

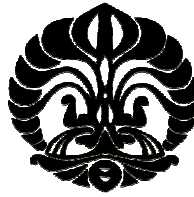
UNIVERSITAS INDONESIA

**PENGEMBANGAN DAN EVALUASI *SMART COUNTER*:
PENGHITUNG JUMLAH PENGUNJUNG RUANGAN
BERBASIS FPGA**

SKRIPSI

**DANI TRI SUTRISNO KURNIAWAN
0906602521**

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**PENGEMBANGAN DAN EVALUASI *SMART COUNTER*:
PENGHITUNG JUMLAH PENGUNJUNG RUANGAN
BERBASIS FPGA**

SKRIPSI

Diajukan untuk melengkapi sebagian persyaratan menjadi Sarjana Teknik

DANI TRI SUTRISNO KURNIAWAN
0906602521

**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
JUNI 2012**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri,
dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**



Nama : Dani Tri Sutrisno Kurniawan
NPM : 0906602521
Tanda Tangan : 
Tanggal : 03 Juli 2012


HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Dani Tri Sutrisno Kurniawan
NPM : 0906602521
Program studi : Teknik Elektro
Judul Skripsi : PENGEMBANGAN DAN EVALUASI
SMART COUNTER: PENGHITUNG
JUMLAH PENGUNJUNG RUANGAN
BERBASIS FPGA

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia

DEWAN PENGUJI

Pembimbing : Prof. Dr.Ing. Ir. Kalamullah Ramli, M.Eng. ()

Penguji 1 : Yan Maraden ST. MSc. ()

Penguji 2 : I Gde Dharma Nugraha ST. MT. ()

Ditetapkan di : Depok

Tanggal : 03 Juli 2012

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada penguasa jagat raya ini Allah SWT yang telah melimpahkan rahmat dan karunia-Nya serta petunjuknya, sehingga penulis dapat menyelesaikan Skripsi ini. Walaupun dalam penyusunan Skripsi ini penulis menemukan berbagai macam kesulitan, tetapi Allah SWT senantiasa memberikan rahmat dan petunjuk-Nya sehingga rintangan dan tantangan dapat dilalui dengan ridha-Nya.

Penyusunan Skripsi yang berjudul **“Pengembangan dan Evaluasi Smart Counter: Penghitung Jumlah Pengunjung Ruang Berbasis FPGA”** bertujuan untuk memenuhi syarat menyelesaikan pendidikan program Sarjana Teknik pada program studi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

Dalam melaksanakan Skripsi ini Penulis mendapatkan banyak bantuan dan bimbingan serta motivasi dari berbagai pihak. Untuk itu pada kesempatan ini penulis ingin menyampaikan terima kasih dan rasa hormat kepada :

1. Prof. Dr.-Ing. Ir. Kalamullah Ramli, M.Eng selaku dosen pembimbing yang telah memberikan petunjuk, kemudahan dalam berpikir dan bimbingan dalam penyelesaian Skripsi ini.
2. Bundaku yang sangat ku sayangi, selalu sabar mendoakanku disetiap hari dalam langkahku untuk menuntut ilmu serta berbagai aktivitasku yang terkadang membuatku tidak pulang kerumah.
3. Dosen-dosen pengajar dan pegawai Jurusan Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia.
4. Hana Afifah dan Martin selaku kawan kerja dalam pembuatan tugas akhir ini, terima kasih atas kerja samanya.
5. Seluruh teman-teman Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

6. Senior-senior dan junior-junior terima kasih atas bantuan selama penulis menyelesaikan Skripsi ini.

Semua pihak yang secara tidak langsung terlibat dalam pembuatan skripsi ini dan tidak mungkin dapat disebutkan satu persatu, semoga amal baik yang telah dilakukan senantiasa dibalas oleh Allah SWT.

Akhirnya, saran dan kritik yang mendukung sangat penulis harapkan dan nantikan terutama untuk kesempurnaan laporan ini. Semoga laporan ini dapat berguna bagi penulis sebagai pembuka gerbang kelulusan serta pembaca yang akan membaca dan ingin mempelajarinya.

Wassalam

Depok, Juni 2012

Penulis

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademika Universitas Indonesia, saya bertanda tangan di bawah ini:

Nama : Dani Tri Sutrisno Kurniawan

NPM : 0906602521

Program studi : Teknik Elektro

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Non-eksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

**PENGEMBANGAN DAN EVALUASI *SMART COUNTER*: PENGHITUNG
JUMLAH PENGUNJUNG RUANGAN BERBASIS FPGA**

Dengan Hak Bebas Royalti Non Eksklusif ini Universitas Indonesia berhak menyimpan, mengalih media / formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta sebagai pemegang Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

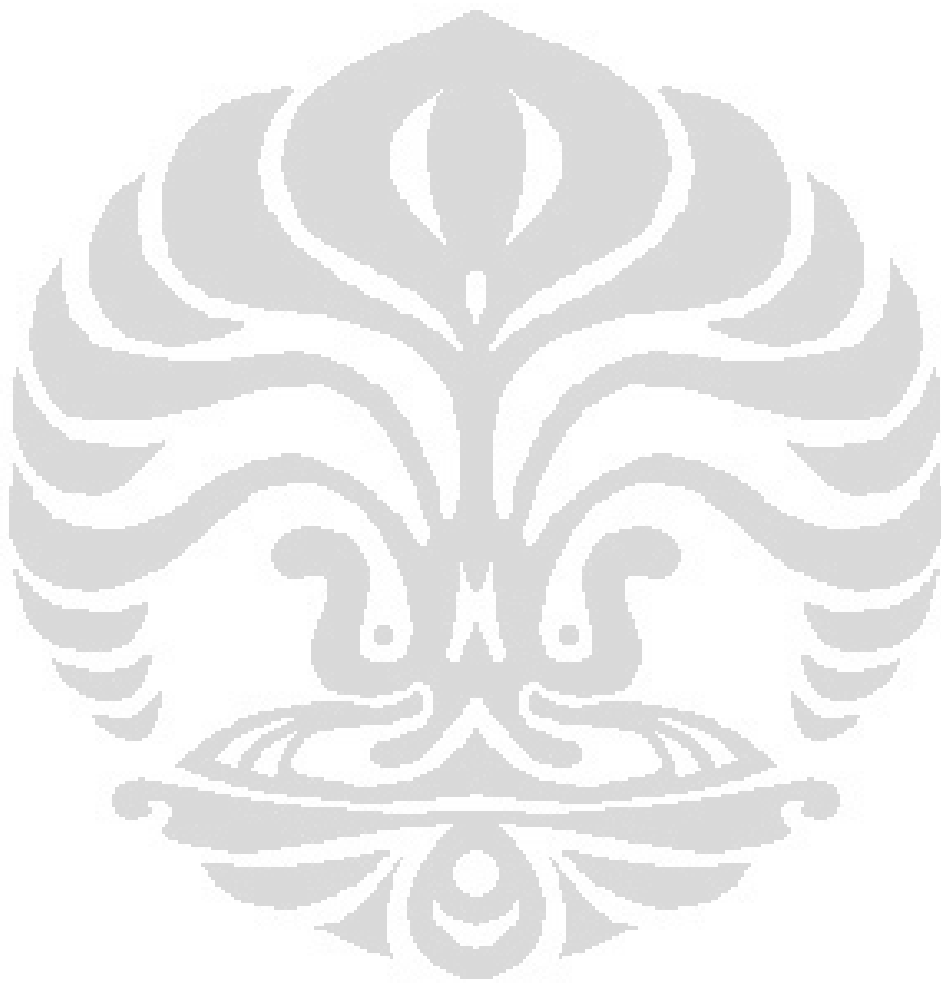
Dibuat di : Depok

Pada tanggal : 03 Juli 2012

Yang menyatakan,



(Dani Tri Sutrisno Kurniawan)



ABSTRAK

Nama : Dani Tri Sutrisno Kurniawan
Program Studi : Teknik Elektro
Judul : Pengembangan dan Evaluasi Smart Counter: Penghitung
Jumlah Pengunjung Ruang Berbasis FPGA

Penulis telah membuat sebuah alat Penghitung jumlah pengunjung dalam suatu ruangan yang dilakukan secara otomatis berbasis *FPGA*. Dimana pada alat ini kita mempergunakan *sensor LDR* sebagai komponen untuk mendeteksi pengunjung yang lewat dan *keypad* digunakan sebagai komponen untuk menginput jumlah kapasitas dalam suatu ruangan. Pada alat ini menggunakan menggunakan *FPGA* sebagai proses pengendali untuk menghitung serta *buzzer* sebagai penanda bahwa kapasitas telah penuh dari suatu ruangan. Implementasi dari alat ini adalah mampu melakukan perhitungan baik penambahan maupun pengurangan. Hasil perhitungan ditampilkan kedalam layar LCD. Bahasa pemrograman yang dipakai dalam kendali FPGA adalah VHDL. Dalam pemrograman dibagi menjadi dua yaitu program utama dan subprogram. Proram utama sebagai fungsi pengendali sedangkan pada subprogram sebagai fungsi tampilan ke layar LCD. Percobaan menunjukkan bahwa semua sistem berfungsi untuk digunakan dengan benar.

Kata Kunci : *FPGA, LDR, LCD, dan Buzzer.*

ABSTRACT

Name : Dani Tri Sutrisno Kurniawan
Study Program : Electrical Engineering
Title : Development and Evaluation of Smart Counter: FPGA based Room Visitors Counter.

Author designs automatic counters that counts number of visitors in a room. The device is develop based on FPGA. The tools used are LDR as a sensor for detecting passing visitors and keypad as components to set the maximum capacity of a room. Buzzer is used as the output device to signal audience that the room are full with visitor. The implementation of this tool is capable of performing the calculations for both the addition and subtraction. The calculation result is displayed into the LCD screen. VHDL is programming language used to the control FPGA. The programming is divided into two: main programs and sub programs. The main program as a fuction of controller while the sub program as a fuction of the display to the LCD screen. The experiment shows that all to function systems used correctly.

Keyword: *FPGA, LDR, LCD, dan Buzzer.*

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vii
ABSTRAK	viii
ABSTRACT.....	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR	xii
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan.....	1
1.3 Pembatasan Masalah.....	2
1.4 Metode Penelitian	2
1.5 Sistematika Penulisan.....	3
BAB 2 TEORI SISTEM DIGITAL DAN PENGENALAN FPGA XILINX SPARTAN 3E SERTA VHDL	4
2.1 Rangkaian Kombinasional.....	4
2.2 Rangkaian Sekuensial.....	5
2.3 Field Programmable Gate Array (FPGA).....	7
2.3.1 Sejarah FPGA.....	8
2.3.2 Konfigurasi FPGA.....	10
2.3.3 Cara Kerja FPGA.....	12
2.3.4 FPGA Design Software	13
2.4 VHDL	14
2.5 Light Dependent Resistor (LDR)	17
2.5.1 Sejarah Penemuan Foto-resistor / LDR.....	17
2.5.2 Simbol Foto-resistor / LDR.....	17
2.5.3 Mekanisme Foto-resistor / LDR.....	18
2.5.4 Aplikasi Foto-resistor / LDR.....	18
2.5.5 Rangkaian Elektronik Foto-resistor / LDR.....	18
2.6 (Light Emitting Diode) LED	20
2.6.1 Macam-macam LED.....	21
2.6.1.1 Diode Emiter Cahaya.....	21
2.6.1.2 LED Warna Tunggal	22
2.6.1.3 LED Tiga Warna Tiga Kaki	22
2.6.1.4 LED Tiga Warna Dua Kaki.....	22
2.6.1.5 Led Seven Segmen	22

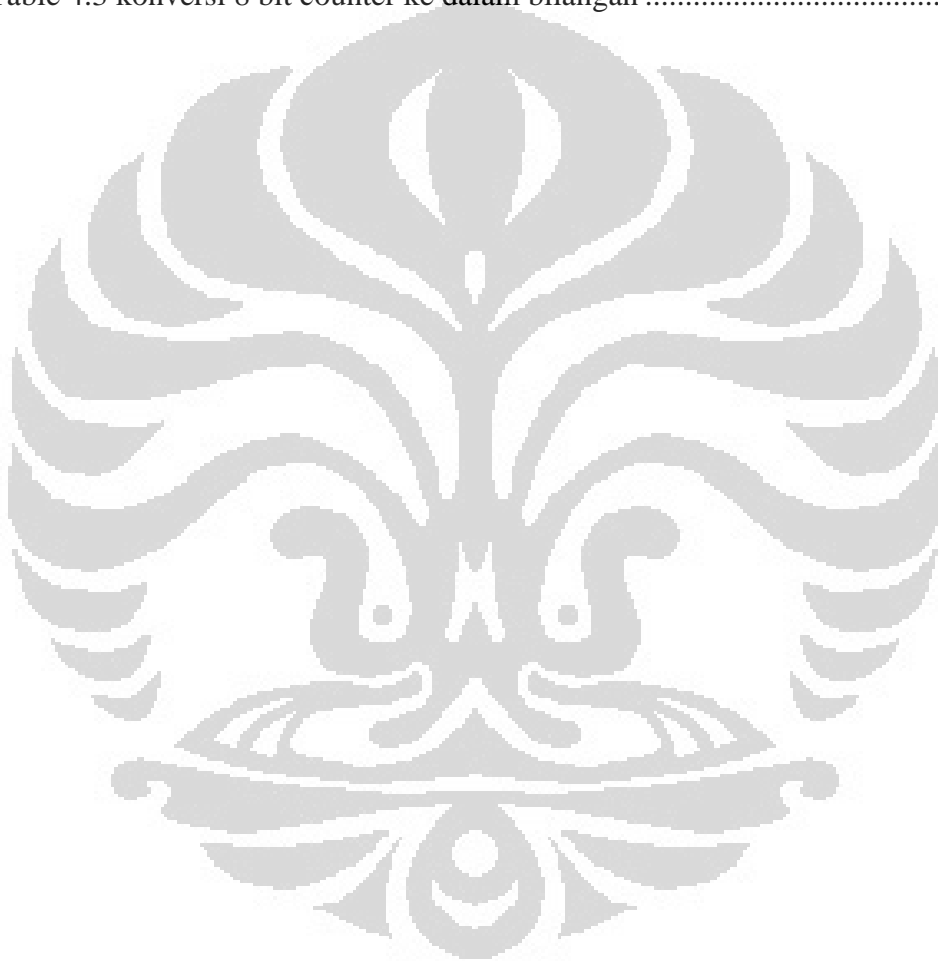
BAB 3 CARA KERJA SISTEM DAN PERENCANAAN PROGRAM	23
3.1 Desain Sistem	23
3.2 Perancangan Sistem.....	24
3.3 Rangkaian Sensor	25
3.4 Rangkaian Buzzer.....	25
3.5 Catu Daya	26
3.6 Modul FPGA SPARTAN 3E.....	26
3.7 Software Xilinx ISE Webpack 13.2	27
3.8 Perancangan Software	30
BAB 4 IMPLEMENTASI DAN ANALISIS.....	33
4.1 Implementasi	33
4.1.1 <i>Hardware</i>	33
4.1.1.1 FPGA	33
4.1.1.2 DI-MLDR	33
4.1.1.3 Voltage Regulator.....	35
4.1.1.4 Buzzer.....	35
4.1.1.5 LCD	35
4.1.2 Implementasi Perangkat Lunak (<i>Software</i>)	37
4.1.2.1 Program Counter.....	37
4.1.2.2 Program LCD	40
4.1.2.3 Program Buzzer	41
4.2 Uji Coba Sistem.....	41
4.2.1 Percobaan 1 – Pembacaan Sensor DI-MLDR	41
4.2.2 Percobaan 2 – Pembacaan Buzzer.....	43
4.2.3 Percobaan 3 – Pembacaan Hasil pada LCD	43
4.3 Analisis Hasil Percobaan	43
BAB 5 KESIMPULAN.....	45
DAFTAR ACUAN	46
DAFTAR PUSTAKA	47
LAMPIRAN	48

