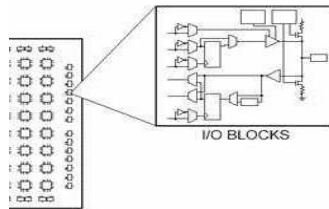
Gambar 1: Bagian-bagian FPGA

Configure Logic Blocks (CLB) merupakan bagian yang akan memproses segala bentuk rangkaian logika yang dibuat oleh user/pemakai.



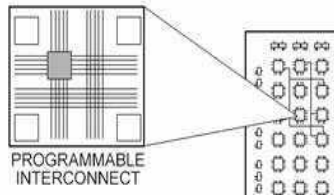
Gambar 2: Configure Logic Blocks

I/O Blocks berfungsi sebagai interface antara external pin dari device dan internal user logic.



Gambar 3: I/O Blocks

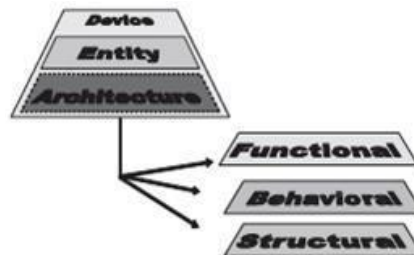
Programmable Interconnect berisi wire segments dan programmable switches, selain itu bagian ini juga akan menghubungkan antara CLB satu dengan CLB lainnya.



Gambar 4: Programmable Interconnect

## 2.2 VHDL

VHDL (*Very high speed integrated Hardware Description Language*) adalah sebuah bahasa pemrograman VHSIC (*Very High Speed Integrated Circuit*) yang dikembangkan oleh IEEE (*Institute of Electrical and Electronic Engineering*). VHDL merupakan bagian dari perancangan pemrograman FPGA. VHDL termasuk bahasa pemodelan yang digunakan untuk merancang atau memodelkan rangkaian digital. VHDL mampu melakukan desain hardware hingga sampai system yang lebih kompleks, mudah dalam mencari dan mendeteksi kesalahan dengan lebih mudah dalam simulasi, bahasa pemrograman yang mudah dimengerti dan dipelajari dengan cepat. Di dalam VHDL juga terdapat main body program, procedure serta function. VHDL memiliki kemampuan untuk menggunakan gabungan level dari model yang memiliki arsitektur yang berbeda seperti ditunjukkan sebagai berikut :



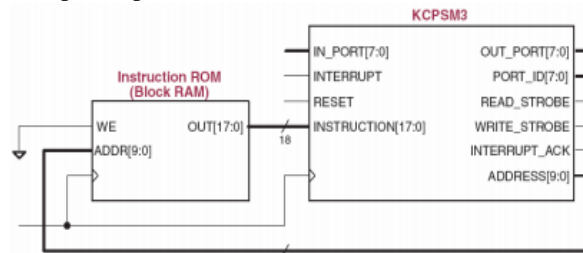
Gambar 5: Arsitektur Model VHDL

### 2.3 Picoblaze

Picoblaze adalah sebuah prosesor yang memiliki 3 buah core didalam satu chip. Picoblaze memiliki arsitektur 8-bit RISC dan memiliki kecepatan hingga 100 MIPS pada keluarga Virtex 4 FPGA. Desain picoblaze awalnya bernama KCPSM yang merupakan singkatan dari "Constant (K) Kode Programmable State Machine"[4][5]. Desain dasar untuk menggunakan picoblaze adalah :

- Picoblaze ditulis dengan menggunakan bahasa pemrograman assembler yang menghasilkan file dengan ekstensi .psm
- KCPSM3 assembler hanya dapat dijalankan pada file berekstensi .psm dan file berekstensi .vhd akan menjalankan instruksi yang berada di dalam Blok Memori sebagai outputnya

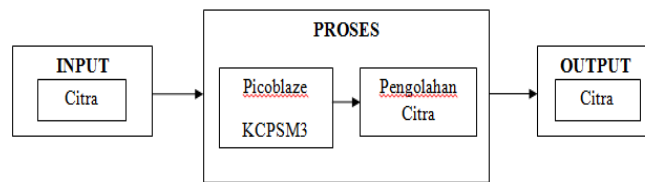
Memori program picoblaze diimplementasikan pada blok tunggal RAM didalam FPGA, yang selanjutnya dikonfigurasi hingga berfungsi sebagai ROM 1K × 18-bit. Program yang akan dieksekusi akan diinisialisasi dalam Blok RAM. Dibawah ini adalah komponen picoblaze.



