



UNIVERSITAS INDONESIA

**PERANCANGAN DAN IMPLEMENTASI SISTEM
PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS
BERBASIS FPGA**

SKRIPSI

**HANA AFIFAH
0806459785**

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



UNIVERSITAS INDONESIA

**PERANCANGAN DAN IMPLEMENTASI SISTEM
PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS
BERBASIS FPGA**

SKRIPSI

Skripsi ini diajukan untuk melengkapi sebagian persyaratan menjadi Sarjana
Teknik

HANA AFIFAH

0806459785

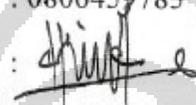
**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPARTEMEN TEKNIK ELEKTRO
DEPOK
JUNI 2012**

HALAMAN PERNYATAAN ORISINALITAS

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

Nama : Hana Afiifah

NPM : 0806459785

Tanda Tangan : 

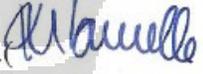
Tanggal : 03 Juli 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Hana Afifah
NPM : 0806459785
Program studi : Teknik komputer
Judul Skripsi : PERANCANGAN DAN IMPLEMENTASI
SISTEM GOLONGAN DAN TARIF TOL
OTOMATIS BERBASIS FPGA

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

Puji syukur penulis panjatkan kehadiran Allah SWT, karena atas segala rahmat dan hidayat-Nya penulis dapat menyelesaikan skripsi yang berjudul “Perancangan dan Implementasi Sistem Penentuan Golongan dan Tarif Tol Otomatis Berbasis FPGA”. Penulis menyadari bahwa skripsi ini tidak akan terselesaikan tanpa bantuan dari berbagai pihak. Mulai dari proses pembelajaran, analisis yang telah dijalani, dan proses penyusunan skripsi ini, penulis ingin mengucapkan terima kasih kepada:

1. Prof. Dr.-Ing. Ir. Kalamullah Ramli, M.Eng. selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Orang tua, suami dan keluarga saya yang telah memberikan bantuan dukungan materil, moral, dan doa;
3. Teman – teman dari Teknik Komputer angkatan 2008, yang tiada hentinya mendukung saya baik secara langsung maupun tidak langsung.
4. Teman-teman dari Teknik Komputer angkatan 2008, yang tiada hentinya mendukung saya baik secara langsung maupun tidak langsung.
5. Rekan asisten dari Laboratorium Digital Departemen Teknik Elektro.
6. Seluruh keluarga besar Civitas Akademika Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia yang tidak dapat saya sebutkan satu persatu.

Akhir kata, saya berharap semoga Allah SWT berkenan membalas kebaikan semua pihak yang telah membantu. Semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan.

Depok, 03 Juli 2012

Hana Afiifah

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

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

Nama : Hana Afiifah

NPM : 0806459785

Program studi : Teknik Komputer

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:

**PERANCANGAN DAN IMPLEMENTASI SISTEM PENENTUAN
GOLONGAN DAN TARIF TOL OTOMATIS 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,



(Hana Afiifah)

ABSTRAK

Nama : Hana Afiifah
Program Studi : Teknik Komputer
Judul : Perancangan dan Implementasi Sistem Golongan dan Tarif Tol Otomatis Berbasis FPGA

Dalam rangka meningkatkan pelayanan jalan tol di Indonesia, maka sistem pengumpulan tol atau aspek transaksi pembayaran di pintu gerbang tol perlu diupayakan untuk dipercepat. Salah satu pilihan untuk mempercepat tersebut dapat dilakukan dengan menerapkan sistem penentuan golongan dan tarif tol otomatis. Berkaitan dengan hal tersebut, maka dalam skripsi ini dilakukan penelitian dengan sebuah perancangan dan implementasi sistem penentuan golongan dan tarif tol otomatis berbasis FPGA. FPGA yang digunakan adalah Xilinx Spartan 3E, sedangkan peralatan lain yang digunakan sebagai *vehicle identification* berupa *limit switch* dan DI-MLDR. Sistem ini dilengkapi dengan *interface* melalui layar LCD. Pemodelan dan purwarupa sistem ini telah diuji dengan akurat dan siap untuk dirancang menjadi sebuah sistem yang utuh, dan siap untuk dikembangkan lebih lanjut.

Kata kunci: Jalan tol, FPGA Xilinx Spartan 3E, *vehicle identification*.

ABSTRACT

Name : Hana Afifah
Study Program : Computer Engineering
Title : Design and Implementation of FPGA Based Toll Road
Group and Tariff Classification System

To enhance the toll road service in Indonesia, the toll collection system need to be made faster. One of the choice for fastening could be done by system to define classes and toll rates automatically. Relating with this, then this final assignment is conducted by a research of class-determination and toll rates automatically system design and implementation based on FPGA. FPGA used is the Xilinx Spartan 3E, while other equipment used for vehicle identification in the form of *limit switch* and DI-MLDR. The system is equipped with an interface via LCD display. Modeling and prototype system has been tested with the results of accurate and ready to be designed into the whole system, and ready to be developed further.

Keyword: Toll Road, FPGA Xilinx Spartan 3E, vehicle identification.

DAFTAR ISI

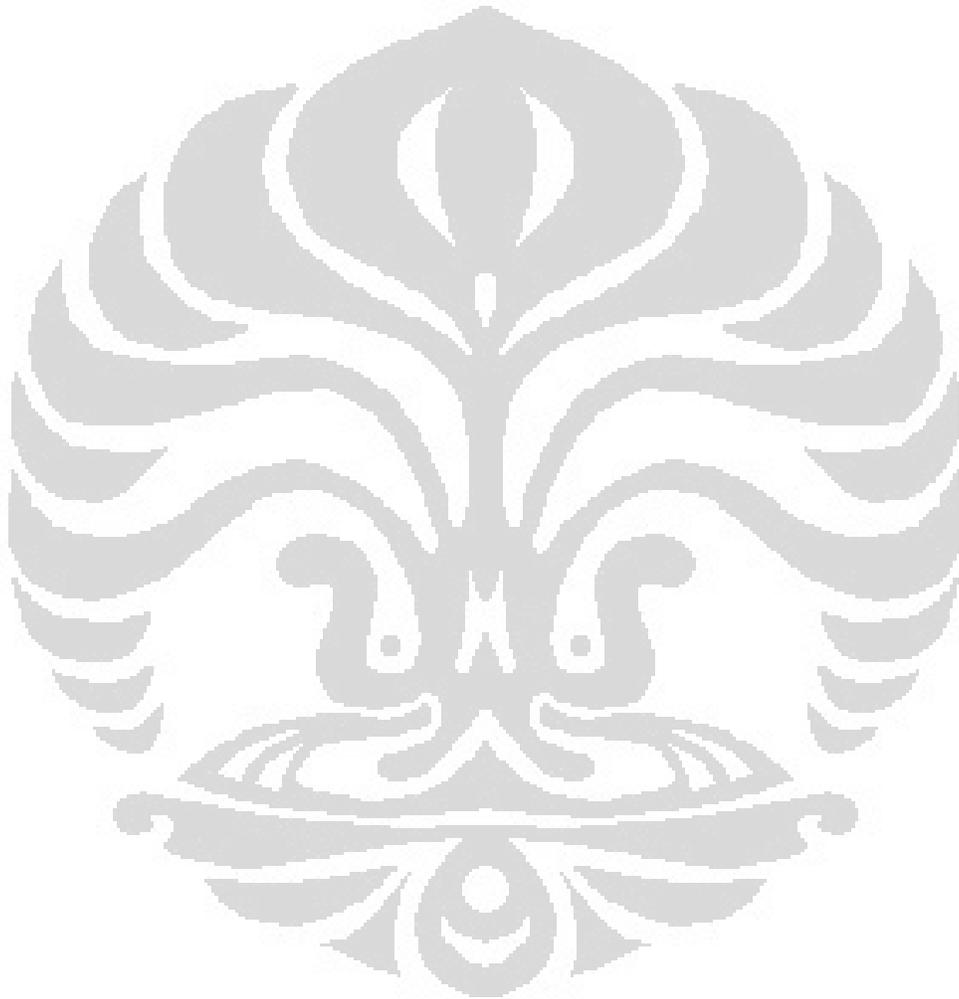
| | |
|---|------|
| HALAMAN JUDUL | ii |
| HALAMAN PERNYATAAN ORISINALITAS | vii |
| HALAMAN PENGESAHAN | viii |
| KATA PENGANTAR | ix |
| HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI | x |
| ABSTRAK | xi |
| ABSTRACT | xii |
| DAFTAR ISI | xiii |
| DAFTAR GAMBAR | xv |
| DAFTAR TABEL | xvii |
| BAB 1 PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Tujuan Penelitian | 2 |
| 1.3 Pembatasan Masalah | 2 |
| 1.4 Sistematika Penulisan | 3 |
| BAB 2 KONSEP JALAN TOL DAN FPGA XILINX SPARTAN 3E | 4 |
| 2.1 Jalan Tol | 4 |
| 2.1.1 Golongan dan Tarif | 4 |
| 2.1.2 Sistem Pembayaran | 6 |
| 2.2 FPGA | 6 |
| 2.2.1 Arsitektur FPGA | 7 |
| 2.2.2 VHDL | 9 |
| 2.3 FPGA SPARTAN 3E | 10 |
| 2.3.1 Komponen Board Starter Kit Spartan 3E | 11 |
| 2.3.1.1 <i>Push Button</i> | 12 |
| 2.3.1.2 <i>Slide Switches</i> | 12 |
| 2.3.1.3 LED | 13 |
| 2.3.1.4 <i>Clock Sources</i> | 14 |
| 2.3.1.5 Konektor Ekspansi | 15 |
| 2.3.1.6 SPI Serial Flash | 16 |
| 2.4 <i>Treadle Axle Counting</i> | 18 |
| 2.5 <i>Laser Rangefinder</i> | 18 |
| 2.6 <i>Double Wheel Detection Treadle</i> | 18 |
| 2.7 LCD | 19 |
| BAB 3 PERANCANGAN SISTEM PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS BERBASIS FPGA | 22 |
| 3.1 User Requirement | 22 |
| 3.2 Desain Sistem | 23 |
| 3.3 Alur Kerja Sistem | 27 |
| 3.4 Tahapan Implementasi | 29 |
| 3.4.1 Implementasi Perangkat Keras (<i>Hardware</i>) | 29 |

| | | |
|--|---|----|
| 3.4.1.1 | <i>Field Programmable Gate Array (FPGA)</i> | 29 |
| 3.4.1.2 | DI-MLDR | 30 |
| 3.4.1.3 | <i>Limit Switch</i> | 30 |
| 3.4.1.4 | LCD | 31 |
| 3.4.2 | Implementasi Perangkat Lunak (<i>Software</i>) | 31 |
| 3.5 | Rencana Uji Coba..... | 32 |
| BAB 4 IMPLEMENTASI SISTEM PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS BERBASIS FPGA | | 34 |
| 4.1 | Implementasi Sistem Penentuan Golongan dan Tarif Otomatis | 34 |
| 4.1.1 | Implementasi Perangkat Keras (<i>Hardware</i>)..... | 34 |
| 4.1.1.1 | FPGA..... | 34 |
| 4.1.1.2 | LCD | 35 |
| 4.1.1.3 | <i>Limit Switch</i> | 37 |
| 4.1.1.4 | DI-MLDR | 39 |
| 4.1.2 | Implementasi Perangkat Lunak (<i>Software</i>) | 41 |
| 4.2 | Uji Coba Sistem..... | 47 |
| 4.2.1 | Perangkat Keras (<i>Hardware</i>)..... | 47 |
| 4.2.2 | Analisis Hasil Percobaan..... | 57 |
| BAB 5 KESIMPULAN | | 59 |
| DAFTAR ACUAN | | 60 |
| DAFTAR PUSTAKA | | 61 |
| LAMPIRAN | | 62 |

DAFTAR GAMBAR

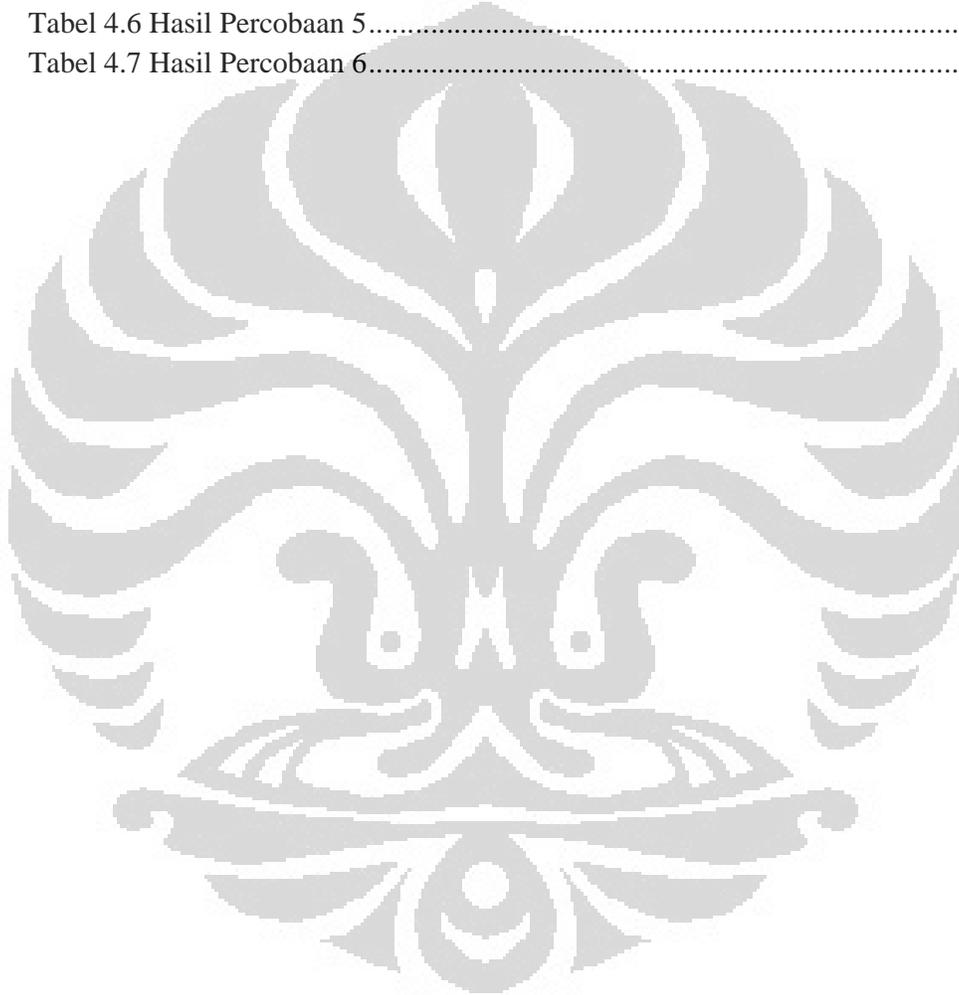
| | |
|---|----|
| Gambar 2.1 Klasifikasi Jenis Kendaraan [1]..... | 5 |
| Gambar 2.2 Golongan Kendaraan pada jalan tol [2] | 5 |
| Gambar 2.3 Arsitektur FPGA [3] | 7 |
| Gambar 2.4 SPARTAN 3E [4]..... | 10 |
| Gambar 2.5 Arsitektur FPGA SPARTAN 3E [4]..... | 11 |
| Gambar 2.6 Push Button dan 1 Rotary Switch..... | 12 |
| Gambar 2.7 UCF <i>Push Button</i> [4] | 12 |
| Gambar 2.8 Slide Switches [4]..... | 12 |
| Gambar 2.9 UCF <i>Slide Switches</i> [4] | 13 |
| Gambar 2.10 LED [4] | 13 |
| Gambar 2.11 UCF LED [4]..... | 13 |
| Gambar 2.12 Clock Source [4]..... | 14 |
| Gambar 2.13 UCF <i>Clock Source</i> [4]..... | 14 |
| Gambar 2.14 Konektor Ekspansi [4] | 15 |
| Gambar 2.15 UCF Konektor Ekspansi [4]..... | 16 |
| Gambar 2.16 Blok Diagram Untuk Komfigurasi SPI [4] | 17 |
| Gambar 2.17 Konfigurasi SPI serial Flash [4] | 17 |
| Gambar 2.18 (a) Treadle axle counting; (b) Double wheel detection treadle [5] . | 19 |
| Gambar 2.19 LCD 16 x 2 [6] | 20 |
| Gambar 2.20 Antarmuka LCD 16 x 2 [6] | 20 |
| Gambar 3.1 Use Case Diagram | 23 |
| Gambar 3.2 Desain Gerbang Tol..... | 24 |
| Gambar 3.3 Double Wheel Detection Treadle [5]..... | 25 |
| Gambar 3.4 Blok diagram perangkat utama..... | 26 |
| Gambar 3.5 Identifikasi Golongan Tol | 27 |
| Gambar 3.6 Diagram Alir Sistem | 28 |
| Gambar 3.7 Model VHDL dengan Arsitektur yang Berbeda..... | 32 |
| Gambar 4.1 Koneksi FPGA ke Header J1 | 35 |
| Gambar 4.2 Karakteristik Antarmuka LCD | 35 |
| Gambar 4.3 Set Karakter Kode ASCII | 37 |
| Gambar 4.4 (a) Tombol tekan (b) Tombol Fleksibel (c) Roller | 38 |
| Gambar 4.5 Layout Rangkaian Sensor DI-MLDR | 39 |
| Gambar 4.6 Sensor DI-MLDR aktif | 48 |
| Gambar 4.7 Kendaraan Golongan 1 dan Led Sebagai Indikator..... | 48 |
| Gambar 4.8 Led Sebagai Indikator 2 Limit Switch..... | 50 |
| Gambar 4.9 Kendaraan Golongan 2 dan Led Sebagai Indikator..... | 50 |
| Gambar 4.10 Kendaraan Golongan 3 dan Led Sebagai Indikator..... | 52 |
| Gambar 4.11 Kendaraan Golongan 4 dan Led Sebagai Indikator..... | 53 |

Gambar 4.12 Kendaraan Golongan 5 dan Led Sebagai Indikator 55
Gambar 4.13 Kendaraan Golongan 5 dan Led Sebagai Indikator 56



DAFTAR TABEL

| | |
|---|----|
| Tabel 4.1 Sinyal Antarmuka LCD..... | 36 |
| Tabel 4.2 Fungsi PIN Sensor DI-MLDR | 40 |
| Tabel 4.2 Hasil Percobaan 1..... | 49 |
| Tabel 4.3 Hasil Percobaan 2..... | 51 |
| Tabel 4.4 Hasil Percobaan 3..... | 52 |
| Tabel 4.5 Hasil Percobaan 4..... | 54 |
| Tabel 4.6 Hasil Percobaan 5..... | 55 |
| Tabel 4.7 Hasil Percobaan 6..... | 57 |



BAB 1

PENDAHULUAN

1.1 Latar Belakang

Jalan tol merupakan bagian dari jalan nasional yang perlu dijaga pengoperasiannya agar tetap dapat berfungsi secara optimal dalam mendukung pergerakan lalu lintas secara aman, nyaman dan efisien. Salah satu aspek pengoperasian itu adalah terkait dengan sistem pengumpulan tol yang dilakukan.