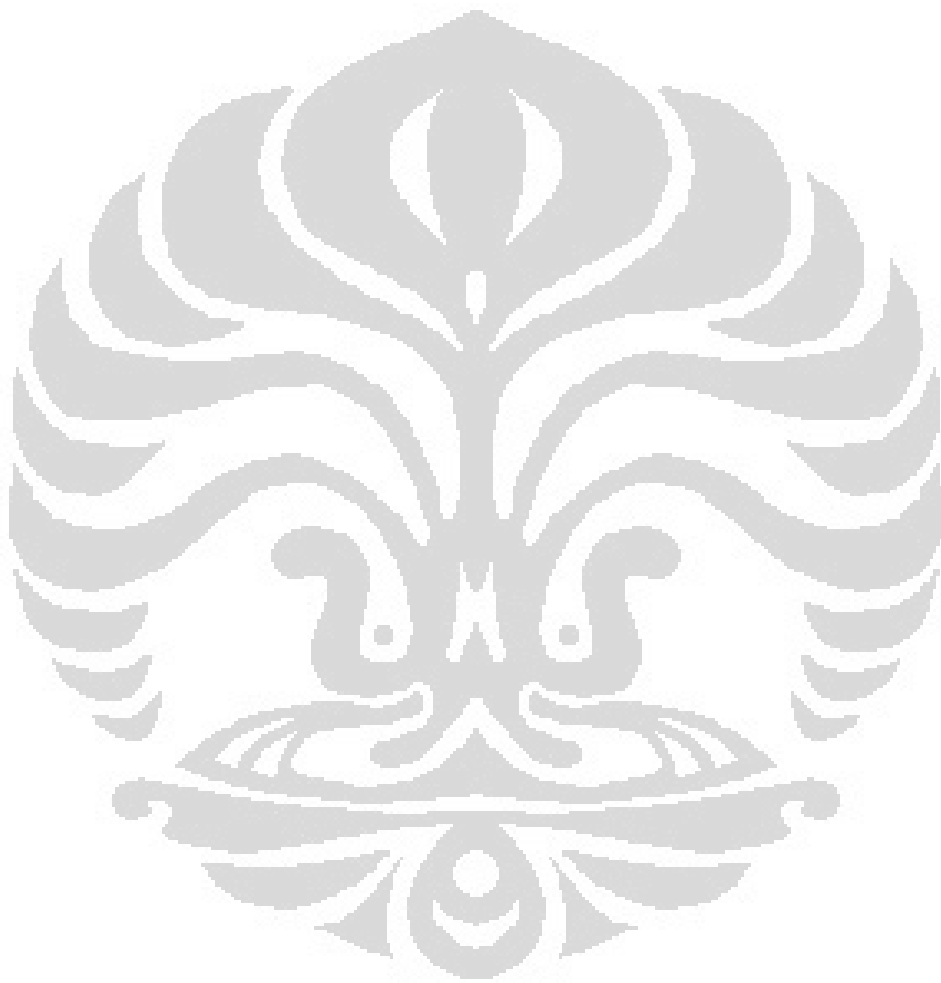
DAFTAR GAMBAR

Gambar 2.1 Diagram Blok Rangkaian Kombinasional [1].....	4
Gambar 2.2 Simbol, Ekspresi Boolean, dan Tabel Kebenaran dari: (a) Gerbang AND; (b) Gerbang OR; dan (c) Gerbang NOT [1].....	5
Gambar 2.3 Diagram Blok Rangkaian Sekuensial [1].....	6
Gambar 2.4 Simbol, Implementasi, dan Tabel Kebenaran dari: (a) Flip-Flop S-R; (b) Flip-Flop D; (c) Flip-Flop J-K [1].....	7
Gambar 2.5 FPGA[2].....	9
Gambar 2.6 Proses Download Ke FPGA[2].....	11
Gambar 2.7 Isi FPGA buatan Xilinx[2].....	12
Gambar 2.8 Salah Satu Development Board milik Xilinx[2].....	13
Gambar 2.9 Sebuah entitas VHDL terdiri dari interface (entitas deklarasi) dan tubuh (deskripsi arsitektur) [3].....	16
Gambar 2.10 Simbol dari LDR[4].....	18
Gambar 2.11 Rangkaian LDR[4].....	19
Gambar 2.12 Penambahan Kapasitor[4].....	19
Gambar 3.1 Desain Sistem.....	23
Gambar 3.2 Diagram Perancangan Alat.....	24
Gambar 3.3 Perancangan Rangkaian Sensor[6].....	25
Gambar 3.4 Perancangan Rangkaian Buzzer[6].....	26
Gambar 3.5 Perancangan Rangkaian Power Supply[6].....	26
Gambar 3.6 Board Starter Kit Xilinx Spartan-3E beserta komponen-komponennya[7].....	27
Gambar 3.7 ISE Project Navigator GUI: Eksplorasi, Proses, Entry dan Log[7]... ..	28
Gambar 3.8 ISE iMPACT: Konfigurasi Boundary Scan (JTAG) [7].....	28
Gambar 3.9 Metodologi desain sistem digital menggunakan ISE: entri.....	29
desain, sintesis, implementasi dan pemrograman[7].....	29
Gambar 3.10 Flowchart awal kapasitas.....	30
Gambar 3.11 Flowchart (A) Sensor masuk; (B) Sensor keluar.....	31
Gambar 3.12 Flowchat kapasitas telah penuh.....	32
Gambar 4.1 Koneksi FPGA ke Header J1.....	33
Gambar 4.5 Layout Rangkaian Sensor DI-MLDR.....	34
Gambar 4.2 Karakteristik Antarmuka LCD.....	36
Gambar 4.3 Set Karakter Kode ASCII.....	37

DAFTAR TABEL

Tabel 2.1 Persamaan karakteristik S-R, D, dan J-K Flip-Flop	7
Tabel 4.2 Fungsi PIN Sensor DI-MLDR	34
Tabel 4.1 Sinyal Antarmuka LCD	36
Table 4.1 Simulasi Objek Masuk Ruangan.....	41
Table 4.2 Simulasi Objek Keluar Ruangan.....	42
Table 4.3 konversi 8 bit counter ke dalam bilangan	42





BAB 1

PENDAHULUAN

1.1 Latar Belakang

Setiap hari kita selalu berada dalam ruangan yang terbuka maupun yang tertutup untuk melakukan kegiatan dari kegiatan formal ataupun non formal. Akhir-akhir ini sering terjadi sejumlah orang pingsan ataupun meninggal akibat padatnya suatu ruangan dan sesak bernafas. Hal ini disebabkan karena penyelenggara acara tidak memperhitungkan jumlah pengunjung dengan kapasitas ruangan yang di pakai dalam suatu acara atau pertunjukan.

Dengan berkembang teknologi semakin pesat. Manusia seolah-olah tidak henti-hentinya melakukan inovasi dalam penggunaan teknologi. Seiring perkembangan teknologi di segala bidang kehidupan baik komunikasi, industri, pendidikan, penelitian, kesehatan, dan lain-lain, maka semakin banyak pula alat yang diciptakan untuk mempermudah manusia dalam melakukan pekerjaan sesuai dengan bidangnya tersebut.

Oleh karena itu, untuk mengetahui jumlah pengunjung dalam suatu ruangan diperlukan suatu alat yang dapat menghitung berapa jumlah pengunjung yang ada di dalam suatu ruangan tersebut. Hal ini juga bisa dilakukan diruang kelas, diruang rapat, diruang pertunjukan, di cafe, dan diberbagai ruang lainnya.

Alat ini dibuat menggunakan kontrol oleh suatu sistem board yang sedang berkembang yaitu Field Programable Gate Array (FPGA). FPGA ditambah dengan beberapa sensor dan actuator untuk dapat menghitung dilengkapi dengan pemograman VHDL sebagai sistem kendali dari alat ini.

1.2 Tujuan

Penulisan Tugas Akhir ini merupakan salah satu syarat dalam menyelesaikan kurikulum Program S1 Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia. Tujuan membuat sistem menghitung jumlah pengunjung adalah untuk membuat suatu program VHDL yang dapat menghitung jumlah pengunjung dalam suatu ruangan menggunakan Field Programable Gate Array yang di singkat dengan FPGA.

1.3 Pembatasan Masalah

Dalam hal ini penulis membahas tentang simulasi dari sistem perhitungan dan pemrograman alat tersebut. Mulai dari pembuatan simulasi, yakni sistem pengendali sensor dan tampilan pada display yang seluruhnya diatur dengan bahasa pemrograman.

1.4 Metode Penelitian

a. Literatur

Penulis menggunakan metode ini untuk memperoleh informasi yang berkaitan dengan penelitian yang penulis buat. Study literatur ini mengacu pada buku-buku pegangan, data sheet dari berbagai macam komponen yang di pergunakan, data yang didapat dari internet, dan makalah-makalah yang membahas tentang proyek yang penulis buat.

b. Perancangan dan Pembuatan Alat

Berisi tentang proses perencanaan alat berupa sistem monitoring dan data logger. Pada bagian data logger akan membahas pembuatan rangkaian atenuator, arsitektur dari FPGA. Pada bagian sistem monitoring akan membahas masalah pembuatan program VHDL, pengambilan data, penganalisaan data, dan sistem pengendalian menggunakan FPGA.

c. Pengambilan Data

Pada bab ini akan diuraikan tentang kinerja dari masing-masing blok dengan harapan dalam pengujian tidak terdapat kesalahan yang fatal.

d. Uji Sistem

Dari alat yang dibuat maka dilakukan pengujian terhadap masing-masing bagian dengan tujuan untuk mengetahui kinerjanya apakah sudah sesuai dengan apa yang diharapkan atau belum

e. Penulisan Penelitian

Dari hasil pengujian dan pengambilan data kemudian dilakukan suatu analisa sehingga dapat diambil suatu kesimpulan. Dengan adanya beberapa saran juga dapat kita ajukan sebagai bahan perbaikan untuk penelitian lebih lanjut.

1.5 Sistematika Penulisan

Sistematika penulisan Tugas Akhir ini terdiri dari bab-bab yang memuat beberapa sub-bab. Untuk memudahkan pembacaan dan pemahaman maka Tugas Akhir ini dibagi menjadi beberapa bab yaitu :

BAB 1 Pendahuluan

Pendahuluan berisi latar belakang permasalahan, tujuan penulisan, pembatasan masalah, metode penulisan dan sistematika penulisan dari Tugas Akhir ini.

BAB 2 Teori Dasar

Teori dasar berisi landasan-landasan teori sebagai hasil dari studi literature yang berhubungan dalam FPGA, VHDL dan pemrograman alat.

BAB 3 Perancangan dan Cara Kerja Sistem

Berisi tentang penjelasan perancangan alat, program atau perangkat lunak (*software*) dan cara kerja dari program yang akan dibuat.

BAB 4 Hasil Eksperimen dan Analisa Data

Berisi tentang hasil dan analisa data yang telah diperoleh dalam perancangan atau pembuatan program baik kesalahan maupun kendala yang didapat.

BAB 5 Kesimpulan

Berisi kesimpulan dari keseluruhan perancangan sistem hingga hasil penelitian yang didapat.

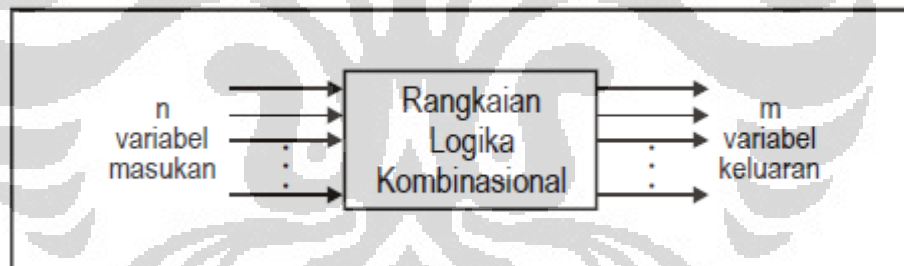
BAB 2

TEORI SISTEM DIGITAL DAN PENGENALAN FPGA XILINX SPARTAN 3E SERTA VHDL

Berbagai sistem digital telah banyak digunakan dalam berbagai bidang. Sistem digital merupakan kumpulan dari berbagai gerbang logika yang dapat dibedakan menjadi dua yaitu: rangkaian kombinasional dan rangkaian sekuensial.

2.1 Rangkaian Kombinasional

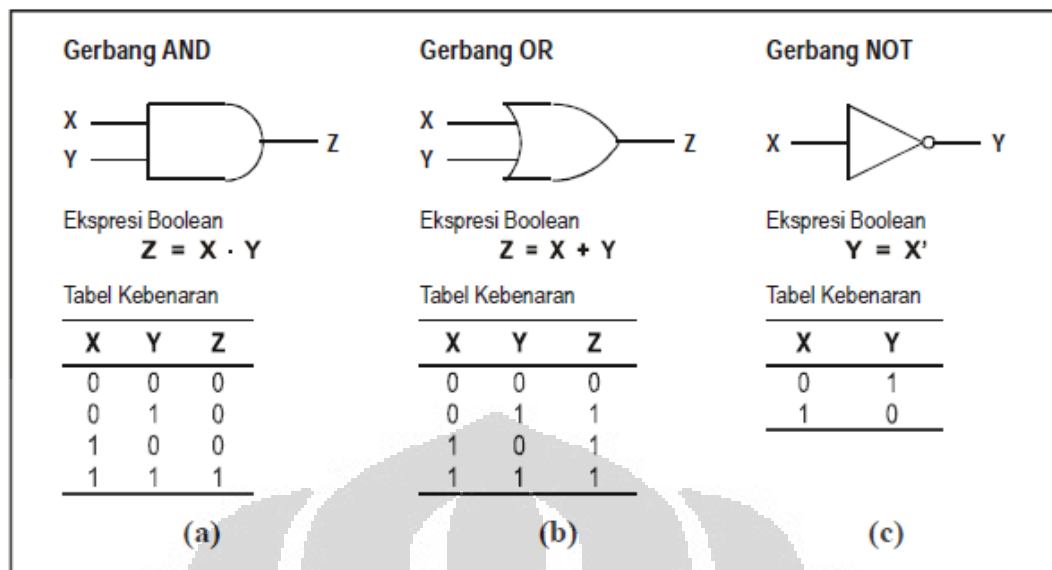
Rangkaian kombinasional adalah rangkaian yang nilai keluaran rangkaian di suatu waktu hanya ditentukan oleh nilai dari masukannya di waktu tersebut. Tidak ada penyimpanan informasi atau ketergantungan terhadap nilai sebelumnya. Misalnya: multiplexer, enkoder, dekoder, demux, dan ALU. Kombinasi tersebut dapat direpresentasikan dengan sebuah himpunan fungsi logika (disebut juga fungsi atau ekspresi *Boolean*). Gambar 2.1 mengilustrasikan digram blok rangkaian kombinasional.



Gambar 2.1 Diagram Blok Rangkaian Kombinasional [1]

Untuk setiap n variabel masukan, terdapat 2^n kombinasi nilai masukan biner yang dapat terjadi, masing-masing dapat diuraikan menjadi sebuah fungsi logika. Untuk setiap fungsi logika itu terdapat hanya satu variabel keluaran. Jadi untuk n buah variabel masukan, terdapat m fungsi logika dan m variabel keluaran, masing-masing untuk sebuah kombinasi masukan, dengan m sama dengan 2^n . [1]

Pada prinsipnya, semua rangkaian kombinasional dapat diimplementasikan menggunakan tiga buah gerbang logika dasar, yaitu: gerbang AND, OR, dan NOT. Gerbang logika lainnya dapat dibuat dari ketiga gerbang dasar ini. Simbol ketiga gerbang ini beserta tabel kebenaran dan ekspresi *Boolean*-nya ditunjukkan oleh Gambar 2.2.



Gambar 2.2 Simbol, Ekspresi Boolean, dan Tabel Kebenaran dari: (a) Gerbang AND; (b) Gerbang OR; dan (c) Gerbang NOT [1]

Dalam perancangan suatu rangkaian kombinasional, biasanya memperhatikan hal-hal berikut:

- Gerbang yang dipergunakan harus sesedikit mungkin;
- Banyaknya masukan ke sebuah gerbang harus minimum;
- Waktu yang diperlukan suatu isyarat untuk menjalar sepanjang rangkaian harus sesingkat mungkin;
- Interkoneksi sesedikit mungkin; dan
- Batasan kemampuan *fan-out* untuk masing-masing gerbang harus diperhitungkan.

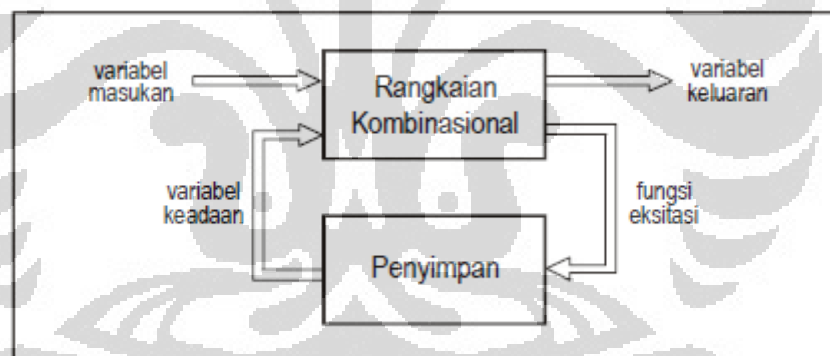
Biasanya tidak semua kriteria di atas dapat terpenuhi sekaligus. Oleh karena itu kriteria yang diutamakan disesuaikan dengan kebutuhan perancangan.

2.2 Rangkaian Sekuensial

Rangkaian sekuensial adalah suatu rangkaian yang nilai keluaran rangkaian di suatu waktu ditentukan oleh nilai masukannya waktu itu dan nilai masukan sebelumnya. Rangkaian sekuensial juga menyertakan ruang untuk menyimpan nilai masukan seperti elemen dasar untuk menyimpan data 1-bit adalah flip-flop. Sebagian besar rangkaian digital adalah sekuensial.

Keluaran sebuah rangkaian sekuensial tidak hanya bergantung pada kombinasi masukan saat itu, tetapi juga pada rentetan (*sequence*) dari masukan-masukan sebelumnya. Oleh karena itu dibutuhkan suatu penyimpan untuk mengingat ‘keadaan sekarang’, yang mencerminkan pengaruh rentetan input sebelumnya. ‘Keadaan’ dari sebuah rangkaian sekuensial adalah kumpulan ‘variabel keadaan’, yang nilainya menunjukkan semua informasi sebelumnya yang diperlukan untuk menentukan perilaku rangkaian berikutnya. Jadi, selain variabel masukan dan keluaran, rangkaian sekuensial juga memiliki variabel keadaan. Rangkaian sekuensial sering disebut juga ‘mesin keadaan’ (*state machine*).

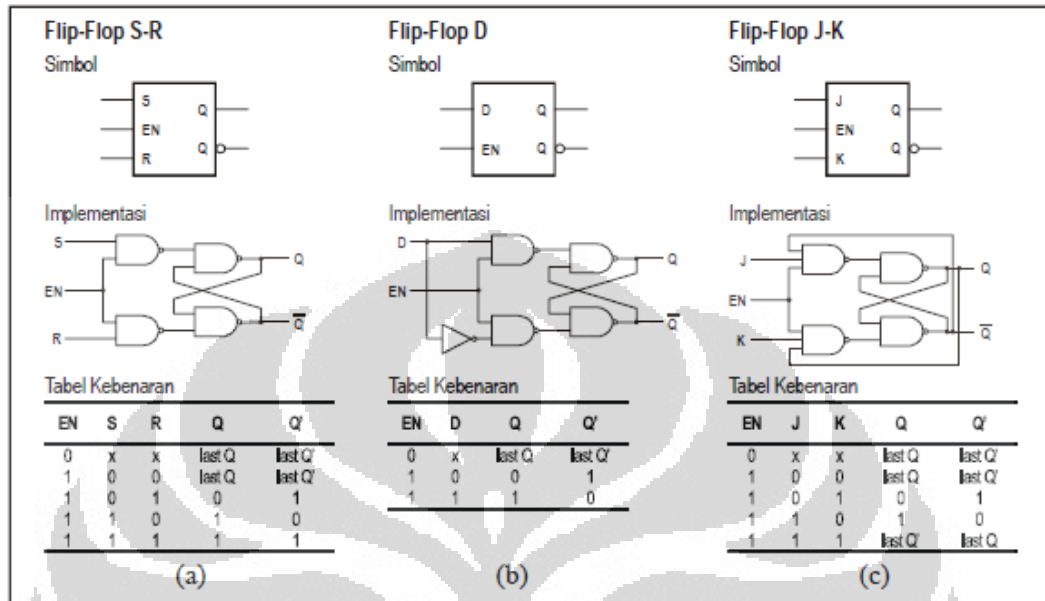
Dalam rangkaian digital, variabel keadaan bernilai biner. Sebuah rangkaian yang memiliki n buah variabel keadaan, memiliki 2^n kemungkinan keadaan. Variabel keadaan dipengaruhi oleh kombinasi input yang membentuk suatu fungsi eksitasi. Gambar 2.3 mengilustrasikan secara fungsional blok diagram dari rangkaian sekuensial.[1]



Gambar 2.3 Diagram Blok Rangkaian Sekuensial [1]

Ciri khas sebuah rangkaian sekuensial adalah adanya elemen penyimpan (biasa disebut ‘register’) yang berfungsi untuk mengingat keadaan rangkaian saat itu. Penyimpan berupa *flip-flop* atau *latch*. Flip-flop merupakan sel biner yang dapat menyimpan 1 *bit* informasi (disebut register 1-bit). Secara fisik, flip-flop berupa rangkaian gerbang logika yang disertai rangkaian umpan balik.

