



UNIVERSITAS INDONESIA

**PENGEMBANGAN DAN EVALUASI SMART STICK
SEBAGAI ALAT BANTU UNTUK ORANG DENGAN
KECACATAN VISUAL**

SKRIPSI

**MARTIN CHORAZON ALFARENO
0906602793**

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



UNIVERSITAS INDONESIA

**PENGEMBANGAN DAN EVALUASI SMART STICK
SEBAGAI ALAT BANTU UNTUK ORANG DENGAN
KECACATAN VISUAL**

SKRIPSI

Skripsi ini diajukan untuk melengkapi sebagian persyaratan menjadi Sarjana
Teknik

MARTIN CHORAZON ALFARENO

0906602793

**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 : Martin Chorazon Alfareno

NPM : 0906602793

Tanda Tangan :




Tanggal : 03 Juli 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Martin Chorazon Alfareno
NPM : 0906602793
Program studi : Teknik Elektro
Judul Skripsi : PENGEMBANGAN DAN EVALUASI
SMART STICK SEBAGAI ALAT BANTU
UNTUK ORANG DENGAN
KECACATAN VISUAL

Telah dipresentasikan dan diterima sebagai persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia.

Dosen 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

Segala puji dan syukur penulis panjatkan kepada Allah SWT karena atas segala limpahan karunia dan rahmat-Nya penulis dapat menyelesaikan skripsi ini dalam rangka memenuhi salah satu syarat untuk mencapai gelar Sarjana Teknik Elektro pada Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia.

Tanpa bantuan dan dukungan dari berbagai pihak sejak masa perkuliahan dan masa penyusunan skripsi, sulit bagi penulis untuk menyelesaikan skripsi ini. Oleh karena itu pada kesempatan ini penulis ingin mengucapkan terimakasih dan rasa hormat dengan segala kerendahan dan ketulusan hati kepada :

1. Prof. Dr.-Ing. Ir. Kalamullah Ramli, M.Eng. sebagai dosen pembimbing yang telah menyediakan waktu dan pikiran serta memberikan ilmu-ilmu bermanfaat untuk membantu dan mengarahkan penulis dalam penyusunan skripsi ini;
2. Bapak, Emak, Koko Joni, dan Adik Wenny tercinta yang telah memberi dukungan, kasih sayang, semangat dan doa yang menyertai penulis;
3. Sahabat-sahabat EXEL'09 dan teman-teman Laboratorium Digital terutama Dani dan Hanna sebagai rekan kerja penulis;
4. Saudara-saudaraku di Pondok Abadi, Andri, Om Dhanu, Olala, Kak Zaki, Bang Dhatu, Ipan, Gobit, Eros, Jabrul, Mamad, Bintang Ardi dan para Alumni Pondok Abadi;

Akhir kata, penulis menyadari bahwa penelitian dan penyusunan skripsi ini masih jauh dari kesempurnaan. Oleh karena itu, penulis dengan senang hati menerima segala kritik dan saran demi perbaikan di masa yang akan datang. Semoga skripsi ini dapat memberikan manfaat bagi pengembangan ilmu pengetahuan pada umumnya dan ilmu Teknik Elektro pada khususnya.

Depok, 03 Juli 2012



Martin Chorazon Alfareno

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

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

Nama : Martin Chorazon Alfareno

NPM : 0906602793

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 Nonoksklusif (*Non-exclusive Royalty Free Right*)** atas karya ilmiah saya yang berjudul:

**PENGEMBANGAN DAN EVALUASI *SMART STICK* SEBAGAI ALAT
BANTU UNTUK ORANG DENGAN KECACATAN VISUAL**

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,



(Martin Chorazon Alfareno)

ABSTRAK

Nama : Martin Chorazon Alfareno
Program Studi : Teknik Elektro
Judul : Pengembangan dan Evaluasi *Smart Stick* Sebagai Alat Bantu Untuk Orang Dengan Kecacatan Visual.

Pada tugas akhir ini telah dilakukan perancangan dan evaluasi. Keseluruhan sistem dan pengembangan menggunakan FPGA Xilinx Spartan-3E. *Smart Stick* berfungsi sebagai alat bantu berjalan bagi orang buta dalam menentukan arah. Sistem *Smart Stick* menggunakan 3 buah sensor ultrasonik dengan posisi pemasangan pada bagian depan, kanan, dan kiri. Sensor ultrasonik sebagai modul *input* akan mendeteksi keberadaan objek disekitar *Smart Stick*. Pembatasan jarak baca sensor sejauh 100 cm yang berfungsi sebagai jarak aman bagi pengguna *smart stick*. FPGA yang digunakan adalah FPGA Xilinx Spartan-3E sebagai *controller*, *buzzer* sebagai modul *output* akan mengeluarkan bunyi peringatan sebagai tanda bahwa sensor mendeteksi adanya penghalang. Percobaan ini menunjukkan semua kinerja subsistem yang memuaskan.

Kata kunci : sensor ultrasonik, FPGA Xilinx Spartan-3E, *buzzer*

ABSTRACT

Name : Martin Chorazon Alfareno
Study Program : Electrical Engineering
Title : Development and Evaluate of Smart Stick as Assistive Device For People With Visual Disability .

In this final project Smart Stick has been designed and evaluated. The system and development use FPGA Xilinx Spartan-3E. Smart Stick serves as a tool for blind people to determine the direction for walking. Smart Stick system uses 3 pieces of ultrasonic sensors with mounting position on the front, right, and left. Ultrasonic sensors as the input module detect the presence of objects around the Smart Stick. The sensors read having distance limitation as far as 100 cm, this is a safe distance for the Smart Stick user. FPGA used is FPGA Xilinx Spartan-3E as a controller, buzzer as output modules will issue a warning sound as a sign that the sensor detects the presence of a barrier. The experiment show me all the subsystems performed satisfactorily.

Keywords : *ultrasonic sensor*, FPGA Xilinx Spartan-3E, *buzzer*

DAFTAR ISI

HALAMAN JUDUL.....	ii
HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	vi
ABSTRAK	vii
ABSTRACT.....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL.....	xii
DAFTAR LAMPIRAN.....	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian.....	1
1.3 Pembatasan Masalah.....	2
1.4 Metodologi Penulisan.....	2
1.5 Sistematika Penulisan.....	2
BAB 2 LANDASAN TEORI.....	4
2.1 Sejarah dan Deskripsi FPGA.....	4
2.1.1 Sejarah FPGA.....	4
2.2 Pengenalan FPGA.....	4
2.2.1 Pengertian FPGA.....	4
2.2.2 Cara Kerja FPGA.....	7
2.2.3 Konfigurasi FPGA.....	9
2.2.4 FPGA Design Software.....	11
2.3 Verilog.....	12
2.4 LCD 16x2	12
2.5 Sensor Ultrasonik PING.....	13
2.6 <i>Buzzer</i>	16
BAB 3 PERANCANGAN SMART STICK UNTUK ORANG BUTA BERBASIS FPGA XILINX SPARTAN-3E.....	17
3.1 Perancangan Alat.....	17
3.2 ISE Webpack 14.1	18
3.2.1 Pcerancangan Software.....	21
3.3 Cara Kerja Sistem.....	22
3.4 Desain Smart Stick	23
BAB 4 PENGUJIAN DAN ANALISA DATA	25
4.1 Pengujian dan Analisa Pada Modul Input Sensor Ultrasonik.....	25
4.1.1 Deskripsi Pengujian.....	25
4.1.2 Prosedur Pengukuran dan Pengujian	25
4.1.2.1 Daftar Peralatan Pengujian	25

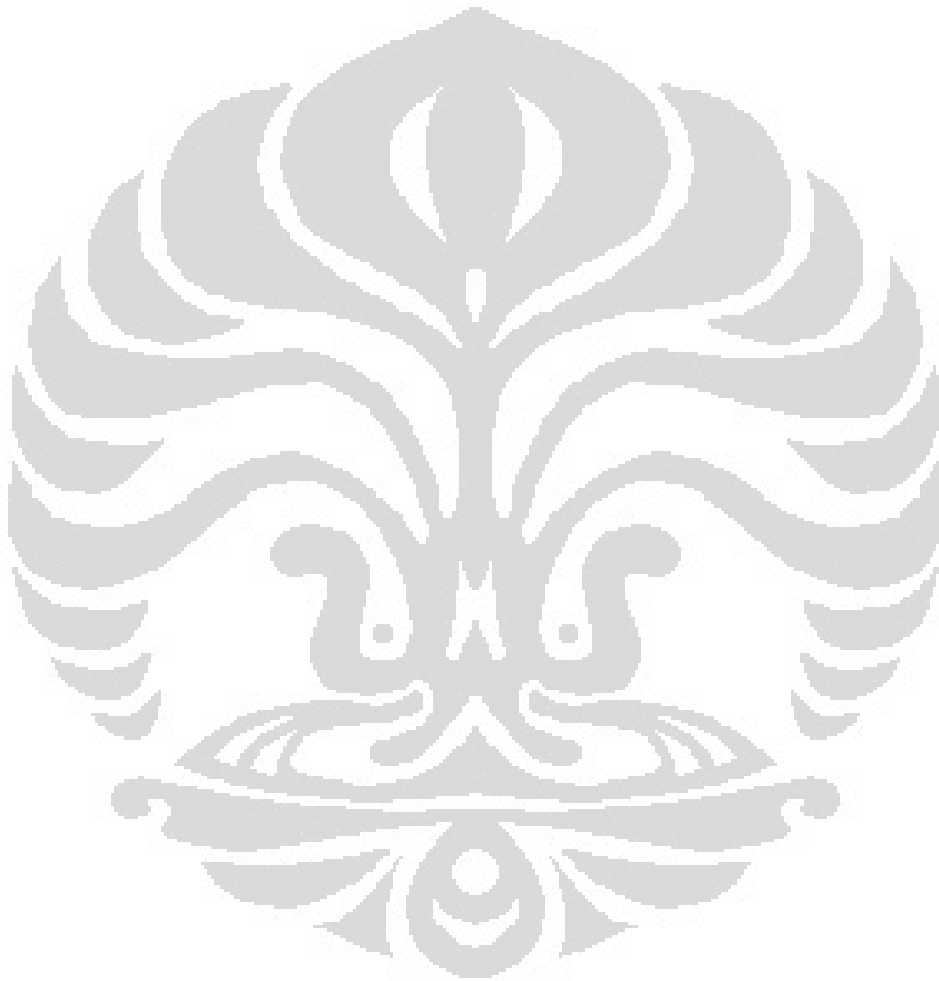
4.1.2.2	Prosedur Pengujian Modul Input Sensor Ultrasonik	26
4.1.3	Hasil Pengujian dan Pengukuran	27
4.1.4	Prosedur Pengujian Modul <i>Input</i> Sensor Ultrasonik Pada FPGA	27
4.1.5	Pengujian Sistem Sensor Ultrasonik.....	28
4.1.6	Analisa	29
4.1.7	Wiring Modul Input Sensor Ultrasonik.....	29
4.2	Pengujian dan Analisa Pada Modul <i>Output Buzzer</i>	30
4.2.1	Deskripsi Pengujian	30
4.2.2	Prosedur Pengujian dan Pengukuran	31
4.2.2.1	Daftar Peralatan Pengujian	31
4.2.2.2	Prosedur Pengujian Modul <i>Output Buzzer</i>	31
4.2.3	Hasil Pengujian dan Pengukuran	32
4.2.4	Pengujian Sistem <i>Buzzer</i>	32
4.2.5	Analisa	32
4.3	Program	33
4.3.1	Progeam LCD	33
4.3.2	Program <i>Buzzer</i>	33
4.3.3	Program LED.....	33
4.4	Analisa Keseluruhan Sistem	33
BAB 5 KESIMPULAN.....		35
DAFTAR ACUAN		36
DAFTAR PUSTAKA		38
LAMPIRAN.....		39

DAFTAR GAMBAR

Gambar 2.1 FPGA Xilinx Spartan [9].....	5
Gambar 2.2 Arsitektur FPGA [9].....	6
Gambar 2.3 Modul Xilinx Spartan-3E [2]	7
Gambar 2.4 Bagian Dalam FPGA [9].....	8
Gambar 2.5 Isi Logic Cell [9].....	8
Gambar 2.6 Logic Cell yang Terhubung satu sama lain [9]	9
Gambar 2.7 Proses Download dari Komputer ke FPGA [9].....	11
Gambar 2.8 LCD 16x2 [8]	12
Gambar 2.9 Interface LCD 16x2 [8].....	13
Gambar 2.10 Sensor Ultrasonik PING [7]	14
Gambar 2.11 Prinsip Kerja Sensor Ultrasonik [7]	14
Gambar 2.12 Pulsa Ping))) Parallax Ultrasonic Range Finder [10].....	15
Gambar 2.13 Buzzer [11].....	16
Gambar 3.1 Blok Diagram Smart Stick	17
Gambar 3.2 ISE Project Navigator GUI: Explorer, Proses, Entry dan Log [4] ...	19
Gambar 3.3 ISE IMPACT: Konfigurasi Boundary Scan (JTAG) [4].....	19
Gambar 3.4 Metodologi desain sistem digital menggunakan ISE: entri.....	21
desain, sintesis, implementasi dan pemrograman [4].....	21
Gambar 3.5 Diagram Alir Cara Kerja Sistem	22
Gambar 3.6 Diagram Alir Smart Stick Untuk Orang Buta Berbasis FPGA Xilinx Spartan-3E.....	23
Gambar 3.7 Desain Smart Stick	24
Gambar 4.1 Tampilan Data Pada LCD Ketika Sensor Mendeteksi Benda.....	29
Gambar 4.1 Wiring Modul Input Sensor Ultrasonik.....	30
Gambar 4.2 LED Sebagai Indikator Proses Kerja Sistem.....	33
Gambar 4.3 Prototype Smart Stick	34

