



**UNIVERSITAS INDONESIA**

**PENGEMBANGAN DAN EVALUASI KINERJA SISTEM  
KONTROL BERBASIS *EMBEDDED* WEB**

**SKRIPSI**

Diajukan sebagai salah satu syarat untuk memperoleh  
gelar sarjana teknik

**YUDHI ASMARA**

0706200056

**FAKULTAS TEKNIK ELEKTRO**

**PROGRAM SARJANA EKSTENSI**

**DEPOK**

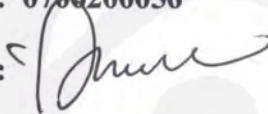
**JUNI 2010**

## HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya saya sendiri,  
Dan semua sumber baik yang dikutip maupun dirujuk  
Telah saya nyatakan dengan benar

Nama : Yudhi Asmara

NPM : 0706200056

Tanda Tangan : 

Tanggal : 30 Juni 2010

## HALAMAN PENGESAHAN


Skripsi ini diajukan oleh:

Nama : Yudhi Asmara  
NPM : 0706200056  
Program Studi : Sarjana Elektro Ekstensi  
Judul Skripsi : Pengembangan dan Evaluasi Kinerja Sistem Kontrol  
Berbasis *Embedded Web*

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro Fakultas Teknik, Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : **Prof. Dr.-Ing. Ir. Kalamullah Ramli, M.Eng** (  ..... )

Penguji : **Prof. Dr. Ir. Bagio Budiardjo** (  ..... )

Penguji : **Prof. Dr. Ir. Riri Fitri Sari, M.Sc., MM.** (  ..... )

Ditetapkan di : Depok

Tanggal : 1 Juli 2010

## UCAPAN TERIMAKASIH

Puji syukur saya panjatkan kepada Tuhan Yang Maha Esa, karena atas berkat dan rahmat-Nya, saya dapat menyelesaikan skripsi ini. Penulisan skripsi ini dilakukan dalam rangka memenuhi salah satu syarat untuk mencapai gerlar Sarjana Teknik Jurusan Teknik Elektro pada Fakultas Teknik Universitas Indonesia. Saya menyadari bahwa, tanpa bantuan dan bimbingan dari berbagai pihak, dari masa perkuliahan sampai pada penyusunan skripsi ini sangatlah sulit bagi saya untuk menyelesaikan skripsi ini. Oleh karena itu, saya mengucapkan terima kasih kepada:

1. Bpk. Prof. Dr. Kalamullah Ramli, selaku dosen pembimbing yang telah menyediakan waktu, tenaga, dan pikiran untuk mengarahkan saya dalam penyusunan skripsi ini;
2. Orang tua dan keluarga saya yang telah memberikan dukungan material dan moral; dan
3. Sahabat yang telah banyak membantu saya dalam menyelesaikan skripsi ini.

Akhir kata, saya berharap Tuhan Yang Maha Esa berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini membawa manfaat bagi pengembangan ilmu. Kritik dan saran yang bersifat membangun dengan senang hati saya tampung melalui [yudhi777@gmail.com](mailto:yudhi777@gmail.com)

Depok, Juni 2010

*Yudhi Asmara*

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS

---

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

Nama : Yudhi Asmara  
NPM : 0706200056  
Program Studi : Sarjana Elektro Ekstensi  
Fakultas : Teknik  
Jenis Karya : Skripsi

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

### **Pengembangan dan Evaluasi Kinerja Sistem Kontrol Berbasis *Embedded Web***

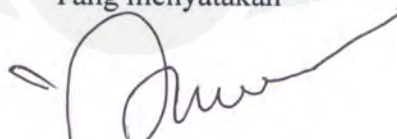
Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/formatkan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya tanpa meminta izin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada Tanggal : 30 Juni 2010

Yang menyatakan



**Yudhi Asmara**

## ABSTRAK

Nama : Yudhi Asmara  
Program Studi : Teknik Elektro  
Judul : Pengembangan dan Evaluasi Kinerja Sistem Kontrol Berbasis *Embedded Web*

Seiring dengan perkembangan teknologi mikroprocessor, kinerja dari embedded system semakin meningkat. Karena kehandalannya, sistem embedded telah diterapkan dalam berbagai bidang seperti peralatan industri, telekomunikasi, alat kesehatan, entertainment, otomotif, perbankan, antariksa, bahkan peralatan rumah tangga dan mainan.

Skripsi ini membahas tentang implementasi Web Server sebagai bagian dari antarmuka dalam sistem kontrol tertanam (*embedded*). Dalam aplikasi tertentu, embedded system merupakan solusi yang tepat, efisien dan hemat energi. *Embedded system* merupakan suatu sistem yang kompak yang dirancang secara terpadu untuk memenuhi kebutuhan yang spesifik. Berbeda dengan komputer pribadi (*PC: Personal Computer*) yang dirancang sangat fleksibel dan multifungsi. Dalam Embedded-System, aplikasi perangkat lunak, komponen elektronik, elektrik dan sistem mekanik, dipadukan secara kompak dalam sebuah perangkat.

Sistem yang dikembangkan adalah *embedded system* yang mengontrol dan memonitor peralatan rumah, yaitu lampu, suhu ruangan dan pintu gerbang. Sistem dirancang menggunakan layanan WEB sebagai antar-muka dengan bahasa pemrograman dynamic-C pada processor Rabbit seri 4000.

Hasil evaluasi atas sistem menunjukkan performa sistem yang cukup baik. Sistem bekerja pada kecepatan 58,98MHz dengan konsumsi daya 1,92 Watt mampu memberikan respon 2,7 detik untuk melayani akses web pada jaringan LAN dan akurasi sensor suhu mencapai 99,1%.

Kata Kunci:  
Sistem Embedded, Web, HTTP Server, Embedded Web

## **ABSTRACT**

Name : Yudhi Asmara  
Study Program : Electrical Engineering  
Title : The Development and Performance Evaluation of  
Embedded Web Control System

Along with the development of micro-processor technology, the performance of embedded systems is enhanced. Because of its ability, embedded systems have been applied in various fields such as industrial equipment, telecommunications, medical equipment, entertainment, automotive, banking, aerospace, and even home appliances and toys.

