

PERANCANGAN ANTARMUKA PENGENALAN OBJEK MENGUNAKAN PERANGKAT LUNAK XILINX ISE DESIGN SUITE VERSI 14.5

Ragiel Hadi Prayitno ¹, Ary Bima Kurniawan ², Antonius Irianto ³
Dosen Universitas Gunadarma ^{1,2,3}

Jl. Margonda Raya No. 100 Pondok Cina Depok

Email : ragielhp@staff.gunadarma.ac.id ¹, bima@staff.gunadarma.ac.id ²,
antonius@staff.gunadarma.ac.id ³

ABSTRAK

Pengenalan citra (Image Recognition) banyak digunakan dalam berbagai hal, baik dalam bidang industri, kedokteran, keamanan dan lain sebagainya. Pengenalan citra/objek memiliki tingkat kebutuhan yang besar. Penerapan antarmuka ini memerlukan sistem yang mempunyai kecepatan yang tinggi dalam prosesnya. Namun dalam penerapan antarmukanya sistem berbasis perangkat keras memiliki kecepatan proses yang lebih tinggi, dibandingkan dengan sistem berbasis perangkat lunak. Penelitian ini membuat rancangan antarmuka perangkat keras untuk sistem pengenalan objek dengan menggunakan modul kamera OmniVision (OV) 7670. Penelitian ini memiliki 3 komponen utama yaitu modul kamera ov7670, antarmuka kamera dan monitor. Hasil penelitian menghasilkan susunan blok – blok antarmuka modul kamera ov7670 yang dapat berjalan sesuai dengan fungsinya masing – masing. Susunan blok yang dirancang terdiri dari top_level, freq_divider, debounce, ov7670_capture, vga_pll, ov7670_controller, frame_buffer, adress_generator, RGB, dan VGA. Dalam penelitian ini secara umum hasil sintesis dapat diterjemahkan kedalam bentuk skematik.

Kata Kunci : Objek, Antarmuka, ov7670, Xilinx, VHDL

ABSTRACT

Image Recognition is widely used in many ways, in the fields of industry, medicine, security, and so on. The introduction of images / objects has a high level of need. For the application of this interface a system is required with a high speed during the process. But in the application of interface, hardware-based systems have a higher processing speed compared to software-based systems. This research produced a hardware interface design for object recognition systems using the OmniVision (OV) 7670 camera module. This study has 3 main components, namely the ov7670 camera module, camera and monitor interface. The results of the study resulted in the compilation of overview70 interface blocks of the camera modules that can operate in accordance with their respective functions. The ordered block layout consists of top_level, freq_divider, debounce, ov7670_capture, vga_pll, ov7670_controller, frame_buffer, adress_generator, RGB and VGA. In this research in general, the results of the synthesis can be translated into a schematic form.

Keywords: Object, Interface, ov7670, Xilinx, VHDL

PENDAHULUAN

Dewasa ini pengenalan objek digunakan secara luas di berbagai bidang, baik dalam bidang industri, kedokteran, maupun dalam bidang keamanan. Pada bidang industri pengenalan objek digunakan untuk mengetahui kesalahan komponen, memilih komponen pada conveyor, perakitan perangkat dan lain sebagainya. Di bidang keamanan pengenalan objek dapat digunakan untuk mendeteksi penyusup[1], sedangkan dalam bidang kedokteran untuk menganalisa objek asing pada tubuh[2].

Proses penerapan antarmuka pengenalan objek, memerlukan sistem yang berkecepatan tinggi. Pada penerapan antarmuka, sistem pengenalan objek berbasis perangkat keras memiliki kecepatan proses yang lebih tinggi dibandingkan dengan sistem berbasis perangkat lunak. Jika dibandingkan dengan sistem berbasis perangkat lunak, sistem berbasis perangkat keras tidak fleksibel, membutuhkan waktu lama dalam pengerjaan, serta biayanya relatif lebih mahal.

Untuk mengatasi perbedaan sifat antara kedua basis sistem ini, dikembangkan sistem perangkat keras yang bersifat fleksibel, yang dapat dikonfigurasi berulang-ulang dengan bantuan perangkat lunak. FPGA (*Field Programmable Gate Array*) yang bersifat *reconfigurable* menawarkan solusi kecepatan dan fleksibilitas yang baik, serta dengan biaya yang rendah untuk mengimplementasikan antarmuka. Antarmuka yang telah diterjemahkan dalam rangkaian elektronika digital, dapat dibuat kode-kode perangkat lunaknya menggunakan bahasa HDL (*Hardware Description Language*). Kemudian kode tersebut dimasukkan pada perangkat FPGA.