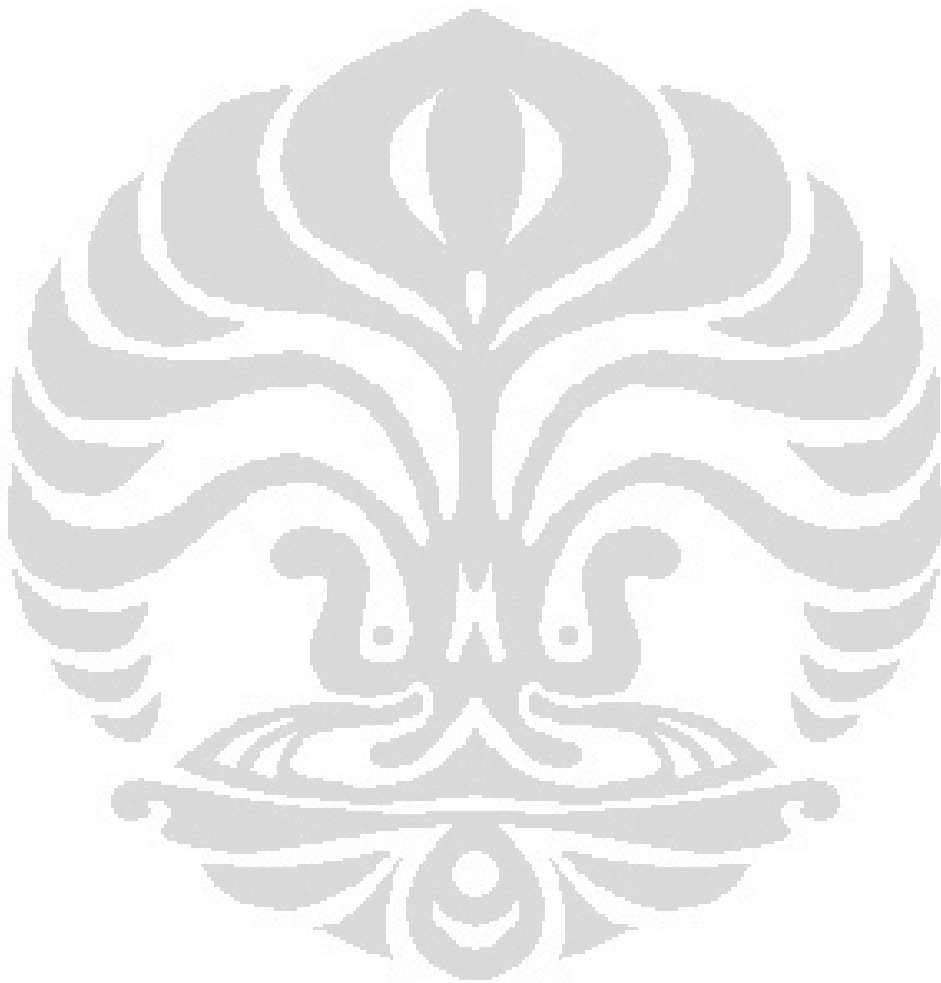
DAFTAR TABEL

Tabel 4.1 Daftar Alat Pengujian Modul Input Sensor Ultrasonik.....	26
Tabel 4.2 Data Hasil Pengukuran Modul Input Sensor Ultrasonik	27
Tabel 4.3 Simulasi Sistem Sensor Ultrasonik.....	28
Tabel 4.4 Daftar Alat Pengujian Modul Output Buzzer	31
Tabel 4.5 Data Hasil Pengukuran Modul Output <i>Buzzer</i>	32
Tabel 4.6 Simulasi Sistem <i>Buzzer</i>	32



DAFTAR LAMPIRAN

Lampiran 1. Keseluruhan Program	38
Lampiran 2. Program LCD	42



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi dewasa ini sungguh sangat pesat, sehingga mendorong sebagian individu ataupun kelompok mau tidak mau harus menerapkannya dalam segala aktivitas di dalam kehidupannya. Dengan kemajuan teknologi ini, setiap individu maupun kelompok dapat dengan mudah melakukan setiap aktivitas yang tidak dapat dilakukan dengan baik oleh manusia. Perkembangan tersebut didukung teknologi tinggi seperti tersedianya perangkat keras (hardware) dan perangkat lunak (software) yang semakin hari semakin meningkat kemampuannya.

Dengan memanfaatkan teknologi yang canggih saat ini, dapat kita manfaatkan untuk membantu sesama yang mengalami kebutaan agar dapat berjalan dengan baik dengan menciptakan sebuah alat bantu berjalan. Alat ini akan dikendalikan sepenuhnya oleh sebuah *Field Programmable Gate Array* (FPGA) dan alat bantu berjalan ini akan dilengkapi oleh 3 buah sensor ultrasonik yang akan berfungsi untuk membaca ada tidaknya benda yang menghalangi sehingga alat ini akan memberikan jarak aman terhadap orang buta yang menggunakan alat ini dan memudahkan dalam menentukan arah berjalan. Selain itu juga alat bantu ini akan dilengkapi dengan *output* berupa *buzzer* yang akan mengeluarkan bunyi peringatan apabila terdapat suatu benda yang menghalangi sensor ultrasonik.

1.2 Tujuan Penelitian

Tujuan Pengembangan dan Evaluasi *Smart Stick* Sebagai Alat Bantu Untuk Orang Dengan Kecacatan Visual agar dapat memberikan kemudahan kepada orang buta dalam menentukan arah berjalan. Serta memberikan penjelasan penggunaan FPGA serta membuat suatu program yang menggunakan bahasa pemrograman Verilog.

Penulisan Tugas Akhir ini merupakan salah satu syarat dalam menyelesaikan kurikulum Program S1 Ekstensi Teknik Elektro, Departemen Teknik Elektro, Fakultas Teknik, Universitas Indonesia.

1.3 Pembatasan Masalah

Dalam Pengembangan dan Evaluasi *Smart Stick* Sebagai Alat Bantu Untuk Orang Dengan Kecacatan Visual ini akan dibahas tentang hal-hal sebagai berikut :

1. Pembatasan jarak baca sensor ultrasonik sejauh 100 cm.
2. Pembatasan kondisi pembacaan benda pada masing-masing sensor ultrasonik dan ketika tidak adanya penghalang.
3. FPGA sebagai *controller* yang menghubungkan modul *input* dan modul *output*.
4. Sensor ultrasonik sebagai *switch* memberikan sinyal input berupa logik.
5. Output sistem berupa bunyi yang dikeluarkan *buzzer*.
6. LCD dan LED sebagai indikator sistem berjalan secara terus menerus.

1.4 Metodologi Penulisan

Metode yang digunakan penulis dalam penyusunan tugas akhir ini adalah sebagai berikut :

1. Metode Wawancara / *Interview*
Dengan metode wawancara ini, penulis mengadakan peninjauan langsung dan bertanya kepada pembimbing untuk memperoleh informasi yang dibutuhkan untuk menunjang penyusunan laporan.
2. Metode Observasi / Pengamatan
Dengan metode observasi ini, penulis mengamati dan menyelidiki alat bantu berjalan orang buta berbasis FPGA.
3. Metode Pengumpulan Data
Dengan metode pengumpulan data ini, penulis mengumpulkan data-data referensi yang dibutuhkan dan berhubungan dengan judul seminar yang diajukan penulis.

1.5 Sistematika Penulisan

Laporan seminar ini terdiri dari tiga bab yang terbagi ke dalam beberapa sub

bab dan disusun untuk memudahkan pemahaman. Sistematika penulisan laporan seminar ini adalah sebagai berikut :

BAB I PENDAHULUAN

Bab pertama ini berisi latar belakang masalah, perumusan masalah, pembatasan masalah, tujuan penulisan, metodologi penulisan, dan sistematika penulisan skripsi ini.

BAB II DASAR TEORI

Bab kedua ini berisi penjelasan dasar teori tentang FPGA, input, dan output, serta instruksi dan bahasa yang dipergunakan.

BAB III PERANCANGAN DAN REALISASI

Bab ketiga ini berisi tentang berbagai aspek yang berhubungan dengan perancangan dan realisasi dari *smart stick* untuk orang buta berbasis FPGA.

BAB IV PENGUJIAN DAN ANALISA DATA

Bab keempat ini berisi pengujian sistem serta analisa data dan penjelasan tentang software yang digunakan.

BAB V PENUTUP

Bab kelima ini berisi penutup berdasarkan rancang bangun *smart stick* untuk orang buta berbasis FPGA.

BAB 2

LANDASAN TEORI

2.1 Sejarah dan Deskripsi FPGA

2.1.1 Sejarah FPGA

Field Programmable Gate Array (FPGA) telah 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. Sementara itu Lattice, Actel, dan Quicklogic merupakan perusahaan-perusahaan yang lebih kecil dan punya “pasar khusus”.

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.

2.2 Pengenalan FPGA

2.2.1 Pengertian FPGA

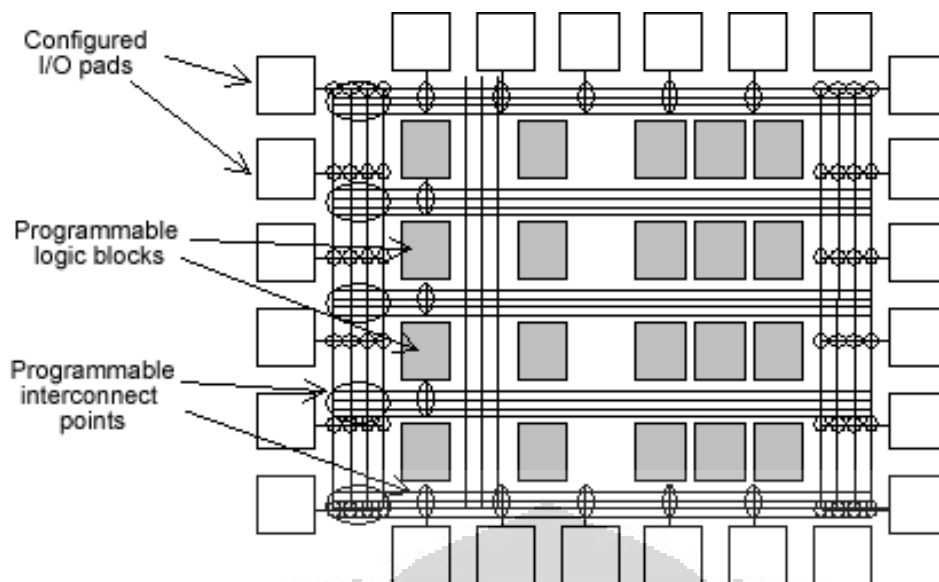
Field-Programmable Gate Array (FPGA) adalah komponen elektronika dan semikonduktor yang mempunyai komponen gerbang terprogram (programmable logic) dan sambungan terprogram. Komponen gerbang terprogram yang dimiliki meliputi jenis gerbang logika biasa (AND, OR, XOR, NOT) maupun jenis fungsi matematis dan kombinatorik yang lebih kompleks (decoder, adder, subtractor, multiplier, dll). Blok-blok komponen di dalam FPGA bisa juga mengandung elemen memori (register) mulai dari flip-flop sampai pada RAM (Random Access Memory). FPGA merupakan kombinasi antara rangkaian kombinasional dan

sekuensial sehingga dapat diimplementasikan pada silikon yang merupakan salah satu bagian dari *Application-Specific Integrated Circuit* (ASIC).



Gambar 2.1 FPGA Xilinx Spartan [9]

FPGA (Field Programmable Gate Array) merupakan sebuah IC digital sering digunakan untuk implementasi rangkaian digital. IC digital ini umumnya terdiri atas 3 bagian yaitu : Configure Logic Blocks (CLB), Input/ Output Blocks (IOB), dan *Programmable Switch Matrix* (PSM). CLB merupakan blok untuk membangun komponen-komponen *combinational/ sequential*. Agar dapat membentuk suatu CLB dibutuhkan tabel kebenaran dalam membangun jalur logika. IOB merupakan blok yang digunakan untuk mengirim sinyal keluar dari *chip* dan sekaligus membaca sinyal yang masuk ke dalam *chip*. PSM merupakan penghubung yang bisa diatur untuk menghubungkan antara CLB dan CLB dengan IOB secara horizontal dan vertikal, serta mampu mengirimkan data ke banyak sumber untuk dijadikan input sekaligus (*fan out*) untuk *multiple output*. Output dari CLB yang satu dapat dihubungkan ke input CLB yang lain melalui *multiple PSM*.

