



UNIVERSITAS INDONESIA

**RANCANG BANGUN DAN ANALISIS SISTEM PEMANTAU
POLUSI UDARA PADA AREA PARKIR TERTUTUP
MENGUNAKAN FPGA XILINX SPARTAN 3E
DAN SENSOR GAS CO MQ7**

SKRIPSI

**Rr. WULAN APRILIYANTI P.
0806339313**

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



UNIVERSITAS INDONESIA

**RANCANG BANGUN DAN ANALISIS SISTEM PEMANTAU
POLUSI UDARA PADA AREA PARKIR TERTUTUP
MENGUNAKAN FPGA XILINX SPARTAN 3E
DAN SENSOR GAS CO MQ7**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana.

**Rr. WULAN APRILIYANTI P.
0806339313**


**FAKULTAS TEKNIK UNIVERSITAS INDONESIA
DEPARTEMEN TEKNIK ELEKTRO
TEKNIK KOMPUTER
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 : Rr. Wulan Apriliyanti P.

NPM : 0806339313

Tanda Tangan : 

Tanggal : Rabu, 13 Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Rr. Wulan Apriliyanti P.
NPM : 0806339313
Program Studi : Teknik Komputer
Judul Skripsi : Rancang Bangun dan Analisis Sistem
Pemantau Polusi Udara pada Area Parkir
Tertutup Menggunakan FPGA Xilinx
Spartan 3E dan Sensor Gas CO MQ7

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang dilakukan untuk memperoleh gelar Sarjana Teknik pada program studi Teknik Komputer, Fakultas Teknik, Universitas Indonesia.

DEWAN PENGUJI

Pembimbing : Prima Dewi Purnamasari S.T., M.Sc (.....)

Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng (.....)

Penguji : Ir. A. Endang Sriningsih M.T. Si (.....)

Ditetapkan di : Depok
Tanggal : 22 Juni 2012

KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Allah SWT, karena atas segala rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi ini. Saya menyadari bahwa skripsi ini tidak akan terselesaikan tanpa bantuan dari berbagai pihak. Mulai dari proses pembelajaran, analisa yang telah dijalani dan proses penyusunan dari skripsi ini, saya ingin mengucapkan terima kasih kepada:

1. Prima Dewi Purnamasari S.T., M.Sc., selaku pembimbing telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Orang tua dan keluarga yang telah memberikan bantuan dukungan moril dan material;
3. Vicky D. Kurniawan, Evan G. Sumbayak, dan teman-teman dari Teknik Komputer angkatan 2008, yang tiada hentinya mendukung saya baik secara langsung maupun tidak langsung;
4. Rekan-rekan asisten dari Laboratorium Digital Departemen Teknik Elektro; dan
5. Irwan Kustianto, ST. yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

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

Depok, 13 Juni 2012

Penulis

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI
SKRIPSI UNTUK KEPENTINGAN AKADEMIS

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

Nama : Rr. Wulan Apriliyanti P.

NPM : 0806339313

Program Studi : Teknik Komputer

Departemen : Teknik Elektro

Fakultas : Teknik

Jenis Karya : Skripsi

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

“Rancang Bangun dan Analisis Sistem Pemantau Polusi Udara pada Area Parkir Tertutup Menggunakan FPGA Xilinx Spartan 3E dan Sensor Gas CO MQ7”

Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia Berhak menyimpan, mengalihmediakan/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 13 Juni 2012

Yang menyatakan



(Rr. Wulan Apriliyanti P.)

ABSTRAK

Nama : Rr. Wulan Apriliyanti P.
Program Studi : Teknik Komputer
Judul : Rancang Bangun dan Analisis Sistem Pemantau
Polusi Udara pada Area Parkir Tertutup
Menggunakan FPGA Xilinx Spartan 3E dan
Sensor Gas CO MQ7

Penelitian pada skripsi ini merancang, membuat, dan menganalisis sistem tertanam pemantau polusi udara pada area parkir tertutup menggunakan FPGA Xilinx Spartan 3E dan sensor gas CO MQ7. Sistem ini berguna untuk mengatasi secara dini kasus keracunan gas emisi kendaraan bermotor yang terendap pada area parkir tertutup. Metode yang digunakan dalam penelitian mengikuti tahapan *Software Development Life Cycle* (SDLC). Bahasa yang digunakan untuk mengkonfigurasi FPGA Xilinx Spartan 3E adalah VHDL melalui Xilinx ISE Design Suite 13.2. Selain itu, diperlukan dua rangkaian tambahan sebagai antarmuka, yaitu rangkaian *Pulse Width Modulation* (PWM) dan transduser. FPGA ini akan mendapatkan data pembacaan sensor tiap 19,11 ms. Pengambilan data dilakukan dengan pengambilan sampel pada *sense phase* sensor yang diambil tiap 10 detik selama 15 menit. Berdasarkan pengujian, sistem menghasilkan selisih pembacaan sebesar 1,76 ppm (2,45% kesalahan) terhadap data normal.

Kata kunci :
FPGA, Xilinx Spartan 3E, sensor CO MQ7, sistem tertanam, VHDL.

ABSTRACT

Name : Rr. Wulan Apriliyanti P.
Study Program : Computer Engineering
Title : Development and Analysis of Air Pollution Monitoring System at Closed Parking Area Using FPGA Xilinx Spartan 3E and MQ7 CO Gas Sensor

This thesis discusses the design, manufacture, and analyzes the embedded air pollution monitor system in a enclosed parking area using the FPGA Xilinx Spartan 3E and the CO MQ7 gas sensor. This system is useful as a precautionary measure in cases of motor vehicles gas emission poisoning deposited in enclosed parking area. The method used in this research follows the Software Development Life Cycle (SDLC). The programming language used in configuring the FPGA Xilinx Spartan 3E is VHDL using Xilinx ISE Design Suite 13.2. In addition, two additional circuit is needed to act as an interface, a Pulse Width Modulation (PWM) and a transducer. The FPGA reads the data every 19.11 ms. Data extractions is performed by extracting samples from the sense phase sensor every 10 seconds for 15 minutes. The test resulted in a deviation of 1.76 ppm (2.45% error) form normal data.

Key words:

Embedded system, FPGA, MQ7 CO gas sensor , Xilinx Spartan 3E.

DAFTAR ISI

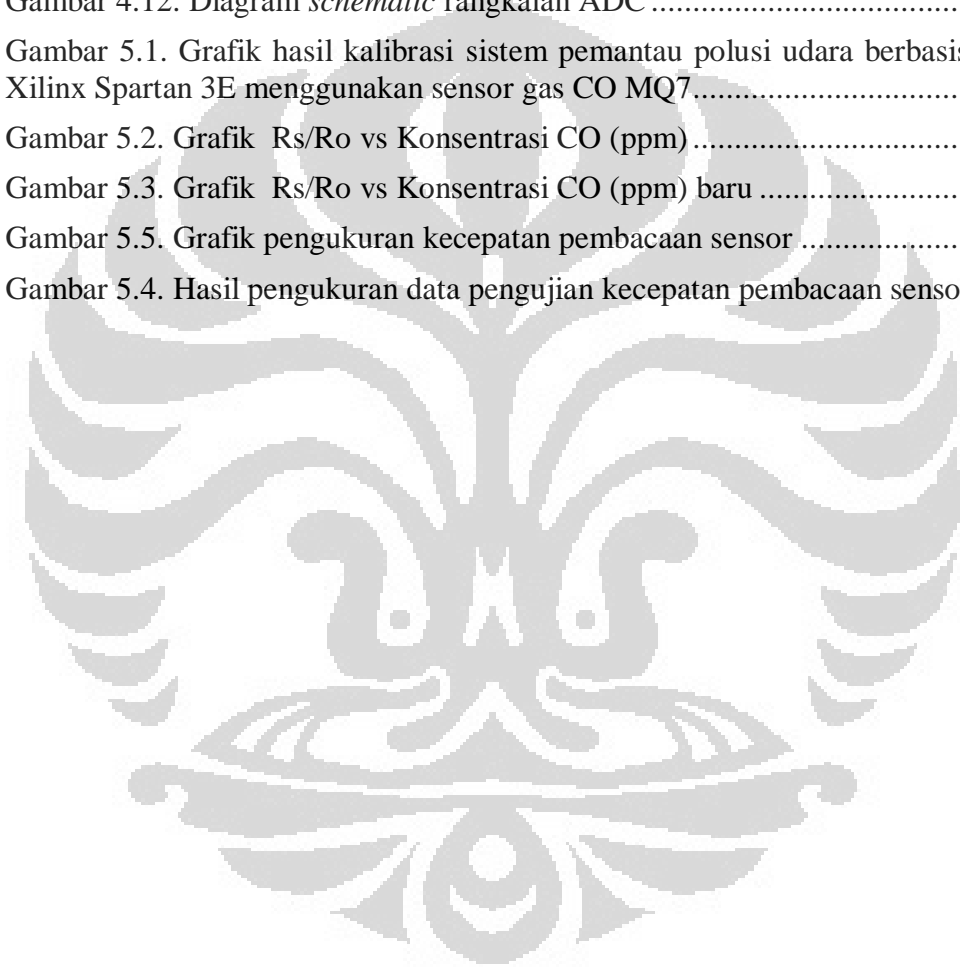
HALAMAN PERNYATAAN ORISINALITAS	iii
HALAMAN PENGESAHAN	iv
KATA PENGANTAR	v
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan Penelitian	2
1.3 Batasan Penelitian	2
1.4 Metodologi Penelitian	2
1.5 Sistematika Penulisan	3
BAB II KONSEP FPGA XILINX SPARTAN 3E DAN SENSOR MQ7	5
2.1 Pencemaran Udara	5
2.1.1 Karbon monoksida (CO)	5
2.2 Field Programmable Gate Array (FPGA)	9
2.2.1 FPGA Xilinx Spartan 3S500E	10
2.2.2 Komponen Board Starter Kit Spartan-3E	14
2.3 VHDL	19
2.3.1 Cara Penulisan dalam VHDL	20
2.3.2 Pendeskripsian VHDL	21
2.4 Pulse-Width Modulation (PWM)	22
2.5 Sensor	23
2.5.1 Hanwei MQ7 CO Gas Sensor	23
2.6 Perlengkapan Sistem Peringatan	26
2.6.1 Exhaust Fan	26
2.6.2 Buzzer	26
2.6.3 LED	27

2.6.4 CRT.....	27
BAB III PERANCANGAN SISTEM PEMANTAU POLUSI UDARA BERBASIS FPGA XILINX SPARTAN 3E MENGGUNAKAN SENSOR GAS CO MQ7.....	28
3.1 System Requirement	28
3.2 Desain Sistem Pemantau Polusi Udara	29
3.2.1 Desain Perangkat Keras (<i>Hardware</i>).....	30
3.2.2 Algoritma Perangkat Lunak	32
BAB IV IMPLEMENTASI SISTEM PEMANTAU POLUSI UDARA BERBASIS FPGA XILINX SPARTAN 3E MENGGUNAKAN SENSOR GAS CO MQ7.....	35
4.1 Implementasi Perangkat Keras (<i>Hardware</i>).....	35
4.2 Implementasi Perangkat Lunak	37
4.2.1 Pemberian Tegangan ke MQ7	38
4.2.2 <i>Analog Capture Circuit</i>	41
4.2.3 Pengaturan Bus <i>Serial Peripheral Interface</i> (SPI).....	43
BAB V PENGUJIAN DAN ANALISA SISTEM PEMANTAU POLUSI UDARA BERBASIS FPGA XILINX SPARTAN 3E MENGGUNAKAN SENSOR GAS CO MQ7.....	44
5.1 Pengujian Pembacaan Kadar Gas CO oleh Sensor MQ7.....	44
5.2 Pengujian Kecepatan Pembacaan Sensor	48
BAB VI KESIMPULAN	50
DAFTAR REFERENSI	51
DAFTAR PUSTAKA.....	53
LAMPIRAN.....	54

DAFTAR GAMBAR

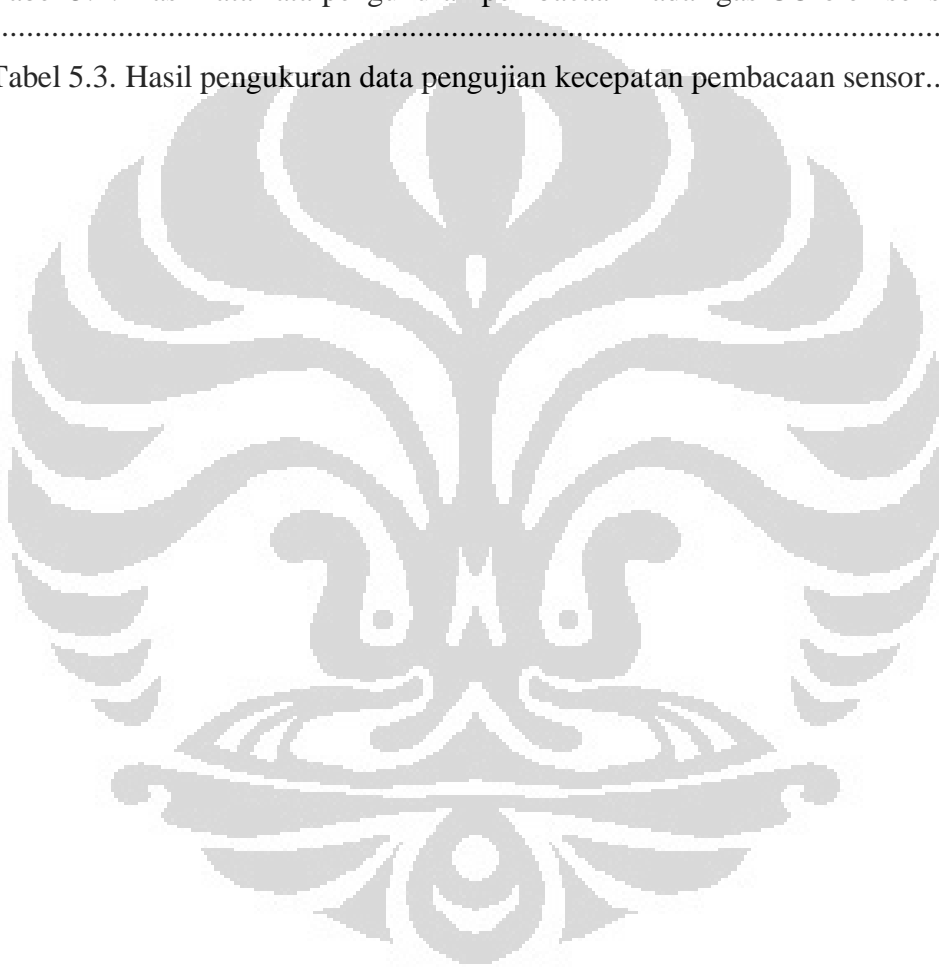
Gambar 2.1. Ikatan senyawa karbon monoksida (CO) [6]	6
Gambar 2.2. Korelasi FPGA dan IC logic lainnya[9].	9
Gambar 2.3. FPGA Spartan 3E[11].....	11
Gambar 2.4. Arsitektur FPGA 3E[12]	11
Gambar 2.5. <i>Logic-cell</i> [12].....	12
Gambar 2.6. <i>Logic-cell</i> yang terhubung melalui <i>carry chain</i> [12].....	12
Gambar 2.8. UCF LED[11].....	14
Gambar 2.7. 8 LED[11]	14
Gambar 2.9. Clock input[11].....	15
Gambar 2.10. UCF <i>clock sources</i> [11]	16
Gambar 2.11. Konektor ekspansi[11]	16
Gambar 2.12. Mapping J1[11]	17
Gambar 2.15. UCF 6-pin <i>accessory header</i> [11]	17
Gambar 2.14. Mapping J4[11]	17
Gambar 2.13. Mapping J2[11]	17
Gambar 2.16. <i>Analog Capture Circuit</i> [11]	18
Gambar 2.17. Detail <i>Analog Capture Circuit</i> [11].....	19
Gambar 2.18. Penulisan <i>entity</i> pada VHDL [9]	20
Gambar 2.20. Penulisan <i>architecture</i> pada VHDL [9]	21
Gambar 2.19. Contoh <i>entity</i> pada gerbang and [9].....	21
Gambar 2.21. <i>Duty cycle</i> [18]	23
Gambar 2.22. Struktur MQ7[13].....	24
Gambar 2.23. Rangkaian sensor MQ7[13]	24
Gambar 2.24. Grafik karakteristik sensitivitas sensor MQ7[13].....	25
Gambar 3.1. <i>Use case diagram</i>	29
Gambar 3.2. Blok diagram sistem pemantau polusi udara berbasis FPGA	30
Gambar 3.3. <i>Prototype</i> area parkir	31
Gambar 3.4. Diagram alir sistem.....	33
Gambar 3.5. <i>Sequence diagram</i>	34
Gambar 4.1. Alur pengembangan perangkat keras.....	35
Gambar 4.3. Jangkauan pembacaan <i>Analog Capture Circuit</i> [11].....	36
Gambar 4.2. Rangkaian antar muka sensor - FPGA.....	36

Gambar 4.4. Rangkaian transduser	37
Gambar 4.5. Alur diagram sistem pemantau polusi udara berbasis FPGA	38
Gambar 4.6. <i>Duty cycle</i> 100%	39
Gambar 4.7. <i>Duty cycle</i> 28%	40
Gambar 4.8. <i>Pseudocode</i> PWM	40
Gambar 4.9. Sinyal PWM untuk merubah tegangan	41
Gambar 4.11. Antarmuka pengkonversi A/D [11]	42
Gambar 4.10. Pewaktuan SPI ketika berkomunikasi dengan <i>amplifier</i> [11]	42
Gambar 4.12. Diagram <i>schematic</i> rangkaian ADC	43
Gambar 5.1. Grafik hasil kalibrasi sistem pemantau polusi udara berbasis FPGA Xilinx Spartan 3E menggunakan sensor gas CO MQ7	45
Gambar 5.2. Grafik R_s/R_o vs Konsentrasi CO (ppm)	46
Gambar 5.3. Grafik R_s/R_o vs Konsentrasi CO (ppm) baru	47
Gambar 5.5. Grafik pengukuran kecepatan pembacaan sensor	49
Gambar 5.4. Hasil pengukuran data pengujian kecepatan pembacaan sensor	48