Ada tiga buah tipe dasar flip-flop, yaitu: flip-flop S-R, flip-flop D, dan flip-flop JK. Simbol masing-masing flip-flop, beserta implementasi menggunakan gerbang NAND dan tabel kebenarannya dapat dilihat pada Gambar 2.4.



Gambar 2.4 Simbol, Implementasi, dan Tabel Kebenaran dari: (a) Flip-Flop S-R; (b) Flip-Flop D; (c) Flip-Flop J-K [1]

Sebagai elemen penyimpanan, tiap flip-flop memiliki persamaan karakteristik yang menunjukkan keadaan flip-flop berikutnya sebagai fungsi dari input dan keadaannya sekarang. Tabel 2.1 menunjukkan persamaan karakteristik untuk tiap flip-flop.

Tabel 2.1 Persamaan karakteristik S-R, D, dan J-K Flip-Flop

Jenis Flip-Flop	Persamaan Karakteristik
Flip-Flop S-R	$Q^* = S + R' \cdot Q$
Flip-Flop S-R	$Q^* = D$
Flip-Flop S-R	$Q^* = J \cdot Q' + K' \cdot Q$

2.3 Field Programmable Gate Array (FPGA)

Field Programmable Gate Array (FPGA) adalah sebuah IC digital yang sering digunakan untuk mengimplementasikan rangkaian digital. FPGA dapat dirancang sesuai kebutuhan dan keinginan dari pemakai. Komponen elektronika

dan semikonduktor yang mempunyai komponen gerbang terprogram (programmable logic) dan sambungan terprogram. Komponen gerbang terprogram yang dimiliki baik jenis gerbang logika biasa yaitu, and, or, not, dan xor maupun jenis fungsi matematis dan kombinatorik yang lebih kompleks (decoder, adder, subtractor, multiplier, dan lain-lain). Blok-blok komponen di dalam FPGA bisa juga mengandung elemen memori (register) mulai dari flip-flop sampai RAM (Random Access Memory). FPGA merupakan rangkaian kombinasional dan sekuensial yang di kombinasikan sedemikian rupa untuk bisa diimplementasikan langsung pada silikon, merupakan salah satu bagian dari Application-specific integrated circuit (ASIC) yang memiliki kinerja tinggi tetapi memiliki fungsi yang spesifik (special purpose). Pengertian terprogram (programmable) dalam FPGA adalah mirip dengan interkoneksi saklar dalam breadboard yang bisa diubah oleh pembuat desain. Dalam FPGA, interkoneksi ini bisa diprogram kembali oleh pengguna maupun pendesain didalam lapangan atau lab (field).

2.3.1 Sejarah FPGA

FPGA ini dikembangkan sejak tahun 1984 oleh perusahaan Xilinx yang berbasis di San Jose, CA. Perkembangan selanjutnya, FPGA ini mulai diproduksi oleh beberapa perusahaan misalnya, Altera, Lattice, dan Quicklogic. Diantara perusahaan-perusahaan tersebut, terdapat 2 perusahaan yang mendominasi produksi FPGA di seluruh dunia yaitu Xilinx dan Altera. Minimal ada 5 perusahaan besar yang memproduksi FPGA[2]. Dua yang pertama merupakan pemain utama di pasar FPGA:

1. Xilinx yang punya nama besar dalam dunia FPGA, masih memimpin dalam densitas dan teknologi.
2. Altera merupakan pemain kedua terkenal di dunia FPGA, terkenal dengan namanya.
3. Lattice, Actel, Quicklogic adalah perusahaan-perusahaan yang lebih kecil dan punya “pasar khusus”.

Seperti yang diketahui, Field Programmable Gate Array atau yang lebih dikenal dengan FPGA merupakan sebuah IC digital yang dapat diprogram sesuai dengan kehendak pemakainya. FPGA berkembang sejak sekitar tahun 1980-an. Awal mula dari adanya FPGA ini tidak lepas dari adanya alat elektronik, yaitu transistor. Transistor merupakan alat semikonduktor yang dipakai sebagai penguat, sebagai komponen pemutus dan penyambung (switching), dan stabilisasi tegangan. Pada umumnya, transistor ini berfungsi sebagai saklar arus listrik yang hendak masuk ke suatu rangkaian elektronik. Perkembangan selanjutnya, adalah munculnya IC (Integrated Circuit) sekitar tahun 1950-an. Komponen elektronik ini lebih dikenal sebagai otak dari sebuah peralatan elektronika seperti televisi, handphone, computer, dsb. Kemudian berturut-turut muncul komponen lain, seperti DRAM, SRAM, microprocessor, ASIC hingga sampai pada FPGA. Gambar 2.5 menunjukkan bentuk dari FPGA. [2]



Gambar 2.5 FPGA[2]

Bila dilihat dari segi bentuknya, FPGA tak berbeda jauh dengan bentuk IC-IC lainnya. Hanya saja, bila dilihat dari isinya FPGA memiliki bagian yang berbeda dengan komponen IC pada umumnya. Berikut isi dari FPGA pada umumnya: 1. Configure Logic Blocks (CLB). Bisa dikatakan, bagian inilah yang akan memproses segala bentuk rangkaian logika yang dibuat oleh user/pemakai. 2. I/O Blocks. Sebagai interface antara external pin dari device dan internal user logic. 3. Programmable Interconnect. Bagian ini berisi wire segments dan programmable switches, selain itu bagian ini juga akan menghubungkan antara

CLB satu dengan CLB lainnya. Sebagian besar FPGA memiliki arsitektur seperti di atas. Entah itu FPGA produksi Xilinx ataupun dari Altera. Bila berbeda pun tak akan sampai menimbulkan perbedaan yang terlalu signifikan.

2.3.2 Konfigurasi FPGA

Sebuah FPGA (Field Programmable Gate Array) dapat terbagi atas dua kondisi yaitu kondisi awal (configuration mode) dan kondisi pemakai (user mode). Ketika pertama kali FPGA dihidupkan, maka otomatis keadaan FPGA ini berada dalam kondisi awal. Hal ini disebabkan karena FPGA masih dalam keadaan awal, fresh dan belum terdapat suatu program apa pun di dalamnya. Untuk menggunakan FPGA tersebut maka perlu adanya proses download program oleh pemakainya (user). Proses mendownload program ke dalam FPGA bertujuan untuk mengirimkan berkas-berkas bilangan biner (0 dan 1) melalui beberapa pin khusus. Setelah proses download dilakukan dan FPGA kemudian telah siap digunakan, maka inilah yang sering disebut kondisi pemakai. Pada kondisi ini FPGA telah aktif dan program yang didownload ke dalamnya dapat digunakan. Inilah yang dinamakan konfigurasi atau pengaturan pada FPGA. Bila pada kondisi pemakai, maka otomatis di dalam FPGA telah terdapat program. Program tersebut dibuat, diuji dan disimulasikan dahulu menggunakan PC (komputer). Setelah itu baru proses download program dilakukan seperti pada Gambar 2.6. Pada umumnya terdapat 3 cara yang biasa digunakan untuk mendownload program ke dalam FPGA yaitu :

1. Pemakai dapat mendownload langsung program ke dalam FPGA menggunakan kabel yang disambungkan ke PC. Program tersebut dibuat dan diolah menggunakan PC (komputer) dan ketika siap digunakan, baru didownload ke FPGA melalui kabel.
2. Pemakai dapat menggunakan mikrokontroler pada board yang ada, dengan firmware yang cukup guna mengirimkan data ke dalam FPGA.

3. Pemakai dapat menggunakan “boot-PROM” pada board yang ada, yang dihubungkan ke FPGA, dan mengatur FPGA tersebut supaya secara otomatis bekerja tanpa proses download dulu ke dalamnya (Perusahaan-perusahaan FPGA pada umumnya memiliki spesifikasi khusus untuk tambahan boot-PROMS).

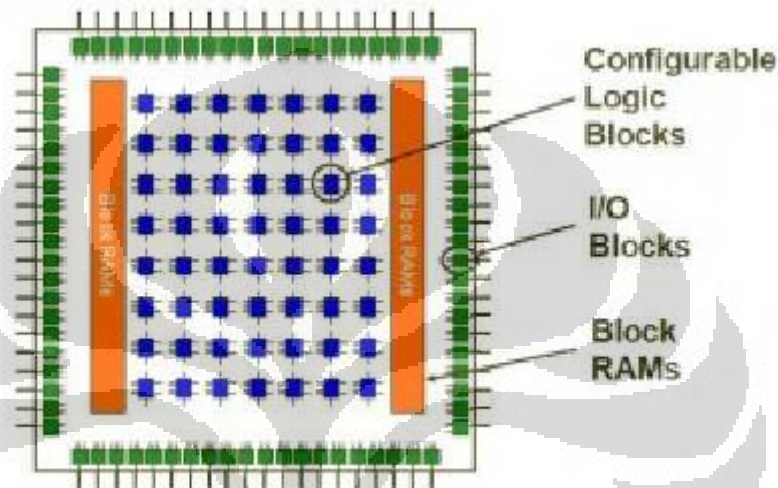


Gambar 2.6 Proses Download Ke FPGA[2]

Dari ketiga cara di atas, cara pertama adalah yang paling sering dan banyak digunakan. Selain itu, cara pertama cukup mudah dan efisien untuk dilakukan. Jadi, ketika pemakai telah mendownload program ke dalam FPGA maka tidak lagi diperlukan PC (komputer). FPGA akan bekerja sesuai dengan program yang telah didownload ke dalamnya tanpa perlu lagi terhubung dengan PC (komputer). Proses konfigurasi FPGA untuk buatan Xilinx maupun Altera hampir sama. Perbedaannya, hanya masalah pemberian nama pada pin-pin FPGA tersebut (nama pin dan jenis operasinya berbeda) . Namun, sebagian besar fungsi dan kegunaannya sama.

2.3.3 Cara Kerja FPGA

FPGA (Field Programmable Gate Array) merupakan sebuah IC digital sering digunakan untuk implementasi rangkaian digital. IC digital ini pada umumnya terdiri atas 3 bagian yaitu configure logic blocks (CLB), I/O Blocks, dan Programmable Interconnect. Baik FPGA buatan Xilinx maupun Altera memiliki 3 bagian seperti yang disebutkan di atas dapat dilihat pada Gambar 2.7.



Gambar 2.7 Isi FPGA buatan Xilinx[2]

Sebuah FPGA tersusun dari sebuah bagian yang bernama “logic cell” (Logic Blocks), yang kemudian pada perkembangannya saling terhubung satu sama lain. Kumpulan-kumpulan dari logic cell ini berjumlah ratusan bahkan ribuan dan membentuk suatu fungsi yang kompleks. Sebuah logic cell pada dasarnya terdiri atas sebuah lookup table(LUT), D flip-flop, dan sebuah multiplexer 2 ke 1.

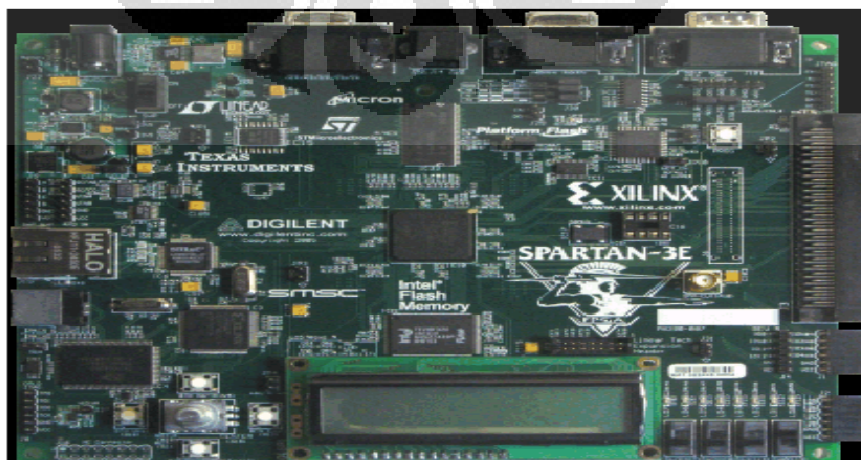
1. Look Up Table (LUT) merupakan sejenis RAM (Random Access Memory) yang berkapasitas kecil. Di dalam FPGA, LUT ini memegang peranan penting dalam proses implementasi fungsi-fungsi logika. Selain itu, LUT ini berciri khas memiliki input sejumlah 4 buah.
2. D Flip Flop. Seperti yang diketahui, flip-flop (Bistable Multivibrator) adalah suatu rangkaian sel biner yang memiliki dua buah output yang saling berkebalikan keadaannya (0 atau 1). Di dalam FPGA, terdapat sebuah jenis flip-flop yaitu D flip-flop atau Data flip flop. Rangkaian D flip-flop ini berfungsi sebagai rangkaian logika sekuensial dimana di dalamnya terdapat peralatan memori dan pewaktu.

3. Multiplexer 2 ke 1. Sebuah multiplexer adalah piranti digital yang bekerja sebagai switch (saklar) yang menghubungkan data dari n masukkan ke sebuah keluaran. Multiplexer berfungsi untuk memilih beberapa input untuk hanya menjadi 1 output saja. Di dalam FPGA, terdapat rangkaian multiplexer 2 ke 1 yang artinya, multiplexer tersebut memiliki 2 buah input dan 1 buah output.

Setiap logic cell tersebut dapat dihubungkan dengan logic cell lainnya melalui jalur/koneksi yang ada. Setiap cell hanya mampu bekerja secara sederhana dan ringkas, Namun bila antara satu cell saling terhubung satu sama lain sebuah fungsi-fungsi logika yang kompleks pun dapat terbentuk. [2]

2.3.4 FPGA Design Software

Field Programmable Gate Array (FPGA) merupakan sebuah IC digital yang sering digunakan untuk mengimplementasikan rangkaian digital. Jadi, bentuk utama sebuah FPGA adalah Integrated Circuit (IC). Dua buah perusahaan yang cukup terkenal sebagai pembuat FPGA adalah Xilinx dan Altera. Pada umumnya, perusahaan tersebut tidak menjual FPGA dalam bentuk terpisah yakni keping IC saja melainkan sudah terintegrasi dengan komponen-komponen elektronik lain seperti kristal, LED, resistor, kapasitor, push button, dsb. Semuanya itu biasanya menjadi satu dan disebut papan pengembang FPGA (FPGA development Board). Bentuk inilah yang biasanya dijual oleh para vendor pembuat FPGA. Berikut Gambar 2.8 adalah FPGA development board :



Gambar 2.8 Salah Satu Development Board milik Xilinx[2]

Perusahaan-perusahaan pembuat FPGA kemudian tidak hanya papan pengembang FPGA saja. Akan tetapi, perusahaan tersebut juga menjual sekaligus perangkat lunak (software) yang akan mendukung penggunaan. Jadi ketika kita membeli FPGA maka yang dimaksud adalah membeli development board FPGA beserta perangkat lunak pendukungnya. Kedua bagian ini tidak bisa terpisah. Tidak bisa hanya membeli papan pengembang FPGA saja atau sebaliknya, perangkat lunaknya saja. Kedua bagian tersebut merupakan satu kesatuan utuh yang saling mendukung proses kerja satu sama lain. Secara garis besar, perancangan perangkat lunak (software) terdiri atas 4 tahap yakni:

- Design-entry.
- Simulation.
- Synthesis and place-and-route.
- Programming through special cables (JTAG).

Perusahaan pembuat FPGA pada umumnya memberikan perangkat lunak secara cuma-cuma alias gratis. Software ini digunakan untuk mendukung proses design entry, simulation, synthesis and place-and-route, dan Programming through special cables (JTAG). Biasanya software yang dibagikan secara gratis hanya untuk jenis FPGA tingkat rendah-menengah atau hanyalah berupa demo saja. Sedangkan untuk jenis FPGA tingkat atas maka software pendukungnya tidaklah gratis melainkan berbayar. Berikut beberapa software pendukung yang gratis atau demo bagi FPGA :

- Perusahaan Xilinx terkenal dengan software miliknya yang bernama ISE WebPack
- Perusahaan Altera terkenal dengan software miliknya yang bernama Quartus II Web Edition

Software di atas cukup bagus dan baik untuk memulai belajar menggunakan FPGA karena software tersebut memiliki fungsi yang hampir sama dengan yang berbayar.