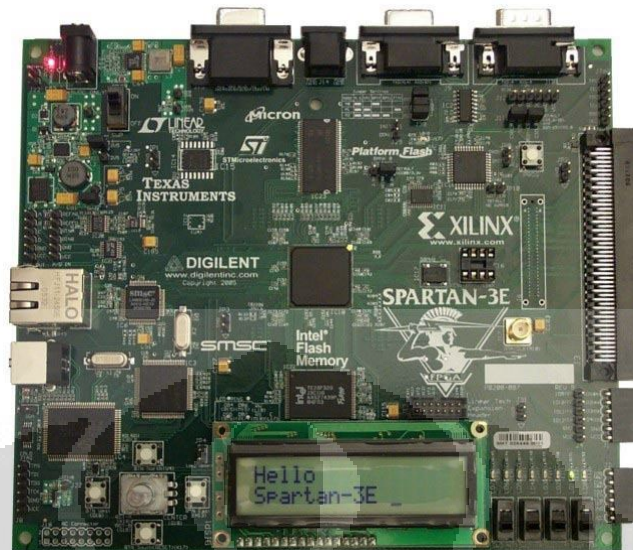


Gambar 2.2 Arsitektur FPGA [9]

Modul FPGA yang penulis gunakan adalah Xilinx Spartan 3-E, pada modul FPGA Xilinx Spartan-3E telah tersedia komponen pendukung lainnya seperti :

- LED (Light Emitting Diode)
LED adalah sebuah komponen yang berfungsi sebagai lampu indikator. Xilinx Spartan 3-E memiliki 8 buah LED.
- Switch
Pada modul FPGA Xilinx Spartan 3-E terdapat 4 switch geser. Switch ini akan memberikan nilai *true* (1) ketika tuasnya berada diposisi atas dan akan memberikan nilai *false* (0) ketika tuasnya berada diposisi bawah.
- VGA (Video Grapichs Array) port
VGA port berfungsi menampilkan data digital menggunakan alat analog, seperti monitor komputer.
- PS/2
FPGA juga menyediakan port untuk menghubungkan mouse dan keyboard PS/2.
- Clock
FPGA menyediakan clock source sebesar 50MHz.
- Dua 9-pin RS-232 serial port
FPGA juga menyediakan serial port yang dapat dihubungkan ke komputer.
- 512-Kbyte SRAM
- Ethernet 10/100 PHY
- 2-Mbyte serial Flash

- 16 karakter – 2 baris LCD

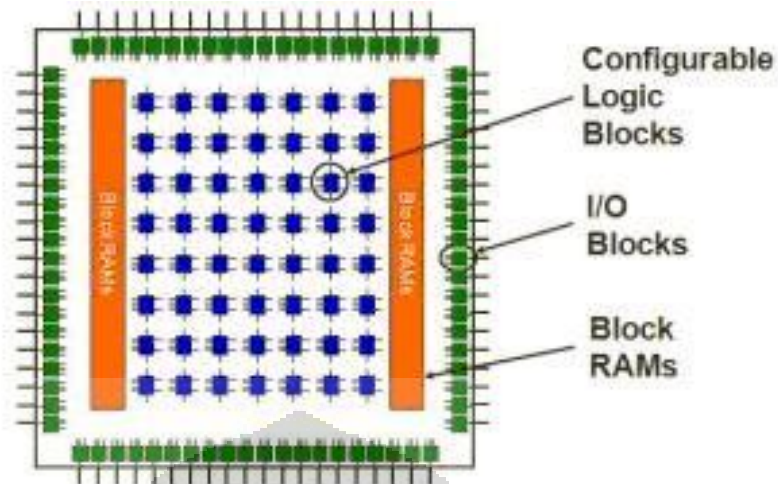


Gambar 2.3 Modul Xilinx Spartan-3E [2]

2.2.2 Cara Kerja FPGA

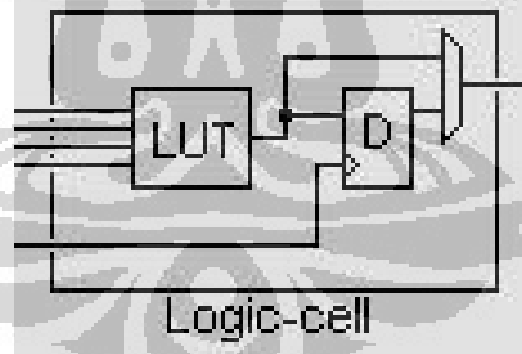
Bentuk sebuah FPGA tak berbeda jauh dengan bentuk IC-IC lainnya. Hanya saja, bila dilihat dari isinya FPGA memiliki bagian yang berbeda dengan komponen IC pada umumnya. Berikut isi dari FPGA pada umumnya:

1. Configure Logic Blocks (CLB). Bagian 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.



Gambar 2.4 Bagian Dalam FPGA [9]

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 look up table (LUT), D flip-flop, dan sebuah multiplexer 2 ke 1.



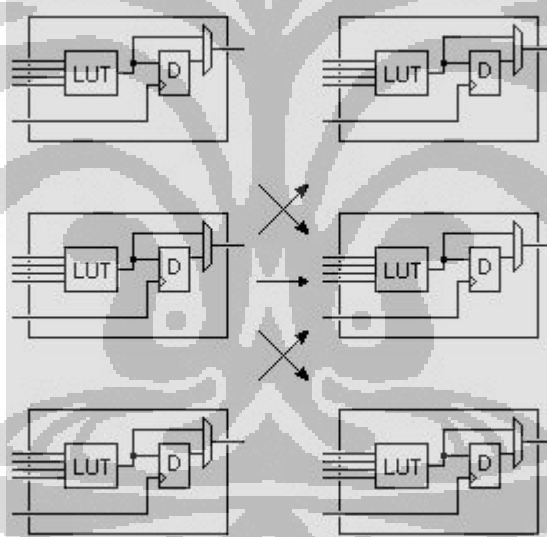
Gambar 2.5 Isi Logic Cell [9]

1. Look Up Table (LUT) merupakan sejenis RAM (Random Acces 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) dalah 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.



Gambar 2.6 Logic Cell yang Terhubung satu sama lain [9]

2.2.3 Konfigurasi FPGA

Sebuah FPGA (Field Programmable Gate Array) terbagi atas dua kondisi yaitu kondisi awal (configuration mode) dan kondisi pemakai (user mode). Ketika pertama kali FPGA dihidupkan, maka FPGA ini berada dalam kondisi awal. Hal ini disebabkan karena FPGA masih dalam keadaan awal, fresh dan belum terdapat suatu program di dalamnya. Sebelum menggunakan FPGA, 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. 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 microkontroller 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).

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).



Gambar 2.7 Proses Download dari Komputer ke FPGA [9]

2.2.4 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). FPGA Board terdapat sebuah IC yang terintegrasi dengan komponen-komponen lain seperti kristal, LED, resistor, kapasitor, push button, dsb.

Perancangan perangkat lunak (software) terdiri atas 4 tahap :

1. *Design Entry.*
2. *Simulation.*
3. *Synthesis and place-and-route.*
4. *Programming through special cables (JTAG).*

Biasannya *software* yang dibagikan secara gratis hanya untuk jenis FPGA tingkat rendah sampai menengah atau demo. Sedangkan untuk tingkat atas maka *software* pendukungnya tidaklah gratis. Berikut beberapa *software* pendukung yang diberikan gratis atau demo :

1. Perusahaan Xilinx terkenal dengan software miliknya yang bernama **ISE WebPack**
2. Perusahaan Altera terkenal dengan software miliknya yang bernama **Quartus II Web Edition**

2.3 Verilog

Verilog HDL adalah bahasa pemrograman yang digunakan untuk mendesain IC seperti membuat sistem digital. Aplikasi yang biasa dibuat menggunakan verilog adalah gerbang-gerbang logika (AND, OR, NOT, NAND), flip-flop, counter.

Keuntungan yang dimiliki oleh bahasa pemrograman verilog dibandingkan dengan VHDL (Very High Speed Hardware Description Language) adalah :

1. Verilog memiliki kecepatan *running* lebih cepat dibandingkan dengan VHDL, karena Verilog dibangun dari bahasa C sedangkan VHDL dibangun dari bahasa PASCAL.
2. Verilog memiliki kode-kode yang lebih sederhana dan mudah di mengerti dibandingkan dengan VHDL.

Kekurangan yang dimiliki oleh Verilog dibandingkan dengan VHDL adalah Verilog tidak mampu merancang sistem digital yang kompleks dan berskala besar.

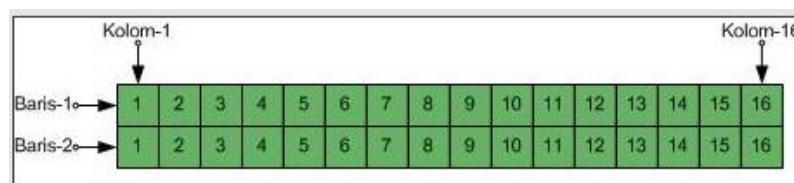
Verilog merupakan bahasa simulasi yang digunakan pada perangkat keras seperti FPGA. Perbedaan antara bahasa simulasi dan bahasa pemrograman adalah pada bahasa simulasi keterangan waktu dapat ditampilkan dan apakah sistem berjalan secara terus menerus.

Tipe pemrograman yang bisa dibuat menggunakan Verilog :

1. Tingkat Perilaku (Behavior Level)
2. Tingkat Transfer Register (Register Transfer Level)
3. Tingkat Gerbang (Gate Level)
4. Tingkat Pensaklaran (Switch Level)

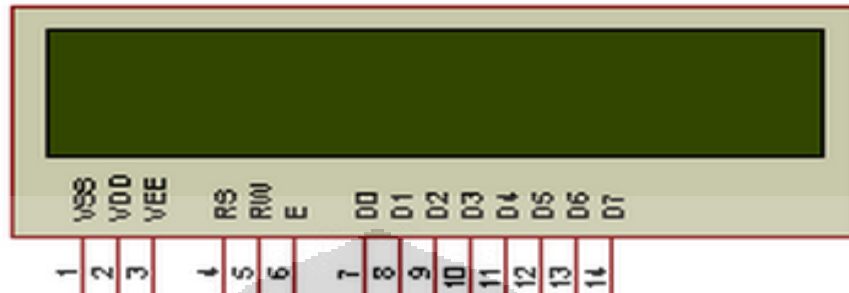
2.4 LCD 16x2

Liquid Cristal Display (LCD) 16x2 adalah media untuk menampilkan karakter seperti huruf dan angka. LCD 16x2 memiliki 2 baris yang masing-masing berisikan 16 kolom karakter seperti Gambar 2.8 berikut.



Gambar 2.8 LCD 16x2 [8]

Sebelum melakukan perancangan *interface* menggunakan LCD 16x2 yang perlu diperhatikan adalah mengetahui pin-pin *interface* pada LCD seperti Gambar 2.9 dibawah ini.



Gambar 2.9 Interface LCD 16x2 [8]

Fungsi masing-masing pin yang terdapat pada LCD 16x2 adalah :

1. VSS = Ground
2. VDD = +5 Volt
3. VEE = Penggerak LCD
4. RS= H/L (H = memasukkan data, L = memasukkan instruksi)
5. RW = H/L (H = baca, L = tulis)
6. E = *Enable Signal*
7. D0
8. D1
9. D2
10. D3
11. D4
12. D5
13. D6
14. D7 = D0 sampai D7 merupakan data bus.

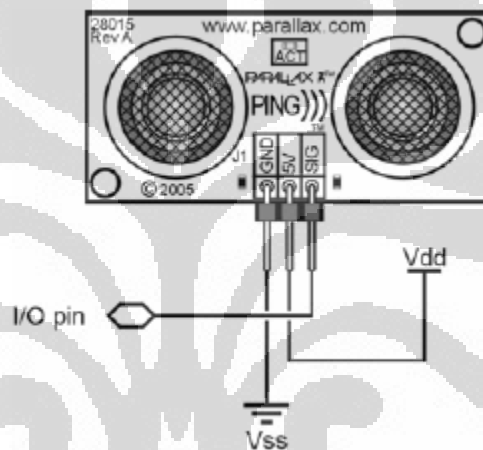
2.5 Sensor Ultrasonik PING

Sensor ultrasonik PING adalah sensor yang bekerja berdasarkan prinsip pantulan gelombang suara dan digunakan untuk mendeteksi keberadaan suatu objek tertentu, frekuensi kerja sensor ultrasonik diatas gelombang suara dari 40KHz hingga 400KHz. Sensor ultrasonik terdiri dari rangkaian pemancar ultrasonik yang disebut *transmitter* dan rangkaian penerima ultrasonik yang

disebut *receiver*. Sinyal ultrasonik yang dihasilkan akan dipancarkan dari *transmitter* ultrasonik ketika sinyal ultrasonik menyentuh adanya benda penghalang maka sinyal tersebut akan dipantulkan dan diterima oleh rangkaian penerima (*receiver*) ultrasonik. Sinyal ultrasonik yang diterima oleh rangkaian *receiver* akan dikirim ke modul FPGA untuk selanjutnya diolah untuk mengaktifkan *output* berupa *buzzer*.