Gambar 6: Komponen Picoblaze

### 3. Implementasi

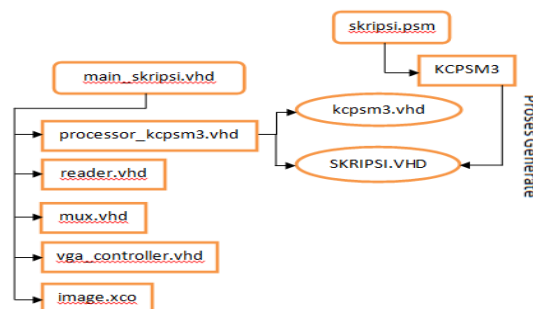
Pada analisa perancangan ini, menjelaskan mengenai bagaimana menganalisa perancangan program yang digunakan dalam Implementasi Picoblaze Pada Pengolahan Citra. Pada analisa hardware ini, akan dijelaskan analisa secara blok diagram. Analisa blok diagram menjelaskan setiap blok program seperti blok input, proses, dan menghasilkan output berdasarkan dari program yang digunakan.



Gambar 7: Blok Diagram Sistem

Pada blok input, didalamnya terdapat sebuah citra yang berfungsi sebagai input awal dari program. Masukan berupa citra tersebut dapat diubah-ubah dengan cara menyimpannya pada RAM FPGA. Pada blok proses, didalamnya terdapat dua buah proses yaitu Picoblaze dan Pengolahan Citra. Citra akan diproses didalam mikrokontroler picoblaze dengan menggunakan processor KCPSM3 yang tertanam didalamnya. Pada blok output hanya terdapat satu keluaran yaitu citra yang telah diolah untuk ditampilkan atau dikeluarkan pada layar sebuah monitor melalui interface VGA yang ada pada FPGA .

Pada analisa blok program ini, akan dijelaskan langkah-langkah dalam pembuatan program secara blok diagram beserta RTL Schematic dari masing-masing blok program.



Gambar 8 : Blok Program

Program yang digunakan pada perancangan ini terbagi menjadi beberapa blok. Blok program pertama adalah “main\_skripsi.vhd” yang merupakan program utama. Berfungsi untuk mendeskripsikan komponen-komponen beserta inialisasi port dan sinyal yang digunakan.

Blok program yang kedua adalah “processor\_kcpsm3.vhd”. Program ini merupakan program utama dari mikrokontroler Picoblaze yang digunakan. Terdapat dua komponen didalam program ini yaitu “kcpsm3.vhd” dan “ram.vhd”.

Komponen pertama berfungsi untuk mendeklarasikan alamat,instruksi,clock dan sinyal yang digunakan untuk dapat saling terhubung dengan komponen lainnya. Sinyal yang digunakan adalah Fundamental control & decode, stack, ALU, ALU multiplexer dan Read & Write strobe. Didalam program ini juga terdapat pendeklariansian terhadap atribut yang digunakan oleh processor kcpsm3. Komponen kedua didapatkan dari file “skripsi.psm” yang ditulis menggunakan bahasa pemrograman assembler. Didalamnya terdapat perintah untuk mengaktifkan perangkat I/O VGA dan RAM kcpsm3 agar dapat digunakan untuk memproses citra.

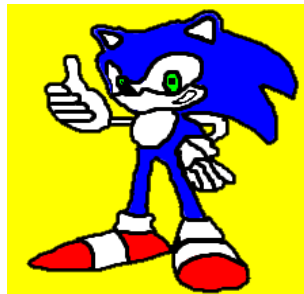
```
;;; IO PORTS FOR PERIPHERALS: VGA AND TEXT_RAM
constant VGA_CTL_PORT, 01
constant VGA_CRY_PORT, 02
constant VGA_CRX_PORT, 04
constant RAM_LOA_PORT, 08
constant RAM_HIA_PORT, 10
constant RAM_DOA_PORT, 20

;;; Backups
constant reg_sA, 00
constant reg_sB, 01
constant reg_sC, 02
constant reg_sD, 03
constant reg_sE, 04
constant reg_sF, 05

start:
;; Set VGA peripheral config
load s9, B2
output s9, VGA_CTL_PORT
```

