

**IMPLEMENTASI SISTEM UKUR SUHU DAN PENGENDALIAN  
KELEMBABAN INKUBATOR MISELUM J.TIRAM  
PADA XILINX SPARTAN-3**

**SKRIPSI**

**KONSENTRASI TEKNIK ELEKTRONIKA**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun Oleh:

**SILVESTER EKA JEMALI**  
**NIM. 0810632008-63**

**KEMENTERIAN PENDIDIKAN NASIONAL  
UNIVERSITAS BRAWIJAYA  
FAKULTAS TEKNIK  
MALANG  
2010**

**LEMBAR PERSETUJUAN**

**IMPLEMENTASI SISTEM UKUR SUHU DAN PENGENDALIAN  
KELEMBABAN INKUBATOR MISELUM J.TIRAM  
PADA XILINX SPARTAN-3**

**SKRIPSI  
KONSENTRASI TEKNIK ELEKTRONIKA**

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik



Disusun Oleh:

**SILVESTER EKA JEMALI**  
**NIM. 0810632008**

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dosen Pembimbing II

**Panca Mudjirahardjo, ST., MT.**  
NIP. 19700329 200012 1 001

**Ir. Bambang Siswojo, MT.**  
NIP. 19621211 198802 1 001

**IMPLEMENTASI SISTEM UKUR SUHU DAN PENGENDALIAN  
KELEMBABAN INKUBATOR MISELUM J.TIRAM  
PADA XILINX SPARTAN-3**

Disusun oleh:

**SILVESTER EKA JEMALI  
NIM. 0810632008**

Skripsi ini telah diuji dan dinyatakan lulus pada  
tanggal 27 Desember 2010

**DOSEN PENGUJI**

**Dr. Agung Darwamansyah, ST., MT.**  
NIP. 19721218 199903 1 002

**Mochammad Rif'an, ST., MT.**  
NIP. 19710301 200012 1 001

**Ir. M. Julius St., MS.**  
NIP. 19540720 198203 1 002

**Mengetahui,  
Ketua Jurusan Teknik Elektro**

**Rudy Yuwono, ST., M.Sc.**  
NIP. 19710615 199802 1 003

## ABSTRAK

SILVESTER EKA JEMALI, Jurusan Teknik Elektro, Fakultas Teknik Universitas Brawijaya. Desember 2010, **IMPLEMENTASI SISTEM UKUR SUHU DAN PENGENDALIAN KELEMBABAN INKUBATOR MISELUM J.TIRAM PADA XILINX SPARTAN-3,**

Dosen Pembimbing : Panca Mudjirahardjo, ST., MT., Ir. Bambang Siswojo, MT.

Untuk memperoleh kelembaban relatif tertentu pada ruangan maka dibutuhkan suatu sistem yang dapat mengendalikan kelembaban tersebut sesuai dengan set point yang telah ditentukan. Untuk mendapatkan sistem seperti diatas maka pada skripsi ini penulis mencoba merancang dan membuat sistem monitor suhu dan pengendali kelembaban relatif pada ruangan inkubator rumah jamur.

Kontroler yang digunakan dalam alat ini yaitu FPGA dalam Spartan-3 development board yang berbasis pada program VHDL. Alat ini memiliki 16 masukan, 2 keluaran untuk logika kelembaban dan 7bit keluaran untuk display 7-Segment. Dari keluaran ini akan dihasilkan pengatur kelembaban seperti yang diinginkan dengan penampil suhu lingkungan inkubator. Data suhu dan kelembaban yang diperoleh dari keluaran sensor HSM-20G, data ini berupa data analog yang dikonversi ke digital menggunakan ADC 8 bit 0809 dimana keluaran dari ADC merupakan masukan untuk FPGA. Dengan menggunakan metode Quine McCluskey keluaran FPGA akan menentukan aksi dari aktuator yang disesuaikan dengan kebutuhan Miselum J. Tiram.

Dari penelitian ini didapatkan suatu inkubator Miselum J. Tiram yang sederhana, yang dapat bekerja sesuai dengan rencana, dimana terdapat penyimpangan rata-rata 1.26% untuk kelembaban dan 0,95% untuk suhunya pada saat terjadi gangguan.

**Kata Kunci :** Spartan-3, Kelembaban Miselum J.Tiram,Suhu dan Kelembaban

## PENGANTAR

Puji syukur ke hadirat Tuhan Yang Maha Esa yang telah memberikan segala nikmat dan rahmat-Nya sehingga penyusunan skripsi ini dapat diselesaikan. Karena hanya dengan pertolongan-Nya semata penulis mampu melewati segala kendala yang ada selama penyusunan skripsi ini.

Skripsi dengan judul ” **IMPLEMENTASI SISTEM UKUR SUHU DAN PENGENDALIAN KELEMBABAN INKUBATOR MISELUM J.TIRAM PADA XILINX SPARTAN-3** “ ini disusun sebagai salah satu syarat untuk mendapatkan gelar Sarjana Teknik, di Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.

Dalam penyusunan skripsi ini, tidak lepas dari bantuan, bimbingan, dorongan dan motivasi dari berbagai pihak. Oleh karena itu dengan segala kerendahan hati penulis menyampaikan terima kasih kepada:

1. Bapak dan Ibu tercinta yang selalu memberikan segala perhatian dan doa tulusnya.
2. Rudy Yuwono, ST., M.Sc selaku Ketua Jurusan Teknik Elektro yang telah mengarahkan dan menentukan dosen pembimbing sehingga tercapai kesuksesan ini.
3. Bapak M. Aziz Muslim, ST., MT., Ph.D selaku Sekretaris Jurusan Teknik Elektro, Fakultas Teknik, Universitas Brawijaya.
4. Bapak Panca Mudjirahardjo, ST., MT. dan Ir. Bambang Siswojo, selaku dosen pembimbing yang telah banyak memberikan bimbingan, petunjuk, maupun saran yang sangat membantu penulis dalam menyelesaikan skripsi ini.
5. Bapak dan Ibu dosen beserta staf dan karyawan Jurusan Teknik Elektro, baik secara langsung ataupun tidak langsung telah membantu menyelesaikan skripsi ini.
6. Rekan-rekan di Jurusan Teknik Elektro Universitas Brawijaya angkatan 2005 - 2008 dan keluarga SAP 2006-2009, terima kasih atas kebersamaannya selama ini.
7. Secara khusus saya ucapkan terima kasih kepada Dwi Joko Waloyo, Siswo Adi Wibowo U, Arif Abdul Mannan, M. Samsul Rijal, Leonardo Rommie E dan Tommy Gandes S.P yang selalu memberi dorongan dan dukungan tanpa henti untuk menyelesaikan skripsi ini.
8. Serta semua pihak yang tidak dapat disebutkan satu per satu yang turut membantu, baik secara langsung maupun tidak langsung dalam penyelesaian skripsi ini.

Penulis menyadari bahwa penyusunan skripsi ini masih jauh dari sempurna, untuk itu kebahagiaan bagi penulis bila ada saran dan kritik yang membangun dari para pembaca yang dapat membantu dalam penyempurnaan skripsi ini. Akhirnya penulis berharap semoga skripsi ini dapat bermanfaat bagi para pembaca.

Malang, Desember 2010

Penulis



## DAFTAR ISI

<b>HALAMAN JUDUL</b> .....	i
<b>LEMBAR PERSETUJUAN</b> .....	ii
<b>LEMBAR PENGESAHAN</b> .....	iii
<b>ABSTRAK</b> .....	iv
<b>PENGANTAR</b> .....	v
<b>DAFTAR ISI</b> .....	vii
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xi
<b>BAB I PENDAHULUAN</b>	
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Sistematika Penulisan .....	3
<b>BAB II TINJAUAN PUSTAKA</b>	
2.1 Morfologi Jamur Tiram .....	4
2.2 Lingkungan Hidup jamur Tiram .....	5
2.3 Kelembaban Udara .....	6
2.4 Sensor Suhu dan Kelembaban (HSM-20GH) .....	7
2.5 <i>Analog to Digital Converter</i> 0809 .....	9
2.6 <i>Optocoupler</i> .....	10
2.7 VHDL ( <i>VHSIC Hardware Description Language</i> ) .....	11
2.7.1 Sejarah VHDL .....	11
2.7.2 Keunggulan VHDL .....	13
2.7.3 Sintak VHDL .....	14
2.7.4 Operator dalam VHDL .....	18
2.8 FPGA ( <i>Field Programmable Gate Array</i> ) .....	19
2.8.1 Arsitektur FPGA .....	20
2.8.2 Pemrograman IC FPGA .....	20

2.9 Metode Quine-McCluskey .....	21
2.10 Pengkondisi Sinyal .....	23
2.10.1 Operasional Amplifier ( <i>Op-Amp</i> ).....	23
2.10.2 Penguat Tak Membalik ( <i>Non-Inverting Amplifier</i> ).....	24
2.10.3 Penguat Diferensial ( <i>Differetial Amplifier</i> ).....	25

### **BAB III METODE PENELITIAN**

3.1 Penentuan Spesifikasi Masukan dan Keluaran Sistem.....	27
3.2 Perancangan Perangkat Keras ( <i>Hardware</i> ).....	27
3.3 Perencanaan Diagram Alir Perangkat Lunak.....	28
3.4 Pembuatan Alat .....	28
3.5 Pengujian Alat .....	28
3.5.1 Pengujian Perangkat Keras ( <i>Hardware</i> ) .....	29
3.5.2 Pengujian perangkat Lunak ( <i>Software</i> ) .....	29
3.5.3 Pengujian Keseluruhan Sistem.....	29

### **BAB IV PERENCANAAN DAN PEMBUATAN ALAT**

4.1 Spesifikasi alat .....	30
4.2 Perancangan sistem alat .....	31
4.3 Prinsip kerja .....	32
4.4 Perancangan perangkat keras ( <i>hardware</i> ) .....	32
4.4.1 Perancangan sensor dan pengkondisi sinyal .....	32
4.4.2 Perancangan ADC 0809, NE 555, CD 4066BC .....	36
4.4.3 Perancangan rangkaian driver <i>relay</i> untuk <i>fan</i> DC 12V .....	39
4.4.4 Perancangan rangkaian driver <i>relay</i> untuk <i>sprayer</i> .....	40
4.5 Perancangan mekanik .....	43
4.6 Perancangan diagram alir kontrol pengendalian kelembaban menggunakan metode Quine McCluskey.....	43
4.6.1 Proses metode Quine-McCluskey pada <i>bit</i> 00H hingga 0BH .....	45
4.6.2 Proses metode Quine-McCluskey pada <i>bit</i> F5H hingga FFH .....	46
4.6.3 Perancangan perangkat lunak pada SPARTAN-3.....	48

**BAB V PENGUJIAN DAN ANALISA HASIL DESAIN**

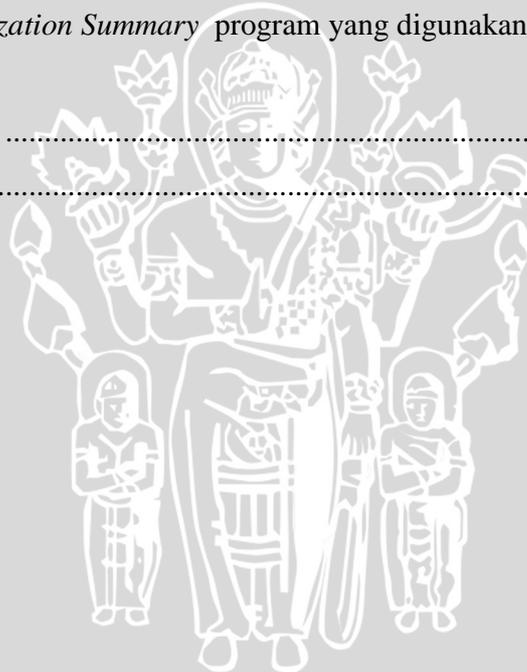
5.1 Pengujian rangkaian kerja sensor HSM-20G .....	49
5.2 Pengujian rangkaian ADC 0809.....	54
5.3 Pengujian Rangkaian <i>Optocoupler</i> .....	56
5.4 Pengujian <i>Driver</i> .....	57
5.5 Pengujian metode Quine-McCluskey pada pengendali kelembaban .....	58
5.6 Pengujian Sistem Secara Keseluruhan ( <i>software</i> ).....	60
5.6.1 Pengujian sistem menggunakan simulasi ModelSim-Altera 6.5b... 60	
5.7 Pengujian praktis alat secara keseluruhan.....	63
5.7.1 Pengujian sistem suhu pada 7-Segment Spartan-3.....	63
5.7.2 Pengujian kelembaban berdasarkan respon waktu.....	64
5.7.3 <i>Utilization Summary</i> program yang digunakan pada sistem.....	66

**BAB VI PENUTUP**

6.1 Kesimpulan .....	67
6.2 Saran .....	67

**DAFTAR PUSTAKA**

**LAMPIRAN**



## DAFTAR TABEL

Tabel 2.1 Karakteristik Standart HSM-20G.....	8
Tabel 2.2 Konfigurasi Pin Sensor Humidity.....	8
Tabel 2.3 Operator Dalam VHDL.....	18
Tabel 4.1 Tabel kebenaran pengendalian kelembaban.....	44
Tabel 4.2 <i>Chart Prime Implicant bit 00H hingga 0BH</i> .....	45
Tabel 4.3 <i>Chart Prime Implicant bit F5H hingga FFH</i> .....	47
Tabel 5.1 Hasil Pengukuran dan Pengujian Suhu.....	51
Tabel 5.2 Hasil Pengukuran dan Pengujian Kelembaban.....	53
Tabel 5.3 Hasil Pengujian Rangkaian ADC 0809.....	55
Tabel 5.4 Hasil Pengujian <i>Optocoupler</i> .....	56
Tabel 5.5 Hasil Pengujian <i>Driver Relay</i> (praktek).....	57
Tabel 5.6 Tabel Kebenaran Setpoint Bawah.....	59
Tabel 5.7 Tabel Kebenaran Setpoint Atas.....	60
Tabel 5.8 Hasil pengujian sistem suhu ditunjukkan dalam 7-Segment.....	63
Tabel 5.9a Hasil Pengujian untuk Pengukuran Kelembaban rendah.....	65
Tabel 5.9b Hasil Pengujian untuk Pengukuran Kelembaban tinggi.....	66

## DAFTAR GAMBAR

Gambar 2.1 Rumah budidaya Jamur Tiram.....	6
Gambar 2.2 Dimensi Sensor Humidity.....	8
Gambar 2.3 Pin Penyemat <i>Optocoupler</i> .....	10
Gambar 2.4 Blok Diagram VHDL.....	15
Gambar 2.5 Arsitektur Bagian Dalam FPGA .....	20
Gambar 2.6 Tahapan Proses FPGA .....	21
Gambar 2.7 Amplifier Operasional .....	24
Gambar 2.8 Penguat Tak Membalik ( <i>Non-Inverting Amplifier</i> ).....	24
Gambar 2.9 Penguat Diferensial ( <i>Differetial Amplifier</i> ) .....	25
Gambar 4.1 Blok Diagram Hardware.....	31
Gambar 4.2 Rangkaian Pengkondisi Sinyal untuk Modul HSM-20G.....	33
Gambar 4.3 Rangkaian Op Amp 1.....	34
Gambar 4.4 Rangkaian Op Amp 2.....	35
Gambar 4.5 Rangkaian ADC, <i>Selector</i> & Pembangkit <i>Clock</i> .....	36
Gambar 4.6 Pin ADC 0809.....	37
Gambar 4.7 Rangkaian Pembangkit Pulsa <i>Clock</i> .....	38
Gambar 4.8 <i>Fan Driver</i> .....	39
Gambar 4.9 <i>Sprayer Driver</i> .....	40
Gambar 4.10 Mekanik Alat .....	43
Gambar 4.11 Diagram Alir Kontrol Kelembaban .....	43
Gambar 4.12 Perancangan Perangkat Lunak.....	48
Gambar 5.1 Pengujian Rangkaian Sensor dan Kelembaban.....	50
Gambar 5.2 Rangkaian Modul HSM 20G .....	50
Gambar 5.3 Grafik Hubungan Antara Suhu T ( °C) dan Hambatan R (k Ω).....	52
Gambar 5.4 Grafik Hubungan Antara Vout dengan Kelembaban (%RH) .....	53
Gambar 5.5 Pengukuran tegangan output op-amp dari HSM-20G .....	54
Gambar 5.6 Pengujian Rangkaian ADC 0809 .....	55
Gambar 5.7 Diagram Blok Pengujian <i>Optocoupler</i> .....	56
Gambar 5.8 Diagram Blok Pengujian <i>Driver</i> .....	67

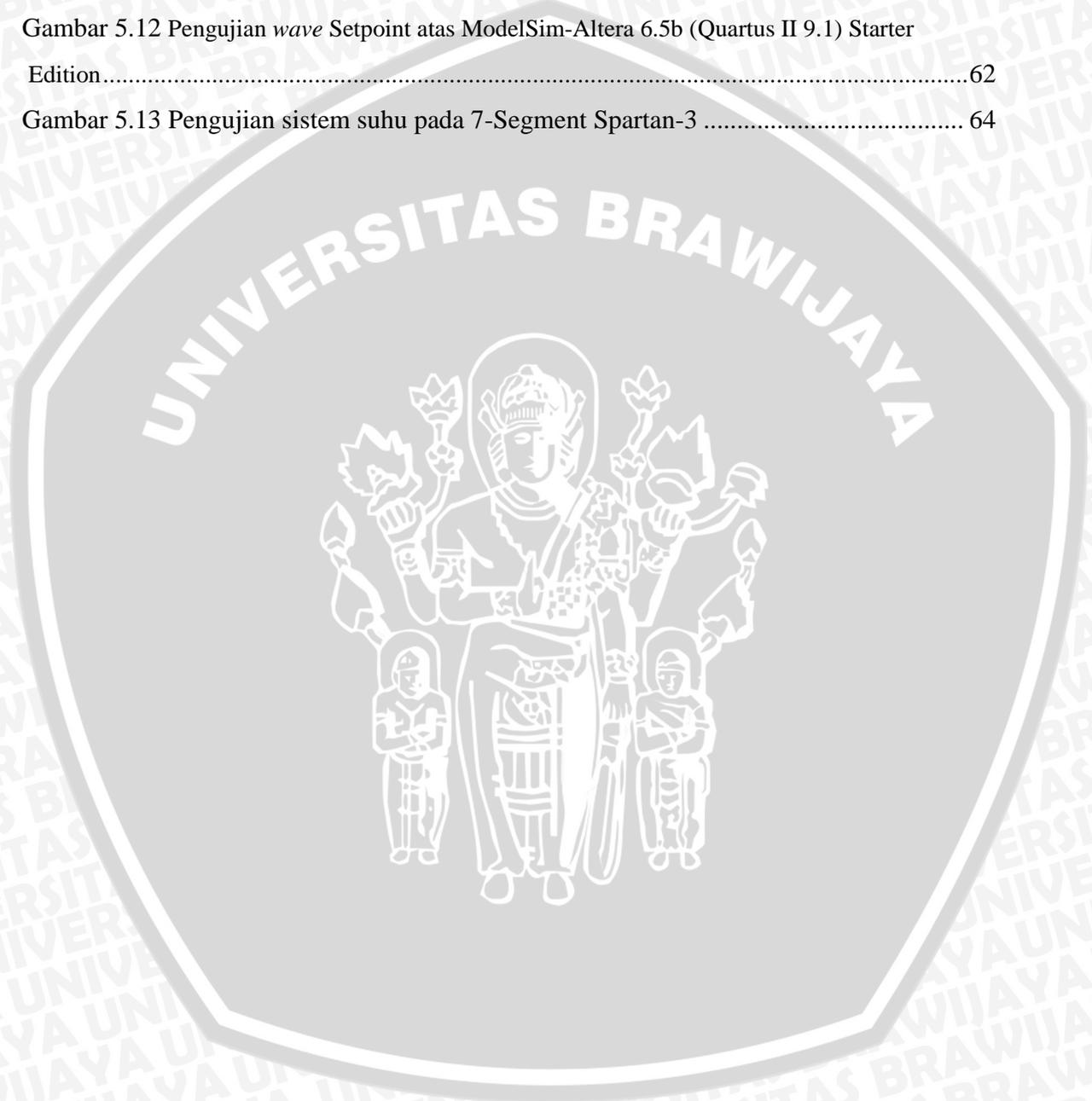
Gambar 5.9 Pengujian Setpoint bawah dengan Multisim Ver.8..... 58

Gambar 5.10 Pengujian Setpoint atas dengan Multisim Ver.8..... 59

Gambar 5.11 Pengujian *wave* Setpoint bawah ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition.....61

Gambar 5.12 Pengujian *wave* Setpoint atas ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition.....62

Gambar 5.13 Pengujian sistem suhu pada 7-Segment Spartan-3 ..... 64



# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Jamur Tiram disebut juga sebagai *Shimeji* atau *Oyster Mushrooms*. Jamur tiram merupakan jenis jamur kayu yang enak dimakan serta mempunyai kandungan gizi yang tinggi. Seiring dengan popularitas dan memasyarakatnya jamur tiram sebagai makanan lezat dan bergizi, maka permintaan konsumen jamur tiram di berbagai daerah terus meningkat.

Suhu dan Kelembaban merupakan hal penting yang mempengaruhi pertumbuhan jamur tiram. Dengan mengatur kondisi lingkungan menggunakan ruang inkubator, diharapkan dapat member kondisi lingkungan seperti yang dibutuhkan jamur tiram. Walaupun ruang inkubator untuk pembudidayaan jamur tiram tersebut sudah banyak dikembangkan, namun pengaturan kelembapan pada ruang inkubator tersebut masih bersifat manual yaitu menggunakan hygrometer udara yang diletakkan di sekitar jamur, ketika kelembapan berada di bawah 65%RH secara manual pekerja melakukan penyemprotan embun pada jamur.

Dengan kemajuan teknologi dewasa ini, khususnya di bidang elektronika, maka makin terasa pula kebutuhan akan peralatan yang dikontrol secara otomatis. Perancangan peralatan elektronik secara otomatis diharapkan dapat mempermudah kegiatan-kegiatan yang dilakukan. Pada sistem kontrol konvensional diperlukan suatu pemahaman mendetail akan semua variabel di dalam sistem serta hubungannya satu dengan yang lain. Hal ini tidak menjadi masalah untuk pengontrolan yang melibatkan sistem yang sederhana dan linier.

Tidak semua tempat memiliki lingkungan udara yang mana suhu dan kelembabannya sesuai dengan lingkungan hidup Miselum J.Tiram dan iklim yang tidak pasti yang diakibatkan pemanasan global membuat adanya beberapa bentuk gangguan terhadap lingkungan Miselum J.Tiram. Maka alat pengatur kelembapanMudara sangat dibutuhkan untuk daerah-daerah yang beriklim kering guna pencapaian produksi jamur tiram secara maksimal.