METODE PENELITIAN

Penelitian ini merupakan suatu bagian dari implementasi Jaringan Saraf Tiruan pada FPGA dengan sistem yang efisien dalam penggunaan sumber daya. Dengan penggunaan sumber daya FPGA yang efisien, diharapkan proses komputasi dapat berlangsung lebih cepat (*real time*) dan mempunyai hasil yang akurat.



Gambar 1. Diagram susunan perangkat keras penelitian

Penelitian ini memiliki 3 komponen utama yaitu kamera, antarmuka kamera dan monitor. Ketiga komponen tersebut saling terhubung untuk mewujudkan pengaplikasian Jaringan Saraf Tiruan pada FPGA. Penelitian ini akan mengambil bagian dalam pembuatan antarmuka modul kamera ov7670. Antarmuka tersebut bertujuan untuk mendeteksi objek yang didapat oleh modul kamera ov7670. Objek yang didapat oleh modul kamera kemudian akan dideteksi oleh Jaringan Saraf Tiruan untuk mengetahui objek apa yang didapat oleh kamera modul tersebut.

a) Analisis Kebutuhan

Pada tahap persiapan, kebutuhan penelitian disiapkan. Persiapan dimulai dari perangkat keras, dan perangkat lunak, hingga konfigurasi perangkat lunak. Pada tabel 1, terdapat daftar kebutuhan perangkat yang dibutuhkan dalam penelitian.

Tabel 1: Daftar kebutuhan perangkat penelitian

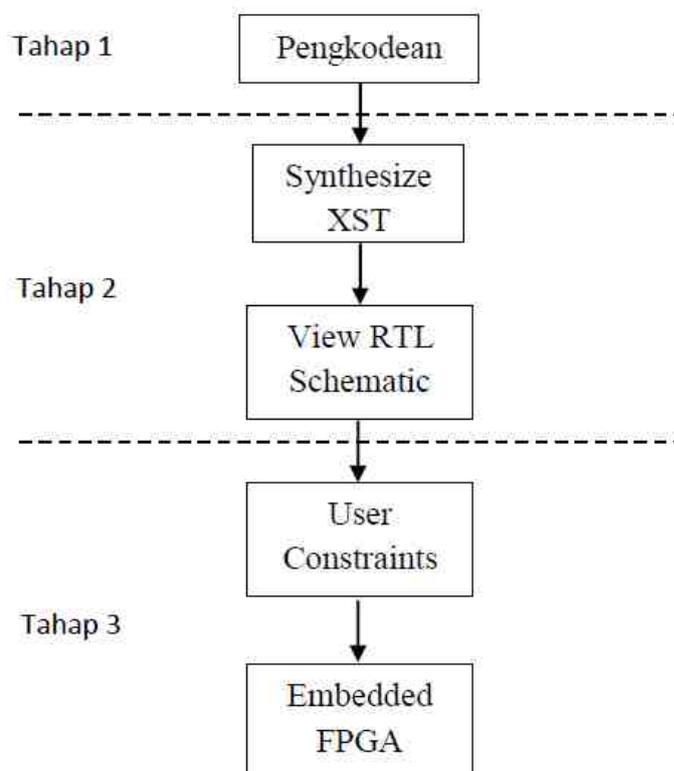
No	Nama Perangkat	Jenis Perangkat	Keterangan
1	Komputer / Laptop	Perangkat Keras	Intel Core i3
2	Monitor	Perangkat Keras	LCD / CRT
3	FPGA	Perangkat Keras	Nexys 3 Spartan 6
4	ov7670	Perangkat Lunak	Kamera Modul
5	Xilinx ISE Design Suite	Perangkat Lunak	Versi 14.5

