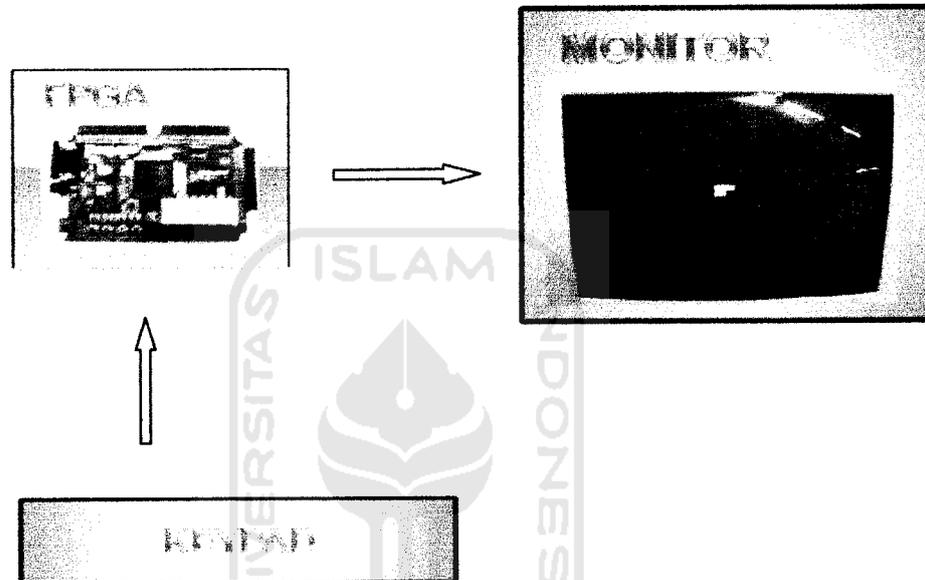


**Bab III**  
**PERANCANGAN SISTEM**



**Gambar 3.1** Blok diagram perancangan sistem.

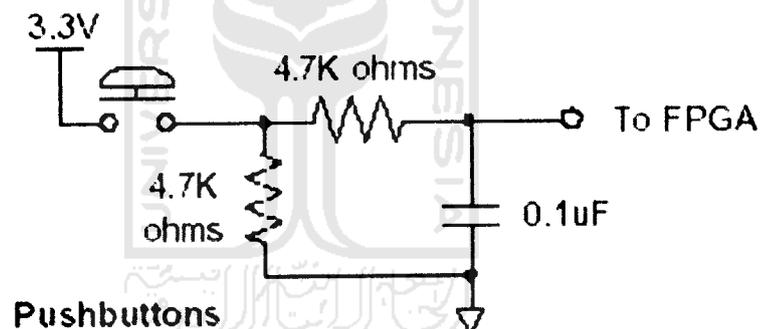
Gambar diatas adalah blok diagram yang menggambarkan sistem yang akan dirancang. Data yang diberikan melalui *keypad* akan ditampilkan dalam VGA monitor.

Fungsi masing - masing blok :

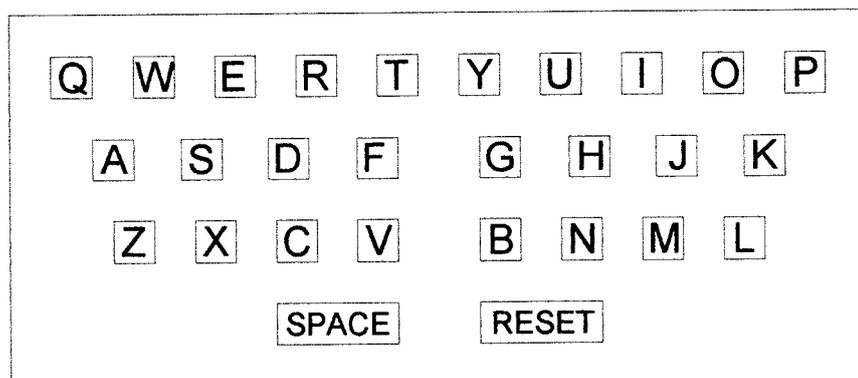
1. *Keypad* : sebagai input atau tempat memberikan data. Melalui *keypad* user dapat menentukan karakter apa yang akan ditampilkan pada monitor.
2. *FPGA board XC2S50* : akan mengolah data yang diberikan melalui *keypad* menjadi sinyal video.
3. Monitor akan menampilkan huruf yang telah diketik pada *keypad*.

### 3.1 Keypad

*Keypad* digunakan sebagai masukan, *port* yang digunakan sebagai masukan adalah *Port B1* pada modul pegasus dengan jumlah *pin* sebanyak 28 *pin*, dan satu *pin* untuk *ground* (*pin 1*) satu *pin* lagi untuk positif 3,3 Vdc (*pin 2*). Masing-masing *pin* mewakili 1 masukan. Setiap masukan berupa *pushbutton* harus melalui rangkaian RC terlebih dahulu, agar didapatkan masukan yang stabil dari *keypad*, sehingga FPGA dapat mengeksekusi perintah dengan benar, hal dikarenakan FPGA merupakan sistem digital (*high* dan *low*) dan dengan rangkaian RC maka masukan yang dihasilkan pasti berupa *low* (0) atau *high* (1).