Gambar 9: Kutipan skripsi.psm

Blok program ketiga yaitu “reader.vhd”. Berfungsi untuk melakukan proses pembacaan citra yang akan diolah, pembacaan dimulai dari posisi vertikal dan horizontal serta pendeklarasian nilai dari file .coe yang disimpan didalam RAM. Blok program keempat adalah “mux” yang berfungsi untuk melakukan enkripsi terhadap citra yang telah dibaca. Enkripsi dilakukan untuk dapat memberikan data keluaran yang kemudian akan diolah selanjutnya oleh VGA. Selanjutnya adalah blok program “vga\_controller”, program ini berfungsi untuk menghasilkan output berupa citra yang telah diolah didalam processor ke sebuah monitor dengan menggunakan interface VGA yang terdapat di FPGA, pada program ini juga dilakukan pendeklarasian terhadap resolusi citra yang digunakan. Blok program yang terakhir adalah “image.xco” untuk menyimpan citra didalam RAM. Program ini akan terbentuk secara otomatis disaat user menginput gambar kedalam RAM menggunakan ICore [1]. Gambar dibawah ini merupakan input yang akan di proses oleh FPGA.



Gambar 10: Citra Masukan

Sebelum dimasukkan kedalam blok memori , citra tersebut dikonversi terlebih dahulu menggunakan matlab sehingga menghasilkan file baru dengan ekstensi .coe.

```
clear all
close all
clc

x = imread('Image Sonic.bmp');

y = zeros(size(x));

y(:, :, 1) = x(:, :, 1) ~ = 0;
y(:, :, 2) = x(:, :, 2) ~ = 0;
y(:, :, 3) = x(:, :, 3) ~ = 0;

z = y(:, :, 1) * (2.^2) + y(:, :, 2)

fid = fopen('Image Sonic.coe',
```

Selanjutnya file .coe tersebut akan dibaca oleh blok program “reader.vhd” dan kemudian akan diproses oleh blok program “vga\_controller.vhd” yang kemudian akan ditampilkan pada layar monitor melalui interface VGA yang terdapat di FPGA[3][6].



Gambar 11: Hasil Output Citra Pada Monitor

#### 4. Kesimpulan dan Saran

Program yang dijalankan didalam FPGA akan terlebih dahulu diproses oleh processor KCPSM3 lalu kemudian akan diteruskan oleh program-program pendukung lainnya. Apabila ingin dilakukan perubahan ataupun pengembangan terhadap program tersebut dapat dilakukan dengan mudah dikarenakan program telah tertanam didalam processor sehingga tidak diperlukannya perubahan struktur program secara keseluruhan. Implementasi dengan menggunakan Spartan 3E-500 membutuhkan slice kurang dari 10%.

#### DAFTAR PUSTAKA

- [1] B. Muralikrishna, K. Deepika, B. Kauth dan V. Venna, “image processing using IPcore generator through FPGA”, International Journal of Computer Applications vol 46/23, 2012
- [2] D. Saptono, V. Brost, and F. Yang, “VLIW architecture compilation-simulation and its implementation into FPGA”, Journal of IRECOS, Italy, 2011.
- [3] J.A. Gomez-Pulido, “Introduction to FPGA design with Xilinx ISE 13 and prototyping boards”, 2012
- [4] J. Banks, “Introductions to using picoblaze microcontroller“, 2012
- [5] Kai-feng Zhang, Hua-min Tao, Shan-zhu Xiao, “Evolutionary Design of Image Filter Using PicoBlaze Embedded Processor”, Springer, 190-197, 2012.
- [6] UG230, “Spartan-3E starter kit, 2006.
- [7] V. Brost, C. Meunier, D. Saptono, and F. Yang, “Flexible VLIW processor based on FPGA for real-time image processing”, International Conference of DASIP, Finlandia, 2011.

#### Hak Cipta

Semua naskah yang tidak diterbitkan, dapat dikirimkan di tempat lain. Penulis bertanggung jawab atas ijin publikasi / pengakuan gambar, table dan bilangan dalam naskah yang dikirimkannya. Naskah bukanlah naskah jiplakan dan naskah tidak melanggar hak-hak lain dari pihak ketiga. Penulis setuju bahwa keputusan untuk menerbitkan/ tidak menerbitkan naskah dalam prociding yang dikirimkan penulis, adalah sepenuhnya hak Panitia. Sebelum penerimaan terakhir naskah, penulis diharuskan menegaskan secara tertulis, bahwa tulisan yang dikirimkan merupakan hak cipta penulis dan menugaskan hak cipta ini pada Panitia Seminar.