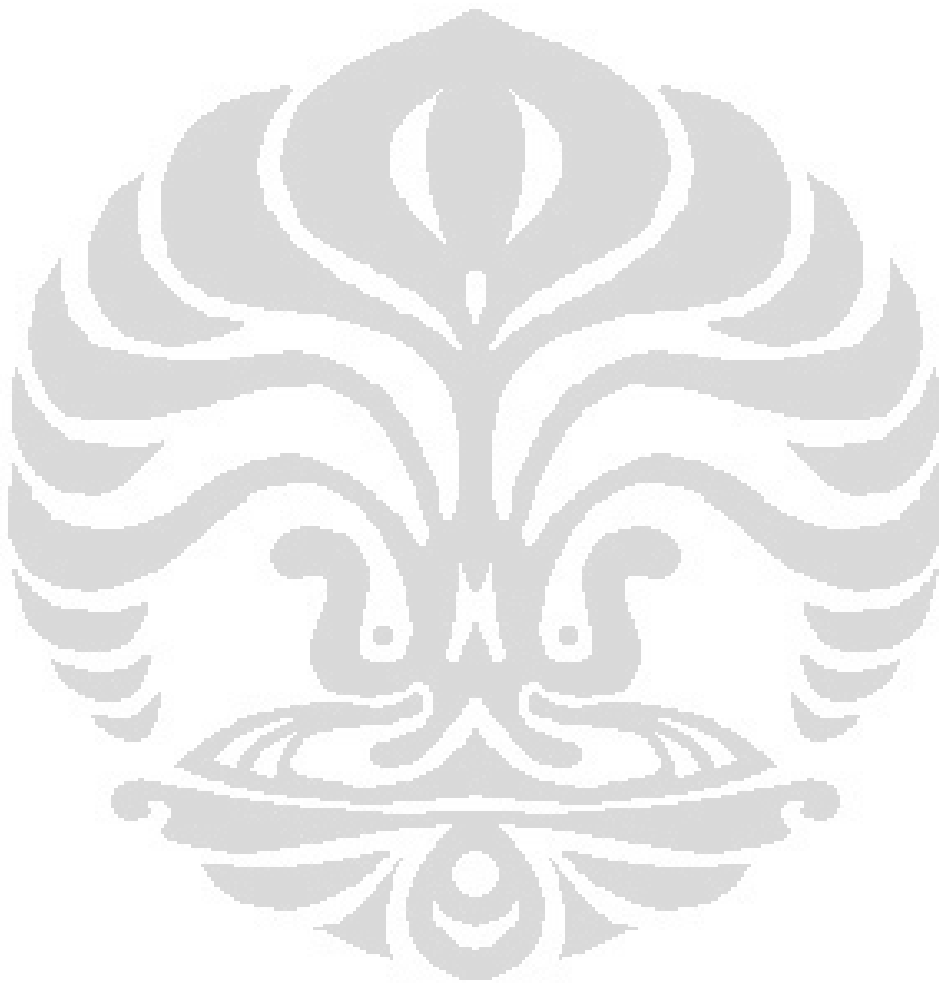
DAFTAR TABEL

Tabel 2.1. Pengaruh Kadar CO dalam darah terhadap kondisi tubuh. [8].....	7
Tabel 2.2. Pengaturan gain untuk <i>pre-amplifier</i> [11]	19
Tabel 2.3. Tabel Karakteristik Sensitivitas Sensor[13]	25
Tabel 4.1. Konfigurasi bus SPI [11]	43
Tabel 5.1. Kadar CO terhadap gas yang diujikan.....	44
Tabel 5.2. Hasil rata-rata pengukuran pembacaan kadar gas CO oleh sensor MQ7	46
Tabel 5.3. Hasil pengukuran data pengujian kecepatan pembacaan sensor.....	48



DAFTAR LAMPIRAN

Tabel A-1 Data pengukuran kalibrasi sistem	54
Gambar A-1 Perangkat keras sistem pemantau polusi udara di area parkir tertutup berbasis FPGA Xilinx Spartan 3E dan menggunakan sensor MQ7	54



BAB I

PENDAHULUAN

1.1 Latar Belakang

Laju pertumbuhan penduduk di kota besar di Indonesia, terutama di Jakarta, dari tahun ke tahun selalu meningkat. Jumlah pendatang pun selalu bertambah menyebabkan Jakarta semakin padat. Lahan kosong pun makin berkurang diimbangi dengan berbagai pembangunan gedung-gedung bertingkat, apartemen, serta perumahan. Pada awal 1970-an[1], gubernur Ali Sadikin menyatakan bahwa Jakarta tertutup bagi pendatang. Namun pada akhirnya, kebijakan tersebut sulit untuk dilakukan bahkan dilupakan oleh masa pemerintahan gubernur-gubernur selanjutnya. Hingga saat ini, Jakarta masih harus bergelut dengan masalah-masalah yang terjadi akibat kepadatan penduduk, seperti banjir, kemacetan, serta kekurangan alat transportasi umum yang memadai. Alat transportasi umum yang kurang memadai pun menjadi penyebab banyaknya individu di Jakarta ini yang berkeinginan untuk membeli alat transportasi pribadi. Dikaitkan dengan kepadatan lahan, banyaknya kendaraan pribadi ini membutuhkan area parkir yang luas, sebagai solusi dibangunlah gedung-gedung parkir tertutup agar dapat menampung banyaknya kendaraan pribadi tersebut, terutama di pusat perbelanjaan.

Gedung yang menyediakan parkir tertutup tentunya harus memperhatikan adanya sirkulasi udara dalam ruangan. Beberapa waktu yang lalu, tepatnya pada tanggal 27 Juli 2010[2], terjadi insiden keracunan gas buang di Pelabuhan Merak, Banten, yang mengakibatkan kematian pada salah satu penumpang bus yang sedang tidur dalam bus yang diparkir. Beberapa kali pula, kasus yang hampir serupa terjadi di salah satu gedung pertokoan di Jalan Jend. Sudirman, Jakarta. Hal ini tentunya perlu diperhatikan dengan baik agar jangan sampai terulang.

Penyebab utama dari kasus-kasus tersebut di atas adalah gas buang (dalam hal ini yang dominan adalah karbon monoksida) yang terendap dalam ruang parkir

yang tertutup. Sedangkan menurut Regina Bailey[3], batas normal yang ditolerir oleh tubuh manusia adalah 35 ppm CO yang terpapar selama 8 jam.

Salah satu bentuk pencegahan yang mungkin dapat dilakukan adalah membentuk sistem peringatan kepada masyarakat yang berada disekitar gedung parkir. Dalam hal ini, perancangan sistem berbasis FPGA dengan menggunakan sensor CO diharapkan mampu menjadi jawaban dari permasalahan di atas.

1.2 Tujuan Penelitian

Tujuan dari penelitian yang dilakukan ini adalah sebagai berikut:

1. Merancang dan membuat prototipe sistem pemantau polusi udara pada area parkir tertutup menggunakan FPGA Xilinx Spartan 3E dan sensor gas CO MQ7.
2. Menganalisis akurasi pembacaan kadar konsentrasi CO oleh MQ7.
3. Menganalisis kecepatan pembacaan kadar konsentrasi CO dari sensor oleh FPGA Xilinx Spartan 3E.

1.3 Batasan Penelitian

Penelitian dalam skripsi ini terbatas pada prototipe sistem *embedded* pemantau polusi udara pada area parkir tertutup. FPGA yang digunakan adalah Xilinx Spartan 3E (XC3S500E-4FG320C). Sensor yang digunakan adalah sensor gas CO MQ7, Pengujian dilakukan pada kotak simulasi berukuran 687 x 478 x 390 mm (82 liter).

1.4 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam pembuatan sistem ini mengacu pada *System Development Life Cycle* (SDLC)[4], antara lain:

1. Inisiasi ide

Menentukan tema yang akan dikembangkan berkaitan dengan isu masyarakat saat ini.

2. *System requirement*

Mendefinisikan sistem dan fitur yang digunakan.

3. Desain

Merubah kebutuhan-kebutuhan yang sudah dirangkum pada tahapan sebelumnya sehingga menjadi suatu sistem.

4. Pengembangan dan implementasi

Merubah desain menjadi bentuk sistem utuh.

5. Uji coba dan pemeliharaan sistem

Melakukan perubahan, penambahan, ataupun pengembangan sistem setelah dilakukan uji coba sistem.

1.5 **Sistematika Penulisan**

Dalam penulisan skripsi ini penulis menyajikan sistematika penulisan dengan uraian sebagai berikut:

BAB I PENDAHULUAN

Bab ini menjelaskan tentang latar belakang masalah, batasan masalah, tujuan penulisan, metode penulisan serta sistematika penulisan.

BAB II KONSEP FPGA XILINX SPARTAN 3E DAN SENSOR MQ7

Bab ini memaparkan tentang definisi udara, pencemaran udara, Field Programmable Gate Array beserta komponennya, VHDL, dan sensor yang digunakan, yaitu Hanwei MQ7.

BAB III PERANCANGAN SISTEM PEMANTAU POLUSI UDARA PADA AREA PARKIR TERTUTUP MENGGUNAKAN FPGA XILINX SPARTAN 3E DAN SENSOR GAS CO MQ7

Bab ini menjelaskan tentang user requirement, desain sistem yang meliputi tahapan perancangan dan rencana uji coba berdasarkan metode penelitian yang telah dipaparkan sebelumnya.

BAB IV IMPLEMENTASI SISTEM PEMANTAU POLUSI UDARA PADA AREA PARKIR TERTUTUP MENGGUNAKAN FPGA XILINX SPARTAN 3E DAN SENSOR GAS CO MQ7

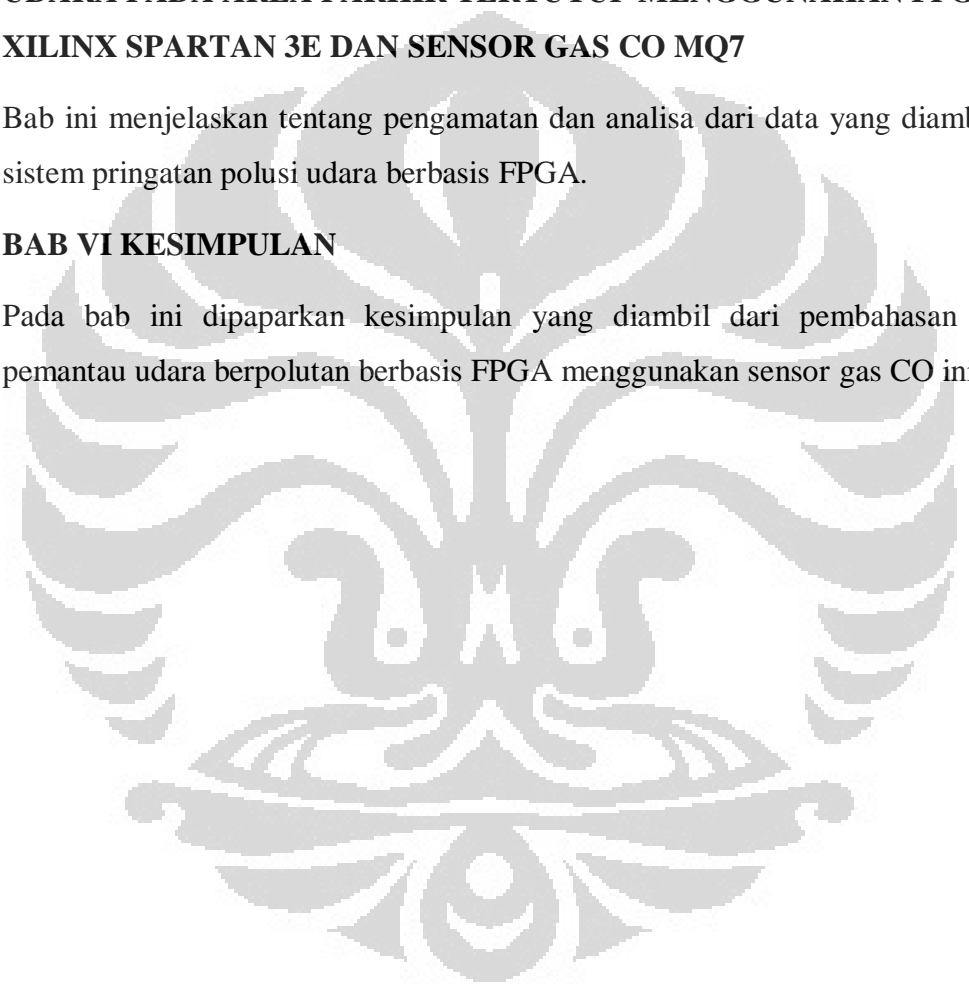
Bab ini menjelaskan tentang tahapan implementasi sistem yang dibagi menjadi dua tahapan, yaitu perangkat keras dan perangkat lunak.

BAB V PENGUJIAN DAN ANALISA SISTEM PEMANTAU POLUSI UDARA PADA AREA PARKIR TERTUTUP MENGGUNAKAN FPGA XILINX SPARTAN 3E DAN SENSOR GAS CO MQ7

Bab ini menjelaskan tentang pengamatan dan analisa dari data yang diambil dari sistem peringatan polusi udara berbasis FPGA.

BAB VI KESIMPULAN

Pada bab ini dipaparkan kesimpulan yang diambil dari pembahasan sistem pemantau udara berpolutan berbasis FPGA menggunakan sensor gas CO ini.



BAB II

KONSEP FPGA XILINX SPARTAN 3E DAN SENSOR MQ7

Pada bab ini akan dijelaskan teori yang melandasi sistem yang akan dibuat, yaitu terdiri dari teori tentang pencemaran udara, *Field Programmable Gate Array* (FPGA), sensor gas, dan perangkat lainnya yang berkaitan dengan sistem peringatan dan antar muka.

2.1 Pencemaran Udara

Berdasarkan KBBI, udara merujuk kepada campuran berbagai gas yang tidak berwarna dan tidak berbau (seperti oksigen dan nitrogen) yang memenuhi ruang di atas bumi seperti yang kita hirup apabila kita bernapas. [5]

Pencemaran udara adalah kehadiran satu atau lebih substansi fisik, kimia, atau biologi di atmosfer, sehingga mutu udara akan menurun hingga tingkat tertentu. Jika bahan pencemar terlalu banyak, maka dapat membahayakan kesehatan manusia, hewan, dan tumbuhan, mengganggu estetikadan kenyamanan, atau merusak properti. Pencemaran udara dapat disebabkan oleh beberapa faktor, yaitu faktor alam dan faktor eksternal (manusia).

2.1.1 Karbon monoksida (CO)

Karbon monoksida merupakan zat pembuangan emisi kendaraan bermotor yang cukup berbahaya. Dalam sub-bab ini akan dijelaskan mengenai sifat fisika dan kimia, sumber dan distribusi, dampak terhadap kesehatan, dan penanggulangan yang dapat dilakukan untuk mengurangi dampak buruk karbon monoksida terhadap tubuh manusia.

1. Sifat Fisika dan Kimia



Gambar 2.1. Ikatan senyawa karbon monoksida (CO) [6]

Karbon monoksida (CO), yang struktur atomnya ditunjukkan oleh Gambar 2.1, merupakan hasil pembakaran yang tidak sempurna. Senyawa ini tidak berbau, tidak berasa dan pada suhu udara normal berbentuk gas yang tidak berwarna dan mempunyai potensi bersifat racun yang berbahaya karena mampu membentuk ikatan yang kuat dengan pigmen darah yaitu hemoglobin.

2. Sumber dan Distribusi

CO di lingkungan dapat terbentuk secara ilmiah, tetapi sumber utamanya adalah dari kegiatan manusia. Contoh sumber CO yang berasal dari kegiatan manusia, antara lain, kendaraan bermotor. Berdasarkan estimasi[7], jumlah CO dari sumber buatan diperkirakan mendekati 60 juta ton per tahun. Separuh dari jumlah ini berasal dari kendaraan bermotor yang menggunakan bahan bakar bensin.

Pada beberapa kasus, terdapat juga kelompok masyarakat yang berpotensi keracunan CO. Kelompok masyarakat yang paling berpotensi terpapar CO termasuk penjaga parkir, polisi lalu lintas, pekerja bengkel mobil, petugas industri logam, industri bahan bakar bensin, industri gas kimia, dan pemadam kebakaran. Pemaparan CO dari lingkungan kerja perlu mendapat perhatian. Secara pribadi masyarakat yang bergelut pada bidang-bidang tersebut di atas, misalkan, penjaga parkir harus mempunyai kemampuan untuk mendeteksi adanya perubahan kecil dalam lingkungannya yang terjadi pada saat yang tidak dapat diperkirakan sebelumnya dan membutuhkan kewaspadaan tinggi dan terus menerus.

Perlu diketahui bahwa manusia sendiri dapat memproduksi CO akibat proses metabolismenya yang normal. Produksi CO didalam tubuh manusia (endogenous) berkisar 0,1+1% dari total HbCO dalam darah.

1. Dampak Terhadap Kesehatan

Tabel 2.1. Pengaruh Kadar CO dalam darah terhadap kondisi tubuh. [8]

CO (ppm)	Lama Paparan	Gejala
35	8 jam	Taraf yang masih diperbolehkan di dalam lingkungan kerja dalam kurun waktu 8 jam kerja per hari.
200	2 – 3 jam	Sakit kepala ringan, rasa lelah, rasa mual, disorientasi.
400	1 – 2 jam	Sakit kepala berat, ancaman kematian setelah 3 jam.
800	45 menit	Pusing hebat, mual, kejang. Kemungkinan tidak sadarkan diri selama 2 jam. Kematian dalam 2-3 jam.
1600	20 menit	Sakit kepala hebat, mual, mengakibatkan kematian dalam 1 jam.
3200	5 – 10 menit	Sakit kepala hebat, mual, mengakibatkan kematian dalam 1 jam.
6400	1 – 2 menit	Sakit kepala hebat, mual, mengakibatkan kematian dalam 25-30 menit.
12.800	1 – 3 menit	Kematian.

Karakteristik yang paling penting dari CO adalah kemampuannya untuk berikatan dengan hemoglobin, pigmen sel darah merah yang mengangkut oksigen (O_2) keseluruh tubuh. Sifat ini menghasilkan pembentukan carboxyhemoglobin (HbCO) yang 200x lebih stabil dibandingkan oksihemoglobin (HbO_2). Penguraian HbCO yang relatif lambat menyebabkan terhambatnya kerja molekul sel pigmen tersebut dalam fungsinya membawa O_2 keseluruh tubuh. Kondisi seperti ini bisa berakibat serius, bahkan fatal, karena dapat menyebabkan keracunan. Selain itu, metabolisme otot dan fungsi enzim intra-seluler juga dapat terganggu dengan adanya ikatan CO yang stabil tersebut[8]. Secara lebih lengkap, pengaruh kadar CO dalam darah terdapat pada Tabel 2.1.

Dampak keracunan CO sangat berbahaya bagi orang yang telah menderita gangguan pada otot jantung atau sirkulasi darah perifer yang parah. Dampaknya pun bervariasi tergantung dari status kesehatan seseorang pada saat terpapar. Pada beberapa orang yang berbadan gemuk dapat mentolerir paparan CO sampai kadar HbCO dalam darahnya mencapai 40% dalam waktu singkat. Tetapi seseorang

yang menderita sakit jantung atau paru-paru akan menjadi lebih parah apabila kadar HbCO dalam darahnya sebesar 5–10% [8].

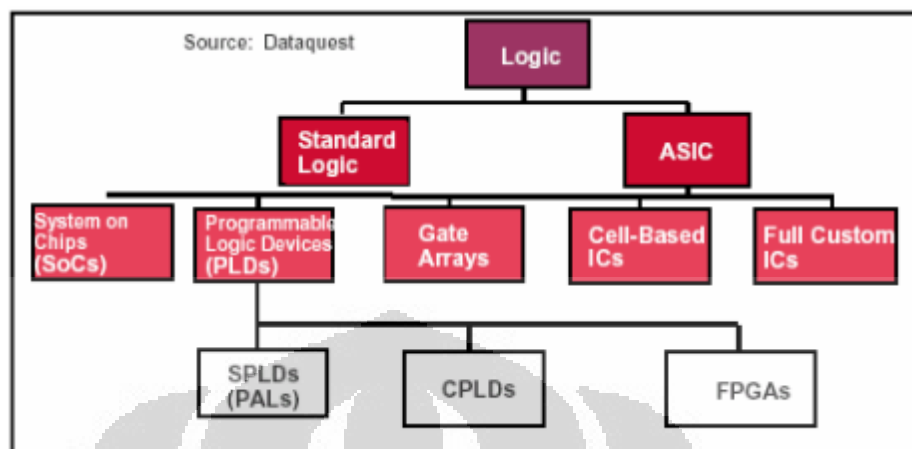
Meski telah banyak penelitian yang menjelaskan mengenai pengaruh karbon monoksida kadar tinggi terhadap sistem syaraf pusat dan sistem kardiovaskular. Namun respon dari masyarakat berbadan sehat terhadap pemaparan karbon monoksida kadar rendah dan dalam jangka waktu panjang, masih sedikit diketahui. Misalnya kinerja para petugas jaga parkir dapat terganggu atau terhambat pada kadar HbCO yang berada dibawah 10% dan bahkan sampai 5% (hal ini secara kasar ekivalen dengan kadar karbon monoksida di udara masing-masing sebesar 80 dan 35 mg/m³) [8].