Work of this skripsi is about implementing a Web Server as part of the interface in embedded control systems. In certain applications, embedded systems provide the right solution for efficient auto control system and energy saving. Embedded systems is a compact system that is designed in an integrated manner to meet specific needs. Unlike a personal computer (PC), which is designed very flexible and multifunctional, Embedded-System, software applications, electronic components, electrical and mechanical systems is integrated in a compact device with limited purposes and less flexible.

This skripsi, discusses the theory and design process of embedded systems applications in controlling and monitoring home devices, i.e. light, room temperature and the gate. System is designed using Web services as an interface with Dynamic-C programming languages on the Rabbit processor 4000 series.

Evaluation result of the system shows a fairly good performance. The system works at clock speed of 58.98 MHz with power consumption of 1.92 Watts. It requires only about 2.7 seconds to respond the web access on the LAN with the accuracy of the temperature sensor 99.1%.

Key Words:

Embedded systems, Web, HTTP Server, Embedded Web

# DAFTAR ISI

## BAB 1 PENDAHULUAN

1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Manfaat .....	2
1.4. Tujuan .....	2
1.5. Batasan Masalah .....	2
1.6. Sistematika Penulisan .....	3

## BAB 2 TEORI DASAR

<b>2.1. Modul Rabbit RCM 4300.....</b>	<b>5</b>
2.1.1. Spesifikasi .....	6
<b>2.2. Relay .....</b>	<b>7</b>
2.2.1. Prinsip Kerja Relay .....	7
<b>2.3. Pulse Widht Modulator .....</b>	<b>8</b>
<b>2.4. Thermistor .....</b>	<b>8</b>
2.4.1. Cara Kerja Thermistor.....	9

## BAB 3 RANCANGAN

<b>3.1. Rancangan Sistem .....</b>	<b>12</b>
3.1.1. Fitur-fitur .....	12
3.1.2. Spesifikasi yang Diharapkan .....	13
3.1.3. Diagram Blok Sistem .....	13
3.1.4. Diagram Blok Embedded Web .....	16
<b>3.2. Rancangan Perangkat Keras .....</b>	<b>19</b>
3.2.1. Rangkaian Utama .....	19
3.2.2. Rangkaian ADC .....	21
3.2.3. Rangkaian Sensor Suhu .....	22
3.2.4. Rangkaian Pengendali Lampu .....	23
3.2.5. Rangkaian Pengendali Motor .....	24
3.2.6. Rangkaian Power Supply .....	25
<b>3.3. Komponen yang Digunakan .....</b>	<b>26</b>
<b>3.4. Rancangan Perangkat Lunak .....</b>	<b>27</b>
3.4.1. Rancangan Antarmuka ( <i>Web Interface</i> ) .....	27
3.4.2. Flow Chart .....	28
<b>3.5. Model Analisa .....</b>	<b>31</b>
3.5.1. Skema Jaringan Uji Coba .....	31
3.5.2. Parameter Evaluasi .....	33



## **BAB 4 IMPLEMENTASI DAN EVALUASI**

<b>4.1. Implementasi</b> .....	<b>34</b>
4.1.1. Hardware .....	34
4.1.2. Software .....	37
<b>4.2. Evaluasi Kinerja</b> .....	<b>48</b>
4.2.1. Peralatan yang Digunakan .....	48
4.2.2. Parameter Evaluasi .....	49
<b>4.3. Hasil Evaluasi</b> .....	<b>52</b>
4.3.1. Evaluasi Uji Fungsi .....	52
4.3.2. Evaluasi Tingkat Akurasi .....	53
4.3.3. Evaluasi Keandalan .....	54

## **BAB 5 KESIMPULAN**

## **DAFTAR REFERENSI**

## DAFTAR TABEL

<b>Tabel 3.1.</b>	Spesifikasi yang Diharapkan .....	13
<b>Tabel 3.2.</b>	Tabel Kebenaran Pengendali Motor H-Bridge .....	25
<b>Tabel 3.3.</b>	Daftar Komponen .....	26
<b>Tabel 4.1.</b>	Design Radio Button .....	38
<b>Tabel 4.2.</b>	Design Tampilan Suhu .....	41
<b>Tabel 4.3.</b>	Hasil Evaluasi Uji Fungsi .....	52
<b>Tabel 4.4.</b>	Hasil Evaluasi Tingkat Akurasi Sensor Suhu .....	53
<b>Tabel 4.5.</b>	Hasil Evaluasi Keandalan .....	54

## DAFTAR GAMBAR

<b>Gambar 2.1.</b> (a) Modul RCM4300 (b) Mikroprocessor Rabbit 4000 .....	5
<b>Gambar 2.2.</b> Arsitektur Modul Rabbit Core 4300 .....	6
<b>Gambar 2.3.</b> Bentuk Fisik Relay .....	7
<b>Gambar 2.4.</b> Thermistor NTC .....	9
<b>Gambar 3.1.</b> Ilustrasi Aplikasi Sistem Kontrol Berbasis Web .....	12
<b>Gambar 3.2.</b> Diagram Blok Implementasi .....	14
<b>Gambar 3.3.</b> Diagram Blok Embedded Web .....	16
<b>Gambar 3.4.</b> Rangkaian Utama .....	19
<b>Gambar 3.5.</b> Rangkaian ADC ADS7870 & Memory .....	21
<b>Gambar 3.6.</b> Rangkaian Sensor Suhu .....	22
<b>Gambar 3.7.</b> Rangkaian Pengendali Lampu .....	23
<b>Gambar 3.8.</b> Rangkaian Pengendali Motor H-Bridge .....	24
<b>Gambar 3.9.</b> Rangkaian Power Supply .....	25
<b>Gambar 3.10.</b> Tampilan Antarmuka Utama .....	27
<b>Gambar 3.11.</b> Tampilan Antarmuka Konfigurasi .....	27
<b>Gambar 3.12.</b> Tampilan Antarmuka bagian Keterangan .....	27
<b>Gambar 3.13.</b> Flow Chart .....	28
<b>Gambar 3.14.</b> Skema Jaringan Uji Coba Model Peer-to-peer .....	31
<b>Gambar 3.15.</b> Skema Jaringan Uji Coba Model LAN .....	31
<b>Gambar 3.16.</b> Jaringan Wireless LAN .....	32
<b>Gambar 3.17.</b> Jaringan Internet (WAN) .....	32
<b>Gambar 4.1.</b> Fisik Modul Rabbit RCM4300 .....	35
<b>Gambar 4.2.</b> Pinout Modul RCM4300 .....	35
<b>Gambar 4.3.</b> USB Downloader .....	36
<b>Gambar 4.4.</b> Konfigurasi Hardware Keseluruhan .....	36
<b>Gambar 4.5.</b> Ilustrasi Kontrol Lampu .....	37
<b>Gambar 4.6.</b> Ilustrasi Kontrol Pintu Gerbang .....	40
<b>Gambar 4.7.</b> Komponen Halaman Utama .....	45
<b>Gambar 4.8.</b> Komponen Halaman Konfigurasi .....	46
<b>Gambar 4.9.</b> Komponen Halaman Informasi .....	47
<b>Gambar 4.10.</b> Implementasi Alat Berbentuk Miniatur Rumah .....	47
<b>Gambar 4.11.</b> Hasil tes kecepatan layanan internet .....	54