Sistem pengumpulan tol adalah suatu rangkaian kegiatan yang berkaitan dengan transaksi tol berupa proses pelayanan kepada pemakai jalan, kontrol atas pelaksanaan transaksi, proses pengadministrasian pendapatan tol serta proses lain yang mendukungnya. Pada prinsipnya, proses sistem pengumpulan tol harus didasarkan pada hal-hal berikut ini:

1. Memberi pelayanan yang cepat, tepat, aman dan nyaman pada pengguna tol;
2. Memberi jaminan kepada pengguna jalan dan Badan Usaha Jalan Tol (BUJT) bahwa transaksi sudah berjalan sesuai dengan tarif yang ditentukan;
3. Dapat diintegrasikan dengan sistem yang sudah ada maupun sistem yang akan dikembangkan;
4. Senantiasa memperhatikan pengembangan teknologi, manajemen sumber daya manusia, yang pada akhirnya memberikan pelayanan yang maksimal pada pengguna tol di satu sisi dan memberikan efisiensi di segala bidang bagi Badan Usaha Jalan Tol.

Dalam pelaksanaan penyelenggaraan transaksi di gerbang tol, BUJT harus memenuhi standar minimum yang ada pada Peraturan Menteri Pekerjaan Umum Nomor 392/PRT/M/2005 tentang Standar Pelayanan Minimal Jalan Tol. Dalam rangka meningkatkan pelayanan jalan tol di Indonesia, maka sistem pengumpulan tol atau aspek transaksi pembayaran di pintu gerbang tol perlu diupayakan untuk dipercepat. Salah satu pilihan untuk mempercepat tersebut dapat dilakukan dengan menerapkan sistem penentuan golongan dan tarif tol secara otomatis.

Beberapa keuntungan penerapan sistem penentuan golongan dan tarif tol otomatis ini adalah :

1. Mempercepat waktu transaksi dan meningkatkan kapasitas pelayanan
2. Meningkatkan tingkat akurasi transaksi dan menghindari kesalahan manusia
3. Meningkatkan efisiensi jumlah SDM untuk pelayanan di gerbang tol

Mengingat banyaknya operator jalan tol yang ada, maka penerapan sistem penentuan golongan dan tarif tol otomatis tersebut perlu diatur dan distandarisasi untuk menjamin kemudahan pengontrolan dan penggunaannya. Standarisasi sistem penentuan golongan dan tarif tol otomatis ini disusun dengan mengacu pada peraturan perundangan terkait dan dimaksudkan untuk menjamin berlangsungnya transaksi secara aman dan terpercaya.

Adapun tujuan dari standarisasi ini adalah untuk dijadikan pedoman pada penerapan sistem transaksi elektronik jalan tol di semua ruas jalan tol yang dioperasikan untuk memudahkan para operator dan pengguna jalan tol. Tulisan ini memberikan gambaran tentang penelitian rancangan sistem penentuan golongan dan tarif tol otomatis berbasis FPGA.

1.2 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan adalah sebagai berikut:

1. Perancangan sistem penentuan golongan dan tarif tol otomatis berbasis FPGA Xilinx Spartan 3E menggunakan *laser rangefinder*, *treadle axle counting*, *double wheel detection treadle* dan LCD.
2. Implementasi sistem penentuan golongan dan tarif tol otomatis berbasis FPGA Xilinx SPARTAN 3E menggunakan *limit switch*, DI-MLDR dan LCD.

1.3 Pembatasan Masalah

Pada penelitian ini dibatasi pada pembuatan purwarupa sistem penentuan golongan dan tarif tol otomatis berbasis FPGA dengan ketentuan sebagai berikut :

3. Sistem dengan asumsi identifikasi golongan dengan membedakan banyaknya gandar (as roda) dan adanya roda ganda pada roda belakang.
4. Sistem dengan FPGA sebagai controller dari semua perangkat.
5. Input sistem berupa nilai atau data dari limit switch dan DI-MLDR.

6. Output sistem berupa hasil pengolahan data yang ditampilkan pada LCD.
7. Antarmuka pengguna menggunakan bahasa pemrograman VHDL.

1.4 Sistematika Penulisan

Agar penulisan skripsi ini dapat terarah dengan baik, maka penulisan skripsi ini dibagi atas beberapa bab, yaitu:

BAB 1 PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, tujuan penulisan, pembatasan masalah, dan sistematika penulisan laporan.

BAB 2 KONSEP JALAN TOL DAN FPGA XILLINX SPARTAN 3E

Bab ini menjelaskan tentang definisi jalan tol, golongan dan tarif jalan tol, definisi FPGA, komponen FPGA Spartan 3E, laser rangefinder, tradle axle counting, double wheel detection tradle, dan LCD.

BAB 3 PERANCANGAN SISTEM PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS BERBASIS FPGA

Bab ini menjelaskan tentang *user requirement*, desain sistem yang meliputi tahapan perancangan, alur kerja sistem, rencana implementasi dan rencana uji coba.

BAB 4 IMPLEMENTASI SISTEM PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS BERBASIS FPGA

Bab ini menjelaskan tentang implementasi dari sistem dari segi perangkat keras maupun perangkat lunak dan analisa hasil implementasi.

BAB 5 KESIMPULAN

Bab ini memaparkan kesimpulan yang diambil dari hasil implementasi sistem penentuan golongan dan tarif jalan tol otomatis berbasis FPGA.

BAB 2

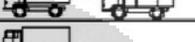
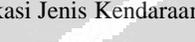
KONSEP JALAN TOL DAN FPGA XILINX SPARTAN 3E

2.1 Jalan Tol

Jalan tol adalah jalan umum yang merupakan bagian sistem jaringan jalan dan sebagai jalan nasional yang penggunanya diwajibkan membayar tol (UU RI No.38 Tahun 2004 tentang jalan). Dalam tingkatan jalan raya, jalan tol adalah satu-satunya fasilitas yang menyediakan arus bebas hambatan yang sempurna. Penyelenggaraan jalan tol dimaksudkan untuk mewujudkan pemerataan pembangunan dan hasil-hasilnya serta keseimbangan dalam pengembangan wilayah dengan memperhatikan keadilan, yang dapat dicapai dengan membina jaringan jalan yang dananya berasal dari pengguna jalan. Jalan tol juga bertujuan meningkatkan efisiensi pelayanan jasa distribusi guna menunjang peningkatan pertumbuhan ekonomi terutama di wilayah yang sudah tinggi tingkat perkembangannya (PP No.15 Tahun 2005 Bab 2 Pasal 2).

2.1.1 Golongan dan Tarif

Para pengguna jalan tol harus membayar sesuai tarif yang berlaku. Penetapan tarif didasarkan pada golongan kendaraan. Bangunan atau fasilitas di mana tol dikumpulkan dapat disebut gerbangtol. Pada dasarnya jenis kendaraan yang beroperasi di Indonesia dapat diklasifikasikan kedalam 12 golongan, termasuk sepeda motor dan kendaraan tidak bermotor, seperti diperlihatkan dalam Gambar 2.1, yaitu : Kendaraan ringan, Truk/Bus Sedang, Bus Besar, Truk Berat, Truk dan Trailer dengan berbagai konfigurasi sumbu, serta Sepeda Motor dan Kendaraan tidak bermotor .

| Golongan | Kelompok jenis kendaraan | Jenis kendaraan | Konfigurasi sumbu |
|----------|--|--|---|
| 1 | Sepeda motor, kendaraan roda-3 |  | |
| 2 | Sedan, jeep, station wagon |  |  |
| 3 | Angkutan penumpang sedang |  |  |
| 4 | Pick up, micro truk dan mobil hantaran |  |  |
| 5a | Bus kecil |  |  |
| 5b | Bus besar |  |  |
| 6a | Truk ringan 2 sumbu |  |  |
| 6b | Truk sedang 2 sumbu |  |  |
| 7a | Truk 3 sumbu |  |  |
| 7b | Truk gandengan |  |  |
| 7c | Truk semitrailer |  |  |
| 8 | Kendaraan tidak bermotor |  | |

Gambar 2.1 Klasifikasi Jenis Kendaraan [1]

Dalam penetapan tarif tol untuk kendaraan yang berbeda, dilakukan penggolongan kendaraan berdasarkan karakteristik kendaraan seperti terlihat pada Gambar 2.2 (kecuali sepeda motor dan kendaraan tidak bermotor dikeluarkan dari klasifikasi tersebut, karena kendaraan jenis tersebut tidak diperkenankan lewat di jalan tol).



Gambar 2.2 Golongan Kendaraan pada jalan tol [2]

2.1.2 Sistem Pembayaran

Terdapat dua sistem pengumpulan tol pada Jalan tol di Indonesia, yaitu:

1. Sistem terbuka, pengemudi kendaraan hanya satu kali melakukan transaksi di gerbang tol, pengemudi langsung membayar tol dan memperoleh tanda terima transaksi pada saat memasuki gerbang tol.
2. Sistem tertutup, pengemudi kendaraan diwajibkan berhenti dua kali untuk melakukan transaksi di gerbang tol. Di gerbang masuk pengemudi akan menerima tiket tanda masuk dari petugas, kemudian pada gerbang keluar akan melakukan pembayaran sesuai tarif tol yang berlaku.

2.2 FPGA

FPGA (Field Programmable Gate Array) adalah Integrated Circuit (IC) digital yang berisi sekumpulan blok logika yang saling terkoneksi dan dapat dikonfigurasi dan dapat dirancang sesuai dengan kebutuhan. Blok logika dapat diprogram untuk melakukan fungsi gerbang logika dasar (AND, OR, XOR, NOT) maupun jenis fungsi matematis sederhana dan kombinatorik yang lebih kompleks (decoder, adder, subtractor, multiplier, dan lain-lain).

Dalam beberapa FPGA, blok logika juga mencakup unsur-unsur memori (register) dari flip-flop yang sederhana sampai pada blok memori yang lebih kompleks (RAM). 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 di dalam lab atau lapangan (field). Oleh karena itu jajaran gerbang logika (Gate Array) ini disebut field-programmable. Jenis gerbang logika yang bisa diprogram meliputi semua gerbang dasar untuk memenuhi kebutuhan yang manapun.

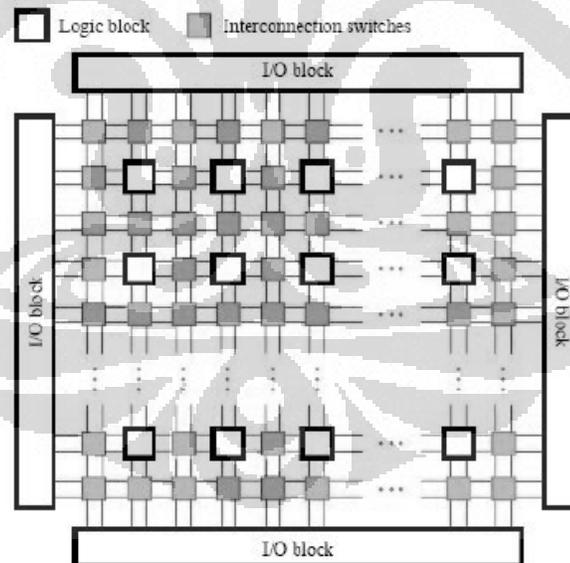
Secara umum FPGA akan lebih lambat jika dibandingkan dengan jenis chip yang lain seperti pada chip Application-Specific Integrated Circuit (ASIC). Hal ini karena FPGA menggunakan power (daya) yang besar dan bentuk desain yang kompleks. Beberapa kelebihan dari FPGA antara lain adalah harga yang murah, bisa diprogram mengikuti kebutuhan, dan kemampuan untuk di program kembali untuk mengoreksi adanya bugs. Jenis FPGA dengan harga murah biasanya tidak

bisa diprogram dan dimodifikasi setelah proses desain dibuat (fixed-version). Chip FPGA yang lebih kompleks dapat diperoleh dari jenis FPGA yang dikenal dengan CPLD (Complex-Programmable Logic Device).

FPGA merupakan IC tipe HDL (Hardware Description Language) yang dimana pengguna dapat mendesain hardware dalam IC FPGA sesuai kebutuhan. FPGA di program dengan menggunakan circuit diagram logic atau source code untuk menentukan bagaimana cara kerja chip FPGA tersebut. Setiap chip FPGA terdiri dari puluhan hingga puluhan ribu blok logika. Secara umum, setiap blok logika menggabungkan beberapa input biner menjadi satu atau dua keluaran sesuai dengan fungsi logika yang ditetapkan.

2.2.1 Arsitektur 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.



Gambar 2.3 Arsitektur FPGA [3]

Fungsi dari ketiga bagian tersebut adalah :

1. Configure Logic Blocks (CLB). 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 langsung.

Sebuah FPGA tersusun dari sebuah bagian yang bernama blok logika (*Logic Blocks*), yang kemudian pada perkembangannya saling terhubung satu sama lain. Kumpulan-kumpulan dari blok logika ini berjumlah ratusan bahkan ribuan dan membentuk suatu fungsi yang kompleks. Sebuah blok logika 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 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*) 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 blok logika tersebut dapat dihubungkan dengan blok logika lainnya melalui jalur atau koneksi yang ada. Setiap blok hanya mampu bekerja secara sederhana dan ringkas, Namun bila antara satu blok saling terhubung satu sama lain sebuah fungsi-fungsi logika yang kompleks pun dapat terbentuk.

2.2.2 VHDL

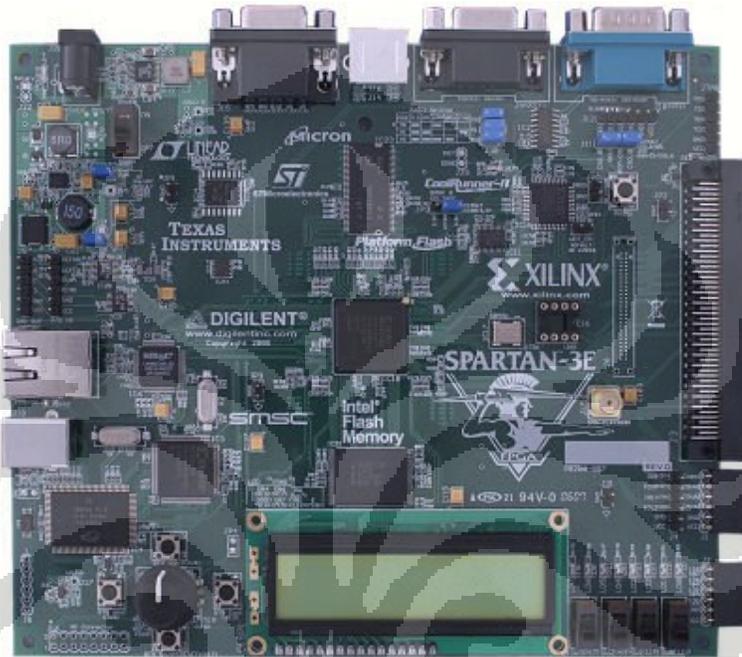
VHDL (*VHSIC Hardware Description Language*) adalah bahasa dari kelas bahasa komputer yang berfungsi untuk mendeskripsikan hardware elektronika digital. VHDL merupakan standar IEEE/ANSI dan Departemen Pertahanan US. Pertama kali dipublikasikan oleh IEEE pada tahun 1987, dengan label IEEE Std 1076-1987. Bahasa ini telah mengalami modifikasi dan revisi, dengan versi terbaru berlabel IEEE Std 1076-1993.