3. Penanggulangan

Setelah dipaparkan mengenai dampak CO terhadap kesehatan, di bawah ini adalah bentuk-bentuk penanggulangan dari pencemaran CO: [8]

- Mengatur pertukaran udara di dalam ruangan, seperti menggunakan exhaust-fan.
- Bila terjadi korban keracunan, berikan pernapasan buatan dan kirim segera ke rumah sakit terdekat.
- Memasang detektor karbon monoksida. Detector akan mendeteksi keberadaan gas karbon monoksida pada tingkat tertentu dalam sebuah ruangan atau kendaraan dan memberi peringatan berupa suara alarm.

2.2 Field Programmable Gate Array (FPGA)



Gambar 2.2. Korelasi FPGA dan IC logic lainnya[9].

FPGA merupakan singkatan dari *Field Programmable Gate Array*. Posisi FPGA di antara IC logic lainnya ditunjukkan pada Gambar 2.2.

Field Programmable di dalam FPGA dapat diartikan sebagai interkoneksi dari blok logika di dalam FPGA yang dapat diprogram kembali oleh *user* atau pengguna sesuai dengan keinginan dan kebutuhan yang diinginkan. Sedangkan gate array dapat diartikan bahwa FPGA ini terdiri atas gerbang-gerbang logika, dimana masing-masing dari gerbang logika tersebut dapat dikonfigurasi antara satu sama lainnya.

Bila dilihat dari segi bentuknya FPGA tidak berbeda jauh dengan bentuk IC-IC pada umumnya. Hanya saja jika dilihat dari isi yang terdapat dalam FPGA, FPGA memiliki bagian yang berbeda dengan komponen IC pada umumnya. Setiap chip FPGA dapat memiliki puluhan hingga ratusan ribu sel logika. Masing-masing sel logika tersebut mempunyai keluaran dan masukan yang dapat dikonfigurasi sesuai dengan kebutuhan.

FPGA berbeda dengan mikroprosesor dan mikrokontroler, pada FPGA mempunyai struktur dari perangkat keras yang dapat dirubah konfigurasinya. Contohnya pada banyaknya I/O yang digunakan, atau jumlah register dapat dikonfigurasi sesuai dengan yang dibutuhkan oleh si pemakai. Meskipun mikrokontroler dapat diprogram ulang, namun mikrokontroler hanya dapat melakukan sesuatu berdasarkan konfigurasi dari perangkat keras yang tetap. Pada

FPGA rancangan perangkat keras dapat diserahkan sepenuhnya ke *user*. Sebelum diprogram, FPGA hanya merupakan sekumpulan blok-blok logika yang belum dikonfigurasi ataupun dihubungkan satu sama lainnya. Karena sebenarnya FPGA ini terdiri dari blok-blok yang belum disusun ataupun dikonfigurasi dan dalam perancangannya *user* diharuskan dapat mengkonfigurasi blok-blok ini hingga menghasilkan konfigurasi blok yang sesuai, maka sebenarnya kata yang lebih tepat seharusnya adalah mengkonfigurasi FPGA bukan memrogramnya.

FPGA memiliki beberapa kelebihan dibandingkan dengan chip yang lain, diantaranya:

- Kemampuannya untuk dikonfigurasi secara manual oleh *user* disesuaikan dengan kebutuhan.
- Dapat diterapkan pada berbagai rangkaian gerbang logika, baik yang sederhana maupun yang kompleks.
- Dapat didesain ulang untuk mengubah fungsi logika tanpa merubah arsitektur sistem.
- *Instant manufacturing* sehingga merupakan prototype yang tergolong lebih murah.
- Komputasi dapat dilakukan secara paralel dan memungkinkan siklus desain logika yang bekerja secara paralel[10].

Produsen FPGA, antara lain Xilinx, Alcatel, Lattice, Actel, dan SiliconBlue. Dalam pengerjaan skripsi ini, FPGA yang digunakan adalah Xilinx Spartan 3E yang akan dijelaskan pada sub-bab selanjutnya.

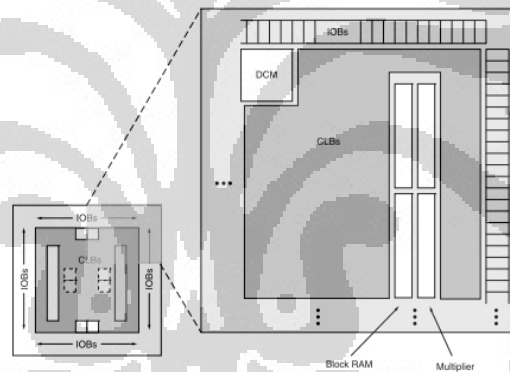
2.2.1 FPGA Xilinx Spartan 3S500E

Spartan 3S500E, ditunjukkan pada Gambar 2.3, merupakan salah satu keluarga FPGA yang diproduksi oleh Xilinx. FPGA ini dapat dipogram dengan VHDL pada *software* Xilinx Development Kit. Selama pengerjaan skripsi ini, penulis menggunakan Xilinx Integrated Software Environment (ISE) Design Suite 13.2. Perangkat lunak ini dapat mensintesis kode VHDL.



Gambar 2.3. FPGA Spartan 3E[11]

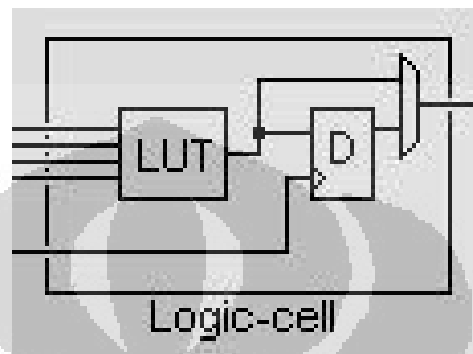
Secara umum, arsitektur chip FPGA Xilinx Spartan 3E terlihat pada Gambar 2.4, terdiri dari:



Gambar 2.4. Arsitektur FPGA 3E[12]

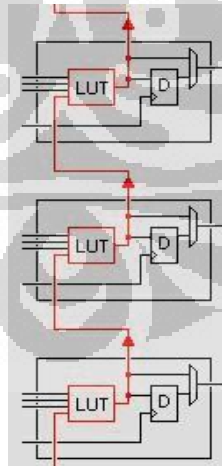
1. *Configurable Logic Blocks (CLB)* yang terdiri dari *Look-Up Tables (LUTs)*. CLB melakukan fungsi logika dan berfungsi pula sebagai penyimpanan data. *Lookup table (LUT)* merupakan sejenis memory yang berkapasitas kecil di dalam *logic-cell*. *Lookup table* ini memiliki peranan penting dalam proses implementasi fungsi-fungsi logika. Pada awalnya lookup table memiliki 4 buah inputan dan akan mengkonversi inputannya tersebut menjadi keluaran berdasarkan fungsi logikanya. Namun akhir-akhir ini perusahaan yang memproduksi FPGA telah menggantinya dengan 6 input LUT untuk dapat meningkatkan performanya. D Flip-flop pada *logic-cell* berfungsi sebagai rangkaian logika sekuensial dimana didalamnya terdapat rangkaian memori

dan pewaktu. Keluaran dari D flip-flop bergantung terhadap sinyal pewaktu yang masuk. Multiplexer 2 ke 1 pada *logic-cells* merupakan sebuah rangkaian digital yang berfungsi sebagai pemilih dari 2 masukan pada multiplexer menjadi sebuah keluaran saja. *Logic-cells* yang dipaparkan dapat digambarkan seperti yang terdapat pada Gambar 2.5.



Gambar 2.5. *Logic-cell*[12]

Untuk tujuan jalur interkoneksi antar *cell*, FPGA memiliki jalur penghubung singkat diantara *cell-cell* yang berdekatan. Tipe jalur interkoneksi ini sering disebut *carry chains*. *Carry chains* berfungsi dalam pembuatan fungsi-fungsi logika agar dapat berjalan dengan efisien. *Carry chain* ditunjukkan pada Gambar 2.6.



Gambar 2.6. *Logic-cell* yang terhubung melalui *carry chain*[12]

2. *Input/Output Blocks* mengontrol aliran data antara pin I/O dan logika internal dalam FPGA.
3. *Block RAM* merupakan tempat penyimpanan data dual-port 18 Kbit blok.

4. *Multiplier* RAM melakukan tugas utama yaitu perkalian numerik *two's complement* tapi bisa juga melakukan beberapa aplikasi seperti menyimpan data dan *barrel shifting*.
5. *Digital Clock Manager* (DCM) memberikan fleksibilitas kelengkapan kontrol clock sinyal, antara lain distribusi, delay, mengkali, membagi, *phase shift* dan *skew*.

FPGA Xilinx Spartan 3E, sama seperti FPGA pada umumnya, dapat terbagi atas dua kondisi yaitu kondisi awal (*configuration mode*) dan kondisi pemakai (*user mode*). Ketika pertama kali FPGA dihidupkan, maka otomatis keadaan FPGA ini berada dalam kondisi awal. Hal ini disebabkan karena FPGA masih dalam keadaan awal dan belum terdapat suatu program apapun di dalamnya. Untuk menggunakan FPGA tersebut maka perlu adanya proses *download* program oleh *user*. Proses *men-download* program ke dalam FPGA bertujuan untuk mengirimkan berkas-berkas bilangan biner (0 dan 1) melalui beberapa pin khusus. Setelah proses *download* dilakukan dan FPGA kemudian telah siap digunakan, maka inilah yang sering disebut kondisi pemakai. Pada kondisi ini FPGA telah aktif dan program yang di-*download* ke dalamnya dapat digunakan. Inilah yang dinamakan konfigurasi atau pengaturan pada FPGA.

Dalam melakukan konfigurasi pada FPGA dapat dilakukan melalui urutan seperti berikut :

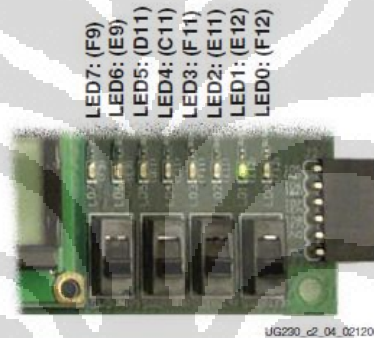
1. Menggunakan komputer untuk mendeskripsikan fungsi logika yang diinginkan. Dalam mendeskripsikan fungsi logika ke dalam FPGA dapat dilakukan dengan dua metode yaitu pertama dengan menggambar skematik diagramnya atau yang kedua dapat dilakukan dengan menggunakan Bahasa Deskripsi Perangkat keras (VHDL/Verilog).
2. Meng-*compile* fungsi logika dengan menggunakan *software* yang telah disediakan oleh vendor FPGA, kemudian membuat file biner yang dapat di-*download* ke dalam FPGA
3. Menghubungkan kabel dari komputer ke FPGA dan *men-download* file biner tersebut ke dalam FPGA.

2.2.2 Komponen Board Starter Kit Spartan-3E

Board yang digunakan dalam penelitian ini adalah Xilinx FPGA Spartan-3E (XC3S500E-4FG320C) yang memiliki 232-pin I/O, 320-pin paket FPGA dan lebih dari 10.000 sel logika. Dalam implementasi digunakan pula Analog Capture Circuit yang akan dijelaskan pada sub-bab ini.

- LED

LED yang terdapat di FPGA Xilinx Spartan 3E berjumlah 8 buah ditunjukkan pada Gambar 2.7. Konstrain 6-pin *accessory header* didefinisikan di file UCF meliputi koneksi I/O, *slew-rate* dan arus keluaran ditunjukkan pada Gambar 2.8.



Gambar 2.7. 8 LED[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 2.8. UCF LED[11]

- *Clock Sources*

Spartan-3E Starter Kit board memiliki 3 sumber daya clock, dimana setiap input clock (ditunjukkan pada gambar 2.9.) terhubung dengan *Digital Clock Manager*.

- *50 MHz On-Board Oscillator*

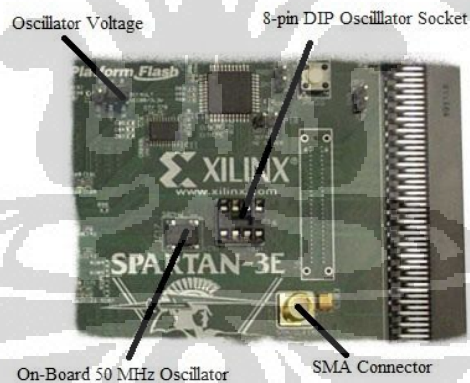
Board ini memiliki 50 MHz osilator dengan 40% - 60% *duty cycle*. Osilator ini mampu bekerja secara akurat sampai dengan ± 2500 Hz atau ± 50 ppm.

- *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 atau Output Connector*

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



Gambar 2.9. Clock input[11]

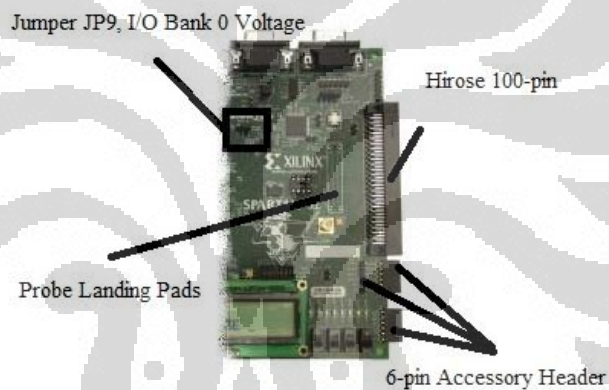
Konfigurasi konstrain UCF *clock source* ditunjukkan pada Gambar 2.10.

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

Gambar 2.10. UCF *clock sources*[11]

- Konektor Ekspansi

Spartan-3E Starter Kit board menyediakan berbagai macam konektor ekspansi untuk *interface*, ditunjukkan pada Gambar 2.11., yang mudah untuk komponen tambahan dari luar board.



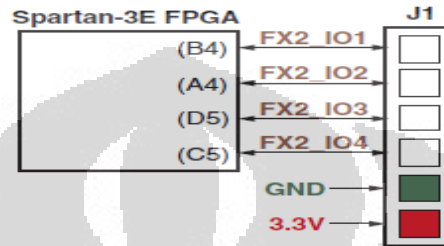
Gambar 2.11. Konektor ekspansi[11]

Secara default konektor ekspansi ini memberikan suplai 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:

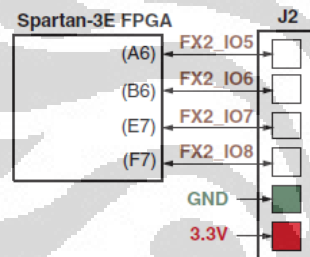
1. Hirose 100-pin edge
2. 3 buah 6-pin peripheral modul connections
3. Probe landing pads

- 6-Pin Accessory Header

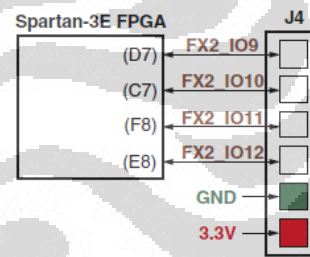
Terdiri dari J1 (ditunjukkan pada Gambar 2.12), J2 (ditunjukkan pada Gambar 2.13) dan J4 header (ditunjukkan pada Gambar 2.14), dimana J1 dan J2 header adalah female 6-pin socket dan J4 adalah male 6-pin socket. 4-pin paling atas sebagai I/O dan 2-pin terakhir sebagai Ground dan *power supply* sebesar 3.3V.



Gambar 2.12. Mapping J1[11]



Gambar 2.13. Mapping J2[11]



Gambar 2.14. Mapping J4[11]

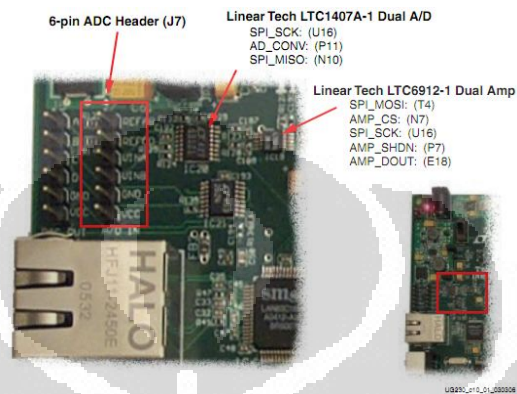
Konstrain 6-pin accessory header didefinisikan di file UCF meliputi koneksi I/O, slew-rate dan arus keluaran, sebagai pada Gambar 2.15.

```
# ==== 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 6-pin accessory header[11]

- *Analog Capture Circuit*

Analog Capture Circuit yang ada dalam FPGA Xilinx Spartan 3E terdiri dari 2 *channel* memiliki programmable scalling pre-amplifier (LTC6912-1) dan dual ADC (LTC1407A-1). Rangkaianannya ditunjukkan pada Gambar 2.16. Port perantara dengan dunia luar adalah melalui port J7 yang memiliki 6 pin.



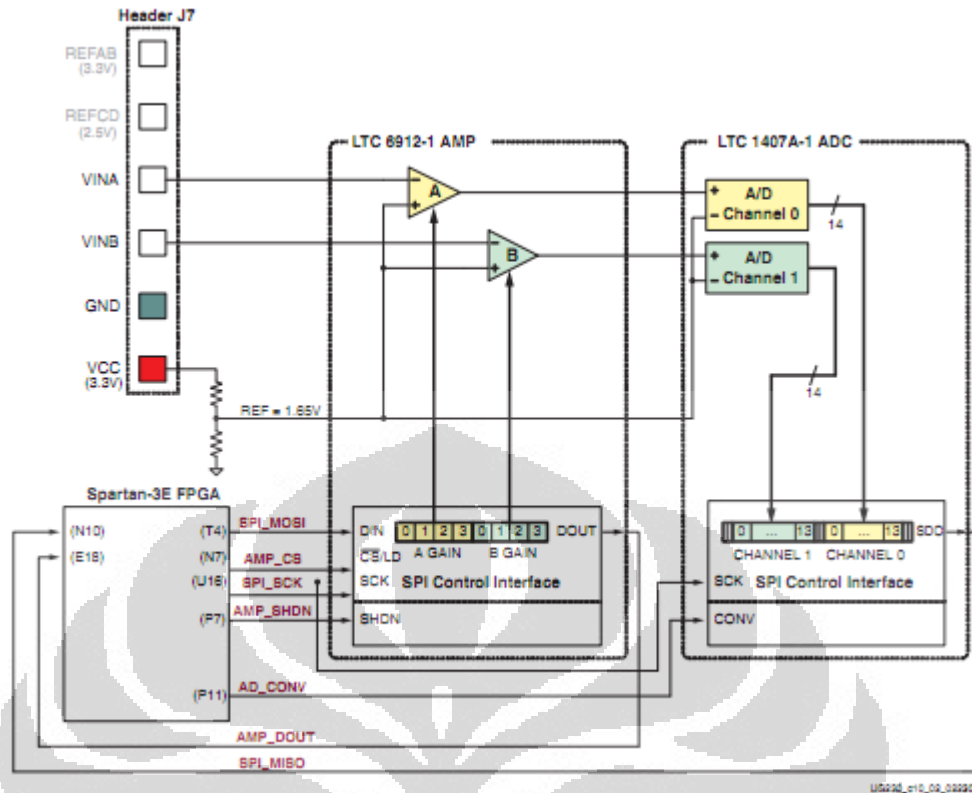
Gambar 2.16. Analog Capture Circuit [11]

Pada Gambar 2.17 ditunjukkan bahwa rangkaian ini memiliki dua *inverting amplifier* yang dapat berdiri sendiri. Tegangan referensi *amplifier* adalah 1,65 V yang dihasilkan oleh *voltage divider*. ADC kemudian akan mengambil *sample* sinyal secara simultan segera ketika AC_CONV aktif.

Output yang diberikan berupa 14 bit (D[13:0]) two's complement dihitung dengan menggunakan Persamaan 2.1.

$$D[13:0] = GAIN \times \frac{V_{in} - 1,65 V}{1,25 V} \times 8192 \quad (2.1)$$

Gain dapat diprogram melalui *programmable pre-amplifier*. Gain bernilai antara -1 hingga -100 (op-amp negatif) seperti yang ditunjukkan pada Tabel 2.2. Sedangkan, jangkauan maksimum ADC adalah $\pm 1,25V$.



Gambar 2.17. Detail Analog Capture Circuit [11]

Tabel 2.2. Pengaturan gain untuk pre-amplifier[11]

Gain	A3	A2	A1	A0	Input Voltage Range	
	B3	B2	B1	B0	Minimum	Maksimum
0	0	0	0	0		
-1	0	0	0	1	0,4	2,9
-2	0	0	1	0	1,025	2,275
-5	0	0	1	1	1,4	1,9
-10	0	1	0	0	1,525	1,775
-20	0	1	0	1	1,5875	1,7125
-50	0	1	1	0	1,625	1,675
-100	0	1	1	1	1,6375	1,6625

2.3 VHDL

VHDL (*VHSIC Hardware Description Language*) merupakan sebuah bahasa pemrograman dari VHSIC (*Very High Speed Integrated Circuit*) yang dikembangkan pertama kali oleh IEEE (*Institute Electrical and Electronic*

Universitas Indonesia

Engineering) pada tahun 1987, digunakan untuk merancang dan memodelkan sebuah rangkaian digital. VHDL merupakan kode yang sering digunakan pada FPGA untuk melakukan konfigurasi ke dalam FPGA. Selain menggunakan kode VHDL dalam mengkonfigurasi FPGA, Verilog juga dapat digunakan untuk melakukan konfigurasi di dalam FPGA.

Perbedaan mendasar dari VHDL dengan Verilog adalah mengenai konteks dari kedua bahasa itu sendiri. Verilog berasal dari tradisi “bottom-up” desain yang telah sering digunakan dalam industri IC dalam hal rancangan dasar IC. Sedangkan kode VHDL dikembangkan lebih kepada persepektif “top-down”.

2.3.1 Cara Penulisan dalam VHDL

Pendeskripsian VHDL terdiri dari dua bagian, yaitu *entity* dan arsitektur. Bagian penting lainnya antara lain, *library*, sinyal, variabel, konstan, *array*, fungsi, dan prosedur.

Library merupakan kumpulan modul-modul perintah yang digunakan dalam program. Sebelum digunakan di dalam program, modul yang berada di dalam *library* perlu diinisialisasi terlebih dahulu.