Dari kasus di atas, penulis dalam penelitian ini merancang dan membuat alat monitor suhu dan pengendali kelembaban udara untuk proses budidaya jamur tiram.

## 1.2 Rumusan Masalah

Dari masalah yang telah dijelaskan diatas, maka dalam perancangan ini rumusan masalahnya adalah:

- 1) Bagaimana mendeteksi perubahan kelembaban dan suhu udara untuk proses budidaya Miselum J.Tiram ?
- 2) Bagaimana memproses data untuk monitoring suhu dan pengendalian kelembaban udara menggunakan FPGA pada Spartan-3 *development board* ?
- 3) Bagaimana merancang sistem yang dapat mengkondisikan kelembaban udara pada jangkauan kelembaban optimal, dengan range 66%RH hingga 89%.RH?

## 1.3 Tujuan

- Tujuan penelitian ini adalah menerapkan FPGA pada Spartan-3 *development board* untuk pengukuran suhu dan pengendalian kelembaban udara dalam Miselum J.Tiram.

## 1.4 Batasan Masalah

Karena begitu luasnya objek kajian maka perlu dilakukan pembatasan masalah agar pembahasan lebih terfokus pada rumusan masalah. Adapun batasan masalah dalam penelitian ini adalah :

- 1) Pengendalian kelembaban ruang inkubator Miselum J.Tiram disesuaikan dengan kebutuhan *Miselum J.Tiram* serta tanpa membahas teknik pembuatan dan instalasi dari pelembab. Serta tidak membahas masalah termodinamika.
- 2) Tidak membahas catu daya dengan dampaknya.
- 3) *Monitoring* suhu dan pengatur kelembaban udara dalam inkubator budidaya pada jangkauan 20°C sampai 50°C untuk suhunya kemudian 65%RH dan 90%RH untuk kelembabannya.

### **1.5 Sistematika Pembahasan**

Sistematika penulisan yang digunakan dalam penyusunan laporan skripsi ini adalah sebagai berikut:

#### **BAB I PENDAHULUAN**

Memuat latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian, dan sistematika pembahasan.

#### **BAB II LANDASAN TEORI**

Membahas teori-teori yang berkaitan dengan pengaturan kelembaban udara dalam incubator budidaya Miselum j.tiram.

#### **BAB III METODOLOGI PENELITIAN**

Membahas metode penelitian yang digunakan dalam perancangan alat.

#### **BAB IV PERENCANAAN DAN PEMBUATAN ALAT**

Berisi penjelasan mengenai perancangan dan pembuatan bagian-bagian dari sistem pengatur kelembaban udara dalam inkubator.

#### **BAB V PENGUJIAN DAN ANALISIS HASIL DESAIN**

Membahas tentang metode dan langkah-langkah pengujian serta analisa alat yang dibuat.

#### **BAB VI PENUTUP**

Berisi tentang kesimpulan dan saran.

## BAB II

### LANDASAN TEORI

Dalam bab ini akan dijelaskan mengenai teori dasar komponen yang digunakan. Adapun dasar teori yang akan dijelaskan meliputi dasar teori tentang Miselum J.Tiram, kelembaban udara, sensor suhu dan kelembaban HSM-20G, Analog Digital Converter 0809, dan VHDL (*VHSIC Hardware Description Language*), FPGA (*Field Programmable Gate Array*), Metode Quine-McCluskey, Operational Amplifier (*Op-Amp*).

#### 2.1 Morfologi Jamur Tiram

Jamur Tiram termasuk keluarga *Agaricaceae* atau *Tricholomataceae* dari kelas *Basidiycetes*. Klasifikasi jamur tiram menurut Alexopolous (1962) adalah sebagai berikut :

Super Kingdom	: <i>Eukaryota</i>
Kingdom	: <i>Myceteae</i>
Divisio	: <i>Amastigomycota</i>
Sub-Divisio	: <i>Basidiomycotae</i>
Kelas	: <i>Basidiomycetes</i>
Ordo	: <i>Agaricales</i>
Familia	: <i>Agaricaceae</i>
Genus	: <i>Pleurotus</i>
Species	: <i>Pleurotus spp.</i>

Jamur tiram adalah jamur kayu yang tumbuh berderet menyamping pada batang kayu lapuk. Tubuh buah jamur ini memiliki tudung (pileus) dan tangkai (stipe). Jamur ini memiliki tubuh buah yang tumbuh mekar membentuk corong dangkal seperti kerang (tiram) berukuran 5-15 cm dan permukaan bagian bawah berlapis-lapis seperti insang berwarna putih dan lunak. Sedangkan tangkainya dapat pendek atau panjang tergantung pada kondisi lingkungan dan iklim.

Jamur tiram adalah tanaman makroskopik yang tidak memiliki klorofil. Jamur ini memiliki inti, berspora dan merupakan sel-sel lepas atau bersambungan membentuk benang yang bersekat atau tidak bersekat yang disebut hifa. Hifa jamur terdiri dari sel-sel yang berinti satu dan haploid, hifa jamur menyatu membentuk jaringan disebut miselum. Miselum ini bercabang-cabang dan pada titik pertemuannya membentuk bintik kecil yang disebut sporangium yang akan tumbuh menjadi pin head (tunas dan calon tubuh buah jamur) dan akhirnya berkembang menjadi jamur (tubuh buah).

## 2.2 Lingkungan Hidup jamur Tiram

Air dibutuhkan untuk memperlancar transportasi atau aliran partikel kimia antar sel yang menjamin pertumbuhan dan perkembangan miselum membentuk tubuh buah sekaligus menghasilkan spora. Pada umumnya, pertumbuhan spora dan miselum membutuhkan kelembaban udara optimal. Jamur tiram memiliki toleransi dan ketahanan terbatas terhadap keasaman (pH) substrat (media tumbuh), kelembaban dan suhu udara lingkungannya. Jamur tiram tumbuh pada tempat-tempat yang mengandung nutrisi berupa senyawa karbon, nitrogen, vitamin serta mineral. Sebagian besar senyawa karbon digunakan sebagai sumber energi sekaligus unsur pertumbuhan. Nitrogen diperlukan dalam sintesis protein, purin dan pirimidin. Jamur menggunakan nitrogen dalam bentuk nitrat, ion amonium, nitrogen organik ataupun nitrogen bebas.

Vitamin diperlukan sebagai katalisator sekaligus berfungsi sebagai koenzim. Macam vitamin yang sangat dibutuhkan untuk pertumbuhan jamur tiram adalah (vitamin B1), Biotin (vitamin B7), inositol, pyridoxine, asam nikotinal (vitamin B3), dan asam pantothenat (vitamin B5). Pada Gambar 2.1 ditunjukkan rumah budidaya jamur tiram yang terdapat pada kalangan umum.



Gambar 2.1 Rumah budidaya Jamur Tiram

Sumber: Sashaoyster's.htm

Jamur tiram tumbuh dan berkembang sepanjang tahun di daerah yang beriklim dingin sampai daratan tropis beriklim panas. Miselum jamur tumbuh optimal pada suhu 25°C sampai 30°C. Masa pertumbuhan miselum membutuhkan kelembaban udara antara 65%RH sampai 90%RH. Tunas dan tubuh buah jamur yang tumbuh pada kelembaban dibawah 60% akan mengalami gangguan absorbs nutrisi sehingga menyebabkan kekeringan dan gangguan pertumbuhan atau kematian. (Sumber: <http://wikimapia.org/11989066/id/Jamur-Tiram-Putih-Kumbang-Agro-Sugih-Makmur>)

### 2.3 Kelembaban Udara

Kelembaban udara menyatakan banyaknya uap air dalam udara. Jumlah uap air dalam udara ini sebetulnya hanya merupakan sebagian kecil saja dari seluruh atmosfer. Akan tetapi uap air ini merupakan komponen udara yang sangat penting ditinjau dari segi cuaca dan iklim. Sebagian gas yang menyusun atmosfer yang dekat dengan permukaan laut relatif konstan dari tempat satu ke tempat yang lain. Sedangkan uap air merupakan bagian yang tidak konstan.

Jumlah uap air maksimum yang dapat dikandung oleh udara pada suatu temperatur disebut kapasitas udara. Besar kecilnya kapasitas udara itu tergantung temperatur. Semakin tinggi temperatur maka semakin besar kapasitas udara dan sebaliknya. Kapasitas udara itu tidak selalu dapat dicapai. Jika kapasitas udara tersebut dapat dicapai berarti udara itu jenuh dengan uap air.

Kandungan uap air dalam atmosfer dinyatakan dalam beberapa cara:

1) Tekanan uap

Bagian dari tekanan atmosfer yang disebabkan oleh uap air. Dinyatakan dalam ukuran yang sama dengan tekanan udara total. Misalnya: atm, milibar atau cm/mm Hg.

2) Kelembaban spesifik

Berat uap air persatuan berat udara (termasuk uap airnya). Biasanya dinyatakan dalam gram tiap air per kg udara. Kelembaban spesifik hampir sama dengan tekanan uap.

3) Kelembaban absolut

Berat uap air persatuan volume udara. Misalnya gram/m<sup>3</sup> udara. Ini kurang digunakan dalam meteorologi oleh karena volume berubah-ubah jika udara naik. Yang berarti kelembaban absolut juga berubah-ubah.

4) Kelembaban relatif

perbandingan antara uap air yang betul-betul ada di udara dengan jumlah uap air dalam udara tersebut jika pada temperatur dan tekanan yang sama udara tersebut jenuh dengan uap air. Jika kelembaban relatif mencapai harga 100% (= 1) berarti udara itu jenuh dengan uap air.

Misalnya udara pada temperatur 34°C untuk mencapai kejenuhan harus ada 8 gram uap air dan ternyata hanya mengandung 6 gram, berarti kelembaban relatifnya =  $6/8 \times 100\% = 75\%$ . Kelembaban relatif dapat pula dirumuskan sebagai: (Soekardi, 1983:51)

$$\frac{\text{kelembaban absolut}}{\text{kapasitas udara}} \quad \text{atau} \quad \frac{\text{kelembaban spesifik}}{\text{kapasitas udara}}$$

Dengan syarat satuan yang digunakan harus sama. Kelembaban relatif dapat berubah baik dengan mengubah jumlah uap air atau mengubah kapasitas udara.

#### 2.4 Sensor Suhu dan Kelembaban (HSM-20G)

Sensor ini mempunyai beberapa karakteristik batas *input* tegangan DC  $5 \pm 0.2$  volt, batas *output* tegangan adalah sebesar DC 1-3 volt, akurasi pengukuran  $\pm 5\%$  RH, operasi arus maksimum 2 mA, batas *storage* RH 0-99% RH, batas operasi RH 20-95% (100% RH *intermittent*), kondensasi *transient*  $< 3\%$  RH, batas *storage temperature*  $-20^{\circ}\text{C}$  sampai dengan  $70^{\circ}\text{C}$ , batas operasi *temperature*  $0^{\circ}\text{C}$ - $50^{\circ}\text{C}$ , *hysteresis* (RH@ $25^{\circ}\text{C}$ ) maksimal 2% RH, sangat linier, respons waktu (63% perubahan *step*) 1 menit.

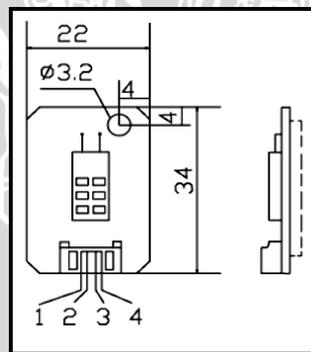
Semua standart alat ini berdasarkan variasi kelembaban di bawah 60%RH pada saat 25<sup>o</sup>C. kelengkapan semua tes-tes yang ada, *module* ini akan melewati batas bawah nominal lingkungan. Dan juga kelembaban untuk 24 jam. Karakteristik standar HSM-20G ditunjukkan Tabel 2.1

Tabel 2.1 Karakteristik Standart HSM-20G

<b>%RH</b>	<b>10</b>	<b>20</b>	<b>30</b>	<b>40</b>	<b>50</b>	<b>60</b>	<b>70</b>	<b>80</b>	<b>90</b>
<b>OutputV</b>	<b>0.74</b>	<b>0.95</b>	<b>1.31</b>	<b>1.68</b>	<b>2.02</b>	<b>2.37</b>	<b>2.69</b>	<b>2.99</b>	<b>3.19</b>

Sumber: Visi Tech Electronics

Pada Tabel 2.1 di atas dapat dilihat *range* atau batas untuk nilai kelembaban pada sensor ini sebagaimana terlihat bahwa nilai tegangan keluaran berbanding lurus dengan persentase kelembaban. Nilai yang tercantum diatas bahwa nilai batas kelembaban maksimum 90%RH dan batas minimum 10%RH dengan tegangan 0.74Volt dan maksimal 3.19Volt. Dimensi sensor *humidity* ditunjukkan Gambar 2.2



Gambar 2.2 Demensi Sensor Humidity

Tabel 2.2. Konfigurasi Pin Sensor Humidity

Pin	Function
1	Temperature Output
2	GND
3	Humidity Output
4	Vcc (+5.0V)

Sumber: Visi Tech Electronics

Pada Tabel 2.2 ditunjukkan konfigurasi pada sensor kelembaban dan temperatur ini. Pada setiap kaki tentu mempunyai fungsi yang berbeda-beda untuk dihubungkan antara satu dengan yang lainnya. Pada sensor ini ada 4 kaki yaitu untuk kelembaban, temperatur, *ground*, dan juga *Vcc*.

## 2.5 Analog to Digital Converter 0809

*Analog to Digital Converter* (ADC) 0809 adalah komponen akuisisi data CMOS monolit dengan *analog-to-digital converter* 8-bit, multiplekser 8-kanal dan kontrol logika yang cocok dengan mikroprosesor. *A/D converter* 8-bit ini menggunakan *successive approximation* sebagai metode konversinya. Waktu konversi selama 100  $\mu$ s cukup cepat untuk aplikasi pengukuran dengan perubahan besaran yang lambat. *Devais* ini tidak membutuhkan pengaturan nol dan skala penuh serta memiliki *interface* yang mudah dengan mikroprosesor. Resolusi 8-bit mencukupi untuk aplikasi pengukuran yang tidak memerlukan ketelitian tinggi.

Untuk mengolah data dari sensor maka dibutuhkan suatu alat yang dapat mengolah data yang disebut mikrokontroler. Dalam skripsi ini pusat sistem yang digunakan adalah FPGA. Akan tetapi FPGA tidak bisa langsung mengolah data dari sensor melainkan membutuhkan perangkat lain sehingga data dari sensor bisa diolah oleh FPGA, perangkat lain ini disebut dengan ADC yang berfungsi untuk mengubah data analog menjadi data digital Berikut beberapa persamaan yang digunakan dalam perhitungan yang terkait dengan ADC. Besar tegangan yang digunakan dalam menentukan tegangan masukan dalam pin-12 ADC ditunjukkan dalam persamaan 2.1

$$V_{out} = V_{ref} \left( \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{256} \right) \quad (2.1)$$

$$\text{Besar tegangan setiap step} = \frac{V_{ref}}{2^n - 1} \quad (2.2)$$

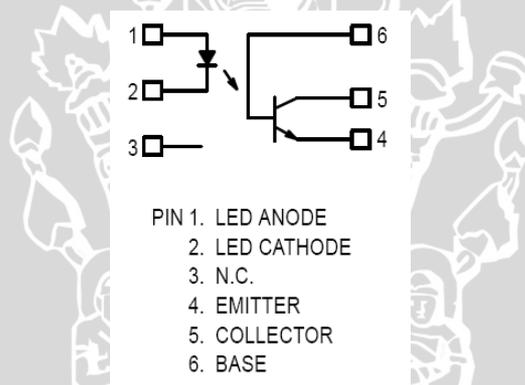
$$\text{Perhitungan bit digital (out)} = \frac{V_{in}}{V_{ref}} \times 256 \quad (2.3)$$

$$\text{Perhitungan dari rangkaian clock ADC adalah } f = \frac{1,44}{(1 + 2.R).C} \quad (2.4)$$

Karena data keluaran dari ADC berupa sinyal digital, maka dapat dihubungkan dengan sistem kontrol utama.

## 2.6 Optocoupler

*Optocoupler* (disebut juga opto isolator optic) yang menggabungkan LED dan photodiode dalam satu kemasan. Optocoupler yaitu suatu alat yang memancarkan suatu foton yang fluknya melalui bahan insulator yang transparan ke jenis detector, alat pemancar foton ini dapat berupa lampu pijar neon ataupun LED. Insulator transparan dapat berupa kaca, udara atau serat optic. Detectornya dapat berupa fotokonduktor, fotodiode, fototransistor, fotoFET, fotoTRIAC, fotoSCR, dan lain sebagainya. Opto mempunyai LED pada sisi masukan dan photodiode pada sisi keluaran. Keuntungan utama optocoupler adalah memisah secara listrik antar rangkaian. Dengan optocoupler, hubungan yang ada antara masukan dan keluaran hanya seberkas cahaya. Karena hal itu, maka kita dapat memperoleh resistansi penyekatan diantara dua rangkaian tersebut dalam ribuan Mega $\Omega$ . Optocoupler yang dipakai adalah yang terdiri dari satu LED dan satu transistor foto seperti yang ditunjukkan Gambar 2.3



Gambar 2.3 Pin Penyemat Optocoupler

Sumber : Motorola, Optoelectronics device data, 1995 : 5 – 6

Optocoupler dirancang sebagai rangkaian isolasi yang memberikan isolasi tingkat tinggi antara terminal input dan terminal output. Keuntungan lain dari optocoupler adalah kecepatan operasi tinggi, ukuran kecil, tahan terhadap getaran dan benturan, dan tidak mempunyai bagian bergerak yang dapat saling melekat.

Isolasi merupakan parameter yang sangat penting pada optocoupler. Tiga parameter isolasi kritis adalah resistansi, kapasitansi isolasi dan ketahanan dielektrik. Resistansi isolasi adalah resistansi DC dari input ke output dari optocoupler. Resistansi isolasi dengan 10 mega ohm adalah umum, nilai ini mungkin lebih tinggi dari pada resistansi antara

penempatan dari PCB dimana optocoupler ditempatkan. Oleh karena itu, penanganan PCB harus hati-hati agar tingkat parameter tidak turun.

## 2.7 VHDL (*VHSIC Hardware Description Language*)

### 2.7.1 Sejarah VHDL

Pada pertengahan tahun 1990-an, Industri elektronik mengalami ledakan dalam kebutuhan akan komputer pribadi, telepon seluler, dan piranti komunikasi data *berkecepatan* tinggi. Berlomba-lomba memperebutkan pangsa pasar, para pengusaha (vendor) membuat produk yang semakin tinggi fungsionalitasnya,

Kinerja yang lebih bagus, harga rendah, konsumsi listrik rendah, dan ukuran yang semakin kecil. Untuk melakukan hal ini, para pengusaha menciptakan sistem-sistem kompleks yang sangat terintegrasi dengan peralatan IC yang lebih sedikit dan area Printed Circuit Board (PCB) yang lebih kecil. Proses semikonduktor sub-mikron pembuatan PCB, dan teknologi pengemasan yang ditunjang integrasi ditingkatkan. Kemacetan atau hambatan bagi sejumlah pengusaha tampaknya berhubungan dengan kemampuan desainer atau perancang menyesuaikan dengan desain-desain yang semakin kompleks, alat-alat automasi desain elektronik (EDA) yang sudah ada, dan jadwal waktu ke pasar (time to market) yang dipercepat. Situasi ini mengangkat kebutuhan akan adopsi yang meluas dari metodologi modern dalam desain dan pengujian. Baik *High Density Programmable Logic Devices (PLDs)* dan *VHDL* menjadi elemen kunci dalam metodologi-metodologi ini.

Piranti-piranti programmable logic dengan kepadatan tinggi, termasuk kompleks *PLDs (CPLDs)* dan *Field Programmable Gate Arrays (FPGAs)*, dapat digunakan untuk mengintegrasikan sejumlah besar logic dalam sebuah IC. Peralatan Application Specific Integrated Circuit (ASIC) baik yang semi custom maupun full custom juga digunakan untuk mengintegrasikan sejumlah besar digital logic, tetapi *CLPDs* dan *FPGAs* menyediakan fleksibilitas tambahan, keduanya dapat digunakan dengan jadwal yang lebih ketat, untuk produk dengan volume rendah, dan untuk perjalanan produksi awal bahkan dengan produk-produk dengan volume tinggi. Keduanya juga menarik untuk proyek-proyek yang memerlukan biaya Non-recurring Engineering (Rekayasa Tak Berulang) yang rendah.

*VHDL* khususnya amat sesuai untuk perancang dengan piranti Programmable logic, dan sedang mencapai popularitas. Untuk merancang dengan *CPLDs* dan *FPGAs* berkapasitas lebih besar 500 sampai dengan lebih dari 100.000 gate, para desainer tidak lagi menggunakan persamaan Boolean atau deskripsi gate level secara cepat dan efisien melengkapinya sebuah desain. *VHDL* menyediakan konstruksi bahasa level tinggi yang memungkinkan perancang untuk mengambarkan sirkuit-sirkuit besar dan membawa produk-produk ke pasar secara cepat. Hal ini menunjang penciptaan perpustakaan desain untuk menyimpan komponen-komponen untuk digunakan lagi pada desain berikutnya. Karena ini merupakan bahasa standart (IEEE Standart 1076). *VHDL* menyediakan kemudahan pemindahan kode antara peralatan sintesis dan simulasi, seperti halnya desain yang tidak tergantung alat. Ini juga memfasilitasi perubahan suatu desain dari sebuah programmable logic menjadi sebuah penerapan ASIC.

Dalam proses pembuatannya, pembuatan menggunakan skematik murni juga mempunyai kelemahan-kelemahan :

- 1) Kontrol logika masih harus dibuat dengan menggunakan teknik-teknik desain tradisional.
- 2) Skematik-skematik dapat sulit dipertahankan sebab maksud dari desain sering dikaburkan oleh penerapannya.
- 3) Sebuah skematik harus sering diikuti dengan dokumentasi untuk mengambarkan sebuah fungsionalitas dari desain, belum lagi penggambaran dalam bahasa inggris atau bahasa lain yang terbuka untuk diinterpretasikan (bebas diartikan).
- 4) Lingkungan pembuatan skematik adalah bersifat kepemilikan, sehingga seorang desainer yang bekerja dalam sebuah lingkungan pembuatan skematik dimana untuk satu proyek yang tidak bisa menggunakan kembali material ketika bekerja pada sebuah proyek baru yang memerlukan penggunaan lingkungan pembuatan skematik yang baru.
- 5) Lingkungan simulasi yang ditunjang oleh alat penangkapan skematik PLD bisa tidak sesuai dengan lingkungan desain sistem, membuat pengujian desain sulit untuk maksimal.

