



UNIVERSITAS INDONESIA

**RANCANG BANGUN DAN ANALISIS ANTARMUKA SISTEM
PEMANTAU DAN PERINGATAN POLUSI UDARA PADA
RUANG PARKIR TERTUTUP BERBASIS FPGA XILINX
SPARTAN 3E**

SKRIPSI

**EVAN G. SUMBAYAK
0806339080**

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



UNIVERSITAS INDONESIA

**RANCANG BANGUN DAN ANALISIS ANTARMUKA SISTEM
PEMANTAU DAN PERINGATAN POLUSI UDARA PADA
RUANG PARKIR TERTUTUP BERBASIS FPGA XILINX
SPARTAN 3E**

SKRIPSI

Diajukan sebagai salah satu syarat memperoleh gelar sarjana.

**EVAN G. SUMBAYAK
0806339080**

**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 : Evan G. Sumbayak

NPM : 0806339080

Tanda Tangan : 

Tanggal : 28 Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh

Nama : Evan G. Sumbayak
NPM : 0806339080
Program Studi : Teknik Komputer
Judul Skripsi : Rancang Bangun dan Analisis
Antarmuka Sistem Pemantau dan
Peringatan Polusi Udara pada Ruang
Parkir Tertutup Berbasis FPGA Xilinx
Spartan 3E

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 : Ir. A. Endang Sriningsih M.T., Si. (.....)
Penguji : Dr. Ir. Anak Agung Putri Ratna M.Eng. (.....)

Ditetapkan di : Depok
Tanggal : 22 Juni 2012

KATA PENGANTAR

Puji syukur saya panjatkan kehadirat Tuhan Yang Maha Kuasa, karena atas karunia-Nya lah saya dapat menyelesaikan skripsi ini. Saya menyadari bahwa skripsi ini tidak akan dapat diselesaikan tanpa bantuan dari berbagai pihak. Mulai dari proses pembelajaran, analisis 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 saya yang telah memberikan bantuan dukungan materil, moral serta doa.
3. Rekan saya April dan Vicky yang telah mendukung dan membantu menyelesaikan skripsi ini.
4. Junita H. Situmorang yang telah membantu saya dalam penulisan skripsi ini.
5. Teman-teman kontrakan yang selalu siap membantu saya dalam segala hal.
6. Teman-teman dari Teknik Komputer angkatan 2008, yang tiada hentinya mendukung saya baik secara langsung maupun tidak langsung.
7. Rekan-rekan asisten dari Laboratorium Digital Departemen Teknik Elektro.
8. Responden yang sudah bersedia meluangkan waktunya untuk mengisi kuesioner yang saya buat.
9. Seluruh keluarga besar Civitas Akademika Departemen Teknik Elektro Fakultas Teknik Universitas Indonesia yang tidak dapat saya sebutkan satu persatu.

Akhir kata, semoga Tuhan Yang Maha Kuasa berkenan membalas kebaikan semua pihak yang telah membantu. Semoga Skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan.

Depok, 27 Juni 2012

Evan G. Sumbayak



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

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

Nama : Evan G. Sumbayak

NPM : 0806339080

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 Ekklusif (Non-exclusive Royalty Free Right)** Atas karya ilmiah saya yang berjudul:

“Rancang Bangun dan Analisis Antarmuka Sistem Pemantau dan Peringatan Polusi Udara pada Ruang Parkir Tertutup Berbasis FPGA”

Dengan Hak Bebas Royalti Nonekskulif ini Universitas Indonesia Berhak menyimpan, mengalihmediakan/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan mempublikasikan tugas akhir 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 : 28 Juni 2012

Yang menyatakan



(Evan G. Sumbayak)

ABSTRAK

Nama : Evan G. Sumbayak
Program Studi : Teknik Komputer
Judul : Rancang Bangun dan Analisis Antarmuka Sistem Pemantau dan Peringatan Polusi Udara pada Ruang Parkir Tertutup Menggunakan FPGA Xilinx Spartan 3E

Penelitian yang dilakukan pada skripsi ini adalah merancang, membuat dan menganalisis sistem pemantau dan peringatan polusi udara pada ruang parkir tertutup berbasis FPGA Xilinx Spartan 3E. Metode yang digunakan dalam pembuatan antarmuka ini mengikuti *System Development Life Cycle* (SDLC) dan mengikuti kaidah *Human Computer Interaction*. Bahasa yang digunakan adalah VHDL dengan *software* Xilinx ISE. Antarmuka sistem pemantau dan peringatan polusi ini menampilkan kadar CO dalam ppm, grafik, dan tingkatan kondisi polusi udara (aman, waspada, berbahaya).

Antarmuka sistem menggunakan layar CRT atau LCD dengan ukuran 1030x788 pixel melalui port VGA. Antarmuka yang dibuat dapat melakukan *update* ketika data masukan berubah. Tampilan kadar gas CO (ppm) akan diperbaharui setiap dua detik. Kecepatan menampilkan satu *frame* di layar monitor adalah 32,67 ms.

Kata kunci :

Area parkir tertutup, FPGA Xilinx Spartan 3E, *port VGA*.

ABSTRACT

Name : Evan G. Sumbayak
Study Program : Computer Engineering
Title : Development and Analysis Interface of Air
Pollution Monitoring and Warning System at
Closed Parking Area Using FPGA Xilinx Spartan
3E

Research conducted in this thesis is to design, create and analyze the system of monitoring and warning of air pollution at the closed parking area based on Xilinx Spartan 3E FPGA. The method used in the design of this interface follows the System Development Life Cycle (SDLC) and the rules of Human Computer Interaction. The language used is VHDL with Xilinx ISE software. The interface pollution monitoring and warning system displays CO levels in ppm, graphics, and levels of air pollution conditions (safe, alert, dangerous).

The Interface systems use CRT screens with LCD or with size 1030x788 pixel through the VGA port. The interface that can be made an update when the data input changes. Display of levels CO gas (ppm) will be updated every two seconds. Speed of display a single frame on the screen is 32.67 ms.

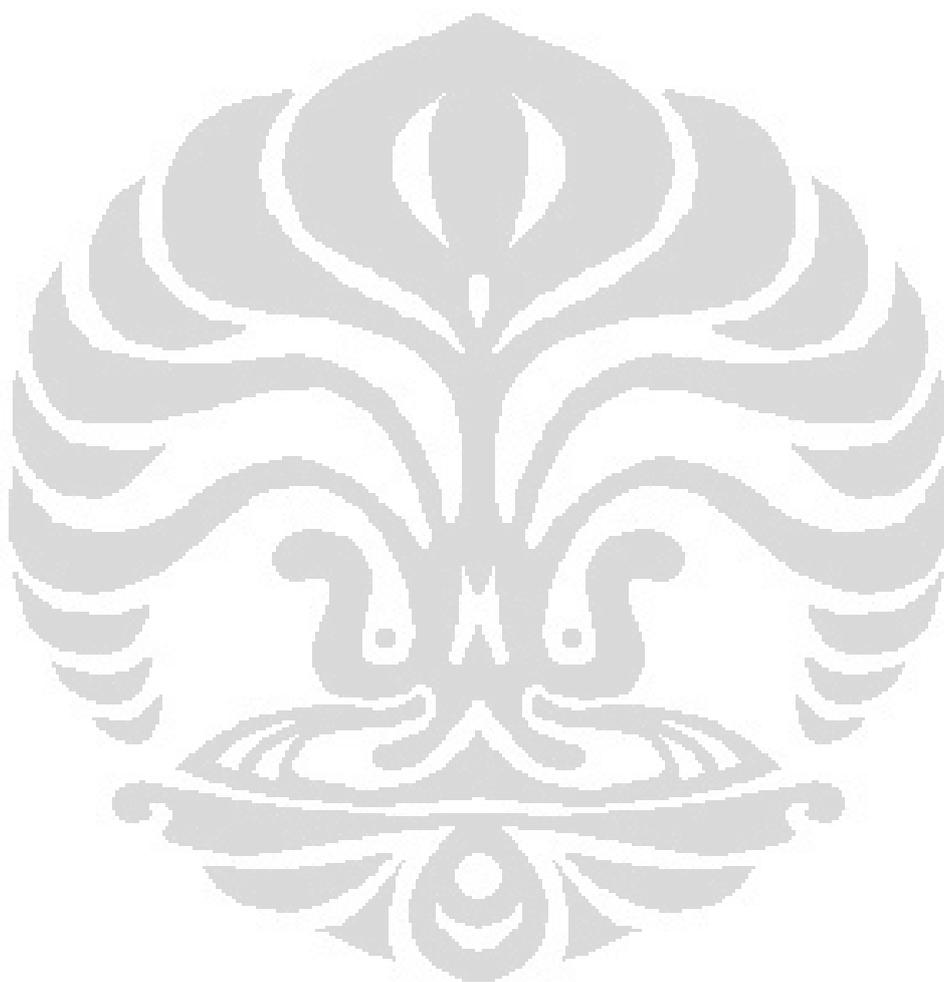
Key words :
Closed Parking Area, FPGA Xilinx Spartan 3E, VGA port.

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS.....	iii
HALAMAN PENGESAHAN.....	iv
KATA PENGANTAR	v
ABSTRAK.....	viii
ABSTRACT	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Tujuan Penelitian.....	2
1.3 Pembatasan Masalah	2
1.4 Metodologi Penelitian	2
1.5 Sistematika Penulisan.....	3
BAB II FPGA DAN POLUSI UDARA.....	5
2.1 Komposisi Udara Normal.....	5
2.2 Pencemar Udara.....	6
2.3 Pencemar Udara dalam Ruang Parkir Tertutup.....	8
2.4 FPGA (Field Programmable Gate Array).....	8
2.5 Arsitektur FPGA.....	9
2.6 FPGA SPARTAN 3E.....	10
2.6.1 Slide Switches.....	12
2.6.2 VGA Display Port.....	13
2.6.3 Clock Source.....	14
2.7 VHDL (Very High speed integrated Hardware Description Language).....	15
2.7.1 Library.....	16
2.7.2 Entity.....	18
2.7.3 Architecture.....	19
2.8 Perlengkapan Sistem	19

2.8.1	Layar CRT	19
2.8.2	Sensor.....	20
2.8.3	Exhaust fan.....	20
2.8.4	Buzzer	20
2.8.5	LED.....	20
2.9	Interaksi Manusia dan Komputer Melalui Penglihatan	21
BAB III PERANCANGAN ANTARMUKA SISTEM PEMANTAU DAN PERINGATAN POLUSI UDARA PADA RUANG PARKIR TERTUTUP BERBASIS FPGA		
22		
3.1	System Requirement.....	23
3.2	Desain.....	24
3.2.1	Desain Hardware.....	24
3.2.2	Skematik Diagram Hardware.....	24
3.2.3	Algoritma	25
3.2.4	Perancangan Tampilan pada Layar monitor	29
BAB IV IMPLEMENTASI, PENGUJIAN DAN ANALISIS ANTARMUKA SISTEM PEMANTAU DAN PERINGATAN POLUSI UDARA.....		
31		
4.1	Mengatur Gambar.....	31
4.1.1	Mengatur Letak Gambar di Layar.....	31
4.1.2	Mengatur Gambar Statis pada Monitor.....	33
4.1.3	Mengatur Gambar Dinamis pada Monitor	35
4.2	Skenario Pengujian.....	38
4.2.1	Tampilan pada Layar Monitor	38
4.2.2	Sistem mulai dari FPGA sampai tampil di Monitor	42
4.3	Data Pengujian.....	43
4.3.1	Tampilan pada layar Monitor.....	43
4.3.2	Sistem mulai dari FPGA sampai tampil di Monitor	44
4.4	Analisis.....	47
4.4.1	Analisis Penilaian Responden.....	47
4.4.2	Analisis Sistem mulai dari FPGA sampai tampil di Monitor	48
BAB V KESIMPULAN.....		
50		
DAFTAR ACUAN.....		
51		

DAFTAR PUSTAKA	52
LAMPIRAN	53



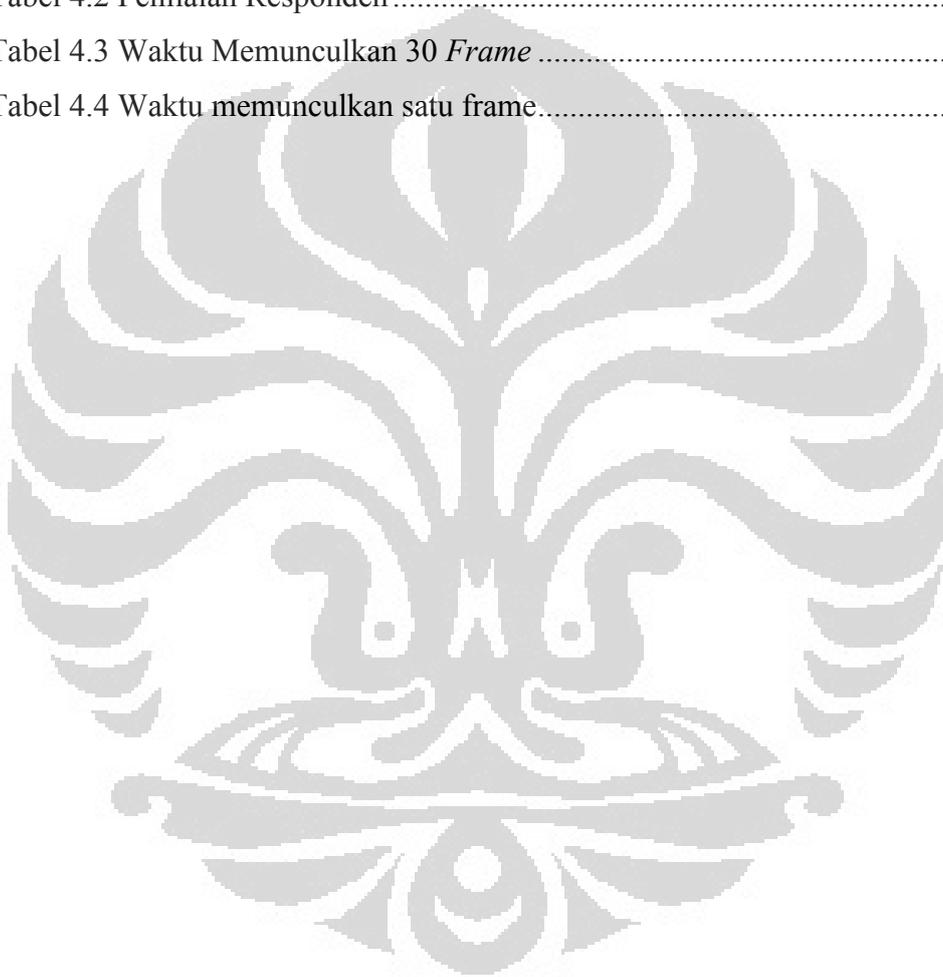
DAFTAR GAMBAR

Gambar 2.1 Arsitektur FPGA[7].....	9
Gambar 2.2 Logic cell[8]	10
Gambar 2.3 FPGA Spartan 3E[9]	11
Gambar 2.4 Arsitektur IC FPGA Spartan 3E[8]	12
Gambar 2.5 Empat Saklar Spartan 3E (four slide switches)[9]	12
Gambar 2.6 UCF constrain untuk empat saklar spartan 3E[9]	13
Gambar 2.7 VGA Connections pada FPGA Spartan 3E[9]	13
Gambar 2.8 Clock Input [9]	14
Gambar 2.9 Ucf <i>clock source</i> [9]	15
Gambar 2.10 Arsitektur VHDL[10].....	16
Gambar 2.11 <i>Syntax</i> deklarasi library	16
Gambar 2.12 <i>Syntax</i> deklarasi package ieee, std, dan work.....	17
Gambar 2.13 <i>Syntax</i> untuk membuat package dan cara memanggilnya	18
Gambar 2.14 <i>Syntax</i> untuk entity	18
Gambar 2.15 <i>Syntax</i> Architecture	19
Gambar 2.16 Pola menampilkan gambar pada CRT	19
Gambar 2.17 Sensor MQ-7 [11].....	20
Gambar 3.1 Layout area parkir tertutup	22
Gambar 3.2 Use Case Diagram	23
Gambar 3.3 Skematik Diagram	25
Gambar 3.4 Flowchart cara kerja sistem.....	26
Gambar 3.5 Pseudocode cara kerja sistem	27
Gambar 3.6 Sequence Diagram.....	28
Gambar 3.7 Tampilan saat kondisi aman	29
Gambar 3.8 Tampilan saat kondisi waspada.....	30
Gambar 3.9 Tampilan saat kondisi berbahaya	30
Gambar 4.1 Posisi gambar di layar	32
Gambar 4.2 Pseudocode mengatur posisi tulisan.....	33

Gambar 4.3 Array untuk gambar FPGA	33
Gambar 4.4 Tampilan <i>constant</i> char_tulisan	34
Gambar 4.5 <i>line code</i> untuk menampilkan tulisan.....	34
Gambar 4.6 Tampilan gambar statis	35
Gambar 4.7 Pseudocode menampilkan titik pada grafik	36
Gambar 4.8 Pseudocode menampilkan kondisi	36
Gambar 4.9 Pseudocode menghitung digit ratusan.....	37
Gambar 4.10 Pseudocode menampilkan puluhan	37
Gambar 4.11 Pseudocode menampilkan satuan.....	38
Gambar 4.12 Tampilan saat kondisi aman.....	40
Gambar 4.13 Tampilan saat kondisi waspada.....	41
Gambar 4.14 Tampilan saat kondisi berbahaya	41
Gambar 4.15 Tampilan pada layar LCD	42
Gambar 4.16 Grafik rata-rata nilai responden terhadap pernyataan	44
Gambar 4.17 Pengukuran <i>response time</i> dengan <i>oscilloscope</i>	45

DAFTAR TABEL

Tabel 2.1 Pengaruh CO pada Tubuh[4]	7
Tabel 2.2 Pengaruh Kadar CO terhadap kondisi tubuh[5].....	7
Tabel 2.3 Output warna [9]	14
Tabel 4.1 <i>Input</i> sesuai posisi saklar.....	42
Tabel 4.2 Penilaian Responden	43
Tabel 4.3 Waktu Memunculkan 30 <i>Frame</i>	45
Tabel 4.4 Waktu memunculkan satu frame.....	46



BAB I

PENDAHULUAN

1.1 Latar Belakang

Populasi penduduk yang banyak di kota besar menyebabkan padatnya daerah permukiman atau perkantoran. Terbatasnya lahan di kota-kota besar membuat perubahan yang signifikan dalam cara membangun bangunan, baik gedung perkantoran, pusat perbelanjaan maupun tempat tinggal. Bangunan yang besar diharapkan dapat menampung banyak orang, dan transportasi orang tersebut antara lain kendaraan umum dan kendaraan pribadi. Dengan menggunakan prinsip menggunakan lahan yang sempit untuk menampung banyak orang, dibangunlah tempat parkir di ruang bawah tanah gedung dengan beberapa tingkat (lantai) biasanya tergantung besarnya gedung atau banyak orang yang mengakses gedung tersebut. Pembangunan ruang bawah tanah untuk tempat parkir dimaksudkan agar luas lahan untuk membangun gedung dapat dimaksimalkan.