```
entity entity_name is
  [generics]
  [ports] [declarations {constants, types, signals};]
  [begin statement]
entity [entity] entity_name;
```

Gambar 2.18. Penulisan *entity* pada VHDL [9]

Entity mendeskripsikan antar muka atau pendeskripsian *port*. Pendeskripsian *port* ini mendefinisikan jenis dari koneksi dan arah yang sesuai. Gambar 2.18 adalah format penulisan *entity*.

Contoh *entity* gerbang and ditunjukkan pada Gambar 2.19.

```
entity andGate is
  port {
    a, b: in std_logic;
    c: out std_logic
  };
entity andGate;
```

Gambar 2.19. Contoh *entity* pada gerbang and [9]

```
architecture architecture_name of entity_name is
  [architecture_declarative_part]
begin
  [architecture_statement_part]
end [architecture] [architecture_name];
```

Gambar 2.20. Penulisan *architecture* pada VHDL [9]

Arsitektur adalah mendeskripsikan implementasi rangkaian yang akan dibuat. Penulisan arsitektur ditunjukkan pada Gambar 2.20.

2.3.2 Pendeskripsian VHDL

Pendeskripsian rangkaian dapat dilakukan dengan menggunakan metode *behavioral*, *structural*, dan *data flow*.

- Behavioral

Arsitektur dapat didesain sesuai dengan prinsip kerja alat. Kunci dalam desain *behavior* adalah proses. Proses adalah kumpulan dari *sequential statement* yang dipasang paralel dengan *statement* yang sama dan proses yang lainnya.

- Structural

Arsitektur struktural menghubungkan modul-modul yang ada pada sistem (FPGA). Ia mendefinisikan struktur interkoneksi yang tepat dari sinyal dan *entity*. Arsitektur ini sama dengan representasi skematik karena hubungan tiap komponen diperlihatkan.

- Data flow

Pada implementasi data flow ini, VHDL menggambarkan aliran data pada rangkaian menggunakan pendekatan bahasa transfer register (*Register*

Transfer Language). Arsitektur ini memperlihatkan bagaimana input dan output dari primitive component yang berada di dalam terhubung bersama.

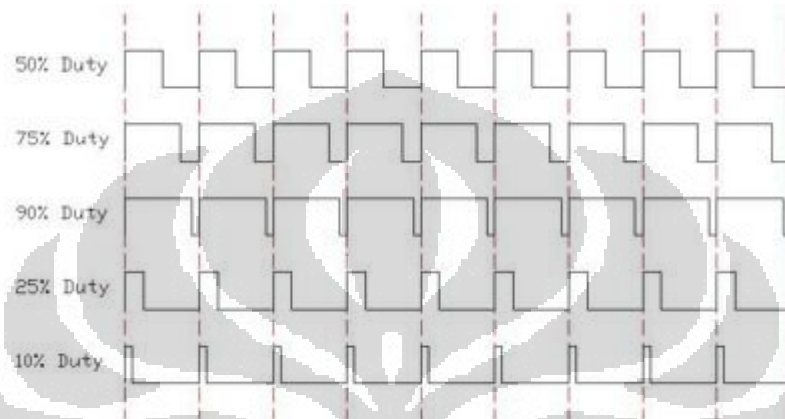
Bagian yang juga penting dalam arsitektur adalah proses dan konfigurasi. Proses adalah sub dari suatu arsitektur. Dalam suatu arsitektur bisa terdapat satu atau lebih proses yang saling *independent* ataupun *dependent*. Sedangkan konfigurasi digunakan untuk mendeklarasikan kumpulan desain *entity* tertentu sebagai komponen instan,

- Sinyal
Sinyal dapat dianalogikan sebagai kabel yang menjadi penghubung antar bagian dalam sistem yang dirancang. Sinyal dideklarasikan dalam arsitektur. Nilai logika
- Variabel
Variabel memiliki nilai yang langsung berubah, tanpa harus menunggu selesainya suatu proses. Variabel dideklarasikan dalam proses.
- Konstan
Konstan memiliki nilai tetap.
- Array
Array yang merupakan sekumpulan elemen yang mempunyai tipe yang sama dan setiap elemen yang mempunyai *array index* yang berbeda-beda.
- Fungsi
Fungsi dapat menerima suatu data masukan dengan tipe data tertentu dan menghasilkan keluaran yang sesuai dengan tipe data yang keluaran yang telah didefinisikan dalam spesifikasinya.
- Prosedur
Prosedur hampir sama dengan fungsi, namun prosedur tidak dapat mengembalikan nilai balikan. Prosedur dalam VHDL harus menyertakan argumen dengan spesifik dengan tipe out atau inout.

2.4 Pulse-Width Modulation (PWM)

PWM merupakan salah satu jenis modulasi yang berkerja dengan cara mengatur lebar pulsa high dan low. PWM berkerja dengan membuat gelombang

persegi dengan perbandingan pulsa yang telah ditentukan oleh sinyal informasi hingga membentuk *duty cycle* yang berbeda berdasarkan referensi dari sinyal informasi. Tujuan dari penggunaan PWM itu sendiri adalah untuk memberikan referensi tegangan rata-rata yang berbeda yang bergantung terhadap perbandingan lebar pulsa high dan low. Sinyal PWM diharapkan agar tetap berada pada frekuensi yang konstan agar sinyal informasi yang diterima menjadi valid.



Gambar 2.21. *Duty cycle* [18]

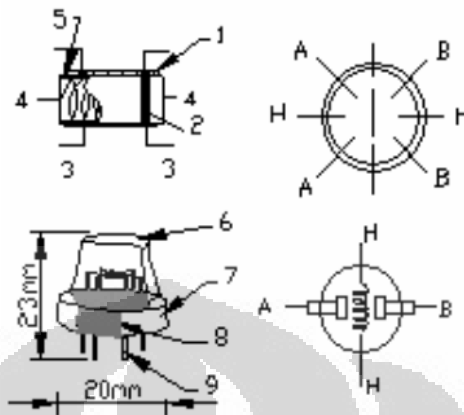
Pada gambar 2.21 terlihat variasi dari *duty cycle* sinyal PWM, besar dari *duty cycle* sinyal PWM merupakan besarnya perbandingan lama waktu nilai high terhadap nilai low dalam satu periode gelombang. *Duty cycle* biasanya ditentukan dalam bentuk persen, seperti terlihat pada gambar 2.11 nilai *duty cycle* 50% adalah 50% periode gelombang merupakan sinyal high dan 50% periode gelombang sinyal low. Untuk *duty cycle* 75% terlihat bahwa 75% periode gelombang adalah sinyal high dan 25% periode gelombang merupakan sinyal low. Dan begitu juga seterusnya, dimana nilai *duty cycle* menjadi persentase lama waktu sinyal high terhadap periode gelombangnya.

2.5 Sensor

2.5.1 Hanwei MQ7 CO Gas Sensor

Sensor buatan Hanwei, Electronics Co., LTD ini digunakan untuk mendeteksi keberadaan gas CO yang merupakan salah satu gas emisi dari kendaraan bermotor. Sensor ini terdiri dari keramik Al_2O_3 , lapisan tipis SnO_2 , elektroda serta *heater* yang digabungkan dalam suatu lapisan kerak yang terbuat

dari plastik dan *stainless*. Struktur dan konfigurasi MQ7 ditunjukkan pada Gambar 2.22.

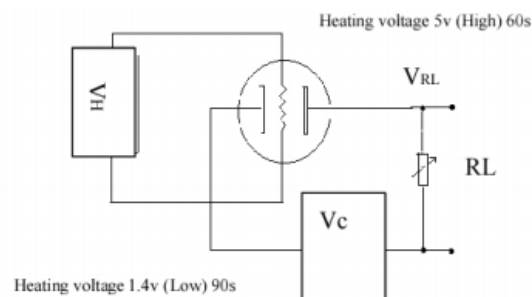


Gambar 2.22. Struktur MQ7[13]

Apabila terdeteksi gas CO, maka tegangan output pada sensor akan naik, sehingga konsentrasi gas akan menurun dan terjadi proses deoksidasi. Akibatnya permukaan dari muatan negatif oksigen akan berkurang, ketinggian permukaan sambungan penghalang pun akan ikut terjadi. Hal ini mengakibatkan penurunan resistansi sensor yang juga memiliki sebuah *heater* yang berfungsi sebagai pembersih dari kontaminasi udara di dalam ruangan sensor. Karakteristik output sensor bila mendeteksi keberadaan CO, yaitu output tegangan semakin besar sesuai dengan besarnya kadar ppm. Sensor ini menghasilkan sinyal analog.

Sistem yang dikembangkan ini menggunakan rangkaian untuk *interfacing* sesuai dengan Gambar 2.23. Nilai RL yang digunakan adalah 10 k Ω . Vc digunakan tegangan DC sebesar 5 Volt[13].

Semakin banyak asap, maka resistansi semakin menurun dan nilai VRL semakin membesar. Tabel karakteristik sensor terdapat pada Tabel 2.3.

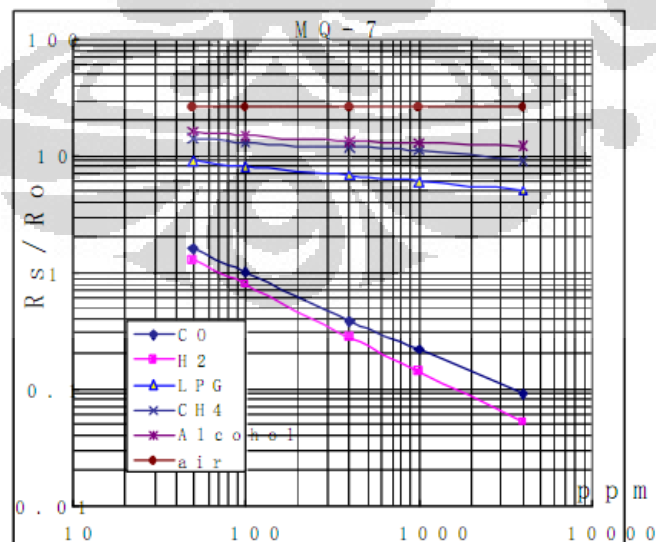


Gambar 2.23. Rangkaian sensor MQ7[13]

Tabel 2.3. Tabel Karakteristik Sensitivitas Sensor[13]

Simbol	Parameter	Parameter Teknis	Keterangan
R_s		2 – 20 K	Pada 100 ppm CO
s (300 / 100 ppm)		Kurang dari 0,5	R_s (300 ppm / 100 ppm)
Kondisi standar kerja sensor	Temperatur: $-20^{\circ}\text{C} \pm 2^{\circ}\text{C}$; Kelembaban relatif: $65\% \pm 5\%$;		
	RL: $10\text{K}\Omega \pm 5\%$; V_c : $5\text{V} \pm 0,1 \text{ V}$; V_H : $5\text{V} \pm 0,1 \text{ V}$; V_L : $1,4\text{V} \pm 0,1 \text{ V}$		
Preheat time	Tidak kurang dari 48 jam	Range deteksi kadar CO: 20 ppm – 2000 ppm	

Untuk mendapatkan ppm CO, hasil pembacaan berupa tegangan kemudian diubah untuk mendapatkan perbandingan antara resistansi sensor pada saat terdapat gas (R_s) dengan resistansi sensor pada saat udara bersih (R_o). Pada proses pendeteksian CO, nilai resistansi R_s pada sensor akan berkurang seiring dengan semakin besarnya konsentrasi CO yang terdeteksi. Sensor menggunakan V_C sebesar 5V. Berikut akan dijelaskan mengenai bagaimana cara mengkonversi tegangan output sensor V_{RL} ke satuan konsentrasi gas ppm (*part per million*). Persamaan 2.1. merupakan gambaran matematis hubungan antara resistansi permukaan R_s dan resistansi beban R_L berdasarkan grafik yang ditunjukkan pada Gambar 2.24.



Gambar 2.24. Grafik karakteristik sensitivitas sensor MQ7[13]

$$\frac{R_S}{R_L} = \frac{V_C - V_{RL}}{V_{RL}} = \text{rasio} \quad (2.1)$$

Perbandingan $\frac{R_S}{R_L}$ berbanding terbalik dengan nilai konsentrasi gas, maka

$$\frac{R_S}{R_L} = \frac{1}{\text{konsentrasi}} \quad (2.2)$$

$$\text{rasio} = \frac{1}{\text{konsentrasi}} \quad (2.3)$$

Diketahui saat $\frac{R_S}{R_0} = 0,9$ konsentrasi CO adalah 100 ppm.

$$0,9 = \frac{1}{100} \quad (2.4)$$

Perhitungan konversi ke satuan konsentrasi ppm dilakukan berdasarkan Persamaan 2.3 dan 2.4.

$$\frac{0,9}{\text{rasio}} = \frac{\frac{1}{100}}{\frac{1}{\text{konsentrasi}}} \quad (2.5)$$

$$\frac{0,9}{\text{rasio}} = \frac{1}{100} \times \frac{\text{konsentrasi}}{1} \quad (2.6)$$

$$\frac{0,9}{\text{rasio}} = \frac{\text{konsentrasi}}{100} \quad (2.7)$$

2.6 Perlengkapan Sistem Peringatan

2.6.1 Exhaust Fan

Exhaust fan memiliki prinsip kerja sebagai penghisap udara di dalam ruangan untuk dibuang ke luar, dan pada saat bersamaan menarik udara segar dari luar ke dalam ruangan. *Exhaust fan* membutuhkan daya listrik untuk dapat digunakan. Alat ini dapat membantu sirkulasi udara dalam ruangan sehingga bisa berjalan lebih cepat dan optimal[15]. Contoh *exhaust fan* ditunjukkan pada Gambar 2.25.

2.6.2 Buzzer

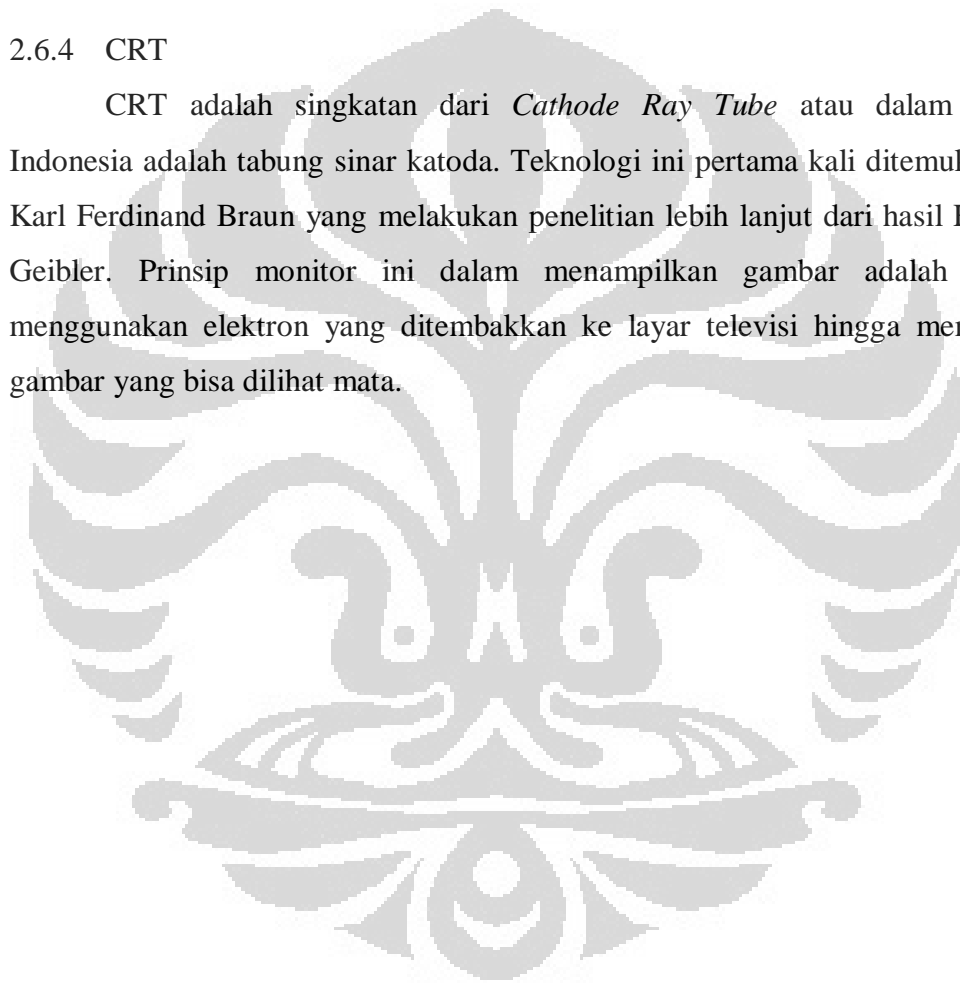
Buzzer atau beeper adalah perangkat sinyal audio. *Buzzer* dapat digunakan sebagai perangkat alarm, timer, dan *feedback* dari masukan *user*. *Buzzer* dapat berupa mekanik, elektromekanik, atau piezoelektrik. Contoh *buzzer* ditunjukkan pada Gambar 2.26.

2.6.3 LED

LED merupakan singkatan dari *Light Emitting System*. LED merupakan suatu semikonduktor yang memancarkan cahaya. Saat LED di forward bias, elektron dapat bergabung kembali dengan electron hole, sehingga LED akan melepaskan foton. Warna yang dihasilkan ditentukan oleh *energy gap* semikonduktor[16]. Contoh *exhaust fan* ditunjukkan pada Gambar 2.27.

2.6.4 CRT

CRT adalah singkatan dari *Cathode Ray Tube* atau dalam Bahasa Indonesia adalah tabung sinar katoda. Teknologi ini pertama kali ditemukan oleh Karl Ferdinand Braun yang melakukan penelitian lebih lanjut dari hasil Heinrich Geibler. Prinsip monitor ini dalam menampilkan gambar adalah dengan menggunakan elektron yang ditembakkan ke layar televisi hingga membentuk gambar yang bisa dilihat mata.



BAB III

PERANCANGAN SISTEM PEMANTAU POLUSI UDARA BERBASIS FPGA XILINX SPARTAN 3E MENGGUNAKAN SENSOR GAS CO MQ7

Berdasarkan *System Development Life Cycle* (SDLC), setelah menentukan tema dan batasan masalah serta tujuan, maka tahapan selanjutnya dalam pembuatan sistem pemantau polusi udara di gedung parkir berbasis FPGA dengan menggunakan sensor CO adalah perencanaan. Tahapan perencanaan ini meliputi *user requirement*, desain, dan implementasi. Pengujian sistem dan analisa tidak akan dibahas pada bab ini. Dalam mendokumentasikan setiap tahapan SDLC, *Unified Modeling Language* (UML) akan digunakan sebagai metode standar. Dengan UML, rancangan perangkat lunak dapat direpresentasikan ke dalam diagram-diagram yang memiliki fungsi masing-masing.

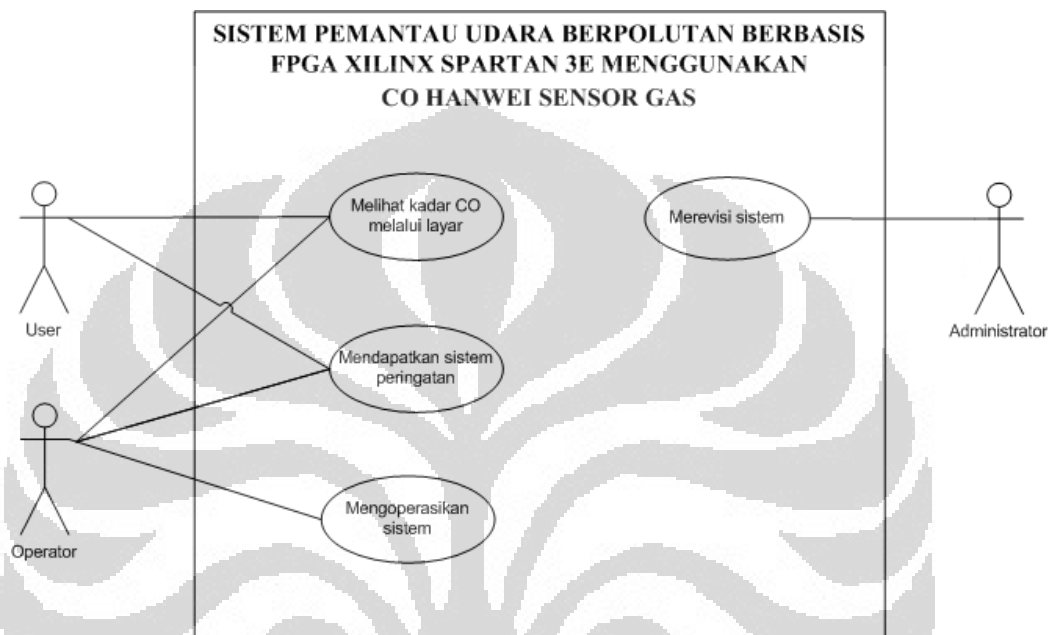
3.1 System Requirement

System requirement merupakan tahapan yang mendefinisikan sistem dan fitur yang dibutuhkan. Tahapan ini memegang peranan penting dalam perancangan fungsionalitas sistem. Terdapat 3 (dua) pengguna yang akan ada dalam sistem ini. Yang pertama yaitu *user*, dalam hal ini *user* adalah orang yang merasakan fasilitas sistem secara langsung. Yang kedua adalah operator, yaitu orang yang mengoperasikan sistem. Yang ketiga adalah administrator. Namun, secara umum, jika dikatakan *user* tanpa ada penjelasan lebih lanjut adalah semua orang yang terlibat dari mulai penyalaan sistem hingga hasil yang diperoleh melalui media *interface* dan sistem peringatan sistem.

Di bawah ini adalah hasil pengumpulan *requirement* yang didapat melalui tukar wawasan dengan beberapa *user*.

1. Operator dapat mengoperasikan sistem dengan mudah melalui Graphical User Interface yang ditampilkan di layar CRT.

2. *User* dapat melihat kadar CO secara *real time* melalui layar CRT secara interaktif. Dalam hal ini akan digunakan grafik dan warna.
3. *User* dapat mendapatkan sistem peringatan berupa LED, *buzzer*, dan *exhaust fan*.
4. Administrator dapat melakukan perubahan pada sistem.



Gambar 3.1. Use case diagram

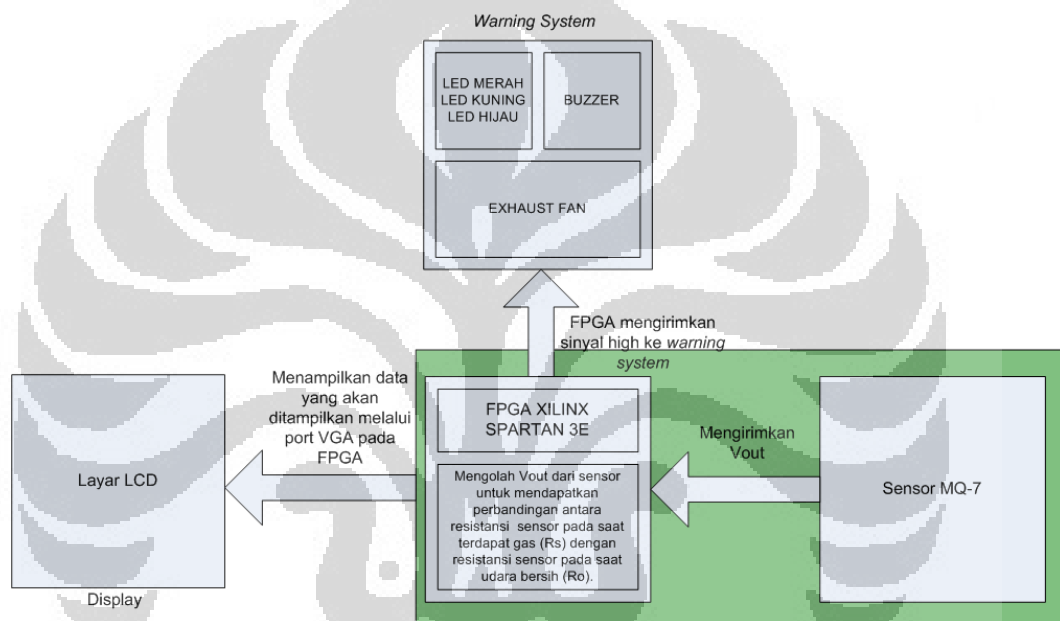