Setelah mengetahui perangkat yang dibutuhkan untuk rancangan desain, proses selanjutnya mengkonfigurasi setiap perangkat lunak, khususnya perangkat lunak untuk saling terhubung dan dapat dijalankan pada sistem operasi windows 7. Pada sistem operasi windows 7, perangkat lunak yang akan dijalankan adalah Xilinx ISE Design Suite 14.5 yang berfungsi untuk membuat rancangan program menggunakan bahasa pemrograman VHDL (*VHSIC Hardware Description Language*), serta untuk melakukan proses *flashing* ke dalam perangkat FPGA. Kemudian hasil dari *flashing* akan dimunculkan kedalam monitor sesuai dengan *capture* yang didapat pada kamera modul ov7670.

b) Langkah – Langkah Penelitian

Pada umumnya sebelum mendesain suatu rancangan elektronika, terlebih dahulu dilakukan metode alur desain. Metode alur desain digunakan sebagai langkah - langkah sistematis yang dilakukan untuk menguji coba suatu rancangan desain. Metode alur desain yang digunakan pada penelitian ini dapat dilihat pada Gambar 2.

Pada gambar 2 metode alur desain dilakukan secara tiga tahap yaitu. Tahap pertama desain VHDL merupakan desain awal dengan mendeskripsikan melalui bahasa pemrograman perangkat keras. Tahap kedua membuat schematic dengan cara melakukan proses sintesis pada desain VHDL yang telah dibuat dengan menggunakan Xilinx ISE. Proses sintesis yang dimaksud adalah untuk menguji apakah dalam desain masih terdapat kesalahan, jika tidak terdapat kesalahan hasil sintesis akan berupa schematic. Tahap selanjutnya adalah tahap desain constraint, yaitu desain pengaturan fungsi dari setiap variable yang digunakan, dan untuk uji coba desain dalam bentuk sinyal. Setelah melakukan proses constraints, selanjutnya adalah proses *flashing* program VHDL kedalam FPGA yang akan menghasilkan output ke layar monitor.



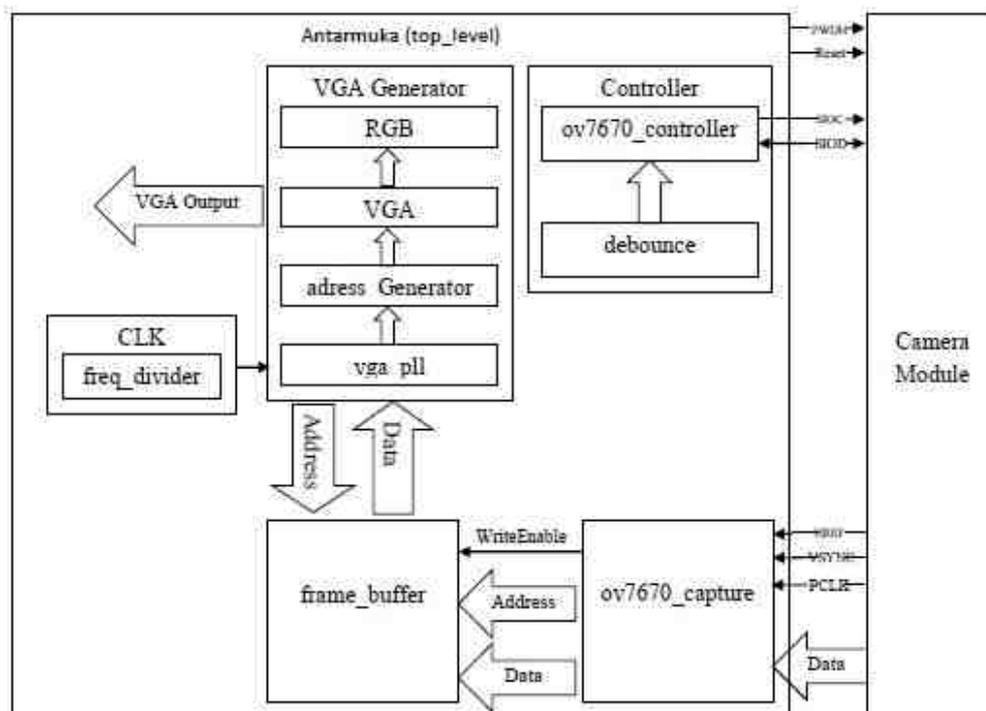
Gambar 2. Diagram langkah – langkah pembuatan program

Perancangan antarmuka pengenalan objek dalam penelitian ini hanya menggunakan dua tahap pertama saja, yaitu tahap pengkodean menggunakan VHDL dan membuat schematic.