Pada saat ini hanya ada dua bahasa yang memenuhi kebutuhan ini untuk digital logic desain : *VHDL* dan Verilog. Verilog dianggap tidak terlalu rumit dibandingkan *VHDL*, namun seseorang bisa berpendapat bahwa ini adalah akibat dari kekayaan bahasa dan penggambaran-penggambaran. Untuk maksud gabungan dari dokumentasi, sintesis dan simulasi untuk piranti-piranti maupun sistem-sistem, *VHDL* adalah pilihan yang terbaik.

*VHDL* adalah salah satu bahasa deskripsi perangkat keras, yang menyerupai bahasa *C / C++*. Dengan tools ini kita cukup mendeskripsikan *behaviour* perangkat keras yang kita rancang dalam sebuah bahasa pemrograman, dan kemudian melakukan simulasi output perangkat keras tersebut. Setelah rancangan selesai, deskripsi perangkat keras tersebut secara otomatis di *synthesis* oleh tools khusus menjadi rangkaian gerbang-gerbang logika (*AND/OR, XOR.. dsb*). Rancangan dalam bentuk gerbang logika ini kemudian di petakan ke rangkaian transistor untuk dilakukan proses pengaturan penempatan (*placement*) dan interkoneksi (*routing*) diatas silicon Tetapi berbeda dengan bahasa pemrograman algoritma seperti bahasa *C* atau *Java*, dimana proses eksekusi diselesaikan satu persatu, maka pada bahasa pemrograman *VHDL* proses eksekusi berlangsung secara bersamaan. Hal ini dikarenakan *VHDL* mendesain *hardware* yang bekerja secara paralel atau bersamaan.

### 2.7.2 Keunggulan *VHDL*

Setiap simulasi pada industry elektronika dapat mempelajari *VHDL*. Dengan perancangan *VHDL* yang terdiri dari sejumlah gate-gate (gerbang) ini bisa membuat ribuan gerbang yang mana apabila desain dengan menggunakan skema atau persamaan boolean akan memakan waktu yang lebih lama. Selain itu kelebihan dari *VHDL* adalah :

- 1) *Efisiensi dan fleksibel* : *VHDL* adalah bahasa yang dapat digunakan untuk menuliskan kode deskripsi yang lebih efisien untuk mengontrol logika sehingga bisa lebih kompleks. *VHDL* juga menyediakan *library design* yang dapat digunakan sebagai desain maupun simulasi.
- 2) *Desain alat sendiri* : *VHDL* dapat mendesain alat tanpa harus memilih terlebih dahulu alat apa yang akan didesain tersebut dapat diimplementasikan dan waktu yang ada bisa dikonsentrasikan pada desainnya.

- 3) *Protobilita* : Karena VHDL merupakan suatu standar dari setiap deskripsi desain sehingga dapat dipakai untuk berbagai macam simulasi sintesa.
- 4) *Kemampuan brchmarking* : VHDL bisa mendesain alat dengan arsitektur alatmaupun sintesa yang berbeda-beda dan tidak perlu memilih terlebih dahulu apakah menggunakan CPLD atau FPGA. Desain dan sintesa terlebih dahulu dilakukan baru memilih IC yang akan digunakan. Sehingga IC-IC yang ada bisa dibandingkan untuk memperoleh IC yang tepat untuk desain.
- 5) *Perpindahan ke ASIC* : Dengan tingkat efisiensi yang dihasilkan oleh VHDL, maka setiap produk yang dibuat bisa memiliki fungsi seperti yang diharapkan, sehingga IC yang dibuat menjadi spesifik.
- 6) *Kecepatan proses dan biaya rendah* : Dengan menggunakan VHDL maka kecepatan proses n dapat ditingkatkan dan biaya dapat ditekan. Sebuah IC PLD dapat menggantikan banyak IC logika biasa. Karena mempunyai kemampuan luas, VHDL merupakan bahasa yang melayani kebutuhan desainer-desainer pada banyak level.

### 2.7.3 Sintak VHDL

Secara umum sintak VHDL terdiri dari *library*, *entity*, dan *architecture*. Blok diagram dari sintak *VHDL* yang ditunjukkan Gambar 2.4, *library* digunakan untuk menjelaskan *library* yang diperlukan oleh bagian *architecture*. Dari sebuah desain. Pada bagian *entity* dijelaskan pin-pin yang digunakan sebagai input dan output. Sedangkan bagian *architecture* mendefinisikan secara detail arsitektur program yang dibuat.

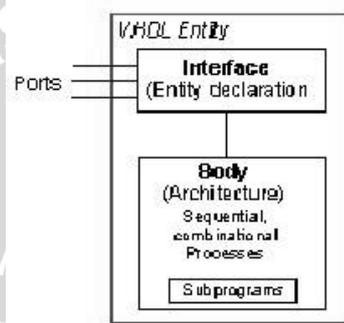
```
library <library_name>;  
use <library_name>;  
use <library_name>;  
....  
entity <entity_name> is  
    port (  
        <port_name>: <port_type>;  
        <port_name>: <port_type>;  
        ....  
    );
```

```

end <entity_name>;

architecture <architecture_name> of <entity_name> is
    local_declaration;
    ....
begin
    program_statement;
    ....
end <architecture_name>;

```



Gambar 2.4 Blok Diagram VHDL

Sumber :Spiengel, 2001:4

Ada tiga pendekatan dalam merancang *architecture* dalam VHDL, antara lain :

a) **Dataflow architecture**

Pada *dataflow architecture*, rangkaian dijelaskan dengan menunjukkan hubungan input output diantara berbagai macam komponen *built-in* pada VHDL, seperti AND, OR, XOR, dan sebagainya. Sehingga pada pendekatan ini kita dapat menjelaskan secara detail bagaimana rancangan suatu rangkaian akan diimplementasikan dalam hardware. *Dataflow architecture* memiliki kelebihan yaitu proses aliran data dalam rangkaian dapat dilihat dengan menganalisis kode VHDLnya. Akan tetapi, *dataflow* bekerja dengan baik untuk rangkaian sederhana. Sehingga *dataflow architecture* tidak efisien apabila diterapkan pada rangkaian yang semakin rumit dan kompleks. pendesain berdasarkan pada proses data transfer ( dari sinyal atau dari input ke output tanpa statement sequential) yang merupakan sekumpulan dari concurrent assignment statement. Perbedaan utama antara data flow dengan behavioral adalah yang satu menggunakan proses dan yang satunya

tidak. Penulisan dengan menggunakan data flow lebih ringkas yaitu menggunakan *Conditional signal assignment* (when-else) statement atau selected assignment (with – select – when) statement. Pada data flow menggunakan *concurrent assignment* lebih banyak digunakan daripada menggunakan proses dan sequential statement.

b) ***Behavioral Architecture***

Sebagai perbandingan dengan dataflow architecture, behavioural architecture tidak menjelaskan secara detail bagaimana rancangan suatu rangkaian akan diimplementasikan dalam hardware. VHDL yang ditulis dengan behavioural architecture tidak menunjukkan bagaimana rangkaian diimplementasikan setelah disintetis. Behavioural architecture menggambarkan bagaimana output rangkaian bekerja atau menunjukkan reaksi pada output rangkaian. pendesain berdasarkan pada proses pengerjaan statement antara masukan dan keluaran port secara berurutan/ step by step yaitu dengan menggunakan statement sequential. Keuntungan dari behavioral description merupakan *high level description* adalah kita tidak perlu menfokuskan pada gate yang menakutkan model fungsi. Proses statement dimuai dengan menggunakan sebuah label yang akan diikuti oleh tanda (:), kemudian kata 'proses' dan sensitivity list dan dibawahnya diikuti dengan sequential statement, setelah bagian sequential statement selesai diakhiri dengan 'End proses'.

c) ***Structural Architecture***

Pada Structural Architecture, suatu rangkaian digambarkan sebagai interkoneksi sinyal sub bagian dalam rangkaian tersebut, Tiap sub bagian dapat dideskripsikan sebagai interkoneksi sinyal sub bagian, dan seterusnya hingga mencapai komponen yang paling sederhana yang hanya dideskripsikan dengan behaviour-nya saja. Pendesain didasarkan pada pemakaian komponen-komponen (logical gate) pada library dan hubungan antara komponen-komponen tersebut. Tipe deskripsi pernyataan pada VHDL adalah concurrent statements. Tipe ini ada beberapa macam yaitu :

### ❖ *Concurrent Signal Assignment Statements*

Deskripsi sintaks dari bahasa pemrograman VHDL menggunakan konsep paralel atau bersamaan. Pernyataan dalam VHDL hamper serupa dengan pernyataan dalam pemrograman algoritma, akan tetapi dalam VHDL terdapat eksekusi pernyataan secara bersamaan. Pada Sintaks VHDL terdapat operator “<=” yang merupakan operator signal assignment.

```
architecture <Architecture_name> of <Entity_name_name> is
begin
target <= statement;
target <= statement;
end <Architecture_name>;
```

### ❖ *Conditional signal assignment statements*

Pada deskripsi pernyataan sebelumnya yaitu *concurrent signal assignment statements*, satu target hanya diasolasijan dengan satu pernyataan. Pada *Conditional signal assignment statements*, satu target dapat diasosiasikan dengan suatu kondisi tertentu. Sintaks untuk *Conditional signal assignment statements* ditunjukkan sebagai berikut :

```
architecture <Architecture_name> of <Entity_name_name> is
begin
target <= statement when condition else
statement when condition else statement;
end <Architecture_name>;
```

### ❖ *Selected signal assignment statements*

*Selected signal assignment statements* adalah tipe ketiga dari *concurrent statements*. Deskripsi ini memiliki satu target yang diasosiasikan dengan satu pernyataan yang menentukan pilihan operasi yang akan dieksekusi. Sintaks untuk *selected signal assignment statements* ditunjukkan sebagai berikut:

```

architecture <Architecture_name> of <Entity_name_name> is
begin
with <choose_statement> select
target <= statement when choices
    statement when choices;
end <Architecture_name>;
    
```

❖ **Process statement**

Pada dasarnya pernyataan dengan *process statements* ini hampir sama dengan *concurrent signal assignment statements*. Yang membedakan adalah pada *process statement* terdapat *sequential statement*. Sintaks untuk *process statement* ini ditunjukkan sebagai berikut:

```

architecture <Architecture_name> of <Entity_name_name> is
label: process <sensitivity_list>

begin
target <= sequential_statement;
target <= sequential_statement;
end <Architecture_name>;
    
```

**2.7.4 Operator Dalam VHDL**

Dalam VHDL terdapat operator-operator yang masing-masing memiliki fungsi tersendiri dan ditunjukkan Tabel 2.3

Tabel 2.3 Operator Dalam VHDL

Fungsi	Operator
Logika	Not, and, or, nand, nor, xor
Relasi	=, /, <, <=, >, >=
Aritmatika	+, -, *, /, **, mod, rem, abs
Tanda	+, -
Penggabung	&

(David E. Vanden Bout, 2001 : 5)

Untuk operator dalam kelompok logika yang hanya dapat digunakan oleh tipe Boolean, bit, dan array. Operator dalam kelompok relasi dapat digunakan oleh semua macam tipe dan hasilnya dalam bentuk boolean (*true or false*). Operator dalam kelompok



aritmatika hanya dapat digunakan oleh tipe integer dan real. pada table di atas terdapat operator-operator ‘\*\*’ yang berate perangkat, sedangkan ‘mod’ singkatan dari modulo (hasil bagi bulat), ‘rem’ singkatan dari remainder (sisa hasil pembagian), dan ‘abs’ singkatan dari absolute value (harga mutlak).

Operator dalam kelompok tanda hanya dapat digunakan oleh tipe integer dan real, juga untuk tanda positif dan negative. Sedangkan untuk penggabung ‘&’ digunakan untuk menggabungkan dua buah array menjadi satu, contohnya (‘1’, ‘0’, ‘0’) & (‘1’, ‘1’, ‘1’) menjadi (‘1’, ‘0’, ‘0’, ‘1’, ‘1’, ‘1’).

## 2.8 FPGA (*Field Programmable Gate Array*)

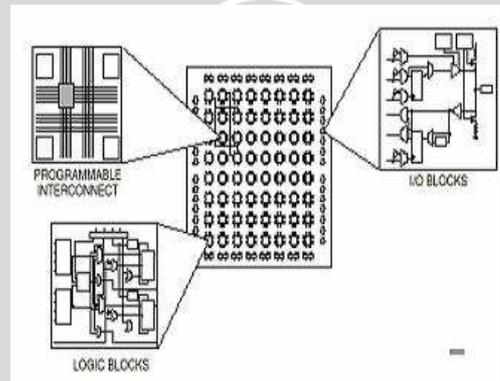
*Field Programmable Device* (FPD) adalah jenis IC yang digunakan untuk mengimplementasikan *hardware* digital, dimana penggunaanya dapat merancang bermacam desain dengan IC ini. FPD memiliki banyak jenis, yaitu PAL, PLD, CPLD dan salah satunya adalah FPGA. PAL, PLD dan CPLD biasanya memiliki kapasitas lebih kecil daripada FPGA dan dapat diimplementasikan dengan *Sum of Products*, dan *Products of Sum*. FPGA biasanya berbasis Flash, SRAM, EEPROM, dan konektivitas *Anti-Fuse*. Dengan FPGA kita dapat melakukan perancangan sistem digital yang selanjutnya menjadi *prototype* rancangan sistem untuk dapat dimanufaktur. FPGA menggunakan *Hardware description Language* (HDL) seperti VHDL atau Verilog. Beberapa keunggulan dari FPGA antara lain :

- 1) Tidak terikat dengan ketersediaan IC
- 2) Perancangan IC tanpa soldering
- 3) Desain sepenuhnya *software-based*
- 4) *Time to market* lebih singkat
- 5) Dapat menggantikan seperangkat rak penyimpanan IC dan kabel jumper.
- 6) Ideal untuk untuk aplikasi pembuatan *prototype*
- 7) Bisa diimplementasikan untuk hardware yang bias dikonfigurasi ulang secara cepat.

### 2.8.1 Arsitektur FPGA

Seperti yang ditunjukkan Gambar 2.5 secara umum arsitektur bagian dalam dari IC FPGA terdiri atas tiga elemen utama yaitu :

- 1) Configurable Logic Blocks :
  - Look up table based complex structure
  - Implement the sequential circuit
- 2) Programmable Interconnect :
  - Berisi wire segments dan programmable switches
  - Menghubungkan anata Configurable Logic Blocks yang berbeda
- 3) Input/Output Block (IOB) :
  - Sebagai interface antara external package pin dari device dan internal user logic



Gambar 2.5 Arsitektur Bagian Dalam FPGA

Sumber : Spiengel, 2001:4

### 2.8.2 Pemrograman IC FPGA

Seperti yang ditunjukkan Gambar 2.6 proses pemrograman IC FPGA adalah sebagai berikut :

#### 1) *Synthesis*

Pada tahap ini *VHDL Source code* dikonversi menjadi gerbang-gerbang logika menghasilkan file yang disebut *netlist*.

#### 2) *Map, Place and Route*

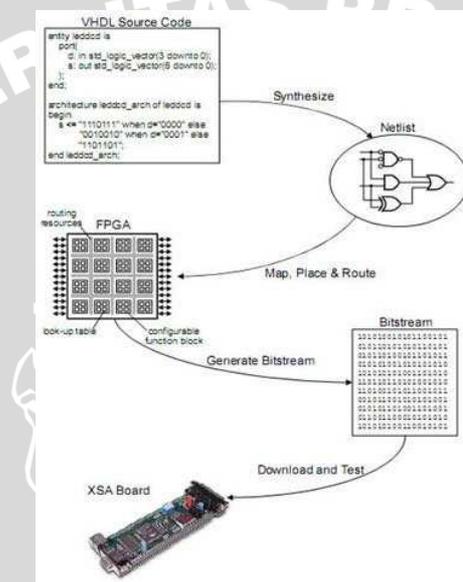
Setelah disintesis, dilakukan proses implementasi yang meliputi proses *map*, *place* dan *route*. Pada tahap ini proses disesuaikan dengan spesifikasi yang disediakan *vendor*, seperti Xilinx atau Altera.

### 3) Generate Bitstream

Setelah dimplementasi, dibuat file bitstream yang merupakan file yang akan di-*download* ke hardware (FPGA chip).

### 4) Download and Test

Pada tahap ini file bitstream yang dihasilkan akan di-*download* ke hardware (FPGA chip). *Download* dapat dilakukan dengan koneksi serial, kabel JTAG atau diprogram dalam ROM



Gambar 2.6 Tahapan proses FPGA

Sumber : Spiengel, 2001:4

## 2.9 Metode Quine-McCluskey

Metode peta Karnaugh hanya cocok digunakan jika fungsi Boolean mempunyai jumlah peubah yang tidak besar. Jika peubah yang terlibat pada suatu fungsi Boolean dalam jumlah yang besar maka penggunaan peta Karnaugh menjadi semakin rumit, sebab ukuran peta bertambah besar. Selain itu, metode peta Karnaugh lebih sulit diprogram dengan komputer karena diperlukan pengamatan visual untuk mengidentifikasi *minterm* – *minterm* yang akan dikelompokkan. Untuk itu diperlukan metode penyederhanaan yang

lain yang dapat diprogram dan dapat digunakan untuk fungsi Boolean dengan sembarang jumlah peubah. Metode alternatif tersebut adalah metode Quine-McCluskey.

Metode Quine-McCluskey adalah sebuah metode yang digunakan untuk menyederhanakan fungsi Boolean, khususnya fungsi Boolean yang memiliki jumlah peubah yang besar (di atas 6 buah). Metode Quine-McCluskey dikembangkan oleh W.V. Quine dan E.J. McCluskey pada tahun 1950.

Metode ini mengubah sebuah fungsi Boolean menjadi sebuah himpunan bentuk prima, dimana sebanyak mungkin peubah dieliminasi (dihilangkan) secara maksimal, hingga didapat fungsi Boolean yang paling sederhana. Ini dapat dilakukan dengan melakukan perulangan penggunaan hukum komplemen,  $a + a' = 1$

Secara umum, langkah-langkah metode Quine-McCluskey untuk menyederhanakan ekspresi Boolean dalam bentuk SOP adalah sebagai berikut :

- 1) Nyatakan tiap *minterm* dalam  $n$  peubah menjadi *string bit* yang panjangnya  $n$ , yang dalam hal ini peubah komplemen dinyatakan dengan '0', peubah yang bukan komplemen dengan '1'.
- 2) Kelompokkan tiap *minterm* berdasarkan jumlah '1' yang dimilikinya.
- 3) Kombinasikan *minterm* dalam  $n$  peubah dengan kelompok lain yang jumlah '1'-nya berbeda satu, sehingga diperoleh bentuk prima (*prime-implicant*) yang terdiri dari  $n - 1$  peubah. *Minterm* yang dikombinasikan diberi tanda "√".
- 4) Kombinasikan *minterm* dalam  $n - 1$  peubah dengan kelompok lain yang jumlah '1'-nya berbeda satu, sehingga diperoleh bentuk prima yang terdiri dari  $n - 2$  peubah.
- 5) Teruskan langkah 4 sampai diperoleh bentuk prima yang sesederhanan mungkin.
- 6) Ambil semua bentuk prima yang tidak bertanda "√". Buatlah Tabel baru yang memperlihatkan *minterm* dari ekspresi Boolean semula yang dicakup oleh bentuk prima tersebut (tandai dengan "x"). Setiap *minterm* harus dicakup oleh paling sedikit satu buah bentuk prima.
- 7) Pilih bentuk prima yang memiliki jumlah literal paling sedikit namun mencakup sebanyak mungkin *minterm* dari ekspresi Boolean semula. Hal ini dapat dilakukan dengan cara berikut :

- a. Tandai kolom – kolom yang mempunyai satu buah tanda “×” dengan tanda “\*”, lalu beri tanda “√” di sebelah kiri bentuk prima yang berasosiasi dengan tanda “\*” tersebut. Bentuk prima ini telah dipilih untuk fungsi Boolean sederhana.
- b. Untuk setiap bentuk prima yang telah ditandai dengan “√”, beri tanda *minterm* yang dicakup oleh bentuk prima tersebut dengan tanda “√” (di baris bawah setelah “\*”).
- c. Periksa apakah masih ada *minterm* yang belum dicakup oleh bentuk prima terpilih. Jika ada, pilih dari bentuk prima yang tersisa yang mencakup sebanyak mungkin *minterm* tersebut. Beri tanda “√” bentuk prima yang dipilih itu serta *minterm* yang dicakupnya.
- d. Ulangi langkah c sampai seluruh *minterm* sudah dicakup oleh semua bentuk prima.

Langkah – langkah penyederhanaan metode Quine-McCluskey di atas juga berlaku untuk penyederhanaan fungsi Boolean dalam bentuk POS. Bentuk fungsi *output* selalu sama dengan bentuk fungsi *input*, artinya *input* dalam bentuk SOP akan menghasilkan *output* dalam bentuk dalam SOP, dan demikian pula untuk bentuk POS.

## 2.10 Pengkondisi sinyal

Rangkaian pengkondisi sinyal dibentuk berdasarkan hubungan yang linier antara tegangan keluaran dan tegangan masukan rangkaian. Hubungan ini dapat dinyatakan dalam bentuk persamaan garis lurus, yaitu:

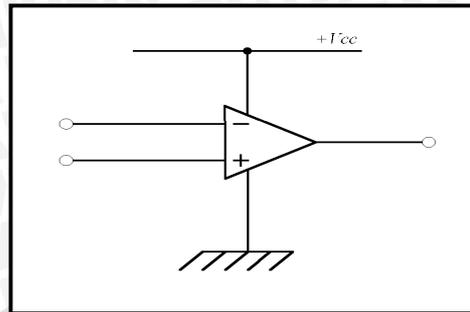
$$V_{out} = mV_{in} + V_o \quad \dots\dots\dots(2.1)$$

Dengan : m adalah kemiringan garis yang menyatakan penguatan

$V_o$  adalah tegangan *offset* keluaran

### 2.10.1 Operasional Amplifier (*Op-Amp*)

Amplifier Operasional sebenarnya dikembangkan dari amplifier diferensial yang digunakan untuk membandingkan dua buah sinyal input. Susunan sirkuit amplifier operasional/*operasional amplifiers (op-amp)* yang ditransistorisasi menjadikannya sangat cocok untuk interasi, sehingga tersedia berbagai jenis op-amp dalam paket IC.