# BAB 1

## PENDAHULUAN

### 1.1 Latar Belakang

Komputer pribadi (*PC: Personal Computer*) merupakan komputer yang dirancang sangat fleksibel untuk memenuhi kebutuhan yang sangat bervariasi dan umum dijumpai seperti pengolahan kata, kalkulasi dalam bentuk tabel (*spreadsheet*), pengolahan gambar dan lain-lain. Dilihat dari segi layanan, sebuah PC bisa dijadikan sebagai *server* atau bisa juga sebagai *client* hanya dengan mengganti software yang sesuai.

‘Dunia ada dalam genggaman’, ini adalah ungkapan yang menggambarkan gaya hidup dimana ruang dan waktu sudah tidak lagi menjadi hambatan yang berarti, segala sesuatu bisa dikontrol dari manapun dan kapanpun. Dari mulai transaksi perbankan, pemesanan tiket, meeting dan lain-lain semuanya bisa dilakukan hanya dengan menggunakan *handset*.

Hal ini didukung oleh teknologi Jaringan komputer yang tumbuh sangat pesat, hambatan jarak dan waktu dapat dipecahkan begitu mudah. Bahkan dengan di depan komputer saja, seseorang dapat berkomunikasi dengan orang yang berada di manapun di dunia ini. Kecepatan akses juga semakin tinggi, sehingga mendekati real time.

Timbul keinginan menghubungkan peralatan rumah seperti lampu-lampu dalam rumah, ruang depan dan taman serta pintu gerbang dengan jaringan komputer. Sehingga dapat dipantau kapanpun dan dimanapun, yang kemudian dihubungkan dengan web server sehingga dapat diakses melalui *web browser*. Sekaligus dapat memantau kondisi suhu rumah.

Menggunakan Komputer sebagai alat kontrol sangatlah memungkinkan bahkan sudah lumrah implementasinya, namun besarnya ukuran dan besarnya daya listrik yang dikonsumsi juga perlu dipertimbangkan.

Sistem kontrol *embedded* nampaknya mampu menjawab tantangan tersebut, dengan ukuran yang relatif kecil, konsumsi daya listrik yang juga kecil memberikan solusi yang lebih efisien jika dibandingkan dengan penggunaan komputer sebagai alat kontrol.

Lalu, teknologi apa yang bisa diterapkan pada sistem *embedded* yang memiliki kecepatan dan memori yang sangat terbatas?

## 1.2 Rumusan Masalah

Pada seminar ini akan membahas rancang bangun sistem kontrol menggunakan teknologi *embedded* dengan tampilan web sebagai antarmuka. Aplikasi dari perancangan ini adalah mengontrol peralatan rumah:

1. Mengontrol 3 buah lampu, yang diasumsikan sebagai lampu ruang utama, lampu depan dan lampu taman.
2. Mengontrol pintu gerbang elektrik
3. Memantau suhu didalam rumah.

Untuk memenuhi kebutuhan tersebut rancang bangun akan meliputi:

1. Skema dan *flowchart*
2. Hardware yang digunakan dan rancanganya
3. Kode Program dan bahasa pemrograman yang digunakan
4. Protokol yang digunakan, yaitu TCP/IP
5. Teknologi *Web Server* dan desain

## 1.3 Manfaat

1. Sebagai solusi efisien dalam mengontrol peralatan rumah.
2. Sebagai acuan penerapan teknik kontrol pada aplikasi yang serupa

## 1.4 Tujuan

1. Mempelajari web server dalam aplikasi *embedded*
2. Merancang dan membangun sistem kontrol berbasis *embedded web* sehingga dapat berfungsi dengan baik.

## 1.5 Batasan

Pembahasan akan dibatasi pada implementasi *web server* secara *embedd* pada modul hardware dan pemanfaatannya dalam mengontrol *relay* sebagai aktuator lampu, motor dc sebagai kendali pintu gerbang dan sensor suhu jenis NTC.

## 1.6 Sistematika Penulisan

### BAB 1 Pendahuluan

Bab pendahuluan memuat latar belakang dan tujuan penelitian.

- a) **Latar belakang** berisi perumusan masalah dan faedah yang dapat diharapkan.
- b) **Perumusan masalah** memuat penjelasan mengenai alasan-alasan mengapa masalah yang dikemukakan dalam seminar itu dipandang menarik, penting dan perlu diriset. Kecuali itu juga diuraikan kedudukan masalah yang akan diriset dalam lingkup permasalahan yang lebih luas.
- c) **Faedah yang dapat diharapkan** adalah faedah bagi pengembangan dan penerapan ilmu pengetahuan.
- d) **Tujuan penelitian** menjelaskan secara spesifik hal-hal yang ingin dicapai.
- e) **Batasan masalah** menjelaskan hal-hal/parameter-parameter yang menjadi pembatas dalam riset yang dilakukan.
- f) **Sistematika penulisan** berisi secara sistematis keseluruhan penulisan seminar.