(a)



(b)

**Gambar 3.2** Rangkaian RC untuk *input* FPGA (a), *Keypad* (b).

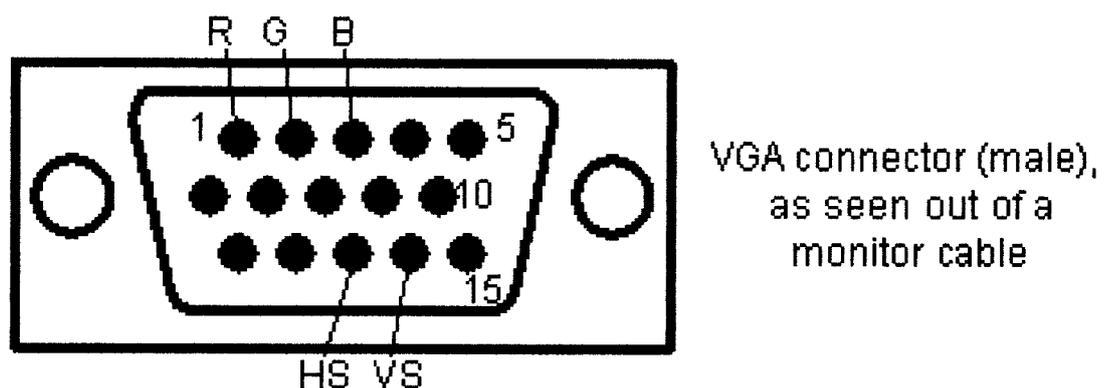
Masing-masing tombol pada *keypad* terbagi beberapa fungsi sebagai berikut:

1. Tombol A-Z : sebagai masukan karakter yang akan ditampilkan pada monitor.
2. Tombol *space* : sebagai tombol spasi.
3. Tombol reset : sebagai tombol reset keseluruhan, sehingga monitor akan menjadi *blank*.

### 3.2 Pengendali (*Controller*)

Pengendali yang digunakan adalah FPGA XC2S50. Blok ini berfungsi sebagai otak atau CPU, karena dalam blok semua proses komputasi diproses dan di eksekusi. Ketika ada data yang dimasukkan oleh *user* melalui *keypad* maka FPGA akan memproses dan mengolah data tersebut menjadi sinyal video, kemudian sinyal video tersebut akan ditampilkan ke monitor.

### 3.3 Port VGA



Gambar 3.3 konektor VGA.

Gambar 3.3 diatas adalah *port* VGA DB 15 yang terdapat pada FPGA sebagai penghubung antara monitor dan FPGA. Pin 1 atau *Red* pada FPGA mempunyai

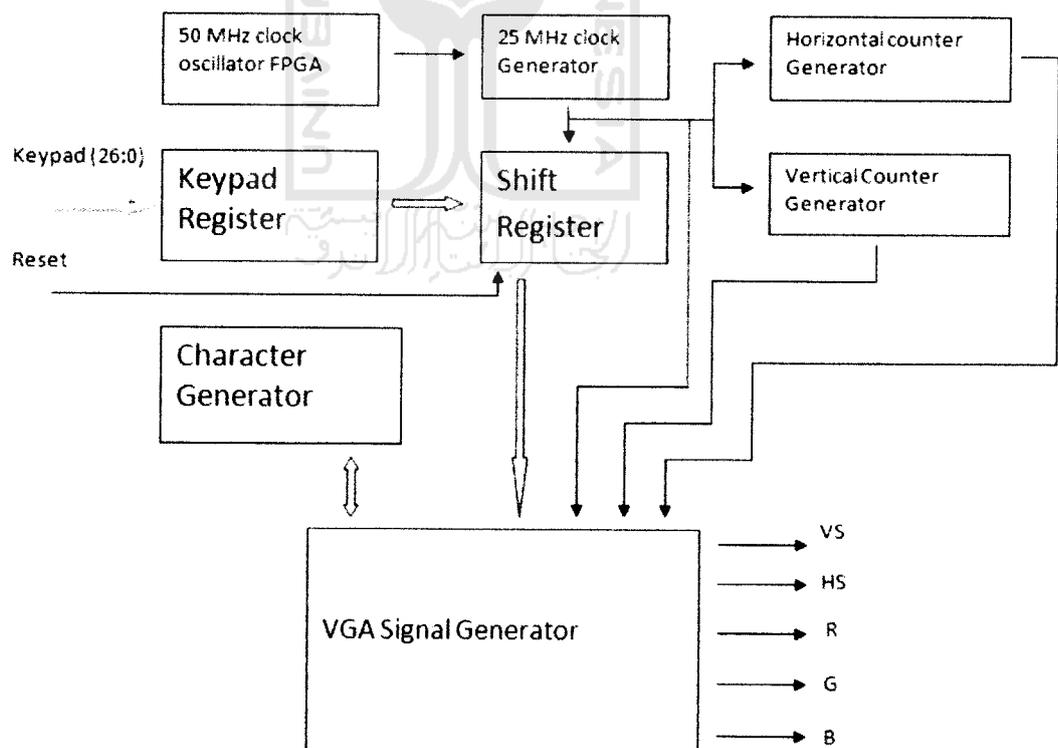
alamat pin p33, untuk G atau *Green* p31, untuk B atau *Blue* p30 sedangkan untuk Hs p29 dan Vs p27.