Ruangan tanpa ventilasi udara tentu harus mendapat perhatian khusus dalam hal sirkulasi udara. Untuk memanfaatkan ruang tertutup seperti ini harus ada jaminan terjadi sirkulasi udara yang baik terutama bila digunakan sebagai tempat parkir. Jika tidak ada sirkulasi udara, asap yang keluar dari knalpot kendaraan bermotor akan berkumpul di dalam ruangan, membuat polusi udara di tempat tersebut menjadi tinggi sehingga berbahaya bagi pengguna kendaraan bermotor.

Ada beberapa senyawa yang berpengaruh dalam pencemaran udara, yang paling banyak berpengaruh adalah Karbon monoksida (CO), Nitrogen Oksida (NO_x), Belerang Oksida (SO_x), Hidro Karbon(HC), Timah hitam (Pb). Tempat parkir di ruang tertutup di Indonesia belum dilengkapi dengan deteksi polusi udara. Kesadaran bahwa polusi sangat berbahaya bagi kesehatan belum dirasa penting bagi banyak orang.

Penggunaan ruang tertutup sebagai tempat parkir tentu akan menjadi tempat menampung pembuangan kendaraan bermotor. Kadar polusi di ruang parkir tertutup ini berpotensi lebih tinggi daripada polusi di ruang terbuka. Hal ini

tentu sangat berbahaya bagi pengunjung jika mereka tidak tahu udara di ruangan beracun yang dapat menyebabkan masalah kesehatan bahkan kematian.

Untuk itulah diperlukan sistem yang dapat mendeteksi kandungan udara beracun sehingga pengunjung tahu bahwa polusi udara masih dalam tingkat aman atau sangat berbahaya. Membuat sistem pendeteksi polusi udara menjadi mungkin dengan berkembangnya teknologi VLSI. Salah satunya adalah FPGA. FPGA dapat diprogram sesuai dengan kebutuhan.

Pengunjung juga perlu mengetahui secara pasti konsentrasi gas beracun dalam udara, karena itu perlu dibuat antarmuka sistem sehingga pengunjung dalam area parkir tertutup dapat melihat tingkat polusi udara dan dapat memperkirakan tingkat polusi berikutnya dengan melihat apakah kadar gas beracun tersebut cenderung naik atau cenderung turun.

1.2 Tujuan Penelitian

Penelitian ini bertujuan untuk

- Rancang bangun antarmuka sistem pendeteksi dan peringatan polusi gas CO pada area parkir tertutup.
- Menghitung dan menganalisis kecepatan respon menampilkan antarmuka dari data pada sensor ke layar monitor menggunakan port VGA.

1.3 Pembatasan Masalah

Dalam penelitian ini, FPGA yang digunakan adalah XILINX Spartan 3E. Komponen peringatan yang ditampilkan adalah tingkatan kondisi polusi (aman, waspada, dan berbahaya). Komponen pemantau yang ditampilkan adalah grafik kadar gas CO dan nilai kadar gas CO dalam ppm. Alat antarmuka yang digunakan adalah layar monitor CRT dan LCD menggunakan port VGA.

1.4 Metodologi Penelitian

Metodologi penelitian yang digunakan mengacu pada *System Development Life Cycle* (SDLC), antara lain:

a. Inisiasi ide

Menentukan tema yang akan dikembangkan berkaitan dengan isu dalam masyarakat.

b. *System Requirement*

Mendefinisikan fitur-fitur yang diperlukan dalam sistem yang akan dibuat.

c. Desain

Merangkum fitur sistem dan menerjemahkannya ke dalam bentuk desain sistem.

d. Pengembangan dan Implementasi

Menerjemahkan desain sistem kedalam bentuk bahasa mesin sehingga sistem dapat bekerja sesuai kebutuhan.

e. Uji coba

Melakukan uji coba terhadap fitur-fitur yang ada dalam sistem.

f. Pemeliharaan

Menjaga sistem dari kemungkinan yang dapat merusak sistem dan melakukan pengembangan jika diperlukan.

1.5 Sistematika Penulisan

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

BAB I PENDAHULUAN

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

BAB II FPGA DAN POLUSI UDARA

Bab ini berisi tentang penjelasan udara yang baik, udara tercemar, *Field Programmable Gate Array* beserta komponennya, VHDL, display, sensor.

BAB III PERANCANGAN ANTARMUKA SISTEM PEMANTAU DAN PERINGATAN POLUSI UDARA PADA RUANG PARKIR TERTUTUP BERBASIS FPGA

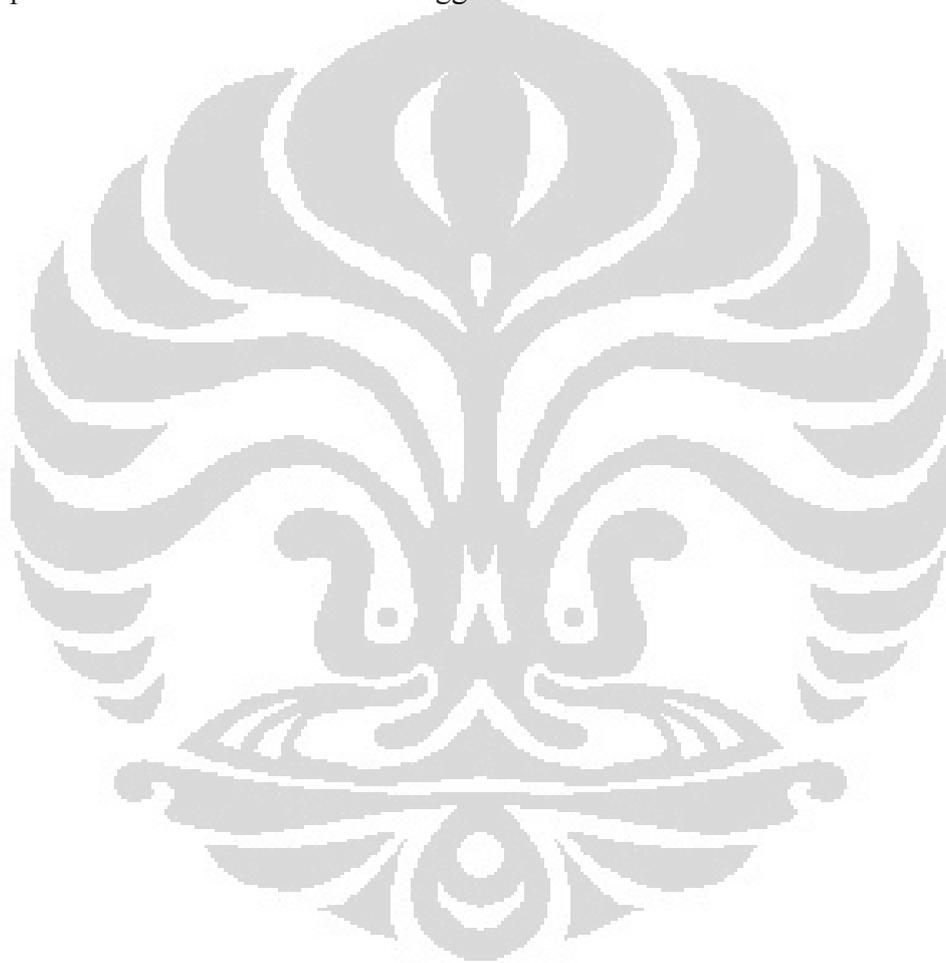
Bab ini berisi tentang kebutuhan pengguna, desain sistem, uji coba sesuai dengan metode penelitian yang sudah dipaparkan sebelumnya.

BAB IV IMPLEMENTASI, PENGUJIAN DAN ANALISIS ANTARMUKA SISTEM PEMANTAU DAN PERINGATAN POLUSI UDARA

Bab ini berisi tentang implementasi desain tampilan ke dalam bahasa VHDL dan pengujian tampilan yang *user friendly* dengan cara meminta responden untuk mengisi kuesioner.

BAB IV KESIMPULAN

Bab ini merupakan kesimpulan yang didapat dari pembahasan sistem pemantau polusi udara berbasis FPGA menggunakan sensor CO.



BAB II

FPGA DAN POLUSI UDARA

Udara adalah kumpulan/campuran gas, yang terbanyak adalah nitrogen dan oksigen. Oksigen sangat penting untuk mendukung kehidupan makhluk hidup dan memungkinkan terjadinya pembakaran bahan bakar. Nitrogen merupakan penyubur tanaman. Bakteri menggunakan nitrogen dari udara untuk menyuburkan tanah. Udara juga melindungi bumi dari radiasi berbahaya yang berasal dari ruang angkasa[1].

Dalam Udara juga terdapat gas yang berbahaya bagi manusia jika kadarnya terlalu banyak. Sebagai contoh gas Karbon Monoksida (CO), gas ini akan menjadi gas yang mematikan jika kandungannya dalam udara banyak. Untuk memantau kadar gas dalam udara dapat digunakan sensor. Data dari sensor harus diproses agar dapat diterjemahkan ke dalam bahasa manusia. Untuk memproses data tersebut dapat digunakan FPGA.

2.1 Komposisi Udara Normal

Komposisi udara bersih sangat bervariasi dari satu tempat ke tempat lain. Rata-rata persentase gas dalam udara bersih dan kering adalah[1]:

- Nitrogen 78 %
- Oksigen 20,8 %
- Argon 0,9 %
- Karbon dioksida 0,03 %
- Gas lain 0,27% (meliputi helium, neon, krypton, xenon, hidrogen, dan metan).

Saat seseorang menghirup 1 liter udara maka banyaknya udara yang dihirup dan dikeluarkan adalah sebagai berikut:

- Dihirup
Oksigen 208 cm³
Nitrogen 780 c

Karbon dioksida 0.3 cm^3

Gas lain 11.7 cm^3

- Dihembuskan

Oksigen 170 cm^3

Nitrogen 780 cm^3

Karbon dioksida $38,3 \text{ cm}^3$

Gas lain $11,7 \text{ cm}^3$

2.2 Pencemar Udara

Pencemaran udara adalah kehadiran satu atau lebih substansi fisik, kimia, atau biologi di atmosfer dalam jumlah yang dapat membahayakan kesehatan manusia, hewan, dan tumbuhan, mengganggu estetika dan kenyamanan, atau merusak properti[2].

Parameter-parameter dasar untuk Index Standar Pencemar Udara (ISPU) adalah[3]:

1. Partikulat (PM10)
2. Sulfur Dioksida (SO_2)
3. Carbon Monoksida (CO)
4. Ozon (O_3)
5. Nitrogen Dioksida (NO_2)

Dalam penelitian ini, digunakan parameter Karbon monoksida sebagai sampel pencemar udara. CO dipilih sebagai sampel karena gas ini tidak berasa, tidak berbau, tidak berwarna pada suhu kamar sehingga tidak dapat dideteksi dengan indra penciuman. Selain itu gas CO juga banyak dihasilkan kendaraan bermotor sebagai akibat pembakaran tidak sempurna.

Gas CO juga sangat mematikan. Hal ini disebabkan karena hemoglobin dalam sel darah merah manusia 200 kali lebih kuat mengikat CO daripada Oksigen[4].

Gejala-gejala klinis dari saturasi darah oleh karbon monoksida ditunjukkan oleh Tabel 2.1

Tabel 2.1 Pengaruh CO pada Tubuh[4]

Konsentrasi CO dalam darah	Gejala-gejala
Kurang dari 20%	Tidak ada gejala
20%	Nafas menjadi sesak
30%	Sakit kepala, lesu, mual, nadi dan pernafasan meningkat
30%-40%	Sakit kepala berat, kebingungan, hilang daya ingat, lemah, hilang daya koordinasi
40%-50%	Kebingungan makin meningkat, setengah sadar
60%-70%	Tidak sadar, kehilangan daya mengontrol faeces dan urin
70%-80%	Koma, nadi menjadi tidak teratur, kematian karena kegagalan pernafasan

Pengaruh gas CO dalam tubuh manusia juga dapat dilihat melalui Tabel 2.2. Tabel ini memaparkan pengaruh CO berdasarkan kadarnya dalam udara dan lama paparannya serta efek yang ditimbulkan pada tubuh manusia.

Tabel 2.2 Pengaruh Kadar CO terhadap kondisi tubuh[5]

Ppm CO	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.

2.3 Pencemar Udara dalam Ruang Parkir Tertutup

Kesadaran akan bahaya gas CO membuat antisipasi terhadap keracunan gas ini sangat diperlukan. Gas CO dihasilkan oleh pembakaran tidak sempurna, jadi gas CO juga dihasilkan oleh kendaraan bermotor yang tidak mengalami pembakaran sempurna. Tempat parkir ruang tertutup menjadi tempat yang sangat rawan dalam pencemaran gas CO ini. Hasil pembuangan dari kendaraan bermotor akan berkumpul di dalam ruangan parkir sehingga konsentrasi gas CO lebih tinggi di dibandingkan di ruang terbuka.

Ruang parkir tertutup bisa menjadi tempat yang sangat berbahaya bagi pengunjung maupun petugas di ruang parkir, oleh karena itu dibutuhkan beberapa cara untuk mengantisipasi tingginya kadar CO di ruang parkir tertutup seperti memperbaiki sistem sirkulasi udara dan menyediakan sistem yang dapat memantau kadar gas CO di ruang parkir tertutup.

2.4 FPGA (Field Programmable Gate Array)

FPGA adalah alat elektronika yang terbuat dari bahan semikonduktor yang mengandung interkoneksi antara gerbang logika yang dapat di program sesuai dengan kebutuhan pengguna. FPGA memiliki komponen gerbang terprogram meliputi gerbang logika biasa (AND, OR, XOR, NOT) maupun jenis fungsi matematis (*decoder, adder, subtractor, multiplier, dll*)[6]. Blok komponen di dalam FPGA juga bisa mengandung elemen memori (*register*) mulai dari *flip-flop* sampai pada RAM (*Random Access Memory*)[7].

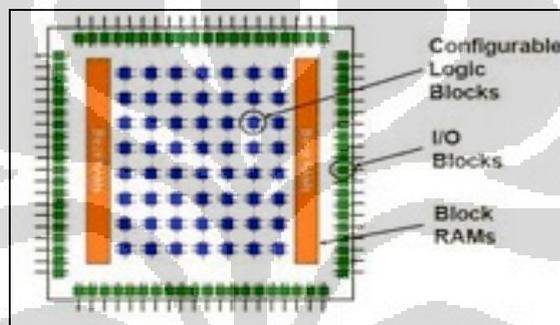
Secara umum, FPGA akan lebih lambat dibandingkan jenis chip lain seperti pada chip *Application Specific Integrated Circuit* (ASIC)[7]. Hal ini karena FPGA menggunakan daya yang besar dan bentuk desain yang lebih kompleks. Untuk mengkonfigurasi FPGA digunakan *Hardware Description Language* (HDL). HDL lebih mudah digunakan dalam program skala besar karena bahasa pemrograman ini memiliki visualisasi desain yang lebih sederhana.

FPGA dikembangkan sejak tahun 1984 oleh perusahaan Xilinx yang berbasis di San Jose oleh Ross Freeman dan Bernard Vonderschmitt sebagai pendiri perusahaan Xilinx. FPGA berhasil di komersialkan pada tahun 1985 dengan tipe XC2064. Selanjutnya, FPGA mulai diproduksi oleh beberapa

perusahaan lain misalnya, Altera, Lattice, dan Quicklogic. Namun perusahaan yang berhasil mendominasi FPGA di seluruh dunia adalah Xilinx dan Altera. Sementara itu perusahaan-perusahaan lain (Lattice, Actel dan Quicklogic) merupakan perusahaan yang lebih kecil dan mempunyai pasar khusus[6].

2.5 Arsitektur FPGA

Secara umum bagian dalam IC FPGA terdiri atas I/O Block (IOB), *Configurable Block* (CLB), *Programmable Interconnect*. Sumber daya internal IC FPGA terdiri dari matriks *Configurable Logic Block* yang dikelilingi oleh I/O Block (IOB)[7] dapat dilihat pada Gambar 2.1.



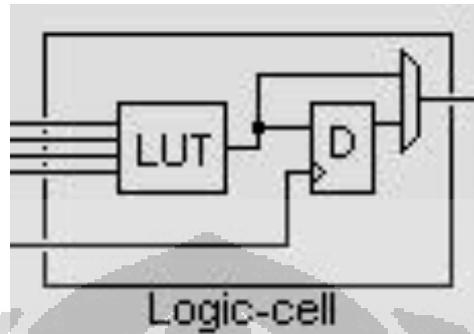
Gambar 2.1 Arsitektur FPGA[7]

CLB berfungsi untuk mengimplementasikan fungsi-fungsi logika yang dibuat oleh user, I/O Block berfungsi sebagai *interface* antara eksternal pin dari *device* dan internal user logic, dan *programmable Interconnect* yang merupakan bagian yang berisi wire segment dan *programmable switches* yang berfungsi untuk menghubungkan antara CLB yang satu dengan CLB yang lain[7].