Fungsi-fungsi di atas merupakan fungsi-fungsi yang harus ada pada sistem ini guna memenuhi kebutuhan *user* yang akan menggunakannya. *Requirement* di atas merupakan *requirement* dasar yang masih dapat dikembangkan lagi sehingga menjadi sebuah sistem yang lebih baik. Gambar 3.1 menggambarkan fungsi yang sudah diolah dalam *use case diagram*.

3.2 Desain Sistem Pemantau Polusi Udara

Sistem pemantau polusi udara yang dibuat dalam penelitian ini dirancang untuk mengatasi permasalahan pertukaran udara yang terjadi di area parkir tertutup. Sistem ini akan dikendalikan oleh FPGA Xilinx Spartan 3S500E. Sedangkan, untuk peralatan sistem pemantau polusi udara lainnya adalah sensor gas Hanwei CO MQ7, LED, *buzzer*, *exhaust fan*, dan layar CRT.

Secara sederhana, sistem ini akan menampilkan hasil pemrosesan data kadar CO melalui layar CRT dan bekerja secara *real time*. Harapannya, masyarakat sekitar atau dalam hal ini pengguna area parkir akan mengetahui seberapa bahaya gas CO yang berada di area tersebut. Ditambah lagi dengan sistem peringatan yang komprehensif, yaitu berupa *buzzer*, LED, dan *exhaust fan*, yang secara efektif dapat menambah kewaspadaan pengguna area parkir.

Blok diagram dari perangkat keras yang digunakan pada sistem secara keseluruhan ditunjukkan pada Gambar 3.2.



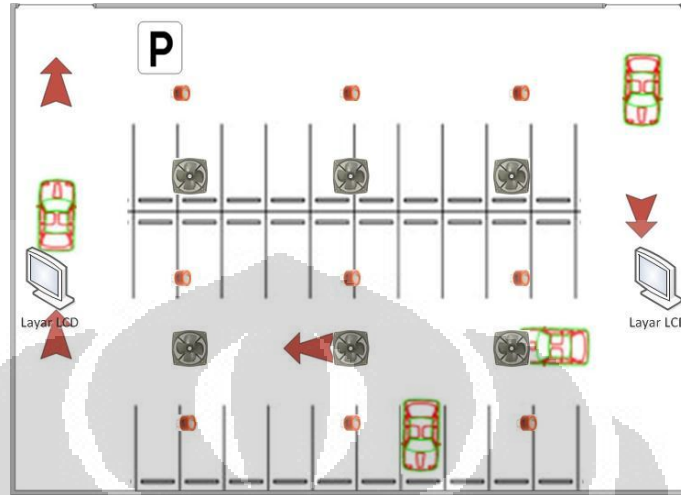
Gambar 3.2. Blok diagram sistem pemantau polusi udara berbasis FPGA

Sesuai dengan Gambar 3.2. sistem ini dibagi menjadi tiga bagian yang memiliki fungsi berbeda. Bagian-bagian tersebut, antara lain blok pengontrol, pembacaan dan pemrosesan data dari sensor gas CO, blok pemberian sistem peringatan, dan blok antar muka. Lingkup yang dibahas pada penelitian ini adalah sistem pengontrol, pembacaan dan pemrosesan data dari sensor gas CO. Pada gambar di atas, cakupan penelitian yang dilakukan ditunjukkan oleh bagian yang berwarna hijau.

3.2.1 Desain Perangkat Keras (*Hardware*)

Jangkauan sensor MQ7 sebesar 3 meter. Sehingga, dapat diasumsikan bahwa sensor MQ7 yang digunakan pada area seluas 35x19 meter sebanyak 12 buah. Dimana sensor ditempatkan berdampingan dengan LED, *buzzer*, *exhaust*

fan dan layar CRT. Adapun area parkir tertutup dapat diasumsikan seperti pada Gambar 3.3.



Gambar 3.3. Prototype area parkir

Alat yang digunakan pada sistem ini, antara lain FPGA Xilinx Spartan 3E, Hanwei MQ7 sensor gas CO, dan peralatan pembantu yaitu resistor 10 k Ω , kabel pelangi, dan konektor.

Sensor yang digunakan pada sistem ini adalah sensor gas CO. Ini merupakan satu-satunya sensor yang digunakan karena sistem ini masih merupakan sistem perintis. Alasan diambilnya sensor CO ini sebagai parameter polusi udara adalah karena CO merupakan gas yang mematikan yang paling banyak terkandung dalam emisi gas buang kendaraan bermotor. Untuk perkembangan selanjutnya, dapat ditambahkan beberapa sensor gas lain yang akan mendukung sistem untuk menentukan kadar polusi pada area parkir. Berdasarkan spesifikasi dari *datasheet* Hanwei, sensor gas MQ7 yang digunakan tidak dapat langsung dihubungkan dengan FPGA Xilinx Spartan 3E. Namun harus terlebih dahulu disambungkan dengan resistor 10 K Ω .

Untuk pengkonfigurasiannya FPGA, secara umum, program yang diimplementasikan akan akan membaca tegangan masukan dari sensor, mengambil data, memproses data, dan mengirimkan hasil proses ke antar muka *user*.

Program dimulai dengan melakukan inisialisasi I/O port dan inisialisasi *variable* pada FPGA. Setelah inisialisasi, program akan mulai menjalankan *powercycle* sensor dan menghitung besarnya Vout yang selanjutnya akan dikonversi menjadi nilai hambatan. Kemudian dari perbandingan nilai hambatan akan diperoleh besarnya nilai ppm. Nilai ppm ini akan menentukan besarnya nilai sinyal kontrol yang akan digunakan untuk mengatur besarnya input tegangan AC pada beban.

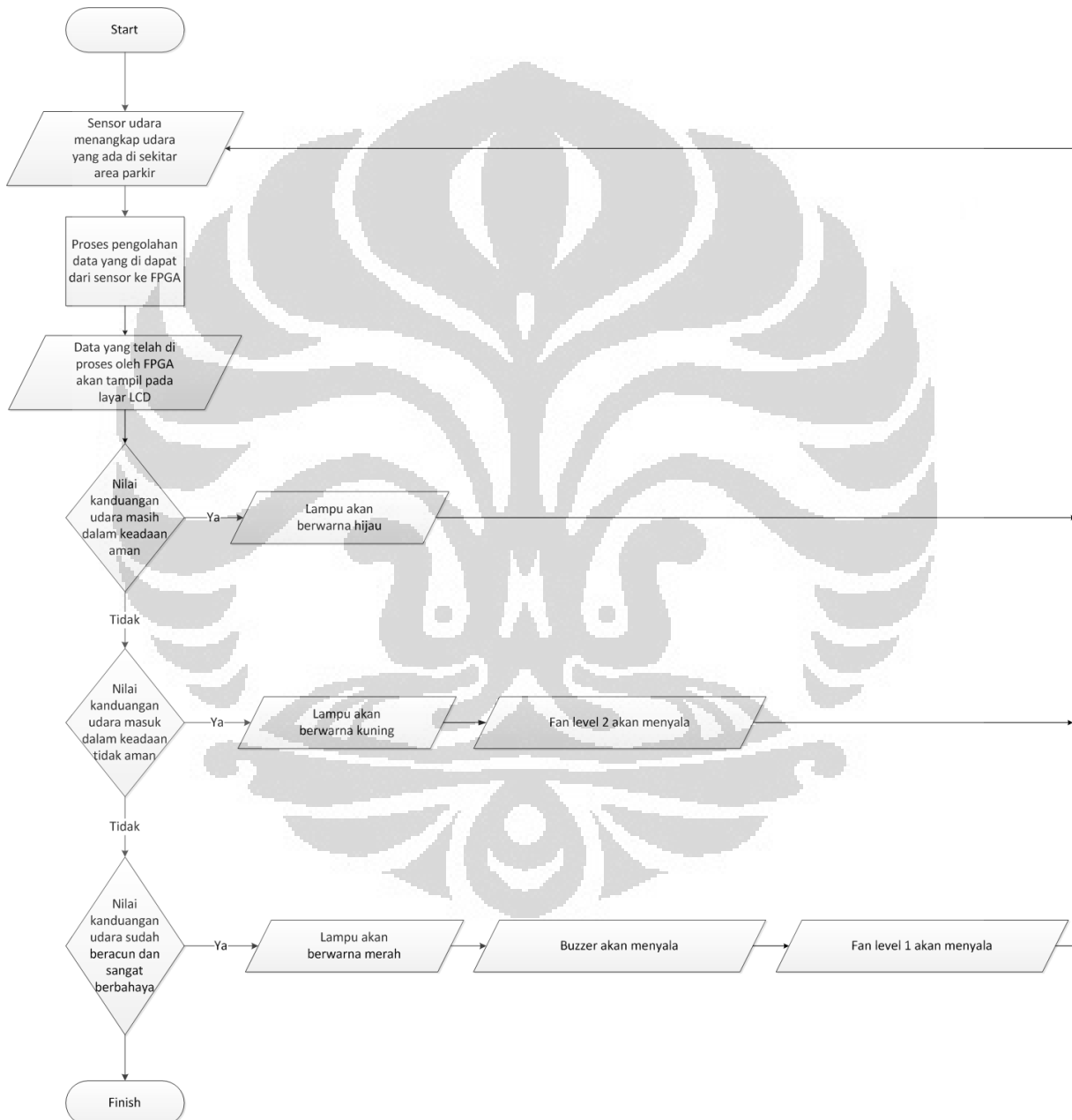
3.2.2 Algoritma Perangkat Lunak

Setelah membahas mengenai perancangan perangkat keras, maka tahapan selanjutnya adalah perancangan perangkat lunak atau dengan kata lain pembuatan program dengan menggunakan VHDL. Hal ini tidak kalah penting dibandingkan dengan tahapan perancangan sebelumnya karena FPGA yang digunakan tentunya harus dikonfigurasi terlebih dahulu. Untuk membantu perancangan ini, *flowchart* atau diagram alir dan UML akan digunakan. Secara garis besar, sistem akan digambarkan dengan menggunakan diagram alir. Untuk alur sistem secara terperinci akan digambarkan dengan menggunakan *sequence* diagram.

Berdasarkan Gambar 3.4., sistem pemantau polusi udara berbasis FPGA menggunakan sensor CO memiliki tahapan kerja sebagai berikut:

1. Pertama-tama, sistem dikalibrasi, disebut sebagai *purge phase*, dalam udara bersih selama tidak kurang dari 48 jam dengan tegangan sebesar 5V (sensor bekerja tiap 60 detik) sebelum pada akhirnya digunakan pada area parkir tertutup. Pengondisian udara bersih cukup dengan diletakkan di area dengan kadar CO kurang dari 35 ppm.
2. Setelah dikalibrasi, barulah sistem siap digunakan.
3. Sistem dinyalakan dengan menghubungkannya dengan catu daya AC dan menekan tombol On pada board FPGA. Hal ini dapat dilakukan oleh penjaga area parkir tersebut sebagai operator. Selbihnya sistem akan bekerja secara otomatis.
4. Sensor kemudian akan bekerja tiap 90 detik (menggunakan tegangan 1.4V, *sense phase*).

5. FPGA memproses data yang diambil dari sensor dengan menggunakan program yang dibuat dalam VHDL. Data yang diperoleh FPGA adalah berupa nilai tegangan sensor yang berupa analog. Oleh karena itu, nilai ini akan diproses dengan menggunakan ADC yang sudah ada pada FPGA yang akan digunakan. Operator tidak mengetahui program ini secara langsung karena program ini sudah diprogramkan ke dalam FPGA.



Gambar 3.4. Diagram alir sistem

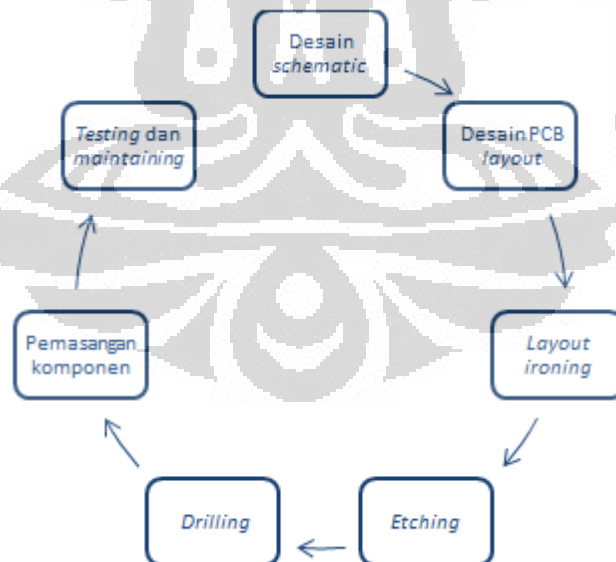
BAB IV

IMPLEMENTASI SISTEM PEMANTAU POLUSI UDARA BERBASIS FPGA XILINX SPARTAN 3E MENGGUNAKAN SENSOR GAS CO MQ7

Setelah selesai dengan tahapan analisa dan perancangan, maka tahapan berikutnya adalah implementasi. Implementasi merupakan penggabungan antara perangkat keras dan lunak pada sistem.

4.1 Implementasi Perangkat Keras (*Hardware*)

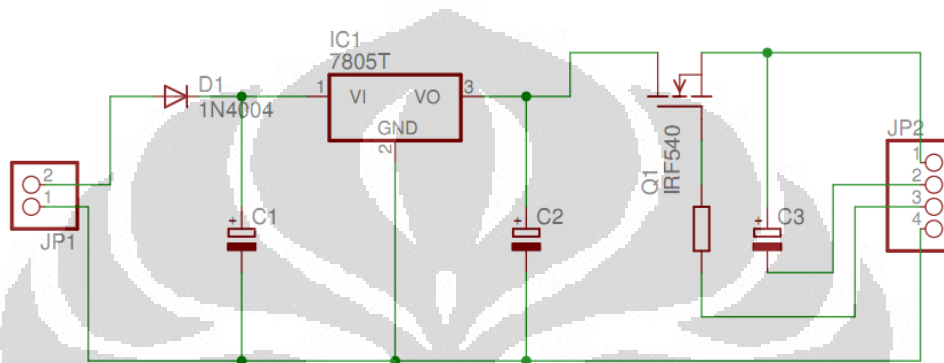
Desain perangkat keras merupakan perancangan dan pengembangan perangkat keras yang digunakan oleh sistem untuk dapat berfungsi sebagai mana mestinya sesuai dengan ketentuan yang telah ditetapkan sebelumnya pada *user requirement*. Tahap ini dilakukan berdasarkan *hardware life cycle* seperti yang ditunjukkan pada Gambar 4.1.



Gambar 4.1. Alur pengembangan perangkat keras

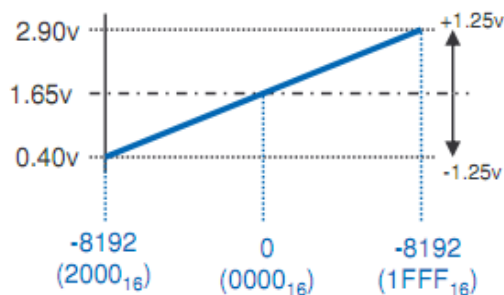
Sensor Gas CO Hanwei MQ7 menggunakan dua (2) *heating voltage*, yaitu sebesar 5V dan 1,4V yang dilakukan selama masing-masing 60 detik dan 90 detik. Dalam pengaturannya diperlukan rangkaian sebagai antar muka dengan FPGA

yang digunakan. Rangkaian yang dimaksudkan ditunjukkan pada Gambar 4.2. adalah untuk dapat memberikan tegangan 5V dan 1,4V secara bergantian dan terus menerus. Rangkaian akan dihubungkan dengan *power supply* pada JP1. Pada JP2, pin 1 adalah output yang akan dihubungkan dengan sensor, pin 2 adalah ground FPGA, pin 3 adalah PWM yang diatur oleh FPGA melalui 1 pin header pada konektor ekspansi J4 (E8), dan pin 4 adalah ground *power supply*. Penentuan port pada FPGA dilakukan melalui UCF dalam file *sensorucf.ucf*.



Gambar 4.2. Rangkaian antar muka sensor - FPGA

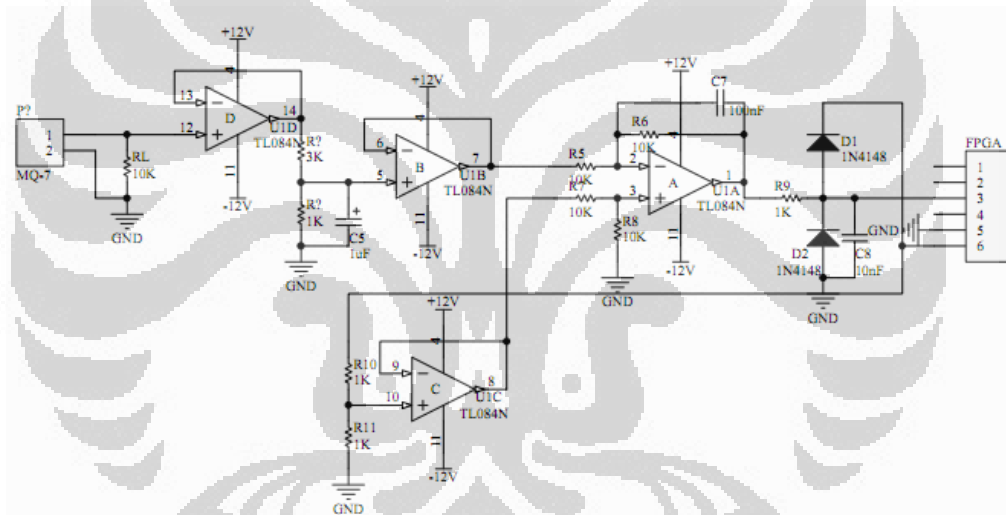
Sedangkan untuk antar muka ADC, dibutuhkan rangkaian transduser yang bertujuan untuk membagi tegangan V_{OUT} dari sensor untuk kemudian masuk ke *Analog Capture Circuit board* FPGA yang digunakan. *Analog Capture Circuit* ini memiliki jangkauan pembacaan sebesar 2,5V, seperti yang terlihat pada Gambar 4.3., namun untuk penyederhanaan sistem, pada penelitian ini hanya akan digunakan jangkauan $0000_{16} - 1FFF_{16}$. Sehingga, jangkauan tegangan yang dapat diterima FPGA adalah sebesar 1,25V. Rangkaian yang digunakan untuk membagi tegangan V_{OUT} sensor ditunjukkan pada Gambar 4.4. Rangkaian berguna untuk membagi tegangan V_{OUT} dengan 4, sesuai dengan Persamaan 4.1.



Gambar 4.3. Jangkauan pembacaan *Analog Capture Circuit* [11]

$$V_L = \frac{R_2}{R_1 + R_2} V_S \quad (4.1)$$

Dengan V_S adalah V_{OUT} dari sensor, R_1 adalah 1Ω , R_2 adalah 3Ω , dan V_L adalah tegangan beban yang merupakan tegangan di R_2 . Yang perlu diperhatikan adalah jika tegangan masukan berubah-ubah (tidak stabil) maka V_L juga akan berubah tidak stabil, maka dibutuhkan suatu rangkaian agar tegangan masukan menjadi stabil. Bagian tersebut dilakukan oleh IC TL084N pada pin 12,13, dan 14. Kemudian untuk membuat nilai $1,65V$ menjadi nilai $0V$ baru, maka hasil dari *voltage divider* tersebut dikurangi dengan $1,65V$. Setelah itu, barulah kemudian terbentuk ADC dengan jangkauan 0 hingga $1,25V$. Gain yang digunakan pada *pre-amplifier* adalah -1 . Rangkaian ini nantinya akan terhubung pada konektor *Analog Capture Circuit J7*. Penentuan port ini dilakukan melalui file UCF.



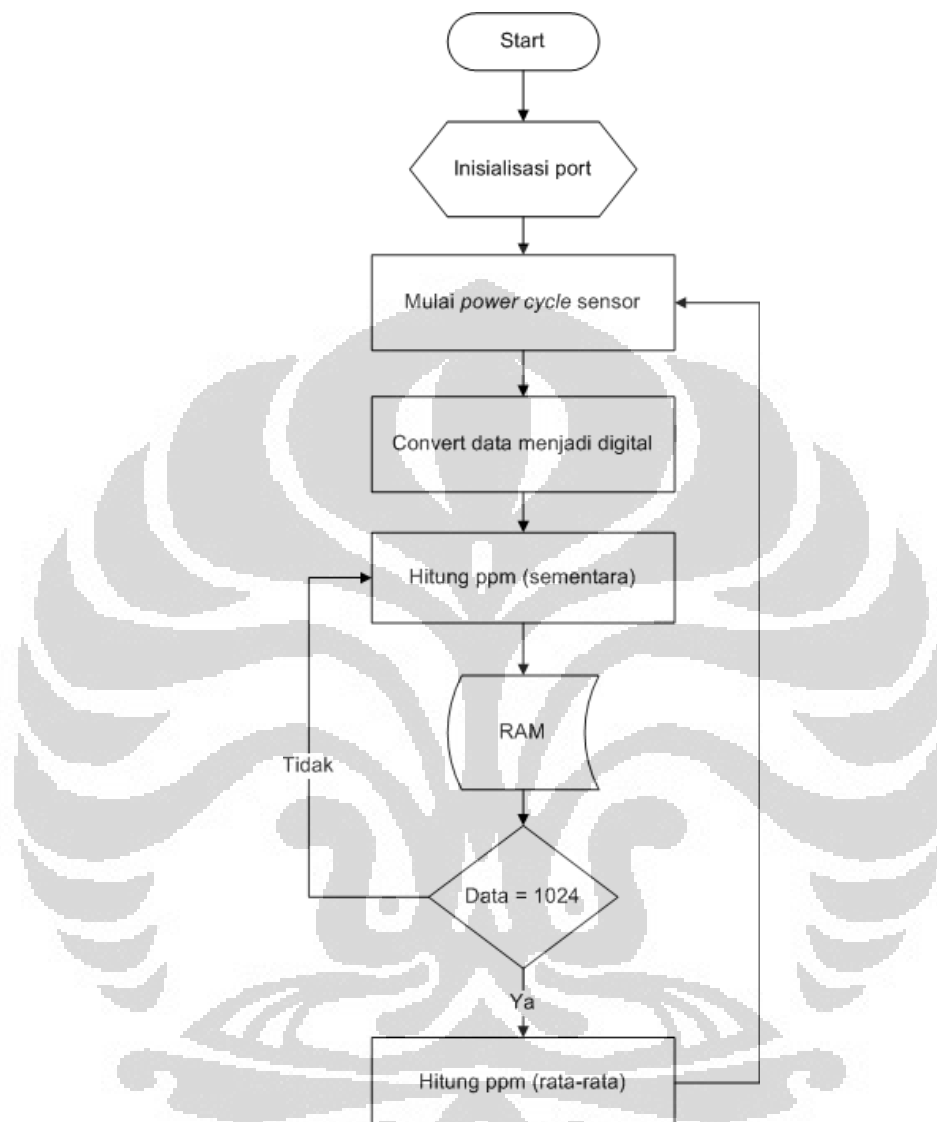
Gambar 4.4. Rangkaian *transducer*

Kedua rangkaian antarmuka ini kemudian dihubungkan dengan sensor dan FPGA. Sistem secara keseluruhan diperlihatkan pada Lampiran Gambar A-1.

4.2 Implementasi Perangkat Lunak

Pengimplementasian sistem ini didasarkan pada diagram alir yang ditunjukkan pada Gambar 4.5. Gambar tersebut menjelaskan bahwa untuk dapat melakukan pembacaan ppm gas CO diperlukan tiga tahapan, antara lain pemberian tegangan yang sesuai yang dibutuhkan sensor untuk membentuk *process cycle*-nya, pembacaan tegangan V_{OUT} sensor ketika menangkap gas CO di

udara (dalam hal ini membutuhkan ADC pada FPGA karena keluaran sensor berupa analog), dan pengubahan nilai tegangan V_{OUT} menjadi ppm.



Gambar 4.5. Alur diagram sistem pemantau polusi udara berbasis FPGA

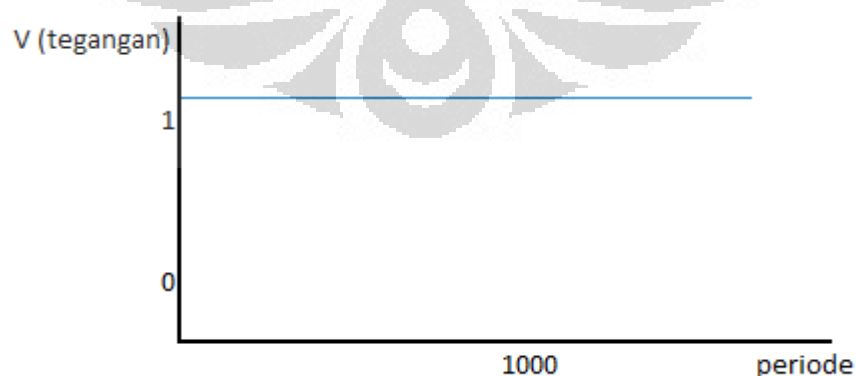
4.2.1 Pemberian Tegangan ke MQ7

Tegangan yang dibutuhkan oleh sensor MQ7 sebagai *heating voltage* adalah 5V dan 1,4V untuk masing-masing *purge phase* dan *sense phase*. Setelah dibantu dengan beberapa komponen dan rangkaian yang telah dijelaskan pada Gambar 4.2. tegangan yang dikeluarkan diatur melalui *Pulse Width Modulation* (PWM).

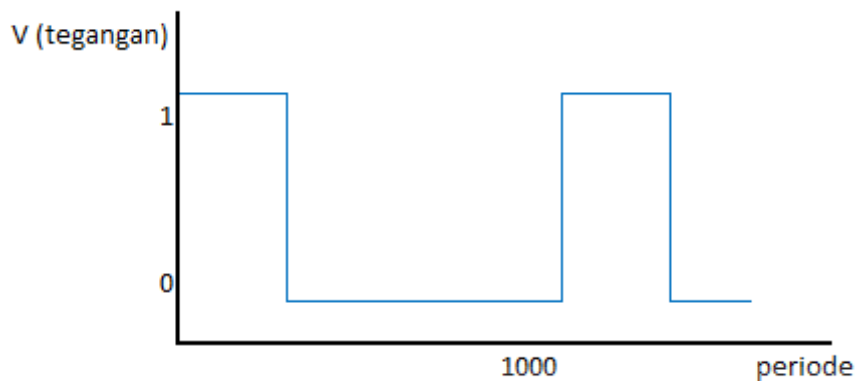
File `sensor.vhd` berisikan program untuk mengatur PWM yang dimaksudkan sebelumnya. Untuk dapat mengatur besarnya tegangan yang

diberikan kepada sensor, dibutuhkan pengaturan pada variabel PWM_accumulator. Pengaturan yang dimaksudkan adalah berupa perbandingan setara. Jika 1000 adalah pemberi tegangan 5V, maka 280 adalah pemberi tegangan 1,4V. Selama counter pada PWM_accumulator masih memenuhi syarat nilai 1 akan tetap diberikan. Sedangkan untuk dapat mengatur tegangan sesuai dengan waktu yang telah ditentukan, dibutuhkan perubahan pada *duty cycle* FPGA. Frekuensi yang digunakan adalah 50 MHz. Frekuensi 50 MHz menunjukkan bahwa dalam 1 detik terdapat 50.000.000 kali *clock*. *Power cycle* sensor membutuhkan 60 detik dan 90 detik. Dalam hal ini, PWM akan membutuhkan variabel *co_pwr_time* sebanyak 50.000.000 dan count sejumlah 150 yang berasal dari 60 detik ditambah 90 detik untuk tiap fase.

Syarat pertama adalah ketika PWM_Accumulator < 1000 dalam jangkauan count 60, maka akan diberikan nilai 1 atau setara dengan 5V selama 60 detik. Perlakuan ini merupakan implementasi dari *duty cycle* sebesar 100% yang ditunjukkan pada Gambar 4.6. Syarat kedua adalah ketika PWM_Accumulator < 280 dalam jangkauan count antara 60 dan 150, maka akan diberikan nilai 1 atau setara dengan 1,4V selama 90 detik. Perlakuan ini merupakan implementasi dari *duty cycle* sebesar 28% yang ditunjukkan pada Gambar 4.7. Ketika PWM_Accumulator melebihi 1000 dan *co_pwr_time* melebihi 50.000.000, maka PWM_Accumulator dan *co_pwr_time* akan di-*reset*. Hasil akhir PWM ditunjukkan pada Gambar 4.9. Pseudocode dari syarat-syarat yang disebutkan di atas ditunjukkan oleh Gambar 4.8.



Gambar 4.6. *Duty cycle* 100%



Gambar 4.7. Duty cycle 28%