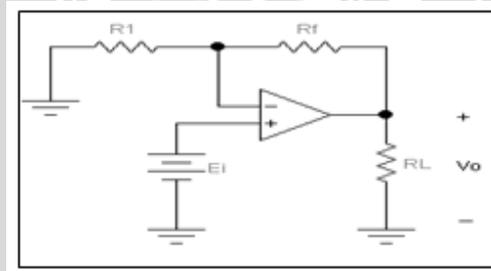
Gambar 2.7 Amplifier Operasional

Sumber : Barry Woollard, 2003

Parameter-parameter op-amp yang ideal adalah : kemampuan, lebar gelombang, dan impedansi *input* yang tidak terbatas, serta arus *input*, *offset*, dan impedansi *output* nol.

### 2.10.2 Penguat Tak Membalik (*Non-Inverting Amplifier*)

Penguat umpan balik tegangan tak membalik adalah penguat tegangan yang mendekati ideal karena impedansi masukan tinggi dan impedansi keluarannya rendah. Pada penguat jenis ini sinyal masuk menggerakkan masukan tak membalik dari penguat, sebagian dari tegangan keluar kemudian dicuplik dan diumpankan kembali ke masukan membalik yang ditunjukkan Gambar 2.8



Gambar 2.8 Penguat Tak Membalik

Sumber : Robert F. Coughlin dan frederich F. Driscoll, 2003

Tegangan keluaran  $V_o$  mempunyai polaritas yang sama dengan tegangan masukan  $E_i$ . Tahanan masukan dari masukan pembalik adalah  $R_i$  tetapi tahanan masukan dari masukan tak membalik adalah besar, biasanya melebihi 100 M $\Omega$ . karena tegangan antara masukan (+) dan masukan (-) secara praktis adalah 0 maka kedua masukan itu berada pada potensial yang sama yaitu  $E_i$ .

Karena  $E_i$  melintasi  $R_1$  dan  $E_i$  menyebabkan arus  $I$  mengalir seperti diberikan oleh

$$I = \frac{E_i}{R_1} \dots\dots\dots (2-2)$$

Arah  $I$  tergantung pada polaritas  $E_i$ , arus yang mengalir lewat masukan (-) diabaikan karenanya  $I$  mengalir melalui  $R_f$  dan penurunan tegangan melintasi  $R_f$  dinyatakan oleh  $V_{Rf}$  dan dinyatakan sebagai

$$V_{Rf} = I.(R_f) = \frac{R_f}{R_1} \times E_i \dots\dots\dots (2-3)$$

Maka tegangan keluaran  $V_o$  adalah

$$V_o = E_i + \frac{R_f}{R_1} .E_i \dots\dots\dots (2-4)$$

$$V_o = \left(1 + \frac{R_f}{R_1}\right) .E_i \dots\dots\dots (2-5)$$

dengan begitu gain tegangannya yaitu

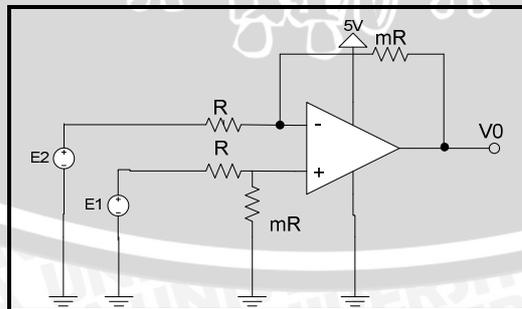
$$A_{CL} = \frac{V_o}{E_i} = 1 + \frac{R_f}{R_1} \dots\dots\dots (2-6)$$

Keterangan:

- $E_i$  = Tegangan Masukan
- $R_i$  = Tahanan Masukan dari Masukan Pembalik
- $R_f$  = Tahanan Referensi

**2.10.3 Penguat Diferensial (*Differetial Amplifier*)**

Penguat diferensial ditunjukkan Gambar 2.9



Gambar 2.9 Penguat Diferensial

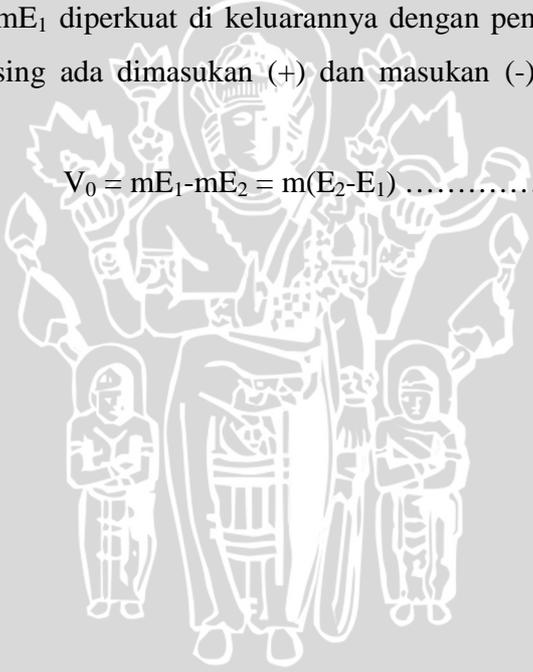
Sumber : Robert F. Coughlin dan frederich F. Driscoll, 2003



Penguat diferensial bisa mengukur maupun memperkuat isyarat-isyarat kecil yang tebenam dalam isyarat-isyarat yang jauh lebih besar. Empat tahanan presisi dan sebuah op amp membentuk sebuah penguat diferensial, seperti terlihat dalam Gambar 2.9 Terminal memasukkannya ada dua, masukan bertanda (-), dan masukan (+) dihubungkan dengan terminal op amp yang terdekat.

Jika  $E_1$  diganti oleh sebuah hubungan singkat,  $E_2$  menghadapi penguat pembalik dengan gain sebesar  $-m$ . karena itu, tegangan keluar akibat  $E_2$  adalah  $-mE_2$ . Sekarang misalkan  $E_2$  dihubungkan singkatkan.  $E_1$  akan terbagi diantara  $R$  dan  $mR$  untuk menerapkan tegangan sebesar  $E_1m/(1+m)$  pada masukan (+) op amp tersebut. Tegangan yang terbagi ini menghadapi penguat tak-membalik dengan gain sebesar  $(m+1)$ . Tegangan keluaran akibat  $E_1$  adalah tegangan yang terbagi  $E_1m/(1+m)$ , dikali gain penguat tak-membalik itu,  $(1+m)$ , yang memberikan  $mE_1$  diperkuat di keluarannya dengan pengali  $m$  menjadi  $mE_1$ . Bila  $E_1$  dan  $E_2$  masing-masing ada dimasukkan (+) dan masukan (-) maka besarnya  $V_0$  adalah  $mE_1 - mE_2$ , atau

$$V_0 = mE_1 - mE_2 = m(E_1 - E_2) \dots\dots\dots (2-7)$$



## BAB III

### METODE PENELITIAN

Penyusunan skripsi ini didasarkan pada masalah yang bersifat aplikatif, yaitu perencanaan dan perealisasiannya dengan mengacu pada rumusan masalah. Pemilihan komponen berdasarkan pada perencanaan dan disesuaikan dengan komponen yang ada di masyarakat atau umum di pasaran. Langkah-langkah yang perlu dilakukan untuk merealisasikan alat yang akan dibuat, secara umum adalah sebagai berikut :

#### 3.1 Penentuan spesifikasi masukan dan keluaran sistem

Penentuan spesifikasi masukan dan keluaran sistem pada prototipe ini meliputi:

- 1) Spesifikasi besaran fisik yang akan diolah dalam hal ini adalah suhu dan kelembaban.
- 2) Penentuan keluaran sistem yaitu *sprayer* dan *fan*.
- 3) Penentuan blok diagram sistem

#### 3.2 Perancangan Perangkat Keras (*Hardware*)

Perencanaan perangkat keras meliputi perencanaan rangkaian mekanik dan perencanaan rangkaian elektronik. Perancangan mekanik yang akan dibuat terdiri dari susunan komponen yang membentuk sebuah ruangan yang dapat mengendalikan suhu dan kelembaban secara otomatis, sedangkan dalam perencanaan rangkaian elektronik dimulai dengan pembuatan diagram blok sistem kendali dan diagram blok sistem alat secara keseluruhan. Setiap bagian dari blok diagram dianalisis, diperhitungkan dan dilakukan perencanaan berdasarkan fungsinya masing-masing dengan mengacu pada spesifikasi keseluruhan yang telah ditentukan, yaitu :

- 1) Perancangan rangkaian pengkondisi sinyal, masukan dari keluaran sensor dan keluaran menjadi masukan ADC.
- 2) Perancangan parameter ADC.
- 3) Perancangan interfacing ADC dengan FPGA

### 3.3 Perencanaan diagram alir perangkat lunak

Perencanaan diagram alir perangkat lunak pada system SPARTAN-3 ini adalah :

- 1) Menentukan logika kombinasi sistem.
- 2) Menulis logika kombinasi tersebut ke dalam bahasa VHDL menggunakan project navigator ISE 7.1i
- 3) Mensimulasikan dengan simulator Model Sim 6.5 Altera Starter Edition sebelum program ditanamkan ke dalam FPGA. Simulasi ini dimaksudkan untuk mengetahui fungsi dan kinerja program yang telah dibuat dengan ISE7.1i
- 4) Melakukan Pemrograman ke dalam Xilinx Spartan-3 *development board* menggunakan kabel downloader JTAC.

### 3.4 Pembuatan alat

Pembuatan inkubator pengendali kelembaban pada ruangan ini meliputi:

- 1) Pembuatan rangkaian mekanik yang membentuk susunan pengukur suhu dan pengendali kelembaban ruangan.
- 2) Pembuatan ruangan inkubator yang dilengkapi bahan triplex yang dilapisi bahan anti air sesuai dengan yang telah direncanakan.
- 3) Pembuatan rangkaian elektronik sesuai dengan komponen yang telah dirancang diatas PCB.
- 4) Pembuatan perangkat lunak untuk menjalankan sistem sesuai dengan diagram alir yang telah dirancang dan selanjutnya diproses system control utama.

### 3.5 Pengujian alat

Pengujian alat ini dilakukan untuk mengetahui unjuk kerja sistem yang telah dirancang, baik perbagian maupun secara keseluruhan, apakah sudah sesuai dengan yang diharapkan. Pengujian alat dilakukan dengan mengubah-ubah suhu dan kelembaban dalam ruangan serta dibantu dengan software simulasi dan program-program yang dibuat khusus untuk tujuan pengujian sistem. Pengujian ini meliputi:

### 3.5.1 Pengujian Perangkat Keras (*Hardware*)

Pengujian perangkat keras dilakukan dengan tujuan untuk menyesuaikan nilai tegangan dan arus pada alat dengan cara pengukuran yang dibandingkan dengan nilai tegangan dan arus yang diijinkan bekerja dalam komponen berdasarkan *datasheet* dari komponen tersebut.

### 3.5.2 Pengujian Perangkat Lunak (*Software*)

Pengujian perangkat lunak dilakukan dengan melakukan pengujian pada subrutin-subrutin program dengan memberi masukan dan mengamati keluarannya.

### 3.5.3 Pengujian Keseluruhan Sistem

Setelah perangkat keras (*hardware*) dan perangkat lunak (*software*) diintegrasikan menjadi satu, kemudian dilakukan pengujian sistem secara keseluruhan yang bertujuan untuk mengetahui unjuk kerja dari alat tersebut.



## BAB IV

### PERENCANAAN DAN PEMBUATAN ALAT

Bab ini menjelaskan tentang perencanaan dan pembuatan alat secara keseluruhan. Alat yang dirancang diharapkan mampu memonitor suhu dan mengendalikan kelembaban di dalam model inkubator jamur. Adapun perencanaan serta pembuatan alat ini terdiri dari dua bagian utama yaitu bagian perencanaan dan pembuatan perangkat keras (*hardware*) serta bagian perencanaan dan pembuatan perangkat lunak (*software*). Perencanaan dan pembuatan alat didasarkan pada spesifikasi alat yang telah ditentukan serta diagram blok sistem yang telah direncanakan sebelumnya.

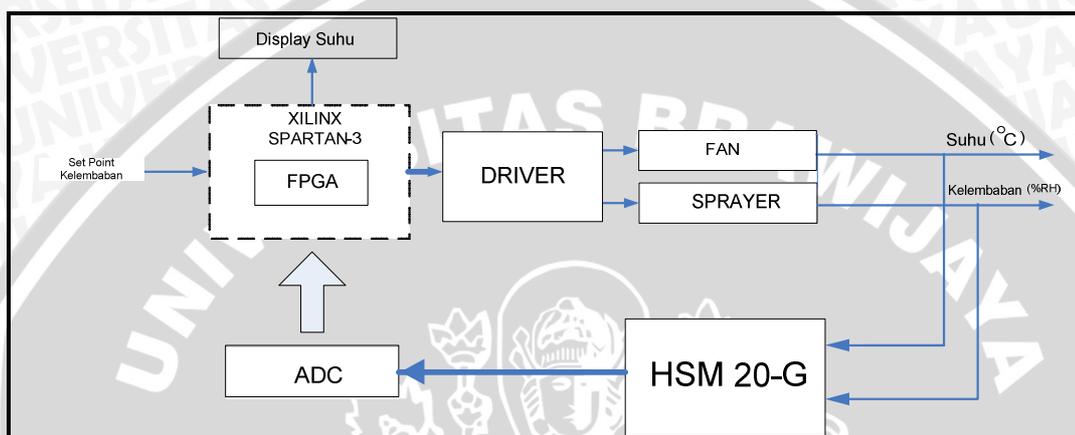
#### 4.1. Spesifikasi alat

Perancangan alat ini mempunyai beberapa spesifikasi diantaranya :

- 1) Inkubator Jamur yang digunakan berupa pemodelan berukuran 90cm x 40cm x 40cm yang berbahan dari kayu dengan pelapis tahan air.
- 2) Menjaga kelembaban di dalam inkubator, yaitu 66%RH-89%RH yang disesuaikan dengan kelembaban lingkungan *Miselum J.Tiram*.
- 3) Pembacaan suhu dengan keadaan terendah adalah 20°C hingga kemampuan maksimal sensor sebesar 50°C.
- 4) Menggunakan sensor HSM-20G
- 5) Menggunakan pengkondisi differensial sebagai pengkondisi sinyal
- 6) Menggunakan ADC 0809 sebagai konverter masukan sinyal analog dari pengkondisi sinyal menjadi sinyal digital sehingga terbaca oleh FPGA.
- 7) Pengendali utama menggunakan FPGA dalam modul Xilinx Spartan-3
- 8) Penampil suhu menggunakan 7-Segment pada modul Xilinx Spartan-3
- 9) Menggunakan rangkaian transistor yang berfungsi sebagai saklar
- 10) Aktuator berupa sebuah *sprayer* dan sepasang *fan*.

## 4.2 Perancangan sistem alat

Diagram blok sistem merupakan salah satu bagian yang penting dalam perancangan suatu alat, karena dari diagram blok dapat di ketahui cara kerja (prinsip kerja) keseluruhan rangkaian. Tujuan lain diagram blok ini adalah memudahkan proses perancangan dan pembuatan pada masing-masing bagian, sehingga akan terbentuk suatu sistem yang sesuai dengan perancangan sebelumnya. Diagram blok sistem ditunjukkan pada Gambar 4.1



Gambar 4.1 Blok Diagram Hardware

Berdasarkan Gambar 4.1 ditunjukkan mengenai bagian-bagian yang menyusun keseluruhan sistem alat ini diantaranya :

### 1) Sensor Suhu dan Kelembaban

Sensor suhu yang digunakan adalah HSM-20G yang berfungsi untuk mendeteksi perubahan suhu dan kelembaban pada inkubator jamur. Perubahan suhu dan kelembaban yang terjadi akan mengakibatkan perubahan tegangan keluaran pada *output* sensor.

### 2) Pengkondisi Sinyal

Pengkondisi sinyal berupa rangkaian yang digunakan untuk mengkondisikan keluaran sensor agar dapat diolah menggunakan Metode Quine-McCluskey.

### 3) *Analog to Digital Converter* (ADC)

ADC yang di gunakan adalah 0809. ADC berfungsi sebagai pengkonversi (*converter*) tegangan analog menjadi digital agar dapat diproses oleh modul Xilinx Spartan-3.

#### 4) *Field Programmable Gate Array (FPGA)*

Pusat Sistem yang digunakan adalah FPGA. Divais ini memiliki beberapa fungsi diantaranya memberika instruksi pada ADC untuk mengkonversikan sinyal analog menjadi data digital, mengkonversikan dan mengolah data keluaran ADC, memberikan instruksi pada 7 Segment untuk menampilkan hasil pengkonversian berupa angka dalam puluhan dan satuan.

#### 5) *Fan*

*Fan* merupakan alat yang digunakan sebagai aktuator untuk membantu dalam pengendalian kelembaban bila kelembabanya  $\geq 90\%RH$ .

#### 6) *Sprayer*

*Sprayer* merupakan alat yang digunakan sebagai aktuator untuk mengendalikan kelembaban bila kelembabanya  $\leq 65\%RH$ .

### 4.3 Prinsip Kerja

Sensor suhu dan kelembaban (HSM-20G) akan memberikan keluaran ke Analog Digital Converter (ADC) yang disesuaikan pada perubahan kondisi udara inkubator berupa data analog kemudian keluaran digital ADC yang berupa data digital menjadi masukan untuk Xilinx Spartan-3. Kemudian sinyal masukan diolah dalam FPGA yang berfungsi sebagai pengendali utama sistem dengan sistem tabulasi. Dengan keadaan kelembaban dibawah 65%RH maka *sprayer* akan aktif (ON) dan *fan* pengering tidak aktif (OFF), kelembaban diantara 66%RH hingga 89%RH *sprayer* dan *fan* tidak aktif (OFF), kelembaban diatas 90%RH maka *sprayer* tidak aktif (OFF), dan *fan* akan aktif (ON). Keadaan suhu di dalam inkubator ditampilkan pada *seven segment* Xilinx Spartan-3.

### 4.4. Perencanaan Perangkat Keras (*hardware*)

Perencanaan perangkat keras (*hardware*) meliputi bagian sensor dan pengkondisi sinyal, ADC 0809, Xilinx Spartan-3, *driver* aktuator

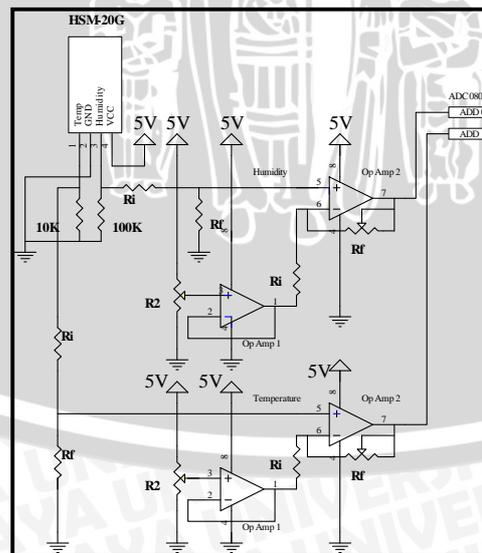
#### 4.4.1. Perencanaan sensor dan pengkondisi sinyal

Sensor suhu dan kelembaban yang digunakan adalah modul HSM-20G, sensor ini mempunyai beberapa karekteristik batas *input* tegangan DC  $5\pm 0.2$ volt, batas *output* tegangan adalah sebesar 1-3volt DC, akurasi pengukuran  $\pm 5\%RH$ , operasi arus maksimum

2mA, batas *storage* RH 0-99%RH, batas operasi RH 20-95% (100%RH *intermittent*), kondensasi *transient* <3%RH, batas *storage temperature* -20°C - 70°C, batas operasi *temperature* 0°C-50°C, *hysteresis* (RH@25°C) maksimal 2%RH, sangat linier, respons waktu (63% perubahan *step*) 1 menit.

Sensor ini dapat mengubah masukan berupa suhu dan kelembaban menjadi keluaran tegangan. Karena tegangan keluaran HSM-20G rendah, maka diperlukan pengondisi sinyal atau penguat tegangan. Sehingga jika terjadi perubahan suhu dan kelembaban, maka perubahan tegangan keluaran pada sensor masih dapat diamati dalam skala yang cukup besar. Dalam perancangan sistem ini nilai *set point* kelembaban yang dikendalikan di dalam inkubator jamur ditetapkan 65%RH.

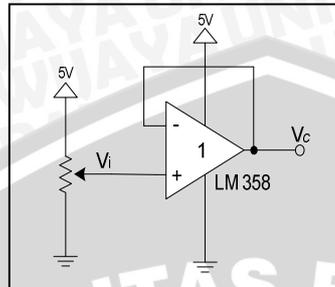
Rangkaian pengkondisi sinyal untuk modul HSM-20G yang dirancang pada sistem ini terbagi menjadi dua macam rangkaian op-amp, yang kemudian digunakan istilah *zero* dan *span*. Rangkaian *zero* berfungsi *virtual input* bagi rangkaian *span*, dimana nilai tegangan keluaran pada rangkaian *zero* tersebut diatur hingga sama dengan nilai tegangan untuk menyatakan batas kondisi suhu dan kelembaban pada range kerja untuk pengukuran suhu dan pengendalian kelembaban. Rangkaian *span* merupakan rangkaian penguat selisih tegangan (penguat differensial) yang berfungsi untuk menguatkan sinyal-sinyal tegangan yang nilainya kecil. Rangkaian pengkondisi sinyal untuk modul HSM-20G ditunjukkan Gambar 4.2.