Desain digital yang akan digunakan dalam VHDL akan digambarkan dengan menggunakan *external view* dengan satu atau beberapa *internal view*. *External view* merupakan interface dari rancangan sedangkan *internal view* menyatakan fungsi atau struktur dari rancangan yang akan dibuat, dimana satu rancangan mungkin memiliki satu atau lebih *internal view*. Selain itu desain digital dapat juga digambarkan dalam VHDL dengan menggunakan beberapa *external view* yang berbeda, dimana tiap *external view* dan salah satu darinya berkaitan dengan *internal view*, yang secara bersama-sama merupakan representasi tertentu dari device, disebut dengan *entity design*. Entity juga bisa diartikan sebagai tempat untuk menentukan variabel input dan output.

VHDL memungkinkan seseorang untuk menggambarkan sistem digital di struktural atau tingkat perilaku. Tingkat perilaku dapat dibagi lagi menjadi dua jenis gaya: Data flow dan algoritmik. Data flow menggambarkan bagaimana data bergerak melalui sistem. Ini biasanya dilakukan dalam hal aliran data antara register (Register transfer level). Model aliran data penggunaan bersamaan membuat pernyataan yang dijalankan secara paralel data segera setelah tiba di masukan. Di sisi lain, pernyataan sekuensial dijalankan dalam urutan yang mereka ditetapkan. VHDL memungkinkan baik bersamaan dan sinyal berurutan tugas yang akan menentukan cara di mana mereka dieksekusi.

2.3 FPGA SPARTAN 3E

Spartan 3E merupakan salah satu dari keluarga FPGA yang di produksi oleh Xilinx yang bisa dilihat pada Gambar 2.4. Bahasa pemrograman yang digunakan untuk memprogram FPGA ini adalah VHDL (*Very high Hardware Description Language*) dengan bantuan *software development kit* dari Xilinx.

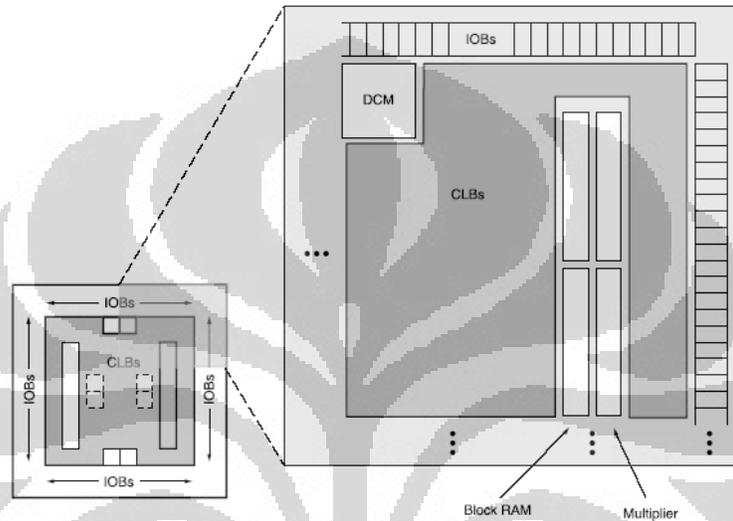


Gambar 2.4 SPARTAN 3E [4]

Secara umum arsitektur bagian dalam dari IC FPGA terdiri atas tiga bagian yaitu *Input Output Block (IOB)*, *Configurable Logic Block (CLB)*, dan Interkoneksi (kanal-kanal *routing*). Sedangkan arsitektur bagian dalam dari IC FPGA Spartan 3E dapat dilihat pada Gambar 2.5.

1. Configurable Logic Blocks (CLBs) memiliki Look-Up Tables (LUTs) yang fleksibel dimana dapat mengimplementasikan logika ditambah tempat penyimpanan yang digunakan sebagai flip-flop atau latches. CLBs melakukan berbagai fungsi logika dan juga menyimpan data.
2. Input/Output Blocks (IOBs) mengontrol aliran data antara I/O pin dan logika internal. Setiap IOB mendukung aliran data dua arah ditambah operasi 3-state. Selain itu IOB juga mendukung berbagai standar sinyal, termasuk empat high performance differential standards.

3. Block RAM menyediakan penyimpanan data dalam bentuk dual-port blok 18-Kbit.
4. Block Multiplier menerima dua angka biner 18-bit sebagai input dan menghitungnya.
5. Digital Clock Manager menyediakan kalibrasi, delay, multiplying, dividing dan fase pergeseran sinyal clock.



Gambar 2.5 Arsitektur FPGA SPARTAN 3E [4]

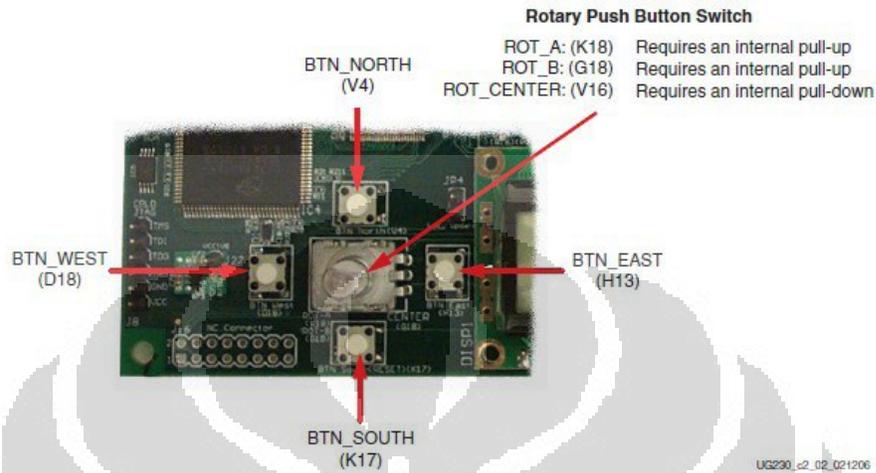
Semua bagian dari arsitektur ini terorganisir dengan baik seperti yang ditunjukkan pada Gambar 2.5. Sebuah cincin IOBs mengelilingi array CLBs. Setiap perangkat memiliki dua blok kolom RAM. Setiap kolom RAM terdiri dari beberapa blok 18-Kbit RAM. Setiap blok RAM dikaitkan dengan multiplier. DCMs diposisikan di tengah dengan dua di atas dan dua di bagian bawah perangkat.

2.3.1 Komponen Board Starter Kit Spartan 3E

Board yang digunakan dalam penelitian ini adalah Xilinx FPGA Spartan-3E (XC3S500E-4FG320C) dimana memiliki 232-pin I/O, 320-pin paket FPGA dan lebih dari 10.000 blok logika.

2.3.1.1 Push Button

Push button pada board FPGA Spartan 3E dengan 1 rotary switch dapat dilihat pada Gambar 2.6.



Gambar 2.6 Push Button dan 1 Rotary Switch

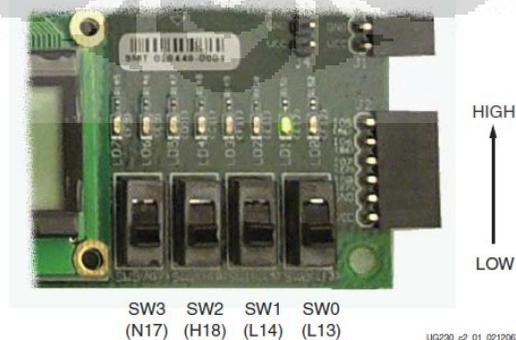
Konstrain push-button didefinisikan pada file UCF meliputi koneksi pin I/O, resistor pull-down, dan tegangan TTL low-voltage yang dapat dilihat pada Gambar 2.7.

```
NET "BTN_EAST" LOC = "H13" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "BTN_NORTH" LOC = "V4" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "BTN_SOUTH" LOC = "K17" | IOSTANDARD = LVTTTL | PULLDOWN ;
NET "BTN_WEST" LOC = "D18" | IOSTANDARD = LVTTTL | PULLDOWN ;
```

Gambar 2.7 UCF *Push Button*[4]

2.3.1.2 Slide Switches

Slide switches pada board FPGA Spartan 3E dapat dilihat pada Gambar 2.8.



Gambar 2.8 Slide Switches [4]

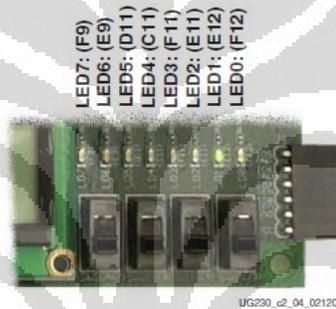
Konstrain *slide switches* didefinisikan pada file UCF meliputi koneksi pin I/O, *resistor pull-down*, dan tegangan TTL *low-voltage* yang dapat dilihat pada Gambar 2.9.

```
NET "SW<0>" LOC = "L13" | IOSTANDARD = LVTTTL | PULLUP ;
NET "SW<1>" LOC = "L14" | IOSTANDARD = LVTTTL | PULLUP ;
NET "SW<2>" LOC = "H18" | IOSTANDARD = LVTTTL | PULLUP ;
NET "SW<3>" LOC = "N17" | IOSTANDARD = LVTTTL | PULLUP ;
```

Gambar 2.9 UCF Slide Switches [4]

2.3.1.3 LED

LED pada board FPGA Spartan 3E dapat dilihat pada Gambar 2.10.



Gambar 2.10 LED [4]

Konstrain 6-pin accessory header didefinisikan di file UCF meliputi koneksi I/O, *slew-rate*, dan arus keluaran yang dapat dilihat pada Gambar 2.11.

```
NET "LED<7>" LOC = "F9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<6>" LOC = "E9" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<5>" LOC = "D11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<4>" LOC = "C11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<3>" LOC = "F11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<2>" LOC = "E11" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<1>" LOC = "E12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
NET "LED<0>" LOC = "F12" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 8 ;
```

Gambar 2.11 UCF LED [4]

2.3.1.4 Clock Sources

Spartan-3E Starter Kit board memiliki 3 sumber daya clock yang dapat dilihat pada Gambar 2.12, dimana setiap input clock terhubung dengan Digital Clock Manager.

- *50 MHz On-Board Oscillator*

Board ini memiliki 50 MHz osilator dengan 40% - 60% duty cycle. Osilator ini mampu bekerja secara akurat sampai dengan ± 2500 Hz atau ± 50 ppm.

- *Auxiliary Clock Oscillator Socket*

Soket 8-pin ini digunakan untuk osilator clock tambahan yang memiliki 8-pin DIP footprint. Soket ini digunakan apabila membutuhkan frekuensi lain dari 50 MHz.

- *SMA Clock Input or Output Connector*

Digunakan untuk memberikan clock dari sumber eksternal, dimana sinyal input clock dihubungkan ke konektor SMA.



Gambar 2.12 Clock Source [4]

Untuk menggunakan salah satu atau semua sumber daya clock, kita harus mengkonfigurasi lokasi konstrain UCF yang dapat dilihat pada Gambar 2.13.

```
NET "CLK_50MHZ" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
NET "CLK_SMA" LOC = "A10" | IOSTANDARD = LVCMOS33 ;
NET "CLK_AUX" LOC = "B8" | IOSTANDARD = LVCMOS33 ;
```

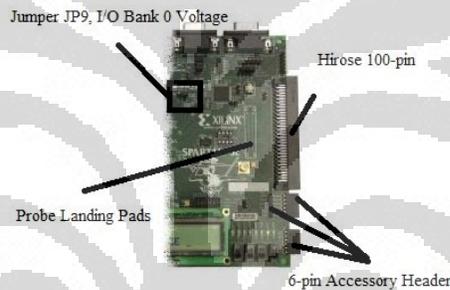
Gambar 2.13 UCF Clock Source [4]

2.3.1.5 Konektor Ekspansi

Spartan-3E Starter Kit board menyediakan berbagai macam konektor ekspansi untuk fleksibilitas *interface* yang mudah untuk komponen tambahan dari luar board, dapat dilihat pada Gambar 2.14.

Secara default konektor ekspansi ini memberikan supply daya sebesar 3.3V dimana dapat dirubah menjadi 2.5V dengan mengubah jumper yang ada di port JP9. Konektor Ekspansi yang ada pada board terdiri dari 3 jenis, yaitu:

1. Hirose 100-pin edge
2. 3 buah 6-pin peripheral modul connections
3. Probe landing pads



Gambar 2.14 Konektor Ekspansi [4]

Konstrain 6-pin *accessory header* didefinisikan di file UCF meliputi koneksi I/O, *slew-rate*, dan arus keluaran yang dapat dilihat pada Gambar 2.15.

```

# ===== 6-pin header J1 =====
#NET "J1<0>" LOC = "B4" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J1<1>" LOC = "A4" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J1<2>" LOC = "D5" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J1<3>" LOC = "C5" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;

# ===== 6-pin header J2 =====
#NET "J2<0>" LOC = "A6" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J2<1>" LOC = "B6" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J2<2>" LOC = "E7" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J2<3>" LOC = "F7" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;

# ===== 6-pin header J4 =====
#NET "J4<0>" LOC = "D7" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J4<1>" LOC = "C7" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J4<2>" LOC = "F8" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;
#NET "J4<3>" LOC = "E8" | IOSTANDARD = LVTTTL | SLEW = SLOW | DRIVE = 6 ;

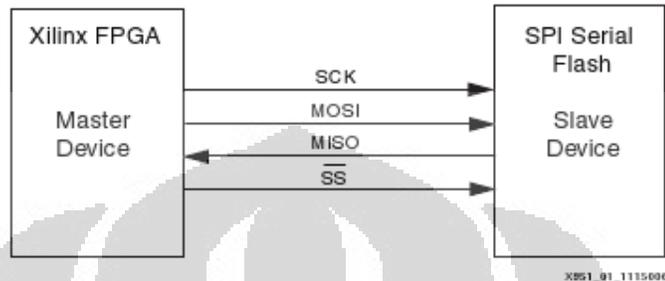
```

Gambar 2.15 UCF Konektor Ekspansi [4]

2.3.1.6 SPI Serial Flash

FPGA Xilinx adalah CMOS dikonfigurasi latch (CCL) yang berbasis dan harus dikonfigurasi pada power up. Secara tradisional, Xilinx FPGA konfigurasi dilakukan melalui Std IEEE 1149,1 (JTAG) *interface*, mikroprosesor, atau PROMs Xilinx (Flash Platform PROMs). Selain metode tradisional, untuk konfigurasi antarmuka secara langsung dapat menggunakan SPI serial flash. Konfigurasi antarmuka secara langsung untuk memori flash SPI serial di Spartan-3E FPGA memperluas solusi konfigurasi yang tersedia untuk pembuat Xilin. SPI serial flash memory populer karena mereka dapat dengan mudah diakses pasca-konfigurasi, menawarkan *random-access*, penyimpanan data non-volatile ke FPGA.

Sebuah sistem SPI biasanya terdiri dari perangkat master dan perangkat slave (Gambar 2.16). Bila menggunakan empat sinyal antarmuka untuk mengkonfigurasi FPGA Xilinx dari flash SPI serial, FPGA adalah perangkat master dan flash SPI serial adalah perangkat slave.

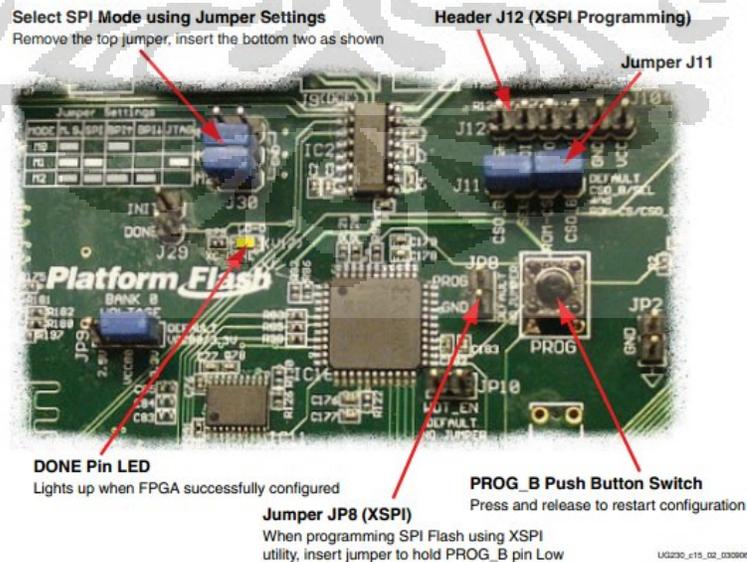


Gambar 2.16 Blok Diagram Untuk Komfigurasi SPI [4]

SPI serial flash juga digunakan untuk beberapa tujuan, seperti:

- Penyimpanan data non-volatile sederhana
- Penyimpanan untuk kode pengenalan, nomor seri, alamat IP, dll
- Penyimpanan kode prosesor MicroBlaze yang dapat dibayangi ke DDR SDRAM.