### BAB 2 Teori Penunjang

Teori penunjang memuat penjelasan tentang konsep dan prinsip dasar yang diperlukan untuk memecahkan masalah riset dan untuk merumuskan hipotesis apabila memang diperlukan. Teori penunjang dapat berbentuk uraian kualitatif, model matematis, atau persamaan-persamaan yang langsung berkaitan dengan permasalahan yang akan diriset.

### BAB 3 Rancangan

Menyusun perencanaan dari mulai spesifikasi yang dibutuhkan, pemilihan komponen, desain antar muka, diagram blok, *flowchart*, dan rangkaian elektronik.

## **BAB 4 Implementasi dan Evaluasi**

Pada bab ini merupakan penjabaran dari proses implementasi dari awal hingga sistem yang dibuat bisa digunakan dan memberikan hasil evaluasi mengenai unjuk kerja dan kehandalan dari sistem tersebut.

## **BAB 5 Kesimpulan**

Kesimpulan merupakan pernyataan singkat dan tepat yang dijabarkan dari hasil studi literatur atau landasan teori dari penyusunan skripsi.

## BAB 2

# TEORI PENUNJANG

Sistem kontrol berbasis *embedded web* merupakan integrasi dari beberapa teknologi. Diantaranya adalah teknologi jaringan menggunakan protokol *TCP/IP*, *HTTP*, *Microprocessor*, komponen elektrik dan elektronika lainnya.

### 2.1 Modul *Rabbit RCM4300*

*Rabbit Core Module 4300* (RCM4300, lihat **Gambar 2.1.a**), adalah salah satu generasi modul yang menggunakan Mikroprosesor terbaru dari *Rabbit*, yaitu menggunakan mikroprosesor *Rabbit 4000* (lihat **Gambar 2.1.b**) dengan fitur seperti *DMA*, kecepatan clock sampai dengan 60 MHz, Jalur *I/O* yang digunakan bersama maksimum enam serial port dengan empat level fungsi pin ganda termasuk didalamnya *PWM*, *I/O ekstra*, *kuadratur decoder*, dan *input capture*. Digabungkan dengan lebih dari 500 instruksi opcode baru yang membantu mengurangi ukuran kode dan meningkatkan kecepatan pemrosesan, ini setara dengan modul yang cepat, efisien, dan merupakan solusi ideal untuk berbagai aplikasi *embedded*. Modul inilah yang akan digunakan sebagai host dari aplikasi *embedded web*, yang kemudian akan digabungkan dengan rangkaian pengendali & pengkondisi sinyal lainnya dalam melakukan proses kontrol.

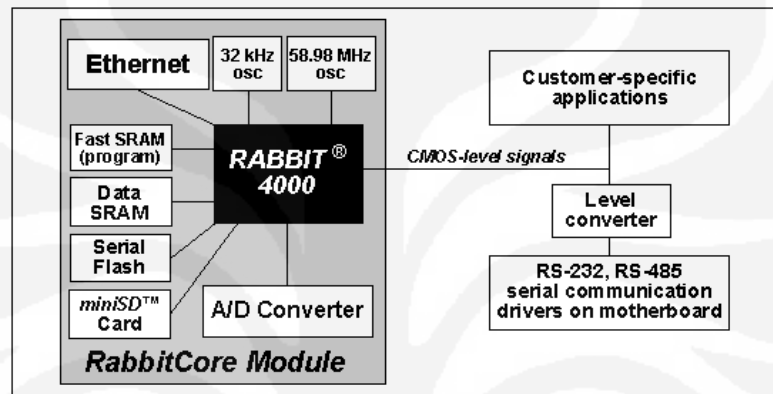


Gambar 2.1. (a) Modul RCM4300 (b) Mikroprosesor Rabbit 4000 <sup>[7]</sup>

Seperti yang terlihat pada **Gambar 2.2**, arsitektur modul RCM4300 juga memiliki fitur yang telah terintegrasi seperti port *Ethernet 10/100Base-T*, *A/D*



converter, dan kartu memori *MMC hot swap*. RCM4300 menggunakan mikroprosesor *Rabbit4000* yang beroperasi hingga 58,98MHz, program SRAM, data SRAM, memori flash serial, sebuah *A/D converter 8-channel*, dilengkapi dengan dua *clock* (osilator utama dan osilator penjaga ketepatan waktu), dan sirkuit yang diperlukan untuk melakukan reset dan pengelolaan baterai cadangan *internal RTC* dan *RAM* statis sebesar 512K.



**Gambar 2.2.** Arsitektur Modul Rabbit Core 4300 <sup>[4]</sup>

RCM4300 juga telah dilengkapi dengan suply tegangan 3,3V dan dapat digabungkan dengan semua jenis perangkat digital CMOS yang kompatibel melalui *motherboard*.

### 2.1.1 Spesifikasi RCM4300

Ukuran : 1.84" × 2.85" × 0.84" (47 mm × 72 mm × 21 mm)

*Microprocessor* : Rabbit 4000 kecepatan maksimum 58.98 MHz

I/O : 36 I/O dengan 4 fungsi alternatif

Tegangan : 3.3 V

Serial Port : 6 port *CMOS-compatible*

ADC : 12-bit *analog inputs*

Memori : 2 MB *flash memory*, 1 MB *fast program-execution SRAM*, Dan 512K *data SRAM*

Media Simpan : *miniSD™ Card memory*

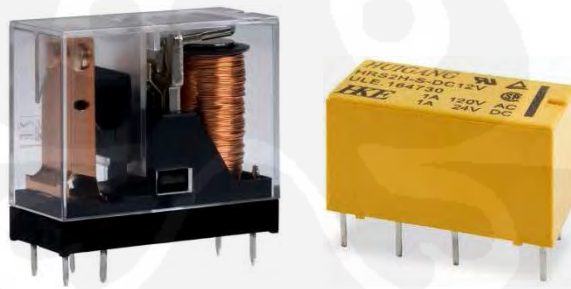
## 2.2 Relay

*Relay* adalah sebuah saklar elektrik yang terbuka dan tertutupnya dikendalikan oleh rangkain listrik lain. Pada bentuk aslinya (lihat **Gambar 2.3**), relay terdiri dari *Coil* (lilitan) dan kontaktor. Saklar digerakan oleh sebuah elektromagnet untuk dapat membuka dan menutup kontaktor. Relay pertama kali ditemukan oleh *Joseph Henry* pada tahun 1835. Keunggulan dari relay adalah:

- Mampu mengendalikan daya yang lebih besar dari pengendali relay tersebut
- Antara pengendali dengan output relay terisolasi dengan baik.