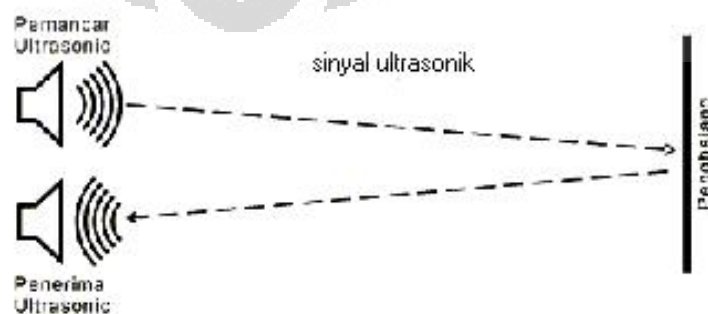
Sensor jarak ultrasonik ping adalah sensor 40 khz produksi parallax yang banyak digunakan untuk aplikasi atau kontes robot cerdas. Kelebihan sensor ini adalah hanya membutuhkan 1 sinyal (SIG) selain jalur 5 v dan ground.

Perhatikan gambar 2.10 dibawah ini :



Gambar 2.10 Sensor Ultrasonik PING [7]

Besar amplitudo sinyal elektrik yang diterima oleh rangkaian penerima tergantung dari jauh dekatnya objek yang dideteksi. Proses *sensing* pada sensor ini menggunakan metode pantulan untuk menghitung jarak antara sensor dengan obyek sasaran.



Gambar 2.11 Prinsip Kerja Sensor Ultrasonik [7]

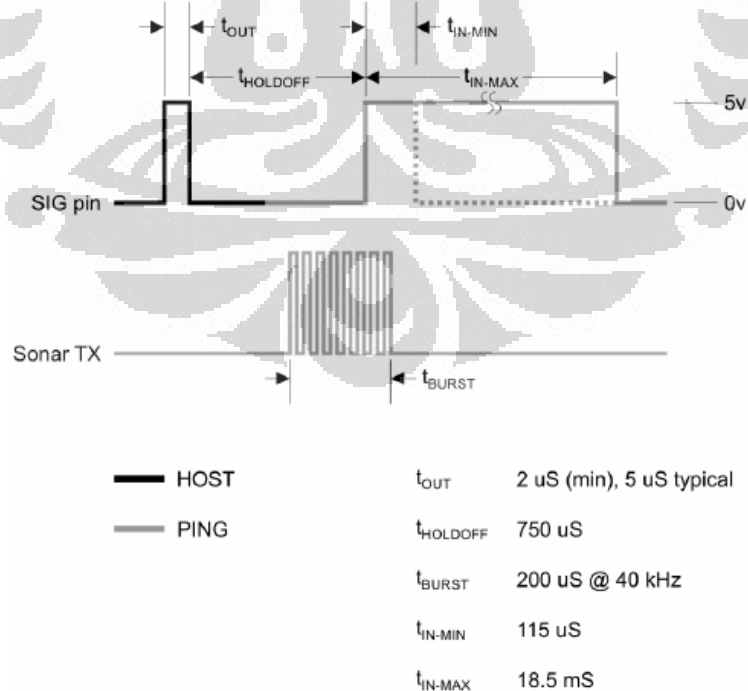
Prinsip kerja sensor ultrasonik adalah :

1. Sinyal ultrasonik yang dipancarkan oleh pemancar ultrasonik (transmitter) memiliki frekuensi 40KHz digunakan untuk mengukur jarak benda.
2. Sinyal yang dipancarkan akan merambat sebagai sinyal/ gelombang suara yang memiliki kecepatan bunyi berkisar 340 m/s atau 1 cm setiap 29.034 uS. Sinyal tersebut akan dipantulkan dan diterima oleh rangkaian penerima (receiver) ultrasonik. Selama rangkaian penerima menunggu pantulan, sensor ultrasonik akan menghasilkan pulsa dan pulsa akan berhenti ketika suara pantulan diterima oleh rangkaian penerima pada sensor ultrasonik. Lebar pulsa yang dihasilkan dari jarak tempuh gelombang ultrasonik yang dipancarkan hingga memantul kembali karena adanya objek yang menghalangi sensor.
3. Setelah sinyal diterima oleh rangkaian penerima (transmitter), kemudian sinyal akan diproses untuk menghitung jaraknya menggunakan rumus :

$$\text{Jarak} = (\text{Lebar Pulsa}/29.034\mu\text{S})/2 \quad (\text{dalam cm})$$

Atau

$$\text{Jarak} = (\text{Lebar Pulsa}/0.034442)/2 \quad (\text{dalam cm})$$



Gambar 2.12 Pulsa Ping))) Parallax Ultrasonic Range Finder [10]

2.6 Buzzer

Buzzer adalah sebuah komponen elektronika yang berfungsi untuk mengubah getaran listrik menjadi getaran suara. Pada dasarnya prinsip kerja buzzer hampir sama dengan loud speaker, jadi buzzer juga terdiri dari kumparan yang terpasang pada diafragma dan kemudian kumparan tersebut dialiri arus sehingga menjadi elektromagnet, kumparan tadi akan tertarik ke dalam atau keluar, tergantung dari arah arus dan polaritas magnetnya, karena kumparan dipasang pada diafragma maka setiap gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga membuat udara bergetar yang akan menghasilkan suara. Buzzer biasa digunakan sebagai indikator bahwa proses telah selesai atau terjadi suatu kesalahan pada sebuah alat (alarm).



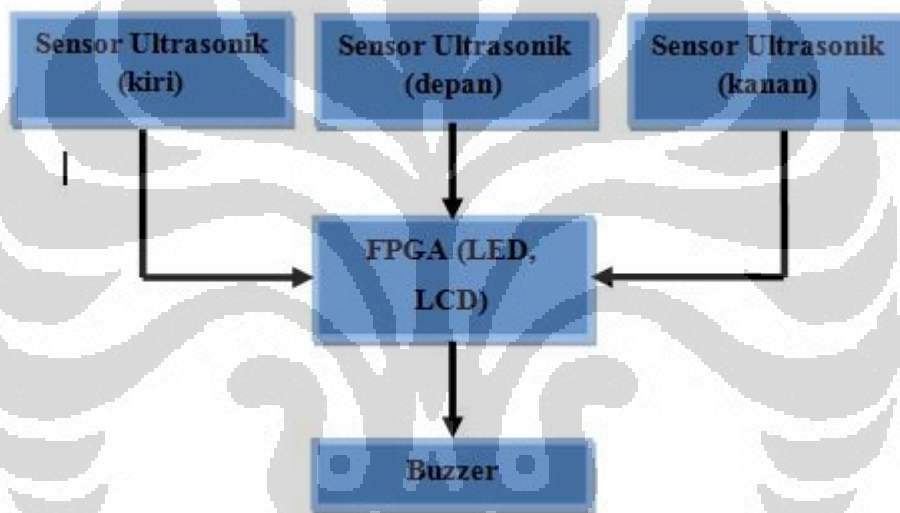
Gambar 2.13 Buzzer [11]

BAB 3

PERANCANGAN SMART STICK UNTUK ORANG BUTA BERBASIS FPGA XILINX SPARTAN-3E

3.1 Perancangan Alat

Pada bagian ini, akan dibahas tentang deskripsi dan spesifikasi dari FPGA, serta perancangan FPGA dan perancangan modul I/O yaitu modul input sensor ultrasonik, dan modul output *buzzer*. Sebelum melakukan perancangan, terlebih dahulu harus membuat diagram alir sebagai acuan dalam melakukan perancangan modul I/O alat bantu berjalan orang buta berbasis FPGA.



Gambar 3.1 Blok Diagram Smart Stick

Berdasarkan blok diagram diatas dapat kita modul sensor ultrasonik merupakan input utama dari alat ini. Pada alat ini akan digunakan tiga buah sensor ultrasonik sebagai input yang akan dipasang pada sebuah tongkat disisi depan, kanan, dan kiri. Masing-masing sensor ultrasonik ini berfungsi untuk mendeteksi ada tidaknya benda yang menghalangi jalan orang buta tersebut sehingga mengurangi resiko menabrak suatu benda. Sinyal sensor ultrasonik yang diterima FPGA ini akan bernilai “0” apabila sensor tidak membaca adanya benda yang menghalangi namun sinyal sensor ultrasonik akan bernilai “1” apabila sensor mendeteksi adanya benda yang menghalangi dan akan segera diproses sebagai input. FPGA merupakan rangkaian pengendali kerja keseluruhan alat ini. Setelah FPGA menerima input maka FPGA akan mengaktifkan output yaitu berupa

buzzer sebagai penunjuk disebelah mana terdapat benda yang menghalangi. Jika sensor ultrasonik bagian depan mendeteksi adanya penghalang maka *buzzer* akan mengeluarkan bunyi “tit nada sedang” sebagai tanda peringatan untuk tidak meneruskan berjalan dikarenakan terdapat penghalang didepannya. Ketika sensor ultrasonik bagian depan mendeteksi adanya benda penghalang maka sensor akan mengirim sinyal “010” ke FGPA untuk mengaktifkan *buzzer*, sensor ultrasonik sebelah kiri mendeteksi adanya penghalang akan mengirimkan sinyal input “100” ke FPGA dan akan mengaktifkan *buzzer* untuk mengeluarkan suara “tit nada lambat”, sensor ultrasonik sebelah kanan mendeteksi adanya penghalang akan mengirimkan sinyal input “001” ke FPGA dan akan mengaktifkan *buzzer* untuk mengeluarkan suara “tit nada tinggi”.

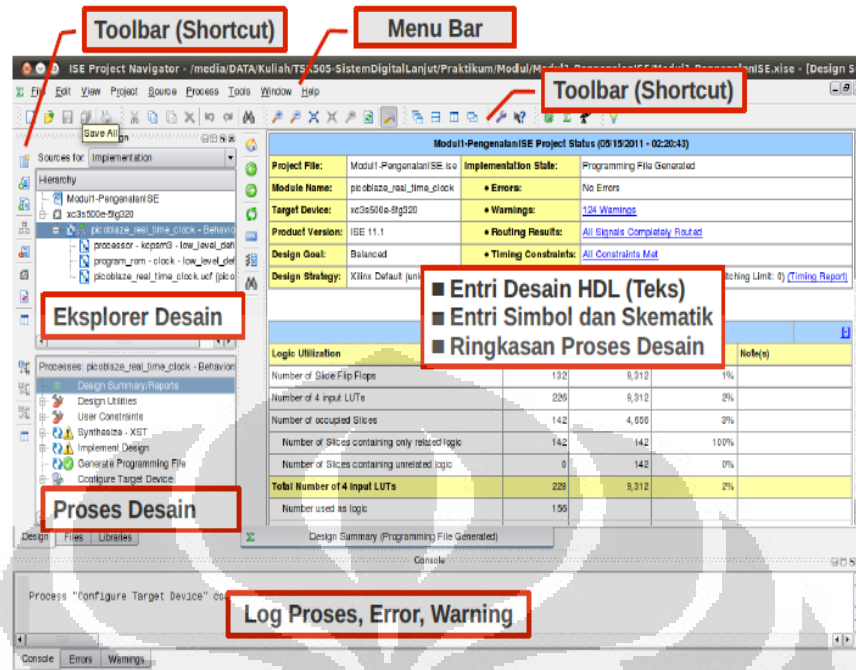
3.2 ISE Webpack 14.1

Software Xilinx ISE Webpack 14.1 akan digunakan sebagai GUI untuk merancang problem sistem digital yang diinginkan. Selanjutnya Xilinx ISE Webpack 14.1 disebut sebagai ISE. ISE berisi tool-tool untuk mengembangkan rancangan sistem digital terprogram. Tool yang akan digunakan dalam perancangan ini adalah:

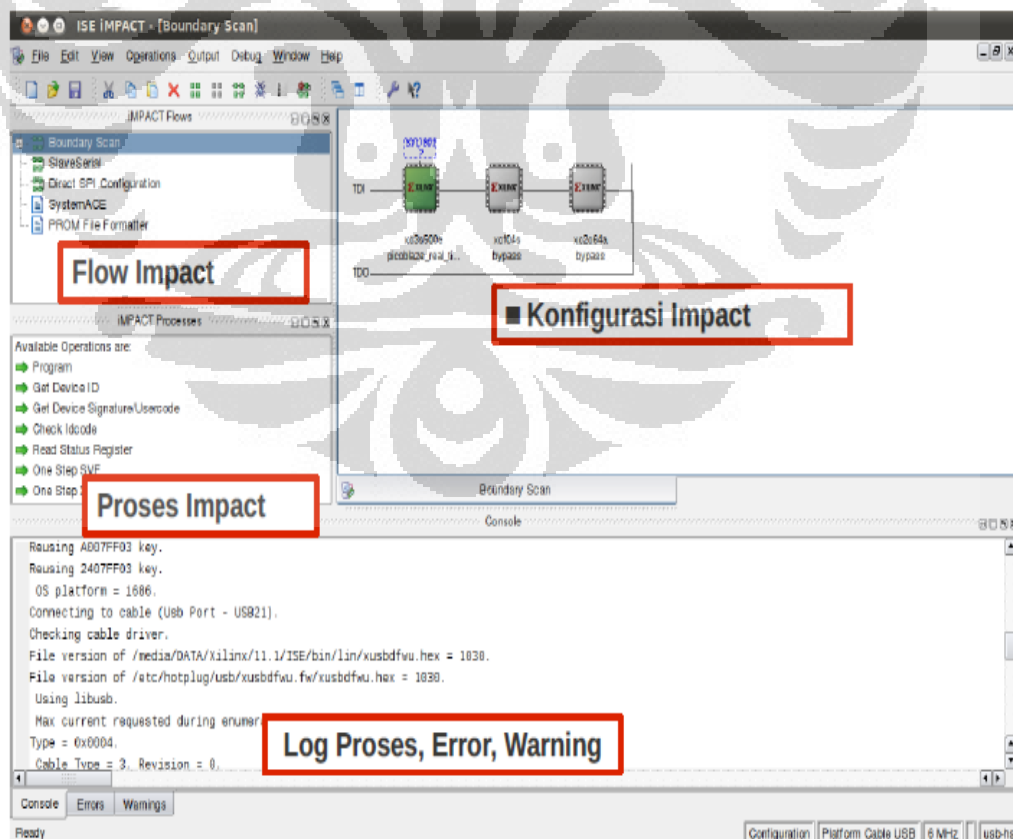
1. **ISE Project Navigator** digunakan untuk membuat proyek baru, memasukan entry desain (HDL, skematik), kompilasi desain (sintesis), mengimplementasikan desain dan membangkitkan file programming (konfigurasi FPGA). Tool lain dari ISE dipanggil dari Project Navigator ini;
2. **ISE iMPACT** digunakan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

GUI dari program ISE Project Navigator diperlihatkan dalam **Gambar 3.2**.