Gambar 4.2 Rangkaian Pengkondisi Sinyal untuk Modul HSM-20G

Rangkaian pengkondisi sinyal dan analisis yang digunakan pada sistem ditunjukkan Gambar 4.3 dan Gambar 4.4

Rangkaian Op amp 1 ditunjukkan Gambar 4.3



Gambar 4.3 Rangkaian Op Amp 1

$$V_{OUT} = V_{IN} \quad (4-1)$$

a) Op Amp 1 Kelembaban:

Misalkan nilai  $R_1 = 100 \Omega$  dan diketahui  $V_i = 0,2V$ , maka untuk mencari nilai  $R_2$  dilakukan perhitungan sebagai berikut:

$$V_i = \frac{R_2}{R_1 + R_2} \cdot V_{CC}$$

$$0,2 = \frac{R_2}{100 + R_2} \times 5$$

$$\frac{0,2}{5} = \frac{R_2}{100 + R_2}$$

$$5R_2 = 0,2R_2 + 20$$

$$4,8R_2 = 20$$

$$R_2 = \frac{20}{4,8} = 4,17 \cong 1k\Omega$$

b) Op Amp 1 Suhu:

Misalkan nilai  $R_1 = 100 \Omega$  dan diketahui  $V_i = 0,2V$ , maka untuk mencari nilai  $R_2$  dilakukan perhitungan sebagai berikut:

$$V_i = \frac{R_2}{R_1 + R_2} \cdot V_{CC}$$

$$0,2 = \frac{R_2}{100 + R_2} \times 5$$

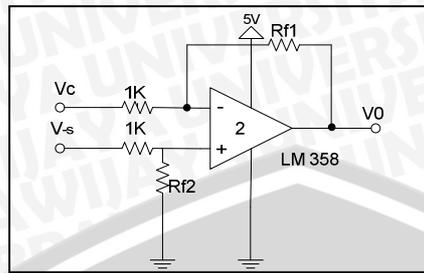
$$\frac{0,2}{5} = \frac{R_2}{100 + R_2}$$

$$5R_2 = 0,2R_2 + 20$$

$$4,8R_2 = 20$$

$$R_2 = \frac{20}{4,8} = 4,17 \cong 1k\Omega$$

Rangkaian Op amp ke 2 ditunjukkan dalam Gambar 4.4



Gambar 4.4 Rangkaian Op Amp 2

Perhitungan untuk op amp 2 adalah sebagai berikut:

a) Op Amp 2 Kelembaban:

Diketahui nilai  $V_a = 4,8V$ ,  $V_{+s} = 3,19 V$  (kondisi pada kelembaban 90%RH),  $R_i = 1k\Omega$  dan  $V_c = 0,2V$  (kelembaban 65%RH), maka untuk mencari besarnya  $R_f$  dilakukan dengan perhitungan sebagai berikut:

$$V_a = \frac{R_f}{R_i} (V_{+s} - V_c)$$

$$R_f = \frac{V_a \cdot R_i}{(V_{+s} - V_c)}$$

$$R_f = \frac{4,8 \cdot (1k)}{(3,19 - 0,2)} \quad R_f = \frac{4,8k}{2,99}$$

$$R_f 1 = 1605,3\Omega \cong 10k\Omega \text{ dan } R_f 2 \cong 1k7\Omega$$

b) Op Amp 2 Suhu:

Diketahui nilai  $V_a = 3,2V$ ,  $V_{+s} = 1,89V$  (kondisi suhu  $50^\circ C$ ),  $R_i = 1k\Omega$  dan  $V_c = 0,2 V$  (suhu  $20^\circ C$ ), maka untuk mencari besarnya  $R_f$  dilakukan dengan perhitungan sebagai berikut:

$$V_a = \frac{R_f}{R_i} (V_{+s} - V_c)$$

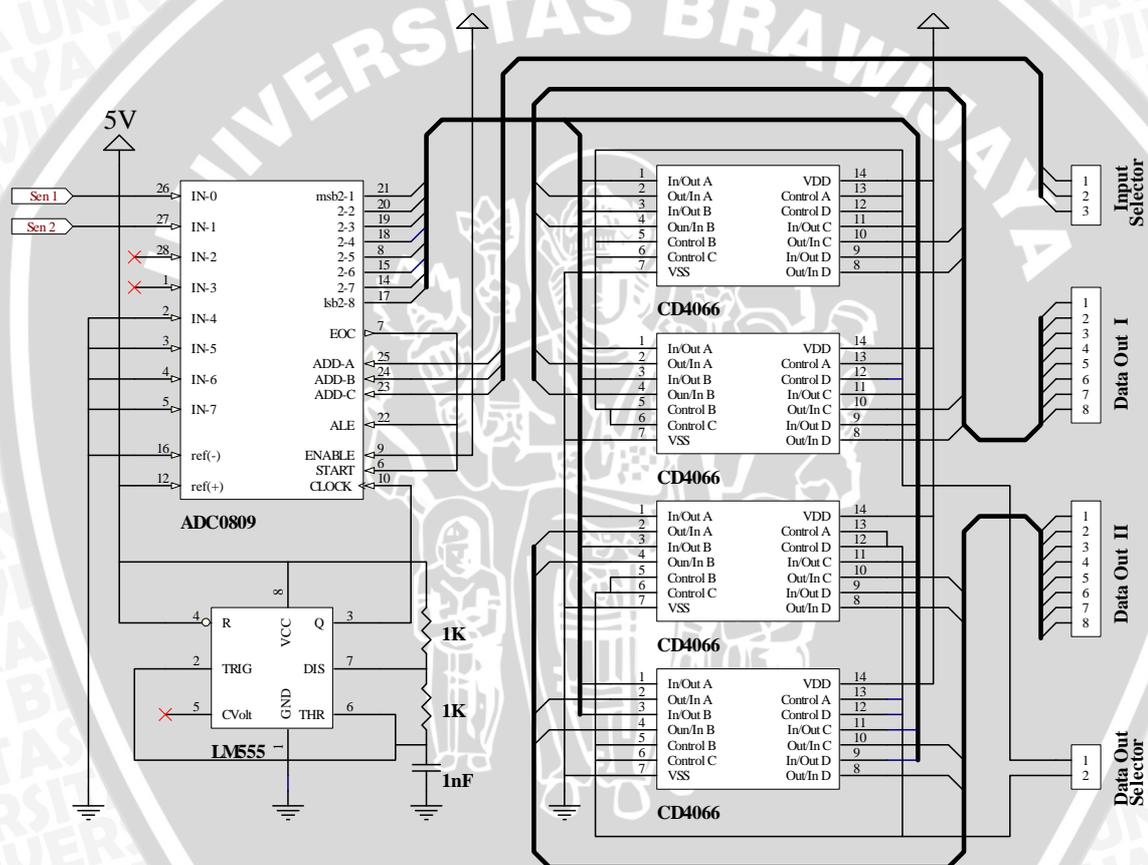
$$R_f = \frac{V_a \cdot R_i}{(V_{+s} - V_c)}$$

$$R_f = \frac{3,2 \cdot (1k)}{(1,89 - 0,2)} \quad R_f = \frac{3,2k}{1,69}$$

$$R_f = 1893,5\Omega \cong 10k\Omega \text{ dan } R_f 2 \cong 1k9 \Omega$$

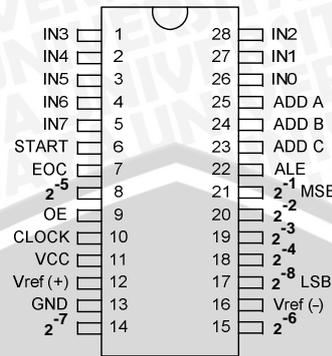
#### 4.4.2. Perencanaan rangkaian ADC 0809, NE 555 dan CD 4066BC

ADC0809 merupakan komponen yang digunakan untuk mengkonversi sinyal analog dari sensor suhu dan kelembaban menjadi data digital. Lebar data keluaran ADC 0809 adalah 8-bit yang nantinya digunakan sebagai masukan untuk FPGA di dalam Spartan-3 yang membutuhkan masukan berupa sinyal digital yang juga lebar datanya 8-bit. Waktu konversi ADC 0809 adalah  $100\mu\text{s}$  yang sangat cepat untuk pengkonversian tegangan dengan perubahan lambat. Rangkaian ADC, NE 555 dan selector ditunjukkan Gambar 4.5



Gambar 4.5 Rangkaian ADC, Selector & Pembangkit Clock

IC ADC 0809 terdiri atas 28 pin yang ditunjukkan dalam Gambar 4.6



Gambar 4.6 Pin ADC 0809

Sumber : National Semiconductor

ADC0809 merupakan pengubah analog ke digital dengan masukan-masukan analog yaitu  $V_{in (+)}$  dan  $V_{in(-)}$  dan mempunyai 8 bit keluaran yaitu DB0 sampai DB7. Dalam perancangan ini ADC mengkonversi antara 0 volt sampai dengan 5 volt, sehingga besar tegangan pada pin 12 ADC 0809 :

$$V_{out} = V_{ref} \left( \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{256} \right)$$

$$5V = V_{ref}(0,996)$$

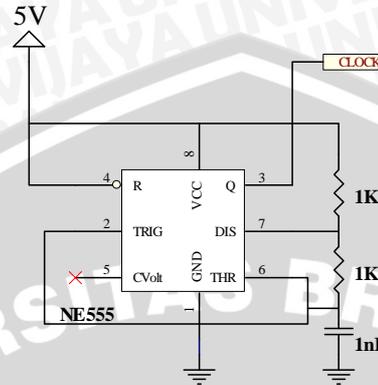
$$V_{ref} = \frac{5}{0,996} = 5,0196 \text{ Volt} \approx 5 \text{ Volt}$$

Sesuai dengan karakteristik ADC0809, tegangan maksimal akan sama dengan 5 volt. Sedangkan tegangan catu 5 volt, keluaran biner ADC akan berada pada range antara 00H (0000 0000B) pada saat tegangan 0 Volt sampai FFH (1111 1111B) pada saat tegangan input mencapai 5 volt. Apabila tegangan maksimalnya 5 volt, maka kenaikan tegangan setiap step adalah:

$$\text{Besar tegangan setiap step} = \frac{V_{ref}}{2^n - 1} = \frac{5V}{2^8 - 1} = 19,6 \text{ mV}$$

Resolusi ADC adalah perubahan terkecil yang terjadi pada masukan analognya yang menyebabkan perubahan 1 bit pada keluarannya. Dengan n adalah jumlah bit keluaran biner IC ADC. Untuk  $V_{maksimal} = 5$  volt dan  $n = 8$ -bit, besar tegangan tiap step adalah  $5/255 = 19,6 \text{ mV}$ .

Untuk menjalankan proses konversi diperlukan sebuah masukan *clock*. Masukan *clock* diperoleh dari rangkaian pembangkit *clock* yang terdiri dari NE555, resistor, dan kapasitor. Rangkaian pembangkit *clock* ditunjukkan Gambar 4.7



Gambar 4.7 Rangkaian Pembangkit Pulsa *Clock*

Perhitungan rangkaian *clock* di atas adalah

$$f = \frac{1,44}{(R1 + 2R2).C}$$

$$f = \frac{1,4}{(1 + 2)k\Omega \cdot 0,001\mu F)}$$

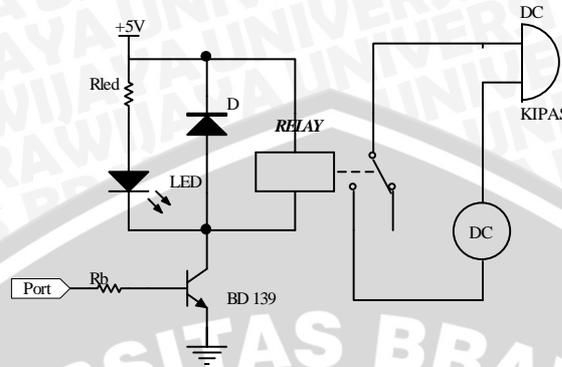
$$f = 480 \text{ kHz}$$

frekuensi tersebut dianggap mampu membuat ADC bekerja secara normal.

Hubungan pin-pin ADC0809 dengan komponen lain ditunjukkan dalam Gambar 4.6. Jalur data 8-bit ADC0809 dihubungkan dengan bus data dari hardware FPGA. Jalur alamat 3-bit, ADD-A, ADD-B, ADD-C dikendalikan secara manual lewat sistem saklar masing – masing melalui jalur alamat A0, A1, A2. Sinyal *Enable* dihubungkan ke keluaran dua rangkaian pengendali ADC. Sinyal EOC dihubungkan ke start ADC dan ALE. Tegangan referensi ADC adalah 0 V untuk  $V_{ref-}$  dan 5 V untuk  $V_{ref+}$ . Karena tegangan kerja rangkaian adalah 5 V, maka tegangan referensi ini dihubungkan dengan catu daya 5Volt yang mencatu rangkaian. Masukan ALE dihubungkan dengan masukan *start* sehingga pensinyalan menjadi lebih sederhana.

#### 4.4.3 Perancangan rangkaian driver *relay* untuk *Fan DC 12V*

Skema perancangan rangkaian driver *relay* untuk *Fan DC 12V* ditunjukkan Gambar 4.8



Gambar 4.8 *Fan Driver*

Driver yang digunakan untuk menggerakkan *relay* AC/DC yang terhubung dengan *fan* DC, menggunakan NPN BD 139 yang difungsikan sebagai saklar, dioda 1N4001 yang dipasang secara paralel dengan *relay* yang berfungsi mencegah terjadinya arus balik pada rangkaian akibat induksi medan magnet yang dihasilkan oleh kumparan *relay* dengan kondisi arus di bawah 1A.

$R_{LED}$  dapat diperoleh dengan perhitungan sebagai berikut :

$$R_{LED} = \frac{V_{cc} - V_{LED} - V_{ceBD139}}{I_{LED}} = \frac{5V - 2V - 0,5}{20mA} = 125\Omega \Rightarrow 220\Omega$$

Diketahui dari pengukuran resistansi dalam *relay* adalah ( $R_{RELAY}$ ) = 400 $\Omega$ .. Maka besarnya arus yang dibutuhkan *relay* adalah :

$$I_{RELAY} \cdot R_{RELAY} = V_{LED} + I_{LED} \cdot R_{LED}$$

$$I_{RELAY} = \frac{2 + 0,02 \cdot 220}{400} = 16mA$$

Sehingga nilai total  $I_c$  bernilai 36mA maka dibutuhkan transistor yang kemampuan  $I_c$ -nya lebih besar daripada 36mA adalah transistor BD 139, dengan spesifikasi  $I_c$  max = 1,5A dengan  $h_{FE}$  63.

$$I_b = \frac{I_c}{h_{FE}}$$

$$I_b = \frac{36mA}{63} = 0,5714 mA.$$

Diketahui *A2 expansion Spartan-3* memiliki port keluaran sebesar 3.3V yang berfungsi sebagai  $V_{bb}$  pada rangkaian driver sehingga untuk mencari nilai resistor dari driver yang menggerakkan relay adalah :

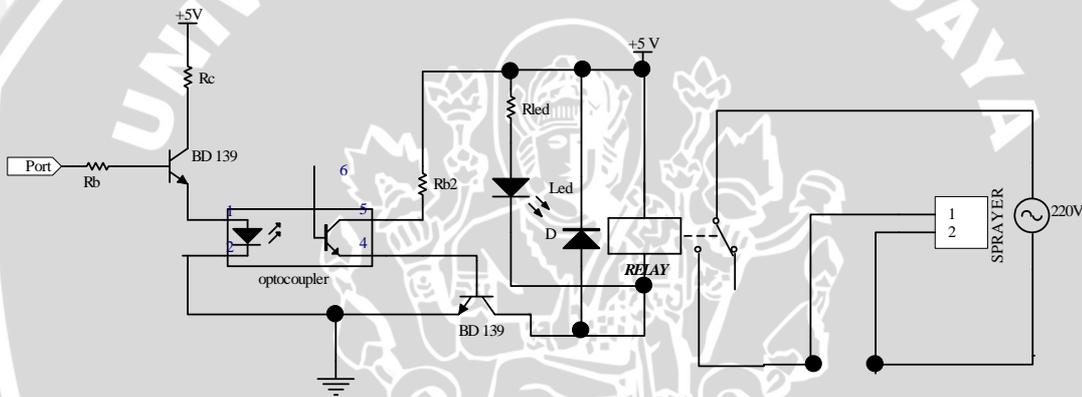
$$V_{bb} - I_b R_b - V_{be} = 0$$

$$R_b = \frac{V_{bb} - V_{be}}{I_b}$$

$$R_b = \frac{3,3 - 0,7}{0,5714mA} = 4550\Omega \approx 4k7\Omega$$

#### 4.4.4 Perancangan rangkaian driver *relay* untuk *sprayer*

Skema perancangan rangkaian driver *relay* untuk *sprayer* ditunjukkan Gambar 4.9



Gambar 4.9 *Sprayer Driver*

Rangkaian driver yang digunakan untuk mengontrol *sprayer* (sebagai penghubung dan pemutus), rangkaian driver ini akan bekerja sesuai dengan perubahan yang terjadi pada input (sensor kelembaban) yang terdiri atas transistor NPN BD139 yang difungsikan sebagai saklar dan dioda 1N4001 yang dipasang secara paralel dengan *relay* yang berfungsi mencegah terjadinya arus balik pada rangkaian akibat induksi medan magnet yang dihasilkan oleh kumparan *relay* dengan kondisi arus di bawah 1A. *Relay* AC/DC yang dikaitkan ke beban yaitu *sprayer* .

Pada rangkaian driver relay menggunakan *optocoupler* jenis 4N25. Berdasarkan *datasheet* didapatkan  $I_{LED}$  optocoupler sebesar 10mA. Sehingga nilai  $R_c$  adalah:

$$I_c = \frac{V_{cc} - V_{ce(sat)}}{R_c}$$

$$R_c = \frac{5V - 0,5V}{10mA} = 450 \Omega = 4k7 \Omega$$

Sehingga dibutuhkan transistor yang kemampuan  $I_c$ -nya lebih besar dari pada 10mA adalah transistor BD139. Dengan spesifikasi  $I_c \text{ max} = 1,5A$  dengan  $h_{FE} = 63$ . Sehingga :

$$I_b = \frac{I_c}{h_{FE}}$$

$$I_b = \frac{10mA}{63} = 0,1587 \text{ mA}$$

Diketahui *A2 expansion Spartan-3* memiliki port keluaran sebesar 3.3V yang berfungsi sebagai  $V_{bb}$  pada rangkaian driver.

Untuk mencari  $R_{b1}$  pada NPN BD139 maka :

$$V_{bb} - I_b R_b - V_{be} - V_{ledopto} = 0$$

$$R_{b1} = \frac{V_{bb} - V_{be} - V_{ledopto}}{I_b}$$

$$R_{b1} = \frac{3,3 - 0,7 - 1,05}{0,1587 \text{ mA}}$$

$$R_{b1} = 9766 \Omega \approx 10k\Omega$$

$R_{LED}$  dapat diperoleh dengan perhitungan sebagai berikut :

$$R_{LED} = \frac{V_{cc} - V_{LED} - V_{ceBD139}}{I_{LED}} = \frac{5V - 2V - 0,5}{20mA} = 125 \Omega \Rightarrow 220 \Omega$$

Diketahui dari pengukuran resistansi dalam *relay* adalah ( $R_{RELAY}$ ) = 400 $\Omega$ . Maka besarnya arus yang dibutuhkan *relay* adalah :

$$I_{RELAY} \cdot R_{RELAY} = V_{LED} + I_{LED} \cdot R_{LED}$$

$$I_{RELAY} = \frac{2 + 0,02 \cdot 220}{400} = 16 \text{ mA}$$

Sehingga nilai total  $I_c$  bernilai 36mA maka dibutuhkan transistor yang kemampuan  $I_c$ -nya lebih besar daripada 36mA adalah transistor BD 139, dengan spesifikasi  $I_c \text{ max} = 1,5A$  dengan  $h_{FE} = 63$ .

$$I_b = \frac{I_c}{h_{FE}}$$

$$I_b = \frac{36mA}{63} = 0,5714 \text{ mA.}$$

Sehingga untuk mencari  $R_b$  dari driver BD139 yang menggerakkan *relay* (ketika optocoupler pada kondisi saturasi) adalah :

$$V_{cc} - I_b R_b - V_{ceopto} - V_{be} = 0$$

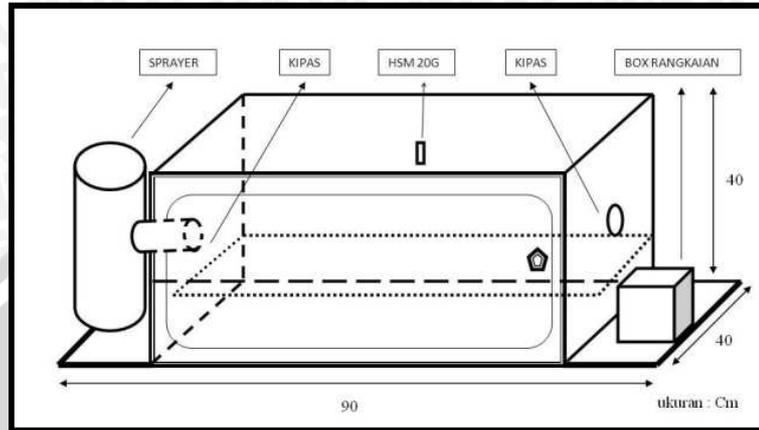
$$R_{b2} = \frac{V_{cc} - V_{ceopto} - V_{be}}{I_b}$$

$$R_{b2} = \frac{5 - 0,5 - 0,7}{0,5714mA} = 6650\Omega \approx 7k2\Omega$$



#### 4.5 Perancangan mekanik

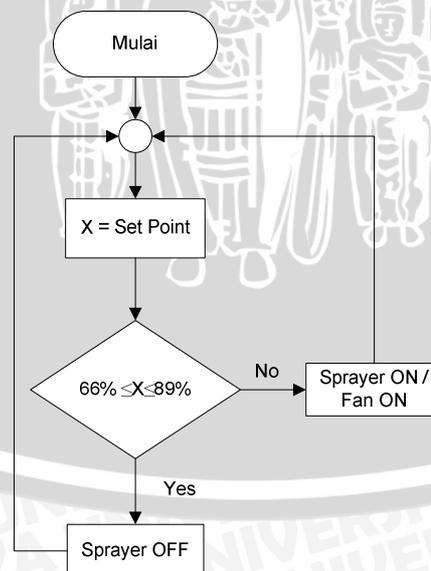
Perancangan mekanik terbuat dari susunan bahan yang berbentuk sebuah model inkubator jamur ditunjukkan Gambar 4.10 :



Gambar 4.10 Mekanik Alat

#### 4.6 Perancangan diagram alir kontrol pengendalian kelembaban menggunakan metode Quine McCluskey

Perancangan diagram alir kontrol pengendalian kelembaban pada inkubator miselum J.Tiram ditunjukkan Gambar 4.11



Gambar 4.11 Diagram Alir Kontrol Kelembaban

Dalam perancangan pengendalian kelembaban yang sebelumnya telah dikalibrasi didapatkan keadaan *setpoint*, yaitu:

- *Setpoint* bawah  $\leq 65\%RH$  dengan tegangan  $\leq 0,2V$  menghasilkan *range* keluaran ADC antara *bit* 00H hingga 0BH untuk mengaktifkan *sprayer*.
- *Setpoint* atas  $\geq 90\%RH$  dengan tegangan  $\geq 4,8V$  menghasilkan *range* keluaran ADC antara *bit* F5H hingga FFH untuk mengaktifkan *fan*.