Perbedaan FPGA dan mikrokontroler adalah FPGA mempunyai struktur dari perangkat keras yang dapat diubah konfigurasinya, misalnya pada banyaknya I/O yang digunakan atau jumlah register yang dapat dikonfigurasi sesuai dengan yang dibutuhkan pemakai. Mikrokontroler memang dapat diprogram ulang, namun mikrokontroler hanya dapat melakukan sesuatu berdasarkan konfigurasi dari perangkat keras yang tetap. FPGA memberikan kebebasan kepada pengguna untuk merancang perangkat keras yang akan digunakan.

Pada awalnya, FPGA tersusun dari sebuah bagian yang bernama "*logic-cell*" (Logic Blocks) seperti pada Gambar 2.2. Kumpulan dari logic cell ini berjumlah ratusan bahkan ribuan dan membentuk suatu fungsi yang kompleks.

Sebuah *logic cell* pada dasarnya terdiri dari *LookUp Table* (LUT), *D flip-flop*, dan sebuah multiplexer 2 ke 1.



Gambar 2.2 Logic cell[8]

LookUp Table merupakan sejenis RAM (*Random Access Memory*) yang berkapasitas kecil. Pada awalnya, FPGA memiliki 4 *input LookUp Table* (LUT). Namun beberapa tahun kemudian perusahaan pembuat FPGA mulai memproduksi FPGA dengan 6 *input LUT* dengan alasan untuk meningkatkan performa. Di dalam FPGA, LUT memegang peranan penting dalam proses implementasi fungsi-fungsi logika[7].

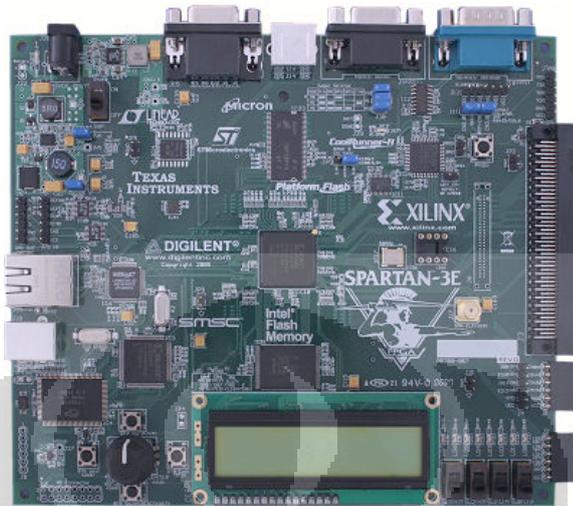
Flip-flop adalah suatu rangkaian sel biner yang memiliki dua buah *output* yang saling berkebalikan keadaannya (0 atau 1). Dalam FPGA terdapat sebuah jenis *flip-flop* yaitu *D flip-flop*. Di dalam logic cell ini, *D flip-flop* berfungsi sebagai rangkaian logika sekunsial dimana di dalamnya terdapat rangkaian memori dan pewaktu. Keluaran dari *D flip-flop* bergantung terhadap sinyal pewaktu yang masuk.

Multiplexer adalah piranti digital yang bekerja sebagai saklar yang menghubungkan data dari n masukan ke sebuah keluaran. Dalam *logic cell* terdapat multiplexer 2 ke 1, artinya multiplexer memilih dari dua masukan sehingga menghasilkan satu keluaran saja.

2.6 FPGA SPARTAN 3E

FPGA Spartan 3E pada Gambar 2.3 merupakan salah satu dari keluarga FPGA yang diproduksi oleh XILINX. Spartan 3E menggunakan bahasa

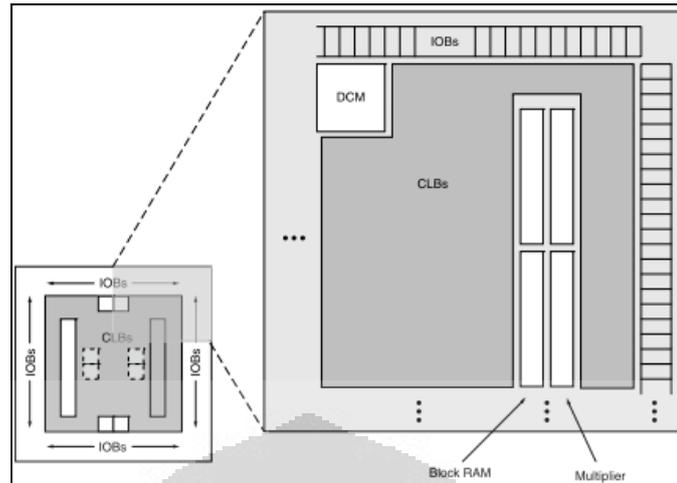
pemrograman VHDL (*Very high Hardware Description Language*) dan menggunakan *software development kit* dari Xilinx dalam memprogramnya.



Gambar 2.3 FPGA Spartan 3E[9]

Secara umum bagian dalam IC FPGA terdiri atas I/O Block (IOB), *Configurable Block* (CLB), *Programmable Interconnect* dan pada FPGA Spartan 3E, arsitektur IC nya dapat dilihat pada Gambar 2.4. Bagian dalam IC FPGA Spartan 3E terdiri dari 5 bagian.

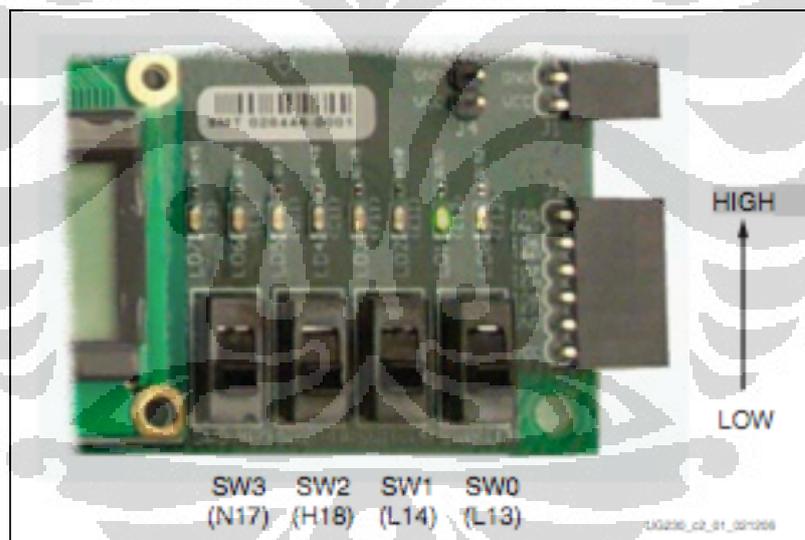
- a. *Configurable Logic Block* (CLB) yang terdiri dari *Look-Up Tables* (LUTs) yang merupakan sejenis memory dengan kapasitas kecil sehingga CLB ini dapat digunakan sebagai tempat penyimpanan data dan melakukan fungsi logika.
- b. *Input/Output Blocks* mengontrol aliran data antara I/O dan logika internal.
- c. *Block RAM* menyediakan tempat penyimpanan data dual port 18 Kbit blok.
- d. *Block Multiplier* memiliki tugas utama untuk melakukan perkalian antara dua angka biner 18 bit.
- e. *Digital Clock Manager* (DCM) menyediakan kalibrasi *delay*, *multiplying*, *dividing*, *phase shift*.



Gambar 2.4 Arsitektur IC FPGA Spartan 3E[8]

FPGA Spartan 3E memiliki beberapa komponen yang terintegrasi antara lain *Switches*, *Buttons*, dan monitor.

2.6.1 Slide Switches



Gambar 2.5 Empat Saklar Spartan 3E (four slide switches)[9]

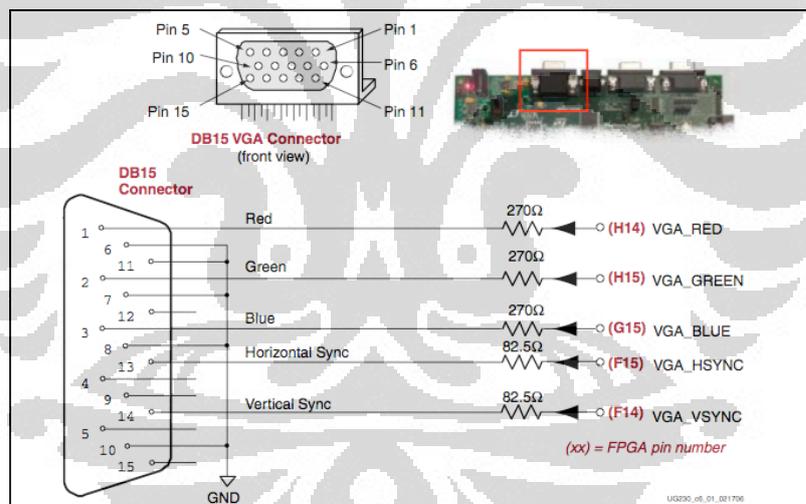
Pada Spartan 3-E Kit Board, slide switches terletak di sudut kanan bawah board seperti yang ditunjukkan Gambar 2.5. Ketika posisi saklar ON, saklar menghubungkan pin FPGA ke 3,3V (*logic High*). Ketika posisi saklar OFF, pin FPGA terhubung dengan *ground* (*logic Low*). Pada Gambar 2.6 diperlihatkan file UCF untuk *four slide switches*, dengan standar I/O yang digunakan, pullup resistor tidak dibutuhkan namun pullup resistor mendefinisikan hasil ketika saklar dalam masa peralihan (*transition*)[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 2.6 UCF constrain untuk empat saklar spartan 3E[9]

2.6.2 VGA Display Port

FPGA Spartan 3E menghasilkan lima sinyal VGA melalui resistor. Tiap jalur warna memiliki resistor yang berbeda, dengan satu bit untuk VGA_RED, VGA_GREEN, dan VGA_BLUE. Resistor dengan tahanan 75Ω digabungkan dengan tahanan kabel VGA untuk memastikan sinyal warna berkisar antara 0V-0,7V. VGA_HSYNC dan VGA_VSYNC menggunakan LVTTTL atau LVCMOS33 I/O standar. Gambar 2.7 menunjukkan gambar port VGA pada FPGA board dan penjelasan tiap pin pada port VGA.



Gambar 2.7 VGA Connections pada FPGA Spartan 3E[9]

Warna pada output diatur melalui output yang dikeluarkan pada pin Red, Green, Blue. Tabel 2.3 menunjukkan warna yang dapat dibentuk dengan mengatur ketiga pin ini 'high' atau 'low'.

Tabel 2.3 Output warna [9]

VGA_RED	VGA_GREEN	VGA_BLUE	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

2.6.3 Clock Source

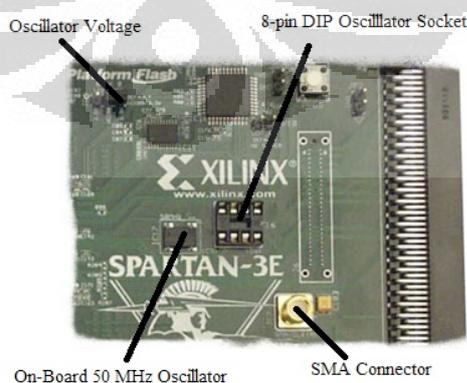
Spartan-3E Starter Kit board memiliki 3 sumber daya *clock* seperti yang ditunjukkan pada Gambar 2.8, dimana setiap *input clock* terhubung dengan *Digital Clock Manager*.

- 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.



Gambar 2.8 Clock Input [9]

- SMA *Clock* Input atau Output Connector

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

Untuk menghitung *clock* pada VGA, perlu diketahui banyaknya piksel horizontal dan vertikal yang akan digunakan dan *refresh rate* pada layar. Misalnya layar memiliki piksel horizontal 800 piksel/baris, vertikal 600 baris/layar dan *refresh rate* 60 layar/ detik, maka *clock* yang digunakan adalah

$$Clock = 800 \frac{\text{piksel}}{\text{baris}} \times 600 \frac{\text{baris}}{\text{layar}} \times 60 \frac{\text{layar}}{\text{detik}} \quad (2.1)$$

$$Clock = 28,8M \frac{\text{piksel}}{\text{detik}}$$

Pada Konfigurasi konstrain UCF *clock source*, ditunjukkan Gambar 2.9.

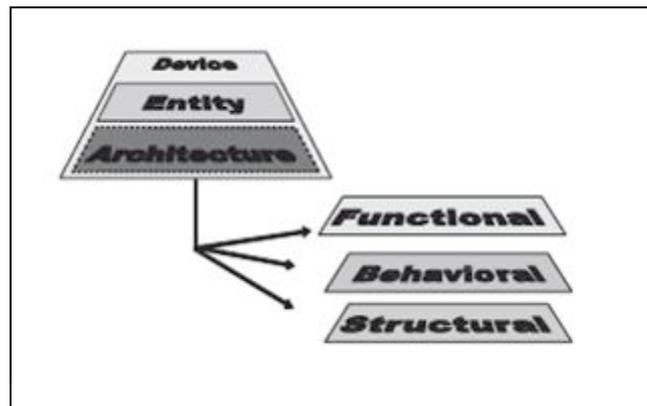
```
NET "CLK_50MHZ" LOC = "C9" | IOSTANDARD = LVCMOS33 ;
NET "CLK_SMA" LOC = "A10" | IOSTANDARD = LVCMOS33 ;
NET "CLK_AUX" LOC = "B8" | IOSTANDARD = LVCMOS33 ;
```

Gambar 2.9 Ucf *clock source* [9]

2.7 VHDL (Very High speed integrated Hardware Description Language)

Pemograman FPGA dilakukan dengan menggunakan bahasa pemograman VHDL. Selain VHDL dapat juga digunakan kode verilog atau systemC, tergantung pada perancang dan ketersediaan software pendukung. Perbedaan VHDL dan verilog terletak pada konteks kedua bahasa. Verilog berasal dari tradisi “bottom-up” yang sering digunakan dalam industri IC. Sedangkan VHDL dikembangkan pada perspektif “top-down”. Perbedaan jelas kedua bahasa ini terlihat pada syntax dasar dan metode dari keduanya[10].

VHDL memiliki keunggulan untuk menggunakan gabungan level dari model yang memiliki arsitektur yang berbeda seperti pada Gambar 2.10.



Gambar 2.10 Arsitektur VHDL[10]

Kode verilog juga memiliki konsep yang sama walaupun hanya terdapat dalam sebuah module. Pembagian sebuah model ke dalam beberapa bagian juga merupakan keunggulan lain dari VHDL. Misalnya, bagian *interface* (dikenal sebagai 'entity') dan bagian *behaviour* (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. VHDL kode minimal terdiri dari tiga bagian, yaitu:

- Library : Terdiri dari daftar library yang digunakan dalam desain
- Entity : Spesifikasi dari pin I/O pada sirkuit
- Architecture : Berisi VHDL code, yang menggambarkan cara kerja sirkuit

2.7.1 Library

Untuk mendeklarasikan library dibutuhkan minimal dua line code, yaitu nama library dan kata *use* seperti *syntax* pada Gambar 2.11.

```
LIBRARY nama_library;
USE nama_library.nama_package.bagian_package;
```

Gambar 2.11 Syntax deklarasi library

Biasanya ada tiga package, dari tiga library yang berbeda yang digunakan dalam desain VHDL, yaitu:

- *ieee.std_logic_1164* (dari library *ieee*),
- *standard* (dari library *std*),

- *work* (library *work*)

Syntax ntuk mendeklarasikan ketiga package diatas ditunjukkan pada Gambar 2.12 berikut:

```

LIBRARY ieee;           --Tanda titik koma (;)
berarti
USE ieee.std_logic_1164.all;  --akhir dari statement atau
                               --deklarasi

LIBRARY std;
USE std.standard.all;       --Tanda (--) digunakan
untuk
                               --memberikan komentar

LIBRARY work;
USE work.all;

```

Gambar 2.12 *Syntax* deklarasi package *ieee*, *std*, dan *work*

Library *std* dan *work* pada *syntax* diatas dibuat tersembunyi pada umumnya, jadi tidak perlu di deklarasikan lagi. Hanya library *ieee* yang harus dideklarasikan. Library *ieee* memiliki beberapa package, diantaranya:

- *std_logic_1164*: Berisi spesifikasi tentang STD_LOGIC dan STD_ULOGIC
- *std_logic_arith*: Berisi spesifikasi tentang SIGNED dan UNSIGNED tipe data dan berhubungan dengan operasi perbandingan dan aritmatika. Package ini juga berisi fungsi data konversi tipe data seperti; *conv_integer*, *conv_unsigned*, *conv_std_logic_vector*.
- *Std_logic_signed*: Berisi fungsi yang memperbolehkan operasi data *std_logic_vector* untuk ditampilkan sebagai data dengan tipe SIGNED.
- *Std_logic_unsigned*: Berisi fungsi yang memperbolehkan operasi data *std_logic_vector* untuk ditampilkan sebagai data dengan tipe UNSIGNED.