c) Rancangan Blok Diagram Alat

Penelitian ini merancang blok diagram yang digunakan dalam membuat sebuah antarmuka modul kamera ov7670 menggunakan FPGA Nexys 3 Spartan 6. Pada rancangan alat antarmuka kamera modul ov7670 terdapat beberapa blok diagram seperti *camera module*, *ov7670_capture*, *frame_buffer*, *vga generator*, *controller*, dan *clk*. Masing – masing blok tersebut memiliki fungsi yang berbeda, yaitu :

- Blok *camera module* merupakan modul kamera yang digunakan untuk menangkap gambar yang diperoleh pada kamera.
- Blok *ov7670_capture* berfungsi untuk menerima data yang diperoleh modul kamera, kemudian mengirimkan data tersebut ke *frame buffer*.
- Blok *frame_buffer* berfungsi untuk mengumpulkan dan mengolah data yang dikirimkan oleh *ov7670_capture* untuk diproses pada *vga generator*.
- Blok *vga generator* berfungsi memproses data yang diterima dari *frame buffer* untuk ditampilkan ke layar monitor.
- Blok *controller* berfungsi untuk mengontrol modul kamera serta melakukan pengiriman register ke modul kamera.
- Blok *clk* berfungsi sebagai pemacu kerja alat.



Gambar 3. Blok diagram rancangan antarmuka modul kamera

PEMBAHASAN

Penelitian ini difokuskan menggunakan bahasa pemrograman perangkat keras HDL yaitu VHDL untuk menganalisis, memverifikasi dan mensintesis persamaan gerbang logika, sampai ke tahap kompleks yaitu kombinasi dari beberapa gerbang logika dalam arsitektur yang terpisah. Untuk mendapatkan bentuk desain skematik, maka

dibutuhkan proses *synthesize XST*. *Synthesize XST* merupakan proses penterjemahan dari Bahasa HDL ke dalam bentuk desain skematik.

1. Tahap Pengkodean

a) a Pengkodean Antarmuka

Pengkodean antarmuka ini merupakan induk dari rancangan program yang nantinya akan memuat seluruh komponen atau bagian yang dibutuhkan dalam merancang bangun antarmuka kamera ov7670. Pada penelitian ini penamaan pada pengkodean antarmuka adalah `top_level`. Berikut listing program dari antarmuka (`top_level`) :

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity top_level is
  Port ( clk_50      : in  STD_LOGIC;
        btn        : in  STD_LOGIC;
        config_finished : out STD_LOGIC;

        vga_hsync : out  STD_LOGIC;
        vga_vsync : out  STD_LOGIC;
        vga_r     : out  STD_LOGIC_vector(2 downto 0);
        vga_g     : out  STD_LOGIC_vector(2 downto 0);
        vga_b     : out  STD_LOGIC_vector(2 downto 1);

        ov7670_pclk : in  STD_LOGIC;
        ov7670_xclk : out STD_LOGIC;
        ov7670_vsync : in  STD_LOGIC;
        ov7670_href : in  STD_LOGIC;
        ov7670_data : in  STD_LOGIC_vector(7 downto 0);
        ov7670_sioc : out STD_LOGIC;
        ov7670_siocd : inout STD_LOGIC;
        ov7670_pwdn : out STD_LOGIC;
        ov7670_reset : out STD_LOGIC
  );
end top_level;

```

Gambar 4. Listing program pendeklarasian port pada antarmuka

Pada gambar 4 merupakan pendeklarasian port yang akan digunakan pada perancangan antarmuka kamera modul ov7670 pada FPGA. Port – port tersebut berfungsi sebagai output dan input sesuai dengan pendeklarasian.

```

COMPONENT VGA
PORT (
  CLK25 : IN std_logic;
  Hsync : OUT std_logic;
  Vsync : OUT std_logic;
  Nblank : OUT std_logic;
  clkout : OUT std_logic;
  activeArea : OUT std_logic;
  Nsync : OUT std_logic;
);
END COMPONENT;

COMPONENT ov7670_controller
PORT (
  clk : IN std_logic;
  resend : IN std_logic;
  sioc : INOUT std_logic;
  config_finished : OUT std_logic;
  sioc : OUT std_logic;
  reset : OUT std_logic;
  pwn : OUT std_logic;
  xclk : OUT std_logic;
);
END COMPONENT;