Dari *setpoint* bawah dan *setpoint* atas didapat tabel kebenaran pengendalian kelembaban yang menentukan aksi dari *fan* dan *sprayer* yang ditunjukkan Tabel 4.1

Tabel 4.1 Tabel kebenaran pengendalian kelembaban

NO	BIT								HEX	Fan	Sprayer
	D7	D6	D5	D4	D3	D2	D1	D0			
0	0	0	0	0	0	0	0	0	00	1	0
1	0	0	0	0	0	0	0	1	01	1	0
2	0	0	0	0	0	0	1	0	02	1	0
3	0	0	0	0	0	0	1	1	03	1	0
4	0	0	0	0	0	1	0	0	04	1	0
5	0	0	0	0	0	1	0	1	05	1	0
6	0	0	0	0	0	1	1	0	06	1	0
7	0	0	0	0	0	1	1	1	07	1	0
8	0	0	0	0	1	0	0	0	08	1	0
9	0	0	0	0	1	0	0	1	09	1	0
10	0	0	0	0	1	0	1	0	0A	1	0
11	0	0	0	0	1	0	1	1	0B	1	0
12	0	0	0	0	1	1	0	0	0C	0	0
13	0	0	0	0	1	1	0	1	0D	0	0
⋮											
243	1	1	1	1	0	0	1	1	F3	0	0
244	1	1	1	1	0	1	0	0	F4	0	0
245	1	1	1	1	0	1	0	1	F5	0	1
246	1	1	1	1	0	1	1	0	F6	0	1
247	1	1	1	1	0	1	1	1	F7	0	1
248	1	1	1	1	1	0	0	0	F8	0	1
249	1	1	1	1	1	0	0	1	F9	0	1
250	1	1	1	1	1	0	1	0	FA	0	1
251	1	1	1	1	1	0	1	1	FB	0	1
252	1	1	1	1	1	1	0	0	FC	0	1
253	1	1	1	1	1	1	0	1	FD	0	1
254	1	1	1	1	1	1	1	0	FE	0	1
255	1	1	1	1	1	1	1	1	FF	0	1

### 4.6.1 Proses metode Quine-McCluskey pada bit 00H hingga 0BH

Dilakukan metode Quine-McCluskey pada bit 00H hingga 0BH menghasilkan logika '1' atau sprayer aktif (ON) .

BATAS BAWAH = 0,2 V  
RANGE TABULASI 00H HINGGA 0BH

LIST 1		LIST 2		LIST 3		LIST 4	
Min	Bit (D) 7654 3210	Min	Bit (D) 7654 3210	Minterm	Bit (D) 3210	Minterm	Bit (D) 7654 3210
00	0000 0000v	00,01	0000 000-v	00,01,02,03	00--	00,01,02,03	0000 00--v
01	0000 0001v	00,02	0000 00-0v	00,01, 04,05	0-0-	00,02, 01,03	0000 00--v
02	0000 0010v	00,04	0000 0-00v	00,01, 08,09	-0-	00,01, 04,05	0000 0-0-v
04	0000 0100v	00,08	0000 -000v	00,02, 01,03	00--	00,04, 01,05	0000 0-0-v
08	0000 1000v	01,03	0000 00-1v	00,02, 04,06	0-0	00,01, 08,09	0000 -00-v
03	0000 0011v	01,05	0000 0-01v	00,02, 08,0A	-0-0	00,08, 01,09	0000 -00-v
05	0000 0101v	01,09	0000 -001v	00,04, 01,05	0-0-	00,02, 04,06	0000 0-0-v
06	0000 0110v	02,03	0000 001-v	00,04, 02,06	0-0-	00,04, 02,06	0000 0-0-v
09	0000 1001v	02,06	0000 0-10v	00,08, 01,09	-00-	00,02, 08,0A	0000 -0-0v
0A	0000 1010v	02,0A	0000 -010v	00,08, 02,0A	-0-0	00,08, 02,0A	0000 -0-0v
07	0000 0111v	04,05	0000 010-v	01,03,05,07	0-1	01,03,05,07	0000 0--1v
0B	0000 1011v	04,06	0000 01-0v	01,03, 09,08	-0-1	01,05, 03,07	0000 0--1v
		08,09	0000 100-v	01,05, 03,07	0-1	01,03, 09,08	0000 -0-1v
		08,0A	0000 10-0v	01,09, 03,08	-0-1	01,09, 03,08	0000 -0-1v
				02,03, 06,07	0-1-	02,03, 06,07	0000 0-1-v
		03,07	0000 0-11v	02,06,03,07	0-1-	02,06,03,07	0000 0-1-v
		03,08	0000 -011v	02,0A, 03,08	-01-	02,0A, 03,08	0000 -01-v
		05,07	0000 01-1v	04,05, 06,07	01--	04,05, 06,07	0000 01--v
		06,07	0000 011-v	04,06,05,07	01--	04,06,05,07	0000 01--v
		09,08	0000 10-1v	08,09,0A,0B	10--	08,09,0A,0B	0000 10--v
		0A,0B	0000 101-v	08,0A, 09,0B	10--	08,0A, 09,0B	0000 10--v
							00,01,02,03,04,05,06,07
							00,01,04,05,02,03,06,07
							00,02,04,06,01,03,05,07
							00,01,08,09,02,0A,03,0B
							00,01,02,03,08,09,0A,0B
							00,02,08,0A,01,03,09,0B
							0000 0--PI <sub>1</sub>
							0000 0--PI <sub>1</sub>
							0000 0--PI <sub>1</sub>
							0000 0--PI <sub>2</sub>
							0000 0--PI <sub>2</sub>
							0000 0--PI <sub>2</sub>

Dilakukan penabelan (*chart prime implicant*) bit 00H hingga 0BH untuk mendapatkan fungsi Boolean sederhana yang ditunjukkan Tabel 4.2

Tabel 4.2 Chart Prime Implicant bit 00 hingga 0B

	√	√	√	√	√	√	√	√	√	√	√	√	√
	Prime	00	01	02	03	04	05	06	07	08	09	0A	0B
*	PI <sub>1</sub>	c	c	c	c	⊙	⊙	⊙	⊙				
*	PI <sub>2</sub>	c	c	c	c					⊙	⊙	⊙	⊙

Dari Tabel 4.2 didapatkan bentuk prima terpilih, yaitu :

- a) **00,01,02,03,04,05,06,07** yang bersesuaian dengan term 0000 0--- adalah

$$D_7' D_6' D_5' D_4' D_3'$$

- b) **00,01,08,09,02,0A,03,0B** yang bersesuaian dengan term 0000 -0--- adalah

$$D_7' D_6' D_5' D_4' D_2'$$

Dengan demikian, fungsi Boolean hasil penyederhanaan dengan metode Quine-McCluskey untuk mengaktifkan *sprayer* adalah

$$\begin{aligned} f(D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0) &= D_7' D_6' D_5' D_4' D_3' + D_7' D_6' D_5' D_4' D_2' \\ &= \mathbf{D_7' D_6' D_5' D_4' \cdot (D_3' + D_2')} \end{aligned}$$

#### 4.6.2 Proses metode Quine-McCluskey pada bit F5H hingga FFH

Dilakukan metode Quine-McCluskey pada bit F5H hingga FFH menghasilkan logika '1' atau *fan* aktif (ON) .

**BATAS ATAS = 4,8 V**  
**RANGE TABULASI F5H HINGGA FFH**

<b>LIST 1</b>		<b>LIST 2</b>		<b>LIST 3</b>		<b>LIST 4</b>	
Min	Bit (D) 7654 3210	Min	Bit (D) 7654 3210	Minterm	Bit (D) 7654 3210	Minterm	Bit (D) 7654 3210
F8	1111 1000 v	F8,F9	1111 100- v	F8,FC,F9, FD	1111 1-0- v	F8,FC,F9, FD, FA,FB, FE,FF	1111 1--- P <sub>1</sub>
6-----		F8,FA	1111 10-0 v	F8,F9, FC, FD	1111 1-0- v	F8,FA, FC, FE, F9, FB, FD,FF	1111 1--- P <sub>2</sub>
F5	1111 0101 P <sub>1</sub>	F8,FC	1111 1-00 v	F8,FA, FC, FE	1111 1-0 v	F8,F9, FA,FB, FC, FE, FD,FF	1111 1--- P <sub>3</sub>
F6	1111 0110 P <sub>2</sub>			F8,FC, FA,FE	1111 1-0 v		
F9	1111 1001 v	F5,F7	1111 01-1 P <sub>3</sub>	F8,F9, FA,FB	1111 10- v		
FA	1111 1010 v	F5, FD	1111 -101 P <sub>4</sub>	F8,FA, F9, FB	1111 10- v		
FC	1111 1100 v	F6, F7	1111 011- P <sub>3</sub>				
7-----		F6, FE	1111 -110 P <sub>5</sub>				
F7	1111 0111 v	F9, FB	1111 10-1 v	F5,F7, FD,FF	1111 -1-1 P <sub>7</sub>		
FB	1111 1011 v	F9, FD	1111 1-01 v	F5, FD, F7,FF	1111 -1-1 P <sub>7</sub>		
FE	1111 1110 v	FA,FB	1111 1-01- v	F6, F7, FE,FF	1111 -11- P <sub>8</sub>		
FD	1111 1101 v	FA,FE	1111 1-10 v	F6, FE, F7,FF	1111 -11- P <sub>8</sub>		
8-----		FC, FE	1111 11-0 v	F9, FB, FD,FF	1111 1-1 v		
FF	1111 1111 v	FC, FD	1111 110- v	F9, FD, FB,FF	1111 1-1 v		
				FA,FB, FE,FF	1111 1-1 v		
				FA,FE, FB,FF	1111 1-1 v		
		F7,FF	1111 -111 v	FC, FE, FD,FF	1111 11- v		
		FB,FF	1111 1-11 v	FC, FD, FE,FF	1111 11- v		
		FE,FF	1111 111- v				
		FD,FF	1111 11-1 v				

Dilakukan penabelan (*chart prime implicant*) bit 00H hingga 0BH untuk mendapatkan fungsi Boolean sederhana yang ditunjukkan Tabel 4.3

Tabel 4.3 *Chart Prime Implicant* bit F5H hingga FFH

	√	√	√	√	√	√	√	√	√	√	√	√
Prima	F5	F6	F7	F8	F9	FA	FB	FC	FD	FE	FF	
<b>PI<sub>1</sub></b>	c											
<b>PI<sub>2</sub></b>		c										
<b>PI<sub>3</sub></b>	c		c									
<b>PI<sub>4</sub></b>	c								c			
<b>PI<sub>5</sub></b>		c	c									
<b>PI<sub>6</sub></b>		c								c		
<b>PI<sub>7</sub></b>	c		c						c		c	
<b>PI<sub>8</sub></b>		c	c							c	c	
* <b>PI<sub>9</sub></b>				©	©	©	©	©	c	c	c	

Dari Tabel 4.3 didapatkan bentuk prima terpilih, yaitu :

- F6** yang bersesuaian dengan term 1111 0110 adalah  $D_7 D_6 D_5 D_4 D_3' D_2 D_1 D_0'$
- F5, F7, FD, FF** yang bersesuaian dengan term 1111 -1-1 adalah  $D_7 D_6 D_5 D_4 D_2 D_0$
- F8, FC, F9, FD, FA, FB, FE, FF** yang bersesuaian dengan term 1111 1--- adalah  $D_7 D_6 D_5 D_4 D_3$

Dengan demikian, fungsi Boolean hasil penyederhanaan dengan metode Quine-McCluskey untuk mengaktifkan *fan* adalah

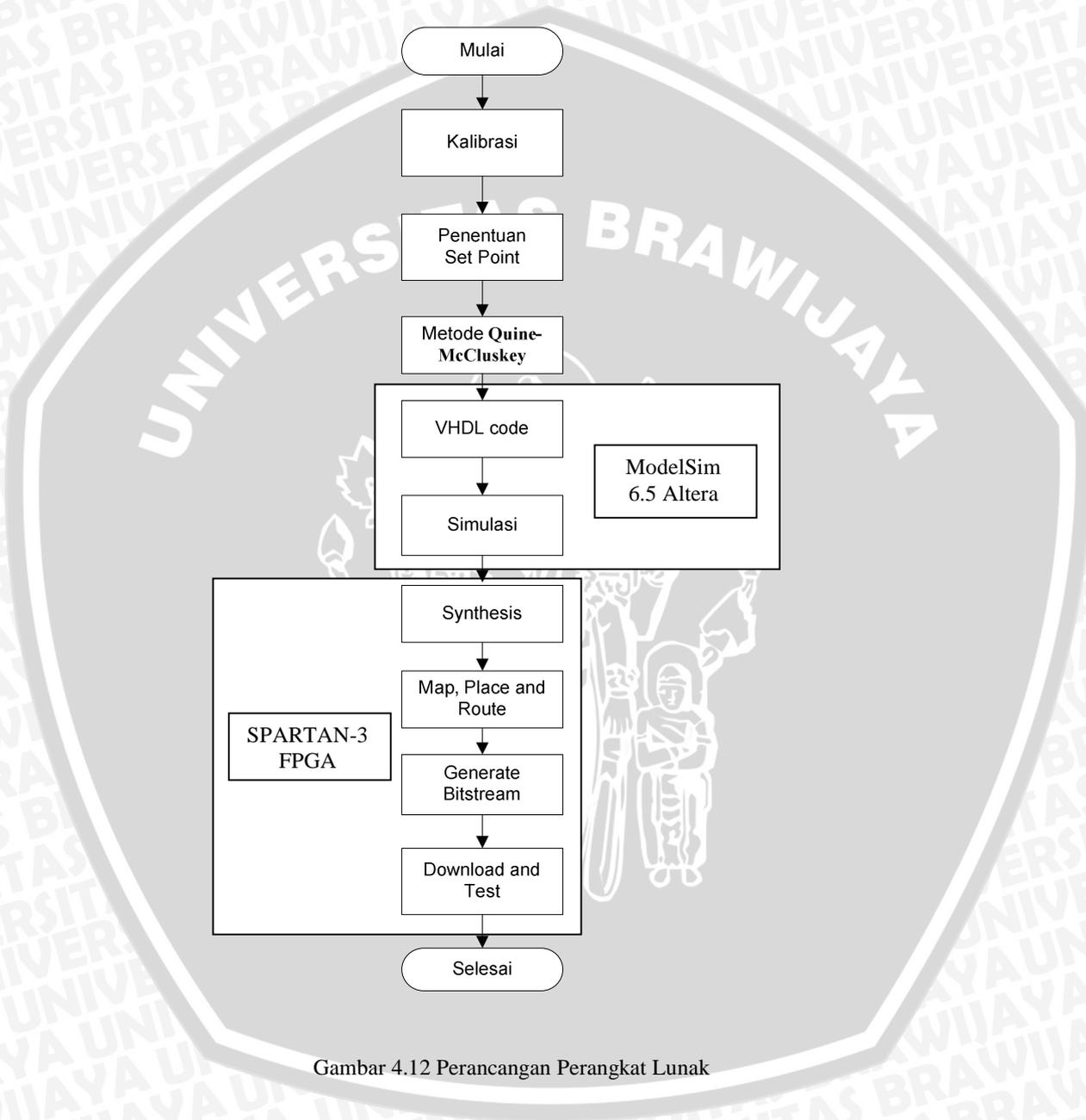
$$f(D_7 D_6 D_5 D_4 D_3 D_2 D_1 D_0) = D_7 D_6 D_5 D_4 D_3' D_2 D_1 D_0' + D_7 D_6 D_5 D_4 D_2 D_0 + D_7 D_6 D_5 D_4 D_3$$

$$= (D_7 D_6 D_5 D_4) \cdot (D_3' D_2 D_1 D_0' + D_2 D_0 + D_3)$$

### 4.6.3 Perancangan perangkat lunak pada SPARTAN-3

Perancangan perangkat lunak pada SPARTAN-3 *development board* ditunjukkan

Gambar 4.12



Gambar 4.12 Perancangan Perangkat Lunak



## BAB V

### PENGUJIAN DAN ANALISIS HASIL DESAIN

Tujuan pengujian alat ini adalah untuk menentukan apakah alat yang telah dibuat berfungsi dengan baik dan sesuai dengan perencanaan. Pengujian ini meliputi pengujian setiap blok maupun pengujian secara keseluruhan. Pengujian setiap blok ini dilakukan untuk mempermudah analisis apabila alat tidak bekerja sesuai dengan perencanaan. Bagian yang akan diuji adalah :

- 1) Pengujian rangkaian sensor suhu dan kelembaban (HSM-20G)
- 2) Pengujian rangkaian ADC (*Analog Digital Converter*)
- 3) Pengujian rangkaian *optocoupler*
- 4) Pengujian *driver*
- 5) Pengujian perangkat lunak (Metode Quine-McCluskey)
- 6) Pengujian sistem keseluruhan

#### 5.1 Pengujian rangkaian kerja sensor HSM-20G

##### 5.1.1 Tujuan :

Untuk mengetahui besarnya tegangan output dari sensor suhu dan kelembaban HSM-20G berdasarkan perubahan kondisi suhu dan kelembaban pada ruangan saat mendeteksi suhu dan kelembaban dan diterima oleh ADC.

##### 5.1.2 Peralatan pengujian

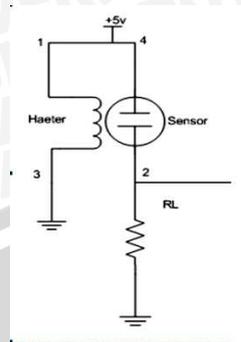
Peralatan pengujian yang digunakan antara lain:

- 1) Multimeter Digital.
- 2) Rangkaian yang akan diuji.
- 3) Catu Daya 5V DC.

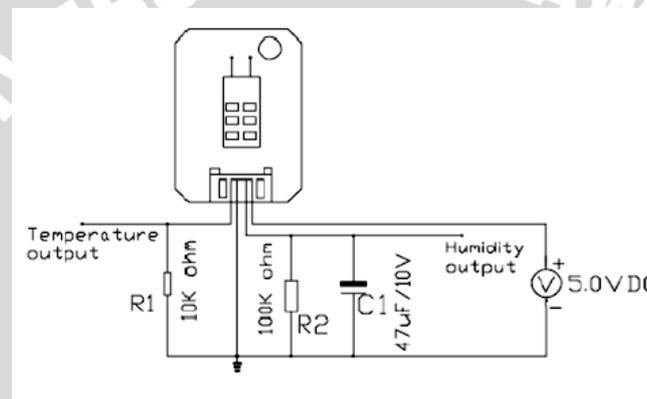
##### 5.1.3 Prosedur pengujian

Prosedur pengujian yang digunakan antara lain:

- 1) Merangkai rangkaian seperti ditunjukkan Gambar 5.1
- 2) Menghubungkan kaki lainnya pada ground.



Gambar 5.1 Pengujian Rangkaian Sensor Suhu dan Kelembaban



Gambar 5.2 Rangkaian Modul HSM 20G

Sumber: Datasheet HSM-20G

Dalam pengujian rangkaian sensor ini dilakukan untuk mengetahui berapa tegangan output dari sensor suhu pada setiap perubahan suhu. Pengujian dan pengukuran rangkaian sensor dilakukan berdasarkan diagram blok di atas. Untuk kalibrasi sensor HSM-20G dengan menggunakan *hygrometer* yang telah dilengkapi dengan termometer. Untuk mencari kemungkinan Error % dapat dicari dengan rumus sebagai berikut:

$$\text{Error(\%)} = \left| \frac{V_{\text{out\_Op-Amp}}(\text{pengukuran}) - V_{\text{out\_Op-Amp}}(\text{perhitungan})}{V_{\text{out\_Op-Amp}}(\text{perhitungan})} \right| \times 100\%$$

### 5.1.4 Hasil pengujian dan Analisis

Diambil suatu sampel, suhu ruangan adalah 30°C. Untuk mengetahui berapa tegangan output yang dihasilkan maka kita dapat diukur pada pin-1 dari HSM 20G.

$$V_{out} = \frac{R_I}{R_s + R_I} \times V_{cc}$$

dengan:

- $V_{out}$  = Tegangan output dari sensor HSM 20G (volt)
- $R_I = 10k\Omega$  dengan  $R_s = 37,43k\Omega$  data sheet untuk 30°C
- Sebagai contoh suhu adalah 30°C maka tegangan dari sensor adalah:

$$V_{out} = \frac{R_I}{R_s + R_I} \times V_{cc}$$

$$V_{out} = \frac{10}{37,43 + 10} \times 5, \text{ dengan keterangan : } V_{out} = 1,054 \text{ V}$$

$$\Delta V = |V_{out\_Op-Amp}(\text{pengukuran}) - V_{out\_Op-Amp}(\text{perhitungan})|$$

$$\Delta V = |1,078 - 1,054| = 0,024$$

$$Error\% = \left| \frac{\Delta V}{V_{out\_Op - Amp}(\text{perhitungan})} \right| \times 100\%$$

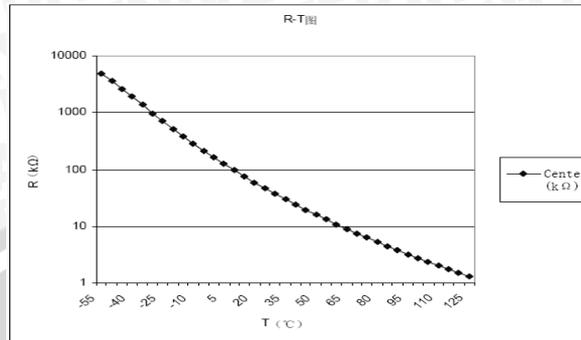
$$Error\% = \left| \frac{0,024}{1,054} \right| \times 100\% \text{ sehingga } Error\% = 2,27\%$$

Dengan cara yang sama, nilai kesalahan relatif ditunjukkan Tabel 5.1 sebagai berikut:

Tabel 5.1 Hasil Pengukuran dan Pengujian Suhu

NO	Temperatur (°C)	Vout Sensor Suhu (Volt)		ΔV	Error%
		Perhitungan	Pengukuran		
1	30	1,054	1,078	0,024	2,27
2	32	1,149	1,153	0,004	0,35
3	34	1,185	1,207	0,022	1,87
4	36	1,357	1,349	0,008	0,59
5	38	1,408	1,393	0,015	1,07
6	40	1,462	1,472	0,010	0,68
<b>Kesalahan rata-rata</b>				0,95 %	

Grafik Hubungan Antara Suhu T ( °C) dan Hambatan R ( kΩ) ditunjukkan Gambar 5.3



Gambar 5.3 Grafik Hubungan Antara Suhu T ( °C) dan Hambatan R ( k Ω)

Diambil suatu sampel, kelembaban adalah 70 %RH. Untuk mengetahui berapa tegangan output yang dihasilkan dapat diukur pada pin-3 dari HSM 20G.

$$V_{out} = \frac{R_1}{R_s + R_1} \times V_{CC}$$

dengan:

- $V_{out} = 2,69$  V data sheet untuk 70%RH
- $R_1 = 100k\Omega$
- $V_{CC} = 5$  V

Sebagai contoh suhu adalah 70%RH maka tegangan dari sensor adalah:

$$V_{out} = \frac{R_1}{R_s + R_1} \times V_{CC}$$

$$R_s = \left( \frac{R_1}{V_{out}} \times V_{CC} \right) - R_1$$

$$R_s = \left( \frac{100}{2,69} \times 5 \right) - 100$$

$$R_s = (37,17 \times 5) - 100 \quad \text{maka } R_s = 85,87 \text{ k}\Omega$$

$$\Delta V = |V_{out\_Op-Amp}(\text{pengukuran}) - V_{out\_Op-Amp}(\text{perhitungan})|$$

$$\Delta V = |2,73 - 2,69| = 0,04$$

$$Error\% = \left| \frac{\Delta V}{V_{out\_Op} - Amp(perhitungan)} \right| \times 100\%$$