```

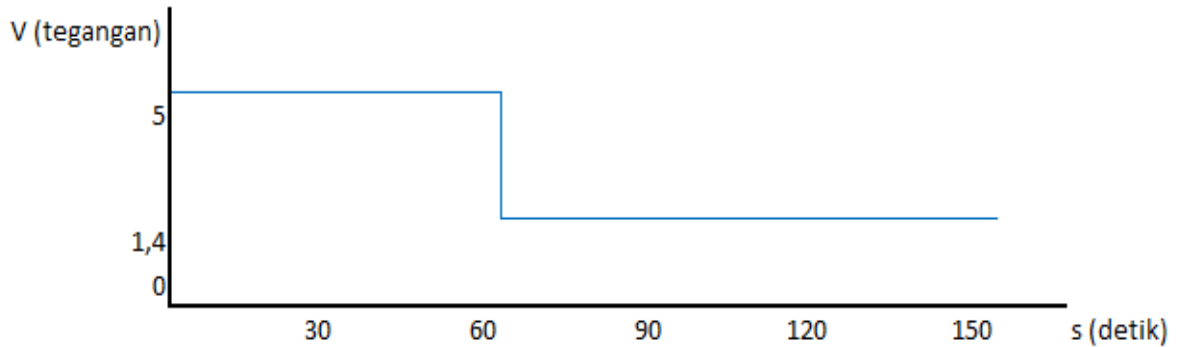
Start
  port : clk, pwm_out, pwm_accumulator,      co_pwr_time,
        count
begin
  if rising_edge (clk){
    co_pwr_time<=co_pwr_time+1;
    if co_pwr_time = 50000000{
      co_pwr_time<=0;
      if count = 75{
        count<=0;
      }
      else
        count<=count+1;
    }
  }

  if count <=60{
    PWM_Accumulator<=PWM_Accumulator+1;
    if(PWM_Accumulator>999){ --reset counter pwm
      PWM_Accumulator<="0000000000";
    }
    if(PWM_Accumulator<1000){ -- output 5v
      PWM_out<='1';
    }
    else
      PWM_out<='0';
  }