Pada umumnya, relay memiliki 3 tipe kontaktor, yaitu:

- *Normally Open (NO)*. Adalah posisi kontaktor dimana pada saat relay tidak aktif, kontaktornya dalam keadaan terbuka. Dan pada saat relay aktif, posisi kontaktornya berubah menjadi tertutup.
- *Normally Close (NC)*. Adalah posisi kontaktor dimana pada saat relay tidak aktif, kontaktornya dalam posisi tertutup. Dan pada saat relay aktif, kontaktornya berubah posisi menjadi terbuka.
- *Change Over (CO)*. adalah kombinasi antara *Normally Open (NO)* dengan *Normally Close (NC)* dengan satu (1) terminal *Common*.



**Gambar 2.3.** Bentuk Fisik Relay <sup>[10]</sup>

### 2.2.1 Prinsip kerja Relay

Ketika sebuah arus mengalir melalui lilitan/kumparan, inti kumparan akan menjadi magnet karena pengaruh dari gaya gerak magnet yang terjadi pada kumparan, sehingga kontaktor yang terbuat dari bahan logam magnetik akan bergerak dan membuat kontaktor tersebut terhubung dengan kontaktor lain.

Ketika arus hilang dari kumparan, gaya gerak magnet juga hilang dan posisi kontaktor kembali pada posisi semula. Posisi kontaktor diatur sedemikian rupa sehingga medan magnet dapat membuat posisinya menjadi *Normally Close* dan *Normally Open*.

Jika kumparan relay dioperasikan menggunakan arus DC, maka dibutuhkan sebuah diode yang dipasang parallel dengan kumparan tersebut dengan posisi reverse terhadap arus DC, ini dibutuhkan untuk menghilangkan arus balik yang sangat tinggi (*spike*) saat relay berubah status dari aktif menjadi tidak aktif.

### **2.3 Pulse Width Modulator**

*Pulse Width Modulator (PWM)* adalah suatu jenis pengendali yang digunakan untuk mengontrol kecepatan motor. Pada aplikasi sederhana untuk mengendalikan motor DC, *PWM* bisa digunakan dengan memanfaatkan beberapa transistor. Transistor ini diaktifkan dan dimatikan dengan cepat dalam teknik yang dikenal sebagai *Pulse Width Modulation (PWM)*.

Suatu transistor bertindak sebagai gerbang untuk mengizinkan sejumlah arus yang tepat untuk mengalir ke motor. Transistor ini akan dengan cepat dinyalakan dan dimatikan, sehingga jumlah arus yang mengalir tergantung pada rasio antara waktu non-aktif dan waktu aktif. Semakin besar rasio ini, semakin banyak arus akan mengalir. Semakin rendah rasio, semakin sedikit arus akan mengalir.

*PWM* sering digunakan untuk mengendalikan motor AC, tapi sistem ini juga digunakan dalam banyak aplikasi termasuk mengendalikan motor DC.

### **2.4 THERMISTOR**

*Thermistor* adalah suatu jenis resistor, dimana nilai resistansinya bergantung pada temperatur. *Thermistor* secara luas digunakan sebagai pembatas arus, sensor suhu, pelindung kelebihan arus, dan system pemanas. *Thermistor* berbeda dengan *RTD (Resistance Temperature Detector)*. Pada umumnya *thermistor* terbuat dari bahan keramik atau polimer, sedangkan *RTD* menggunakan logam murni. Tanggapan Suhnya juga berbeda, *RTD* dirancang untuk temperatur yang lebih

besar, sementara termistor dirancang untuk mencapai ketepatan yang lebih tinggi dalam kisaran suhu yang terbatas, biasanya  $-90^{\circ}\text{C}$  sampai  $130^{\circ}\text{C}$ .

### 2.4.1 Cara kerja thermistor

Dengan asumsi, sebagai sebuah pendekatan orde pertama, bahwa hubungan antara resistensi dan temperatur adalah *linier*, maka:

$$\Delta R = k\Delta T \quad (2.1)$$

(Wikipedia.org, 2010)

Dimana:

$\Delta R$  = Perubahan resistansi

$\Delta T$  = perubahan temperatur

$k$  = koefisien temperature dari restansi



**Gambar 2.4.** Thermistor NTC <sup>[11]</sup>

Bentuk fisik dari Thermistor seperti yang terlihat pada **Gambar 2.4** di atas. Termistor dapat diklasifikasikan ke dalam dua jenis, tergantung pada tanda  $k$ . Jika  $k$  adalah positif, artinya hambatan akan meningkat seiring meningkatnya suhu, dan jenis ini disebut *Thermistor Positif Temperatur Coefisien (PTC)*, atau posistor. Jika  $k$  adalah negatif, nilai hambatan akan berkurang seiring meningkatnya temperatur, dan jenis ini disebut *Thermistor Negatif Temperature Coefisien (NTC)*. Perbedaan antara resistor dan Thermistor adalah dimana resistor dirancang untuk memiliki nilai  $k$  sedekat mungkin dengan nol, sehingga hambatan

resistor akan bernilai tetap dan hampir konstan selama rentang temperatur yang luas.

*Termistor NTC* memiliki persamaan sebagai berikut:

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} \ln \left( \frac{R}{R_0} \right) \quad (2.2)$$

(Wikipedia.org, 2010)

dimana suhu dalam kelvin dan  $R_0$  adalah resistansi pada suhu  $T_0$  (biasanya  $25^\circ\text{C}=298,15\text{ K}$ ).