Untuk mengkonfigurasi FPGA dari SPI serial flash, *select pins* pada FPGA harus diatur secara tepat dan SPI serial flash harus berisi konfigurasi yang valid. Konfigurasi SPI serial flash dapat dilihat pada Gambar 2.17.



Gambar 2.17 Konfigurasi SPI serial Flash [4]

2.4 *Treadle Axle Counting*

Dalam pensinyalan kereta api, *treadle* adalah sebuah perangkat mekanis atau listrik yang mendeteksi bahwa *axle* (poros roda) kereta telah melewati suatu lokasi tertentu. Alat ini digunakan di mana lintasan rangkaian membutuhkan penguat dengan informasi tambahan tentang lokasi kereta, seperti di sekitar persimpangan otomatis, atau dalam rangkaian isyarat, bahwa suara peringatan kereta telah melewati titik yang tepat.

Treadle memiliki 3 tipe: mekanis, elektro-mekanis dan elektronik. Yang akan digunakan dalam penelitian ini adalah tipe elektronik dimana alat ini digunakan untuk menghitung as roda pada mobil. *Treadle axle counting* adalah sebuah sensor yang terdiri dari pipakare yang mengandung dua pelat baja, pelat ini dilas ke dua konduktor listrik, yang akan membawanya sinyal kepada controller. Ketika alat ini terinjak atau tertekan, maka sensor mengurangi impedansi, menghasilkan sinyal output.

2.5 *Laser Rangefinder*

Laser rangefinder (Pengukur jarak laser) adalah sebuah alat yang menggunakan pancaran sinar Laser (*Light Amplification by Stimulated Emission of Radiation*) untuk menentukan jarak ke sebuah benda/objek. Bentuk yang paling umum dari alat ini beroperasi dengan menggunakan prinsip menghitung waktu yang dibutuhkan oleh sinar laser untuk mulai memancar dari sumbernya, kemudian menemui sebuah objek (sehingga sinar laser tersebut memantul kembali), hingga kembali kepada pemancarnya.

Laser rangefinder mempunyai tingkat presisi yang tidak terlalu tinggi dikarenakan gelombang yang digunakan untuk pengukurannya adalah gelombang cahaya, yang bergerak sangat cepat. *Laser rangefinder* baik digunakan untuk melakukan pengukuran jarak yang membutuhkan respon yang cepat dan untuk arah pengukuran yang linear.

2.6 *Double Wheel Detection Treadle*

Alat pendeteksi roda ganda ini secara umum memiliki bahan yang sama dengan *treadle axle counting*. Perbedaannya adalah pada instalasinya, dimana

untuk mendeteksi roda ganda dibutuhkan 2 *treadle* (pedal) untuk mengidentifikasi roda ganda tersebut. Gambar perbedaan antara *treadle axle counting* dengan *double wheel detection treadle* dapat dilihat pada Gambar 2.16.



Gambar 2.18(a) Treadle axle counting; (b) Double wheel detection treadle [5]

2.7 LCD

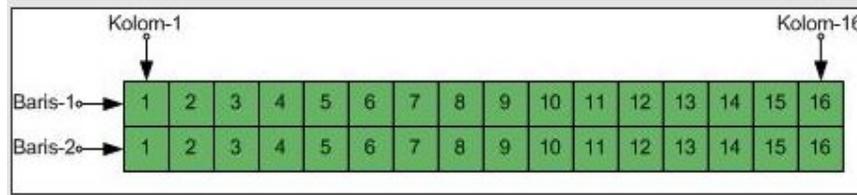
LCD adalah singkatan dari *Liquid Crystal Display*. Dilihat dari jenisnya, LCD memiliki 2 jenis, yaitu :

1. LCD Grafik

LCD Grafik adalah LCD yang tampilannya tidak terbatas, bahkan dapat menampilkan foto. LCD Grafik inilah yang terus berkembang seperti layar LCD yang biasa dilihat di notebook / laptop.

2. LCD Karakter

LCD ini tampilannya terbatas hanya pada tampilan karakter ASCII (seperti karakter-karakter yang tercetak pada keyboard komputer). Jenis LCD karakter biasa dituliskan dengan bilangan matriks dari jumlah karakter yang dapat dituliskan pada LCD tersebut, yaitu jumlah kolom karakter dikali jumlah baris karakter. Sebagai contoh, LCD 16x2, artinya terdapat 16 kolom dalam 2 baris ruang karakter, yang berarti total karakter yang dapat dituliskan adalah 32 karakter. Gambar LCD 16x2 dapat dilihat pada Gambar 2.19.



Gambar 2.19 LCD 16 x 2 [6]

LCD Karakter dalam pengendaliannya lebih mudah dibandingkan dengan LCD Grafik. Namun ada kesamaan diantara keduanya, yaitu inisialisasi. Inisialisasi adalah prosedur awal yang perlu dilakukan dan dikondisikan kepada LCD agar LCD dapat bekerja dengan baik. Hal yang sangat penting yang ditentukan dalam proses inisialisasi adalah jenis *interface* (antarmuka) antara LCD dengan *controller* (pengendali). Pada umumnya terdapat dua jenis antarmuka yang dapat digunakan dalam pengendalian LCD karakter:

1. 4 Bit, dan
2. 8 Bit

Untuk dapat mengendalikan LCD karakter dengan baik perlu koneksi yang benar dan koneksi yang benar dapat diwujudkan dengan cara mengetahui pin-pin antarmuka yang dimiliki oleh LCD karakter tersebut. LCD karakter yang akan digunakan adalah LCD 16x2 dengan antarmuka seperti Gambar 2.20.

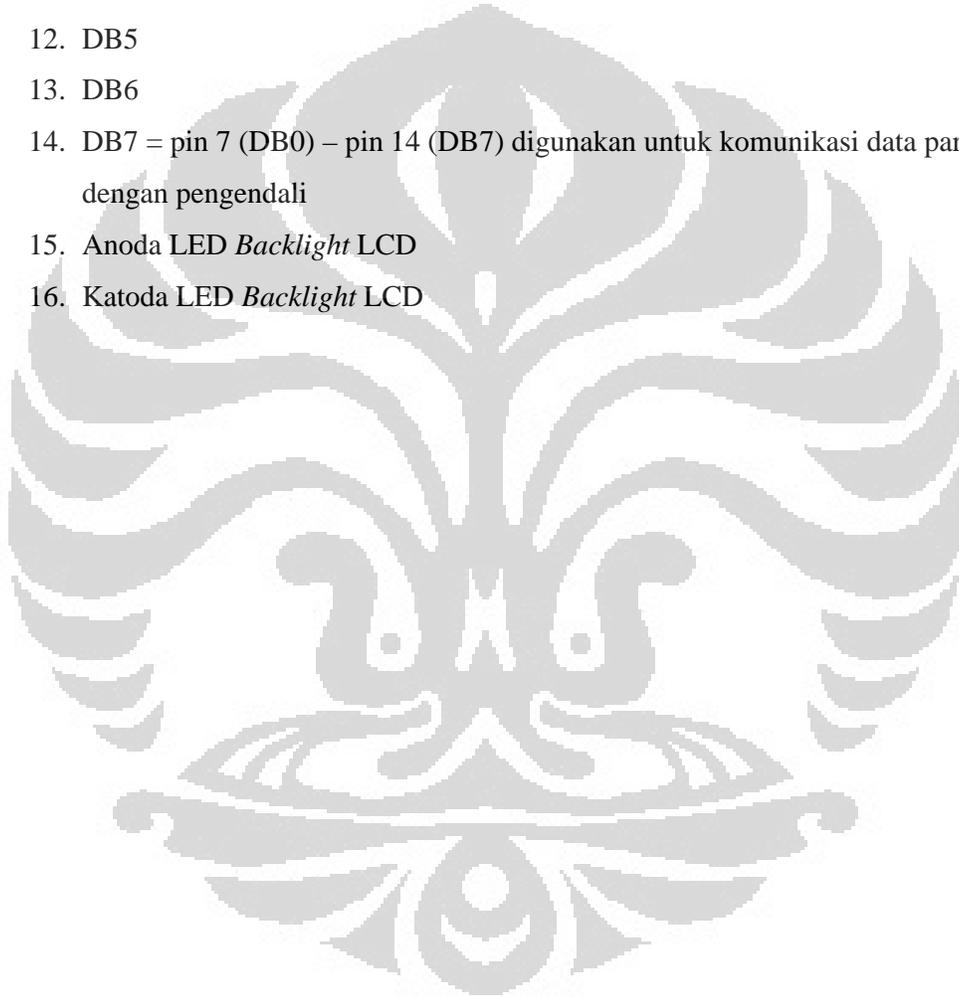


Gambar 2.20 Antarmuka LCD 16 x 2 [6]

Penjelasan antarmuka LCD pada Gambar 2.20 adalah sebagai berikut:

1. VSS = GND (ground)
2. VDD = Positif 5 Volt
3. Vo = Tegangan untuk mengatur kontras dari tampilan karakter
4. RS

5. R/W
6. E = pin 4 (RS) – pin 6 (E) digunakan untuk aktivasi LCD
7. DB0
8. DB1
9. DB2
10. DB3
11. DB4
12. DB5
13. DB6
14. DB7 = pin 7 (DB0) – pin 14 (DB7) digunakan untuk komunikasi data parallel dengan pengendali
15. Anoda LED *Backlight* LCD
16. Katoda LED *Backlight* LCD



BAB 3

PERANCANGAN SISTEM PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS BERBASIS FPGA

Bab ini menerangkan tahap perancangan sistem penentuan golongan dan tarif tol berbasis FPGA. Tahapan perencanaan ini meliputi desain dan implementasi. Dalam mendokumentasikan tahapan ini digunakan metode *Unified Modelling Language* (UML) yang merupakan diagram standar. Dengan UML, rancangan sistem dapat direpresentasikan ke dalam diagram-diagram yang memiliki fungsi masing-masing.

3.1 User Requirement

Menurut standar kamus IEEE, *user requirement(s)* adalah suatu kondisi atau kemampuan yang harus dipenuhi atau dimiliki oleh sistem atau komponen sistem untuk memenuhi kontrak, standar, spesifikasi atau dokumen formal lainnya yang disesuaikan dengan kebutuhan *user*.

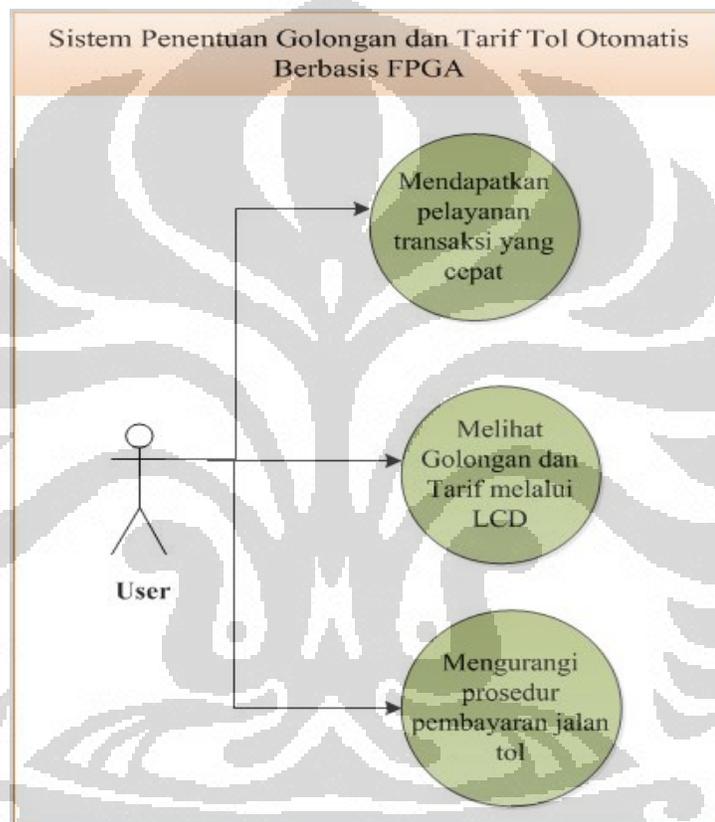
Secara umum, penentuan golongan dan tarif pada jalan tol masih dilakukan secara manual oleh manusia. Dimana pada pengamatan dilapangan masih ada kesalahan pada penentuan tarif dan golongannya. Selain itu kemacetan di jalan tol juga sering terjadi ketika memasuki gerbang tol dikarenakan waktu transaksi yang terlalu lama.

Untuk itu dibutuhkan sebuah sistem gerbang tol yang sesuai dengan *functional requirements* dari beberapa wawancara dengan *user* adalah sebagai berikut:

1. Mampu memberikan pelayanan transaksi yang cepat dan mengurangi kemacetan
2. Mampu mengidentifikasi golongan dan tarif lebih akurat untuk mengurangi kesalahan
3. Memiliki prosedur yang mudah dan tidak mengganggu kenyamanan pengguna jalan
4. Biaya implementasi tidak terlalu tinggi namun tetap dapat diandalkan

5. Dapat bertahan lama dan dapat diaplikasikan secara menyeluruh
6. Mempermudah pengguna jalan

Functional requirements di atas merupakan fungsi-fungsi dasar yang harus ada pada sebuah sistem untuk memenuhi kebutuhan *user* yang akan menggunakannya secara langsung. Untuk menggambarkan fungsional sistem dapat digunakan *use case diagram* yang dapat dilihat pada Gambar 3.1.

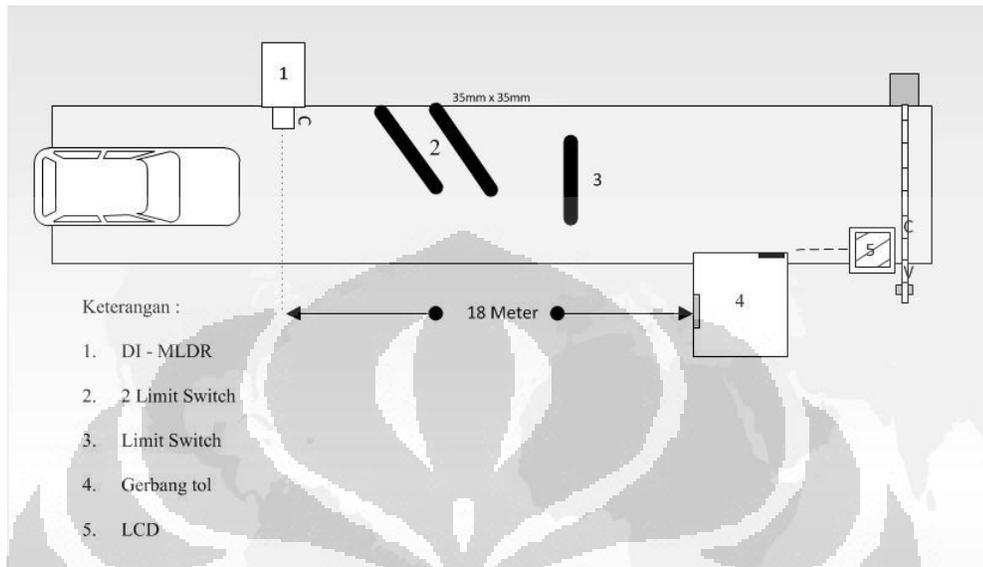


Gambar 3.1 Use Case Diagram

3.2 Desain Sistem

Jalan tol membutuhkan pelayanan yang baik serta tepat sasaran, termasuk pada sistem pengumpulan tol yang memerlukan kecepatan transaksi dan ketepatan pemilihan golongan dan tarif. Sistem pengumpulan tol adalah suatu rangkaian kegiatan yang berkaitan dengan transaksi tol berupa proses pelayanan kepada pemakai jalan, kontrol atas pelaksanaan transaksi, proses pengadministrasian pendapatan tol serta proses lain yang mendukungnya. Untuk itu dibutuhkan

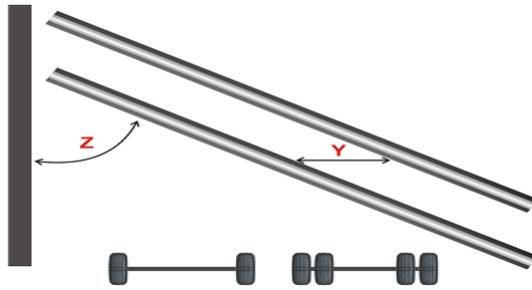
sebuah sistem yang dapat memenuhi kebutuhan tersebut. Asumsikan sebuah gerbang tol dengan ketentuan seperti Gambar 3.2.