Pada program yang cukup besar kadang dibutuhkan package yang dibuat sendiri, hal ini dapat terjadi karena banyaknya data yang dimasukkan secara manual ke dalam program. Untuk mencegah program memiliki baris yang sangat panjang maka data tersebut dapat di deklarasikan dalam package seperti Gambar 2.13.

```

----- Package: -----
LIBRARY ieee;
USE ieee.std_logic_1164.all;
-----

PACKAGE Nama_package_data IS
TYPE vector_array IS ARRAY (NATURAL RANGE <>) OF
STD_LOGIC_VECTOR(7 DOWNT0 0);
END Nama_package_data;
-----

----- Main code: -----
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE work.Nama_package_data.all; -- package yang dibuat
-----

ENTITY mux IS
PORT (inp: IN VECTOR_ARRAY (0 TO 3);
... );
END mux;
... ;
-----

```

Gambar 2.13 Syntax untuk membuat package dan cara memanggilnya

2.7.2 Entity

Entity adalah daftar dari semua pin *input* dan output yang digunakan (PORTS) dalam sirkuit. *Syntax* untuk mendeklarasikan entity ditunjukkan pada Gambar 2.14. Mode dari signal dapat berupa IN, OUT, INOUT, atau BUFFER.

```

ENTITY nama_entity IS
  Port (
    Nama_port    : mode_signal tipe_signal;
    Nama_port    : mode_signal tipe_signal;
    ...);
END nama_entity;

```

Gambar 2.14 Syntax untuk entity

2.7.3 Architecture

Architecture adalah gambaran bagaimana program itu berjalan (berfungsi), *syntax* dari architecture ditunjukkan Gambar 2.15.

```

ARCHITECTURE nama_architecture OF nama_entity IS
    [deklarasi]
BEGIN
    (code)
END nama_architecture;

```

Gambar 2.15 Syntax Architecture

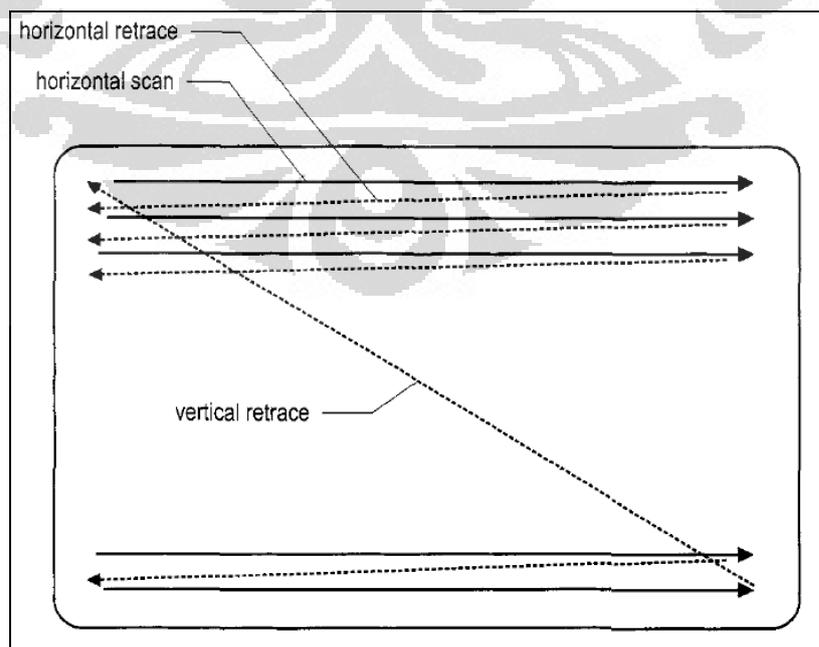
Pada bagian deklarasi di Gambar 2.15, dapat ditambahkan SIGNAL atau jenis data lain yang akan dibutuhkan pada program sesuai dengan kebutuhan dari *user*.

2.8 Perlengkapan Sistem

2.8.1 Layar CRT

Pada penelitian ini, layar yang digunakan untuk menampilkan antarmuka adalah layar CRT (Cathode Ray Tube).

Pada tampilan layar CRT, tampilan pada tabung menggunakan elektron yang ditembakkan. Elektron ditembakkan mulai dari kiri ke kanan secara horizontal dan dari atas ke bawah secara vertikal. Untuk lebih jelasnya pola penampilan gambar pada layar monitor dapat dilihat pada Gambar 2.16



Gambar 2.16 Pola menampilkan gambar pada CRT

2.8.2 *Sensor*

Sensor yang digunakan adalah MQ-7 CO Gas Sensor buatan Hanwei Electronic Co (Gambar 2.17). Sensor ini akan ditempatkan pada tempat yang diperkirakan dapat merepresentasikan kadar CO dalam ruangan, sensor akan dihubungkan dengan FPGA untuk mengolah data hasil pengukuran sensor.



Gambar 2.17 Sensor MQ-7 [11]

2.8.3 *Exhaust fan*

Exhaust fan berfungsi untuk menghisap udara di dalam ruangan untuk dibuang keluar ruangan dan pada saat yang bersamaan menggantikan udara dari dalam ruangan dengan udara dari luar ruangan. Alat ini digunakan untuk melakukan sirkulasi udara pada ruangan tertutup.

2.8.4 *Buzzer*

Buzzer merupakan perangkat yang digunakan untuk memberikan sinyal audio. Pada penelitian ini digunakan perangkat *buzzer* yang menghasilkan frekuensi tinggi yang peka pada telinga manusia karena digunakan sebagai alat untuk memberi peringatan.

2.8.5 *LED*

LED (*Light Emitting Diode*) digunakan untuk memberikan sinyal kepada manusia secara visual. LED yang digunakan adalah berwarna hijau untuk kondisi aman, kuning untuk kondisi waspada dan merah untuk kondisi berbahaya.

2.9 Interaksi Manusia dan Komputer Melalui Penglihatan

Manusia dapat melihat karena ada tahap penerimaan cahaya oleh mata, kemudian cahaya yang diterima mata diproses oleh otak[12]. Mata berfungsi untuk menerima cahaya dan mengubahnya menjadi sinyal elektrik sehingga dapat dikirim ke otak. Cahaya yang tampak oleh mata manusia berkisar antara 400nm-700nm yang terdiri dari 7 warna yaitu merah, jingga, kuning, hijau, biru, dan ungu[13]. Benda dapat menyerap dan memantulkan cahaya, cahaya yang dipantulkan adalah warna benda tersebut yang dapat kita lihat.

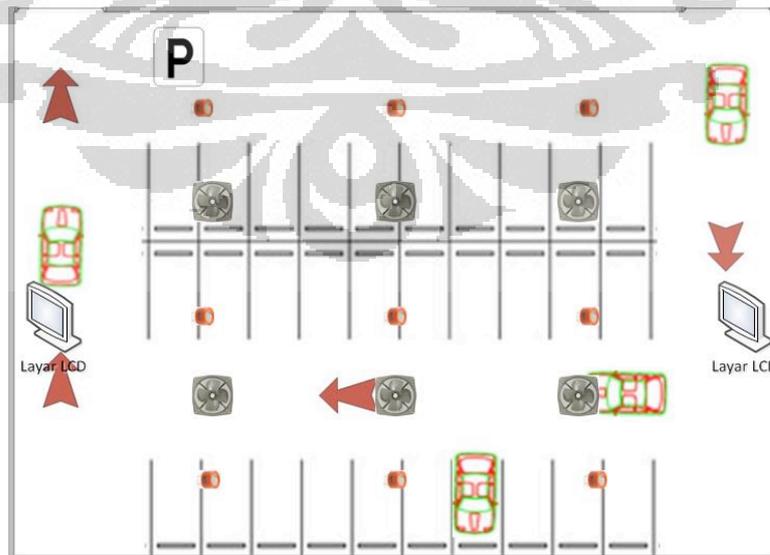
Dalam sistem peringatan selalu ada tiga warna yang sangat sering digunakan yaitu hijau, kuning, dan merah. Hijau adalah warna yang tenang dan sering dikaitkan dengan lingkungan. Warna kuning adalah warna yang ceria dan warna ini sering digunakan untuk mendapatkan perhatian orang. Karena itulah kuning sering digunakan sebagai tanda untuk berhati-hati misalnya dalam rambu lalu lintas[14]. Warna yang ketiga adalah warna merah. Merah adalah warna yang kuat, merah membantu seseorang untuk mengingat, mengoreksi bacaan dan membaca peringatan bahaya[15]. Merah membuat seseorang memberi perhatian yang lebih pada objek dengan warna tersebut.

BAB III

PERANCANGAN ANTARMUKA SISTEM PEMANTAU DAN PERINGATAN POLUSI UDARA PADA RUANG PARKIR TERTUTUP BERBASIS FPGA

Dalam membuat sistem pendeteksi pencemaran udara yang menggunakan gas CO sebagai parameter digunakan FPGA produksi XILINK yaitu Spartan 3E, yang dihubungkan dengan sensor gas CO (MQ-7) sebagai *input* dan lampu, *buzzer*, layar monitor, kipas sebagai Output.

Sistem yang akan dibuat diasumsikan menggunakan 12 buah sensor di tempatkan di 12 titik berbeda di area parkir tertutup seperti Gambar 3.1. Pengguna kendaraan bermotor atau pengunjung pada area parkir diharapkan mengetahui kondisi udara pada ruang parkir tersebut terutama kadar gas CO. Pengunjung akan mengetahui kadar gas CO di area parkir melalui tampilan monitor. Saat kadar gas CO dalam area parkir diatas normal, pengunjung akan diminta untuk waspada dengan memberikan peringatan melalui layar monitor serta lampu dan untuk mengembalikan kadar gas CO pada keadaan normal akan dinyalakan kipas. Saat kadar gas CO berbahaya maka sistem akan memberikan tanda bahaya dan petugas diharapkan membantu evakuasi pengunjung.

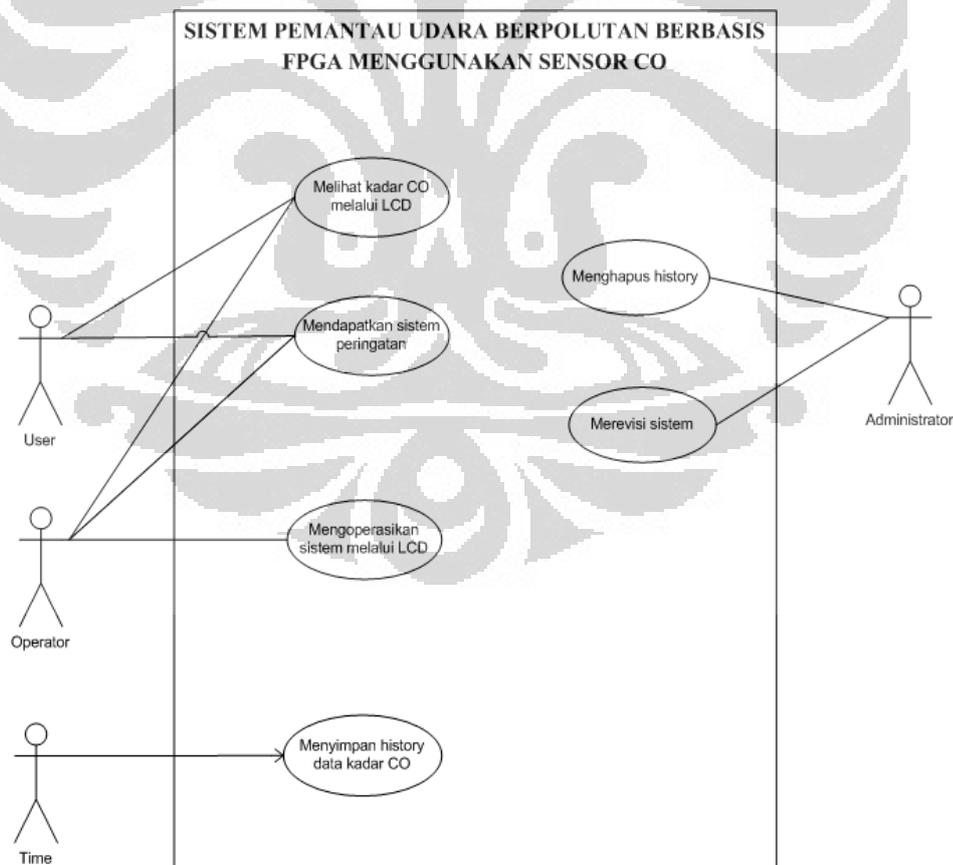


Gambar 3.1 Layout area parkir tertutup

3.1 System Requirement

System requirement adalah informasi mengenai kebutuhan pengguna terhadap sistem yang akan dibuat. Pada tahap ini, pembuat sistem (*developer*) mengumpulkan semua informasi mengenai apa yang dibutuhkan oleh pengguna sehingga menghasilkan *functional requirement*. Dalam sistem pemantau polusi udara yang akan dibuat, *functional requirement* yang didapat melalui proses tukar wawasan dan mengumpulkan informasi adalah

1. Operator dapat mengoperasikan sistem dengan mudah tanpa harus mengerti *programming*.
2. Pengguna (pengunjung) dapat mengetahui kadar gas CO secara *real time* melalui tampilan di layar monitor.
3. Adanya sistem peringatan terhadap *user* apabila kadar gas CO berbahaya dan sistem dapat melakukan tindakan mengurangi polusi.
4. Terdapat database tempat penyimpanan data pembacaan sensor.



Gambar 3.2 Use Case Diagram

Pada Gambar 3.2 dapat dilihat fungsi-fungsi yang dapat di akses oleh pengguna pada sistem yang dibuat. *Use case* diagram digunakan untuk menggambarkan *functional requirement*.

3.2 Desain

Setelah semua kebutuhan user didapat maka kebutuhan-kebutuhan tersebut diterjemahkan kedalam fitur dari sistem yang akan dibuat. Diasumsikan ruang parkir yang akan menggunakan sistem pemantau polusi udara berukuran 39x15 meter. Jangkauan sensor MQ7 yang digunakan adalah 2 m².

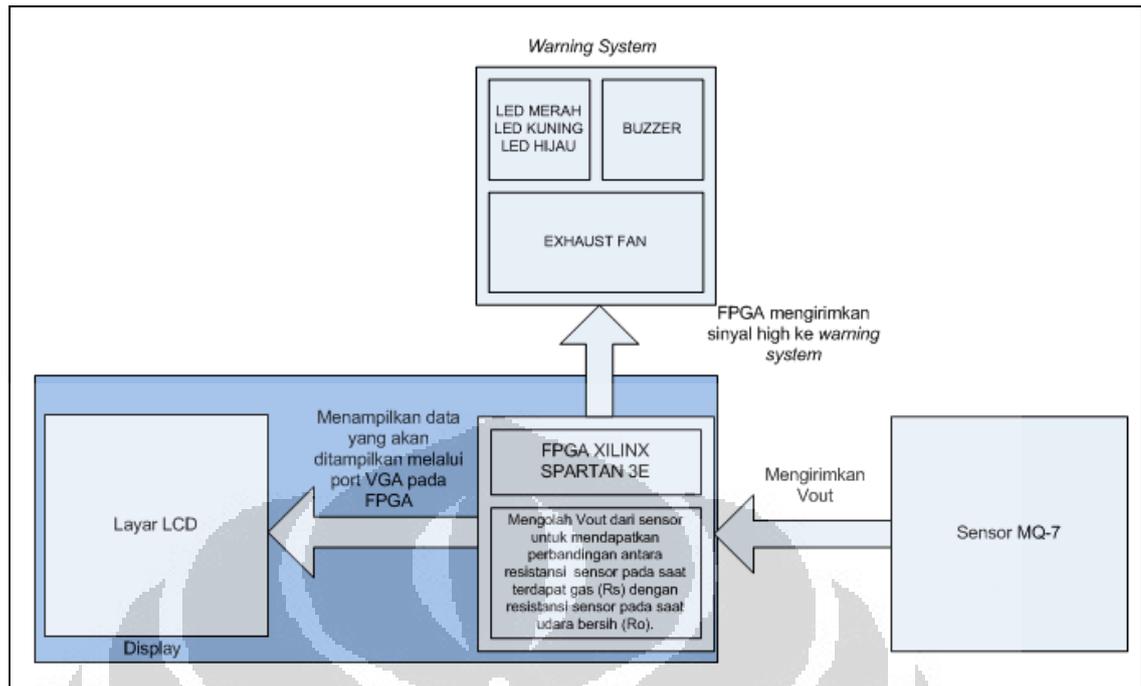
3.2.1 Desain Hardware

Perancangan perangkat keras dibagi menjadi tiga yaitu pembacaan dan pemrosesan data dari sensor, output data ke sistem peringatan dan menampilkan output data ke layar monitor melalui port VGA. Sensor yang digunakan adalah sensor CO MQ-7. Sistem peringatan akan menggunakan lampu LED dan layar monitor sebagai pemberi peringatan secara visual dan *buzzer* sebagai pemberi peringatan secara audio.

Penelitian ini dititikberatkan pada penampilan output data ke monitor melalui port VGA. Layar monitor yang digunakan adalah layar monitor yang menggunakan VGA port sebagai konektornya. Pada FPGA Spartan 3E VGA port sudah terintegrasi. Layar monitor yang digunakan adalah layar monitor yang biasa digunakan pada komputer biasa yang memiliki dua konektor yaitu konektor dengan VGA port dan konektor dengan Sumber daya (PLN).