2.4 VHDL

VHDL adalah Very High-Speed Integrated Circuit (VHSIC) Hardware Description Language. VHDL adalah salah satu bahasa pemrograman yang

digunakan untuk memodelkan sistem digital dengan pemodelan data flow, behavioral dan struktural. Bahasa ini pertama kali diperkenalkan di 1981 oleh Departemen Pertahanan USA (Department of Defense (DoD) dibawah program VHSIC). Selanjutnya pada tahun 1983 IBM, Texas instruments dan Intermetrics mulai mengembangkan bahasa pemrograman ini, sehingga diluncurkanlah VHDL versi 7.2 diluncurkan di tahun 1985. Dan akhirnya di tahun 1987, bahasa pemrograman ini distandarisasi IEEE (Institute of Electrical and Electronic Engineering). Inilah evolusi berdasarkan sumber dari Universitas Waterloo :

1981, VHDL diusulkan sebagai bahasa deskripsi hardware

1986, VHDL diusulkan sebagai standar IEEE

1987, Standar pertama VHDL (IEEE-1076-1987)

1993, Standar VHDL direvisi (IEEE-1076-1993)

2002, Standar VHDL sekarang (IEEE-1076-2002), dan

Sekarang digunakan dengan luas oleh kalangan industri dan akademi, dengan penambahan IEEE-1164-1993 untuk mengenalkan nilai sistem logika

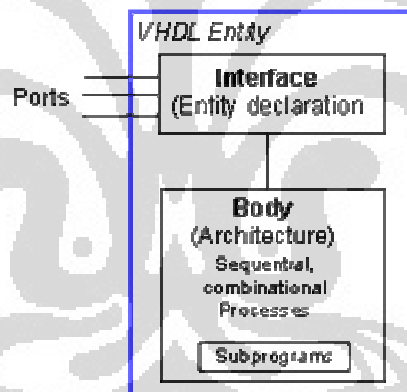
VHDL merupakan singkatan dari VHSIC Hardware Description Language (Bahasa Pemrograman hardware VHSIC). VHDL merupakan bahasa pemrograman hardware/alat/piranti keras yang memiliki fungsi tambahan dari bahasa pemrograman hardware sebelumnya yaitu PLD dan netlist. Hampir semua software VHDL synthesis mendukung bahasa pemrograman VHDL. Bahasa pemrograman VHDL menganut standart yang ditetapkan IEEE.

VHSIC sendiri merupakan IC dengan kecepatan sangat tinggi (Very High Speed Integrated Circuit). Bahasa pemrograman HDL dapat digunakan pada beberapa cara dan dapat dijadikan jalan alternatif untuk merepresentasikan diagram rangkaian untuk rangkaian digital ataupun untuk sebagian sub sistem rangkaian digital. Dua cara untuk memodelkan sistem digital yaitu secara struktur maupun secara behavioral (penjelasan output sistem). [3]

VHDL dapat digunakan untuk dokumentasi, verifikasi, dan sintesis dari perancangan sistem digital. Ini merupakan salah satu keuntungan dari VHDL, dimana kode VHDL yang sama dapat melakukan tiga fungsi di atas, sehingga lebih sederhana dan mengurangi kesalahan dalam penerjemahan sistem yang akan diimplementasikan. Selain itu, VHDL dapat dijalankan dengan tiga cara yang

berbeda untuk mendeskripsikan hardware. Tiga cara tersebut yaitu secara struktural, aliran data, dan penjelasan input output sistem. Biasanya untuk sistem yang besar penggabungan tiga cara ini diterapkan dan memiliki penerjemahan yang berbeda untuk cara yang berbeda pula.

Sebuah sistem digital di VHDL terdiri dari sebuah entitas desain yang dapat berisi banyak entitas lainnya yang kemudian dianggap sebagai komponen (gabungan) dari entitas tingkat atas (top-level). Setiap entitas dimodelkan oleh deklarasi entitas dan tubuh arsitektur. Anda dapat membayangkan bahwa deklarasi entitas sebagai antarmuka (interface) ke dunia luar yang mendefinisikan sinyal input dan output, sedangkan tubuh arsitektur berisi deskripsi entitas dan terdiri dari : entitas yang saling berhubungan, proses dan komponen, semua operasi yang bersamaan (concurrent), seperti skematis yang ditunjukkan pada Gambar 2.9 di bawah. Dalam desain yang umum, akan ada banyak entitas semacam itu dihubungkan bersama untuk melakukan fungsi yang diinginkan.



Gambar 2.9 Sebuah entitas VHDL terdiri dari interface (entitas deklarasi) dan tubuh (deskripsi arsitektur) [3]

VHDL menggunakan kata kunci tertentu (reserved keywords) yang tidak boleh/dapat digunakan sebagai nama sinyal atau pengidentifikasi (identifier). Kata kunci dan pengidentifikasi yang didefinisikan pengguna/programmer TIDAK case sensitive (huruf-besar atau huruf-kecil diperlakukan sama). Baris komentar dimulai dengan dua tanda hubung berdekatan (-) dan akan diabaikan oleh compiler. VHDL juga mengabaikan jeda baris dan ruang ekstra (spasi, tab). VHDL adalah bahasa yang sangat menekankan jenis (type) data (strongly typed),

yang menyiratkan bahwa seseorang harus selalu menyatakan jenis (type) setiap objek yang dapat memiliki nilai, seperti sinyal, konstanta dan variabel.

2.5 Light Dependent Resistor (LDR)

Light Dependent Resistor (LDR), yaitu resistor yang besar resistansi-nya bergantung terhadap intensitas cahaya yang menyelimuti permukaannya. LDR, dikenal dengan banyak nama: foto-resistor, foto-konduktor, sel foto-konduktif, atau hanya foto-sel. Dan yang sering digunakan dalam literatur adalah foto-resistor atau foto-sel.

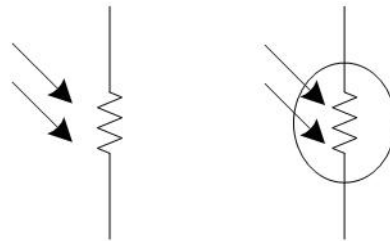
2.5.1 Sejarah Penemuan Foto-resistor / LDR

Foto-resistor, telah sangat banyak digunakan selama bertahun-tahun. Foto-resistor telah terlihat di bentuk awal sejak abad kesembilan belas ketika foto-konduktivitas akan selenium ditemukan oleh Smith pada tahun 1873. Sejak itu banyak varian perangkat foto-konduktif telah dibuat. Banyak pekerjaan yang berguna dilakukan oleh Case TW pada tahun 1920 ketika ia menerbitkan sebuah makalah berjudul “Thalofide Cell – a new photo-electric cell“.

Zat-zat lainnya termasuk PbS, PbSe, dan PbTe dipelajari di tahun 1930-an dan 1940-an, dan kemudian pada tahun 1952, Rollin dan Simmons mengembangkan foto-konduktor menggunakan silikon dan germanium.

2.5.2 Simbol Foto-resistor / LDR

Simbol rangkaian seperti pada Gambar 2.10 yang digunakan untuk Foto-resistor atau LDR adalah penggabungan resistor dan penunjukkan bahwa resistor tersebut sensitif terhadap cahaya. Simbol dasar Foto-resistor / LDR memiliki persegi panjang yang digunakan untuk menunjukkan fungsi resistansi-nya, dan kemudian memiliki dua panah masuk, sama seperti yang digunakan untuk fotodiode dan foto-transistor, untuk menunjukkan sensitivitasnya terhadap cahaya. Sebagian menggunakan lingkaran pada resistor-nya, sebagian lagi tidak. Simbol Foto-resistor atau LDR yang lebih umum digunakan adalah resistor tanpa lingkaran di sekitarnya.



Gambar 2.10 Simbol dari LDR[4]

2.5.3 Mekanisme Foto-resistor / LDR

Sebuah Foto-resistor atau LDR adalah komponen yang menggunakan foto-konduktor di antara dua pin-nya. Saat permukaannya terpapar cahaya akan terjadi perubahan resistansi di antaranya. Mekanisme di balik Foto-resistor atau LDR adalah foto-konduktivitas, yaitu suatu peristiwa perubahan nilai konduktansi bahan semikonduktor saat energi foton dari cahaya diserap olehnya. Ketika digunakan sebagai Foto-resistor atau LDR, bahan semikonduktor hanya digunakan sebagai elemen resistif dan tidak ada koneksi PN-nya. Dengan demikian, Foto-resistor atau LDR adalah murni komponen pasif.

2.5.4 Aplikasi Foto-resistor / LDR

Foto-resistor atau LDR berguna sebagai elemen sensitif cahaya berbiaya rendah dan digunakan selama bertahun-tahun dalam fotografi sebagai pengukur intensitas cahaya serta dalam aplikasi lain seperti detektor api / asap / pencuri, pembaca kartu, dan kendali lampu jalan berdasarkan cahaya.

2.5.5 Rangkaian Elektronik Foto-resistor / LDR

Rangkaian elektronik yang dapat digunakan untuk Foto-resistor atau LDR adalah rangkaian yang dapat mengukur nilai resistansi dari Foto-resistor / LDR tersebut. Dari hukum ohm, diketahui bahwa:

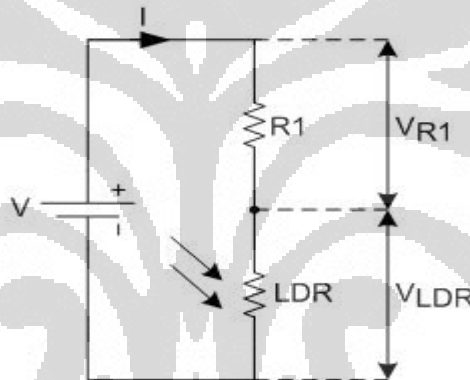
$$V = I.R$$

Dengan V adalah beda potensial antara dua titik, I adalah arus yang mengalir di antara-nya, dan R adalah resistansi di antara-nya. Lebih lanjut dikatakan pula bahwa nilai R tidak bergantung dari V ataupun I. Sehingga, jika ada perubahan

nilai resistansi dari R, maka nilai tegangan V-nya pun akan berubah. Jika beda potensial di-set tetap, maka perubahan resistansi hanya akan mempengaruhi besar arusnya. Dan persamaan tersebut akan menjadi:

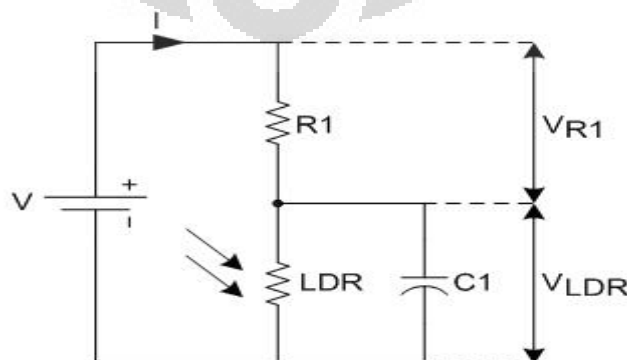
$$I = V / R$$

Kedua persamaan tersebut dapat dimanfaatkan sebagai rangkaian yang dapat mendeteksi perubahan resistansi dari Foto-resistor atau LDR. Pada persamaan pertama, nilai V akan berubah jika resistansi berubah, sedangkan pada persamaan kedua, nilai I yang akan berubah. Namun, pada banyak mikrokontroler, telah terintegrasi rangkaian ADC yang dapat membaca tegangan (V) analog dengan baik. Sehingga pada pembahasan, rangkaian pembacaan nilai resistansi dari Foto-resistor atau LDR adalah yang berdasar pada persamaan pertama.



Gambar 2.11 Rangkaian LDR[4]

Dengan rangkaian sederhana seperti Gambar 2.11 di atas, intensitas cahaya dapat diukur dengan mengukur nilai tegangan VLDR (dalam volt). Karena intensitas cahaya akan mempengaruhi nilai resistansi LDR yang dengan demikian akan mempengaruhi pula nilai VLDR.



Gambar 2.12 Penambahan Kapasitor[4]

Selanjutnya tambahkan kapasitor pada LDR seperti pada Gambar 2.12 di atas. Dengan penambahan kapasitor, nilai VLDR tidak akan berubah secara signifikan. Tetapi respon terhadap perubahan intensitas memang sedikit lebih lambat. Namun, dengan kapasitor tersebut, tegangan VLDR akan lebih stabil. Dengan pemilihan nilai kapasitor yang tepat (0.1 μF – 1 μF), respon terhadap perubahan tetap baik, dan akan didapatkan tegangan VLDR yang stabil.

2.6 (Light Emitting Diode) LED

LED atau singkatan dari Light Emitting Diode adalah salah satu komponen elektronik yang tidak asing lagi di kehidupan manusia saat ini. LED saat ini sudah banyak dipakai, seperti untuk penggunaan lampu permainan anak-anak, untuk rambu-rambu lalu lintas, lampu indikator peralatan elektronik hingga ke industri, untuk lampu emergency, untuk televisi, komputer, pengeras suara (speaker), hard disk eksternal, proyektor, LCD, dan berbagai perangkat elektronik lainnya sebagai indikator bahwa sistem sedang berada dalam proses kerja, dan biasanya berwarna merah atau kuning. LED ini banyak digunakan karena konsumsi daya yang dibutuhkan tidak terlalu besar dan beragam warna yang ada dapat memperjelas bentuk atau huruf yang akan ditampilkan. dan banyak lagi

Pada dasarnya LED itu merupakan komponen elektronika yang terbuat dari bahan semi konduktor jenis dioda yang mampu memancarkan cahaya. LED merupakan produk temuan lain setelah dioda. Strukturnya juga sama dengan dioda, tetapi belakangan ditemukan bahwa elektron yang menerjang sambungan P-N. Untuk mendapatkna emisi cahaya pada semikonduktor, doping yang pakai adalah galium, arsenic dan phosporus. Jenis doping yang berbeda menghasilkan warna cahaya yang berbeda pula. [5]

Keunggulannya antara lain konsumsi listrik rendah, tersedia dalam berbagai warna, murah dan umur panjang. Keunggulannya ini membuat LED digunakan secara luas sebagai lampu indikator pada peralatan elektronik. Namun LED punya kelemahan, yaitu intensitas cahaya (Lumen) yang dihasilkannya termasuk kecil. Kelemahan ini membatasi LED untuk digunakan sebagai lampu penerangan. Namun beberapa tahun belakangan LED mulai dilirik untuk keperluan penerangan, terutama untuk rumah-rumah di kawasan terpencil yang

menggunakan listrik dari energi terbarukan (surya, angin, hidropower, dll). Alasannya sederhana, konsumsi listrik LED yang kecil sesuai dengan kemampuan sistem pembangkit energi terbarukan yang juga kecil.

Riset-riset mutakhir menunjukkan hasil menggembirakan. Kini LED mampu menghasilkan cahaya besar dengan konsumsi energi listrik (tetap) kecil. Berita terakhir adalah ditemukannya OLED (Organic LED) oleh para ilmuwan di University of Michigan dan Princeton University. Temuan ini sukses menghasilkan cahaya dengan intensitas 70 Lumen setiap 1 watt listrik yang digunakan. Sebagai perbandingan, lampu pijar memancarkan 15 lumen per watt, dan lampu fluorescent (misalnya lampu jantung) memancarkan 90 lumen per watt. Keunggulan LED dibanding lampu fluorescent adalah ramah lingkungan, cahaya tajam, umur panjang, dan murah.

Sebelum OLED ditemukan, persoalan yang dihadapi para ahli LED adalah rendahnya efisiensi LED. Bukan karena cahaya yang dihasilkan sedikit, tapi karena sekitar 80% cahaya terperangkap di dalam LED. Sebagai solusi, disain OLED menggunakan kombinasi kisi dan cermin berukuran mikro, bekerja bersama-sama memandu cahaya yang terperangkap di dalam LED keluar. Stephen Forrest, profesor teknik elektro dan fisika di University of Michigan, penemu OLED mengatakan bahwa kini kita bisa bersiap untuk mengganti pencahayaan di dalam bangunan dan rumah yang saat ini menggunakan lampu pijar ataupun fluorescent dengan OLED. [5]

2.6.1 Macam-macam LED

2.6.1.1 Diode Emiter Cahaya

Sebuah dioda emisi cahaya dapat mengubah arus listrik langsung menjadi cahaya. Dengan mengubah-ubah jenis dan jumlah bahan yang digunakan untuk bidang temu PN. LED dapat dibentuk agar dapat memancarkan cahaya dengan panjang gelombang yang berbeda-beda. Warna yang biasa dijumpai adalah merah, hijau dan kuning.

2.6.1.2 LED Warna Tunggal

LED warna tunggal adalah komponen yang paling banya dijumpai. Sebuah LED warna tunggal mempunyai bidang temu PN pada satu keping silicon. Sebuah lensa menutupi bidang temu PN tersebut untuk memfokuskan cahaya yang dipancarkan.

2.6.1.3 LED Tiga Warna Tiga Kaki

Satu kaki merupakan anoda bersama dari kedua LED. Satu kaki dihubungkan ke katoda LED merah dan kaki lainnya dihubungkan ke katoda LED hijau. Apabila anoda bersamanya dihubungkan ke bumi, maka suatu tegangan pada kaki merah atau hijau akan membuat LED menyala. Apabila satu tegangan diberikan pada kedua katoda dalam waktu yang bersama, maka kedua LED akan menyala bersama-sama. Pencampuran warna merah dan hijau akan menghasilkan warna kuning.

2.6.1.4 LED Tiga Warna Dua Kaki

Satu kaki merupakan anoda bersama dari kedua LED. Satu kaki dihubungkan ke katoda LED merah dan kaki lainnya dihubungkan ke katoda LED hijau. Apabila anoda bersamanya dihubungkan ke bumi, maka suatu tegangan pada kaki merah atau hijau akan membuat LED menyala. Apabila satu tegangan diberikan pada kedua katoda dalam waktu yang bersama, maka kedua LED akan menyala bersama-sama. Pencampuran warna merah dan hijau akan menghasilkan warna kuning.

2.6.1.5 Led Seven Segmen

Biasanya digunakan untuk menampilkan angka berupa angka 0 sampai 9, angka – angka tersebut dapat ditampilkan dengan mengubah nyala dari 7 segmen yang ada pada led.

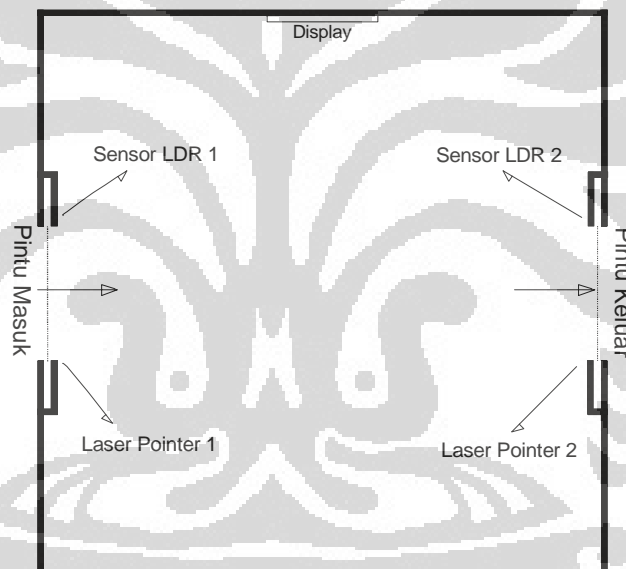
BAB 3

CARA KERJA SISTEM DAN PERANCANGAN PROGRAM

Pada bab ini akan dibahas mengenai perancangan sistem beserta cara kerja sistem dan perancangan software yang digunakan penulis dalam penyusunan pengendalian dari alat ini.