Gambar 3.2 Desain Gerbang Tol

Perancangan sistem ini akan diimplementasikan pada sebuah gerbang tol otomatis yang dikendalikan oleh FPGA. Input FPGA didapat dari DI-MLDR dan laser pointer yang dipasang sejauh 18 m dari gerbang tol sesuai dengan panjang maksimum kendaraan [1]. Laser ini diharapkan dapat mendeteksi adanya kendaraan pada jarak minimum 50 cm.

Selain itu input juga didapat dari 2 *limit switch* yang dipasang tepat setelah sensor DI-MLDR. Sensor ini dipasang secara parallel sebanyak 2 buah, diatur satu sama lain pada jarak (Y), sehingga langkah gardan ban tunggal satu persatu pada sensor, tetapi beberapa langkah poros ban ganda kedua dapat melewati sensor secara bersamaan, sensor yang satunya lagi diletakkan dengan kemiringan sekitar 53° (Z) dari sumbu panjang jalan normal, peletakkan sensor seperti ini memungkinkan untuk mendeteksi roda tunggal atau ganda. Gambar peletakkan sensor ini dapat dilihat pada Gambar 3.3.

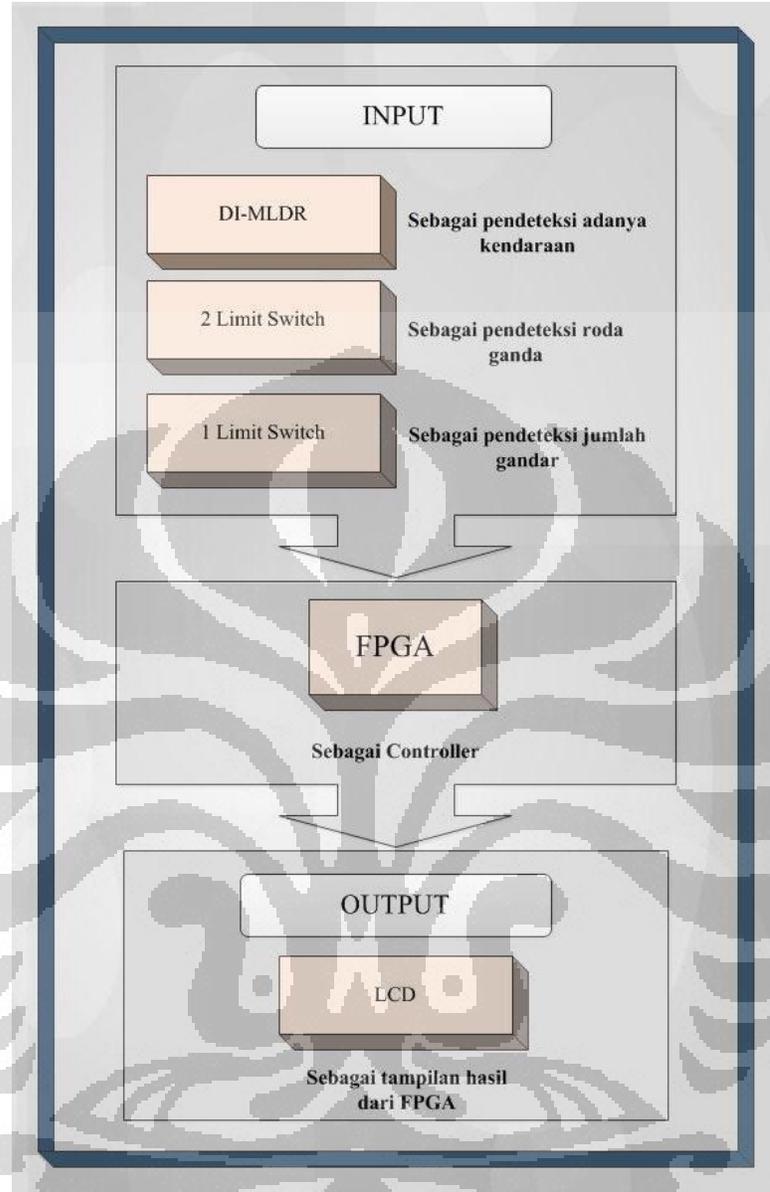


Gambar 3.3 Double Wheel Detection Treadle [5]

Jika pada saat roda tunggal, switch akan dilewati satu persatu sehingga memberi masukan logika '01' atau '10'. Sedangkan pada saat roda terdeteksi ganda, roda akan menekan kedua switch sehingga memberi masukan logika '11'.

Kemudian yang terakhir data diambil dari *limit switch*. *Limit switch* ini diletakkan setelah 2 *limit switch* sebelumnya. Switch ini akan menghitung banyaknya gandar kendaraan yang melintas ketika kendaraan terdeteksi memiliki roda ganda pada sensor sebelumnya. Jika roda terdeteksi tunggal, maka switch ini akan dilewati dan tidak menghitung banyaknya gandar. Dari hasil ini didapatkan jumlah gandar kendaraan yang melintas.

Data-data yang didapatkan dari input adalah berupa data digital antara 0 atau 1 yang bernilai benar atau salah. Selanjutnya data-data tersebut disimpan ke dalam FPGA dan diolah sehingga mendapatkan hasil yang dapat ditampilkan pada sebuah output berupa LCD. Blok diagram dari perangkat utama ini dapat dilihat pada Gambar 3.4.



Gambar 3.4 Blok diagram perangkat utama

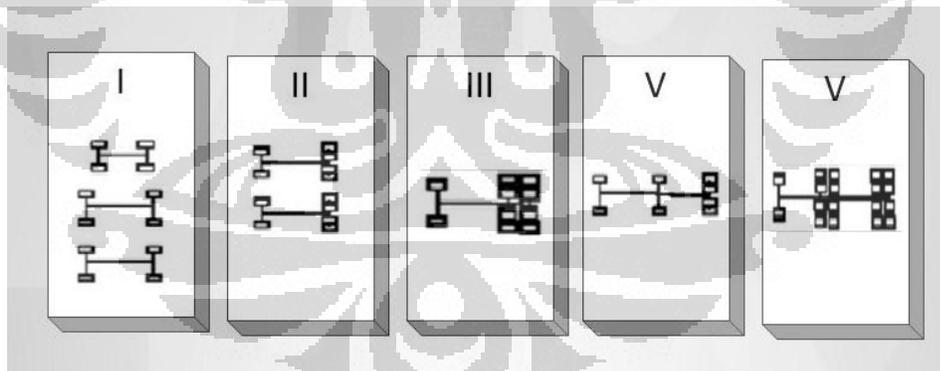
Pengguna jalan hanya perlu melewati gerbang tol seperti biasanya, tidak ada ketentuan khusus untuk menggunakan alat dari implementasi ini. Pengguna hanya tinggal membayar tarif yang sesuai dengan hasil yang ditampilkan pada LCD.

Biaya dari implementasi ini lebih banyak dibebankan kepada pengelola jalan tol bukan kepada pengguna jalan. Akan tetapi diperlukan sosialisasi secara menyeluruh kepada masyarakat guna mengetahui sistem penentuan golongan dan tarif otomatis ini.

3.3 Alur Kerja Sistem

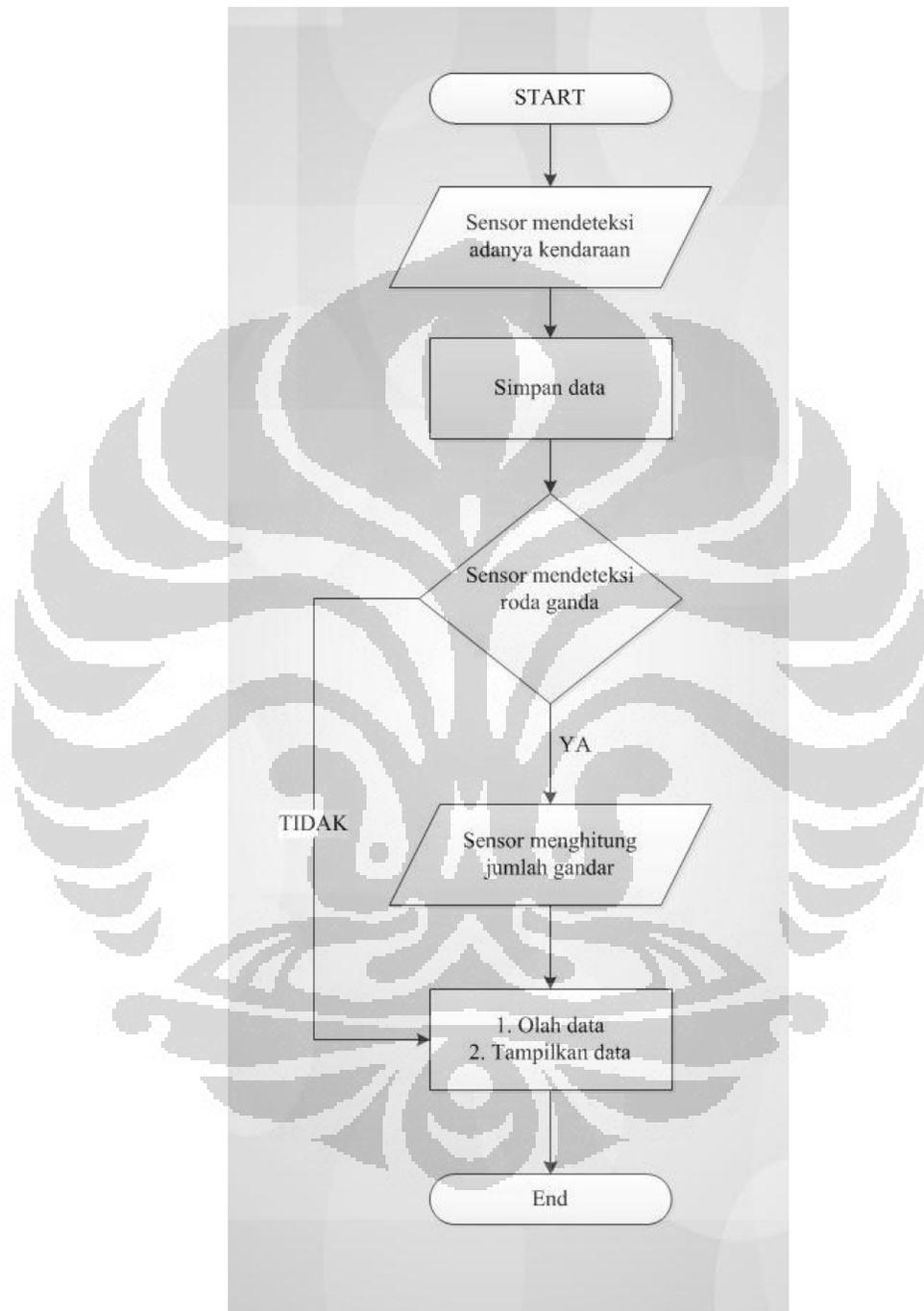
Sistem penentuan golongan dan tarif otomatis berbasis FPGA ini memiliki tahapan sebagai berikut :

1. Ketika mobil memasuki jarak 18 meter dari gerbang tol, sensor DI-MLDR aktif dan mendeteksi adanya kendaraan yang kemudian menyimpan data tersebut ke dalam FPGA.
2. Selanjutnya mobil melewati 2 sensor *limit switch* dengan posisi seperti pada gambar 3.3. Sensor ini digunakan untuk mendeteksi adanya roda ganda pada mobil yang kemudian data tersebut disimpan dalam FPGA
3. Sebelum melewati sensor yang selanjutnya, apabila ada mobil yang melewati sensor DI-MLDR maka data tersebut diolah menjadi data selanjutnya sesuai dengan antrian ketika masuk.
4. Setelah mobil melewati pendeteksian roda ganda, apabila tidak terdeteksi roda ganda, maka LCD langsung menampilkan golongan I.
5. Apabila mobil terdeteksi roda ganda, maka selanjutnya mobil melakukan penghitungan banyaknya gandar. Hasil perhitungan tersebut mempengaruhi identifikasi golongan yang tampil pada LCD. Penggolongan ini dapat dilihat pada Gambar 3.5.



Gambar 3.5 Identifikasi Golongan Tol

Alur kerja sistem ini dapat direpresentasikan dalam sebuah diagram alir (*flowchart*) yang dapat dilihat pada Gambar 3.6.



Gambar 3.6 Diagram Alir Sistem

3.4 Tahapan Implementasi

3.4.1 Implementasi Perangkat Keras (*Hardware*)

Perangkat keras yang digunakan dalam sistem ini adalah FPGA, DI-MLDR, *Limit switch* dan LCD.

3.4.1.1 *Field Programmable Gate Array* (FPGA)

Perangkat keras pertama yang sangat penting adalah FPGA. FPGA ini berlaku sebagai *controller* dari perangkat keras lainnya. Ada banyak macam FPGA dengan produsen yang berbeda-beda. Beberapa langkah untuk melakukan implementasi sistem tertanam menggunakan FPGA adalah sebagai berikut:

- Langkah 1 : Menentukan desain spesifikasi atau persyaratan yang dibutuhkan
- Langkah 2 : Membuat sebuah blok diagram sistem yang didasarkan pada langkah pertama
- Langkah 3 : Memilih perangkat yang sesuai dengan sistem yang dibuat pada langkah kedua

Sebelum lebih jauh menggunakan FPGA, terdapat beberapa hal yang harus diperhatikan diantaranya :

1. Penentuan biaya.
Hal ini penting mengingat harga sebuah FPGA sangat mahal. Perlu diteliti kembali apakah sistem yang ingin dibuat membutuhkan sebuah FPGA atau dapat dikendalikan hanya dengan mikrokontroler.
2. Penentuan sistem.
Hal ini juga penting untuk diperhatikan karena jenis-jenis FPGA yang diproduksi adalah berdasarkan kebutuhan sistem.

FPGA yang digunakan dalam sistem yang akan dibuat adalah FPGA SPARTAN-3E dimana FPGA jenis ini sesuai untuk mengendalikan sistem yang akan dibuat.

3.4.1.2 DI-MLDR

DI-MLDR yang digunakan pada implementasi sistem ini harus memiliki beberapa kriteria sebagai berikut:

1. Rendah biaya.

Kriteria ini penting karena untuk sistem tertanam dibutuhkan semua perangkat yang rendah biaya untuk pemasangan dalam jumlah banyak.

2. Rendah daya.

Kriteria rendah daya juga dibutuhkan untuk sebuah sistem tertanam, karena untuk pemasangan dalam jumlah banyak dan sesuai dengan karakter daya dari pengendalinya (dalam sistem ini FPGA).

3. Mudah diperoleh.

Tentunya agar implementasi sistem ini dapat dilaksanakan, maka alat pendukungnya pun harus mudah diperoleh.

4. Jarak Pengukuran.

Untuk sistem ini, dibutuhkan jarak ukur yang bisa dibaca oleh DI-MLDR minimum 50 cm, sebagai kompensasi antara posisi diletakkannya DI-MLDR tersebut dengan mobil yang melintasinya.

Dengan kriteria seperti yang disebutkan di atas, maka sensor yang digunakan adalah sensor DI-MLDR dengan tambahan *laser-pointer*.

3.4.1.3 *Limit Switch*

Untuk menghitung banyaknya gandar dan mendeteksi adanya roda ganda dibutuhkan bahan yang sama dengan kriteria sebagai berikut:

1. Waktu identifikasi yang cepat.

Sistem ini sangat membutuhkan waktu identifikasi yang cepat untuk mengurangi kemacetan dan antrian yang panjang. Selain itu kendaraan yang melaju juga mungkin bisa dengan kecepatan tinggi sehingga dibutuhkan waktu deteksi yang cepat dan akurat.

2. Kokoh dan handal

Kriteria ini juga sangat dibutuhkan dalam sistem ini dimana kendaraan yang melintas bukan hanya kendaraan yang berukuran kecil dan ringan, akan tetapi

kendaraan yang melintas juga kendaraan yang berukuran besar dan memiliki beban berat. Oleh karena itu dibutuhkan alat yang kokoh dan handal agar tidak mudah rusak.

3. Mudah digunakan dan diperoleh

Alat yang akan digunakan sebisa mungkin adalah alat-alat yang mudah digunakan sehingga dapat dipasang tanpa keahlian yang khusus. Untuk implementasi dalam jumlah besar juga dibutuhkan alat yang mudah diperoleh sehingga tidak memerlukan biaya lebih untuk mendapatkan alat ini.