Untuk mencari nilai  $R$ :

$$R = R_0 e^{B(1/T - 1/T_0)} \quad (2.3)$$

$$R = r_\infty e^{B/T} \quad (2.4)$$

dimana  $r_\infty = R_0 e^{-B/T_0}$  (2.5)

$$T = \frac{B}{\ln(R/r_\infty)} \quad (2.6)$$

$$\ln R = B/T + \ln r_\infty. \quad (2.7)$$

(Wikipedia.org, 2010)

Ini dapat digunakan untuk mengubah fungsi temperatur vs resistansi thermistor menjadi sebuah fungsi linear  $\ln R$  vs  $1/T$ . kemiringan rata-rata fungsi ini akan menghasilkan estimasi dari nilai parameter  $B$ . Kebanyakan thermistor PTC adalah jenis "*switching*", dimana hambatannya akan tiba-tiba meningkat pada nilai temperatur kritis tertentu. *PTC* terbuat dari keramik polikristalin mengandung barium titanate ( $\text{BaTiO}_3$ ) dan senyawa lainnya. Konstanta dielektrik bahan ferroelectric ini bervariasi dengan suhu.

Tipe lain dari termistor PTC adalah polimer, yang biasanya dijual dengan nama seperti "Polyswitch" "Semifuse", dan "Multifuse". Ini terdiri dari sepotong plastik dengan butir karbon yang tertanam di dalamnya. Ketika plastik dingin, semua butir karbon kontak satu sama lain, membentuk jalur konduktif. Ketika plastik memanas, akan mengembang, memaksa butir karbon terpisah, dan menyebabkan hambatan dari perangkat untuk meningkat dengan cepat.

Seperti thermistor  $BaTiO_3$ , perangkat ini memiliki tanggapan resistansi yang sangat nonlinear terhadap suhu, tidak dianjurkan untuk pengukuran temperatur proporsional. Namun jenis lain dari termistor adalah silistor, sebuah silikon yang sensitif terhadap suhu. Silistor dibuat dan beroperasi pada prinsip yang sama seperti termistor lain, tetapi menggunakan silikon sebagai bahan dasar komponen.

## BAB 3

### PERANCANGAN

Dalam bab III ini, akan membahas mengenai kegiatan riset dalam rangka perancangan Sistem Kontrol berbasis *Embedded Web* dari mulai perancangan perangkat keras hingga desain *software*.

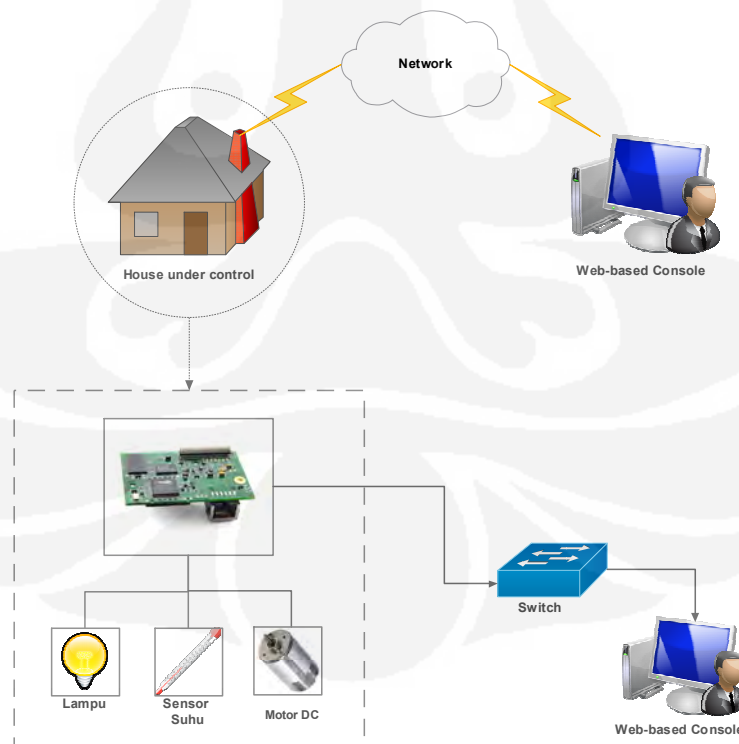
#### 3.1. RANCANGAN SISTEM

##### 3.1.1. Fitur-Fitur

Rancangan dari sistem yang akan dibuat seperti apa yang terlihat pada **Gambar 3.1**. Dimana alat yang akan dibuat digunakan untuk mengontrol tiga (3) komponen, yaitu:

1. lampu sejumlah 3 (tiga) buah,
2. buka-tutup pintu gerbang, dan
3. memonitor suhu ruangan

Ketiga komponen tersebut harus bisa dikontrol secara jarak jauh.



**Gambar 3.1.** Ilustrasi Aplikasi Sistem Kontrol Berbasis Web

Oleh karena itu, tulisan berikutnya akan membahas lebih spesifik tentang bagaimana, teknologi dan teknik yang digunakan untuk memenuhi kebutuhan tersebut.

### 3.1.2. Spesifikasi Yang Diharapkan

**Tabel 3.1.** Spesifikasi yang diharapkan

Spesifikasi	Keterangan
Kontrol Lampu	3 Buah Lampu ON/OFF
Sensor Suhu	0 – 100 °C
Penggerak Pintu Gerbang	Motor DC 5V dengan PWM maks. 2A
Antar muka	Static Web Based, HTML
Sarana Konfigurasi	Static IP-Address melalui Web
<b>Konsumsi Listrik</b>	
Tegangan Kerja	220VAC
Konsumsi Daya. Diluar lampu dan motor	< 2 watt.