3.1 Desain Sistem

Ruangan membutuhkan sistem keamanan yang baik serta dapat memberikan kenyamanan bagi pengunjung. Untuk itu dibutuhkan sebuah sistem yang dapat memenuhi kebutuhan tersebut. Asumsikan sebuah ruangan dengan ketentuan seperti Gambar 3.1.



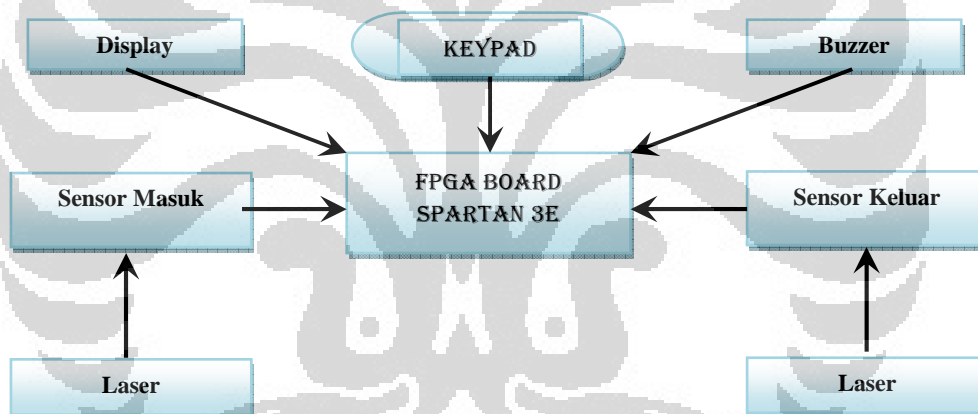
Gambar 3.1 Desain Sistem

Berdasar gambar 3.1 pengunjung masuk dalam suatu ruangan akan melewati laser yang ditembakkan ke sensor masuk yang berupa sensor LDR. Sensor LDR yang tadinya bernilai 1 akan berubah menjadi 0 ketika laser terhalang oleh gerakan pengunjung yang lewat menutupi. Perubahan nilai dari 1 ke 0 tersebut akan dibaca oleh FPGA sebagai input. Input tersebut akan diolah menjadi counter yang akan menambahkan/mengurangkan jumlah perhitungan. Jumlah perhitungan tersebut akan ditampilkan kedalam display yang membuat penambahan jumlah

pengunjung pada display. Jika jumlah melebihi kapasitas ruangan maka buzzer akan berbunyi.

3.2 Perancangan Sistem

Berdasarkan Gambar 3.2 diagram perancangan alat dibawah ini akan dibahas bagaimana sistem kerja dari menghitung jumlah pengunjung dengan FPGA. Pada awal pengunjung masuk dalam suatu ruangan akan melewati laser yang ditembakkan ke sensor masuk yang berupa sensor LDR. Sensor LDR yang tadinya bernilai 1 akan berubah menjadi 0 ketika laser terhalang oleh gerakan pengunjung yang lewat menutupi. Perubahan nilai dari 1 ke 0 tersebut akan dibaca oleh FPGA sebagai input. Input tersebut akan diolah menjadi counter yang akan menambahkan jumlah perhitungan. Jumlah perhitungan tersebut akan ditampilkan kedalam display yang membuat penambahan jumlah pengunjung pada display.



Gambar 3.2 Diagram Perancangan Alat

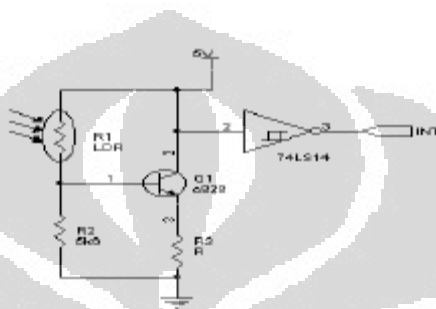
Pada sensor keluar pun terjadi demikian perubahan sensor LDR dari 1 ke 0 yang merupakan input bagi FPGA. Input tersebut bukan untuk menambahkan perhitungan counter akan tetapi akan mengurangi perhitungan counter. Pengurangan tersebut akan menguraingi jumlah pengunjung yang berada pada layar display.

Dalam diagram tersebut terdapat buzzer yang berfungsi sebagai peringatan bahwa ruangan tersebut akan atau sudah penuh dan bahkan melebihi kapasitas. Selain bazzer berbunyi yaitu pemberitahuan bahwa kapasitas nya sudah penuh.

Tombol yang terdapat dalam diagram tersebut berfungsi sebagai pengatur dari kapasitas ruangan untuk settingan awal pada program.

3.3 Rangkaian Sensor

Rangkaian sensor menggunakan LDR yang mampu merespon segala macam cahaya. Digunakan LDR karena luas penampang LDR lebih besar dibandingkan dengan menggunakan phototransistor. Adapun Gambar 3.3 rangkaian sebagai berikut :



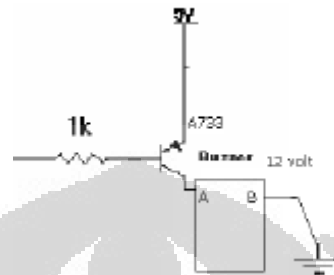
Gambar 3.3 Perancangan Rangkaian Sensor[6]

Rangkaian ini bekerja dengan perbandingan RB antara R2 dan R1, jika $R2 < R1$ maka arus terhubung ground. Dan jika $R1 < R2$ maka arus akan menuju ke basis transistor menyebabkan transistor hidup dan tegangan 5V akan terhubung ground karena transistor melewatkan arus dari kolektor ke emitor menuju ground. Jika Transistor mati seolah-olah menjadi hambatan yang besar sehingga tegangan dari sumber akan menuju IC. Tegangan out put tersebut dimantapkan oleh IC 7414 sebagai pembalik logika dan pemantap. Jika tegangan keluaran dari transistor sekitar 0,13V sampai 2,4 V maka tegangan tersebut oleh IC 7414 dianggap sebagai logika 0, dan jika tegangan sekitar 2,56V sampai 5V dianggap sebagai logika 1. Output keluaran dari IC 7414 dihubungkan dengan pin pada FPGA sebagai masukan sensor.

3.4 Rangkaian Buzzer

Rangkaian penggerak buzzer ini menggunakan transistor A733 sebagai saklar yang akan diberi input. Transistor Bekerja jika tegangan emitor lebih positif dari tegangan basis $V_e > V_b$ dengan cara memberi logika 0 pada input maka transistor akan bekerja atau saklar tertutup dan arus dari emitor akan menuju kolektor yang

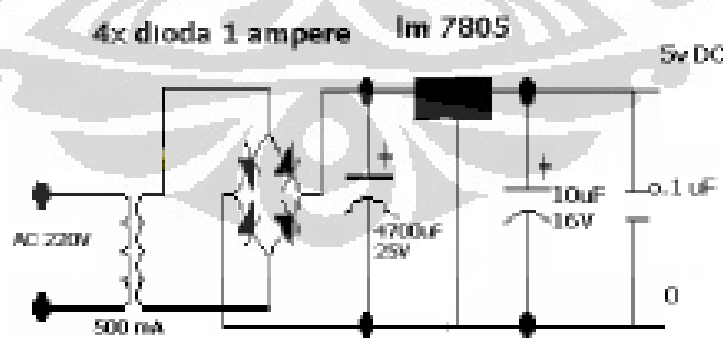
terhubung dengan buzzer menyebabkan buzzer 12 Volt berbunyi. Sedangkan resistor 1k ohm digunakan sebagai pembatas arus bocor dari emitor supaya tidak terhubung langsung dengan mikrokontroler. Adapun Gambar 3.4 rancangan penggerak buzzer adalah sebagai berikut:



Gambar 3.4 Perancangan Rangkaian Buzzer[6]

3.5 Catu Daya

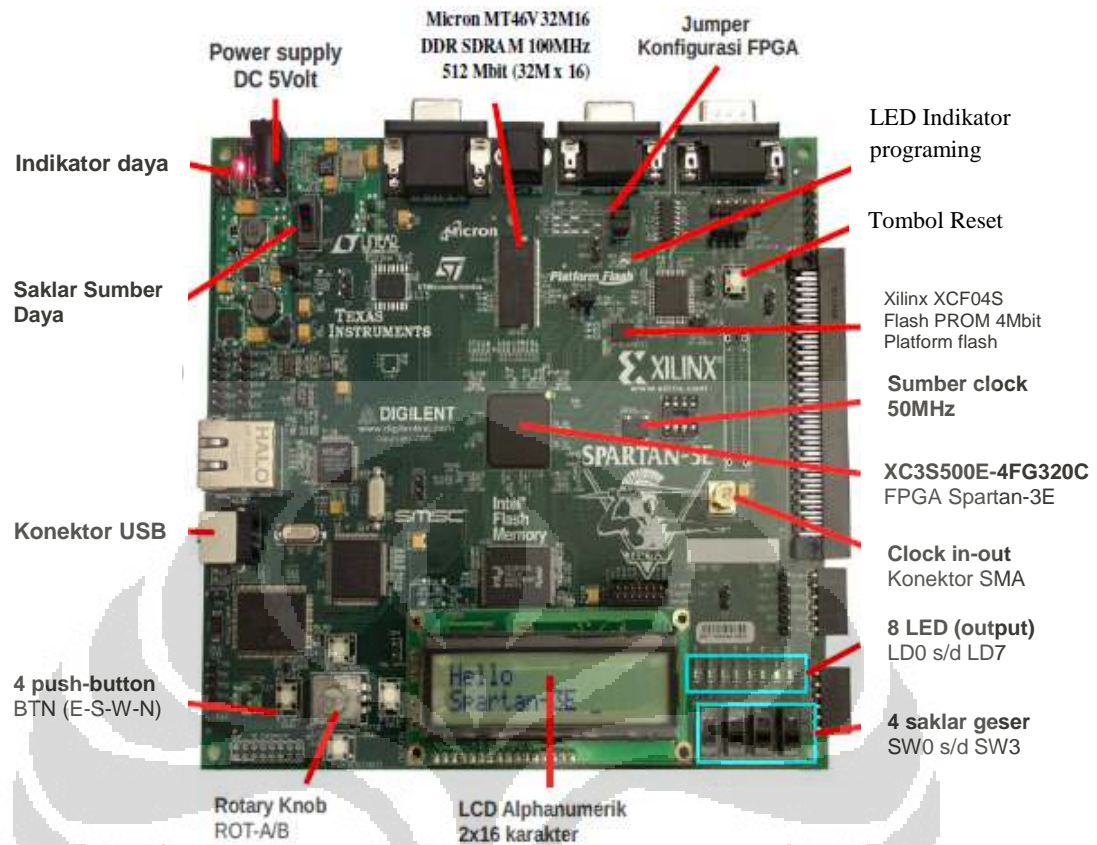
Supaya alat dapat bekerja harus diberi tegangan sumber sebesar 5Vdc. Rangkaian catudaya menggunakan transformator 500mA dan pada keluaran 6Vac dipasang empat buah dioda 1ampere yang disusun menjadi jembatan wheatstone sebagai penyearah gelombang dan mengubah tegangan ac menjadi dc. Untuk memperoleh tegangan yang setabil maka dipasang IC LM7805 sebagai penyetabil tegangan 5Vdc, IC ini memiliki ketahanan arus normal maksimal sebesar 1 ampere. Pada keluaran IC dipasang kapasitor sebesar 10uF/16V dan 0,1uF sebagai filter tegangan AC dari transformator. Berikut Gambar 3.5 rangkaian catu daya.



Gambar 3.5 Perancangan Rangkaian Power Supply[6]

3.6 Modul FPGA SPARTAN 3E

Board starter kit Spartan-3E akan digunakan dalam perancangan ini. Board ini menggunakan Xilinx FPGA Spartan-3E (XC3S500E-4FG320C) yang mempunyai 500K gerbang dan tersusun atas komponen-komponen (Gambar 3.6).



Gambar 3.6 Board Starter Kit Xilinx Spartan-3E beserta komponen-komponennya[7]

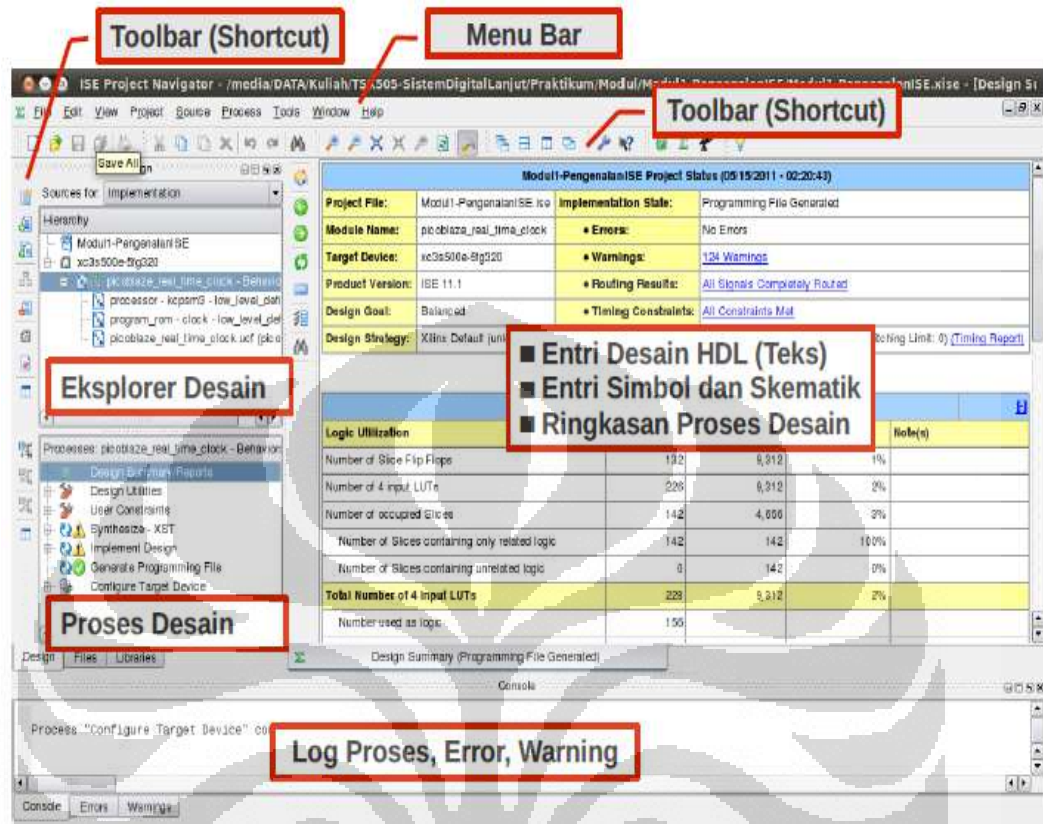
3.7 Software Xilinx ISE Webpack 13.2

Software Xilinx ISE Webpack 13.2 akan digunakan sebagai GUI untuk merancang problem sistem digital yang diinginkan. Selanjutnya Xilinx ISE Webpack 13.2 disebut sebagai ISE. ISE berisi tool-tool untuk mengembangkan rancangan sistem digital terprogram. Tool yang akan digunakan dalam perancangan ini adalah:

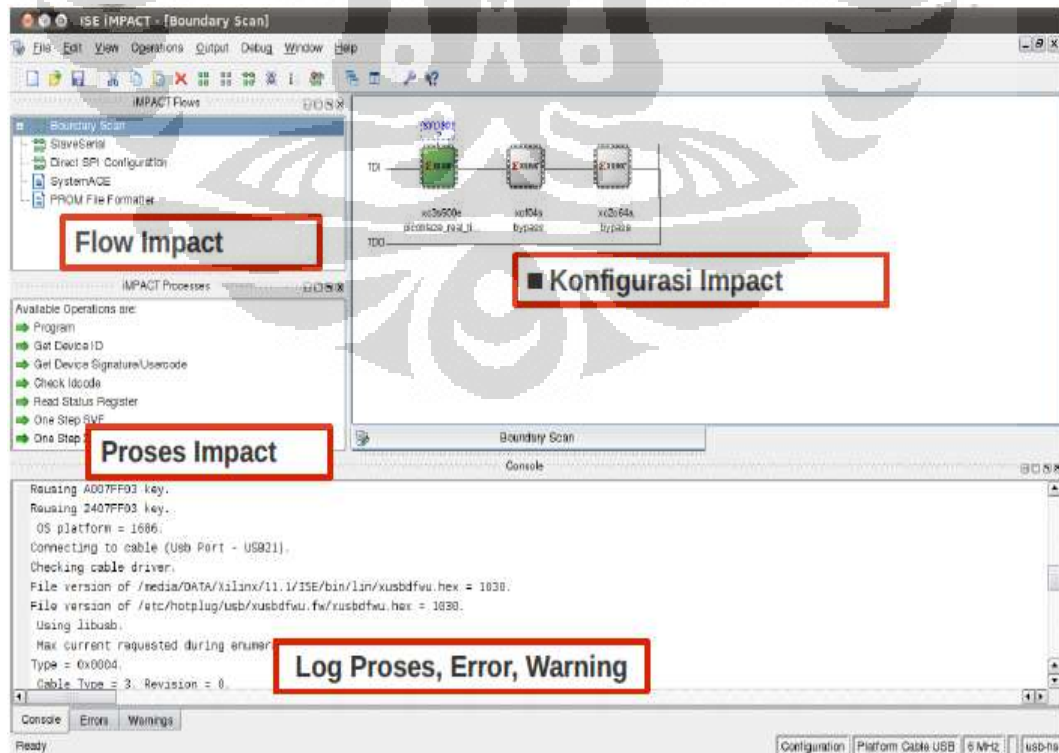
1. ISE Project Navigator digunakan untuk membuat proyek baru, memasukan entry desain (HDL, skematik), kompilasi desain (sintesis), mengimplementasikan desain dan membangkitkan file programming (konfigurasi FPGA). Tool lain dari ISE dipanggil dari Project Navigator ini;
2. ISE iMPACT digunakan untuk menuliskan file konfigurasi FPGA ke board Starter Kit;

GUI dari program ISE Project Navigator diperlihatkan dalam Gambar 3.7.