  elsif (count > 60 and count <= 150)
    PWM_Accumulator<=PWM_Accumulator+1;
    if(PWM_Accumulator>999){ --reset counter pwm
      PWM_Accumulator<="0000000000";
    }
    if(PWM_Accumulator<239){ --output 1,4v
      PWM_out<='1';
    }
    else
      PWM_out<='0';
  }
}
}
end

```

Gambar 4.8. Pseudocode PWM



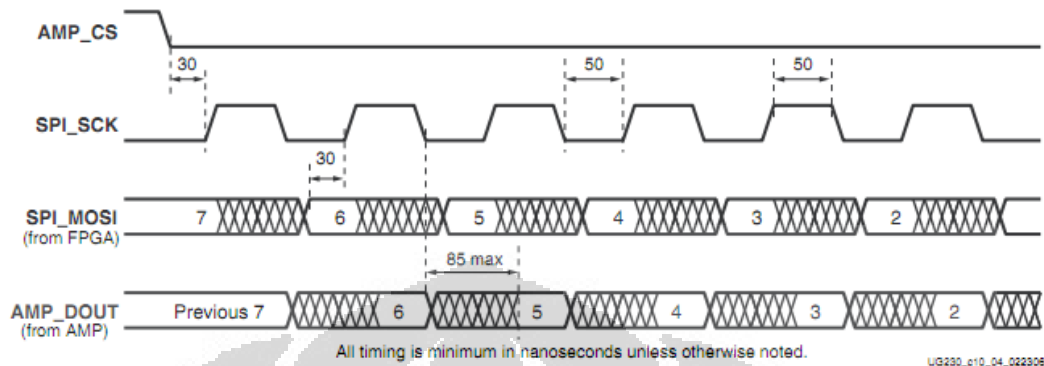
Gambar 4.9. Sinyal PWM untuk merubah tegangan

4.2.2 Analog Capture Circuit

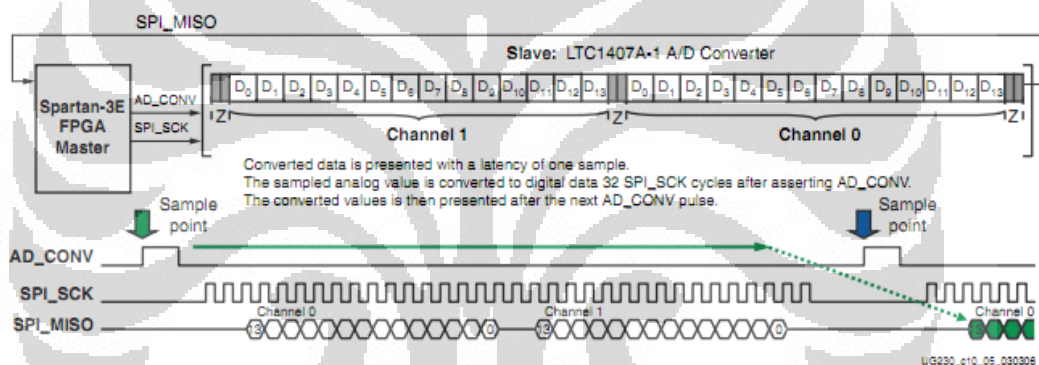
Analog Capture Circuit yang dimaksudkan terdiri dari *inverting amplifier* dan A/D itu sendiri. Keduanya terhubung melalui komunikasi *Serial Peripheral Interface* (SPI). Pada *amplifier* ini, seperti yang telah dijelaskan pada Bab 2 sebelumnya bahwa FPGA Spartan 3E akan berperan sebagai *master* yang akan mengirimkan nilai *gain*. Transaksi data *gain* dari master terjadi saat AMP_CS di *set low* untuk mengaktifkannya. Hal ini dikarenakan AMP_CS merupakan komponen yang *active low* (lihat Gambar 4.10.). *Gain* dikirimkan melalui 8 bit command secara serial dengan ketentuan *most significant bit* B3 dikirimkan lebih dahulu. Kemudian *amplifier* menangkap data di SPI_MOSI saat SPI_SCK *rising edge*. Data tersebut harus valid paling tidak 30 ns sebelum SPI_SCK *rising*. *Amplifier* menghasilkan serial data sebagai *output* di AMP_DOUT saat SPI_SCK *falling edge*. Hasil keluaran *amplifier* akan menjadi masukan bagi A/D. Siklus pada A/D dimulai dengan mengubah nilai AD_CONV menjadi 1. Perubahan ini akan menyebabkan ADC secara simultan *men-sampling* kedua *channel*. Ketentuan yang perlu diperhatikan adalah SPI_SCK harus ada dalam kondisi *high impedance*. Jika kondisi ini tidak terpenuhi, maka ADC akan memblokir komunikasi dengan peripheral SPI lainnya, dalam hal ini *amplifier*, dengan konsekuensi A/D tidak akan dapat *men-sampling* data. Pada praktisnya, SPI_SCK yang digunakan adalah 34 *cycle* (lihat Gambar 4.11.). Nilai ini didapat dari $(2 + 14 + 2 + 14 + 2) = 34$. 14 merupakan jumlah bit *output* A/D.

Nilai *output* dari A/D kemudian akan disimpan dalam memori sebesar 1KB. Hal ini ditujukan agar hasil keluaran dari A/D stabil dan tidak berubah-

ubah. Untuk itu, diperlukan RAM yang dibuat secara virtual menggunakan IP (Core Generator & Architecture Wizard) pada ISE Xilinx.

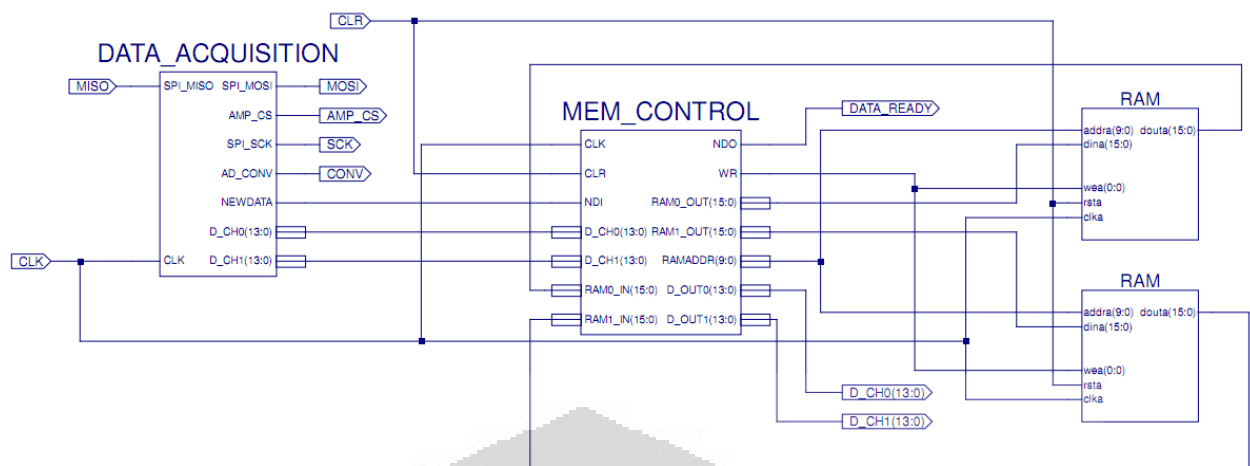


Gambar 4.10. Pewaktuan SPI ketika berkomunikasi dengan *amplifier*[11]



Gambar 4.11. Antarmuka pengkonversi A/D [11]

Untuk dapat merepresentasikan pemahaman dari ketentuan konfigurasi *Analog Capture Circuit*, maka dirancanglah diagram *schematic* yang ditunjukkan oleh Gambar 4.12. Di mana pada gambar tersebut, DATA_ACQUISITION merupakan fungsi *Analog Capture Circuit* yang berisikan *pre-amplifier* dan ADC. Data masuk melalui SPI_MISO dan keluar melalui kanal 0, yaitu D_CH0. Dalam hal ini, D_CH1 dapat diabaikan, alasan mengapa *port* ini digunakan adalah karena *Analog Capture Circuit* FPGA Xilinx Spartan 3E memang memiliki 2 kanal. Selanjutnya, nilai digital hasil dari DATA_ACQUISITION akan dimasukkan ke dalam MEM_CONTROL. Fungsi ini bertugas untuk menyimpan nilai satu per satu ke dalam RAM. Setelah RAM terisi penuh, maka nilai-nilai tersebut akan di rata-ratakan.



Gambar 4.12. Diagram schematic rangkaian ADC

4.2.3 Pengaturan Bus *Serial Peripheral Interface* (SPI)

Bus *Serial Peripheral Interface* (SPI) yang digunakan sebagai penjaluran segala komponen yang ada harus diatur. Pengaturan ini bertujuan agar tidak ada “tabrakan” data dalam bus SPI. Table 4.1 menunjukkan sinyal dan nilai logic yang dibutuhkan untuk men-disable beberapa device yang menggunakan SPI sebagai komunikasi transfer data.

Tabel 4.1. Konfigurasi bus SPI [11]

Signal	Disable Device	Disable Value
SPI_SS_B	SPI Serial Flash	1
AMP_CS	Programmable Pre-Amplifier	1
DAC_CS	DAC	1
SF_CEO	StrataFlash Parallel Flash PROM	1
FPGA_INIT_B	Platform Flash PROM	1

BAB V

PENGUJIAN DAN ANALISA SISTEM PEMANTAU POLUSI UDARA BERBASIS FPGA XILINX SPARTAN 3E MENGUNAKAN SENSOR GAS CO MQ7

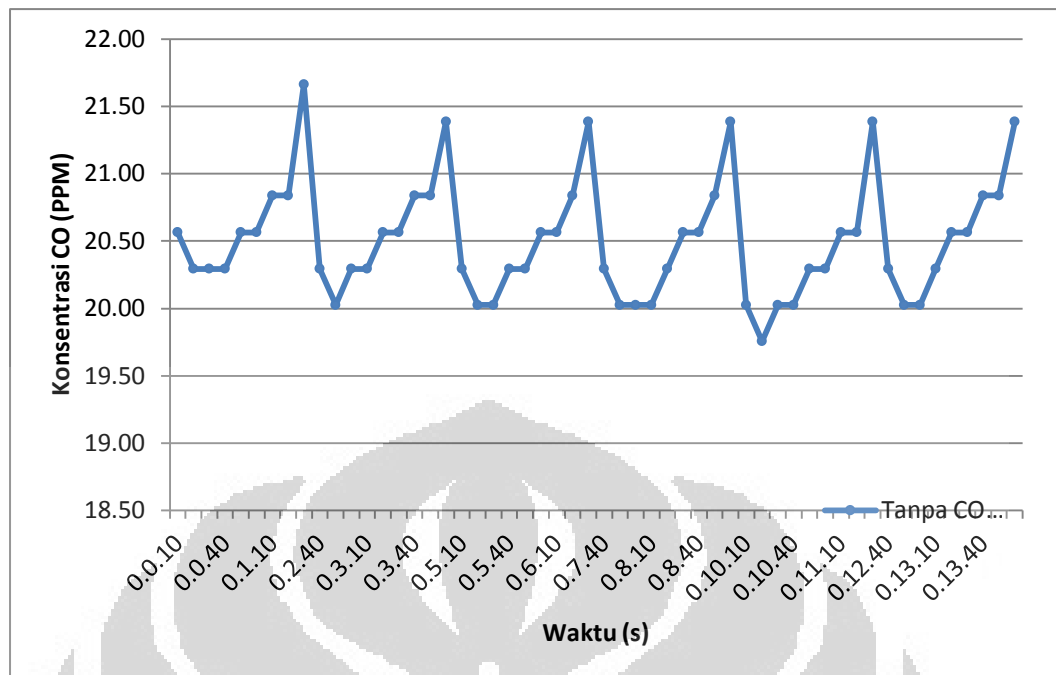
Untuk dapat mengetahui performa sistem pemantau polusi udara di area parkir tertutup berbasis FPGA Xilinx Spartan 3E dan menggunakan sensor MQ7 ini, maka dibutuhkan pengujian dengan pengambilan data-data disertai dengan analisa.

5.1 Pengujian Pembacaan Kadar Gas CO oleh Sensor MQ7

Pengujian pembacaan kadar gas CO oleh MQ7 berguna untuk mengetahui akurasi sistem. Gambar 5.1 adalah grafik hasil kalibrasi sistem. Baik kalibrasi maupun pengukuran dilakukan pengambilan sampel tiap 10 detik sekali dan dilakukan dengan menggunakan kotak simulasi (area tertutup) selama 15 menit. Pembacaan yang dilakukan adalah ketika sensor sedang memasuki *sense phase* atau pada kondisi sedang dialiri tegangan sebesar 1,4V. Untuk pengkalibrasian sistem, sensor membaca kadar CO yang ditempatkan pada kondisi konstan 20 ppm. Suhu pada saat pengambilan data adalah 26°C. Dari gambar, dapat dilihat bahwa sistem mendapatkan nilai yang stabil pada *cycle* kedua atau sekitar 5 menit. Pada pengujian pembacaan kadar gas CO oleh sensor gas CO MQ7 ini, diterapkan lima kondisi lainnya, yaitu pada kondisi 25, 35, 75, 150, dan 350 ppm. Besar pemberian volume gas pada pengujian ditunjukkan pada Tabel 5.1.

Tabel 5.1. Kadar CO terhadap gas yang diujikan

Kondisi	Volume Gas (ml)	Konsentrasi (ppm)
1	-	20
2	600	25
3	1500	35
4	4500	75
5	9000	150
6	15000	350



Gambar 5.1. Grafik hasil kalibrasi sistem pemantau polusi udara berbasis FPGA Xilinx Spartan 3E menggunakan sensor gas CO MQ7

Untuk menghitung nilai *error*, dilakukan perhitungan dengan Persamaan 5.1. Selanjutnya, akan ditunjukkan perhitungan untuk kondisi kalibrasi. Nilai *error* untuk kondisi lain akan langsung diperlihatkan pada Tabel 5.2.

$$\% \text{ error} = \left| \frac{\text{nilai selisih}}{\text{nilai ideal}} \right| \times 100\% \quad (5.1)$$

$$\% \text{ error} = \left| \frac{0,5}{20} \right| \times 100\%$$

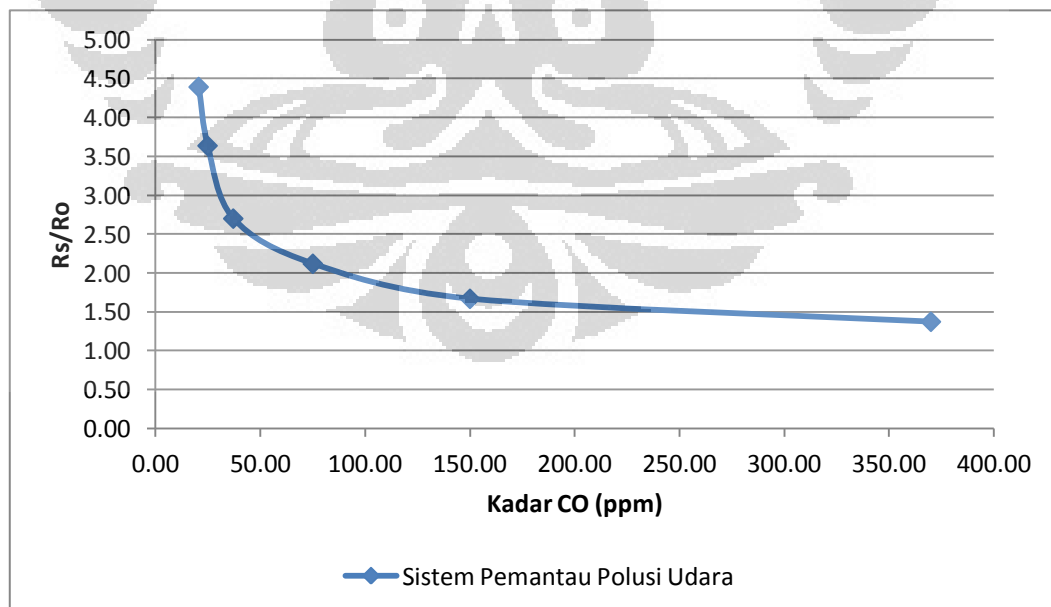
$$\% \text{ error} = 2,5\%$$

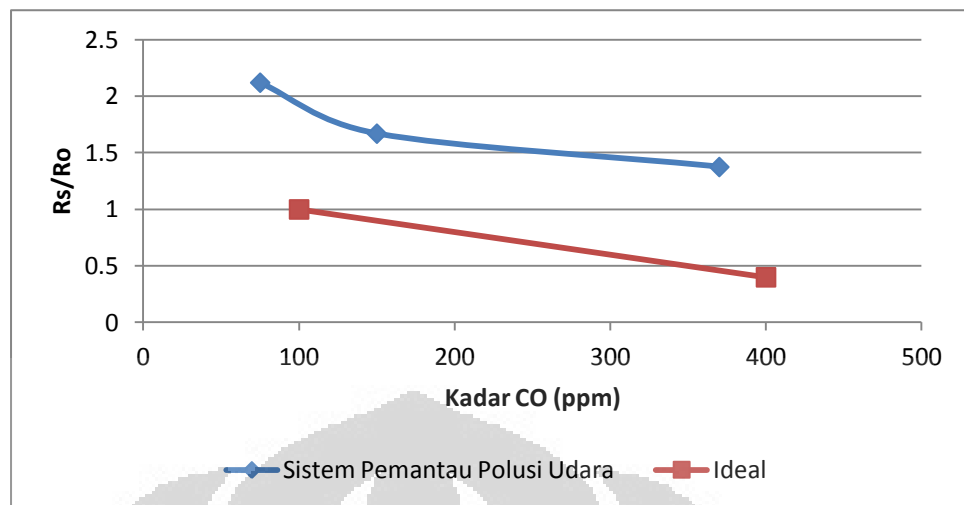
Pada pengujian ini, variabel suhu dan kelembaban dibuat konstan. Data pengujian dengan enam kondisi (termasuk dengan kondisi kalibrasi) terdapat pada Lampiran Tabel A-1. Nilai rata-rata dari enam kondisi pengujian ditunjukkan pada Tabel 5.2.

Tabel 5.2. Hasil rata-rata pengukuran pembacaan kadar gas CO oleh sensor MQ7

Kondisi (ppm)	Hasil Pengukuran (ppm)	Selisih	% Error	Rs/Ro
20	20,50	0,5	2,5	4,39
25	24,74	0,26	1,04	3,64
35	37,02	2,02	5,77	2,70
75	77,22	2,22	2,96	2,12
150	147,83	2,17	1,44	1,67
350	353,39	3,39	0,96	1,38
Rata-rata		1,76	2,45	

Dari kelima kondisi di atas, grafik kerja sistem dapat ditentukan dengan cara menghubungkan titik-titik hasil pengukuran. Titik-titik yang dimaksudkan adalah nilai rata-rata data hasil pengujian sistem yang telah dilakukan. Persamaan ini berguna untuk mengetahui nilai kerja sistem. Axis horizontal yang digunakan adalah konsentrasi CO berupa ppm. Sedangkan axis vertikal yang digunakan adalah R_s/R_o . Grafik kerja sistem ditunjukkan oleh Gambar 5.2. Namun, pada analisis hanya akan dibahas dimulai dari kondisi 75 ppm ke atas untuk menyesuaikan data yang ada pada referensi [13], maka grafik kerja sistem dapat diilustrasikan seperti pada Gambar 5.3.

Gambar 5.2. Grafik R_s/R_o vs Konsentrasi CO (ppm)



Gambar 5.3. Grafik R_s/R_o vs Konsentrasi CO (ppm) dibandingkan dengan referensi

Pada Gambar 5.3. dapat dilihat bahwa grafik kerja sistem (ditunjukkan dengan tanda berwarna biru) sesuai dengan karakteristik ideal sensor (ditunjukkan dengan tanda berwarna merah) walaupun terjadi pergeseran karakteristik kerja. Penyebab terjadinya hal ini akan dijelaskan selanjutnya. Pada Tabel 5.2., dapat dilihat bahwa semakin besar paparan gas CO yang diberikan, rasio resistansi sensor akan menurun. Hal ini disertai dengan semakin besarnya selisih perhitungan sistem dengan kondisi ideal.

Terdapat beberapa penyebab terjadinya *error*, antara lain:

1. Perubahan yang terjadi tiap 10 detik dicatat secara manual, sehingga hasil menjadi kurang presisi.
2. Ketidaksesuaian kondisi suhu dan kelembaban dengan literatur. Hal ini disebabkan karena sulitnya mendapatkan kondisi yang diinginkan. Dalam datasheet, sensor bekerja maksimal pada suhu 20° dan kelembaban (RH) 65%. Namun, pada kenyataannya sistem ini bekerja pada suhu 26°C . Jika mengacu pada Referensi [13], semakin meningkat suhu, semakin tinggi nilai R_s/R_o .
3. Pengukuran dilakukan dengan jarak waktu yang cukup cepat. Sehingga, diduga *heater coil*, yang berupa nikel kromium, belum dapat membersihkan kontaminasi paparan CO pada sensor dari pengukuran sebelumnya. Hal ini menyebabkan terjadinya akumulasi kadar CO dalam sensor yang