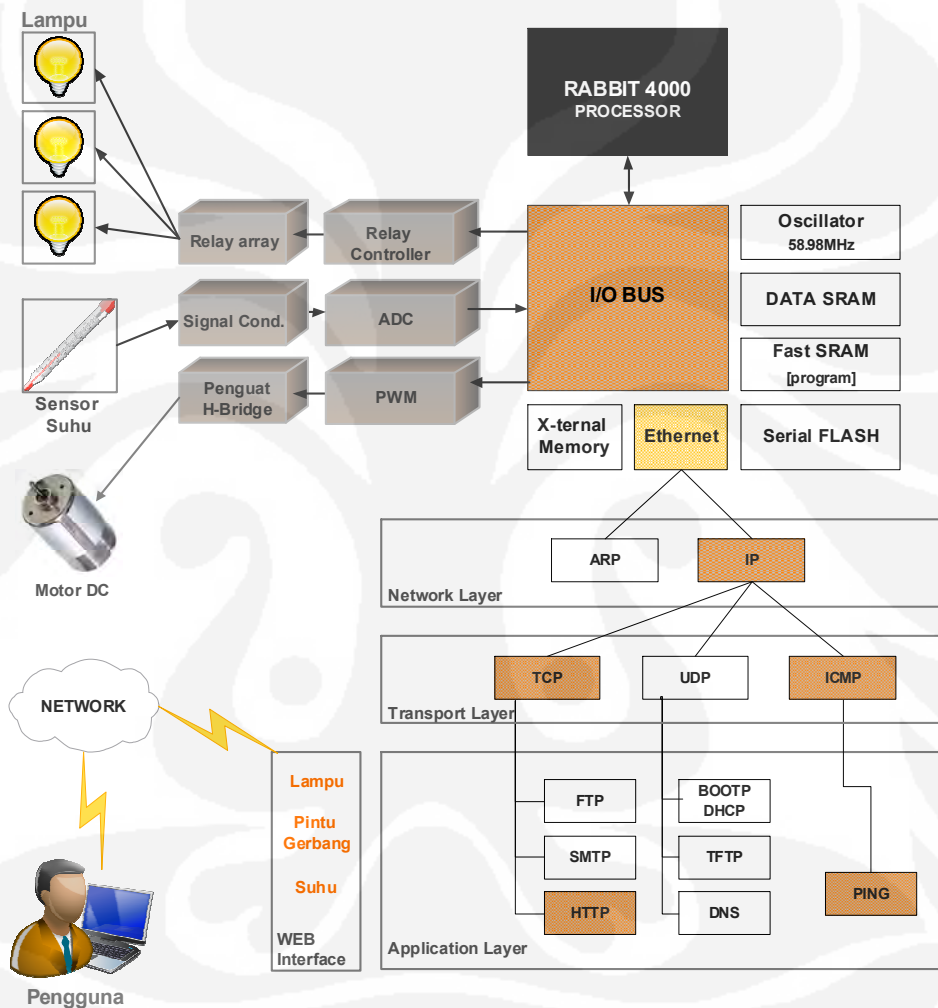
### 3.1.3. Diagram Blok Sistem

Dalam perancangannya, **Sistem Kontrol Berbasis *Embedded Web*** dibagi menjadi beberapa blok (lihat **Gambar 3.2.** Diagram Blok Implementasi) yang bertujuan untuk mengendalikan 3 (tiga) buah lampu pijar, buka/tutup pintu gerbang dan memonitor suhu ruangan. Tiga buah lampu tersebut direpresentasikan sebagai sebuah lampu di ruangan utama, sebuah lampu di ruang depan dan lampu taman, sedangkan untuk buka/tutup pintu gerbang digunakan sebuah motor DC yang akan dikendalikan dengan **PWM** yang diperkuat oleh **penguat arus** maksimal 2 Ampere sehingga kecepatan buka/tutup bisa diatur. Suhu di dalam ruangan akan secara terus menerus dipantau menggunakan **thermistor** sebagai sensornya.

Untuk mengendalikan on/off lampu digunakan **relay** sebagai aktuator yang kemudian relay tersebut dihubungkan dengan **I/O Bus Microprocessor** melalui **Relay Controller**. Disisi lain, suhu ruangan di sensing menggunakan thermistor, nilai analog dari thermistor ini disesuaikan menggunakan rangkaian **Signal Conditioning** yang kemudian dilakukan **sampling** oleh ADC 12-bit agar hasil pembacaan bisa diumpankan pada **I/O Bus Microprocessor**. Pada bagian motor



DC yang digunakan sebagai aktuator untuk membuka/tutup pintu gerbang, digunakan *Penguat H-Bridge* untuk mengendalikan arah putaran dan arus yang dibutuhkan, sedangkan untuk kecepatan putarannya sendiri dikendalikan oleh *PWM (Pulse Width Modulator)* dimana *PWM* tersebut sudah tersedia pada internal microprocessor Rabbit 4000.



**Gambar 3.2.** Diagram Blok Implementasi

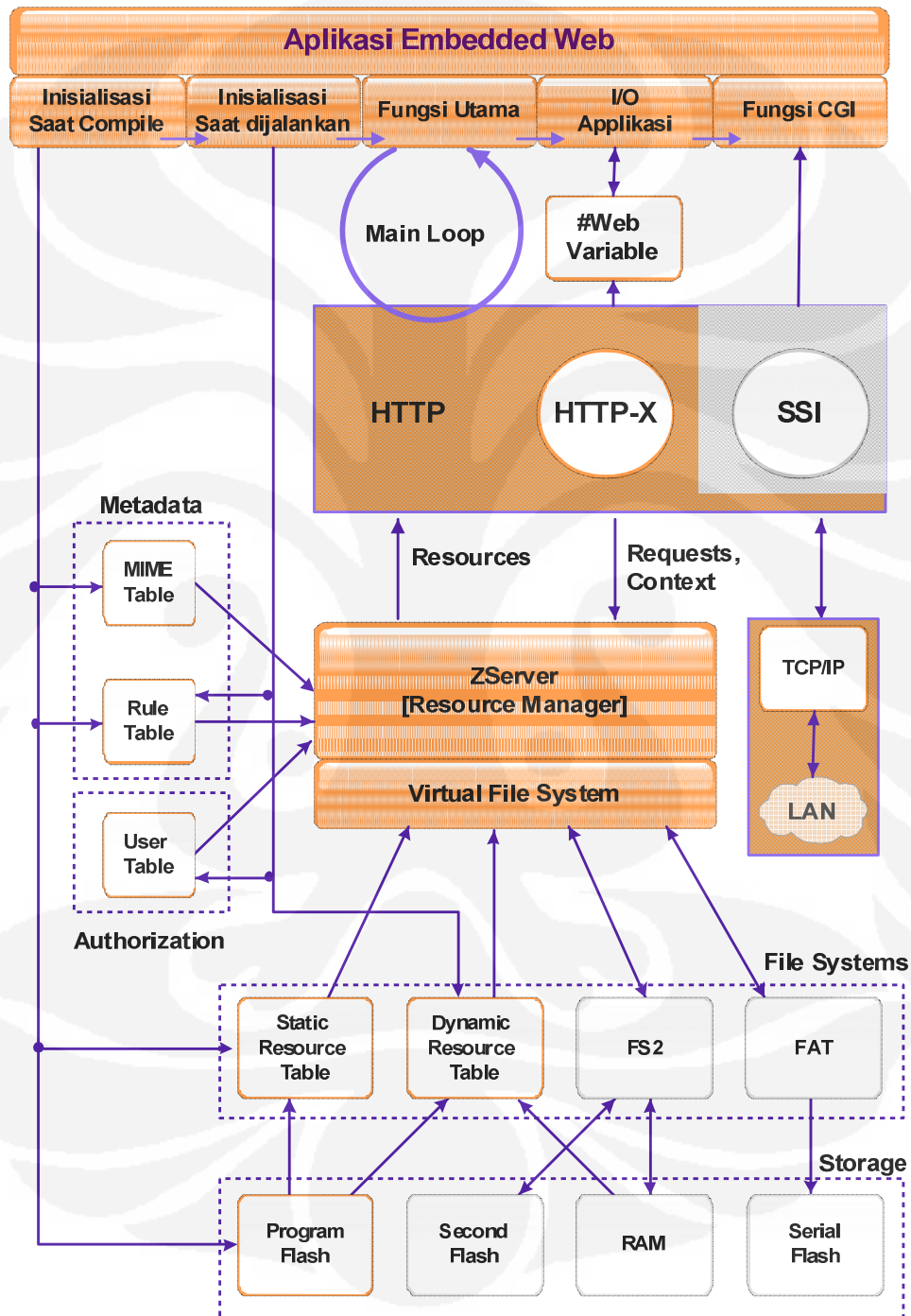
Terlihat di **Gambar 3.2.** di atas, lampu yang merupakan lampu pijar dengan tegangan kerja 220VAC dan maksimum beban 110Watt akan dikontrol menggunakan *Relay Array* sedangkan *Relay Controller* merupakan konverter level tegangan 3,3V ke 5V DC dan transistor sebagai saklar berfungsi sebagai