GUI dari program ISE iMPACT diperlihatkan dalam Gambar 3.3.



Gambar 3.2 ISE Project Navigator GUI: Explorer, Proses, Entry dan Log [4]

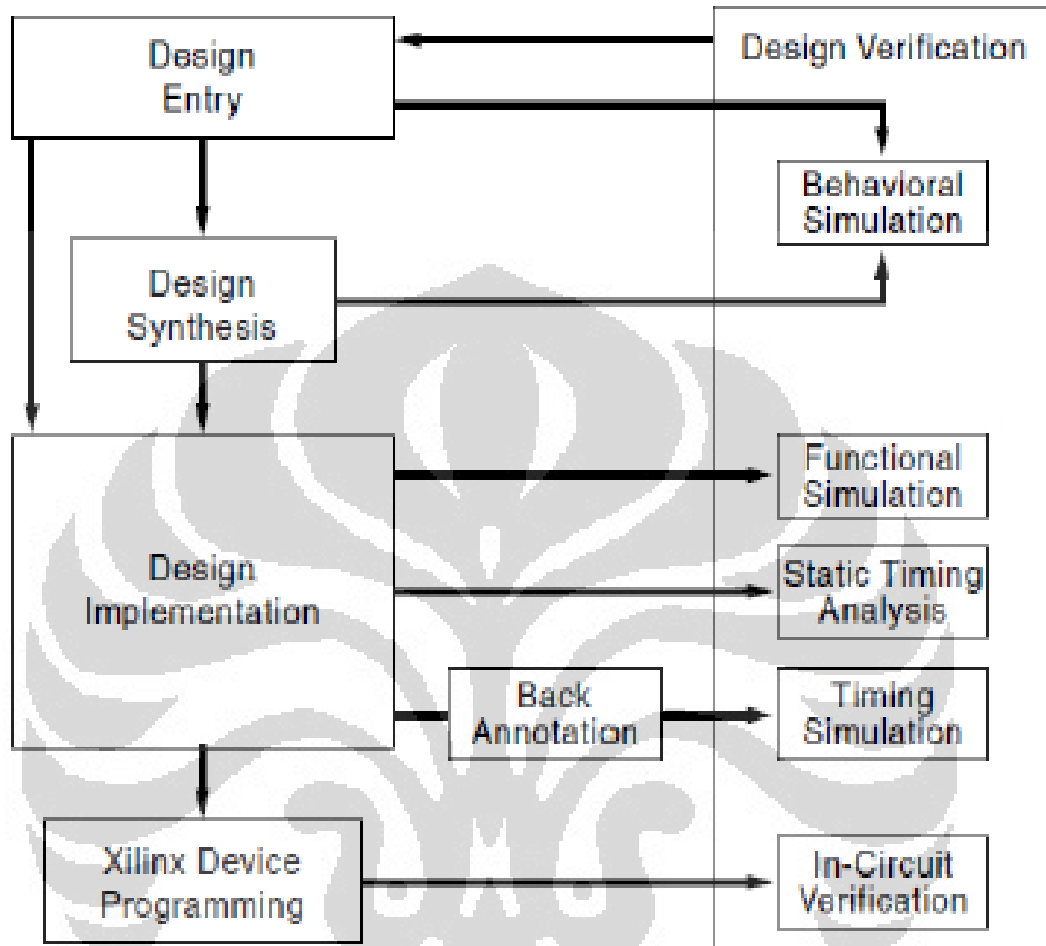


Gambar 3.3 ISE iMPACT: Konfigurasi Boundary Scan (JTAG) [4]

Perancangan sistem digital menggunakan ISE mempunyai metodologi (**Gambar 3.4**). 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.

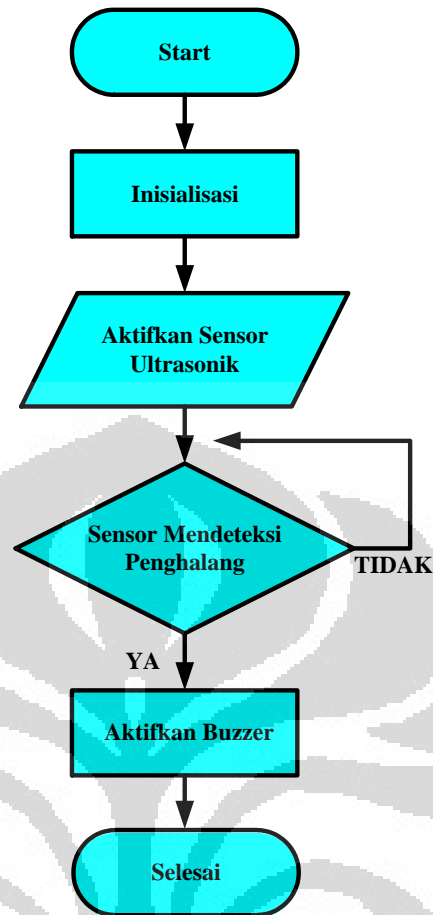
Dalam setiap tahap, verifikasi dilakukan untuk memastikan kebutuhan dan konstrain terpenuhi.



Gambar 3.4 Metodologi desain sistem digital menggunakan ISE: entri desain, sintesis, implementasi dan pemrograman [4]

3.2.1 Perancangan *Software*

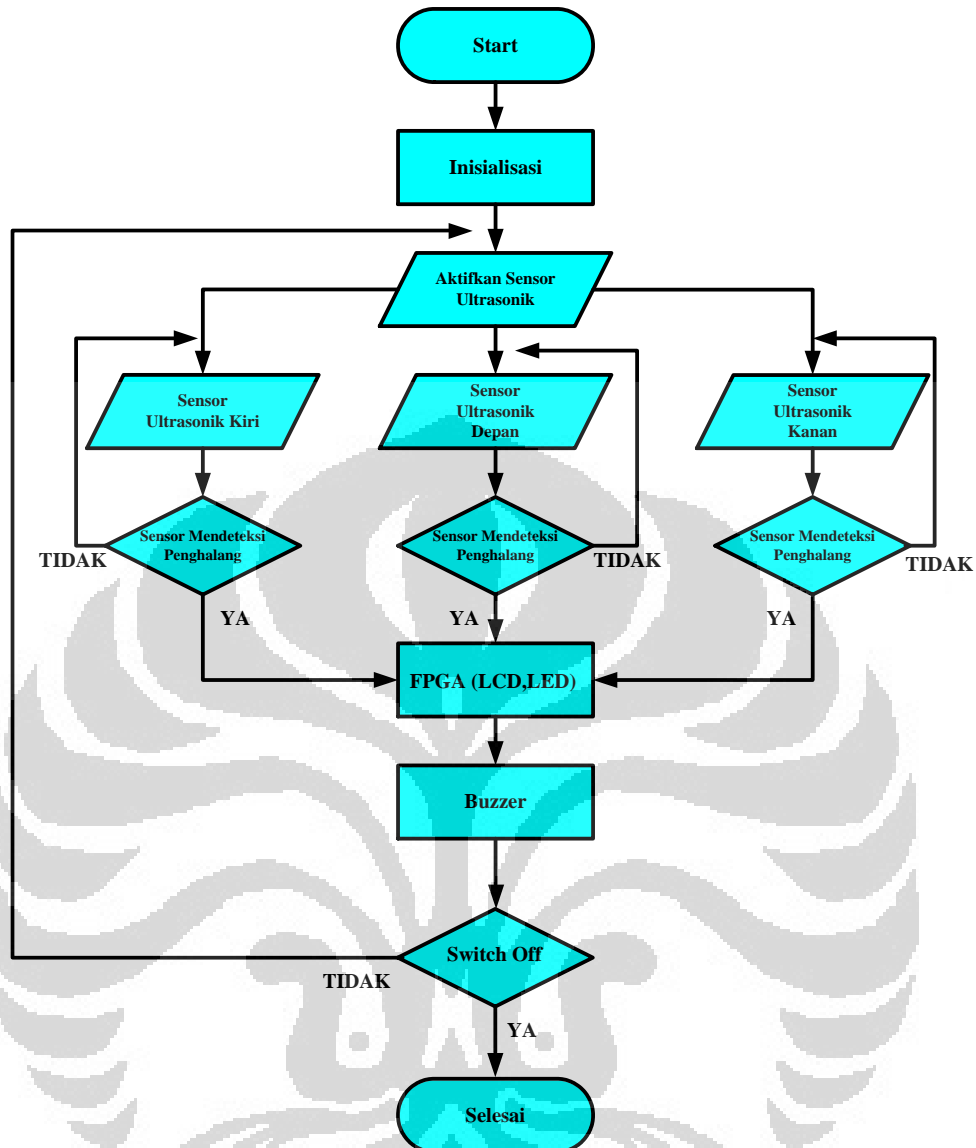
Tahap perancangan software ini untuk mengetahui cara pembuatan program dari sistem alat bantu berjalan orang buta berbasis FPGA dengan menggunakan bahasa pemrograman Verilog. Gambar dibawah ini menunjukkan diagram alir rancangan program alat bantu berjalan orang buta berbasis FPGA. Saat sensor ultasonik diaktifkan maka sensor akan mulai mendeteksi ada tidaknya benda yang menghalangi, ketika sensor mendeteksi keberadaan suatu benda yang menghalangi sensor akan segera memberikan sinyal sebagai input ke FPGA dan akan segera mengaktifkan output berupa buzzer yang akan mengeluarkan suara peringatan.



Gambar 3.5 Diagram Alir Cara Kerja Sistem

3.3 Cara Kerja Sistem

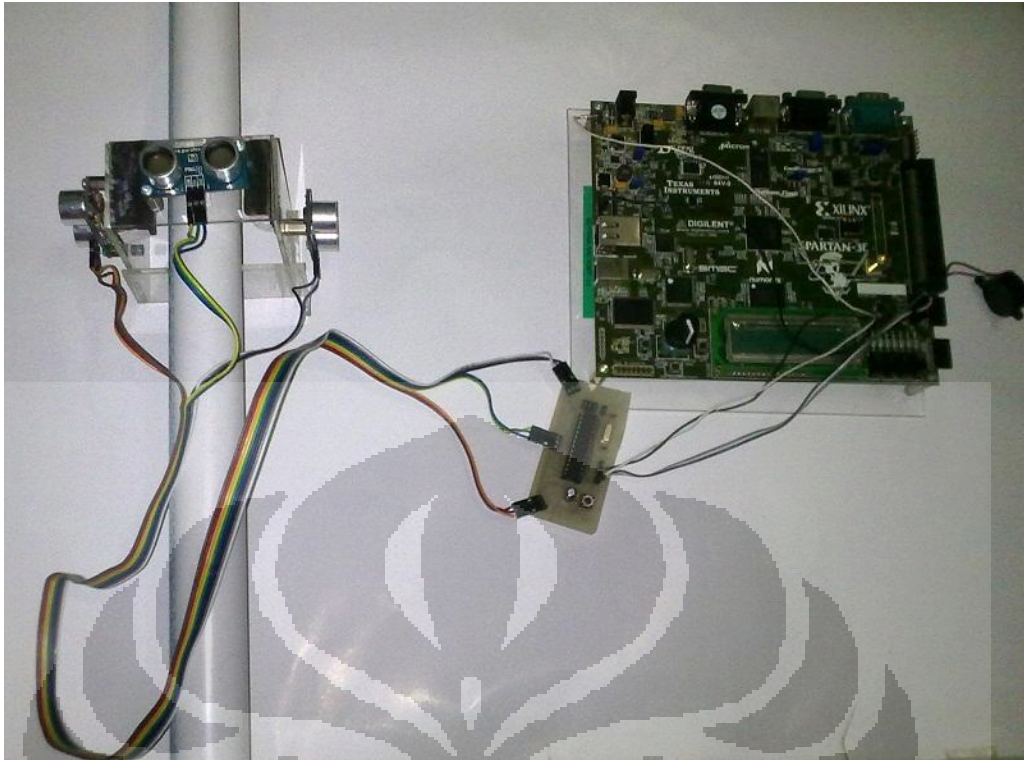
Gambar diagram alir dibawah ini menunjukkan cara kerja keseluruhan sistem yang telah dirancang. Ketika saklar pada FPGA dinyalakan maka Masing-masing sensor ultrasonik akan aktif dan akan memberikan sinyal input berupa *logic* ke FPGA ketika setiap sensor ultrasonik mendeteksi adanya benda penghalang, FPGA akan segera merespon sinyal *input* dari sensor ultrasonik dan segera mengaktifkan modul *output* yaitu mengeluarkan bunyi peringatan melalui sebuah buzzer sehingga orang yang menggunakan alat ini dapat mengetahui keadaan jalan yang akan dilaluinya.