3.4.1.4 LCD

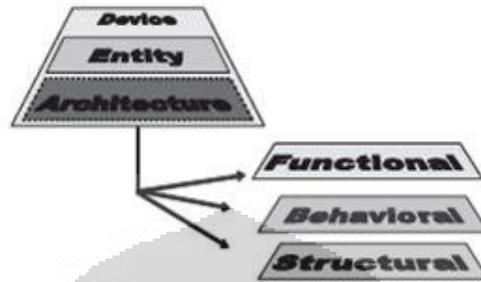
Dalam menentukan sebuah alat untuk output (keluaran) didasarkan pada kebutuhan sistem untuk menampilkan hasil data. Dikarenakan jenis LCD hanya ada 2, maka ini memudahkan untuk menentukan LCD apa yang akan dipakai. Sistem ini hanya membutuhkan keluaran berupa golongan dan tarif tol, sehingga hanya diperlukan sebuah LCD karakter dengan ukuran 16x2.

3.4.2 Implementasi Perangkat Lunak (*Software*)

Tahapan selanjutnya adalah implementasi perangkat lunak. Perangkat lunak yang diimplementasikan yaitu implementasi perangkat lunak pada pengendali. Pada implementasi ini, perangkat lunak yang diimplementasikan berupa sistem antar-muka sebagai jembatan komunikasi antara FPGA dengan sensor-sensor yang digunakan. Oleh karena itu, perangkat lunak ini harus berupa program yang dapat melakukan komunikasi secara langsung dengan FPGA.

Dalam hal ini dipilih program berbasis VHDL dikarenakan VHDL memungkinkan perilaku sistem perlu dijelaskan (model) dan diverifikasi (simulasi) sebelum alat sintesis menterjemahkan desain ke dalam perangkat keras yang nyata (gerbang dan kabel). Selain itu, VHDL memungkinkan deskripsi sistem konkuren (banyak bagian, masing-masing dengan perilakunya sendiri, bekerja bersama-sama pada waktu yang sama). VHDL adalah bahasa dataflow, tidak seperti bahasa komputasi prosedural seperti BASIC, C, dan kode assembly, yang semuanya berjalan secara berurutan, satu instruksi pada satu waktu.

VHDL juga memiliki kemampuan untuk menggunakan gabungan level dari model yang memiliki arsitektur yang berbeda seperti ditunjukkan sebagai berikut :



Gambar 3.7 Model VHDL dengan Arsitektur yang Berbeda

Hal tersebut memang bukanlah keunikan atau ciri khas VHDL. Namun, pada kenyataannya kode Verilog juga memiliki konsep sama walaupun hanya terdapat dalam sebuah “module”. Meskipun demikian, keunggulan itu secara eksplisit didefinisikan dalam VHDL dan secara praktis digunakan bersama oleh rancangan *multi-level* dalam VHDL.

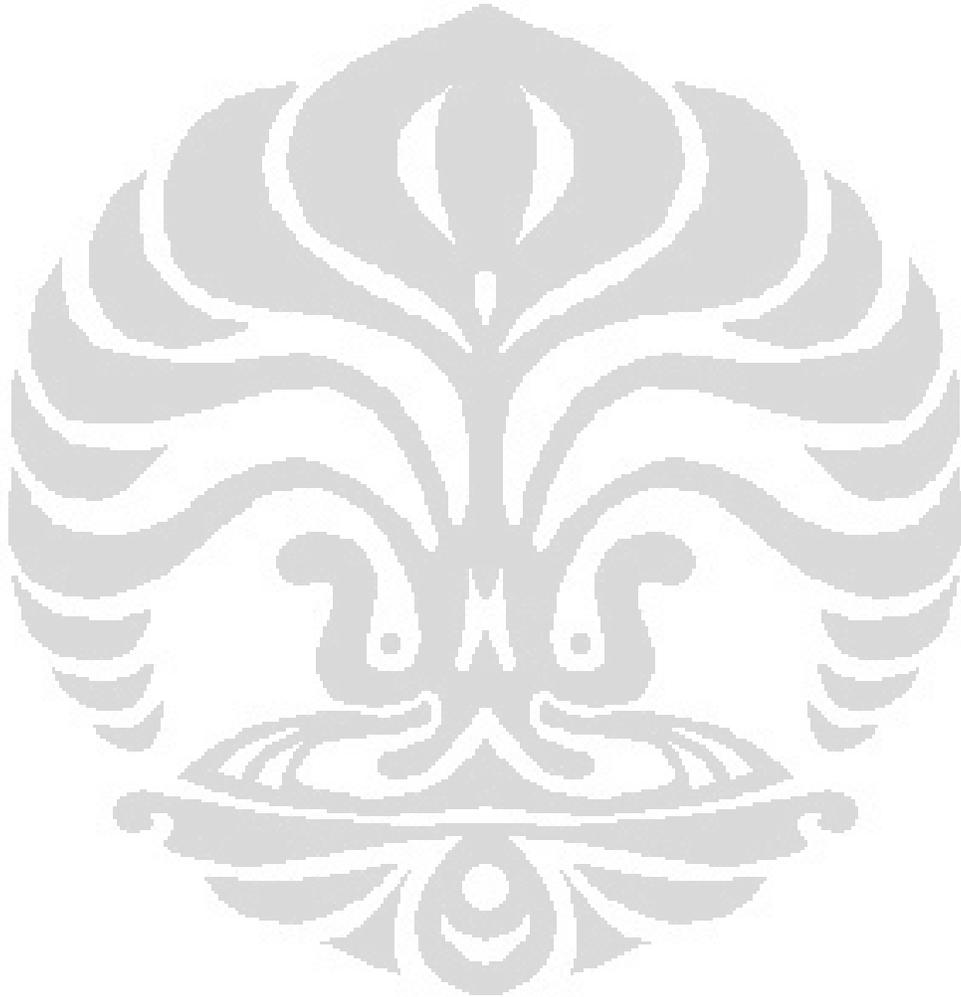
Pembagian sebuah model ke dalam beberapa bagian juga merupakan keunggulan lain dari VHDL. Misalnya, bagian *interface* (dalam VHDL dikenal sebagai *entity*) dan bagian kelakuan atau *behavior* (dalam VHDL dikenal sebagai *architecture*). Bagian-bagian tersebut adalah pendekatan praktis yang luar biasa untuk bentuk model yang memiliki *multiple behaviour* dalam sebuah *interface*. Selain itu bentuk model tersebut memungkinkan terjadi pertukaran dan implementasi *multiple* secara terus menerus.

3.5 Rencana Uji Coba

Rencana pengujian yang pertama adalah melakukan uji coba DI-MLDR. Parameter-parameter yang diuji-cobakan adalah jarak pendeteksian sensor dan ketepatan deteksi. Hasil uji coba ini akan menentukan apakah sensor yang digunakan layak untuk digunakan pada sistem atau tidak.

Kemudian uji coba yang kedua adalah uji coba terhadap *limit switch*. Parameter-parameter yang akan diuji-cobakan adalah ketepatan pendeteksian roda ganda maupun penghitungan jumlah gandar.

Pengujian yang terakhir adalah pada output (keluaran) yaitu LCD. Dimana parameter yang akan diuji-cobakan adalah ketepatan hasil pengolahan data dan koneksi antara LCD dengan FPGA.



BAB 4

IMPLEMENTASI SISTEM PENENTUAN GOLONGAN DAN TARIF TOL OTOMATIS BERBASIS FPGA

4.1 Implementasi Sistem Penentuan Golongan dan Tarif Otomatis

Implementasi sistem penentuan golongan dan tarif ini terbagi menjadi 2 (dua) bagian utama. Pertama adalah implementasi alat (perangkat keras), dan yang kedua adalah implementasi perangkat lunak yang berupa program antar-muka.

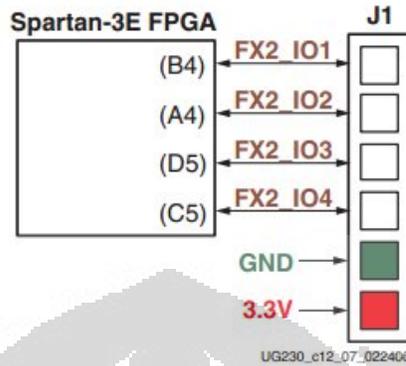
4.1.1 Implementasi Perangkat Keras (*Hardware*)

Karena hasil dari implementasi ini masih berupa purwarupa dari sistem yang akan dibuat, maka ada beberapa perangkat keras yang berbeda dari rancangan sistem yang sudah ada. Perangkat keras yang berbeda antara lain *treadle axle counting* dan *double wheel detection treadle* diganti dengan menggunakan sensor *limit switch*. Selain itu, *laser rangefinder* juga diganti dengan DI-MLDR yang mempunyai fungsi yang hampir sama.

Design implementasi perangkat keras masih sama dengan design sebelumnya hanya dibuat miniatur dari jarak yang seharusnya. Ldr dalam posisi terdepan dari jalan yang dilanjutkan sensor *limit switch* untuk mendeteksi adanya roda ganda. Kemudian sensor *limit switch* untuk menghitung banyaknya gandar mobil dan hasilnya ditampilkan pada layar LCD.

4.1.1.1 FPGA

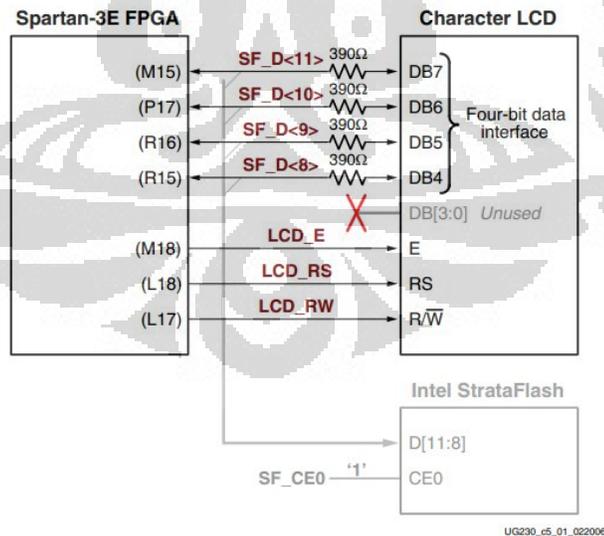
FPGA yang digunakan pada implementasi sistem ini adalah SPARTAN 3E. Port yang digunakan dalam implementasi ini adalah 3 soket konektor ekspansi. Sumber tegangan yang disediakan dalam FPGA adalah 3,3V. Dikarenakan setiap sensor yang digunakan dalam implementasi ini membutuhkan sumber tegangan 5V, maka sumber didapat dari luar FPGA. 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 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 | 1 | 1 | 1 | 1 | 1 | 1 | |
| Lower Data Nibble | xxxx0000 | | | | | 0 | 0 | P | \ | P | - | 9 | 3 | α | ρ | | | |
| | xxxx0001 | ! | 1 | A | Q | a | q | . | . | 7 | 7 | 4 | ä | q | | | | |
| | xxxx0010 | " | 2 | B | R | b | r | 「 | 」 | イ | ツ | × | β | θ | | | | |
| | xxxx0011 | CG RAM | # | 3 | C | S | c | s | 」 | ウ | 〒 | ε | ε | ω | | | | |
| | xxxx0100 | CG RAM | \$ | 4 | D | T | d | t | 、 | エ | ト | ト | μ | Ω | | | | |
| | xxxx0101 | | % | 5 | E | U | e | u | . | . | オ | ナ | 1 | ö | ü | | | |
| | xxxx0110 | | & | 6 | F | V | f | v | ヲ | カ | ニ | ヨ | ρ | Σ | | | | |
| | xxxx0111 | | ' | 7 | G | W | g | w | 7 | キ | ヲ | ラ | q | π | | | | |
| | xxxx1000 | | (| 8 | H | X | h | x | イ | ウ | ネ | リ | 」 | 又 | | | | |
| | xxxx1001 | |) | 9 | I | Y | i | y | ウ | ケ | 」 | ル | 」 | 4 | | | | |
| | xxxx1010 | | * | : | J | Z | j | z | エ | コ | ハ | レ | i | 〒 | | | | |
| | xxxx1011 | | + | ; | K | [| k | [| オ | サ | ヒ | ロ | * | 〒 | | | | |
| | xxxx1100 | | , | < | L | ¥ | l | | ヤ | シ | フ | フ | φ | 〒 | | | | |
| | xxxx1101 | | - | = | M |] | m |] | ユ | ズ | ヘ | ン | も | ÷ | | | | |
| | xxxx1110 | | . | > | N | ^ | n | → | ヨ | セ | ホ | ° | 〒 | | | | | |
| | xxxx1111 | | / | ? | O | _ | o | ← | ッ | ツ | マ | ° | ö | ■ | | | | |

Gambar 4.3 Set Karakter Kode ASCII

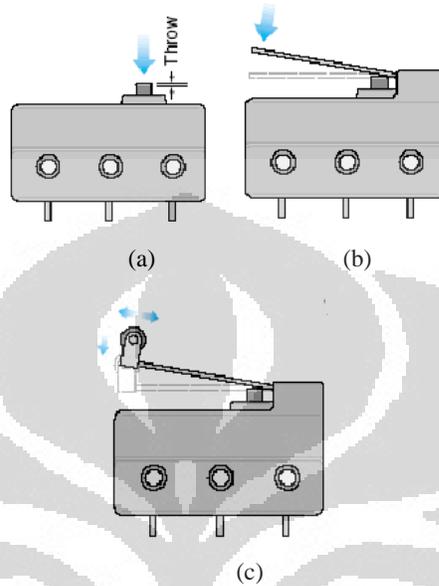
Pada implementasi ini, kode ASCII digunakan untuk menampilkan tulisan golongan dan tarif pada layar LCD. Dalam kode ASCII tulisan golongan dan tarif diubah dalam bentuk biner menjadi :

0100 0111: G ; 0100 1111: O; 0100 0100: L; 0100 1111: O; 0100 1110: N;
0100 0111: G ; 0100 0001: A; 0100 1110:N;

4.1.1.3 Limit Switch

Limit switch dalam implementasi ini adalah untuk menggantikan fungsi dari *treadle*. *Limit switch* adalah alat penghubung dan pemutus arus yang bekerja berdasarkan sentuhan langsung yang digerakan oleh peralatan mekanik sehingga merubah posisi titik sambungannya. *Limit switch* sangat berperan untuk banyak aplikasi, tetapi mereka memiliki dua kekurangan yaitu apabila digunakan

secara terus menerus sebagai peralatan mekanik akhirnya akan rusak, dan *limit switch* membutuhkan sejumlah tekanan fisik untuk digerakkan. Gambar 4.4 menunjukkan beberapa contoh *limit switch*.



Gambar 4.4(a) Tombol tekan (b) Tombol Fleksibel (c) Roller

Limit switch terdiri dari tiga komponen utama, yaitu:

1. Rumah *limit switch*, yaitu tempat kedudukan kontak dan *micro switch*.
2. *Micro switch*, yaitu alat penghubung dan pemutus arus.
3. *Actuator*, merupakan sensor yang mengaktifkan *micro switch*.

Seperti halnya pada push button switch, *limit switch* pun terdapat dua titik sambungan yaitu NO (*Normally Open*) dan NC (*Normally Close*) yang bekerjanya saling bertolak belakang. *Limit switch* yang digunakan dalam implementasi ini adalah *micro switch* dengan spesifikasi sebagai berikut:

- Beroperasi pada tegangan maksimum 250 VAC
- Arus maksimum yang digunakan adalah 5A
- Keluarannya berupa data digital (Logika 0 atau 1)

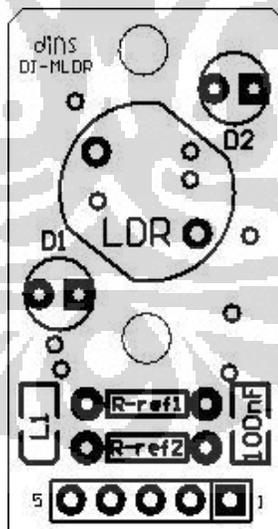
Implementasi limit switch pada sistem ini dikoneksikan dengan slot konektor ekspansi yang ada di FPGA. Input dari 2 limit switch sebagai pendeteksi roda

ganda ada di pin I/O J4 data kesatu dan kedua. Selanjutnya untuk input 1 limit switch yang digunakan sebagai penghitung gandar kendaraan ada di pin I/O J4 data ketiga. Keluaran/output dari limit switch ini adalah led. Led juga berfungsi untuk menjadi indikator untuk pengujian sistem.