3.2.2 Skematik Diagram Hardware

Sistem akan mendeteksi kadar gas CO menggunakan sensor MQ-7, kemudian hasil pengukuran akan dikirimkan ke *board* FPGA. Kemudian FPGA akan mengolah data tersebut dan memberikan hasilnya kepada *warning system* dan display sebagai respon *output*-nya. Blok diagram dari perangkat keras yang digunakan pada sistem ditunjukkan oleh Gambar 3.3

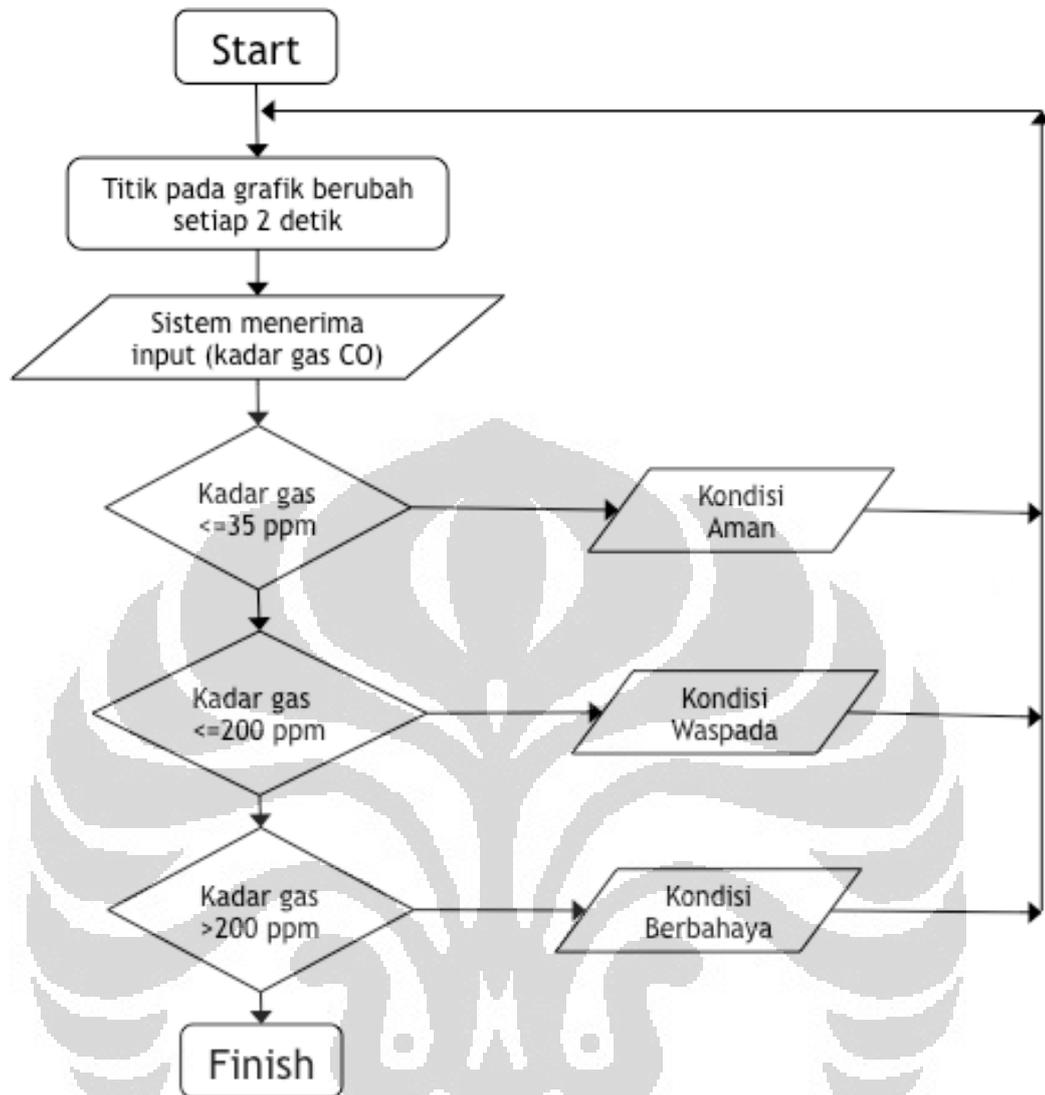


Gambar 3.3 Skematik Diagram

Pada Gambar 3.3 ditunjukkan bahwa sistem yang dibuat dibagi menjadi tiga fungsi utama dan pada penelitian ini dititikberatkan pada bagaimana FPGA menampilkan hasil pengukuran pada layar monitor.

3.2.3 Algoritma

Pemrograman untuk menampilkan data dari FPGA ke monitor akan dibuat dengan menggunakan bahasa VHDL dan bantuan software ISE 13 yang mendukung programing untuk FPGA Spartan 3E. Algoritma program dari sistem akan digambarkan melalui *flowchart* atau diagram alir pada Gambar 3.4 dan *pseudocode* pada Gambar 3.5.



Gambar 3.4 Flowchart cara kerja sistem

Pada Gambar 3.4 dapat dilihat saat sistem mulai dijalankan maka sensor udara yang dihubungkan dengan FPGA akan aktif untuk memeriksa kandungan udara, data dari sensor akan diterima FPGA, setelah diproses maka kandungan udara yang diukur akan ditampilkan di monitor. Ada tiga kondisi yang mempengaruhi output sistem.

1. Kandungan udara aman
Saat kondisi ini, sistem akan menyalakan lampu berwarna hijau.
2. Kandungan udara tidak aman

Saat kondisi ini, sistem akan menyalakan lampu berwarna kuning sebagai tanda untuk berhati-hati dan kipas pengatur sirkulasi udara akan dinyalakan dengan kecepatan sedang.

3. Kandungan udara berbahaya/beracun

Saat kondisi ini, sistem akan menyalakan lampur berwarna merah dan buzzer akan berbunyi sebagai tanda untuk melakukan evakuasi dari ruangan parkir. Kipas pengatur sirkulasi akan dinyalakan dengan kecepatan tinggi.

```

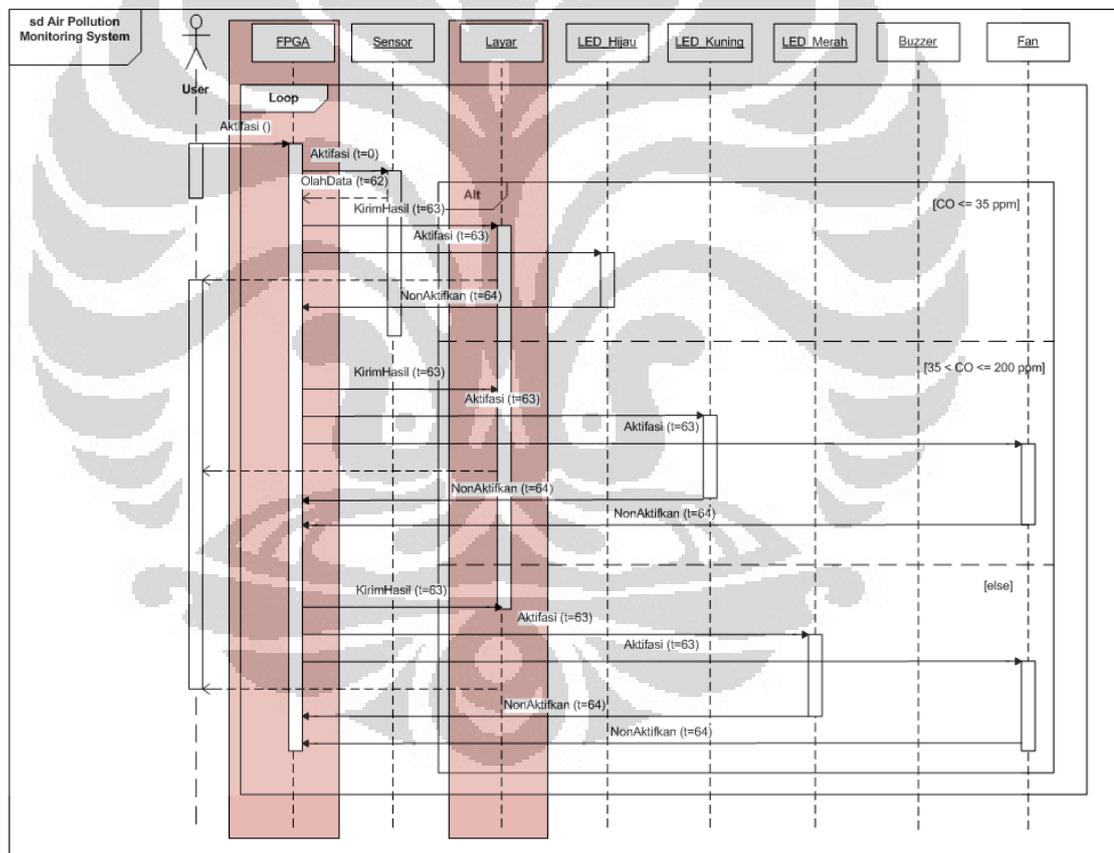
Start
Use variable ppm sensor, LED hijau, LED kuning,
  LED merah, buzzer, fan
Set constrant limit bawah, limit atas
Sistem = 1 ; 1 adalah aktif
A : Sensor = 1 (aktif)
monitor = 1
Read ppm sensor
Print ppm sensor
Jika ppm sensor <= limit bawah {
Lampu hijau = 1
Buzzer =0
Kipas =0
Loop A}
Jika ppm sensor >=limit bawah dan ppm sensor
  <=limit atas{
Lampu kuning = 1
Buzzer=0
Kipas=1
Loop A }
Jika ppm sensor > limit atas{
Lampu merah = 1
Buzzer=1
Kipas=1
Loop A
}

```

Gambar 3.5 Pseudocode cara kerja sistem

Tampilan yang akan dimunculkan pada layar berupa grafik, kadar CO hasil pengukuran dan pemberitahuan apakah kondisi aman, waspada, atau berbahaya. *Background* pada layar adalah warna hitam agar tulisan pada layar terlihat lebih jelas karena tulisan akan terlihat kontras.

Time constraint pada sistem dijelaskan dengan *sequence diagram* seperti pada Gambar 3.6. Sequence diagram menunjukkan bagaimana alat bekerja sesuai dengan aturan waktu. Sistem dimulai saat *user* mengaktifkan sistem dengan bantuan *main screen* sebagai monitor. Saat FPGA aktif, FPGA akan mengaktifkan sensor sehingga sensor mulai mengambil data dari udara. Pada Gambar 3.6 dapat dilihat saat sistem diaktifkan maka sistem akan melakukan pengulangan prosedur kerja (LOOP) sehingga sistem akan bekerja otomatis. Operator hanya dibutuhkan untuk mengaktifkan sistem dan memantau kerja sistem jika terjadi kerusakan pada sistem. User disini adalah pengunjung area parkir, dan interaksi yang dilakukan dengan sistem adalah menerima output sistem baik melalui layar monitor dan lampu peringatan.



*t dalam satuan detik

Gambar 3.6 Sequence Diagram

Ketika kadar CO hasil pengukuran sensor normal ($CO \leq 35$ ppm), FPGA akan memberikan sinyal *high* ke LED hijau untuk menyalakan lampu dan menampilkan output ke layar monitor. Saat kadar CO berada dalam tahap waspada ($35 \text{ ppm} < CO \leq 200 \text{ ppm}$) FPGA akan menyalakan LED kuning dan

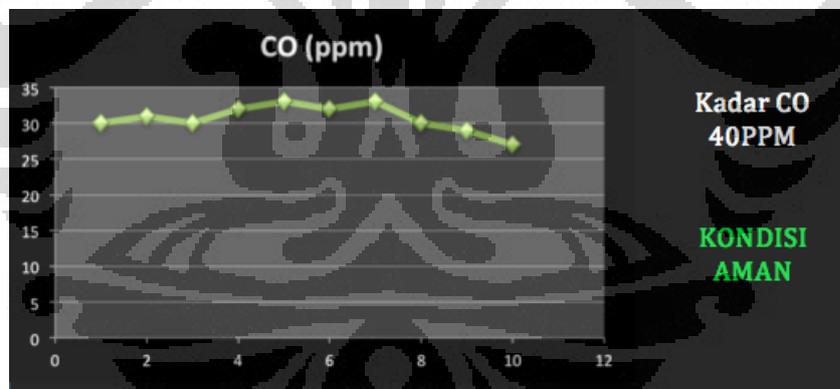
menyalakan kipas untuk membuat kadar udara dalam ruang parkir menjadi normal kembali. Saat kadar CO berbahaya ($CO > 200\text{ppm}$) maka FPGA akan menyalakan LED merah, kipas dan buzzer sebagai peringatan kepada *user* untuk segera melakukan evakuasi.

3.2.4 Perancangan Tampilan pada Layar monitor

Tampilan pada layar monitor yang dirancang memiliki *background* hitam agar tulisan pada layar terlihat lebih kontras sehingga membantu pengguna untuk melihat informasi yang ditampilkan pada layar. Layar akan menampilkan grafik kadar CO selama 30 menit terakhir. Disamping grafik kadar CO akan ditampilkan tulisan kadar CO dalam ppm dalam rentang 1 menit terakhir (*update* terakhir), dan dibawah tulisan kadar CO akan diberikan informasi apakah kondisi udara di ruang parkir aman, waspada, atau dalam keadaan berbahaya.

1. Saat kadar CO ≤ 35 ppm

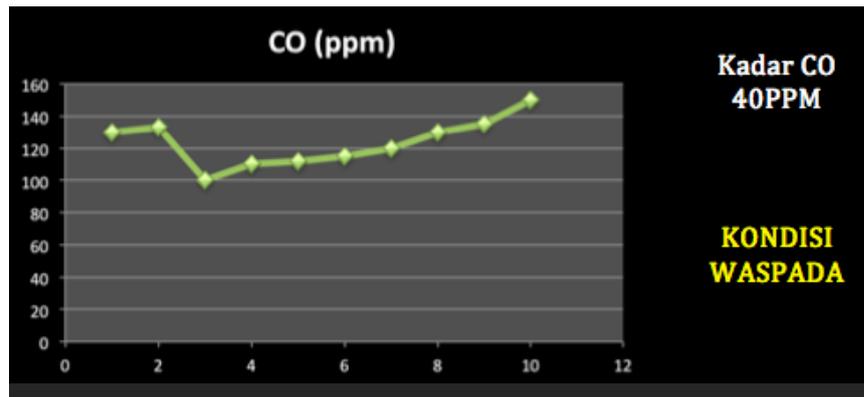
Tulisan kondisi aman ditampilkan dengan warna hijau karena warna hijau memberikan perasaan nyaman kepada yang melihat (Gambar 3.7).



Gambar 3.7 Tampilan saat kondisi aman

2. Saat kadar CO $\leq 200\text{ppm}$

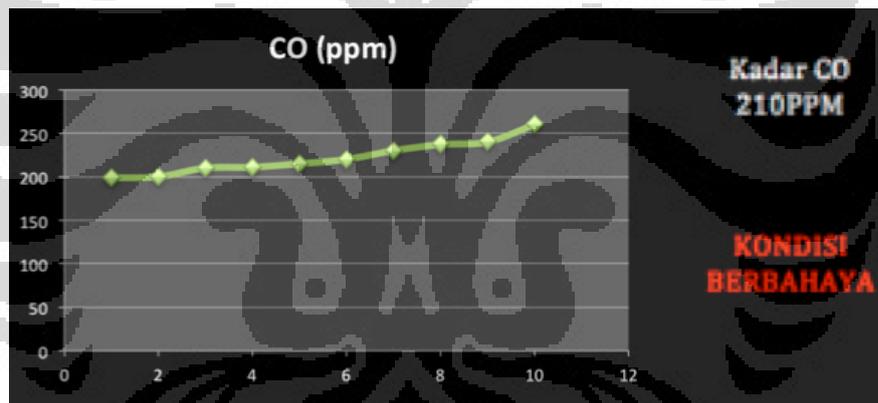
Tulisan kondisi waspada ditampilkan dengan warna kuning karena warna kuning memiliki tingkat kecerahan yang lumayan kuat dan sering diartikan untuk keadaan berhati-hati (Gambar 3.8).



Gambar 3.8 Tampilan saat kondisi waspada

3. Saat kadar CO >200ppm

Tulisan kondisi berbahaya ditampilkan dengan warna merah yang menandakan bahwa hal ini harus diperhatikan. Warna merah biasanya dikaitkan dengan keadaan yang penting atau berbahaya (Gambar 3.9).



Gambar 3.9 Tampilan saat kondisi berbahaya

Grafik yang dimunculkan adalah grafik WAKTU vs PPM, dengan waktu sebagai sumbu X yang menunjukkan waktu mulai dari waktu sekarang sampai waktu –30 menit. PPM sebagai sumbu Y akan menampilkan range mulai dari 0-240. Dengan pertimbangan saat ppm = 200 keadaan sudah sangat berbahaya dan pengunjung area parkir diharapkan tidak berada lagi di lokasi, hal ini juga untuk mencegah tampilan yang timpang atau titik pada grafik yang tidak jelas karena diharapkan grafik akan dominan menampilkan ppm pada range dibawah 35 ppm.

BAB IV

IMPLEMENTASI, PENGUJIAN DAN ANALISIS ANTARMUKA SISTEM PEMANTAU DAN PERINGATAN POLUSI UDARA

Tampilan antarmuka pada monitor terdiri dari dua jenis grafis yaitu grafis yang bersifat statis dan grafis yang bersifat dinamis. Grafis yang bersifat statis adalah grafis yang tidak akan berubah atau tetap selama ada tampilan pada layar. Grafis yang bersifat dinamis adalah grafis yang berubah-ubah sesuai dengan *input* yang didapat dari sensor yaitu nilai konsentrasi gas CO (PPM).

4.1 Mengatur Gambar

4.1.1 Mengatur Letak Gambar di Layar

Pada layar monitor, piksel ditembakkan mulai dari kiri atas sampai ke kanan bawah. Oleh karena itu, pada vhd1 dibuat konstanta yang menghitung ukuran piksel secara horizontal (*horizontal scan*) dan secara vertikal (*vertical scan*). Saat *horizontal scan* mencapai batas yang ditentukan, piksel akan ditembakkan mulai dari awal lagi namun pada baris yang berbeda sehingga *vertical scan* akan bertambah. Begitulah seterusnya sampai gambar pada layar penuh sesuai dengan ukuran yang ditentukan.

Setiap layar monitor memiliki ukuran yang berbeda-beda. Dengan FPGA, ukuran tampilan pada layar dapat dibuat lebih kecil namun tidak dapat melebihi kemampuan layar. Misalnya, layar monitor yang dimiliki berukuran 800x600 maka jika menggunakan tampilan 320x240, piksel yang tampil akan tampak lebih besar dari ukuran 800x600 sesuai perbandingan ukuran layar. Jika desain tampilan berukuran 1024x720, layar monitor tidak akan dapat menampilkannya karena ukuran piksel lebih kecil, melebihi kemampuan layar yang sebenarnya.

Mengatur letak gambar di monitor dilakukan dengan cara menentukan *horizontal scan* dan *vertical scan*. Pada monitor, sisi paling kiri tidak selalu berada pada *horizontal scan* = 0 tergantung pada monitor yang digunakan. Posisi

gambar yang ingin ditempatkan di layar perlu menentukan titik koordinatnya terlebih dahulu. *Horizontal scan* adalah koordinat X dan *vertical scan* adalah koordinat Y. Misalkan huruf “FPGA” dengan ukuran 40x20 akan ditampilkan di layar yang memiliki ukuran 320x240 seperti pada Gambar 4.1.



Gambar 4.1 Posisi gambar di layar

Tulisan “FPGA” pada gambar 4.1 akan ditampilkan di tengah layar, maka perlu ditentukan koordinat tulisan secara horizontal dan vertikal. Titik tengah tulisan harus berada pada titik tengah layar. Titik tengah dapat dihitung seperti berikut ini.