Gambar 3.6 Diagram Alir Smart Stick Untuk Orang Buta Berbasis FPGA Xilinx Spartan-3E

3.4 Desain Smart Stick

Gambar dibawah ini merupakan desain *smart stick* yang akan dibuat, Bagian sensor ultrasonik ditempatkan agak miring dikarenakan apabila pengguna smart stick ini berjalan maka posisi smart stick tidak akan tegak lurus melainkan miring 30 derajat, sehingga tidak akan mengganggu proses pembacaan dari sensor ultrasonik. Pada bagian atas smart stick terdapat buzzer port agar dapat mengeluarkan suara peringatan, selain itu juga terdapat saklar untuk menghidupkan dan mematikan keseluruhan sistem apabila alat ini sedang tidak digunakan. Di bagian bawah tongkat ini akan ditambahkan roda untuk memudahkan pengguna dalam menggunakan *smart stick* dan juga untuk mengurangi adanya getaran yang dapat merusak komponen *smart stick* tersebut.



Gambar 3.7 Desain Smart Stick

BAB 4

PENGUJIAN DAN ANALISA DATA

Tahap pengujian dilakukan untuk mengetahui seberapa besar perbedaan antara perencanaan awal modul I/O dengan keadaan akhir modul I/O setelah diaplikasikan. Modul I/O yang diuji masing-masing adalah modul input sensor ultrasonik, modul output *buzzer*. Dengan dilakukannya tahap pengujian ini, diharapkan dapat menjadi referensi dalam merancang sebuah modul I/O yang handal dan untuk melihat tingkat kemampuan kerja modul I/O ketika sudah diaplikasikan pada sistem.

4.1 Pengujian dan Analisa Pada Modul Input Sensor Ultrasonik

4.1.1 Deskripsi Pengujian

1. Sistem yang Diuji

Sistem yang akan diujikan yaitu modul input sensor ultrasonik.

2. Tujuan Pengujian Modul Input Sensor Ultrasonik adalah :

- a. Untuk memastikan kondisi modul input sensor ultrasonik berfungsi dengan baik.
- b. Untuk mengetahui cara kerja modul input sensor ultrasonik sebagai input.
- c. Menganalisa perancangan program dan *hardware* yang dibuat sesuai dengan keinginan.

3. Target Pengujian Modul Input Sensor Ultrasonik

Modul input sensor ultrasonik dapat berfungsi dengan baik dan sesuai dengan perencanaan awal.

4. Data Lingkungan Pengujian

Lokasi Pengujian : Laboratorium Digital Elektro

Tanggal Pelaksanaan : 11 Juni 2012

Penguji : Martin Chorazon A

4.1.2 Prosedur Pengukuran dan Pengujian

4.1.2.1 Daftar Peralatan Pengujian

Tabel di bawah ini merupakan daftar peralatan yang digunakan dalam pengujian Modul Input Sensor Ultrasonik.

Tabel 4.1 Daftar Alat Pengujian Modul Input Sensor Ultrasonik

No.	Nama Alat	Fungsi
1	Multimeter	Mengukur tegangan output
2	Modul FPGA	Modul input dan output
3	Kabel Penghubung	<i>Wiring</i> modul FPGA
4	<i>Notebook / PC</i>	Menulis dan men- <i>download</i> program ke FPGA

4.1.2.2 Prosedur Pengujian Modul Input Sensor Ultrasonik

Langkah-langkah yang harus dilakukan dalam pengujian modul input sensor ultrasonik adalah sebagai berikut :

1. Memberikan tegangan *supply* 5 Vdc ke modul input sensor ultrasonik.
2. Mengukur tegangan yang keluar dari output modul sensor ultrasonik sebelum dihalangi.

Menghalangi bagian *transmitter* dan *receiver* ultrasonik dengan suatu benda untuk memberi masukan ke input FPGA lalu mengukur tegangan pada posisi seperti langkah 2.

4.1.3 Hasil Pengujian dan Pengukuran

Dari hasil pengukuran yang telah dilakukan, maka diperoleh hasil :

Tabel 4.2 Data Hasil Pengukuran Modul Input Sensor Ultrasonik [12]

Jarak	Waktu Tempuh
10 cm	582 μ S
20 cm	1163 μ S
30 cm	1745 μ S
40 cm	2326 μ S
50 cm	2907 μ S
60 cm	3489 μ S
70 cm	4070 μ S
80 cm	4652 μ S
90 cm	5233 μ S
100 cm	5814 μ S

4.1.4 Prosedur Pengujian Modul Input Sensor Ultrasonik Pada FPGA

Langkah-langkah yang dilakukan untuk melakukan pengujian modul input sensor ultrasonik dengan menggunakan FPGA :

1. Menghubungkan FPGA dan *port* USB pada komputer.
2. Menghubungkan terminal input FPGA dengan modul input sensor ultrasonik.
3. Menghubungkan terminal output FPGA dengan modul output *buzzer*.
4. Mengaktifkan komputer dan *FPGA*.
5. Membuka *software* ISE WEBPACK.
6. Membuat *software* untuk modul input sensor ultrasonik dan meng-*compile*-nya. Bila sudah tidak ada kesalahan, program di-*download* ke *FPGA*.
7. Menjalankan modul dan mengamati hasil pengujian, apakah sudah sesuai dengan rencana.

Program untuk melakukan pengujian modul input sensor ultrasonik dapat dilihat pada BAB LAMPIRAN.

4.1.5 Pengujian Sistem Sensor Ultrasonik

Percobaan ini dilakukan untuk mendeteksi adanya benda yang menghalangi sensor ultrasonik, serta mengatur jarak baca sensor ultrasonik sejauh 100 cm sebagai jarak aman bagi pengguna *smart stick*. Sensor ultrasonik akan berlogik “1” ketika ada benda yang menghalangi, LED dan LCD pada FPGA akan aktif yang berfungsi sebagai indikator sensor ultrasonik yang sebelah mana yang terhalang oleh benda. Sensor ultrasonik yang digunakan pada smart stick sebanyak tiga buah sensor ultrasonik.

Berdasarkan percobaan diatas maka diperoleh hasil pembacaan dari masing-masing sensor ultrasonik yang diterima FPGA, perhatikan Tabel 4.3 berikut :

Tabel 4.3 Simulasi Sistem Sensor Ultrasonik

Sensor Ultrasonik	Logik	Tampilan pada LCD	LED
Depan	1	010	LD6
Kiri	1	100	LD7
Kanan	1	001	LD5
Depan, Kiri, Kanan	0	000	-



Gambar 4.1 Tampilan Data Pada LCD Ketika Sensor Mendeteksi Benda

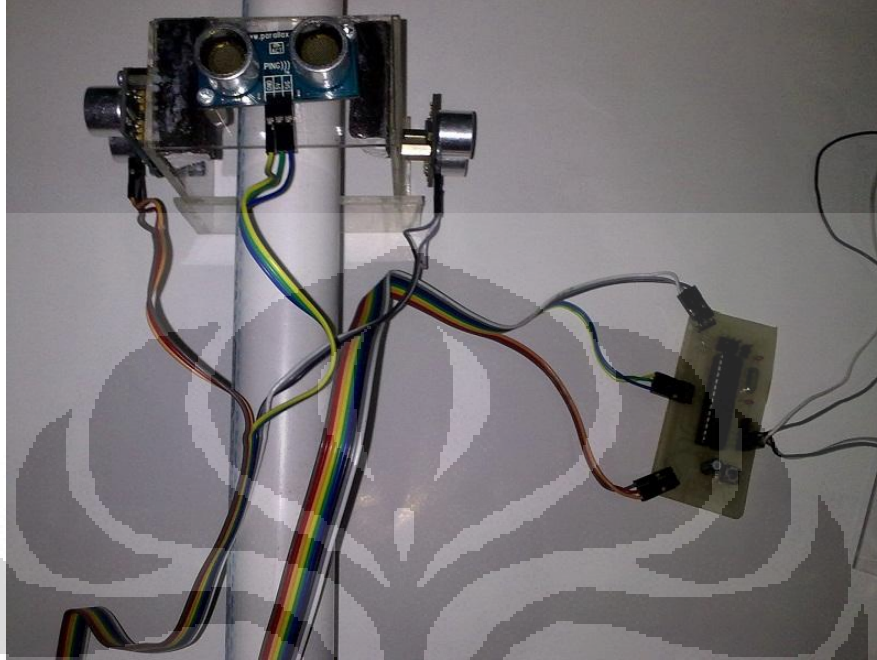
4.1.6 Analisa

Berdasarkan pengamatan hasil pengujian, dapat diketahui :

1. Sensor Ultrasonik dapat digunakan untuk mendeteksi benda yang bergerak di depannya.
2. Sensor Ultrasonik akan aktif apabila bagian *transmitter* dan *receiver* nya dihalangi oleh suatu benda. Pemantulan gelombang suara diterima oleh *receiver* akan menghasilkan logic sebagai sinyal input pada FPGA untuk mengaktifkan modul output *buzzer*.
3. Jauhnya jarak yang dapat dideteksi dipengaruhi oleh besarnya daya yang dipancarkan oleh bagian *transmitter* dan frekuensi kerja yang digunakan.
4. Kelebihan dari sensor ultrasonik adalah memiliki kemampuan mendeteksi benda yang jauh dan sangat sensitif terhadap suatu gerakan benda yang lewat di depannya.

4.1.7 Wiring Modul Input Sensor Ultrasonik

Wiring modul *input* sensor ultrasonic dapat dilihat pada gambar 4.2 dibawah ini.



Gambar 4.2 Wiring Modul Input Sensor Ultrasonik

Ketiga sensor ultrasonik ini dihubungkan ke IC Atmega yang merupakan satu kesatuan dari modul input pada alat ini. IC Atmega telah diisi program terpadu untuk mengendalikan sensor ultrasonik. Pada IC Atmega proses pembacaan sensor akan diproses dan nilai logik yang diperoleh akan dikirim ke FPGA. Penulis menambahkan IC Atmega pada modul *input* untuk meminimalisir penggunaan pin-pin pada FPGA.

4.2 Pengujian dan Analisa Pada Modul Output Buzzer

4.2.1 Deskripsi Pengujian

a. Sistem yang Diuji

Sistem yang akan diujikan yaitu modul output *buzzer*.

b. Tujuan Pengujian Modul Output *Buzzer* adalah :

- Untuk memastikan kondisi modul output *buzzer* berfungsi dengan baik.
- Untuk mengetahui cara kerja modul output *buzzer* sebagai output.

- Menganalisa perancangan program dan *hardware* yang dibuat sesuai dengan keinginan.
- c. Target Pengujian Modul Output *Buzzer*
Modul output *buzzer* dapat berfungsi dengan baik dan sesuai dengan perencanaan awal.
- d. Data Lingkungan Pengujian
Lokasi Pengujian : Laboratorium Elektro
Tanggal Pelaksanaan : 11 Juni 2012
Penguji : Martin Chorazon A

4.2.2 Prosedur Pengujian dan Pengukuran

4.2.2.1 Daftar Peralatan Pengujian

Tabel di bawah ini merupakan daftar peralatan yang digunakan dalam pengujian Modul Output *Buzzer*.

Tabel 4.4 Daftar Alat Pengujian Modul Output Buzzer

No.	Nama Alat	Fungsi
1	Multimeter	Mengukur tegangan output
2	Modul FPGA	Modul input dan output
3	Kabel Penghubung	<i>Wiring</i> modul FPGA
4	<i>Notebook</i> / PC	Menulis dan men- <i>download</i> program ke FPGA

4.2.2.2 Prosedur Pengujian Modul Output Buzzer

Langkah-langkah yang harus dilakukan dalam pengujian modul output *buzzer* adalah sebagai berikut :

1. Memberikan tegangan *supply* 6 Vdc ke modul output *buzzer*.
2. Mengukur tegangan yang keluar dari modul output *buzzer*.