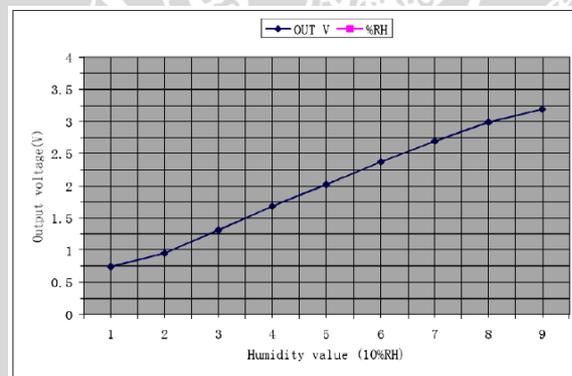
$$Error\% = \left| \frac{0,04}{2,69} \right| \times 100\% \text{ sehingga } Error\% = 1,48\%$$

Dengan cara yang sama, nilai kesalahan relatif ditunjukkan Tabel 5.2

Tabel 5.2 Hasil Pengukuran dan Pengujian Kelembaban

No	Humidity % (RH)	Vout Sensor kelembaban (Volt)		ΔV	Error%
		Perhitungan	Pengukuran		
1	45	1,94	1,91	0,03	1,55
2	50	2,02	2,04	0,02	0,99
3	55	2,20	2,24	0,04	1,8
4	60	2,37	2,40	0,03	1,27
5	65	2,53	2,56	0,03	1,19
6	70	2,69	2,71	0,02	0,74
<b>Kesalahan rata-rata</b>				1,26 %	

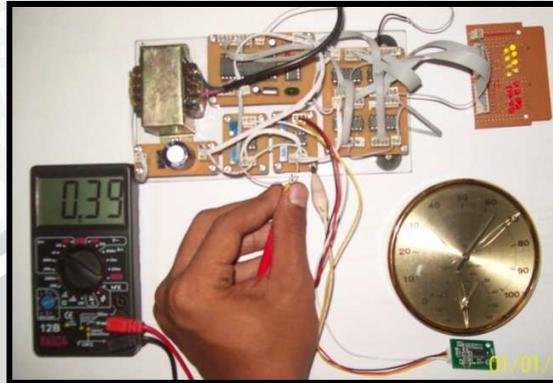
Untuk Grafik Hubungan Antara Vout dengan Kelembaban (%RH) dapat ditunjukkan Gambar 5.4:



Gambar 5.4 Grafik Hubungan Antara Vout dengan Kelembaban (%RH)

Sumber: Datasheet HSM-20G

Pengukuran tegangan output op-amp pada alat yang digunakan sebagai masukan data  $V_{out}$  Op-Amp(pengukuran) ditunjukkan Gambar 5.5



Gambar 5.5 Pengukuran tegangan output op-amp dari HSM-20G

## 5.2 Pengujian rangkaian ADC 0809

### 5.2.1 Tujuan

Pengujian ADC ini dilakukan, untuk mengetahui keluaran digital yang dihasilkan oleh ADC berdasarkan masukan analog yang diberikan serta untuk mengetahui kelinieran konversi ADC

### 5.2.2 Peralatan Pengujian

Peralatan pengujian yang digunakan antara lain:

- 1) Rangkaian ADC 0809
- 2) Catu daya +5 V
- 3) LED

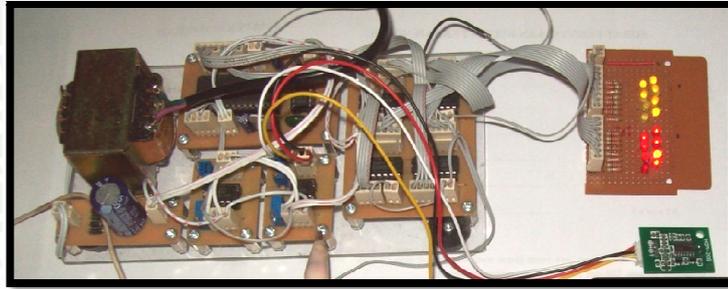
### 5.2.3 Prosedur Pengujian

Prosedur pengujian yang digunakan antara lain:

- 1) Memberikan masukan tegangan pada ADC 0809.
- 2) Mengaktifkan catu daya.
- 3) Mencatat perubahan bit yang tampak pada LED dengan beban sampel tegangan masukan yang berbeda-beda

### 5.2.4 Hasil Pengujian dan Analisis

Rangkaian pengujian ADC 0809 ditunjukkan Gambar 5.6



Gambar 5.6 Pengujian Rangkaian ADC 0809

Hasil pengujian rangkaian ADC 0809 ditunjukkan Tabel 5.3.

Tabel 5.3 Hasil Pengujian Rangkaian ADC 0809

Vin (V)	Keluaran Pengukuran		Keluaran Perhitungan		Error (%)
	Biner	Decimal	Biner	Decimal	
0,5	0001 1001	25	0001 1001	25	0
1	0011 0011	51	0011 0011	51	0
1,51	0101 1100	92	0101 1010	90	2,2
2	0110 0110	102	0110 0110	102	0
2,5	1000 0001	129	1000 0000	128	0,78
3	1001 1010	154	1001 1001	153	0,65
3,5	1011 0100	180	1011 0011	179	0,56
4	1100 1100	204	1100 1100	204	0
4,5	1110 0111	231	1110 0110	230	0,77
5	1111 1111	255	1111 1111	255	0

Dari pengujian didapatkan hasil seperti dalam Tabel 5.3. Kesalahan maksimum ADC yang terjadi hanya sebesar 1 LSB dengan penyimpangan rata-rata sebesar 2,2%, didapat dari :

$$Error\% = \left| \frac{Vo_{Pengukuran} - Vo_{Perhitungan}}{Vo_{Perhitungan}} \right| \times 100\%$$

$$Error\% = \left| \frac{92 - 90}{90} \right| \times 100\% = 2,2\%$$

Dengan kesalahan rata-rata sebesar,

$$Kesalahan\_rata - rata = \frac{4,96}{10} = 0,496\%$$

karena kesalahan rata-rata hanya 0,496% maka ADC ini bisa dikatakan berjalan dengan baik untuk aplikasi yang direncanakan. Kesalahan hasil konversi ADC dapat

disebabkan oleh tingkat akurasi komponen ADC dan juga tidak stabilnya tegangan referensi yang diberikan pada ADC.

### 5.3 Pengujian Rangkaian Optocoupler

#### 5.3.1 Tujuan Pengujian

Untuk mengetahui level tegangan keluaran dari rangkaian sensor posisi ketika cahaya yang dipancarkan oleh LED terhalang atau tidak terhalang oleh plat penghalang dan untuk mengetahui ketepatan dalam pembacaan sensor.

#### 5.3.2 Peralatan Pengujian

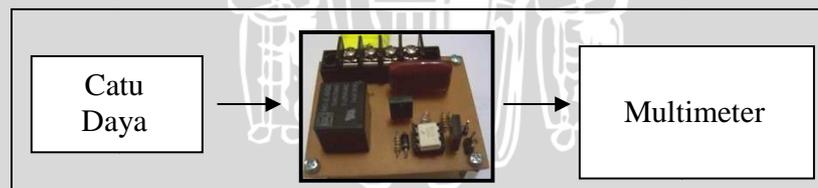
Peralatan pengujian yang dibutuhkan antara lain:

- 1) multimeter digital
- 2) catu daya +5 V
- 3) rangkaian Optocoupler

#### 5.3.3 Langkah Pengujian

Langkah-langkah pengujiannya adalah:

- 1) Merangkai peralatan seperti dalam Gambar 5.7 keluaran rangkaian terhubung dengan multimeter digital.
- 2) Mengaktifkan catu daya.
- 3) Mencatat tegangan yang tampak pada multimeter digital saat kondisi optocoupler gelap dan terang



Gambar 5.7 Diagram Blok Pengujian Optocoupler

#### 5.3.4 Hasil Pengujian dan Analisis

Hasil Pengujian dan Analisis *driver optocoupler* ditunjukkan dalam Tabel 5.4

Tabel 5.4 Hasil Pengujian Optocoupler

Logic	Sprayer
Low	OFF
High	ON

## 5.4 Pengujian Driver

### 5.4.1 Tujuan Pengujian

Pengujian driver bertujuan untuk mengetahui apakah driver berfungsi dengan baik dan sesuai dengan logika keluaran FPGA.

### 5.4.2 Peralatan Pengujian

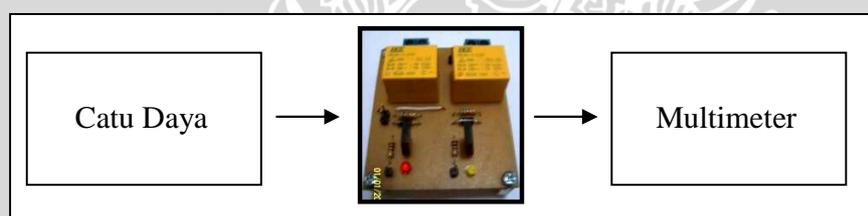
Peralatan pengujian yang digunakan antara lain:

- 1) Multimeter digital
- 2) Catu daya +5 V
- 3) Rangkaian driver

### 5.4.3 Prosedur Pengujian

Prosedur pengujian yang digunakan antara lain:

- 1) Merangkai peralatan ditunjukkan Gambar 5.8 keluaran rangkaian terhubung dengan multimeter digital.
- 2) Mengaktifkan catu daya.
- 3) Mencatat kondisi driver.



Gambar 5.8 Blok Diagram Pengujian Driver

Hasil Pengujian dan Analisis *driver relay* ditunjukkan dalam Tabel 5.5

Tabel 5.5 Hasil Pengujian *Driver Relay* (praktek)

<b>Logic</b>	<b>Fan</b>
Low	OFF
High	ON

## 5.5 Pengujian metode Quine-McCluskey pada pengendali kelembaban

### 5.5.1 Tujuan Pengujian

Pengujian metode Quine-McCluskey bertujuan untuk menganalisis apakah setpoint bawah  $\leq 65\%RH$  dan setpoint atas  $\geq 90\%RH$  telah bekerja sesuai yang direncanakan.

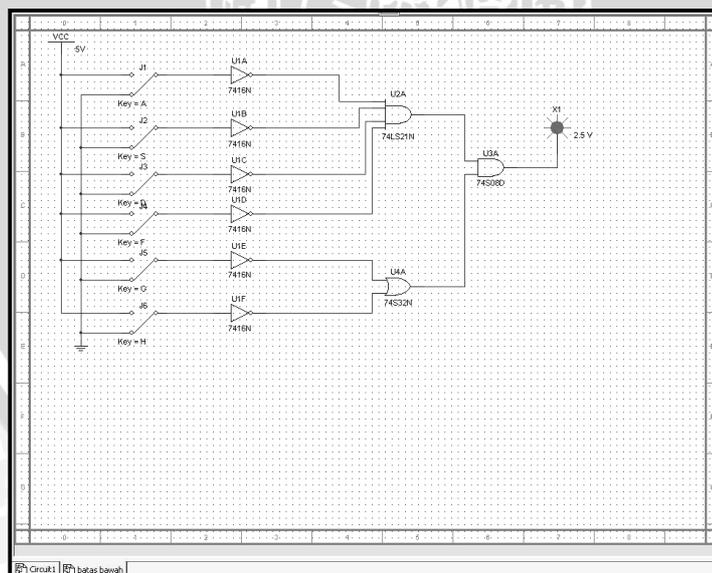
### 5.5.2 Peralatan Pengujian

Peralatan pengujian yang digunakan antara lain:

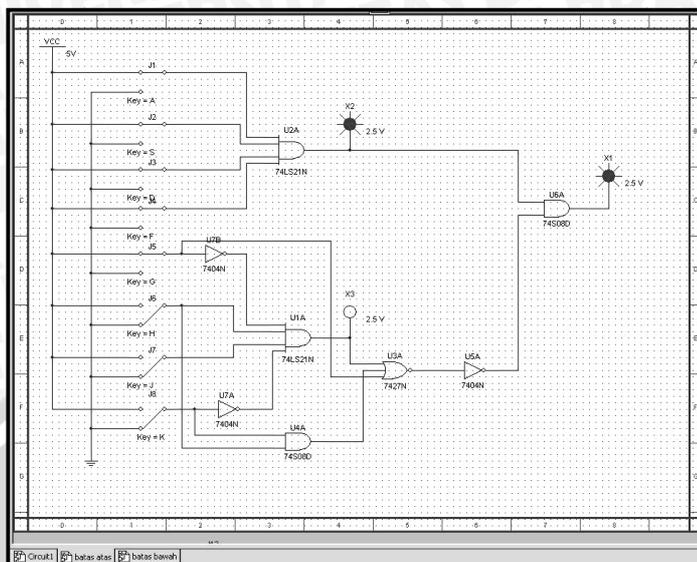
- 1) Hasil Metode Quine-McCluskey
- 2) Multisim Ver.8

### 5.5.3 Prosedur Pengujian dan analisis

- 1) Menentukan batas bawah dan atas yang sebelumnya telah dikalibrasi. Didapat:
  - a) Setpoint bawah  $\leq 65\%RH$  dengan tegangan  $\leq 0,2V$  menghasilkan range Metode Quine-McCluskey antara bit 00H hingga 0BH menghasilkan  $D_7 \cdot D_6 \cdot D_5 \cdot D_4 \cdot (D_3 + D_2)$
  - b) Setpoint atas  $\geq 90\%RH$  dengan tegangan  $\geq 4,8V$  menghasilkan range Metode Quine-McCluskey antara bit F5H hingga FFH menghasilkan  $(D_7 D_6 D_5 D_4) \cdot (D_3 D_2 D_1 D_0 + D_2 D_0 + D_3)$
- 2) Melakukan percobaan dengan software Multisim Versi 8 dengan ditunjukkan Gambar 5.9 dan Gambar 5.10



Gambar 5.9 Pengujian Setpoint bawah Multisim Ver.8



Gambar 5.10 Pengujian Setpoint atas Multisim Ver.8

- 3) Mencatat hasil yang didapat dengan berbagai sample kondisi pada Tabel 5.6 dan Tabel 5.7

Tabel 5.6 Tabel Kebenaran Setpoint Bawah

Input	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	D <sub>7</sub>	D <sub>6</sub>	D <sub>5</sub>	D <sub>4</sub>	A (and)	D <sub>3</sub> '	D <sub>2</sub> '	B (or)	A.B=C
00	0000 0000	1	1	1	1	1	1	1	1	1.1 = 1
01	0000 0001	1	1	1	1	1	1	1	1	1.1 = 1
02	0000 0010	1	1	1	1	1	1	1	1	1.1 = 1
03	0000 0011	1	1	1	1	1	1	1	1	1.1 = 1
04	0000 0100	1	1	1	1	1	1	0	1	1.1 = 1
05	0000 0101	1	1	1	1	1	1	0	1	1.1 = 1
06	0000 0110	1	1	1	1	1	1	0	1	1.1 = 1
07	0000 0111	1	1	1	1	1	1	0	1	1.1 = 1
08	0000 1000	1	1	1	1	1	0	1	1	1.1 = 1
09	0000 1001	1	1	1	1	1	0	1	1	1.1 = 1
0A	0000 1010	1	1	1	1	1	0	1	1	1.1 = 1
0B	0000 1011	1	1	1	1	1	0	1	1	1.1 = 1
0C	0000 1100	1	1	1	1	1	0	0	0	1.0 = 0
0F	0000 1111	1	1	1	1	1	0	0	0	1.0 = 0
10	0001 0000	1	1	1	0	0	1	1	1	0.1 = 0
90	1001 0000	0	1	1	0	0	1	1	1	0.1 = 0
FF	1111 1111	0	0	0	0	0	0	0	0	0.0 = 0

Setpoint bawah  $\leq 65\%RH$  dengan tegangan  $\leq 0,2V$  menghasilkan range Metode Quine-McCluskey antara bit 00H hingga 0BH menjadi logika '1'.

Tabel 5.7 Tabel Kebenaran Setpoint Atas

Input	D <sub>7</sub> D <sub>6</sub> D <sub>5</sub> D <sub>4</sub> D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	A (and)	D <sub>3</sub> D <sub>2</sub> D <sub>1</sub> D <sub>0</sub>	B	D <sub>2</sub> D <sub>0</sub>	C	B+C+ D <sub>3</sub> '= E	A.E = F
F5	1111 0101	1	1100	0	11	1	0+1+0=1	1.1 = 1
F6	1111 0110	1	1111	1	10	0	1+0+0=1	1.1 = 1
F7	1111 0111	1	1110	0	11	1	0+1+0=1	1.1 = 1
F8	1111 1000	1	0001	0	00	0	0+0+1=1	1.1 = 1
F9	1111 1001	1	0000	0	01	0	0+0+1=1	1.1 = 1
FA	1111 1010	1	0011	0	00	0	0+0+1=1	1.1 = 1
FB	1111 1011	1	0010	0	01	0	0+0+1=1	1.1 = 1
FC	1111 1100	1	0101	0	10	0	0+0+1=1	1.1 = 1
FD	1111 1101	1	0100	0	11	1	0+1+1=1	1.1 = 1
FE	1111 1110	1	0111	0	10	0	0+0+1=1	1.1 = 1
FF	1111 1111	1	0110	0	11	1	0+1+1=1	1.1 = 1
F4	1111 0100	1	1101	0	10	0	0+0+0=0	1.0 = 0
F3	1111 0011	1	1010	0	01	0	0+0+0=0	1.0 = 0
F2	1111 0010	1	1011	0	00	0	0+0+0=0	1.0 = 0
F1	1111 0001	1	1000	0	01	0	0+0+0=0	1.0 = 0
F0	1111 0000	1	1001	0	00	0	0+0+0=0	1.0 = 0
00 -> EF	-----	0	-----	-	--	-	-----	0.E=0

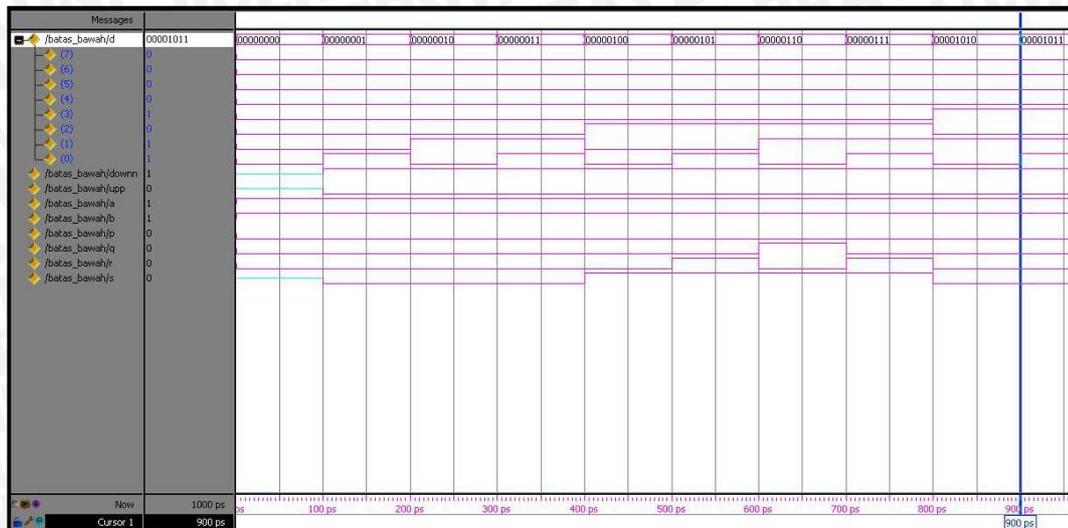
Setpoint atas  $\geq 90\%RH$  dengan tegangan  $\geq 4,8V$  menghasilkan range Metode Quine-McCluskey antara bit F5H hingga FFH menjadi logika '1'.

### 5.6 Pengujian sistem secara keseluruhan (*software*)

Pengujian sistem secara keseluruhan ini bertujuan untuk mengetahui kemampuan sistem yang dirancang dalam pengendalian kelembaban pada nilai set point yang diinginkan, diberikan set point untuk kelembaban dengan setpoint bawah  $\leq 65\%RH$  dan Setpoint atas  $\geq 90\%RH$ . Pengujian ini dilakukan menggunakan *prototype* ruangan.