4.1.1.4 DI-MLDR

DI-MLDR ini digunakan untuk menggantikan fungsi dari *laser rangefinder* dalam mendeteksi kendaraan yang masuk ke dalam gerbang tol. DI-MLDR (DI-Multifunction LDR) 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. Diantara aplikasi dari DI-MLDR ini adalah untuk pendeteksi benda, dalam sistem ini dibutuhkan untuk mendeteksi adanya kendaraan yang masuk ke dalam gerbang tol. 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) |

Rangkaian sensor ini ditambahkan sebuah *laser-pointer* sebagai sumber cahaya dan menggunakan sumber tegangan dari FPGA yang dikoneksikan dengan pin konektor ekspansi I/O pada FPGA. Cara Kerja rangkaian sensor ini terhadap sistem dapat dijelaskan sebagai berikut:

1. Pin VCC pada sensor dihubungkan ke tegangan sumber positif (3,3) dan pin GND pada sensor ke tegangan sumber negatif atau nol (0V);
2. *Laser-pointer* digunakan sebagai sumber cahaya dan diarahkan ke permukaan sensor dengan diberi jarak selebar jalan gerbang tol sebagai lajur dari gerak kendaraan.
3. Data dari COMP akan berupa digital yang selanjutnya menjadi input dari FPGA untuk di proses. COMP akan berlogika 0 (0 VDC) saat mobil menghalangi cahaya yang menuju permukaan LDR dan ditandai dengan menyalnya LED indikator berwarna hijau, dan COMP akan berlogika 1 (3,3 VDC) saat cahaya mengenai permukaan LDR.

4. Saat keluaran dari sensor ini bernilai 0 (*active low*), maka secara otomatis jalur gerbang tol terbuka dan sistem dalam keadaan siap. Sebaliknya, saat keluaran dari sensor ini bernilai 1, maka sistem dalam keadaan *idle*.

Implementasi DI-MLDR pada sistem ini dikoneksikan dengan slot konektor ekspansi (pin I/O J1) pada FPGA. Output (keluaran) dari sensor ini berupa led yang juga digunakan sebagai indikator pengujian sensor DI-MLDR pada sistem.

4.1.2 Implementasi Perangkat Lunak (*Software*)

Pada rangkaian menggunakan bahasa VHDL, terdapat beberapa bagian didalamnya, diantaranya menentukan *library*, *entity*, *architecture* dan program utama. Dalam rangkaian sistem ini, terdapat sebuah *top module* vhdl bernama main dan module lainnya bernama module1 lcd.

1. Library

Dalam bahasa pemrograman vhdl dikenal pula paket library/pustaka yang berfungsi untuk memudahkan programmer untuk menyelesaikan pekerjaannya karena dalam library tersebut terdapat fungsi-fungsi dan tipe data yang sudah didefinisikan sebelumnya untuk digunakan berulang-ulang. Pada rangkaian ini, library yang digunakan adalah sebagai berikut:

```
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;
```

2. Entity

Pada bagian entity berisi komponen input maupun output pada rangkaian beserta tipe dari input atau output tersebut. Entity pada rangkaian ini adalah sebagai berikut:

```
entity main is
    port(
        LDR, DWHEEL1, DWHEEL2, WHEELC, clk, reset: in
        STD_LOGIC;
```

```

        STATUS_LED, DWHEEL_LED, LCD_E, LCD_RS, LCD_RW, SF_CE0:
out STD_LOGIC;

        SF_D : out STD_LOGIC_VECTOR(3 downto 0);

        WHEELC_LED: out STD_LOGIC_VECTOR(3 downto 0)

    );

end main;

```

Input dalam rangkaian ini terdiri dari 6 buah, yaitu:

1. LDR terhubung dengan DI-MLDR
2. DWHEEL1 terhubung dengan *limit switch 1*
3. DWHEEL2 terhubung dengan *limit switch 2*
4. WHEELC terhubung dengan *limit switch 3*
5. Clock terhubung pada internal clock yang ada pada FPGA
6. Reset terhubung pada *btn east* pada FPGA

Output dalam rangkaian ini terdiri dari 8 buah, yaitu:

1. STATUS_LED berupa nilai dari status led
2. DWHEEL_LED berupa keluaran led untuk hasil *limit switch 1* dan *limit switch 2*
3. LCD_E, LCD_RS, LCD_RW, SF_CE0, SF_D berupa keluaran pada LCD
4. WHEELC_LED berupa keluaran led untuk hasil *limit switch 3*.

3. Architecture

Hanya ada 1 Architecture pada entity main ini yaitu Architecture Behavioral. Dalam Architecture Behavioral ini terdapat beberapa signal dan 1 component lcd. Berikut adalah deklarasi component dan signal tersebut.

```

architecture Behavioral of main is
    component lcd

```

```

port(
    clk, reset : in STD_LOGIC;
    gol : in STD_LOGIC_VECTOR(3 downto 0);
    SF_D : out STD_LOGIC_VECTOR(3 downto 0);
    LCD_E, LCD_RS, LCD_RW, SF_CE0 : out STD_LOGIC
);
end component;

type main_status is (waiting, calculate, display);
signal    status : main_status := waiting;
signal    s_led : STD_LOGIC := '0';

type wstatus is (single_wheel, double_wheel);
signal    wheel_status, ws : wstatus := single_wheel;
signal    w_led : STD_LOGIC := '0';

signal    wheel_counter, wc : STD_LOGIC_VECTOR(3 downto 0)
:= "0000";
signal gol : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal Q1, Q2, Q3, Q4, Q5, Q6, QR : std_logic := '0';
signal counter, counter2 : integer := 0;

```

Architecture behavioral dalam entity main ini merupakan satu – satunya architecture yang menentukan bagaimana keseluruhan sistem bekerja. Di awal dari Architecture Definition Part ini terdapat Component Instantiation untuk Component LCD.

```

MODULE1: lcd
port map(
    clk => clk,
    reset => reset,

```

```

SF_D => SF_D,
LCD_E => LCD_E,
LCD_RS => LCD_RS,
LCD_RW => LCD_RW,
SF_CEO => SF_CEO,
gol => gol
);

```

Pada bagian ini hanya terdapat 1 process yang akan dijalankan tergantung dengan sensitivity list-nya yaitu clk, LDR, DWHEEL1, DWHEEL2, WHEELC, dan reset.

```

detect: process(clk, LDR, DWHEEL1, DWHEEL2, WHEELC, reset)

```

Ada beberapa sub process yang akan dijalankan apabila ada nilai dari sensitivity list tersebut yang berubah. Sub process yang pertama adalah sub process yang bekerja apabila terdapat perubahan pembacaan pada sensor LDR yang menandakan masuknya kendaraan ke zona deteksi golongan.

```

detect: process(clk, LDR, DWHEEL1, DWHEEL2, WHEELC, reset)
begin
if(clk'event and clk = '1') then
...
if(LDR = '1') then
status<= calculate;
s_led <= '1';
end if;
...
end if;
end process;

```

Sub process berikutnya yang akan dijalankan apabila kendaraan sudah terdeteksi masuk ke zona deteksi adalah penghitungan jumlah gandar dan deteksi roda ganda menggunakan dua sensor *limit switch*. Kendaraan dianggap memiliki

roda ganda apabila kedua sensor *limit switch* tertekan secara bersamaan. Berikut adalah sub process untuk deteksi roda ganda tersebut.

```

detect: process(clk, LDR, DWHEEL1, DWHEEL2, WHEELC, reset)
begin
  if(clk'event and clk = '1') then
    ...
    if(DWHEEL1 = '1' and DWHEEL2 = '1') then
      wheel_status <= double_wheel;
      w_led <= '1';
    end if;
    ...
  end if;
end process;

```

Hasil pembacaan pada sub process tersebut dikirimkan ke signal `wheel_status`. Sedangkan untuk penghitungan gandar maka sistem akan melakukan deteksi terhadap *limit switch*, dan melakukan inkrementasi jumlah ban setiap switch tersebut tertekan dan menyimpannya pada signal `wheel_counter`.

```

detect: process(clk, LDR, DWHEEL1, DWHEEL2, WHEELC, reset)
begin
  if(clk'event and clk = '1') then
    ...
    -- Switch debouncer
    if(counter = 50000) then
      Q1 <= WHEELC;
      Q2 <= Q1;
      Q3 <= Q2;
      Q4 <= Q3;
      Q5 <= Q4;
      Q6 <= Q5;
      counter <= 0;
    end if;
  end if;
end process;

```

```

else
    counter<= counter + 1;
end if;

if(QR = '1')then
    wheel_counter <= wheel_counter + 1;
end if;

...
end if;
end process;
...
QR <= Q1 and Q2 and Q3 and Q4 and Q5 and(not Q6);

```

Untuk deteksi jumlah gandar ini, *limit switch* terlebih dahulu dihubungkan dengan debouncer, agar pembacaan counter nya lebih akurat. Setelah semua proses deteksi tersebut selesai dilalui oleh kendaraan, maka selanjutnya tombol reset digunakan untuk menampilkan data hasil deteksi golongan kendaraan ke LCD, dan me-reset semua nilai pada counter.

```

detect: process(clk, LDR, DWHEEL1, DWHEEL2, WHEELC, reset)
begin
    if(clk'event and clk = '1') then
        ...
        if(reset = '1') then
            if(wheel_counter = "0010" and wheel_status = single_wheel)
            then

                gol<= "0001";

            elsif(wheel_counter = "0010" and wheel_status =
            double_wheel) then

                gol<= "0010";

            elsif(wheel_counter = "0011") then

                gol<= "0011";

```

```

elseif(wheel_counter = "0100") then
    gol<= "0100";
elseif(wheel_counter >= "0101") then
    gol<= "0101";
end if;

wheel_counter <= "0000";
wheel_status <= single_wheel;
w_led <= '0';
s_led <= '0';
end if;
...
end if;
end process;

```

4.2 Uji Coba Sistem

4.2.1 Perangkat Keras (*Hardware*)

Pengujian dari perangkat keras dibagi menjadi 3 pengujian, yaitu terhadap sensor DI-MLDR, 2 sensor *limit switch* dan 1 sensor *limit switch*. Ketiga sensor tersebut masing-masing memiliki keluaran berupa led sebagai indikator pengujian pada sistem. Pengujian perangkat keras pada sistem ini dilakukan sebanyak 10 kali untuk masing-masing percobaan. Secara garis besar, langkah-langkah pengujian pada sistem adalah sebagai berikut:

1. Percobaan 1 – Golongan 1

Ketika sensor DI-MLDR bernilai '0', maka sensor mendeteksi adanya kendaraan yang melintas. Keadaan ini juga membuat LCD menampilkan gol 0 dan tarif 0 sebagai inisiasi awal (Gambar 4.6).



Gambar 4.6 Sensor DI-MLDR aktif

Kemudian 2 *limit switch* pertama tidak tertekan secara bersamaan (memiliki nilai '10' atau '01') yang berarti tidak mendeteksi adanya roda ganda, maka led pada posisi kedua yang digunakan sebagai indikator tidak menyala.

Setelah itu *limit switch* kedua menghitung jumlah gandar pada kendaraan berjumlah 2. Indikator ini adalah led yang berubah bitnya sesuai dengan jumlah gandarnya. Pada percobaan ini led berubah menjadi bernilai '010' dilihat dari led posisi kelima. Hasil yang tampil pada LCD adalah gol 1 dan tarif 1000 (dapat dilihat pada Gambar 4.7).



Gambar 4.7 Kendaraan Golongan 1 dan Led Sebagai Indikator

Percobaan 1 dilakukan sebanyak 10 kali dengan data hasil percobaan dapat dilihat pada Tabel 4.2.

Tabel 4.2 Hasil Percobaan 1

| Pembacaan ke- | LED | | | | LCD |
|---------------|------|-------|------|------|-------------------|
| | 4 | 3 | 2 | 1 | |
| 1 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 2 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 3 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 4 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 5 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 6 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 7 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 8 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 9 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |
| 10 | Mati | Hidup | Mati | Mati | Gol 1 –Tarif 1000 |

Pembacaan led dihitung dari posisi kedua dari sebelah kanan. Led 1 menunjukkan bahwa tidak ada roda ganda yang terdeteksi sedangkan led 4 sampai led 2 menunjukkan jumlah gandar dengan angka biner yaitu '010' atau bernilai 2 gandar. Hasil pembacaan led ini membuat lcd menampilkan golongan 1 dengan tarif 1000.

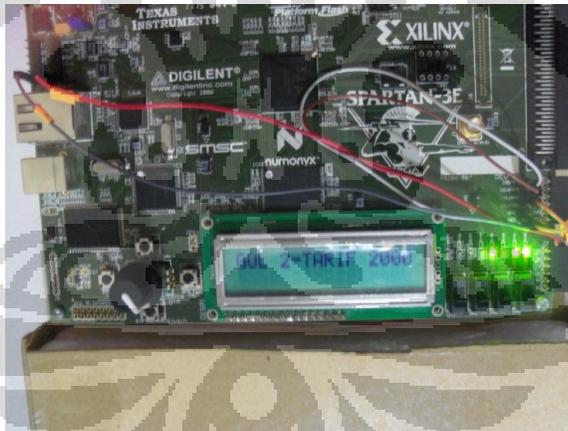
2. Percobaan 2 – Golongan 2

Ketika sensor DI-MLDR bernilai '0', maka sensor mendeteksi adanya kendaraan yang melintas. Kemudian 2 *limit switch* pertama tertekan secara bersamaan (mempunyai nilai '11') yang berarti mendeteksi adanya roda ganda, maka led pada posisi kedua menyala (dapat dilihat pada Gambar 4.8).



Gambar 4.8 Led Sebagai Indikator 2 Limit Switch

Setelah itu, limit switch kedua menghitung jumlah gandar pada kendaraan yang berjumlah 2. Indikator ini adalah led yang berubah bitnya sesuai dengan jumlah gandarnya. Pada percobaan ini led berubah menjadi bernilai '010' dilihat dari led posisi kelima. Hasil yang tampil pada LCD adalah gol 2 dan tarif 2000. Gambar hasil percobaan ini dapat dilihat pada Gambar 4.9.



Gambar 4.9 Kendaraan Golongan 2 dan Led Sebagai Indikator

Percobaan 2 dilakukan sebanyak 10 kali dengan data hasil percobaan dapat dilihat pada Tabel 4.3.

Tabel 4.3 Hasil Percobaan 2

| Pembacaan ke- | Led | | | | LCD |
|---------------|------|-------|------|-------|-------------------|
| | 4 | 3 | 2 | 1 | |
| 1 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 2 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 3 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 4 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 5 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 6 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 7 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 8 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 9 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |
| 10 | Mati | Hidup | Mati | Hidup | Gol 2 –Tarif 2000 |

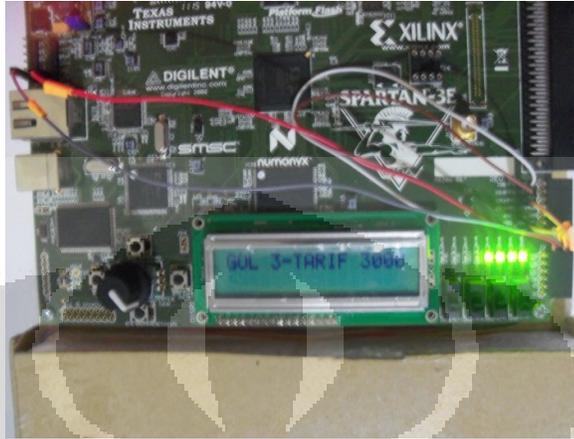
Pembacaan led dihitung dari posisi kedua dari sebelah kanan. Led 1 menunjukkan bahwa ada roda ganda yang terdeteksi sedangkan led 4 sampai led 2 menunjukkan jumlah gandar dengan angka biner yaitu '010' atau bernilai 2 gandar. Hasil pembacaan led ini membuat lcd menampilkan golongan 2 dengan tarif 2000.

3. Percobaan 3 – Golongan 3

Ketika sensor DI-MLDR bernilai '0', maka sensor mendeteksi adanya kendaraan yang melintas. Kemudian 2 *limit switch* pertama tertekan secara bersamaan (mempunyai nilai '11') yang berarti mendeteksi adanya roda ganda, maka led pada posisi kedua menyala (dapat dilihat pada Gambar 4.8).

Setelah itu, *limit switch* kedua menghitung jumlah gandar pada kendaraan yang berjumlah 3. Indikator ini adalah led yang berubah bitnya sesuai dengan jumlah gandarnya. Pada percobaan ini led berubah menjadi bernilai '011' dilihat

dari led posisi kelima. Hasil yang tampil pada LCD adalah gol 3. Gambar hasil percobaan ini dapat dilihat pada Gambar 4.10.



Gambar 4.10 Kendaraan Golongan 3 dan Led Sebagai Indikator

Percobaan 3 dilakukan sebanyak 10 kali dengan data hasil percobaan dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil Percobaan 3

| Pembacaan ke- | LED | | | | LCD |
|---------------|------|-------|-------|-------|-------------------|
| | 4 | 3 | 2 | 1 | |
| 1 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 2 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 3 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 4 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 5 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 6 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 7 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 8 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 9 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |
| 10 | Mati | Hidup | Hidup | Hidup | Gol 3 –Tarif 3000 |

Pembacaan led dihitung dari posisi kedua dari sebelah kanan. Led 1 menunjukkan bahwa ada roda ganda yang terdeteksi sedangkan led 4 sampai led 2 menunjukkan jumlah gandar dengan angka biner yaitu '011' atau bernilai 3 gandar. Hasil pembacaan led ini membuat lcd menampilkan golongan 3 dengan tarif 3000.