mengakibatkan resistansi semakin kecil dan secara lebih jauh menjadikan nilai ppm lebih besar dari yang seharusnya ada.

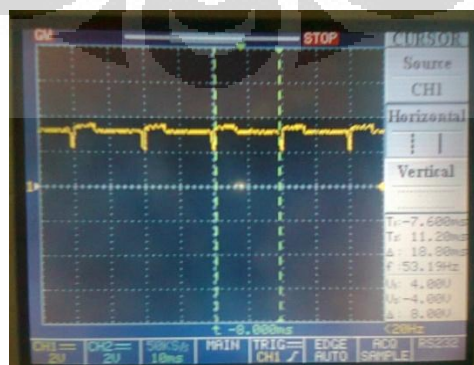
5.2 Pengujian Kecepatan Pembacaan Sensor

Pengujian ini bertujuan untuk mengetahui kecepatan data yang dikirimkan ke FPGA untuk kemudian diolah. Pengambilan data dilakukan sebanyak 10 kali dalam rentang waktu *sense phase* dengan menggunakan *digital storage oscilloscope* (DSO). Sampel diambil dengan cara melihat lebar tiap sinyal.

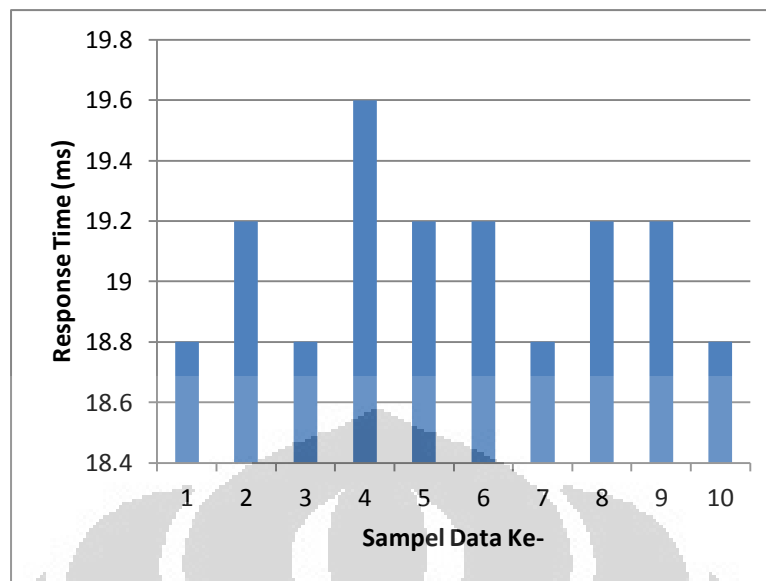
Contoh pengambilan data yang dilakukan ditunjukkan pada Gambar 5.4. Hasil pengukuran kecepatan pembacaan sensor ditunjukkan pada Tabel 5.3 dan Gambar 5.5.

Tabel 5.3. Hasil pengukuran data pengujian kecepatan pembacaan sensor

Sampel Data	Response Time (ms)
1	18,8
2	19,2
3	18,8
4	19,6
5	19,2
6	19,2
7	18,8
8	19,2
9	19,2
10	18,8
Rata-rata	19,11



Gambar 5.4. Hasil pengukuran data pengujian kecepatan pembacaan sensor



Gambar 5.5. Grafik pengukuran kecepatan pembacaan sensor

Untuk pengujian kecepatan pembacaan sensor, berdasarkan data, waktu rata-rata yang dibutuhkan sensor untuk mengirimkan data terbaru dari hasil pembacaannya adalah sebesar 19,11 ms. Waktu ini adalah waktu yang dibutuhkan sensor untuk mengirimkan data kepada FPGA. Lebar sinyal yang diukur merupakan representasi dari berubahnya tegangan aksi dari terdapatnya gas CO di sekitar sensor MQ7.

BAB VI

KESIMPULAN

1. Untuk dapat memberikan tegangan yang sesuai dengan kebutuhan sensor, maka sistem ini membutuhkan antarmuka yang menghubungkan sensor dan FPGA Xilinx Spartan 3E (XC3S500E-4FG320C) dengan algoritma *Pulse Width Modulation* dengan pengaturan *duty cycle*, yaitu sebesar 100% (5V) dan 28% (1,4V).
2. Untuk dapat membaca hasil V_{RL} yang berupa nilai analog, dibutuhkan pula antarmuka yang menghubungkan V_{OUT} sensor dengan *Analog Capture Circuit* yang terdapat pada *board* Xilinx Spartan 3E Starter Kit (XC3S500E-4FG320C) berupa rangkaian transduser. Rangkaian ini berguna sebagai *handle* masukan yang dapat diterima oleh FPGA itu sendiri.
3. Pengujian pembacaan kadar konsentrasi CO menunjukkan bahwa data yang didapat dan data normal memiliki kesesuaian. Bahwa semakin besar konsentrasi gas, semakin kecil nilai R_s/R_o . pada pengujian ini didapatkan nilai rata-rata kesalahan pembacaan sensor sebesar 1,76 ppm atau sebesar 2,45%.
4. FPGA menerima data tiap 19,11 ms sekali.

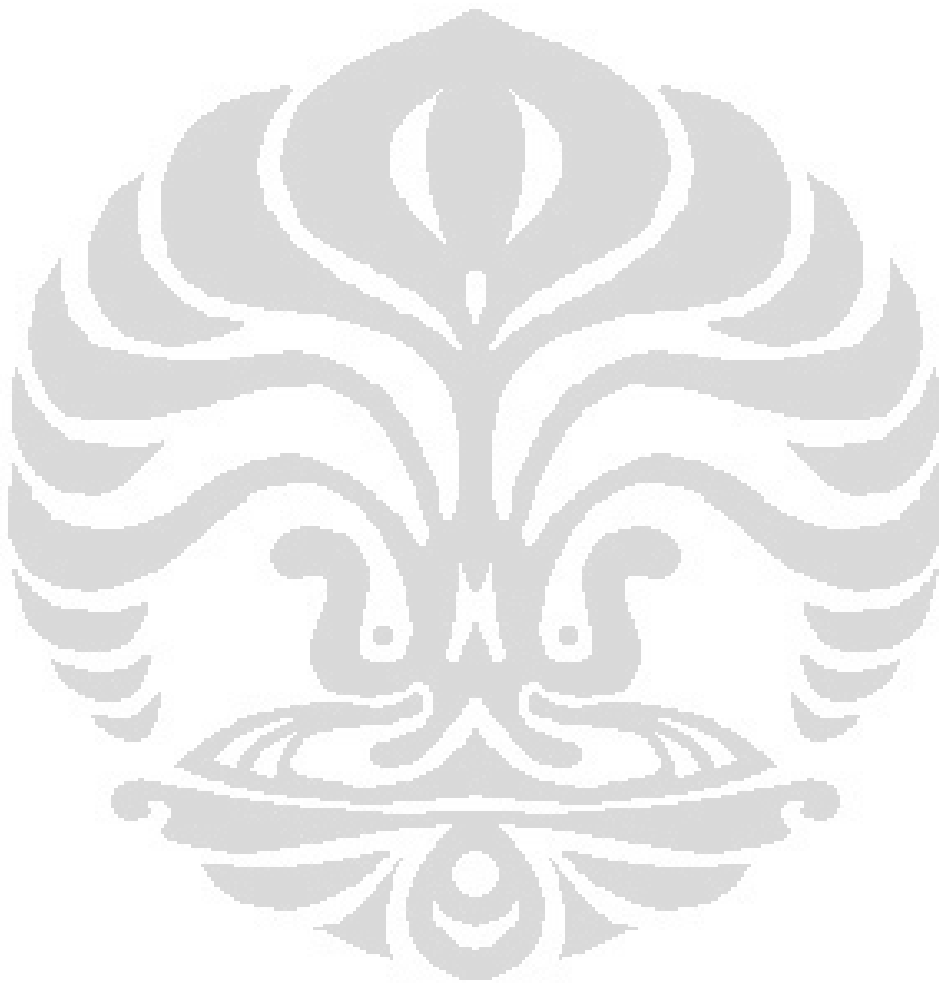
DAFTAR REFERENSI

- [1] Pemerintah Provinsi DKI Jakarta, *Portal Pemerintah Provinsi DKI Jakarta*, Oktober, 2011, [Online]. Available: <http://www.jakarta.go.id> [Accessed: 20 November 2011]
- [2] "Penumpang Bus Ditemukan Tewas Akibat Keracunan Gas Buang" *Vivanews*, Juli, 2010, [Online]. Available: <http://www.video.vivanews.com> [Accessed: 20 November 2011]
- [3] R. Bailey, "Carbon Monoxide" *Biology.about.com*, [Online]. Available: http://www.biology.about.com/od/molecularbiology/a/carbon_monoxide.htm [Accessed: 12 November 2011]
- [4] Y. Berra, "Software Development Life Cycle (SDLC)" *DePaul University*, [Online]. Available: <http://www.condor.depaul.edu/jpetlick/extra/394/Session2.ppt> [Accessed: 13 November 2011]
- [5] "KBBI Daring" *Pusat Bahasa*, Februari, 2008, [Online]. Available: <http://www.badanbahasa.kemdiknas.go.id/kbbi/> [Accessed: 12 November 2011]
- [6] Hollemann-Wiberg, *Inorganic Chemistry*, 1st ed. California: Academic Press, 2001.
- [7] "Udara" *Kementrian Kesehatan Republik Indonesia*, [Online]. Available: <http://www.depkes.go.id/downloads/Udara.pdf> [Accessed: 21 November 2011]
- [8] Nuzulul Z. Haq, "Pencegahan Keracunan Gas" *Nuzulul Official Box*, Juni 2010, [Online]. Available: <http://www.nuzulz.co.cc/kesehatan/pencegahan-keracunan-gas/> [Accessed: 21 November 2011]
- [9] A. Irwansyah, "Modul Praktikum VLSI dan Logic Design" *Institut Teknologi Sepuluh November*, [Online]. Available: <http://www.lecturer.eepis-its.edu> [Accessed: 12 November 2011]
- [10] Jean P. Nicolle, "What are FPGAs?" *fpga4fun.com - where FPGAs are fun*, Desember, 2010, [Online]. Available: <http://www.fpga4fun.com/FPGAinfo1.html> [Accessed: 12 November 2011]
- [11] *All Programmable Technologies from Xilinx Inc*, Januari, 2011, [Online]. Available: <http://www.xilinx.com> [Accessed: 15 November 2011]
- [12] Jean P. Nicolle, "How FPGAs Work" *fpga4fun.com - where FPGAs are fun*, September, 2009, [Online]. Available: <http://www.fpga4fun.com/FPGAinfo2.html> [Accessed: 20 November 2011]

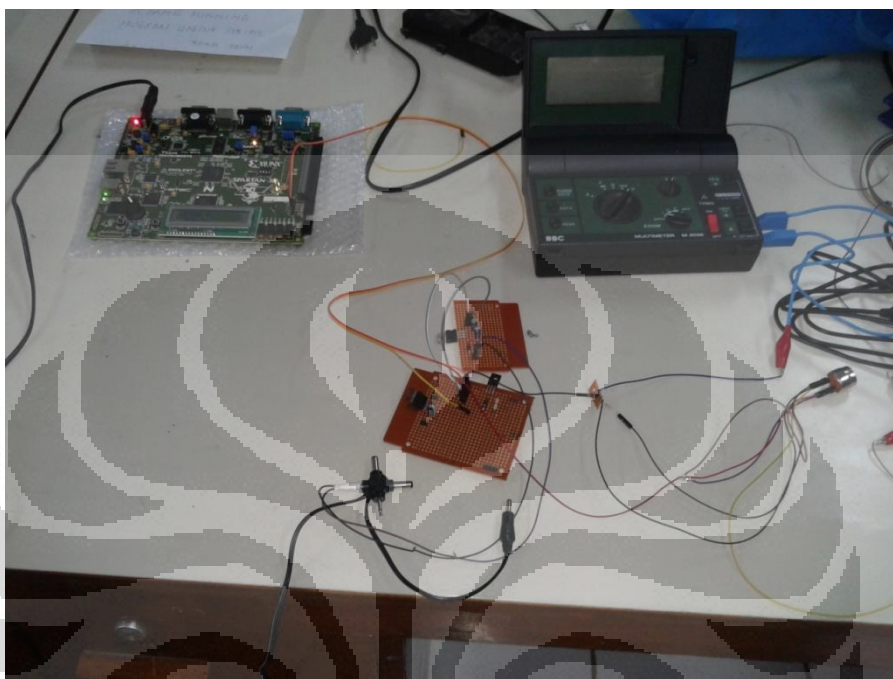
- [13] "Technical Data MQ7 Gas Sensor" *Hanwei Electronics Co. LTD*, [Online]. Available: <http://www.sparkfun.com/datasheets/Sensors/Biometric/MQ7.pdf> [Accessed: 11 November 2011]
- [14] Ganis Rama Pradika, S.T, M.T Sumardi, and S.T, M.T Budi Setiyono. (2011) Pengendalian Putaran Kipas Ventilator pada Smart Smoking Area Menggunakan Mikrokontroler ATMEGA 8535 dan Sensor Gas MQ7. [Online]. Available: <http://ejournal.undip.ac.id/index.php/transmisi/article/view/3616> [Accessed: 11 November 2011]
- [15] *Housing Estate*, [Online]. Available: <http://www.housing-estate.com> [Accessed: 19 Desember 2011]
- [16] (2005) dictionary.com. [Online]. Available: <http://www.dictionary.reference.com/browse/led> [Accessed: 19 Desember 2011]
- [17] M. Bellis, "Television History - Cathode Ray Tube" *About.com*, [Online]. Available: <http://inventors.about.com/od/cstartinventions/a/CathodeRayTube.htm> [Accessed: 19 Desember 2011]
- [18] Badr Ghatasheh. (2012, Januari) DC Motor Bidirectional Speed Control Using PWM. [Online]. Available: <http://eproject-jo.blogspot.com/2012/01/dc-motor-bidirectional-speed-control.html> [Accessed: 3 Januari 2011]

DAFTAR PUSTAKA

Peter J. Ashenden, *The VHDL Cookbook*, 1st ed. Adelaide, South Australia: Dept. Computer Science, University of Adelaide, 1990.



LAMPIRAN



Gambar A-1 Perangkat keras sistem pemantau polusi udara di area parkir tertutup berbasis FPGA Xilinx Spartan 3E dan menggunakan sensor MQ7

Tabel A-1 Data pengukuran kalibrasi sistem

Waktu (s)	Kalibrasi (ppm)	Kondisi 1 (25 ppm)	Kondisi 2 (35 ppm)	Kondisi 3 (75 ppm)	Kondisi 4 (150 ppm)	Kondisi 5 (350 ppm)
0.0.10	20.57	23.35	37.48	72.57	147.88	354.21
0.0.20	20.29	23.64	37.49	72.29	147.60	353.93
0.0.30	20.29	23.92	37.49	72.29	147.60	353.93
0.0.40	20.29	24.21	37.48	72.29	147.60	353.93
0.0.50	20.57	24.21	37.40	72.57	147.88	354.21
0.1.00	20.57	24.21	37.41	72.57	147.88	354.21
0.1.10	20.84	24.21	37.34	72.84	148.15	354.48
0.1.20	20.84	24.21	37.34	76.22	148.15	354.48
0.2.30	21.66	23.92	35.54	73.66	148.97	355.30
0.2.40	20.29	23.92	35.94	72.29	147.60	353.93
0.2.50	20.02	24.21	36.02	72.02	147.33	353.66
0.3.00	20.29	24.50	36.20	72.29	147.60	353.93
0.3.10	20.29	24.50	36.98	72.29	147.60	353.93
0.3.20	20.57	24.50	37.06	72.57	147.88	354.21

0.3.30	20.57	24.50	37.16	72.57	147.88	354.21
0.3.40	20.84	24.50	37.17	72.84	148.15	354.48
0.3.50	20.84	24.50	37.17	72.84	148.15	354.48
0.5.00	21.39	23.92	37.22	73.39	148.70	355.03
0.5.10	20.29	24.50	37.08	72.29	147.60	353.93
0.5.20	20.02	24.50	37.22	76.80	147.33	353.66
0.5.30	20.02	24.80	37.63	72.02	147.33	353.66
0.5.40	20.29	24.80	37.65	72.29	147.60	353.93
0.5.50	20.29	24.50	37.66	76.80	147.60	353.93
0.6.00	20.57	24.50	37.66	77.89	147.88	354.21
0.6.10	20.57	24.80	37.65	77.52	147.88	354.21
0.6.20	20.84	24.80	37.63	77.32	148.15	354.48
0.7.30	21.39	24.80	35.57	77.22	148.70	353.48
0.7.40	20.29	25.09	35.33	72.29	147.60	352.38
0.7.50	20.02	25.09	35.65	72.02	147.33	352.11
0.8.00	20.02	25.09	36.73	72.02	147.33	352.11
0.8.10	20.02	25.38	37.11	72.02	147.33	352.11
0.8.20	20.29	25.38	37.09	72.29	147.60	352.38
0.8.30	20.57	25.38	37.35	72.57	147.88	352.66
0.8.40	20.57	25.38	37.61	72.57	147.88	352.66
0.8.50	20.84	25.38	37.66	72.84	148.15	352.93
0.10.00	21.39	24.80	35.41	77.38	148.70	352.48
0.10.10	20.02	24.80	36.49	77.09	147.33	351.11
0.10.20	19.76	25.09	37.00	71.76	147.07	350.85
0.10.30	20.02	25.09	37.08	72.02	147.33	351.11
0.10.40	20.02	25.38	37.06	72.02	147.33	351.11
0.10.50	20.29	25.38	37.01	72.29	147.60	351.38
0.11.00	20.29	25.38	37.49	72.29	147.60	351.38
0.11.10	20.57	25.09	37.62	72.57	147.88	351.66
0.11.20	20.57	25.38	37.66	72.57	147.88	351.66
0.12.30	21.39	23.92		77.35	148.70	353.08
0.12.40	20.29	25.09		72.29	147.60	351.98
0.12.50	20.02	25.09		72.02	147.33	351.71
0.13.00	20.02	25.38		72.02	147.33	351.71
0.13.10	20.29	25.38		72.29	147.60	351.98
0.13.20	20.57	25.38		72.57	147.88	352.26
0.13.30	20.57	25.38		72.57	147.88	352.26
0.13.40	20.84	25.38		72.84	148.15	352.53
0.13.50	20.84	25.38		72.84	149.34	353.72
0.15.00	21.39	23.35		77.38	148.70	353.08