#### 5.6.1 Pengujian sistem menggunakan simulasi ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition

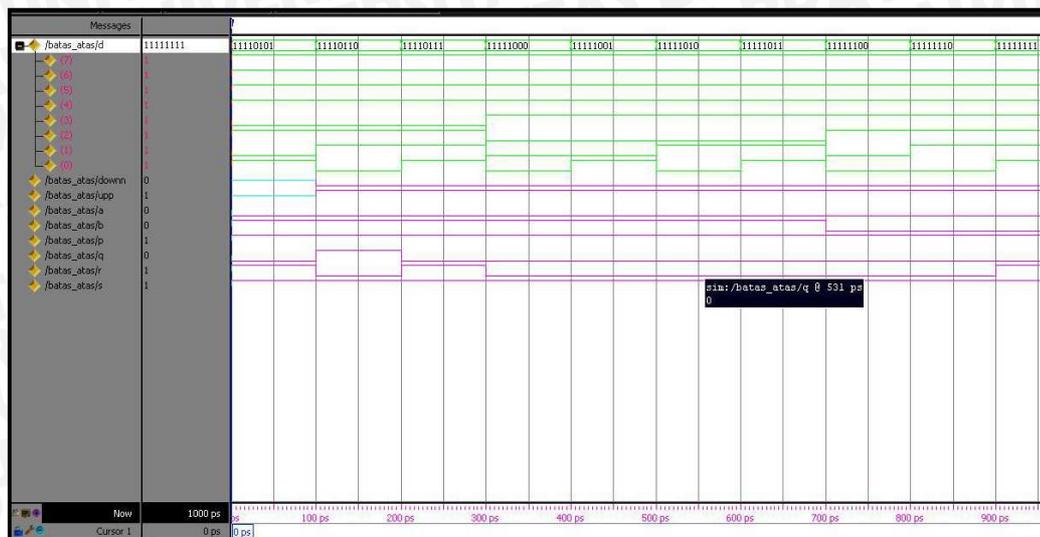
Secara simulasi menggunakan ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition didapatkan *wave* keadaan pusat sistem ketika diberikan kelembaban dengan setpoint bawah  $\leq 65\%RH$  dan Setpoint atas  $\geq 90\%RH$  tanpa memperhitungkan tingkat *error* yang terjadi pada sensor ditunjukkan Gambar 5.11 dan Gambar 5.12



Gambar 5.11 Pengujian wave Setpoint bawah ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition

Keterangan :

- a) D0-D7 : masukan sinyal digital yang berasal dari ADC yang merupakan hasil konverter sinyal analog sensor HSM-20G.
- b) Batas\_bawah down : keluaran untuk batas bawah dengan respon logika '1' ketika kelembaban  $\leq 65\%RH$ .
- c) Batas\_bawah up : keluaran untuk batas atas dengan respon logika '1' ketika kelembaban  $\geq 90\%RH$ .
- d) Batas\_bawah a : not D<sub>7</sub> and not D<sub>6</sub> and not D<sub>5</sub> and not D<sub>4</sub>
- e) Batas\_bawah b : not D<sub>3</sub> or not D<sub>2</sub>
- f) Batas\_bawah p : D<sub>7</sub> and D<sub>6</sub> and D<sub>5</sub> and D<sub>4</sub>
- g) Batas\_bawah q : not D<sub>3</sub> and D<sub>2</sub> and D<sub>1</sub> and not D<sub>0</sub>
- h) Batas\_bawah r : D<sub>2</sub> and D<sub>0</sub>
- i) Batas\_bawah s : P or Q or D<sub>2</sub>
- j) Run Length : 100ps



Gambar 5.12 Pengujian wave Setpoint atas ModelSim-Altera 6.5b (Quartus II 9.1) Starter Edition

Keterangan :

- a) D0-D7 : masukan sinyal digital yang berasal dari ADC yang merupakan hasil konverter sinyal analog sensor HSM-20G.
- b) Batas\_atas down : keluaran untuk batas bawah dengan respon logika '1' ketika kelembaban  $\leq 65\%RH$ .
- c) Batas\_atas up : keluaran untuk batas atas dengan respon logika '1' ketika kelembaban  $\geq 90\%RH$ .
- d) Batas\_atas a : not D<sub>7</sub> and not D<sub>6</sub> and not D<sub>5</sub> and not D<sub>4</sub>
- e) Batas\_atas b : not D<sub>3</sub> or not D<sub>2</sub>
- f) Batas\_atas p : D<sub>7</sub> and D<sub>6</sub> and D<sub>5</sub> and D<sub>4</sub>
- g) Batas\_atas q : not D<sub>3</sub> and D<sub>2</sub> and D<sub>1</sub> and not D<sub>0</sub>
- h) Batas\_atas r : D<sub>2</sub> and D<sub>0</sub>
- i) Batas\_atas s : P or Q or D<sub>2</sub>
- j) Run Length : 100ps



44°C	2,6	1000 0101	1100000	1100000
45°C	2,7	1000 1010	1100000	0100100
46°C	2,8	1000 1111	1100000	1100000
47°C	2,9	1001 0100	1100000	0001111
48°C	3,0	1001 1001	1100000	0000000
49°C	3,1	1001 1110	1100000	0000100
50°C	3,2	1010 0011	0100100	0000001

Keterangan : 7 Segment (a): Puluhan 7 Segment (b): Satuan

Pengujian sistem suhu pada 7-Segment Spartan-3 ditunjukkan Gambar 5.13



Gambar 5.13 Pengujian sistem suhu pada 7-Segment Spartan-3

### 5.7.2 Pengujian kelembaban berdasarkan respon waktu.

Pengujian sistem ini bertujuan untuk mengetahui kemampuan sistem yang dirancang dalam pengendalian kelembaban pada nilai *setpoint*  $\leq 65\%RH$  dan  $\geq 90\%RH$ .

Langkah-langkah yang dilakukan adalah sebagai berikut:

- 1) Menyiapkan inkubator *Miselum J.Tiram* dan memastikan catu daya telah terpasang pada alat.
- 2) Menghidupkan catu daya dan memastikan alat telah bekerja.
- 3) Menentukan nilai *set point* pada sistem, yaitu untuk kelembaban dengan *setpoint* bawah  $\leq 65\%RH$  dan *setpoint* atas  $\geq 90\%RH$ .
- 4) Mengukur nilai suhu pada model miniatur *inkubator* setiap 10 detik.
- 5) Mengamati dan mencatat hasil pengujian.

Pengambilan pengujian dilakukan di bengkel Elka (R1.8). Keadaan *inkubator* dengan kelembaban ruangan sebesar 58%RH (kelembaban rendah) dan 92%RH (kelembaban tinggi), alat melakukan aksi sesuai dengan rules yang diberikan dengan ditunjukkan Tabel 5.9a dan Tabel 5.9b.

Tabel 5.9a Hasil Pengujian untuk Pengukuran Kelembaban rendah

Waktu	RH(%)	Sprayer	Waktu	RH(%)	Sprayer
0	58	on	410 detik	59	on
1menit	58	on	420 detik	59	on
2menit	58	on	430 detik	60	on
3menit	58	on	440 detik	60	on
190 detik	58	on	450 detik	60	on
200 detik	58	on	460 detik	60	on
210 detik	58	on	470 detik	60	on
220 detik	58	on	480 detik	61	on
230 detik	58	on	490 detik	61	on
240 detik	58	on	500 detik	61	on
250 detik	58	on	510 detik	61	on
260 detik	58	on	520 detik	61	on
270 detik	58	on	530 detik	62	on
280 detik	59	on	540 detik	62	on
290 detik	59	on	550 detik	62	on
300 detik	59	on	560 detik	62	on
310 detik	59	on	570 detik	62	on
320 detik	59	on	580 detik	63	on
330 detik	59	on	590 detik	63	on
340 detik	59	on	600 detik	63	on
300 detik	59	on	610 detik	63	on
310 detik	59	on	620 detik	64	on
320 detik	59	on	630 detik	64	on
330 detik	59	on	640 detik	65	on
340 detik	59	on	650 detik	65	on
350 detik	59	on	670 detik	66	off
360 detik	59	on	680 detik	67	off
370 detik	59	on	690 detik	68	off
380 detik	59	on	700 detik	69	off
390 detik	59	on	710 detik	69	off

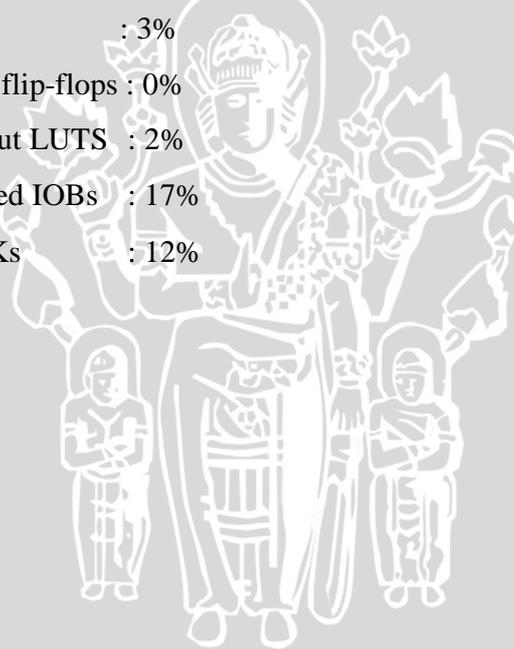
Tabel 5.9b Hasil Pengujian untuk Pengukuran Kelembaban Tinggi

Waktu	RH (%)	Fan
0	92	on
10 detik	92	on
20 detik	92	on
30 detik	91	on
40 detik	91	on
50 detik	91	on
60 detik	90	on
70 detik	90	on
80 detik	89	off

### 5.7.3 Utilization Summary program yang digunakan pada sistem

Kapasitas program desain (*device utilization*) pada xc3s200 yaitu

- Number of slices : 3%
- Number of slices flip-flops : 0%
- Number of 4 Input LUTs : 2%
- Number of bonded IOBs : 17%
- Number of GCLKs : 12%



## BAB VI

### PENUTUP

#### 6.1. Kesimpulan

Dari perancangan dan pembuatan sistem serta dilanjutkan dengan proses pengujian, maka diperoleh kesimpulan sebagai berikut :

- 1) Pendeteksian perubahan kelembaban dan suhu pada perubahan aksi dari *sprayer* dan *fan* kelembaban dalam 3 jangkauan range yaitu  $X \leq 65\%$ ,  $66\% \leq X \leq 89\%$  dan  $X \geq 90\%$  (dalam RH) yang memiliki kesalahan rata-rata sebesar 1,26% serta perubahan digit suhu pada 7-Segment *SPARTAN-3 development board* yang memiliki range monitor  $20^{\circ}\text{C}$  hingga  $50^{\circ}\text{C}$  dengan kesalahan rata-rata 0,95%.
- 2) Penggunaan metode Quine-McCluskey dan *sampling* suhu sehingga didapatkan jumlah alamat eksternal *SPARTAN-3* yang digunakan yaitu 16 untuk masukan dan 2 untuk keluaran dan 7 bit keluaran lainnya sebagai masukan 7-Segment *SPARTAN-3*. Menghasilkan respon perubahan bit  $100\mu\text{S}$  dan pada perangkat lunak menghasilkan 111 instruksi.
- 3) Sistem dapat mengkondisikan kelembaban udara pada jangkauan kelembaban optimal dengan kelembaban udara antara 66%RH sampai 89%RH dengan penggunaan fungsi *setpoint* bawah 65%RH diperoleh tegangan 0,2V dan *setpoint* atas 90%RH diperoleh tegangan 4,8V.

#### 6.2. Saran

Perancangan dan pembuatan alat yang dilakukan pada penelitian ini, masih banyak terdapat kelemahan dan kekurangan yang dimiliki oleh sistem. Oleh karena itu terdapat beberapa hal yang dapat dijadikan saran untuk kesempurnaan alat lebih lanjut, sebagai berikut :

- 1) Untuk mencapai nilai kelembaban udara dalam waktu yang lebih singkat sebaiknya memperhatikan konstruksi inkubator di mana dinding yang digunakan dengan memperhatikan penempatan yang sesuai untuk *fan* dan *sprayer*.
- 2) Penggunaan *sprinkle* sebagai pengganti uap panas pada *sprayer*.

- 3) Dapat dikembangkan dengan penambahan faktor-faktor yang mempengaruhi pertumbuhan *Miselum J.Tiram* itu sendiri. Untuk pengembangan dan penyempurnaan kinerja alat diperlukan riset lebih lanjut.



## DAFTAR PUSTAKA

- Broto, Soekardi, Wisnu. 1983. *Asas-Asas Meteorologi Pertanian*. Ghalia Indonesia. Jakarta.
- Ashenden, P.J. 1990. *The VHDL Cookbook. First Edition*. Adelaide: Dept. Computer Science, University of Adelaide.
- Campobello, Giuseppe dkk. 2003. *Parallel Realization*. Department of Physics University of Messina. Italy.
- Pearson Education. 2004. *Error Detection and Correction*. Pearson Education.
- Spiegel, Jan V. 2001. *Tutorial VHDL*. University of Pennsylvania Departement of Electrical Engineering.
- XESS Corporation. 2001. *FPGA*. XESS Corporation
- Xilinx Inc. 2006. *Progamable Logic Design Quick Start Handbook*. Xilinx Inc.
- Xilinx Inc. 2006. *Synthesis and Simulation Design Guide*. Xilinx Inc.
- Xilinx Inc. 2006. *Synthesis and Verification Design Guide*. Xilinx Inc.
- Xilinx Inc. 2004. *Virtex II FPGA Family : Complete Data Sheet*. Xilinx Inc.
- Xilinx Inc. 2008. *Xilinx Timing Constrain User Guide*. Xilinx Inc.
- Coughlin, R.F. 1992. *Penguat Operasional dan Rangkaian Terpadu Linear*. penerjemah Herman W. S. Erlangga. Jakarta.
- Sutrisno. 1987. *Elektronika Teori dan Penerapannya*. ITB. Bandung.
- Malvino, Albert, Paul. 1996. *Prinsip-prinsip Elektronik*. alih bahasa Hanafi Gunawan. Erlangga. Jakarta.

## DAFTAR LAMPIRAN

### LAMPIRAN A. Foto Alat

- 1) Gambar Box Rangkaian
- 2) Gambar Mekanik Alat

### LAMPIRAN B. Gambar Rangkaian

Gambar Rangkaian Keseluruhan

### LAMPIRAN C. Listing Program

Listing Program Keseluruhan

### LAMPIRAN D. Data Sheet Komponen

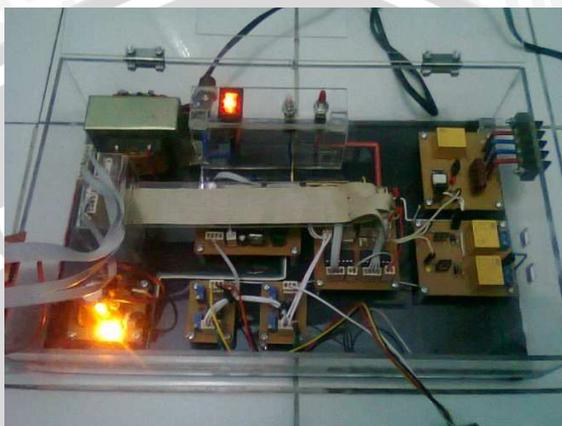
- 1) Data Sheet Sensor Suhu dan Kelembaban HSM-20G
- 2) Data Sheet ADC 0809
- 3) Data Sheet BD 139
- 4) Data Sheet 4N25
- 5) Data Sheet LM 358
- 6) Data Sheet Spartan-3 Starter Kit Board User Guide

**UNIVERSITAS BRAWIJAYA**

**LAMPIRAN A**  
**Foto-Foto Alat**



Gambar Box Rangkaian

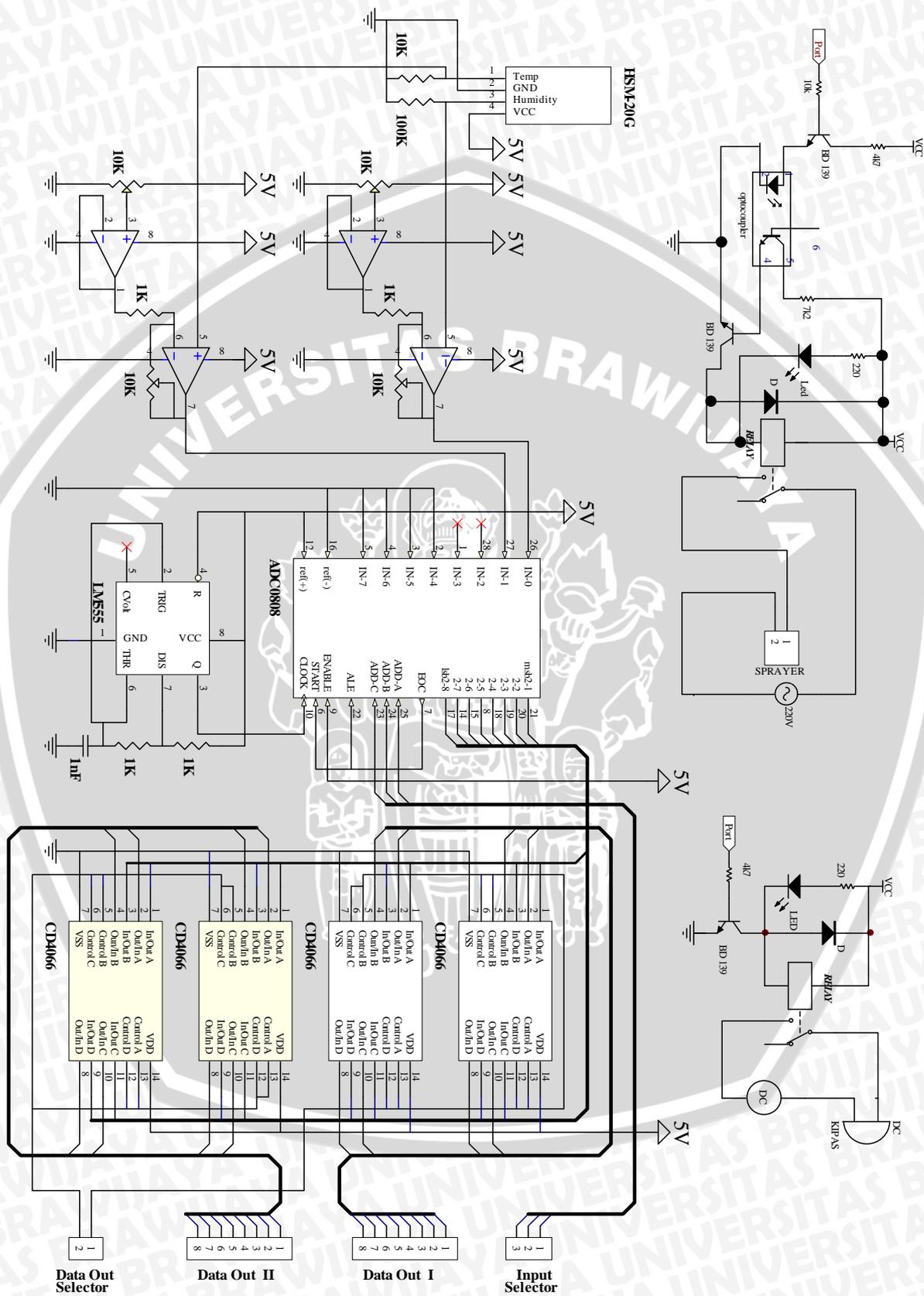


Gambar Mekanik Alat



**LAMPIRAN B**  
**Gambar Skematik Rangkaian Total**





UNIVERSITAS BRAWIJAYA

**LAMPIRAN C**  
**Listing Program (VHDL)**



## LISTING PROGRAM

```

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity batas_bawah is
  Port ( input : in std_logic_vector(15 downto 0);
        sysCLK : in std_logic;
        en0,en1,en2,en3 : out std_logic;
        output : out std_logic_vector(6 downto 0);
        bawahOK,atasOK : out std_logic);
end batas_bawah;

architecture Behavioral of batas_bawah is
  signal output_led3,output_led2 : std_logic_vector(6 downto 0);
  signal InSuhu,InLembab: std_logic_vector (7 downto 0);
  signal state,nextstate : integer range 0 to 1 := 1;
  signal new_CLK : std_logic;
  signal clkdiv : std_logic_vector(23 downto 0):="000000000000000000000000";

begin
  insuhu <= input (7 downto 0);
  inlembab <= input (15 downto 8);

  process (insuhu,inlembab,state) begin

    if inlembab <="00001011" then
      bawahOK <= '1'; atasOK <= '0';
    elsif inlembab>="11110101" then
      bawahOK <= '0'; atasOK <= '1';
    else
      bawahOK <= '0'; atasOK <= '0';
    end if;

    case state is
      when 1 =>
        en3<='0';en2<='1';en1<='1';en0<='1';
        nextstate<=0;
        output<=output_led3;
      when 0 =>
        en3<='1';en2<='0';en1<='1';en0<='1';
        nextstate<=1;
        output<=output_led2;
      end case;
    end process;

```

```

output_led3 <= "1110001" when insuhu < "00001010" --L
           else "0010010" when insuhu >= "00001010" and insuhu < "00111101" --2
           else "0000110" when insuhu >= "00111101" and insuhu < "01110000" --3
           else "1001100" when insuhu >= "01110000" and insuhu < "10100011" --4
           else "0100100" when insuhu = "10100011" --5
           else "1001000"; --H

output_led2 <= "1110001" when insuhu < "00001010" --L
           else "0000001" when (insuhu >= "00001010" and insuhu < "00001111") --0
           or (insuhu >= "00111101" and insuhu < "01000010")
           or (insuhu >= "01110000" and insuhu < "01110101")
           or (insuhu = "10100011")
           else "1001111" when (insuhu >= "00001111" and insuhu < "00010100") --1
           or (insuhu >= "01000010" and insuhu < "01001111")
           or (insuhu >= "01110101" and insuhu < "01111010")
           else "0010010" when (insuhu >= "00010100" and insuhu < "00011001") --2
           or (insuhu >= "01001111" and insuhu < "01001100")
           or (insuhu >= "01111010" and insuhu < "10000000")
           else "0000110" when (insuhu >= "00011001" and insuhu < "00011110") --3
           or (insuhu >= "01001100" and insuhu < "01010001")
           or (insuhu >= "10000000" and insuhu < "10000101")
           else "1001100" when (insuhu >= "00011110" and insuhu < "00100011") --4
           or (insuhu >= "01010001" and insuhu < "01010111")
           or (insuhu >= "10000101" and insuhu < "10001010")
           else "0100100" when (insuhu >= "00100011" and insuhu < "00101000") --5
           or (insuhu >= "01010111" and insuhu < "01011100")
           or (insuhu >= "10001010" and insuhu < "10001111")
           else "0100000" when (insuhu >= "00101000" and insuhu < "00101110") --6
           or (insuhu >= "01011100" and insuhu < "01100001")
           or (insuhu >= "10001111" and insuhu < "10010100")
           else "0001111" when (insuhu >= "00101110" and insuhu < "00110011") --7
           or (insuhu >= "01100001" and insuhu < "01100110")
           or (insuhu >= "10010100" and insuhu < "10011001")
           else "0000000" when (insuhu >= "00110011" and insuhu < "00111000") --8
           or (insuhu >= "01100110" and insuhu < "01101011")
           or (insuhu >= "10011001" and insuhu < "10011110")
           else "0000100" when (insuhu >= "00111000" and insuhu < "00111101") --9
           or (insuhu >= "01101011" and insuhu < "01110000")
           or (insuhu >= "10011110" and insuhu < "10100011")
           else "1001000"; --H

```

```

process (new_CLK)
begin
  if new_CLK='1' and new_CLK'event then
    state<=nextstate;
  end if;
end process;

```

```

process (sysCLK)
begin
  if sysCLK='1' and sysCLK'event then

```



```
clkdiv<=clkdiv+1;
end if;
end process;

new_CLK<=clkdiv(13);
end Behavioral;
```



UNIVERSITAS BRAWIJAYA

**LAMPIRAN D**  
***Datasheet* Komponen**



Filename: Skripsi Silvester E.J (0810632008).docx  
Directory: C:\Users\Dwi Joko Waluyo\Desktop\SILVESTER E.J  
(0810632008)\_ SPARTAN-3 DAN J.TIRAM\3. SKRIPSI\LAPORAN  
Template: C:\Users\Dwi Joko  
Waluyo\AppData\Roaming\Microsoft\Templates\Normal.dotm  
Title:  
Subject:  
Author: Sylvester Eka Jemali  
Keywords:  
Comments:  
Creation Date: 06-Jan-11 11:53:00  
Change Number: 111  
Last Saved On: 07-Jan-11 10:30:00  
Last Saved By: Dwi Joko Waluyo  
Total Editing Time: 398 Minutes  
Last Printed On: 07-Jan-11 10:50:00  
As of Last Complete Printing  
Number of Pages: 91  
Number of Words: 14,590 (approx.)  
Number of Characters: 83,169 (approx.)