4. Percobaan 4 – Golongan 4

Ketika sensor DI-MLDR bernilai '0', maka sensor mendeteksi adanya kendaraan yang melintas. Kemudian 2 *limit switch* pertama tertekan secara bersamaan (mempunyai nilai '11') yang berarti mendeteksi adanya roda ganda, maka led pada posisi kedua menyala (dapat dilihat pada Gambar 4.8).

Setelah itu, *limit switch* kedua menghitung jumlah gandar pada kendaraan yang berjumlah 4. Indikator ini adalah led yang berubah bitnya sesuai dengan jumlah gandarnya. Pada percobaan ini led berubah menjadi bernilai '100' dilihat dari led posisi kelima. Hasil yang tampil pada LCD adalah gol 4 dan tarif 4000. Gambar hasil percobaan ini dapat dilihat pada Gambar 4.11.



Gambar 4.11 Kendaraan Golongan 4 dan Led Sebagai Indikator

Percobaan 4 dilakukan sebanyak 10 kali dengan data hasil percobaan dapat dilihat pada Tabel 4.5.

Tabel 4.5 Hasil Percobaan 4

| Pembacaan ke- | LED | | | | LCD |
|---------------|-------|------|------|-------|-------------------|
| | 4 | 3 | 2 | 1 | |
| 1 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 2 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 3 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 4 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 5 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 6 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 7 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 8 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 9 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |
| 10 | Hidup | Mati | Mati | Hidup | Gol 4 –Tarif 4000 |

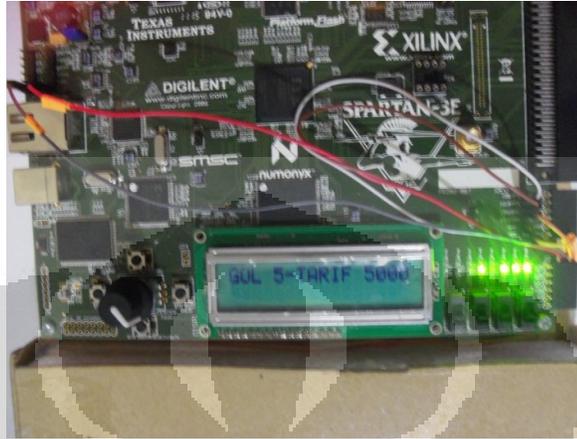
Pembacaan led dihitung dari posisi kedua dari sebelah kanan. Led 1 menunjukkan bahwa ada roda ganda yang terdeteksi sedangkan led 4 sampai led 2 menunjukkan jumlah gandar dengan angka biner yaitu '100' atau bernilai 4 gandar. Hasil pembacaan led ini membuat lcd menampilkan golongan 4 dengan tarif 4000.

5. Percobaan 5 – Golongan 5

Ketika sensor DI-MLDR bernilai '0', maka sensor mendeteksi adanya kendaraan yang melintas. Kemudian 2 *limit switch* pertama tertekan secara bersamaan (mempunyai nilai '11') yang berarti mendeteksi adanya roda ganda, maka led pada posisi kedua menyala (dapat dilihat pada Gambar 4.8).

Setelah itu, *limit switch* kedua menghitung jumlah gandar pada kendaraan yang berjumlah 5. Indikator ini adalah led yang berubah bitnya sesuai dengan jumlah gandarnya. Pada percobaan ini led berubah menjadi bernilai '101' dilihat

dari led posisi kelima. Hasil yang tampil pada LCD adalah gol 5 dan tarif 5000. Gambar hasil percobaan ini dapat dilihat pada Gambar 4.12.



Gambar 4.12 Kendaraan Golongan 5 dan Led Sebagai Indikator

Percobaan 5 dilakukan sebanyak 10 kali dengan data hasil percobaan dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil Percobaan 5

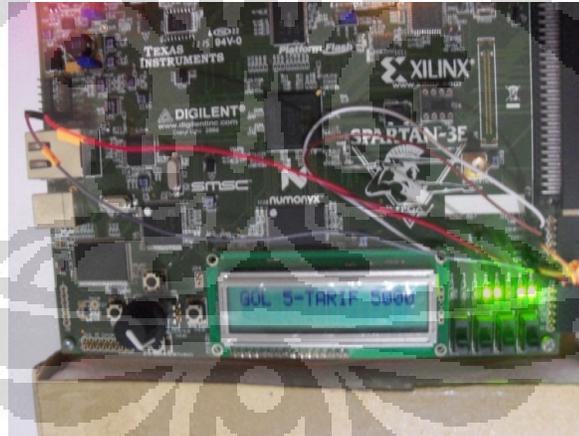
| Pembacaan ke- | LED | | | | LCD |
|---------------|-------|------|-------|-------|-------------------|
| | 4 | 3 | 2 | 1 | |
| 1 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 2 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 3 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 4 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 5 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 6 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 7 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 8 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 9 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |
| 10 | Hidup | Mati | Hidup | Hidup | Gol 5 –Tarif 5000 |

Pembacaan led dihitung dari posisi kedua dari sebelah kanan. Led 1 menunjukkan bahwa ada roda ganda yang terdeteksi sedangkan led 4 sampai led 2 menunjukkan jumlah gandar dengan angka biner yaitu '101' atau bernilai 5 gandar. Hasil pembacaan led ini membuat lcd menampilkan golongan 5 dengan tarif 5000.

6. Percobaan 6 – Golongan 5

Ketika sensor DI-MLDR bernilai '0', maka sensor mendeteksi adanya kendaraan yang melintas. Kemudian 2 *limit switch* pertama tertekan secara bersamaan (mempunyai nilai '11') yang berarti mendeteksi adanya roda ganda, maka led pada posisi kedua menyala (dapat dilihat pada Gambar 4.8).

Setelah itu, *limit switch* kedua menghitung jumlah gandar pada kendaraan yang berjumlah 5. Indikator ini adalah led yang berubah bitnya sesuai dengan jumlah gandarnya. Pada percobaan ini led berubah menjadi bernilai '110' dilihat dari led posisi kelima. Hasil yang tampil pada LCD adalah gol 5 dan tarif 5000. Gambar hasil percobaan ini dapat dilihat pada Gambar 4.13.



Gambar 4.13 Kendaraan Golongan 5 dan Led Sebagai Indikator

Percobaan 6 dilakukan sebanyak 10 kali dengan data hasil percobaan dapat dilihat pada Tabel 4.7.

Tabel 4.7 Hasil Percobaan 6

| Pembacaan ke- | LED | | | | LCD |
|---------------|-------|-------|------|-------|-------------------|
| | 4 | 3 | 2 | 1 | |
| 1 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 2 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 3 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 4 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 5 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 6 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 7 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 8 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 9 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |
| 10 | Hidup | Hidup | Mati | Hidup | Gol 5 –Tarif 5000 |

Pembacaan led dihitung dari posisi kedua dari sebelah kanan. Led 1 menunjukkan bahwa ada roda ganda yang terdeteksi sedangkan led 4 sampai led 2 menunjukkan jumlah gandar dengan angka biner yaitu '110' atau bernilai 6 gandar. Hasil pembacaan led ini membuat lcd menampilkan golongan 5 dengan tarif 5000.

4.2.2 Analisis Hasil Percobaan

Dari keenam percobaan diatas, dapat dilihat bahwa pengujian pada tiap parameter yang telah dibuat adalah akurat 100%. Hal ini berarti sistem yang telah dibuat dapat diandalkan. Masing-masing sensor bekerja dengan baik saat dihubungkan dengan FPGA walaupun sensor yang digunakan adalah *limit switch* dan DI-MLDR. Akan tetapi respon waktu yang diberikan masing-masing sensor termasuk cepat dengan delay yang dibuat untuk tiap-tiap sensor adalah 1ms.

Dengan akurasi pembacaan dan tidak adanya bit loss pada pentransmisian data dari sensor ke FPGA, maka dapat dikatakan sistem ini bekerja dengan baik dan memenuhi standar yang diinginkan untuk digunakan pada implementasi sistem penentuan golongan dan tarif tol otomatis.

Namun, sistem ini masih berupa purwarupa dengan sensor yang digunakan belum merepresentasikan sistem sesungguhnya. Untuk benar-benar dapat mengimplementasikan sistem ini, alat-alat yang dibutuhkan harus sesuai dengan keadaan sebenarnya dan diujikan pada lahan yang nyata.



BAB 5

KESIMPULAN

Teknologi FPGA dapat dikembangkan menjadi suatu sistem tertanam yang terpadu dimana FPGA sebagai pusat kendalinya. Salah satu contoh penggunaannya adalah sebagai pengendali dari sebuah sistem deteksi golongan kendaraan untuk selanjutnya digunakan sebagai pentarifan biaya jalan bebas hambatan. Dalam buku ini telah dilakukan perancangan dan implementasi sistem tersebut dengan menggunakan beberapa sensor pengganti yang memiliki analogi sifat sebagai purwarupa dari sistem yang sesungguhnya.

Secara umum sistem ini dirancang agar dapat mempermudah pekerjaan manusia di masa datang. Sistem ini juga dapat dikembangkan sesuai kebutuhan sehingga mendukung pemanfaatan dalam skala penggunaan yang besar, mencakup lebih banyak pengguna.

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

1. Pemodelan dan purwarupa Sistem Penentuan Golongan kendaraan pada jalan tol ini siap untuk dirancang menjadi sebuah sistem yang utuh, dan siap untuk dikembangkan lebih lanjut.
2. Agar dapat dikembangkan menjadi sistem yang utuh, maka dibutuhkan pengganti terhadap sensor yang digunakan pada purwarupa ini, yaitu sensor-sensor ideal yang sejatinya digunakan pada sistem sejenis seperti yang telah dijelaskan pada dasar teori buku ini.

Dari hasil yang didapat ini, akan dijadikan bahan studi lebih lanjut untuk pengembangan sistem penentuan golongan kendaraan pada jalan tol.

DAFTAR ACUAN

- [1] Rencana Penerapan Sistem Pengumpulan Tol Elektronik (Electronic Toll Collection System) di Indonesia. Diakses tanggal 27 Desember 2011. http://www.ftsl.itb.ac.id/wp-content/02-Rudi_Hermawan_31_agustus.pdf
- [2] Laporan Usaha | Annual Report. Diakses tanggal 20 Oktober 2011. http://jasamarga.com/annual_report/ar2009/laporan_usaha.html
- [3] Tulisan tentang FPGA <http://tenikcal.blogspot.com/2010/10/tulisan-tentang-fpga.html>
- [4] “Xilinx Spartan 3e – Documentations”. Diakses tanggal 01 Januari 2012. <http://www.xilinx.com/support/index.htm>
- [5] Intec – Treadle – Disposition. Diakses tanggal 15 Januari 2012. <http://www.intecar.com.ar/disposition.htm>
- [6] Teknik pengendalian lcd karakter. Diakses tanggal 03 Januari 2012. <http://depokinstruments.com/2010/02/08/teknik-pengendalian-lcd-karakter-i/>
- [7] DI-MLDR. Diakses tanggal 10 Mei 2012. <http://depokinstruments.com/2010/02/08/24/produk-baru-di-mldr-di-multifunction-ldr/>
- [8] Limit Switch. Diakses tanggal 10 Mei 2012. <http://www.scribd.com/doc/78073696/41/II-3-2-2-1-Limit-Switch>
- [9] Library VHDL. Diakses tanggal 12 Juni 2012. <http://viana-chubby.blogspot.com/2010/10/library-vhdl.html>
- [10] Tutorial VHDL. Diakses tanggal 12 April 2012. <http://somejunks.wordpress.com/2011/11/01/tutorial-mari-belajar-vhdl-secara-singkat-1/>

DAFTAR PUSTAKA

- [1]. Rudy Hermawan Karsaman. “Rencana Penerapan Sistem Pengumpulan Tol Elektronik (Electronic Toll Collection System) di Indonesia”. 2005.
- [2]. Muhammad Arzaki. “Konstruksi Formalisasi Klasifikasi Kendaraan Otomatis (AVC/ Automatic Vehicle Classification) dengan Menggunakan Kalkulus Durasi”. 2011.
- [3]. Draft RPP kendaraan 2010. [online]. <http://hubdat.web.id/peraturan-pemerintah/979-draft-rpp-tahun-2010-tentang-kendaraan/download>
- [4]. Kamus Istilah Bidang Pekerjaan Umum. 2008 [online] <http://pustaka.pu.go.id/uploads/resensi/kamusistilah.pdf>
- [5]. Surat Keputusan no.1064/AJ.401/DRJD/2011 Tentang Pengaturan Lalu Lintas.
- [6]. Rudy Hermawan. “Kaji Ulang Penentuan Tarif dan Sistem Penggolongan Kendaraan Jalan Tol di Indonesia”. 2009.
- [7]. Marthyn Hutahaean. “Evaluasi Kapasitas dan Pelayanan Gerbang Tol Tanjung Morawa”. 2007.
- [8]. Hikmat Iskandar. “Kajian Penyesuaian Golongan dan Tarif Tol dengan Mempertimbangkan Pengrusakan Perkerasan Jalan Oleh Kendaran Berat”. 2009.
- [9]. Bayu Tenoyo. “Spesifikasi Sistem Automatic Vehicle Classification Menggunakan B-Method”. 2011.

LAMPIRAN

```

-----
-----
-- Company: Fakultas Teknik Universitas Indonesia - Departemen
Teknik Elektro
-- Engineer: Hana Afiifah
--
-- Create Date: 2012
-- Design Name: Car Group Detection
-- Module Name: main - Behavioral
-- Project Name: Skripsi
-- Target Devices: Spartan 3E Starter Board
-- Tool versions: 13.2
-----
-----
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;

entity main is
    port(
        LDR, DWHEEL1, DWHEEL2, WHEELC, clk, reset: in STD_LOGIC;
        STATUS_LED, DWHEEL_LED, LCD_E, LCD_RS, LCD_RW, SF_CE0: out
STD_LOGIC;
        SF_D : out STD_LOGIC_VECTOR(3 downto 0);
        WHEELC_LED: out STD_LOGIC_VECTOR(3 downto 0)
    );
end main;
architecture Behavioral of main is
    component lcd
        port(
            clk, reset : in STD_LOGIC;
            gol : in STD_LOGIC_VECTOR(3 downto 0);
            SF_D : out STD_LOGIC_VECTOR(3 downto 0);
            LCD_E, LCD_RS, LCD_RW, SF_CE0 : out STD_LOGIC
        );
    end component;

```

```

type main_status is (waiting, calculate, display);
signal status : main_status := waiting;
signal s_led : STD_LOGIC := '0';

type wstatus is (single_wheel, double_wheel);
signal wheel_status, ws : wstatus := single_wheel;
signal w_led : STD_LOGIC := '0';

signal wheel_counter, wc : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal gol : STD_LOGIC_VECTOR(3 downto 0) := "0000";
signal Q1, Q2, Q3, Q4, Q5, Q6, QR : std_logic := '0';
signal counter, counter2 : integer := 0;

begin
  MODULE1: lcd
    port map(
      clk => clk,
      reset => reset,
      SF_D => SF_D,
      LCD_E => LCD_E,
      LCD_RS => LCD_RS,
      LCD_RW => LCD_RW,
      SF_CE0 => SF_CE0,
      gol => gol
    );

  detect: process(clk, LDR, DWHEEL1, DWHEEL2, WHEELC, reset)
  begin
    if(clk'event and clk = '1') then
      if(counter = 50000) then
        Q1 <= WHEELC;
        Q2 <= Q1;
        Q3 <= Q2;
        Q4 <= Q3;
        Q5 <= Q4;
        Q6 <= Q5;
        counter <= 0;
      end if;
    end if;
  end process;
end;

```

```

else
    counter <= counter + 1;
end if;

if(QR = '1')then
    wheel_counter <= wheel_counter + 1;
end if;

if(LDR = '1') then
    status <= calculate;
    s_led <= '1';
end if;
if(DWHEEL1 = '1' and DWHEEL2 = '1') then
    wheel_status <= double_wheel;
    w_led <= '1';
end if;

if(reset = '1') then
    if(wheel_counter = "0010"
and wheel_status = single_wheel) then
        gol <= "0001";
    elsif(wheel_counter = "0010"
and wheel_status = double_wheel) then
        gol <= "0010";
    elsif(wheel_counter = "0011") then
        gol <= "0011";
    elsif(wheel_counter = "0100") then
        gol <= "0100";
    elsif(wheel_counter >= "0101") then
        gol <= "0101";
    end if;

    wheel_counter <= "0000";
    wheel_status <= single_wheel;
    w_led <= '0';
    s_led <= '0';
end if;
end if;
end process;

```

```
WHEELC_LED <= wheel_counter;  
STATUS_LED <= s_led;  
DWHEEL_LED <= w_led;  
QR <= Q1 and Q2 and Q3 and Q4 and Q5 and(not Q6);  
  
end Behavioral;
```