GUI dari program ISE iMPACT diperlihatkan dalam Gambar 3.8.



Gambar 3.7 ISE Project Navigator GUI: Eksplorer, Proses, Entry dan Log[7]

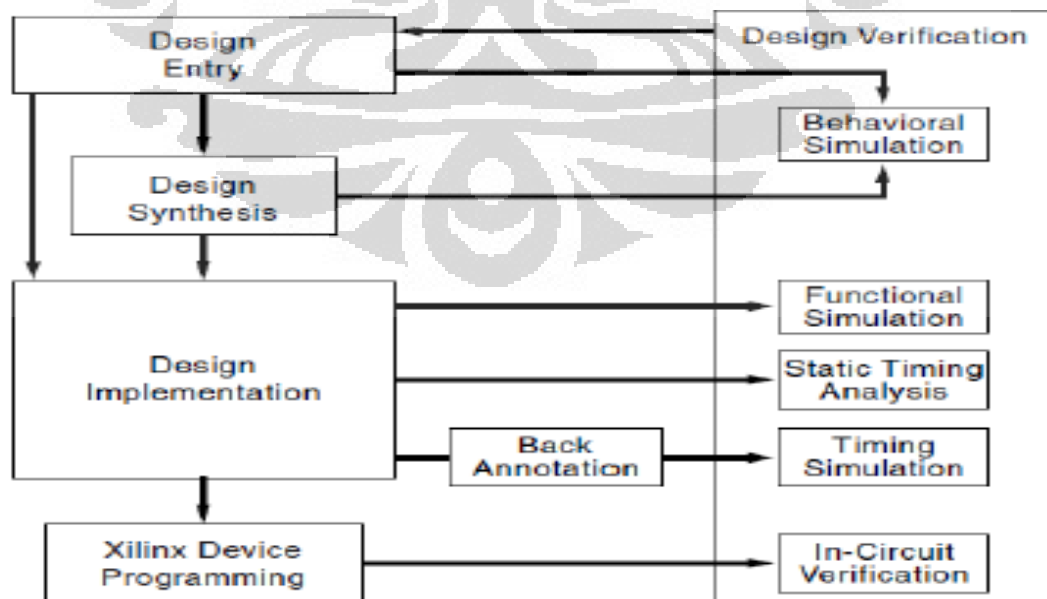


Gambar 3.8 ISE iMPACT: Konfigurasi Boundary Scan (JTAG) [7]

Perancangan sistem digital menggunakan ISE mempunyai metodologi (Gambar 3.9). Metodologi tersebut meliputi tahapan sebagai berikut:

1. Entri desain. Program HDL (Hardware Description Language) dibuat untuk memenuhi kebutuhan dan konstrain sistem yang ingin dirancang. Program ini mendeskripsikan struktural dan perilaku sistem. HDL yang dikenal ada 3, yaitu VHDL, Verilog dan Altera HDL. Rancangan ini memilih bahasa HDL yang akan digunakannya.
2. Sintesis. Dari desain HDL, skematik RTL (Register Transfer Level) dibangkitkan oleh ISE sesuai teknologi yang digunakan (Xilinx FPGA).
3. Implementasi. Tahap meliputi translating, mapping/fitting dan placing&routing untuk mengimplementasikan desain ke teknologi device Xilinx yang sesuai. Desain dioptimasi agar memenuhi kebutuhan fungsional dan konstrainnya (power, area, speed). Hasil implementasi adalah berupa file konfigurasi FPGA (dengan ekstensi *.bit) yang siap untuk diprogramkan ke device XC3S500E;
4. Pemrograman/download ke device Xilinx. Tahap ini dilakukan menggunakan program IMPACT. File konfigurasi didownload ke device melalui interface USB;

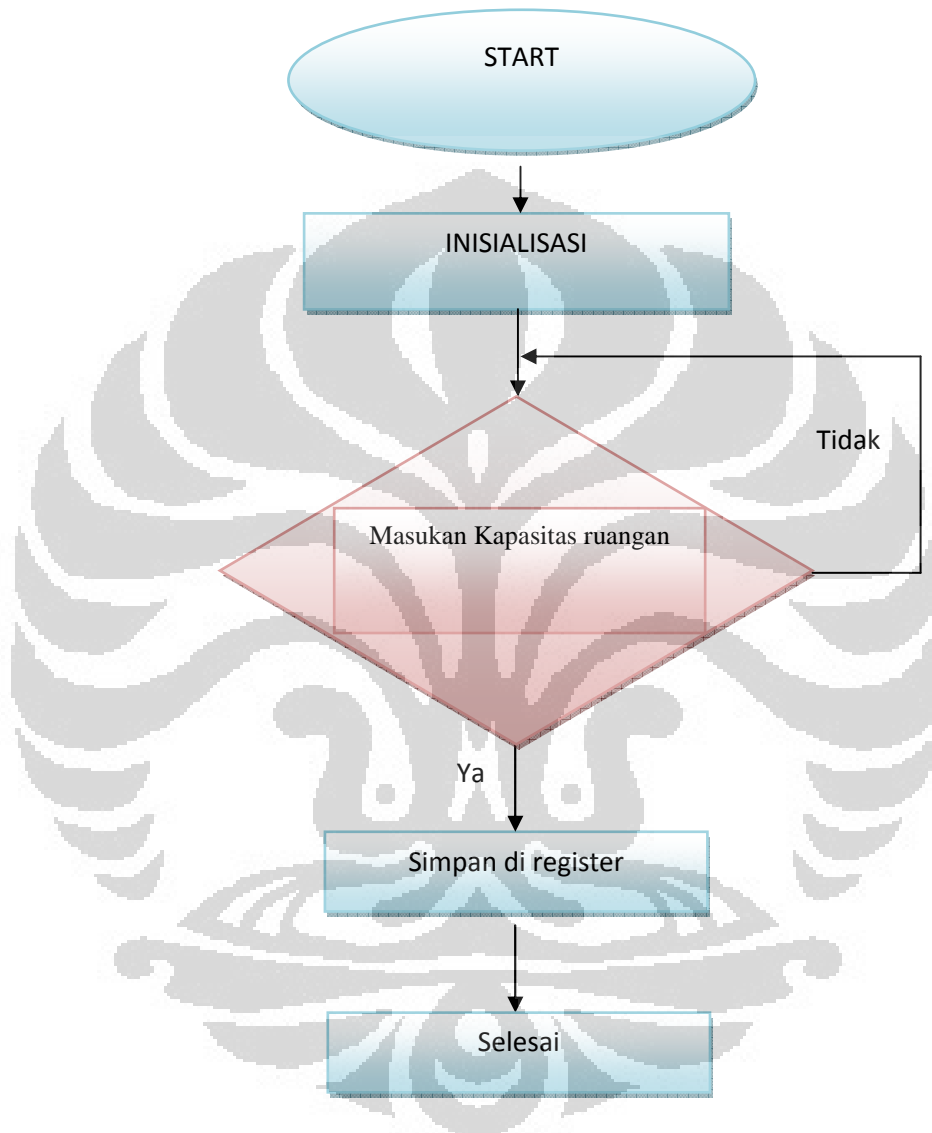
Dalam setiap tahap, verifikasi dilakukan untuk memastikan kebutuhan dan konstrain terpenuhi.



Gambar 3.9 Metodologi desain sistem digital menggunakan ISE: entri desain, sintesis, implementasi dan pemrograman[7]

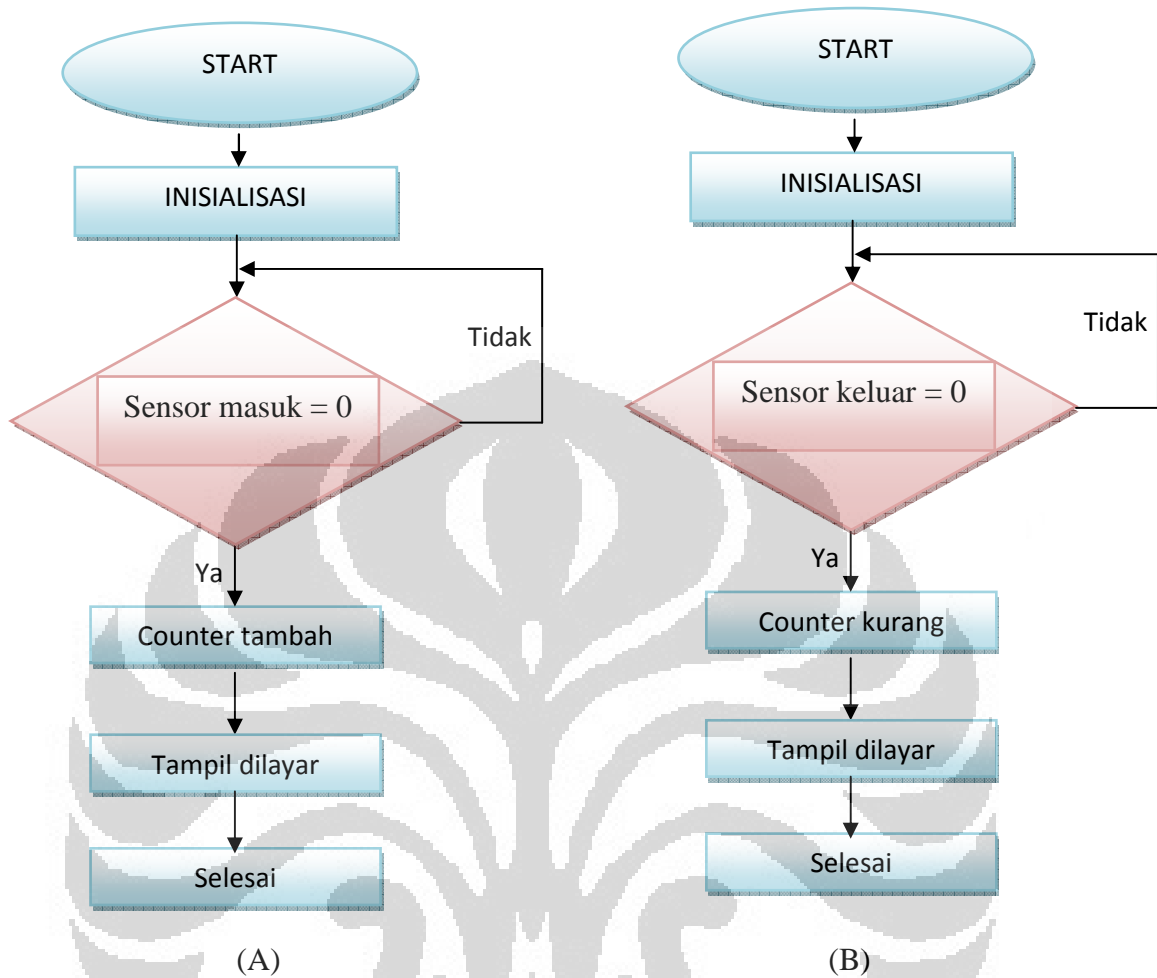
3.8 Perancangan Software

Pada bagian ini akan dijelaskan tentang perancangan software dari sistem yang telah dibuat termasuk flowchart program VHDL yang digunakan. Berikut adalah flowchart program :



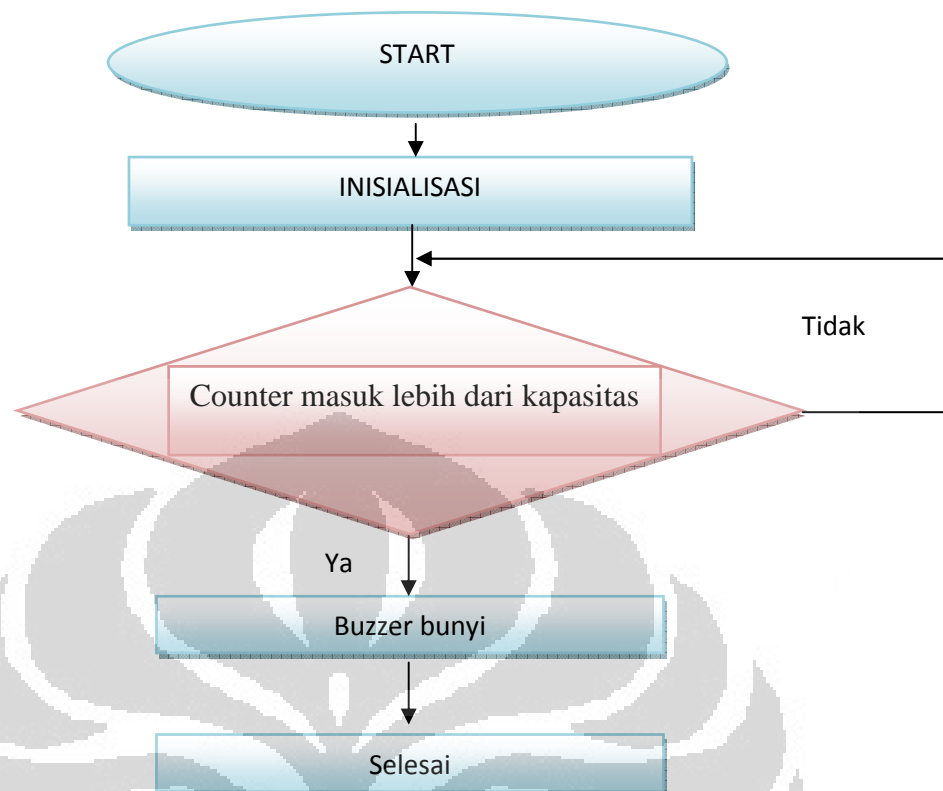
Gambar 3.10 Flowchart awal kapasitas

Pada Gambar 3.10 flowchart awal kapasitas maksudnya adalah menginput kapasitas ruangan yang akan di setting oleh FPGA untuk menentukan batas maksimum kapasitas dari ruangan tersebut. Nilai yang disimpan akan di masukan keregister untuk disimpan dan di bandingkan ketika counter pada sesor masuk bertambah.



Gambar 3.11 Flowchart (A) Sensor masuk; (B) Sensor keluar

Gambar 3.11 merupakan flowchart untuk counter masuk dan keluar serta akan menampilkan ke display berapa jumlah pengunjung yang ada dalam ruangan tersebut. Jika counter masuk melebihi register yang telah ditetapkan pada awal kapasitas maka display akan tertulis penuh dan buzzer akan berbunyi sesuai dengan Gambar 3.12 flowchart kapasitas telah penuh.



Gambar 3.12 Flowchat kapasitas telah penuh

BAB 4

IMPLEMENTASI DAN ANALISIS

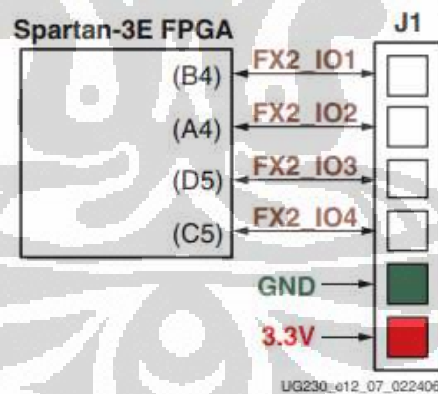
4.1 Implementasi

Implementasi Smart Counter: Penghitung Jumlah Pengunjung Ruangannya berbasis FPGA Xilinx Spartan 3E dibagi menjadi dua yaitu: Hardware dan Software. Sistem hardware membahas tentang semua peralatan yang digunakan berupa simulasi. Sedangkan sistem software membahas tentang pemrograman pada kinerja simulasi alat tersebut.

4.1.1 Hardware

4.1.1.1 FPGA

Simulasi yang digunakan dalam percobaan ini menggunakan FPGA Xilinx Spartan 3E. Port yang digunakan dalam implementasi ini adalah 3 soket konektor ekspansi. Sumber tegangan yang disediakan dalam FPGA adalah 3,3V. Koneksi FPGA ke header J1 yang berisi 6-pin yang terdapat pada slot konektor ekspansi dapat dijelaskan pada gambar 4.1.

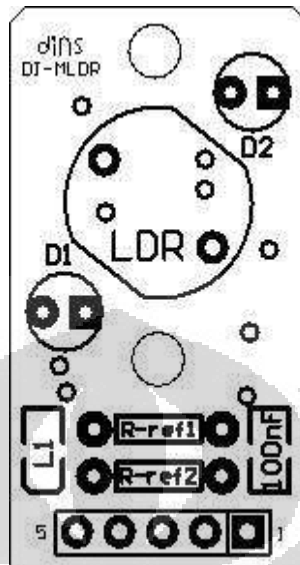


Gambar 4.1 Koneksi FPGA ke Header J1

4.1.1.2 DI-MLDR

DI-MLDR adalah modul sensor cahaya berbasis sensor LDR (Light Dependent Resistor) atau biasa juga disebut dengan photoresistor, yaitu komponen elektronika yang bersifat resistif yang resistansi berubah jika kecerahan cahaya yang mengenai permukaannya berubah. Modul sensor ini didisain untuk

banyak fungsi yang berkaitan dengan aplikasi cahaya. Layout dari rangkaian DI-MLDR dapat dilihat pada Gambar 4.5.



Gambar 4.5 Layout Rangkaian Sensor DI-MLDR

Sensor DI-MLDR ini memiliki spesifikasi sebagai berikut :

- Beroperasi pada tegangan : 3,5 – 5,5 VDC.
- Terdapat dua jenis keluaran (*output*):
 - Analog: 0 VDC s.d 5 VDC.
 - Digital : Logika 0 dan Logika 1.
- Terdapat dua buah LED (*Light Emitting Diode*) berwarna putih terang berdiameter 3 mm yang dapat dikendalikan nyala dan padamnya (*on/off*).
- Tingkat referensi pembanding kecerahan yang dapat diatur untuk keluaran digital.