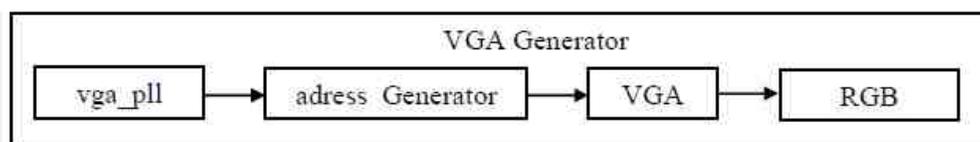
```

Gambar 5. Listing program pendeklarasian komponen pada antarmuka

Gambar 5 merupakan contoh pendeklarasian komponen yang dibutuhkan untuk mendukung perancangan antarmuka kamera modul ov7670. Berdasarkan gambar 5, maka akan terbentuk 2 buah komponen atau 2 sub program dari program induk atau top_level. Komponen tersebut merupakan salah satu komponen pendukung untuk merancang antarmuka modul kamera ov7670. Setelah seluruh komponen yang dibutuhkan telah dideklarasikan, maka selanjutnya akan mulai untuk merancang program untuk masing – masing komponen.

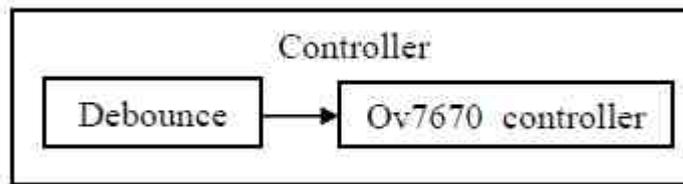
b) Pengkodean VGA Generator

Pengkodean vga generator ini berfungsi untuk mengolah dan menampilkan output atau keluaran yang diterima oleh modul kamera ke layar monitor. Pada pengkodean ini terdapat beberapa komponen pendukung yaitu vga_pll, adress_generator, vga, dan rgb. Komponen – komponen tersebut memiliki fungsi yang berbeda – beda.



Gambar 6. Rancangan Blok VGA Generator

- **Pengkodean vga_pll**
Pengkodean vga_pll berfungsi untuk mendeklarasikan clock yang dibutuhkan untuk mengimplementasikan antarmuka kamera modul ov7670.
 - **Pengkodean Adress_generator**
Pengkodean address_generator berfungsi untuk generate alamat data yang akan disimpan di memori.
 - **Pengkodean VGA**
Program VGA ini akan digunakan untuk menampilkan output ke layar monitor. Resolusi yang digunakan pada penelitian ini adalah 640x480. Resolusi ini diketahui berdasarkan definisi HD dan VD pada listing program vga.
 - **Pengkodean RGB**
Pengkodean RGB ini berfungsi untuk mendefinisikan hasil keluaran yang diperoleh kamera ov7670. Port Din pada komponen ini berfungsi untuk menentukan pixel grayscale. Port Nblank berfungsi untuk menunjukkan area tampilan, sedangkan port R, G, B berfungsi untuk menentukan warna yang diperoleh kamera ov7670.
- c) **Pengkodean Controller**
Pengkodean controller ini berfungsi untuk mengontrol data yang masuk dari modul kamera serta mengolah masukan dari push button. Pada pengkodean ini terdapat beberapa komponen pendukung yaitu debounce dan ov7670_controller. Komponen – komponen tersebut memiliki fungsi yang berbeda – beda.



Gambar 7. Rancangan Blok Controller

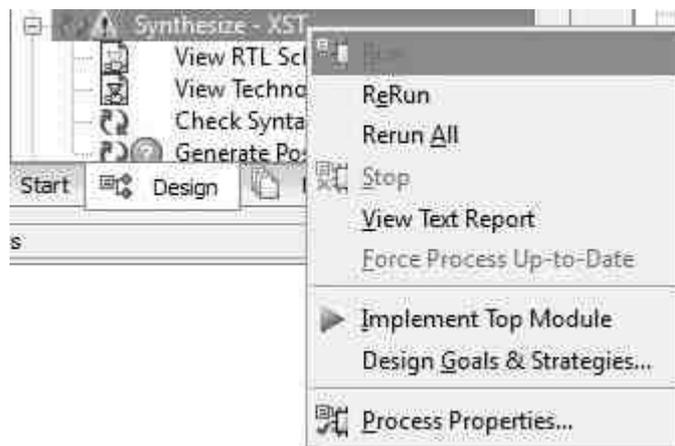