Layar:

- Sumbu X : $\frac{320}{2} = 160$
- Sumbu Y : $\frac{240}{2} = 120$

Tulisan:

- Sumbu X : $\frac{40}{2} = 20$
- Sumbu Y : $\frac{20}{2} = 10$

Setelah titik tengah ditemukan, selanjutnya yang perlu dilakukan adalah menentukan koordinat sudut-sudut tulisan. Misalkan tulisan yang ditampilkan adalah gambar persegi panjang maka keempat titik sudutnya ditentukan sebagai berikut.

titik pertama sumbu x adalah $160 - 20 = 140$,

titik terakhir sumbu X adalah $160 + 20 = 180$,

titik pertama sumbu Y adalah $120-10 = 110$,

titik terakhir sumbu Y adalah $120+10 = 130$.

Dengan demikian, koordinat batas atas tulisan adalah (140, 110) sampai (140, 110) dan koordinat batas bawah tulisan adalah (180, 110) sampai (180, 130).

Setelah koordinat titik diketahui, tulisan dapat ditampilkan di layar dengan menggunakan pseudocode pada Gambar 4.2.

```

Start
Use variable output, horz_scan, vert_scan, tulisan
Proses (output)
if (horz_scan >= 140 )
and (horz_scan <= 180)
and (vert_scan >= 110)
and (vert_scan <= 130) then
output = tulisan
end if;
End;

```

Gambar 4.2 Pseudocode mengatur posisi tulisan

4.1.2 Mengatur Gambar Statis pada Monitor

Gambar statis dalam tampilan di terjemahkan ke dalam bentuk *constant* yang diterjemahkan oleh aplikasi Matlab dengan fungsi *imread*. Berikut contoh penerjemahan gambar pada *implemented code*(Gambar 4.3).

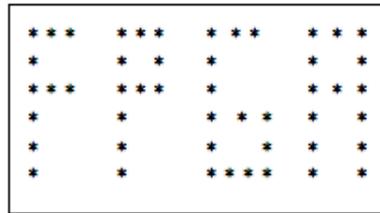
```

Screen_tulisan is ARRAY(0 to 5, 0 to 15);
CONSTANT char_tulisan : Screen_tulisan := (('
'1','1','1','0','1','1','1','0','1','1','1','0','0','1','1','1',
'1','0','0','0','1','0','1','0','1','0','0','0','0','1','0','1',
'1','1','1','0','1','1','1','0','1','0','0','0','0','1','1','1',
'1','0','0','0','1','0','0','0','1','0','1','0','0','1','0','1',
'1','0','0','0','1','0','0','0','1','0','0','0','0','1','0','1',
'1','0','0','0','1','0','0','0','1','1','1','0','0','1','0','1'));

```

Gambar 4.3 Array untuk gambar FPGA

Gambar 4.3 adalah bentuk array, dan apabila diterjemahkan dalam piksel, angka '1' adalah hitam dan angka '0' adalah putih hasilnya dapat dilihat pada Gambar 4.4.



Gambar 4.4 Tampilan *constant* char_tulisan

Seandainya tulisan pada Gambar 4.1 sudah diterjemahkan ke dalam bentuk *constant* dan memiliki ukuran *array* (0 to 20, 0 to 40) seperti yang didefinisikan pada gambar 4.3 maka *code line* untuk menampilkannya adalah sebagai berikut:

```

Start
Use variable horz_scan, vert_scan, tulisan,
Red, green, blue

Proses (output)
if (horz_scan >= 140) and (horz_scan <= 180)
  and (vert_scan >= 110) and (vert_scan <= 130) th
    if (pixel <= 39) then
      red = char_tulisan (pixel, line);
      green = char_tulisan (pixel, line);
      blue = char_tulisan (pixel, line);
    elseif (pixel >= 40) then
      red = 0;
      green = 0;
      blue = 0;
    end if;
  end if;

  if (horz_scan =320) then
    vert_scan = vert_scan +1;
    if (vert_scan >= 110) and (vert_scan <= 130)
      if (line <= 20) then
        line = line +1;
      elseif (line>=21) then
        line = 0;
      end if;
    end if;
  end if;
end if;
End;

```

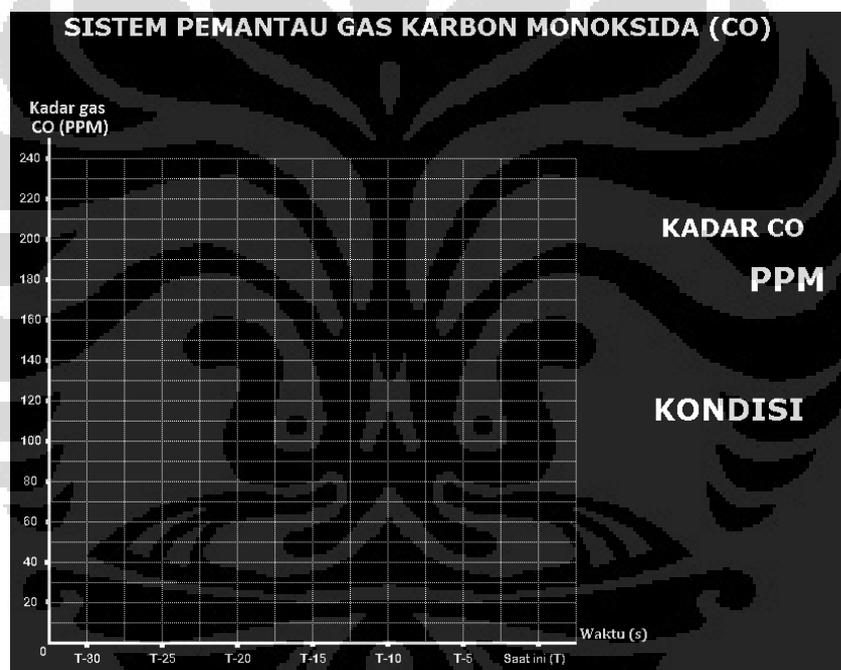
Gambar 4.5 *line code* untuk menampilkan tulisan

Mengatur warna tulisan pada Gambar 4.5 terdapat pada output red, green, dan blue. Apabila semuanya diberi nilai maka warna pada layar akan bernilai “111” yaitu warna putih. Untuk memberikan warna yang merah dapat

memberikan nilai '0' untuk green dan blue. Nilai output untuk warna dapat dilihat pada Tabel 2.3.

Gambar-gambar yang sudah diterjemahkan ke dalam konstanta dibuat ke dalam *package* agar tidak membuat *code line* pada file *code* utama menjadi banyak. *Constanta* dalam *package* dapat dipanggil dengan menggunakan fungsi “use work.nama_package.all;” dengan fungsi ini, semua *constant* atau fungsi yang sudah didefinisikan dapat digunakan langsung pada *main code*.

Pada penelitian ini, gambar grafik dan tulisan statis merupakan satu kesatuan sehingga tidak membutuhkan pengaturan letak tulisan pada *line code*. Gambar yang digunakan memiliki ukuran (830x660) piksel seperti yang ditunjukkan pada Gambar 4.6.



Gambar 4.6 Tampilan gambar statis

4.1.3 Mengatur Gambar Dinamis pada Monitor

Pada penelitian ini, ada tiga gambar dinamis yang ditampilkan. Tetapi gambar dinamis yang digunakan terdiri dari dua jenis, yaitu:

- Gambar titik-titik pada grafik
- Angka yang menunjukkan kadar gas CO dalam ppm. Dan Kondisi Aman, Waspada dan Berbahaya

Kedua gambar ini memiliki pengaturan yang berbeda. Gambar titik dikonfigurasi hanya dengan menentukan letak titik koordinatnya dan ukuran titiknya dalam piksel. Konfigurasi salah satu titik ditunjukkan pada Gambar 4.7.

```

Start
Use variable output, warna, v, h
Proses (output)
  Output = warna when v >= y1 and v <= y1+4
           and h >= x1 and h < x1 +4;
# dimana v= vertical scan pada layar
#        h= horizontal scan pada layar
#        x1= letak pixel horizontal titik satu
#        y1= letak pixel vertical titik satu
#        y1+4 = ukuran vertikal titik adalah 4 pixel
#        x1+4 = ukuran horizontal titik adalah 4 pixel
end

```

Gambar 4.7 Pseudocode menampilkan titik pada grafik

Gambar angka dan tulisan keadaan kondisi merupakan gambar statis yang berubah sesuai dengan kondisi yang diberikan. Untuk keadaan digunakan algoritma seperti pada Gambar 4.8.

```

Start
Use variable ppm sensor, kondisi
Proses (kondisi)
Jika ppm sensor <=35 {
  kondisi = Aman}
Jika ppm sensor <=200 {
  kondisi = Waspada}
Jika ppm sensor >200-{
  kondisi = Berbahaya}
end

```

Gambar 4.8 Pseudocode menampilkan kondisi

Tampilan angka menggunakan tiga digit yaitu digit ratusan, puluhan dan satuan. Untuk mendapatkan nilai tiap digit, digunakan algoritma pada Gambar 4.9, 4.10, 4.11.

```

Start
Use variable data, ratusan, puluhan, satuan,
datapuluhan, datasatuan
Proses (ratusan)
if data >=200 {
  ratusan = 2, datapuluhan = data - 200}
elseif data >=100 {
  ratusan = 1, datapuluhan = data - 100}
elseif data >=0 {
  ratusan = 0, datapuluhan = data}
else ratusan = 0; datapuluhan=0;
end proses

```

Gambar 4.9 Pseudocode menghitung digit ratusan

Pada digit ratusan, digit yang dimunculkan maksimal bernilai “2” karena saat kadar gas CO mencapai 200 harus dilakukan evakuasi sehingga tidak perlu menampilkan digit ppm di layar.

```

Start
Use variable puluhan, datapuluhan, datasatuan
Proses (puluhan)
If puluhan >= 90 {
  puluhan = 9, datasatuan = datapuluhan-90}
elseif puluhan >=80 {
  puluhan = 8, datasatuan = datapuluhan-80}
elseif puluhan >=70 {
  puluhan = 7, datasatuan = datapuluhan-70}
elseif puluhan >=60 {
  puluhan = 6, datasatuan = datapuluhan-60}
elseif puluhan >=50 {
  puluhan = 5, datasatuan = datapuluhan-50}
elseif puluhan >=40 {
  puluhan = 4, datasatuan = datapuluhan-40}
elseif puluhan >=30 {
  puluhan = 3, datasatuan = datapuluhan-30}
elseif puluhan >=20 {
  puluhan = 2, datasatuan = datapuluhan-20}
elseif puluhan >=10 {
  puluhan = 1, datasatuan = datapuluhan-10}
elseif puluhan >=0 {
  puluhan = 0, datasatuan = datapuluhan}
end proses

```

Gambar 4.10 Pseudocode menampilkan puluhan

```

Start
Use variable satuan,datasatuan
Proses (satuan)
Satuan = 1 when datasatuan=1 else
Satuan = 2 when datasatuan=2 else
Satuan = 3 when datasatuan=3 else
Satuan = 4 when datasatuan=4 else
Satuan = 5 when datasatuan=5 else
Satuan = 6 when datasatuan=6 else
Satuan = 7 when datasatuan=7 else
Satuan = 8 when datasatuan=8 else
Satuan = 9 when datasatuan=9 else
Satuan = 0 when datasatuan=0 else
end proses

```

Gambar 4.11 Pseudocode menampilkan satuan

Keluaran dari ketiga proses ini yaitu ratusan, puluhan dan satuan akan di gunakan pada file yang menampilkan gambar dinamis dengan nilai pada ketiga variabel ini sebagai penentu gambar keluarannya.

4.2 Skenario Pengujian

Sistem pengujian yang dilakukan terdiri dari dua, yaitu pengujian tampilan yang melibatkan responden untuk menilai apakah tampilan sudah *user friendly* dan pengujian *respons time* FPGA dari sensor mengambil data sampai memberikan output pada layar monitor.

4.2.1 Tampilan pada Layar Monitor

Pengujian tampilan dilakukan dengan cara melibatkan responden untuk menilai secara langsung tampilan pada layar monitor. Kondisi yang diberikan saat melakukan pengujian tampilan adalah sebagai berikut:

- Memberikan penjelasan mengenai sistem kepada responden.
- Layar monitor di tempatkan dalam ruangan dengan kondisi cahaya yang redup.
- Data *Input* yang digunakan bukan merupakan *input* dari sensor melainkan data yang dimasukkan secara manual dengan *switch* agar responden dapat melihat perubahan yang terjadi apabila rentang kadar CO yang ditampilkan lebar. Misalya untuk melihat bagaimana tampilan saat kondisi aman, waspada dan berbahaya.

Pengujian tampilan pada layar melibatkan tiga pokok penilaian, yaitu:

- Sistem secara umum
- Grafik
- Tulisan

Rentang penilaian yang untuk pernyataan yang diberikan adalah:

1 = Sangat buruk

2 = Buruk

3 = Lumayan

4 = Baik

5 = Sangat baik

Pernyataan yang dinilai oleh responden adalah sebagai berikut:

Sistem Secara umum

1. Sistem pemantau udara seperti ini penting
2. Interface pada layar membantu pengunjung memantau kondisi udara
3. Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas
4. Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya

Grafik

5. Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO
6. Titik penanda kadar CO di grafik terlihat jelas
7. Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik

Tulisan

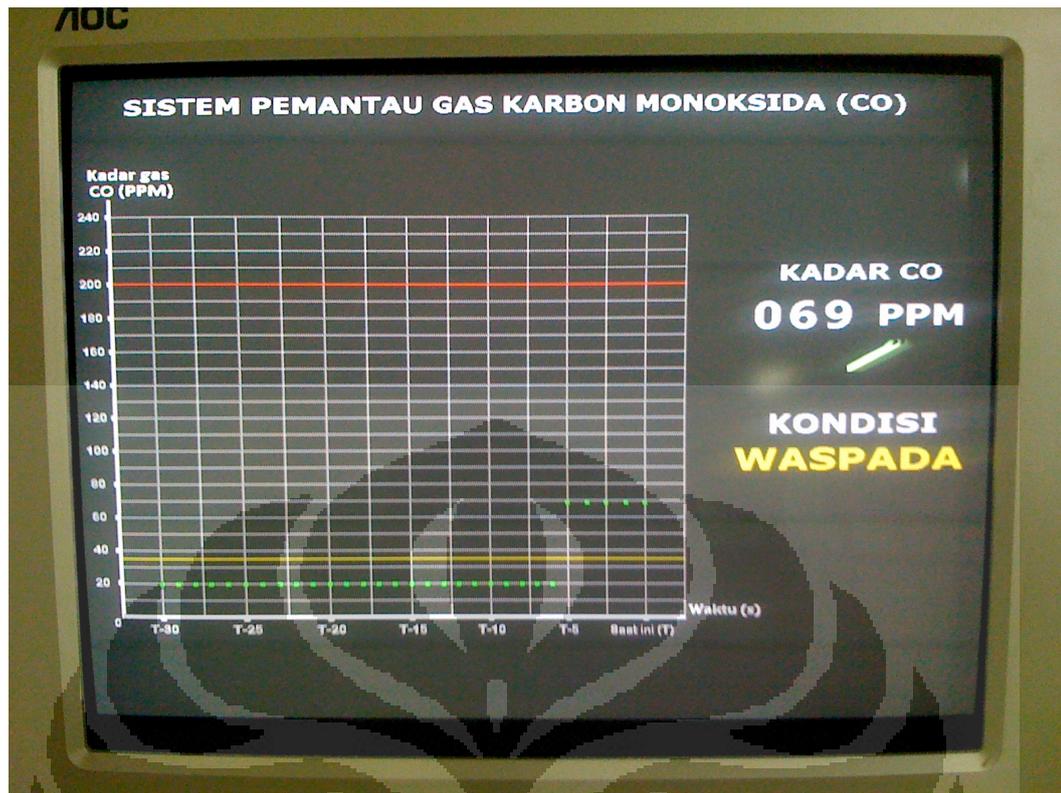
8. Ukuran tulisan dapat dibaca
9. Tulisan jelas (pada kondisi cahaya redup)
10. Background hitam membuat tulisan terlihat jelas

Tampilan pada layar monitor yang diuji adalah tampilan saat kondisi gas CO berada pada kondisi aman, waspada, berbahaya. Pada Gambar 4.12

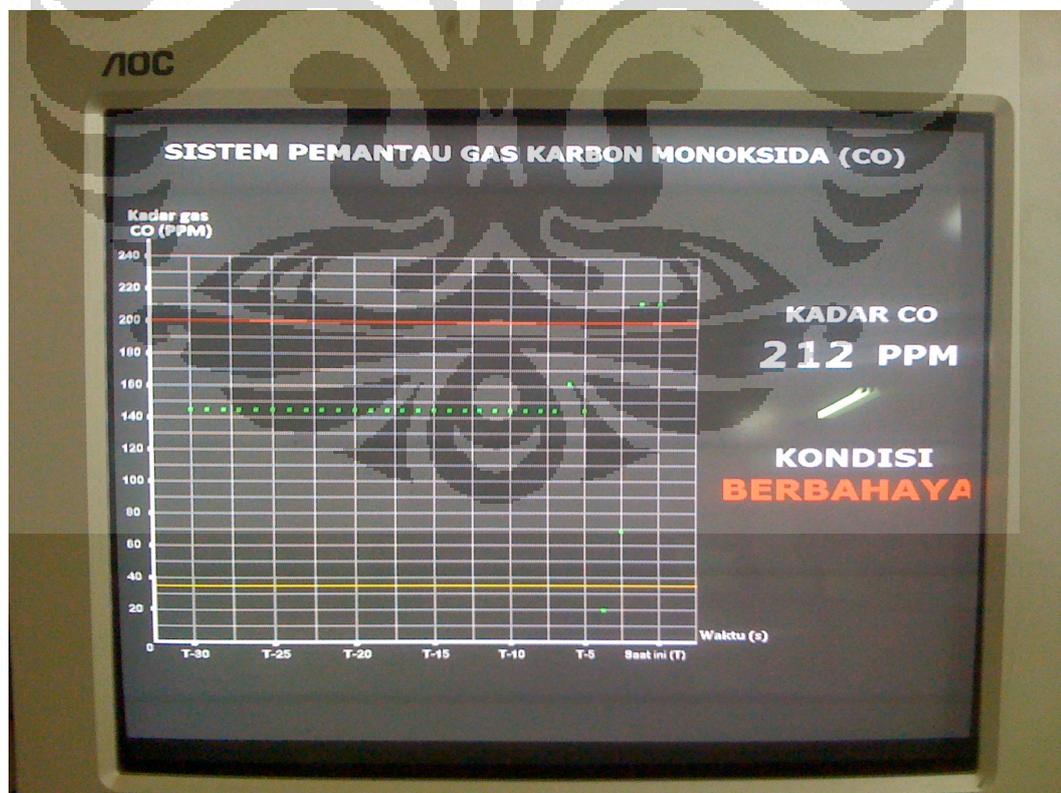
ditunjukkan tampilan pada layar monitor CRT saat kondisi aman, Gambar 4.13 saat kondisi aman dan Gambar 4.14 saat kondisi berbahaya. Gambar 4.15 merupakan tampilan pada layar LCD dengan menggunakan *source code* yang sama dengan tampilan pada layar CRT. Namun, yang digunakan untuk pengujian ini adalah tampilan pada layar CRT karena posisi gambar yang ditampilkan diatur sesuai dengan ukuran pada layar CRT. Data yang didapat dari responden akan menjadi tolak ukur apakah tampilan pada layar sudah *user friendly* atau belum.