Beberapa fungsi dari pin yang terdapat dalam rangkaian sensor DI-MLDR dapat dilihat pada Tabel 4.2.

Tabel 4.2 Fungsi PIN Sensor DI-MLDR

No.	PIN	Fungsi
1.	GND	Sumber tegangan (-) rangkaian
2.	VCC	Sumber tegangan (+) rangkaian
3.	ANA	Vout analog (0 – 5 VDC)
4.	CTRL	Kendali LED (0 = nyala; 1 = padam)
5.	COMP	Vout digital (logika 0 dan 1)

Pada simulasi percobaan ini menggunakan DI-MLDR untuk mendeteksi keberadaan orang. Dengan menambahkan lanser pointer sebagai sumber cahaya yang jaraknya diatur agar orang dapat berlalu-lalang diantara sensor dan laser pointer. Data dari COMP akan berlogika 0 (0 VDC) saat orang menghalangi cahaya yang menuju permukaan LDR ditandai dengan menyalnya LED indikator berwarna hijau dan COMP akan berlogika 1 (5 VDC) saat cahaya mengenai permukaan LDR. Dalam percobaan digunakan dua buah sensor DI-MLDR dan dua buah laser pointer sebagai pendeteksi di pintu masuk dan keluar.

4.1.1.3 Voltage Regulator

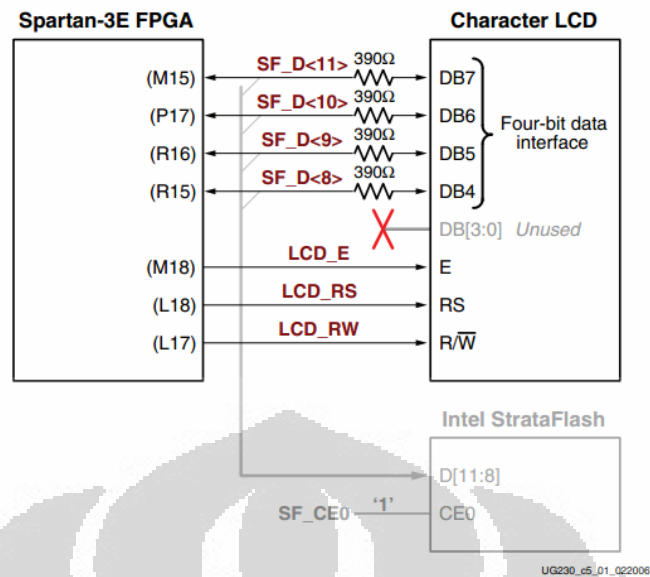
Dalam percobaan ini menggunakan Voltage Regulator 5V sebagai supply tambahan yang berasal dari arus AC 12V yang diubah menjadi arus DC sebesar 5V.

4.1.1.4 Buzzer

Buzzer digunakan dalam simulasi percobaan ini sebagai peringatan bahwa pengunjung dalam ruangan sudah penuh. Buzzer akan menyala jika COMP pada rangkaian Buzzer berlogika 0 menandakan ruangan telah penuh. Sedangkan COMP berlogika 1 jika ruangan tidak penuh.

4.1.1.5 LCD

LCD yang digunakan dalam sistem ini adalah LCD yang ada dalam FPGA dengan seri Sitronix ST7066U. LCD ini merupakan LCD karakter dengan jumlah karakter 16 x 2. Pada umumnya terdapat dua jenis antarmuka yang dapat digunakan dalam pengendalian LCD karakter, yaitu dengan 8 bit atau dengan 4 bit. Akan tetapi, pada LCD yang terdapat dalam FPGA hanya memungkinkan untuk pengendalian LCD karakter dengan 4 bit. Hal ini dikarenakan agar tetap kompatibel dengan pengembangan Xilinx lainnya dan untuk meminimalkan jumlah total pin. Karakteristik antarmuka LCD dapat dilihat pada gambar 4.2.



Gambar 4.2 Karakteristik Antarmuka LCD

Pada Tabel 4.1 dijelaskan karakteristik sinyal antarmuka LCD sebagai penjelasan Gambar 4.2.

Tabel 4.1 Sinyal Antarmuka LCD

Signal Name	Pin FPGA	Function	
SF_D<11>	M15	Data bit BD7	Shared with StrataFlash pins SF_D<11:8>
SF_D<10>	P17	Data bit BD6	
SF_D<9>	R16	Data bit BD5	
SF_D<8>	R15	Data bit BD4	
LCD_E	M18	Read/Write Enable Pulse 0: Disable 1: Read/Write operation enabled	
LCD_RS	L18	Register Select 0: Instruction register during write operations. Busy Flash during read operations 1: Data for read/write operations	
LCD_RW	L17	Read/Write Control 0: WRITE, LCD accepts data 1: READ, LCD presents data	

Kontroller LCD memiliki 3 bagian memori internal yang mempunyai fungsi masing-masing, yaitu DD RAM, CG RAM, dan CG ROM. Dalam menentukan huruf atau simbol yang akan ditampilkan pada layar dibutuhkan kode ASCII. ASCII (American Standar Codes for Internasional Interchange) adalah merupakan kumpulan kode-kode yang dipergunakan untuk interaksi antara user dengan

komputer. Kode ASCII merupakan kode angka sebanyak 255 buah, dimana setiap angka mempunyai karakter khusus. Lokasi set karakter kode ASCII dapat dilihat pada Gambar 4.3.

		Upper Data Nibble																
		DB7	DB6	DB5	DB4	0	0	0	0	0	0	0	1	1	1	1	1	1
		0	0	0	1	1	1	1	1	0	0	1	1	1	1	1	1	
		0	1	1	0	0	1	1	1	1	0	0	1	1	1	1		
		0	0	1	0	1	0	1	0	1	0	1	0	1	0	1		
Lower Data Nibble	xxxx0000			0	@	P	`	P		-	9	ε	α	ρ				
	xxxx0001	!	1	A	Q	a	q	。	ア	チ	△	ä	ç					
	xxxx0010	"	2	B	R	b	r	「	イ	ツ	×	β	θ					
	xxxx0011	CG RAM	#	3	C	S	c	s	」	ウ	〒	ε	ω					
	xxxx0100	CG	\$	4	D	T	d	t	、	エ	ト	μ	Ω					
	xxxx0101	%	5	E	U	e	u	・	オ	ナ	1	σ	ü					
	xxxx0110	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ					
	xxxx0111	'	7	G	W	g	w	フ	キ	ヲ	ラ	q	π					
	xxxx1000	(8	H	X	h	x	イ	ク	ネ	リ	ル	ア					
	xxxx1001)	9	I	Y	i	y	ウ	ケ	ル	レ	ウ						
	xxxx1010	*	:	J	Z	j	z	エ	コ	ハ	レ	i	チ					
	xxxx1011	+	;	K	[k	[オ	サ	ヒ	ロ	×	ア					
	xxxx1100	,	<	L	¥	l		パ	シ	フ	ワ	φ	ア					
	xxxx1101	-	=	M]	m]	ユ	ズ	ハ	ン	モ	÷					
	xxxx1110	.	>	N	^	n	^	ヨ	セ	ホ	ッ	ン						
	xxxx1111	/	?	O	_	o	_	ウ	ツ	マ	°	ö	■					

Gambar 4.3 Set Karakter Kode ASCII

Pada implementasi ini, kode ASCII digunakan untuk menampilkan tulisan pada layar LCD.

4.1.2 Implementasi Perangkat Lunak (*Software*)

4.1.2.1 Program Counter

Program counter pada percobaan ini digunakan untuk menghitung jumlah pengujung yang memperoleh signal dari sensor DI-MLDR.

```
if(clk'event and clk = '1') then
    if(counter1 = 50000) then
```

```

    Q1 <= sig1;
    Q2 <= Q1;
    Q3 <= Q2;
    Q4 <= Q3;
    Q5 <= Q4;
    Q6 <= Q5;
    counter1 <= 0;
else
    counter1 <= counter1 + 1;
end if;
if(counter2 = 50000) then
    S1 <= sig2;
    S2 <= S1;
    S3 <= S2;
    S4 <= S3;
    S5 <= S4;
    S6 <= S5;
    counter2 <= 0;
else
    counter2 <= counter2 + 1;
end if;
if(QR = '1')then
    sat_counter <= sat_counter + 1;
end if;

if(SR = '1')then
    sat_counter <= sat_counter - 1;

end if;

if sat_counter = "01010001" then
    puluhan<="0000";
    satuan<="0001";

elsif sat_counter = "10100010" then

```

```
puluhan<="0000";
satuan<="0010";

elsif sat_counter = "11110011" then
    puluhan<="0000";
    satuan<="0011";

elsif sat_counter = "01000100" then
    puluhan<="0000";
    satuan<="0100";

elsif sat_counter = "10010101" then
    puluhan<="0000";
    satuan<="0101";

elsif sat_counter = "11100110" then
    puluhan<="0000";
    satuan<="0110";

elsif sat_counter = "00110111" then
    puluhan<="0000";
    satuan<="0111";

elsif sat_counter = "10001000" then
    puluhan<="0000";
    satuan<="1000";

elsif sat_counter = "11011001" then
    puluhan<="0000";
    satuan<="1001";

elsif sat_counter = "00101010" then
    puluhan<="0001";
    satuan<="0000";

--10
```

4.1.2.2 Program LCD

Program LCD pada percobaan ini berfungsi sebagai tampilan jumlah batasan pengujung ruangan dan jumlah orang. Dengan menggunakan LCD yang terpasang pada FPGA Spartan 3E.

```

with cur_state select
    tx_byte <= "00101000" when function_set,
              "00000110" when entry_set,
              "00001100" when set_display,
              "00000001" when clr_display,
              "10000000" when set_addr,
              "01001011" when char_K,
              "01100001" when char_a1,
              "01110000" when char_p,
              "01100001" when char_a2,
              "01110011" when char_s,
              "01101001" when char_i,
              "01110100" when char_t,
              "01100001" when char_a3,
              "01110011" when char_s1,
              "00100000" when char_space,
              "0011"&pul_kap when char_pul_kap,
              "0011"&sat_kap when char_sat_kap,
              "00100000" when char_space1,
              "00100000" when char_space30,
              "01001010" when char_J,
              "01110101" when char_u,
              "01101101" when char_m,
              "01101100" when char_l,
              "01100001" when char_a,
              "01101000" when char_h,
              "00100000" when char_space5,
              "0011"&puluhan when char_puluhan,
              "0011"&satuan when char_satuan,
              --"0011"&kap when char_kap,
              "00000000" when others;

```

4.1.2.3 Program Buzzer

Program Buzzer akan aktif jika jumlah pembatas ruangan dan jumlah pengunjung dalam ruangan telah sama.

```
if (pul_kap = "0010" and sat_kap = "0000" and puluhan = "0010" and
satuan <= "0000") then
    bunyi <= '1';
else
    bunyi <= '0';
end if;
```

4.2 Uji Coba Sistem

4.2.1 Percobaan 1 – Pembacaan Sensor DI-MLDR

Pada percobaan sensor ini, parameter yang diuji adalah ketepatan sensor dalam mendeteksi objek yang melintas. Percobaan ini dilakukan sebanyak lima belas kali percobaan. Berikut adalah hasil dari percobaan 1:

Table 4.1 Simulasi Objek Masuk Ruangan

Objek ke	Tampilan Jumlah Orang
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	10
11	11
12	12
13	13
14	14
15	15

Table 4.2 Simulasi Objek Keluar Ruangan

Objek ke	Tampilan Jumlah Orang
1	15
2	14
3	13
4	12
5	11
6	10
7	9
8	8
9	7
10	6
11	5
12	4
13	3
14	2
15	1

Table 4.3 konversi 8 bit counter ke dalam bilangan

Counter	Bit	Counter	Bit	Counter	Bit	Counter	Bit	Counter	Bit
1	01010001	21	10100101	41	11111001	61	01001101	81	10100001
2	10100010	22	11110110	42	01001010	62	10011110	82	11110010
3	11110011	23	01000111	43	10011011	63	11101111	83	01000011
4	01000100	24	10011000	44	11101100	64	01000000	84	10010100
5	10010101	25	11101001	45	00111101	65	10010001	85	11100101
6	11100110	26	00111010	46	10001110	66	11100010	86	00110110
7	00110111	27	10001011	47	11011111	67	00110011	87	10000111
8	10001000	28	11011100	48	00110000	68	10000100	88	11011000
9	11011001	29	00101101	49	10000001	69	11010101	89	00101001
10	00101010	30	01111110	50	11010010	70	00100110	90	01111010
11	01111011	31	11001111	51	00100011	71	01110111	91	11001011
12	11001100	32	00100000	52	01110100	72	11001000	92	00011100
13	00011101	33	01110001	53	11000101	73	00011001	93	01101101
14	01101110	34	11000010	54	00010110	74	01101010	94	10111110
15	10111111	35	00010011	55	01100111	75	10111011	95	00001111
16	00010000	36	01100100	56	10111000	76	00001100	96	01100000
17	01100001	37	10110101	57	00001001	77	01011101	97	10110001
18	10110010	38	00000110	58	01011010	78	10101110	98	00000010
19	00000011	39	01010111	59	10101011	79	11111111	99	01010011
20	01010100	40	10101000	60	11111100	80	01010000		

4.2.2 Percobaan 2 – Pembacaan Buzzer

Pada percobaan ini, pengujian terhadap buzzer sebagai peringatan bahwa ruangan telah penuh. Parameter yang diuji adalah ketepatan bunyi ketika jumlah pengunjung ruangan sama dengan dan melebihi batas kapasitas ruangan. Percobaan ini dilakukan selama sepuluh kali percobaan. Berikut adalah hasil dari percobaan 3:

Pada percobaan ini dilakukan pada saat counter penjumlahan sama dengan kapasitas maka buzzer akan berbunyi. Akan tetapi belum bisa dilakukan karena pendeklarasian pada output dari pin belum berhasil.

4.2.3 Percobaan 3 – Pembacaan Hasil pada LCD

Pada percobaan ini, parameter yang diuji adalah ketepatan LCD dalam menampilkan data hasil pembacaan sensor dengan beberapa keadaan yang dilakukan pada percobaan 1 sampai dengan 3 yaitu:

Ketika sensor DI-MLDR memberikan signal counter penambahan dan pengurangan bisa ditampilkan dalam bentuk puluhan dan satuan begitu juga pada kapasitas bisa ditampilkan akan tetapi kapasitas dilakukan langsung perubahannya dalam program.

4.3 Analisis Hasil Percobaan

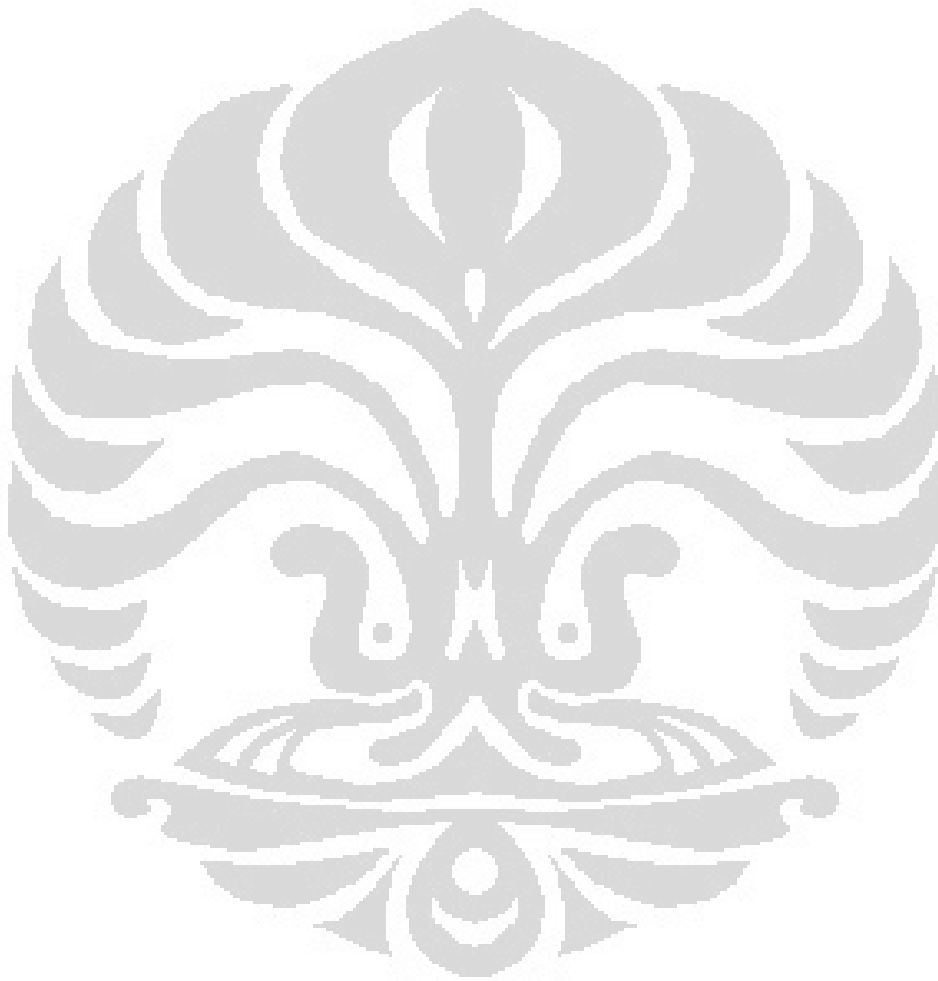
Analisa dilakukan dengan cara menjalankan semua sistem keseluruhan secara bersamaan. Dengan memprogram FPGA menggunakan ISE 13.2 untuk mendownload program kedalam FPGA.

Sistem mulai berjalan dengan memasukan jumlah pembatas kapasitas dari suatu ruangan dengan dibatasi langsung pada program karena belum bisa dibuat menggunakan input dari keypad.

Semua sensor DI-MLDR dan laser pointer disimulasikan menggunakan swith pada FPGA. Sensor pada pintu masuk akan melakukan counter penambahan dapat terlihat pada layar LCD. Begitu pula pada sensor pintu keluar akan melakukan counter pengurangan.