- **Pengkodean ov7670_controller**
Pengkodean ov7670_controller ini berfungsi untuk mengontrol modul kamera ov7670, serta melakukan pengiriman register ke modul kamera ov7670 melalui komponen I2C yang berfungsi seperti bus.
 - **Pengkodean debounce**
Pengkodean debounce ini berfungsi untuk merestart atau mereset inialisasi kamera dengan menggunakan tombol yang ada pada FPGA.
- d) **Pengkodean frame_buffer**
Pengkodean frame_buffer ini berfungsi untuk mengumpulkan data dari objek yang diterima oleh ov7670_capture, kemudian data yang diterima akan dikirimkan ke RGB untuk diolah warna yang diterima oleh ov7670_capture.
- e) **Pengkodean ov7670_capture**
Pengkodean ov7670_capture berfungsi untuk menangkap gambar yang diperoleh oleh kamera ov7670 dan kemudian dikirimkan ke frame_buffer untuk diolah datanya.

f) Pengkodean freq_divider

Pengkodean freq_divider ini dapat berfungsi sebagai pembagi clock pada FPGA agar dapat menampilkan hasil yang diperoleh oleh kamera ke layar monitor.

2. Tahap Synthesize XST

Synthesize XST merupakan proses penterjemahan dari bahasa pemrograman HDL ke dalam bentuk desain skematik. Setelah menyelesaikan perancangan program menggunakan bahasa pemrograman VHDL, langkah selanjutnya adalah melakukan *synthesize XST* dengan cara klik kanan pada Synthesize-XST >> kemudian klik run.



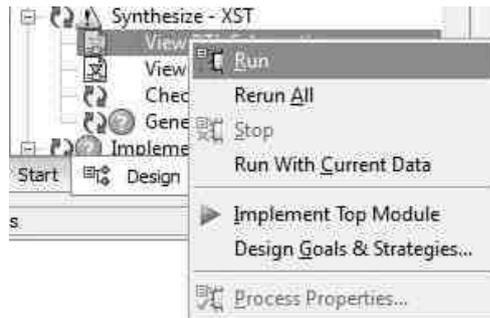
Gambar 8. Langkah untuk synthesize – XST



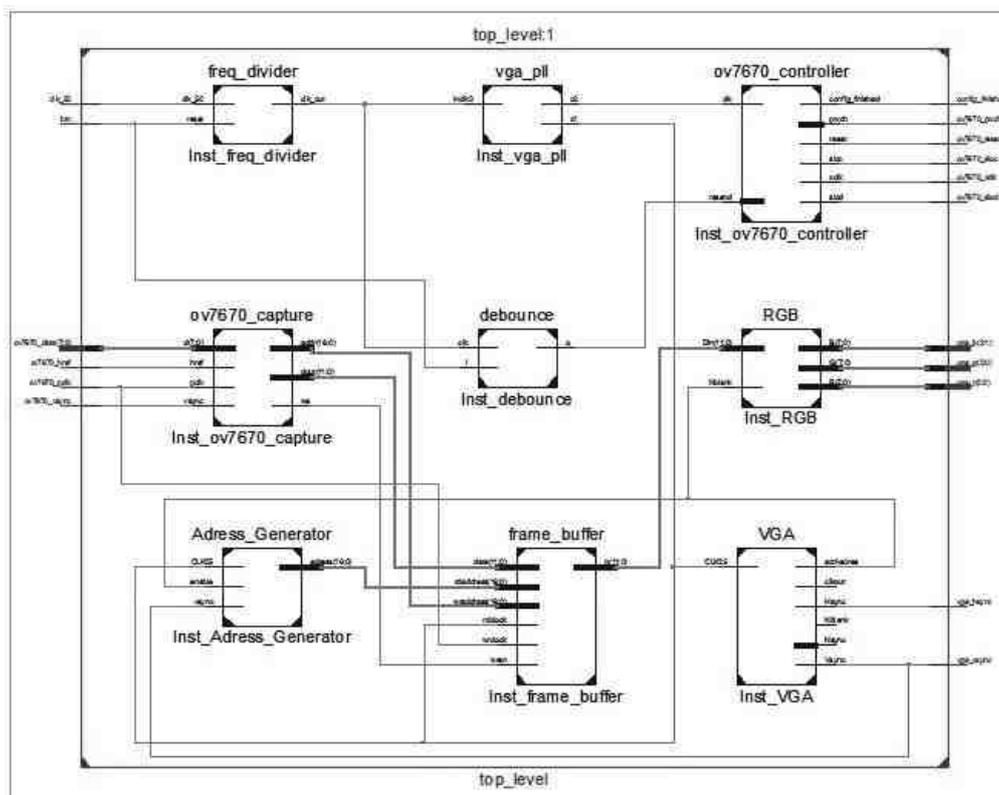
Gambar 9. Proses Synthesize

Dalam tahap View RTL Schematic, akan dilakukan proses penterjemahan dari hasil sintesis bahasa pemrograman HDL ke dalam bentuk desain skematik. Hasil skematik ini diperoleh setelah proses sintesis selesai.

Setelah proses running selesai, maka akan muncul hasil skematik dari rancangan program yang telah dibuat, seperti yang pada gambar 11.



Gambar 10. View RTL Schematic



Gambar 11. RTL Schematic

Gambar 11 merupakan hasil dari perancangan program untuk mengimplementasikan antarmuka kamera modul ov7670. Pada gambar diatas terdapat beberapa blok rangkaian, dimana setiap blok memiliki fungsi yang berbeda – beda. Adapun fungsi dari masing – masing blok program sebagai berikut :

- **Blok freq_divider**
Blok freq_divider ini memiliki fungsi untuk membagi clock yang digunakan agar modul kamera dapat menangkap sebuah objek dan menampilkannya pada monitor.