4.2.3 Hasil Pengujian dan Pengukuran

Dari hasil pengukuran yang telah dilakukan, maka diperoleh hasil :

Tabel 4.5 Data Hasil Pengukuran Modul Output *Buzzer*

<i>Vin Buzzer</i>	<i>Vout Buzzer</i>
6 V	6 V

4.2.4 Pengujian Sistem Buzzer

Percobaan ini dilakukan untuk menerangkan bunyi yang dihasilkan oleh *buzzer* setelah menerima perintah dari FPGA. Bunyi yang dihasilkan oleh *buzzer* menandakan adanya penghalang dan bunyi *buzzer* akan berbeda bergantung pada sensor ultrasonik sebelah mana yang mendeteksi adanya benda. Berikut adalah data yang diperoleh.

Tabel 4.6 Simulasi Sistem *Buzzer*

Sensor Ultrasonik	Logik	Bunyi <i>Buzzer</i>
Depan	1	“tit nada sedang”
Kiri	1	“tit nada lambat”
Kanan	1	“tit nada tinggi”
Depan, Kiri, Kanan	0	-

4.2.5 Analisa

Berdasarkan pengamatan hasil pengujian, dapat diketahui :

1. *Buzzer* dapat digunakan sebagai indikator keadaan bahaya atau untuk mengetahui adanya suatu kerusakan pada sistem yang biasanya berupa suara.
2. Untuk mengatur lamanya *buzzer* berbunyi pada modul output buzzer, dapat diatur pada bagian *software* dengan menggunakan instruksi *timer* yang berisi nilai konstanta pewaktu sesuai dengan lamanya *buzzer* berbunyi.

4.3 Program

4.3.1 Program LCD

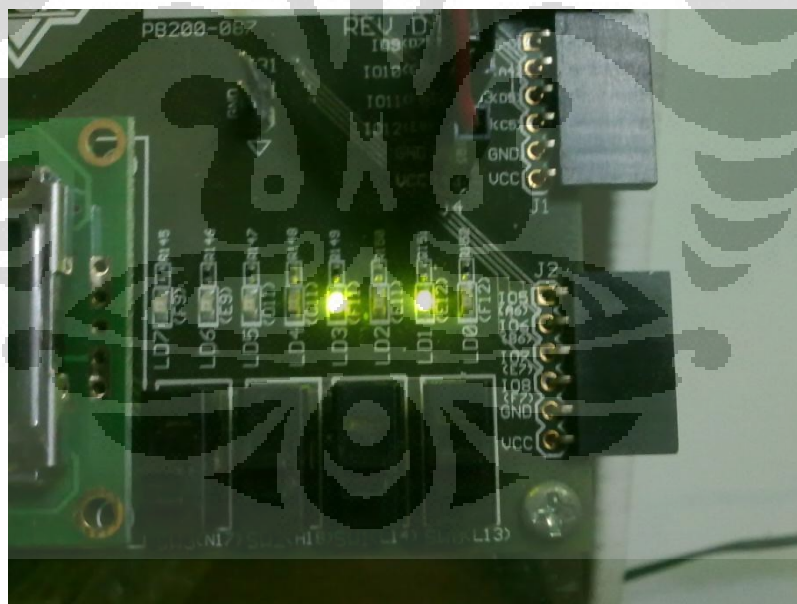
LCD yang terdapat pada FPGA akan dimanfaatkan untuk menampilkan nilai logik yang diterima FPGA dari hasil baca sensor ultrasonik. Tampilan yang akan ditampilkan pada LCD berguna untuk melihat sensor bagian depan, kiri atau kanan yang mendeteksi adanya benda penghalang

4.3.2 Program Buzzer

Program *buzzer* akan aktif setelah sensor ultrasonik membaca adanya penghalang dan *buzzer* akan mengeluarkan bunyi sesuai dengan sensor ultrasonik yang mendeteksi adanya penghalang.

4.3.3 Program LED

Program LED ini akan aktif ketika FPGA dinyalakan sebagai indikator bahwa selama FPGA tidak dalam keadaan off sistem akan terus berlangsung. LED ini juga berfungsi sebagai indikator yang menunjukkan sensor sebelah mana yang mendeteksi adanya benda.



Gambar 4.3 LED Sebagai Indikator Proses Kerja Sistem

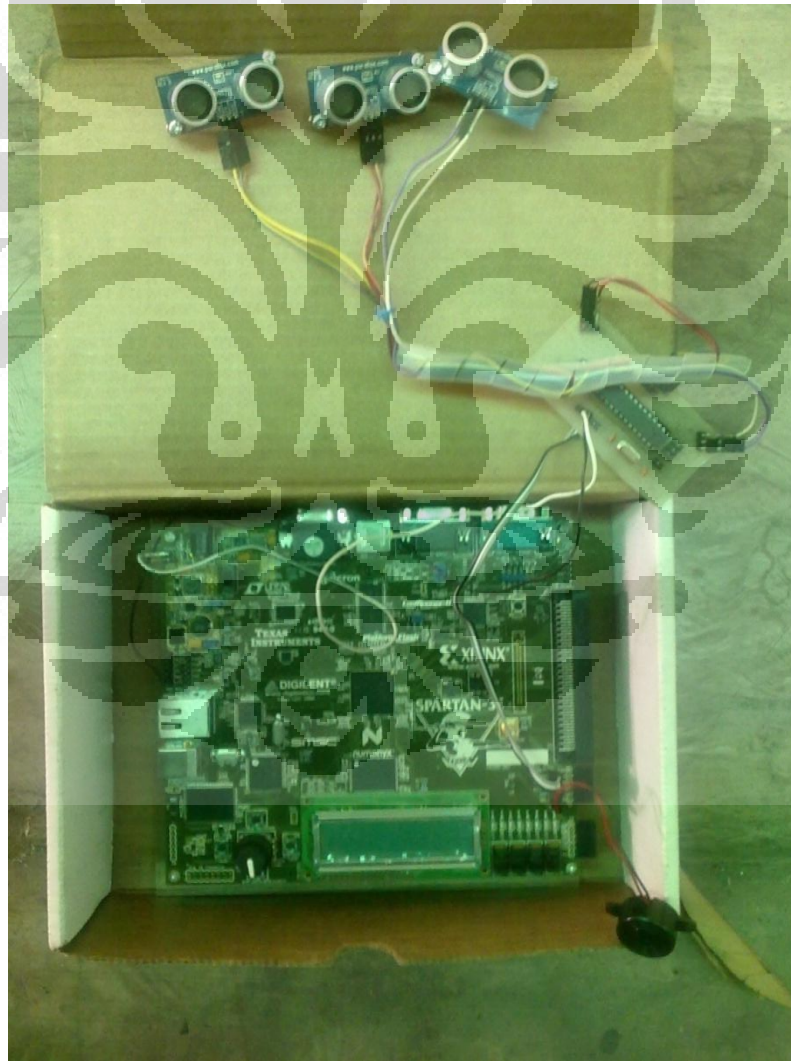
4.4 Analisa Keseluruhan Sistem

Sistem akan mulai berjalan ketika FPGA dinyalakan, ketiga sensor ultrasonik akan dalam kondisi *stand by* sampai terdeteksi adanya benda penghalang. Jarak baca ketiga sensor ultrasonik ini telah dibatasi yaitu sejauh 100 cm yang juga berfungsi sebagai jarak aman bagi pengguna *smart stick*. Ketika sensor ultrasonik

mendeteksi adanya benda penghalang maka sensor ultrasonik akan mengeluarkan logik “1” yang dikirim ke FPGA untuk mengaktifkan *buzzer*.

Buzzer akan mengeluarkan suara “tit nada rendah” ketika sensor ultrasonik sebelah kiri mendeteksi adanya benda, *buzzer* akan mengeluarkan bunyi “tit nada sedang” ketika sensor ultrasonik bagian depan mendeteksi adanya benda penghalang, *buzzer* akan mengeluarkan bunyi “tit nada tinggi” ketika sensor ultrasonik sebelah kanan mendeteksi adanya benda.

Secara bersamaan logik “1” yang diterima FPGA yang menandakan adanya benda penghalang akan mengaktifkan LCD dan LED sebagai indikator yang membuktikan bahwa sensor ultrasonik sebelah kiri, tengah atau kanan yang mendeteksi adanya benda penghalang.



Gambar 4.4 Prototype Smart Stick

BAB 5

KESIMPULAN

Perancangan serta implementasi sistem *Smart Stick* ini terdiri dari modul *input* dan modul *output* yang terhubung ke FPGA sebagai pengendali utama. Modul *input* berupa 3 buah sensor ultrasonik akan mengirimkan logik “1” ke FPGA ketika mendeteksi keberadaan penghalang, FPGA akan merespon logik tersebut untuk segera mengaktifkan *buzzer* untuk mengeluarkan bunyi peringatan sesuai dengan posisi sensor ultrasonik yang mendeteksi keberadaan penghalang.

Smart stick ini dirancang untuk membantu orang-orang yang memiliki keterbatasan dalam hal penglihatan sebagai alat bantu untuk menentukan arah berjalan. Sistem *smart stick* ini masih dapat dikembangkan sehingga pemanfaatannya akan sangat membantu dan mempermudah kegiatan orang-orang yang kurang dalam hal penglihatan.

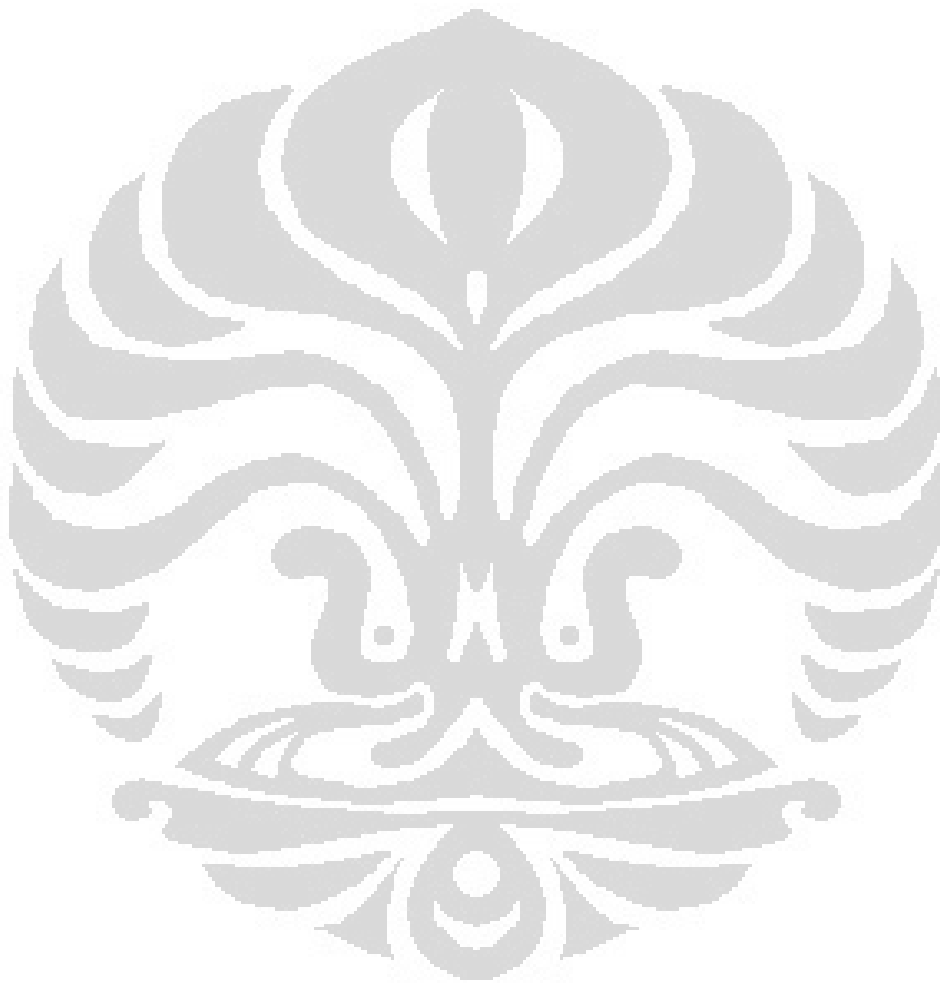
Beberapa hal yang dapat disimpulkan dari perancangan dan pembuatan *Smart Stick* Untuk Orang Buta Berbasis FPGA Xilinx SPARTAN-3E, adalah sebagai berikut.

1. Perancangan serta pembuatan *prototype Smart Stick* sebagai alat bantu berjalan untuk orang buta ini telah diuji dan membutuhkan pengembangan lebih lanjut.
2. Pemanfaatan teknologi yang terdapat pada FPGA akan mempermudah manusia dalam menjalankan aktifitas sehari-hari.