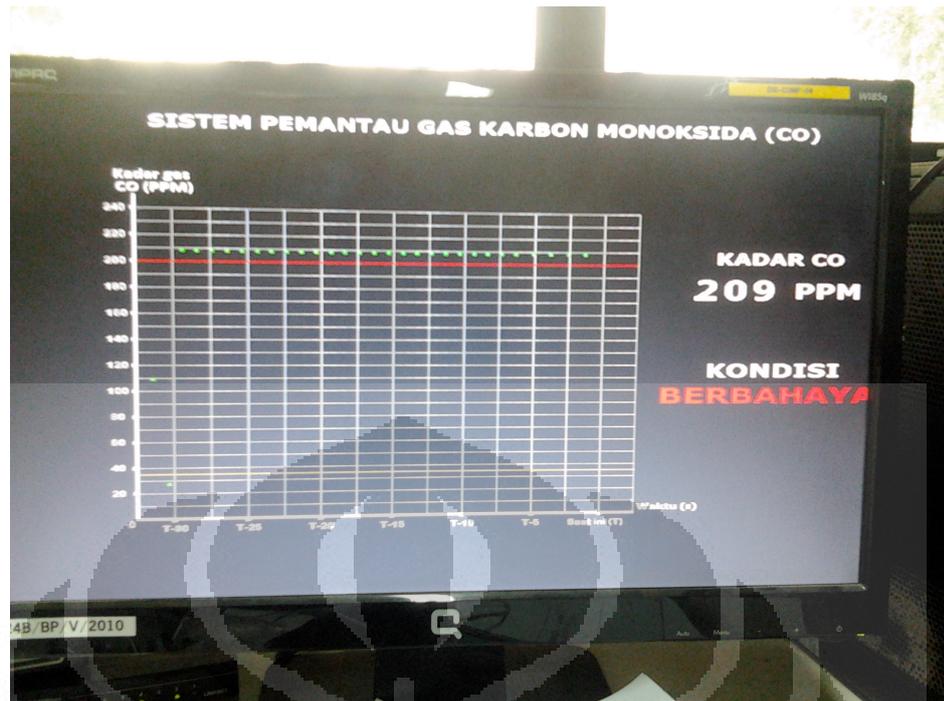
Gambar 4.12 Tampilan saat kondisi aman



Gambar 4.13 Tampilan saat kondisi waspada



Gambar 4.14 Tampilan saat kondisi berbahaya



Gambar 4.15 Tampilan pada layar LCD

4.2.2 Sistem mulai dari FPGA sampai tampil di Monitor

1. Sistem dapat melakukan update informasi sesudah ada *input* masuk

Pengujian dilakukan dengan *input* yang dimasukkan secara manual. *Input* yang digunakan adalah tiga saklar yang ada pada board FPGA dengan nilai pada Tabel 4.1.

Tabel 4.1 *Input* sesuai posisi saklar

Posisi Saklar	Nilai Input (ppm)
000	20
001	35
010	69
011	97
100	145
101	162
110	200
111	212

2. Response time dari FPGA ke monitor

Pengujian dilakukan dengan cara menghitung response time yang dibutuhkan FPGA sampai dapat menampilkan gambar satu *frame*. Pengujian akan dilakukan dengan mengukur waktu yang dibutuhkan saat piksel pertama muncul

sampai piksel terakhir di layar. Pengukuran dilakukan dengan memberikan output 1 dan 0 pada pin FPGA dan dihitung lama waktunya menggunakan *oscilloscope*. Pengukuran dilakukan dengan mengambil 10 kali data yang dapat dilihat pada Tabel 4.3.

4.3 Data Pengujian

4.3.1 Tampilan pada layar Monitor

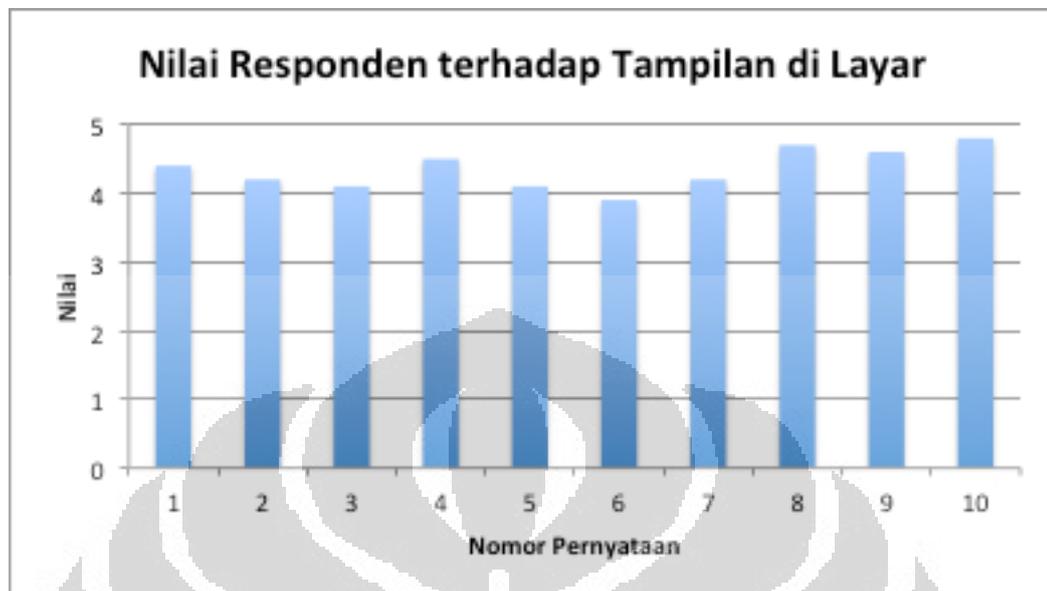
Pengujian dilakukan dengan mengisi kuesioner yang dibagikan kepada responden saat FPGA dihubungkan ke port VGA dan memberikan keluaran pada layar monitor. Pengaturan data kadar CO sebagai *input* dilakukan dengan menggunakan tiga saklar sehingga variasi data menjadi 2^3 yaitu 8 data. Nilai dari setiap variasi dilakukan manual dengan mengubahnya program. Kadar CO yang dimasukkan dengan saklar adalah nilai yang tetap sehingga tidak akan berubah kecuali terjadi perubahan pada saklar.

Hasil pengujian ini ditunjukkan pada Tabel 4.2. Tabel ini adalah penilaian responden terhadap pernyataan yang telah dibuat yang menilai Sistem secara umum, Grafik, dan tulisan.

Tabel 4.2 Penilaian Responden

NO	NAMA	NILAI PERNYATAAN									
		Sistem Secara Umum				Grafik			Tulisan		
		1	2	3	4	5	6	7	8	9	10
1	Responden 1	5	5	5	5	4	4	4	5	5	5
2	Responden 2	5	4	5	4	4	2	3	4	5	5
3	Responden 3	4	4	3	4	4	3	4	5	4	5
4	Responden 4	5	4	4	3	4	3	4	4	4	5
5	Responden 5	4	4	4	5	3	3	4	4	4	4
6	Responden 6	5	4	3	4	4	5	4	5	5	5
7	Responden 7	3	3	3	4	3	3	4	5	5	5
8	Responden 8	4	3	4	4	4	4	3	4	4	4
9	Responden 9	4	4	3	5	4	3	4	5	5	5
10	Responden 10	4	4	4	5	4	5	5	5	4	4
11	Responden 11	5	5	5	5	5	5	5	5	5	5
12	Responden 12	4	5	5	5	5	5	5	5	5	5
13	Responden 13	5	5	5	5	4	5	5	5	5	5
14	Responden 14	5	5	5	5	5	5	5	5	5	5
	Rata-Rata	4,4	4,2	4,1	4,5	4,1	3,9	4,2	4,7	4,6	4,8

Dari data pada Tabel 4.2 dapat dibuat grafik nilai rata-rata responden terhadap pernyataan yang telah diberikan seperti pada Gambar 4.16.



Gambar 4.16 Grafik rata-rata nilai responden terhadap pernyataan

4.3.2 Sistem mulai dari FPGA sampai tampil di Monitor

1. Sistem dapat melakukan update informasi sesudah ada *input* masuk

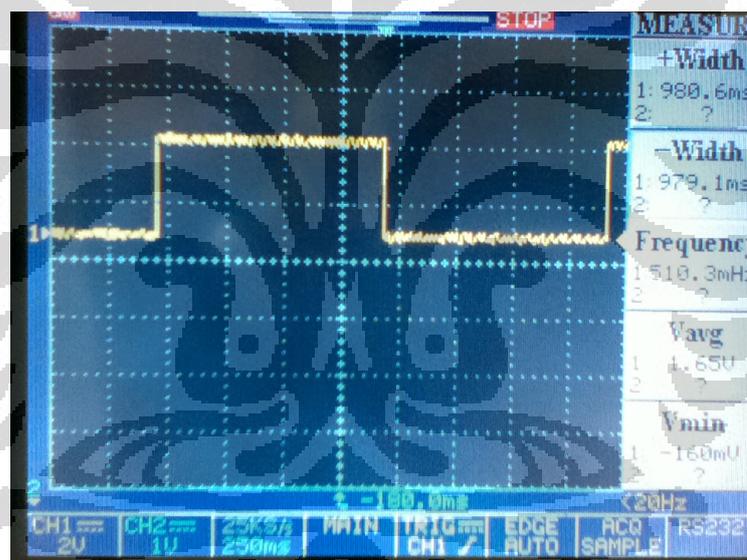
Sistem dapat melakukan update apabila ada data yang masuk dari saklar. Update yang dilakukan terjadi pada tampilan nilai ppm dan titik pada grafik. Keduanya akan berubah sesuai dengan nilai *input* yang dimasukkan seperti yang ditunjukkan pada Gambar 4.12, 4.13, 4.14.

2. Response time dari FPGA ke monitor

Program pengatur *response time* dimasukkan ke dalam *line code* yang menghitung *counter* tampilan sampai 30 kali untuk satu kali tampilan *frame*. Sehingga waktu yang didapat adalah 30 kali waktu yang dibutuhkan FPGA untuk menampilkan piksel dari yang pertama sampai yang terakhir.

Tabel 4.3 Waktu Memunculkan 30 *Frame*

NO	Waktu (ms)
1	979,2
2	979,7
3	980,5
4	980,5
5	979,7
6	980,3
7	980,5
8	980,4
9	980,4
10	980,0
Rata-rata	980,12

Gambar 4.17 Pengukuran *response time* dengan *oscilloscope*

Gambar 4.17 merupakan gambar pada *oscilloscope* saat pengukuran. Karena lebar data saat *high* (+Width) dan saat *low* (-Width) hampir sama, maka yang diambil adalah lebar data saat *high*.

Setelah nilai diatas dibagi 30 maka didapatkan waktu sebenarnya yang dibutuhkan FPGA untuk memunculkan piksel yang pertama sampai piksel yang terakhir. Pada Tabel 4.4 dapat dilihat waktu yang dibutuhkan FPGA untuk menampilkan satu *frame* di layar.

Tabel 4.4 Waktu memunculkan satu frame

NO	Waktu (ms)
1	32,62
2	32,66
3	32,68
4	32,68
5	32,66
6	32,68
7	32,68
8	32,68
9	32,68
10	32,67
Rata-rata	32,67

Nilai ini didapat dari rumus

$$T = \frac{H \times V}{Clock} \quad (4.1)$$

dimana

T = waktu menampilkan gambar

H = Banyak piksel Horizontal

V = Banyak baris Vertikal

Pada program ini, *Clock* yang digunakan adalah 100MHz dimana memiliki nilai '1' dan '0' dan selama *clock* bernilai '1' digunakan setengahnya untuk menampilkan piksel. Jadi dari *clock* yang diberikan sebesar 100Mhz, hanya seperempatnya yaitu 25Mhz yang digunakan untuk menampilkan gambar.

$$T = \frac{1030 \text{ piksel}/\text{baris} \times 788 \text{ baris}/\text{layar}}{25 \times 10^6 \text{ } 1/\text{detik}}$$

$$T = 0,0324656s$$

$$T = 32,47ms$$

$$\Delta T = T_{FPGA} - T_{perhitungan} \quad (4.2)$$

$$\Delta T = 32,67ms - 32,47ms$$

$$\Delta T = 0,2ms$$

$$\text{Persen Kesalahan} = \left| \frac{\text{nilai literatur} - \text{nilai alat}}{\text{nilai literatur}} \right| \times 100\% \quad (4.3)$$

$$\text{Persen Kesalahan} = \left| \frac{0,2}{32,67} \right| \times 100\%$$

$$\text{Persen Kesalahan} = 0,612\%$$

4.4 Analisis

4.4.1 Analisis Penilaian Responden

Sistem Secara umum

1. Sistem pemantau udara seperti ini penting = 4,4
Rata-rata nilai responden adalah 4,4 dari skala 5. Hal ini menunjukkan bahwa sistem pemantau udara gas karbon monoksida penting, karena gas CO secara alami tidak dapat di deteksi oleh indra manusia dan keracunan gas ini dapat menyebabkan kematian.
2. Interface pada layar membantu pengunjung memantau kondisi udara = 4,2
Interface yang ditampilkan pada layar sudah memberikan informasi yang jelas kepada pengunjung.
3. Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas = 4,1
Tampilan mengenai kadar gas CO yang ukurannya paling besar diantara tulisan yang lain sudah memberikan informasi yang jelas mengenai kadar CO secara real time.
4. Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya = 4,5
Penggunaan warna hijau untuk tulisan aman, kuning untuk waspada dan merah untuk berbahaya sudah memberikan tanda sesuai dengan kondisinya.

Grafik

5. Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO = 4,1
Grafik menampilkan kadar gas CO pada kondisi saat ini sampai dengan kondisi 30 menit sebelumnya. Dengan adanya bantuan grafik, pengunjung

dapat memprediksi kadar gas CO berikutnya dengan melihat apakah kadarnya cenderung naik atau cenderung turun.

6. Titik penanda kadar CO di grafik terlihat jelas = 3,9
Titik penanda CO pada grafik sudah cukup terlihat jelas, karena titik diberi warna hijau dengan latar hitam yang membuat warna titik semakin kontras.
7. Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik = 4,2
Garis pembantu membuat pengunjung dapat menentukan letak titik lebih akurat.

Tulisan

8. Ukuran tulisan dapat dibaca = 4,7
Ukuran tulisan dapat dibaca karena tulisan kondisi berukuran 243x39 piksel, angka 26x39 piksel pada layar dengan tampilan berukuran 800x650 piksel. Hal ini membuat ukuran tulisan relatif besar, dan jelas untuk dibaca, kecuali pada tulisan label di sumbu x dan sumbu y.
9. Tulisan jelas (pada kondisi cahaya redup) = 4,6
Tulisan yang berwarna putih terlihat lebih jelas pada latar hitam terutama pada kondisi cahaya yang redup.
10. Background hitam membuat tulisan terlihat jelas = 4,8
Latar yang berwarna hitam membuat tulisan lebih kontras. Hal ini disebabkan karena manusia memiliki tingkat ketertarikan lebih tinggi pada bagian terang di kondisi gelap daripada bagian gelap di kondisi terang. Titik hitam pada latar putih cocok untuk memberikan fokus pada titik tersebut, namun pada desain tampilan ini banyak tulisan yang menyebar di setiap bagian layar sehingga tidak fokus pada bagian tertentu saja. Selain itu, layar akan di tempatkan pada ruang parkir tertutup yang kondisi penerangannya cenderung redup.