Counter mendapatkan perhatian khusus karena jika menggunakan 8 bit tanpa dilakukan pengambilan data dan konversi kebilangan maka akan

menghasilkan tampilan yang tidak sesuai. Pada 4 bit pertama tidak ada masalah namun pada 4 bit setelahnya data masuk tidak berurutan dan acak. Hal ini yang memakan lama dalam proses pemrograman. Dari hal tersebut maka dilakukan pengambilan data acak yang kemudian di konversi ke dalam bilangan angka.



BAB 5

KESIMPULAN

Beberapa hal yang dapat disimpulkan dari perancangan dan implementasi sistem pada skripsi, setelah melalui berbagai rangkaian uji coba ini adalah sebagai berikut:

1. Pada percobaan pertama untuk perhitungan counter penambah dan pengurangan berhasil dilakukan.
2. Pada percobaan kedua untuk membunyikan buzzer ketika jumlah objek dan kapasitas pada ruangan sama blm berhasil dilakukan dan perlu dilakukan percobaan pemograman kembali.
3. Pada percobaan ketiga untuk menampilkan hasil perhitungan counter penambahan dan pengurangan serta kapasitas ruangan ke LCD screen berhasil dilakukan.
4. Pada proses penampilan hasil perhitungan counter diperlukan pengambilan data 99 kali untuk mengkonversi 8 bit ke dalam bilangan angka pada karakter yang tersedia di VHDL.
5. Proses input kapasitas menggunakan keypad belum dapat dilakukan.

Dari hasil yang didapat ini, akan dijadikan bahan studi lebih lanjut untuk pengembangan.

DAFTAR ACUAN

- [1] “Implementasi Model Pembangkit Runtun PN dan Sandi Walsh berbasis FPGA”, Jakarta.
- [2] “Modul Praktikum Pengenalan FPGA Xilinx Spartan 3E”. Diakses tanggal 16 Januari 2012. <http://dc340.4shared.com/doc/CTlhA1Ou/preview.html>.
- [3] “VHDL Struktur Dasar 1”. Diakses tanggal 18 Januari 2012. <http://rizkia.wordpress.com/2011/01/11/vhdl-struktur-dasar-1>.
- [4] “Teori Dasar Rangkaian LDR”. Diakses tanggal 18 Januari 2012. <http://depokinstruments.com/2011/07/29/teori-dasar-ldr-dan-rangkaian-ldr-dalam-pengembangan>.
- [5] “Macam-macam Jenis LED”. Diakses tanggal 19 Januari 2012. <http://rasapas.wordpress.com/2011/03/04/8/>
- [6] “Modul 1 Pengenalan FPGA”. Diakses tanggal 19 Januari 2012. <http://lecturer.eepis-its.edu/~taufiq/Praktikum%20VLSI%20D4LJ/Modul%20Praktikum%20FPGA/MODUL%20I%20rev1-2.pdf>
- [7] Agustina A. R, “Penghitung Jumlah Pengunjung Ruangan Pertunjukan Berbasis Mikrokontroler AT89C51”, Semarang. 2005.
- [8] Teknik pengendalian lcd karakter. Diakses tanggal 03 Januari 2012. <http://depokinstruments.com/2010/02/08/teknik-pengendalian-lcd-karakter-i/>
- [9] DI-MLDR. Diakses tanggal 10 Mei 2012. <http://depokinstruments.com/2010/02/08/24/produk-baru-di-mldr-di-multifunction-ldr/>

DAFTAR PUSTAKA

- [1] Tulisan tentang FPGA <http://tenikcal.blogspot.com/2010/10/tulisan-tentang-fpga.html>
- [2] “Xilinx Spartan 3e – Documenttions”. Diakses tanggal 01 Januari 2012.
<http://www.xilinx.com/support/index.htm>
- [3] Teknik pengendalian lcd karakter. Diakses tanggal 03 Januari 2012.
<http://depokinstruments.com/2010/02/08/teknik-pengendalian-lcd-karakter-i/>
- [4] DI-MLDR. Diakses tanggal 10 Mei 2012.
<http://depokinstruments.com/2010/02/08/24/produk-baru-di-mldr-di-multifunction-ldr/>
- [5] Library VHDL. Diakses tanggal 12 Juni 2012. <http://viana-chubby.blogspot.com/2010/10/library-vhdl.html>
- [6] Tutorial VHDL. Diakses tanggal 12 April 2012.
<http://somejunks.wordpress.com/2011/11/01/tutorial-mari-belajar-vhdl-secara-singkat-1/>

LAMPIRAN

```
-----  
-----  
-- Company: Fakultas Teknik Universitas Indonesia - Departemen  
Teknik Elektro  
-- Engineer: Dani Tri Sutrisno K  
--  
-- Create Date: 21:35:18 06/11/2012  
-- Design Name:  
-- Module Name: main - Behavioral  
-- Project Name:  
-- Target Devices:  
-- Tool versions:  
-- Description:  
--  
-- Dependencies:  
--  
-- Revision:  
-- Revision 0.01 - File Created  
-- Additional Comments:  
--  
-----  
-----  
library IEEE;  
use IEEE.STD_LOGIC_1164.ALL;  
use IEEE.NUMERIC_STD.ALL;  
use IEEE.STD_LOGIC_ARITH.ALL;  
use IEEE.STD_LOGIC_UNSIGNED.ALL;  
  
-- Uncomment the following library declaration if using  
-- arithmetic functions with Signed or Unsigned values  
--use IEEE.NUMERIC_STD.ALL;  
  
-- Uncomment the following library declaration if instantiating  
-- any Xilinx primitives in this code.  
--library UNISIM;  
--use UNISIM.VComponents.all;  
  
entity main is
```

```

    port(
        sig1, sig2, clk, clr: in STD_LOGIC;
        bunyi, LCD_E, LCD_RS, LCD_RW, SF_CE0: out STD_LOGIC;
        SF_D : out STD_LOGIC_VECTOR(3 downto 0)
    );
end main;

```

architecture Behavioral of main is

component lcd

```

    port(
        clk, reset : in STD_LOGIC;
        LCD_E, LCD_RS, LCD_RW, SF_CE0: out STD_LOGIC;
        SF_D : out STD_LOGIC_VECTOR(3 downto 0);
        satuan: in STD_LOGIC_VECTOR(3 downto 0);
        puluhan: in STD_LOGIC_VECTOR(3 downto 0);
        sat_kap: in STD_LOGIC_VECTOR(11 downto 8);
        pul_kap: in STD_LOGIC_VECTOR(15 downto 12)
    );

```

end component;

```

signal sat_counter : STD_LOGIC_VECTOR(7 downto 0) := "00000000";
signal pul_counter : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal kap : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal sat_kap : STD_LOGIC_VECTOR(11 downto 8) := "0000";
signal pul_kap : STD_LOGIC_VECTOR(15 downto 12) := "0000";
signal satuan : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal puluhan : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal Q1, Q2, Q3, Q4, Q5, Q6, QR : std_logic := '0';
signal S1, S2, S3, S4, S5, S6, SR : std_logic := '0';
signal R0, R1, R2, R3, R4, R5, R6, RR : std_logic := '0';
signal counter1, counter2, kap1, jmlh : integer := 0;
begin

```

MODULE1: lcd

```

    port map(
        clk => clk,
        reset => RR,

```



```

        SF_D => SF_D,
        LCD_E => LCD_E,
        LCD_RS => LCD_RS,
        LCD_RW => LCD_RW,
        SF_CE0 => SF_CE0,
        puluhan => puluhan,
        satuan => satuan,
        sat_kap => sat_kap,
        pul_kap => pul_kap
    );

detect: process(clk, sig1, sig2, clr)
begin
    if(clk'event and clk = '1') then
        if(counter1 = 50000) then
            Q1 <= sig1;
            Q2 <= Q1;
            Q3 <= Q2;
            Q4 <= Q3;
            Q5 <= Q4;
            Q6 <= Q5;
            counter1 <= 0;
        else
            counter1 <= counter1 + 1;
        end if;

        if(counter2 = 50000) then
            S1 <= sig2;
            S2 <= S1;
            S3 <= S2;
            S4 <= S3;
            S5 <= S4;
            S6 <= S5;
            counter2 <= 0;
        else
            counter2 <= counter2 + 1;
        end if;
    end if;
end process;

```

```
if(QR = '1')then

    sat_counter <= sat_counter + 1;

end if;

if(SR = '1')then

    sat_counter <= sat_counter - 1;

end if;

if(QR = '1' or SR = '1') then

    pul_kap <= "0010";
    sat_kap <= "0000";
    kap1 <= 20;

    RR <= '1';
else
    RR <= '0';
end if;

if(clr = '1') then

    sat_counter <= "00000000";
    pul_counter <= "0000";
    puluhan <= "0000";
    satuan <= "0000";

end if;

if sat_counter = "01010001" then
    puluhan<="0000";
    satuan<="0001";

elsif sat_counter = "10100010" then
```

```
puluhan<="0000";
satuan<="0010";

elsif sat_counter = "11110011" then
    puluhan<="0000";
    satuan<="0011";

elsif sat_counter = "01000100" then
    puluhan<="0000";
    satuan<="0100";

elsif sat_counter = "10010101" then
    puluhan<="0000";
    satuan<="0101";

elsif sat_counter = "11100110" then
    puluhan<="0000";
    satuan<="0110";

elsif sat_counter = "00110111" then
    puluhan<="0000";
    satuan<="0111";

elsif sat_counter = "10001000" then
    puluhan<="0000";
    satuan<="1000";

elsif sat_counter = "11011001" then
    puluhan<="0000";
    satuan<="1001";

elsif sat_counter = "00101010" then
    puluhan<="0001";
    satuan<="0000";

--10
elsif sat_counter = "01111011" then
    puluhan<="0001";
    satuan<="0001";

elsif sat_counter = "11001100" then
    puluhan<="0001";
    satuan<="0010";

elsif sat_counter = "00011101" then
```

```
        puluhan<="0001";
        satuan<="0011";
    elsif sat_counter = "01101110" then
        puluhan<="0001";
        satuan<="0100";
    elsif sat_counter = "10111111" then
        puluhan<="0001";
        satuan<="0101";
        jmlh <= 15;
    elsif sat_counter = "00010000" then
        puluhan<="0001";
        satuan<="0110";
    elsif sat_counter = "01100001" then
        puluhan<="0001";
        satuan<="0111";
    elsif sat_counter = "10110010" then
        puluhan<="0001";
        satuan<="1000";
    elsif sat_counter = "00000011" then
        puluhan<="0001";
        satuan<="1001";
    elsif sat_counter = "01010100" then
        puluhan<="0010";
        satuan<="0000";
        jmlh <= 20;
--20
    elsif sat_counter = "10100101" then
        puluhan<="0010";
        satuan<="0001";
    elsif sat_counter = "11110110" then
        puluhan<="0010";
        satuan<="0010";
    elsif sat_counter = "01000111" then
        puluhan<="0010";
        satuan<="0011";
```

```
elseif sat_counter = "10011000" then
    puluhan<="0010";
    satuan<="0100";
elseif sat_counter = "11101001" then
    puluhan<="0010";
    satuan<="0101";
elseif sat_counter = "00111010" then
    puluhan<="0010";
    satuan<="0110";
elseif sat_counter = "10001011" then
    puluhan<="0010";
    satuan<="0111";
elseif sat_counter = "11011100" then
    puluhan<="0010";
    satuan<="1000";
elseif sat_counter = "00101101" then
    puluhan<="0010";
    satuan<="1001";
elseif sat_counter = "01111110" then
    puluhan<="0011";
    satuan<="0000";
--30
elseif sat_counter = "11001111" then
    puluhan<="0011";
    satuan<="0001";
elseif sat_counter = "00100000" then
    puluhan<="0011";
    satuan<="0010";
elseif sat_counter = "01110001" then
    puluhan<="0011";
    satuan<="0011";
elseif sat_counter = "11000010" then
    puluhan<="0011";
    satuan<="0100";
elseif sat_counter = "00010011" then
    puluhan<="0011";
    satuan<="0101";
elseif sat_counter = "01100100" then
```

```
        puluhan<="0011";
        satuan<="0110";
    elsif sat_counter = "10110101" then
        puluhan<="0011";
        satuan<="0111";
    elsif sat_counter = "00000110" then
        puluhan<="0011";
        satuan<="1000";
    elsif sat_counter = "01010111" then
        puluhan<="0011";
        satuan<="1001";
    elsif sat_counter = "10101000" then
        puluhan<="0100";
        satuan<="0000";
--40
    elsif sat_counter = "11111001" then
        puluhan<="0100";
        satuan<="0001";
    elsif sat_counter = "01001010" then
        puluhan<="0100";
        satuan<="0010";
    elsif sat_counter = "10011011" then
        puluhan<="0100";
        satuan<="0011";

    elsif sat_counter = "11101100" then
        puluhan<="0100";
        satuan<="0100";
    elsif sat_counter = "00111101" then
        puluhan<="0100";
        satuan<="0101";

    elsif sat_counter = "10001110" then
        puluhan<="0100";
        satuan<="0110";

    elsif sat_counter = "11011111" then
        puluhan<="0100";
        satuan<="0111";

    elsif sat_counter = "00110000" then
        puluhan<="0100";
```

```

        satuan<="1000";
    elsif sat_counter = "10000001" then
        puluhan<="0100";
        satuan<="1001";
    elsif sat_counter = "11010010" then
        puluhan<="0101";
        satuan<="0000";

--50
    elsif sat_counter = "00100011" then
        puluhan<="0101";
        satuan<="0001";
    elsif sat_counter = "01110100" then
        puluhan<="0101";
        satuan<="0010";
    elsif sat_counter = "11000101" then
        puluhan<="0101";
        satuan<="0011";
    elsif sat_counter = "00010110" then
        puluhan<="0101";
        satuan<="0100";
    elsif sat_counter = "01100111" then
        puluhan<="0101";
        satuan<="0101";
    elsif sat_counter = "10111000" then
        puluhan<="0101";
        satuan<="0110";
    elsif sat_counter = "00001001" then
        puluhan<="0101";
        satuan<="0111";
    elsif sat_counter = "01011010" then
        puluhan<="0101";
        satuan<="1000";
    elsif sat_counter = "10101011" then
        puluhan<="0101";
        satuan<="1001";

    elsif sat_counter = "11111100" then
        puluhan<="0110";
        satuan<="0000";

```

```
--60
elsif sat_counter = "01001101" then
    puluhan<="0110";
    satuan<="0001";
elsif sat_counter = "10011110" then
    puluhan<="0110";
    satuan<="0010";
elsif sat_counter = "11101111" then
    puluhan<="0110";
    satuan<="0011";
elsif sat_counter = "01000000" then
    puluhan<="0110";
    satuan<="0100";
elsif sat_counter = "10010001" then
    puluhan<="0110";
    satuan<="0101";
elsif sat_counter = "11100010" then
    puluhan<="0110";
    satuan<="0110";
elsif sat_counter = "00110011" then
    puluhan<="0110";
    satuan<="0111";
elsif sat_counter = "10000100" then
    puluhan<="0110";
    satuan<="1000";
elsif sat_counter = "11010101" then
    puluhan<="0110";
    satuan<="1001";
elsif sat_counter = "00100110" then
    puluhan<="0111";
    satuan<="0000";

--70
elsif sat_counter = "01110111" then
    puluhan<="0111";
    satuan<="0001";
elsif sat_counter = "11001000" then
    puluhan<="0111";
    satuan<="0010";
elsif sat_counter = "00011001" then
```



```
        puluhan<="0111";
        satuan<="0011";
    elsif sat_counter = "01101010" then
        puluhan<="0111";
        satuan<="0100";
    elsif sat_counter = "10111011" then
        puluhan<="0111";
        satuan<="0101";

    elsif sat_counter = "00001100" then
        puluhan<="0111";
        satuan<="0110";
    elsif sat_counter = "01011101" then
        puluhan<="0111";
        satuan<="0111";
    elsif sat_counter = "10101110" then
        puluhan<="0111";
        satuan<="1000";
    elsif sat_counter = "11111111" then
        puluhan<="0111";
        satuan<="1001";
    elsif sat_counter = "01010000" then
        puluhan<="1000";
        satuan<="0000";
--80
    elsif sat_counter = "10100001" then
        puluhan<="1000";
        satuan<="0001";
    elsif sat_counter = "11110010" then
        puluhan<="1000";
        satuan<="0010";

    elsif sat_counter = "01000011" then
        puluhan<="1000";
        satuan<="0011";

    elsif sat_counter = "10010100" then
        puluhan<="1000";
        satuan<="0100";

    elsif sat_counter = "11100101" then
        puluhan<="1000";
```

```

        satuan<="0101";
    elsif sat_counter = "00110110" then
        puluhan<="1000";
        satuan<="0110";
    elsif sat_counter = "10000111" then
        puluhan<="1000";
        satuan<="0111";
    elsif sat_counter = "11011000" then
        puluhan<="1000";
        satuan<="1000";
    elsif sat_counter = "00101001" then
        puluhan<="1000";
        satuan<="1001";
    elsif sat_counter = "01111010" then
        puluhan<="1001";
        satuan<="0000";
--90
    elsif sat_counter = "11001011" then
        puluhan<="1001";
        satuan<="0001";

    elsif sat_counter = "00011100" then
        puluhan<="1001";
        satuan<="0010";
    elsif sat_counter = "01101101" then
        puluhan<="1001";
        satuan<="0011";
    elsif sat_counter = "10111110" then
        puluhan<="1001";
        satuan<="0100";
    elsif sat_counter = "00001111" then
        puluhan<="1001";
        satuan<="0101";
    elsif sat_counter = "01100000" then
        puluhan<="1001";
        satuan<="0110";

    elsif sat_counter = "10110001" then
        puluhan<="1001";

```

```

        satuan<="0111";
    elsif sat_counter = "00000010" then
        puluhan<="1001";
        satuan<="1000";
    elsif sat_counter = "01010011" then
        puluhan<="1001";
        satuan<="1001";

        elsif sat_counter = "00000000" then
            puluhan<="0000";
            satuan<="0000";

        end if;

        if (jmlh = kap) then
            bunyi <= '1';
        else
            bunyi <= '0';
        end if;

    end if;
end process;

sat_LCD <= pul_counter;

QR <= Q1 and Q2 and Q3 and Q4 and Q5 and(not Q6);

SR <= S1 and S2 and S3 and S4 and S5 and(not S6);

end Behavioral;

```