DAFTAR ACUAN

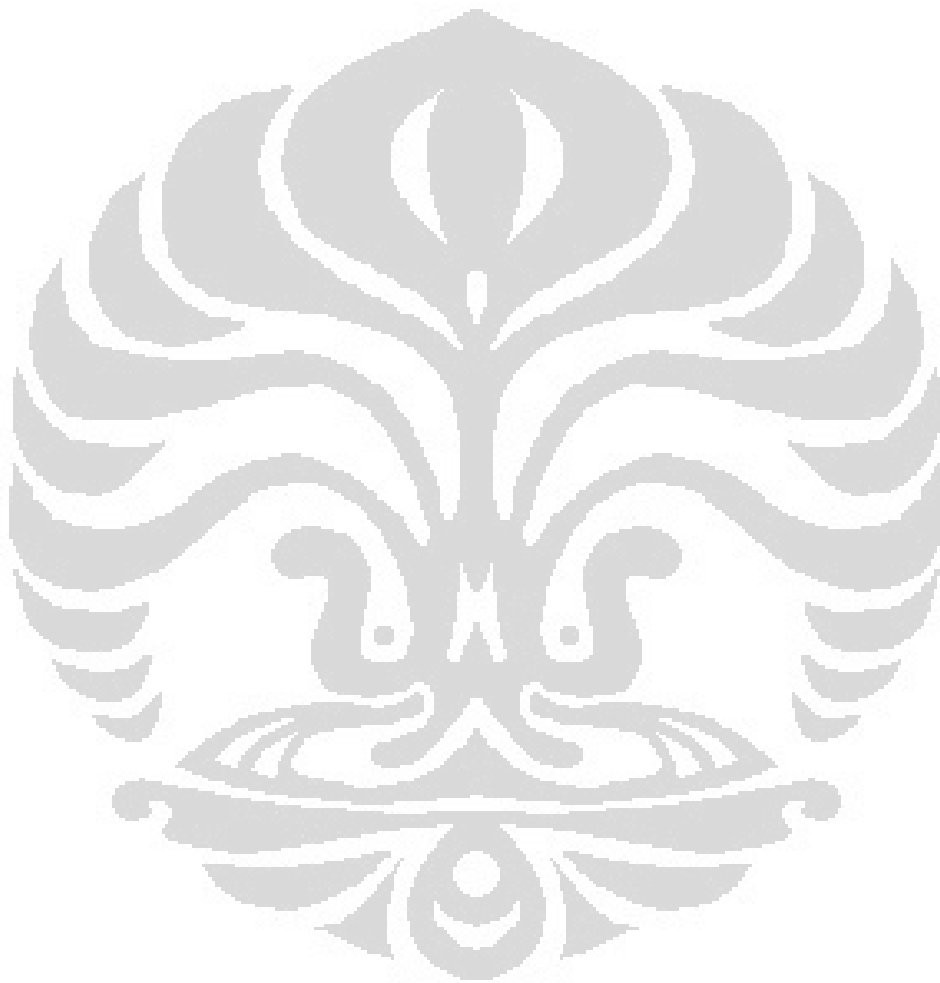
- [1] Nicolle, Jean P. (1997). "FPGA" December 20, 2011.
<http://www.fpga4fun.com/>
- [2] Author : "Xilinx". "*Xilinx Spartan-3E User Guide*". Access : December 20, 2011.
<http://www.xilinx.com/products/silicon-devices/fpga/index.htm>
- [3] Author : "Library Digital". "FPGA". Access December 20, 2011.
http://digilib.itelkom.ac.id/index.php?option=com_content&view=article&id=82:fpga-field-programmable-gate-array&catid=16:mikroprocessorkontroller&Itemid=14
- [4] Author : "undip". "*Praktikum FPGA*". Access : Januari 5, 2012.
http://didik.blog.undip.ac.id/files/2011/09/Modul1-Input_Output-v1.1.pdf
- [5] Author : "Digilent Inc". "*Digilent User Guide*". Access : Januari 5, 2012.
<http://www.digilentinc.com/Data/Documents/Tutorials/Xilinx%20ISE%20WebPACK%20VHDL%20Tutorial.pdf>
- [6] Utomo, Pranoto. (2007). "*FPGA*". Access : Januari 5, 2012
<http://pranotoutomo.wordpress.com/2008/08/03/vhdl-introduction/>
- [7] Mar, Idah. (2007). "Sensor Ultrasonik". Access : Januari 5, 2012.
<http://xsensor232.blogspot.com/2011/05/sensor-ultrasonik.html>
- [8] Author : "Teknik pengendali lcd karakter". Access : Januari 5, 2012
<http://depokinstruments.com/2010/02/08/teknik-pengendalian-lcd-karakter-i/>
- [9] Author : "ndoware". "*FPGA*". Access : Januari 5, 2012.
<http://ndoware.com/kategori/fpga/page/4>
- [10] Author : "hendrawan". "Ping))) Parallax Ultrasonic Range Finder". Access : Januari 5, 2012.
hendrawan.files.wordpress.com/2009/02/ping-paralax-application.pdf
- [11] Author : "mindsetonline.co.uk". "6V Electronic Buzzer". Access : Januari 5, 2012.
http://www.mindsetonline.co.uk/product_info.php?products_id=7784

- [12] Author : “Kiki Prawiroredjo dan Nyssa Asteria”. “Detektor Jarak Dengan Sensor Ultrasonik Berbasis Mikrokontroler”. Access : Januari 5, 2012.
blog.trisakti.ac.id/.../14.3-Kiki-Nyssa-Detektor-Jarak-hal-41-52.pdf



DAFTAR PUSTAKA

- [1] Peter J Ashenden, VHDL Tutorial. 2004.
- [2] Charles H. Roth, Jr, *Digital System Design Using VHDL*. 1998.



LAMPIRAN

Lampiran 1. Keseluruhan Program

```

`timescale 1ns / 1ps

module LCDtest(
    clk,
    lcd_rs, lcd_rw, lcd_e, lcd_4, lcd_5, lcd_6, lcd_7,
    leds, led01, led02, led03, buzzer, ping1, ping2, ping3);

    // input and outputs
    input          ping1;           // Switch
    input          ping2;           // Switch
    input          ping3;           // Switch
    input          clk;             // System Clock
    output         lcd_rs, lcd_rw, lcd_e, lcd_4, lcd_5,
lcd_6, lcd_7; // LCD pinout
    output reg     led01;           // LED Indicator
    output reg     led02;           // LED Indicator
    output reg     led03;           // LED Indicator
    output reg     buzzer;          // Buzzer Output
    output reg     [3:0] leds;      // LED Array

    wire           ping1,ping2,ping3;
    wire           lcd_rs, lcd_rw, lcd_e, lcd_4, lcd_5, lcd_6,
lcd_7;
    wire           [256:0] chars;
    reg            [23:0] state;
    reg            [23:0] delay;
    reg            [15:0] delay2;
    reg            [11:0] delay3;
    reg            [1:0] x;

    initial
        begin
            x <= 0;
            led01 = 1;
            led02 = 1;
            led03 = 1;

```

```

end

/* buzzer routine */
always @( posedge clk) begin
if ((ping1 == 1'b1) && (ping2 == 1'b0) && (ping3 == 1'b0))
begin
delay <= delay + 24'b1;
if (delay == 24'hFFFFFF)
begin
x = ~x;
buzzer = x;
end
end
end

if ((ping2 == 1'b1) && (ping1 == 2'b0) && (ping3 == 1'b0))
begin
delay2 <= delay2 + 16'b1;
if (delay2 == 16'hFFFF)
begin
x = ~x;
buzzer = x;
end
end
end

if ((ping3 == 1'b1) && (ping1 == 1'b0) && (ping2 == 1'b0))
begin
delay3 <= delay3 + 12'b1;
if (delay3 == 12'hFFF)
begin
x = ~x;
buzzer = x;
end
end
end

if ((ping1 == 1'b1) && (ping2 == 1'b1) && (ping3 == 1'b1))
begin
buzzer = 1'b0;
end
end

```

```
if ((ping1 == 1'b0) && (ping2 == 1'b0) && (ping3 == 1'b0))
    begin
        buzzer = 1'b1;
    end

end

/* led indikator routine */
always @( posedge clk) begin

    if (ping1 == 1'b1)
        begin
            led01 = 1;
        end
    else
        begin
            led01 = 0;
        end

    if (ping2 == 1'b1)
        begin
            led02 = 1;
        end
    else
        begin
            led02 = 0;
        end

    if (ping3 == 1'b1)
        begin
            led03 = 1;
        end
    else
        begin
            led03 = 0;
        end

end
```

```

always @( posedge clk) begin
begin
    state <= state + 24'b1;
    if(state == 24'hFFFFFF)
    begin
        leds <= leds + 1'b1;
    end
end
end

// LCD module
LCD lcd(    clk,
          chars,
          lcd_rs,
          lcd_rw,
          lcd_e,
          lcd_4,
          lcd_5,
          lcd_6,
          lcd_7);

// assign character data +
assign chars[255:24] = "Univ.Indonesia Distance : ";
// PING 1 indikator
assign chars[23:16] = {7'b0011000, ping1};
// PING 2 indikator
assign chars[15:8] = {7'b0011000, ping2};
// PING 3 indikator
assign chars[7:0] = {7'b0011000, ping3};

endmodule

```

Lampiran 2. Program LCD

```

module LCD(
    clk,
    chars,
    lcd_rs, lcd_rw, lcd_e, lcd_4, lcd_5, lcd_6, lcd_7);

    // inputs and outputs
    input      clk;
    input [256:0]  chars;
    output      lcd_rs,  lcd_rw,  lcd_e,  lcd_4,  lcd_5,
lcd_6, lcd_7;

    wire [256:0]  chars;
    reg          lcd_rs,  lcd_rw,  lcd_e,  lcd_4,  lcd_5,  lcd_6,
lcd_7;

// internal variables
    reg [5:0]  lcd_code;
    reg [1:0]  write = 2'b10;    // write code has 10 for rs rw

// delays
    reg [1:0]  before_delay = 3; // time before on
    reg [3:0]  on_delay = 13;    // time on
    reg [23:0] off_delay = 750_001; // time off

// states and counters
    reg [6:0]  Cs = 0;
    reg [19:0] count = 0;
    reg [1:0]  delay_state = 0;

// character data
    reg [256:0] chars_hold = " ";
    wire [3:0]  chars_data [63:0];    // array of characters

// redirects characters data to an array
    generate
    genvar i;
        for (i = 64; i > 0; i = i-1)

```



```

        begin : for_name
            assign chars_data[64-i] = chars_hold[i*4-1:i*4-4];
            end
        endgenerate

        always @ (posedge clk) begin

            // store character data
            if (Cs == 10 && count == 0) begin
                chars_hold <= chars;
            end

            // set time when enable is off
            if (Cs < 3) begin
                case (Cs)
                    0: off_delay <= 750_001; // 15ms
                    1: off_delay <= 250_001; // 5ms delay
                    2: off_delay <= 5_001; // 0.1ms delay
                endcase
            end else begin
                if (Cs > 12) begin
                    off_delay <= 2_001; // 40us delay
                end else begin
                    off_delay <= 250_001; // 5ms delay
                end
            end

            // delays during each state
            if (Cs < 80) begin
                case (delay_state)
                    0: begin
                        // enable is off
                        lcd_e <= 0;

                        {lcd_rs,lcd_rw,lcd_7,lcd_6,lcd_5,lcd_4} <= lcd_code;
                        if (count == off_delay) begin
                            count <= 0;
                            delay_state <= delay_state + 1;
                        end
                    end
                endcase
            end
        end

```

```

        end else begin
            count <= count + 1;
        end
    end
end
1: begin
    // data set before enable is on
    lcd_e <= 0;
    if (count == before_delay) begin
        count <= 0;
        delay_state <= delay_state + 1;
    end else begin
        count <= count + 1;
    end
end
2: begin
    // enable on
    lcd_e <= 1;
    if (count == on_delay) begin
        count <= 0;
        delay_state <= delay_state + 1;
    end else begin
        count <= count + 1;
    end
end
3: begin
    // enable off with data set
    lcd_e <= 0;
    if (count == before_delay) begin
        count <= 0;
        delay_state <= 0;
        Cs <= Cs + 1;           // next case
    end else begin
        count <= count + 1;
    end
end
endcase
end

// set lcd_code

```

```

        if (Cs < 12) begin
            // initialize LCD
            case (Cs)
                0: lcd_code <= 6'h03;           // power-on
initialization
                1: lcd_code <= 6'h03;
                2: lcd_code <= 6'h03;
                3: lcd_code <= 6'h02;
                4: lcd_code <= 6'h02;           // function
set
                5: lcd_code <= 6'h08;
                6: lcd_code <= 6'h00;           // entry mode
set
                7: lcd_code <= 6'h06;
                8: lcd_code <= 6'h00;           // display
on/off control
                9: lcd_code <= 6'h0C;
                10: lcd_code <= 6'h00;          // display
clear
                11: lcd_code <= 6'h01;
                default: lcd_code <= 6'h10;
            endcase
        end else begin

            // set character data to lcd_code
            if (Cs == 44) begin // change address at end
of first line
                lcd_code <= {2'b00, 4'b1100}; // 0100 0000
address change
            end else if (Cs == 45) begin
                lcd_code <= {2'b00, 4'b0000};
            end else begin
                if (Cs < 44) begin
                    lcd_code <= {write, chars_data[Cs-12]};
                end else begin
                    lcd_code <= {write, chars_data[Cs-14]};
                end
            end
        end
    end

```

```
end

// hold and loop back
if (Cs == 78) begin
    lcd_e <= 0;
    if (count == off_delay) begin
        Cs          <= 10;
        count       <= 0;
    end else begin
        count <= count + 1;
    end
end
end
end
endmodule
```