### 3.4 Display

Pada bagian ini digunakan monitor CRT sebagai penampil karakter yang telah diketik oleh *user*. Mode yang akan dipakai adalah 640 x 480 dengan frekuensi 25MHz.

### 3.5 Pembuatan *Software* (perangkat lunak)

Perancangan *software* dibuat dalam bahasa VHDL dengan menggunakan Xilinx.



**Gambar 3.4** Blok Diagram Desain Program.

### 3.5.1 Pembangkit *clock* 25MHz

Pada bagian ini akan dibuat suatu proses dalam program untuk membangkitkan *clock* dengan frekuensi 25MHz karena untuk mode monitor 640 x 480 membutuhkan *clock* frekuensi 25Mhz sedangkan *clock* pada FPGA XC2s50 adalah 50MHz.

### 3.5.2 Input

Input yang berupa keypad mempunyai beberapa fungsi, yaitu untuk mengeluarkan kode ASCII dari A sampai Z dan space yang akan digunakan untuk memanggil data yang telah dibuat dari *character map* serta fungsi yang kedua adalah sebagai trigger yang nanti dipakai dalam pembentukan *case digit* dalam program.

### 3.5.3 *Character Map*

Pada bagian ini akan dibentuk data untuk menampilkan huruf A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z dan space. Data dibuat dalam bentuk *array* sebagai contoh untuk huruf A dengan ukuran 12 x 8 pixel adalah sebagai berikut :

ASCII : 65	0 to 7	7 downto 0
08	00001000	00010000 10
1C	00011100	00111000 38
36	00110110	01101100 6C
63	01100011	11000110 C6
63	01100011	11000110 C6
7F	01111111	11111110 FE
63	01100011	11000110 C6
00	00000000	00000000 00
00	00000000	00000000 00

Gambar 3.5 *Bitmap* huruf A.

Dari gambar *Bitmap* huruf A diatas maka dibuat data *array*-nya menjadi :

```
(( '0','0','0','1','0','0','0','0'),
  ( '0','0','1','1','1','0','0','0'),
  ( '0','1','1','0','1','1','0','0'),
  ( '1','1','0','0','0','1','1','0'),
  ( '1','1','0','0','0','1','1','0'),
  ( '1','1','1','1','1','1','1','0'),
  ( '1','1','0','0','0','1','1','0'),
  ( '1','1','0','0','0','1','1','0'),
  ( '1','1','0','0','0','1','1','0'),
  ( '1','1','0','0','0','1','1','0'),
  ( '0','0','0','0','0','0','0','0'),
  ( '0','0','0','0','0','0','0','0'));
```

Data diatas ini adalah yang nantinya akan dikeluarkan melalui R G B dimana untuk bit '0' adalah pixel yang mati sedangkan bit '1' adalah pixel yang hidup sehingga akan membentuk huruf A .

#### 3.5.4 VGA Controller

Dalam bagian ini berisi kode VHDL yang membentuk *output* 5 sinyal video yaitu R, G, B, Vs dan Hs. R, G, B, atau singkatan dari *Red*, *Green*, *Blue* adalah sinyal video untuk mengeluarkan data dan membentuk karakter serta menentukan warna dari karakter melalui sinyal ini.

Sedangkan untuk membuat mode horisontal 640 dan vertikal 480 menggunakan *horisontal counter* dan *vertical counter* untuk mencacahnya. Untuk mode 640 x 480 membutuhkan *horisontal counter* sepanjang 800 dan *vertical counter* sepanjang 524. Angka ini didapat dari panjang horisontal *active video* 640 ditambah *front porch* 16 *sync pulse* 96 dan *back porch* 48 sedangkan untuk lebar vertikal *active video* 480 ditambah *front porch* 11 *sync pulse* 2 dan *back porch* 31.

### 3.6 Simulasi

Setelah semua program dibuat terlebih dahulu disimulasikan dengan menggunakan Active-HDL untuk melihat apakah program yang telah dibuat menghasilkan *output* ataukah tidak.

### 3.7 Test

Setelah simulasi dilakukan maka langkah selanjutnya program yang dibuat dalam Xilinx di-*download* pada FPGA dengan menggunakan JTAG *cable*. Metode yang paling mudah adalah dengan *Boundary-Scan Mode*.