4.4.2 Analisis Sistem mulai dari FPGA sampai tampil di Monitor

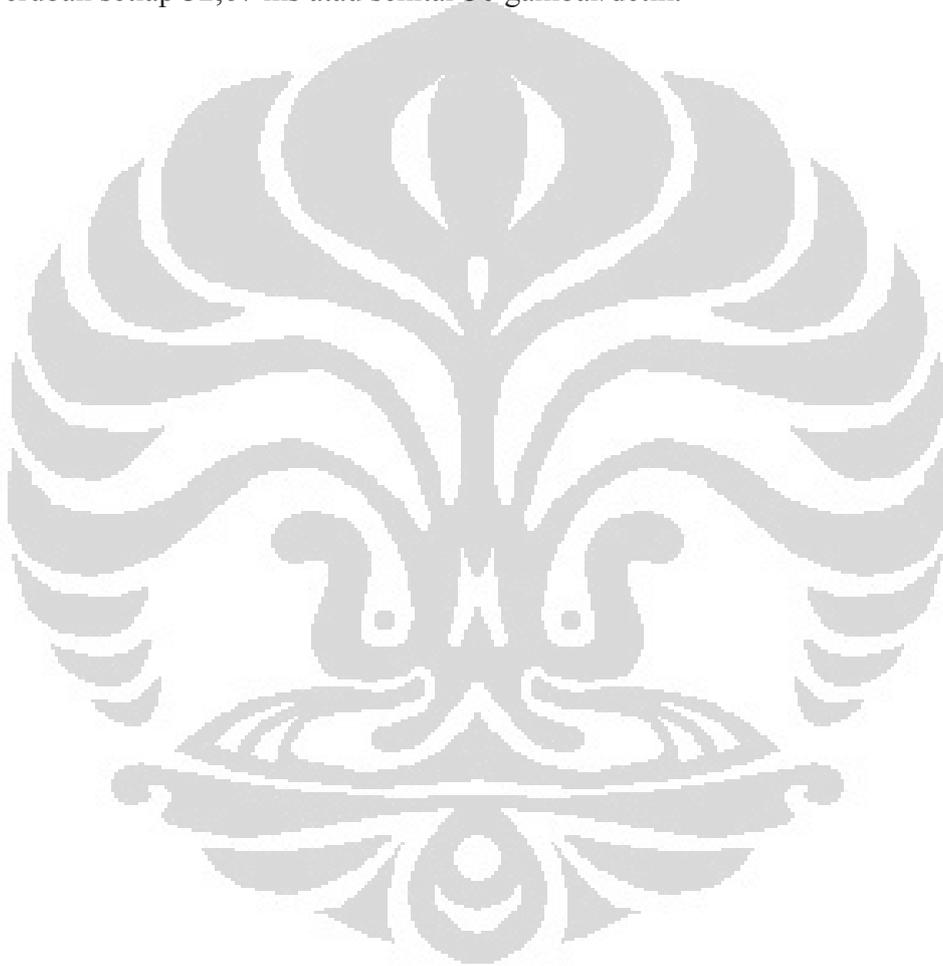
1. Sistem dapat melakukan *update* informasi sesudah ada *input* masuk

Kemampuan sistem melakukan *update* informasi tergantung pada program yang dibuat. Apabila sistem tidak dapat meng-*update* tampilan di layar maka perlu dilakukan analisis lagi pada program yang sudah dibuat.

2. Response time dari FPGA ke monitor

Waktu yang dibutuhkan FPGA menulis seluruh gambar berukuran 1030x788 adalah 32,67ms. Sementara berdasarkan perhitungan, waktu yang seharusnya dibutuhkan adalah 32,47ms. Hampir tidak ada perbedaan antara waktu yang diperoleh melalui perhitungan dengan waktu yang sebenarnya.

Waktu rata-rata yang dibutuhkan untuk menampilkan satu *frame* gambar adalah 32,67 ms. Maka gambar yang ditampilkan dapat di atur untuk berubah setiap 32,67 ms atau sekitar 30 gambar/detik.



BAB V

KESIMPULAN

1. Antarmuka sistem pemantau dan peringatan polusi udara berbasis FPGA Xilinx Spartan 3E berhasil dibuat dengan menggunakan bahasa VHDL dan dapat melakukan *update* bila ada perubahan *input*.
2. Tampilan angka terdiri dari tiga digit dan masing-masing digit ditampilkan secara terpisah.
3. *Clock* yang digunakan untuk menampilkan gambar di layar monitor dengan ukuran 1030x788 piksel adalah 100Mhz.
4. Rata-rata nilai kualitatif yang didapat dari responden adalah 4,35 yang berarti secara keseluruhan sistem ini sudah baik.
5. Rata-rata *response time* FPGA menampilkan satu *frame* di layar monitor dengan ukuran 1030x788 adalah 32,67 ms dengan memiliki persen kesalahan 0,612% dari hasil perhitungan.

DAFTAR ACUAN

- [1] Mambo. (2011). *Udara*. [Online]. Available: http://e-smartschool.co.id/index2.php?option=com_content&do_pdf=1&id=181 (diakses 11 Oktober 2011 pukul 11.23 AM).
- [2] Ocet. (2010). *Polusi Udara*. [Online]. Available: http://ocet.student.umm.ac.id/download-as-pdf/umm_blog_article_139.pdf (diakses 11 Oktober 2011 pukul 11.10AM).
- [3] Republik Indonesia. 1997. Keputusan Kepala Badan Pengendalian Dampak Lingkungan Nomor : KEP-107/KABAPEDAL/11/1997.
- [4] Hadiyani Murti. *Keracunan Karbon Monoksida*. [Online]. Available: <http://www.pom.go.id/public/siker/desc/produk/RacunKarMon.pdf>
- [5] Regina Baley. *Carbon Monoksida*. [Online]. Available : <http://id.shvoong.com/exact-sciences/biology/2195593-carbon-monoksida/> (diakses 21 November 2011 pkl. 10.00 WIB).
- [6] Hafsal. (2008). *FPGA*. [Online]. Available: <http://vetechlib.blogspot.com/2008/09/fpga.html> (diakses 8 November 2011).
- [7] Elean. (2012). *FPGA dan CPLD*. [Online]. Available : <http://abikindo.blogspot.com/2010/05/fpga-dan-cpld.html> (diakses 8 November 2011).
- [8] Annymous. (2009). *How FPGAs Work*. [Online]. Available: <http://www.fpga4fun.com/FPGAinfo2.html> (diakses 8 November 2011).
- [9] Spartan-3D Starter Kit Board User Guide, UG230 (v1.0) March 9, 2006.
- [10] Dzale. (2009). Mengenal VHDL. [Online]. Available: <http://ndoware.com/mengenal-vhdl.html> (diakses 8 November 2011).
- [11] Online shop (2010). *MQ-7 CO, H2 Gas Sensor*. [Online] Available : http://www.ec.in.th/Sensors/Gas_sensor?product_id=855 (diakses 10 Juni 2012)
- [12] A.J. Dix, J.E. Finlay, G.D. Abowd and R. Beale, "Human-Computer Interaction", Third Edition, USA : Prentice Hall, 2003
- [13] Iwan Sugihartono. (2008). *Mengenal Warna Dan Cahaya*. [Online]. Available: <http://excitonindo.wordpress.com/2008/02/19/mengenal-warna-dan-cahaya/> (diakses 26 Desember 2011).
- [14] Richard Fang. (2010). Psikologi Warna Pada Desain Grafis. [Online]. Available: <http://jurusgrafis.com/artikel/psikologi-warna-desain-grafis/> (diakses 26 Desember 2011).
- [15] Maryadi, Windaratie. (2009). Arti dan Efek Warna pada Otak. [Online]. Available: http://kosmo.vivanews.com/news/read/28105-arti_dan_efek_warna_pada_otak (diakses 26 Desember 2011).

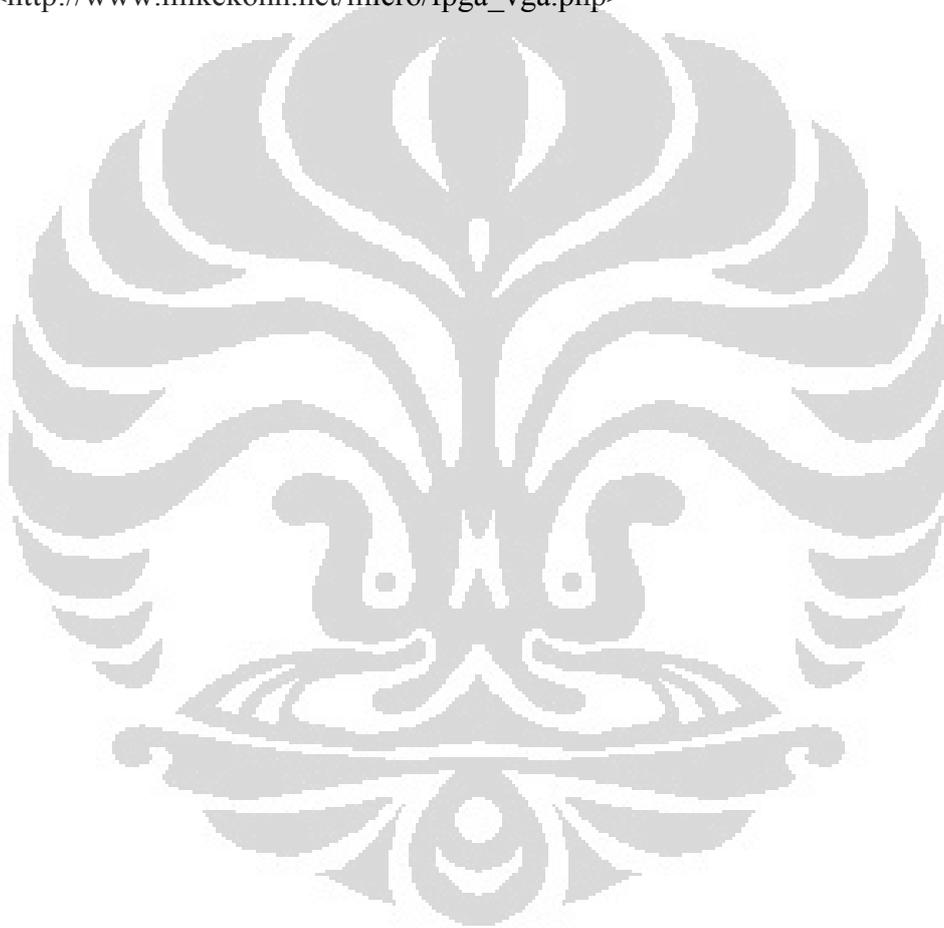
DAFTAR PUSTAKA

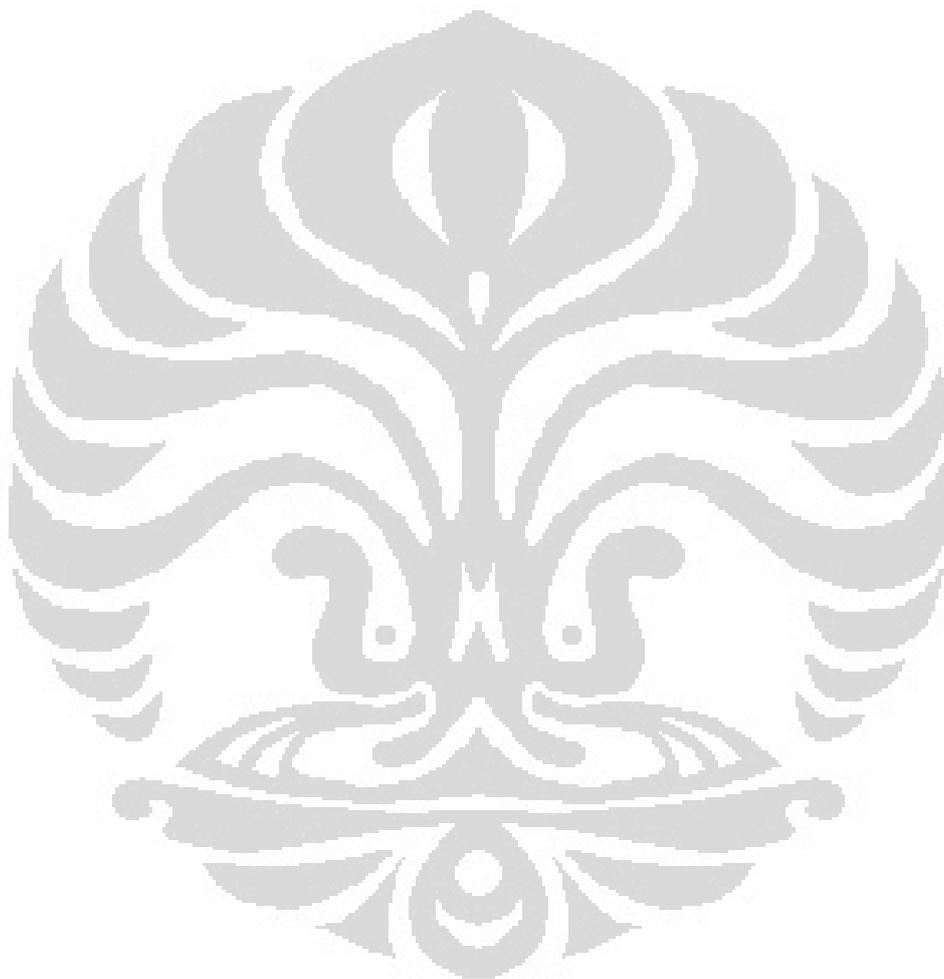
Chu Pong P. (2008). *FPGA Prototyping by VHDL Examples Xilinx Spartan™ -3 Version*. Canada : A John Wiley & Sons, inc.

Immelman, Quintin. *FPGA VGA Text.zip*. FPGA4fun.com. 18 April 2012.
< <http://www.fpga4fun.com/ExternalContributions/>>

Pedroni Volnei A. (2004). *Circuit Desing with VHDL*. London : MIT Press

“VGA Graphics on an FPGA.” Mike Kohn’s website. 20 April 2008.
<http://www.mikekohn.net/micro/fpga_vga.php>



LAMPIRAN

Nama : Henry

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
5	Sistem pemantau udara seperti ini penting	
5	Interface pada layar membantu pengunjung memantau kondisi udara	
5	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
5	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	SUDAH MONTAP!
5	Titik penanda kadar CO di grafik terlihat jelas	
5	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Rani

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
4	Sistem pemantau udara seperti ini penting	
5	Interface pada layar membantu pengunjung memantau kondisi udara	
5	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
5	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	Sangat
5	Titik penanda kadar CO di grafik terlihat jelas	
5	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Herda S Napihpu

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
4	Sistem pemantau udara seperti ini penting	
4	Interface pada layar membantu pengunjung memantau kondisi udara	
3	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
4	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
3	Titik penanda kadar CO di grafik terlihat jelas	
4	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
4	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

SUDAH BAGUS

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Eka Purwa Laksana

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
5	Sistem pemantau udara seperti ini penting	
4	Interface pada layar membantu pengunjung memantau kondisi udara	
4	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
3	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
3	Titik penanda kadar CO di grafik terlihat jelas	
4	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
4	Ukuran tulisan dapat dibaca	
4	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

BAGUS

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Arian C. Adi

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
5	Sistem pemantau udara seperti ini penting	
5	Interface pada layar membantu pengunjung memantau kondisi udara	
5	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
5	Titik penanda kadar CO di grafik terlihat jelas	
5	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : ELWIAN S.

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
5	Sistem pemantau udara seperti ini penting	✓
5	Interface pada layar membantu pengunjung memantau kondisi udara	
5	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
5	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
5	Titik penanda kadar CO di grafik terlihat jelas	
5	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Yolius Dimas H.T

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
4	Sistem pemantau udara seperti ini penting	
4	Interface pada layar membantu pengunjung memantau kondisi udara	
3	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
3	Titik penanda kadar CO di grafik terlihat jelas	
4	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Bagus

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Noni

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
4	Sistem pemantau udara seperti ini penting	
4	Interface pada layar membantu pengunjung memantau kondisi udara	
4	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
5	Titik penanda kadar CO di grafik terlihat jelas	
5	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
4	Tulisan jelas (pada kondisi cahaya redup)	
4	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Rhaka

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
3	Sistem pemantau udara seperti ini penting	
3	Interface pada layar membantu pengunjung memantau kondisi udara	
3	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
4	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
3	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
3	Titik penanda kadar CO di grafik terlihat jelas	
4	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
5	Ukuran tulisan dapat dibaca	
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Helmi

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
4	Sistem pemantau udara seperti ini penting	4
3	Interface pada layar membantu pengunjung memantau kondisi udara	
4	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
4	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
3	Titik penanda kadar CO di grafik terlihat jelas	
3	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
4	Ukuran tulisan dapat dibaca	
4	Tulisan jelas (pada kondisi cahaya redup)	
4	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : *Sulaga D.*

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
<i>4</i>	Sistem pemantau udara seperti ini penting	
<i>4</i>	Interface pada layar membantu pengunjung memantau kondisi udara	
<i>4</i>	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
<i>5</i>	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
<i>3</i>	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
<i>3</i>	Titik penanda kadar CO di grafik terlihat jelas	
<i>4</i>	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
<i>0</i>	Ukuran tulisan dapat dibaca	
<i>4</i>	Tulisan jelas (pada kondisi cahaya redup)	
<i>4</i>	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : *Darma*

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
<i>5</i>	Sistem pemantau udara seperti ini penting	
<i>4</i>	Interface pada layar membantu pengunjung memantau kondisi udara	
<i>3</i>	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
<i>4</i>	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
<i>4</i>	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
<i>5</i>	Titik penanda kadar CO di grafik terlihat jelas	
<i>4</i>	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
<i>5</i>	Ukuran tulisan dapat dibaca	
<i>5</i>	Tulisan jelas (pada kondisi cahaya redup)	
<i>5</i>	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Ahmad Shauqi

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
5	Sistem pemantau udara seperti ini penting	Ya
5	Interface pada layar membantu pengunjung memantau kondisi udara	Ya
5	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	Ya
5	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	Ya
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	Ya
4	Titik penanda kadar CO di grafik terlihat jelas	Ya cukup jelas
4	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	Ya
	Tulisan	
5	Ukuran tulisan dapat dibaca	Ya
5	Tulisan jelas (pada kondisi cahaya redup)	Ya
5	Background hitam membuat tulisan terlihat jelas	Ya

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali

Nama : Bama

Nilai	Pertanyaan	Komentar
	Sistem secara Umum	
5	Sistem pemantau udara seperti ini penting	
4	Interface pada layar membantu pengunjung memantau kondisi udara	
5	Tampilan di monitor sudah memberikan informasi kadar gas CO yang jelas	
4	Penggunaan warna sudah menandakan kondisi aman, waspada dan berbahaya	
	Grafik	
4	Tampilan pada grafik sudah memberikan informasi yang lengkap mengenai kadar CO	
2	Titik penanda kadar CO di grafik terlihat jelas	
3	Garis pembantu pada grafik sudah membantu melihat posisi titik pada grafik	
	Tulisan	
4	Ukuran tulisan dapat dibaca	yg di ganti kr ke ke cilin
5	Tulisan jelas (pada kondisi cahaya redup)	
5	Background hitam membuat tulisan terlihat jelas	

Keterangan nilai

- 1 = Buruk sekali
- 2 = Buruk
- 3 = Lumayan
- 4 = Baik
- 5 = Baik sekali