- **Blok ov7670_capture**
Blok ov7670_capture berfungsi untuk menangkap objek yang diperoleh dari modul kamera.

- **Blok frame_buffer**
Blok frame_buffer berfungsi untuk mengumpulkan data dari objek yang diterima oleh ov7670_capture, kemudian data yang diterima akan dikirimkan ke RGB untuk diolah warna yang diterima oleh ov7670_capture.
- **Blok vga_pll**
Blok vga_pll berfungsi untuk mendeklarasikan clock yang dibutuhkan untuk mengimplementasikan antarmuka kamera modul ov7670.
- **Blok ov7670_controller**
Blok ov7670_controller berfungsi untuk mengontrol modul kamera OV7670 serta melakukan pengiriman register ke modul kamera OV7670 melalui komponen I2C yang berfungsi seperti bus.
- **Blok debounce**
Blok debounce berfungsi untuk merestart atau mereset inisialisasi kamera dengan menggunakan tombol yang ada pada FPGA.
- **Blok adress_generator**
Blok adress_generator berfungsi untuk generate alamat data yang akan disimpan di memori.
- **Blok vga**
Blok vga digunakan untuk menampilkan output ke layar monitor.
- **Blok rgb**
Blok rgb berfungsi untuk mendefinisikan hasil keluaran yang diperoleh kamera ov7670.

KESIMPULAN DAN SARAN

Berdasarkan hasil perancangan yang diperoleh pada saat penelitian, maka dapat diambil kesimpulan bahwa perancangan susunan blok – blok antarmuka modul kamera ov7670 dapat berjalan sesuai dengan fungsinya masing - masing. Susunan blok yang dirancang terdiri dari top_level, freq_divider, debounce, ov7670_capture, vga_pll, ov7670_controller, frame_buffer, adress_generator, RGB, dan VGA. Dalam penelitian ini secara umum hasil sintesis dapat diterjemahkan kedalam bentuk skematik.

Penelitian selanjutnya diharapkan dapat melakukan tahap desain constraint. Tahapan tersebut dilakukan dengan mendesain pengaturan fungsi dari setiap variable yang digunakan, dan pengujian desain dalam bentuk sinyal. Setelah itu melakukan proses constraints, dan proses *flashing* program VHDL kedalam FPGA yang akan menghasilkan output ke layar monitor.

DAFTAR PUSTAKA

- [1] Zen , S Hadi and Entin , Martiana and Aries , Pratiarso and Ellysabeth, J.C., “Intrusion Detection System Based Snort Using Hierarchical Clustering”.2014.
- [2] Nurhayati O., “Metode Segmentasi Pada Analisis Citra Digital Head CT Scan”.2008.