penghubung antara *Relay Array* dengan mikroprocessor. Sebagai sensor suhu digunakan *Thermistor NTC* yang dikondisikan menggunakan *Signal Conditioning* kemudian diumpankan ke *ADC* untuk dikonversi dari tegangan analog menjadi data digital agar bisa dibaca oleh mikroprocessor. Motor DC Digunakan sebagai aktuator buka/tutup pintu gerbang, untuk mengendalikan motor ini digunakan penguat H-Bridge sebagai sumber arus bagi motor sekaligus mengendalikan arah putaran motor. *Pulse Width Modulator (PWM)* digunakan untuk mengendalikan kecepatan motor DC tersebut.

Masing-masing komponen tersebut saling berkomunikasi dengan mikroprocessor melalui *I/O Bus* yang merupakan 36 jalur *Input/Output* yang dimiliki mikroprocessor Rabbit 4000. Disediakan juga *Ethernet* sebagai antarmuka ke jaringan komputer, sementara disisi software dibangun sebuah *Network Layer* sebagai lapisan *virtual* pada software protokol yang mengendalikan koneksi antara sumber dan tujuan pada jaringan, dalam hal ini direpresentasikan sebagai *IP address*. *Transport Layer* akan mengendalikan aliran paket data antara keduanya. Pada sistem ini digunakan *Transmission Control Protocol (TCP)* dan *Internet Control Message Protocol (ICMP)* sebagai protokol utama yang akan mengatur aliran data. *Aplication Layer* memberikan antarmuka untuk aplikasi *embedded* itu sendiri, dimana pada sistem ini digunakan *HyperText Transfer Protocol (HTTP)* dimana halaman WEB akan diletakkan. Dari *Web interface* inilah pengguna melakukan aktifitas kontrol dan monitoring.

### 3.1.4. Diagram Blok *Embedded Web*

Gambar 3.3. di bawah ini merupakan rancangan utama dari sistem yang akan dibangun. Memberikan gambaran mengenai hubungan antara halaman web dengan elemen kontrol lainnya, yaitu sensor dan aktuator.



Gambar 3.3. Diagram Blok Embedded Web

Dari **Gambar 3.3**. Diagram Blok Embedded Web yang ada pada halaman sebelumnya, terdapat Inisialisasi Saat *Compile* yang merupakan tahap paling awal ketika program yang telah dibuat akan di *embedd* kedalam chip, mencakup *library*, inisialisasi konstanta, struktur data dan tabel. Konfigurasi standar untuk jaringan dan *upload* file statis (file eksternal) melalui perintah *#ximport* atau *#zimport*. Sedangkan Inisialisasi Saat Dijalankan adalah serangkaian proses yang dikerjakan pertama kali pada saat aplikasi dijalankan yaitu inisialisasi *HTTP Server* dan inisialisasi *TCP/IP*. Barulah setelah itu Fungsi Utama akan mengambil alih seluruh kegiatan dari sistem.

I/O Aplikasi Merupakan antar-muka antara halaman web dengan sensor dan aktuator. Melalui Fungsi *Common Gateway Interface* atau disingkat CGI adalah suatu standar untuk menghubungkan berbagai program aplikasi ke halaman web. Walau demikian pada sistem ini tidak digunakan CGI melainkan *HTTP-X* yang merupakan bagian dari *HTTP* dimana pada bagian ini akan memberikan fasilitas pemrograman khusus yang bisa dimengerti oleh mikroprosesor rabbit 4000 yang akan mempermudah dalam mendesain model kontrol berbasis WEB.

Pada *Main Loop*, terdiri dari serangkaian fungsi utama yang akan dijalankan terus menerus. Pada proses ini *Main Loop* akan melibatkan *#Web Variabel* yang merupakan sarana komunikasi antara *WEB Server* dengan *I/O* dimana pertukaran variabel bisa dilakukan. *HTTP* merupakan *Web Server* yang bertanggung jawab untuk menanggapi permintaan dari pengguna. Setiap permintaan akan dianalisa untuk menentukan sumber daya yang diminta, siapa pengguna tersebut, dan apakah pengguna tersebut berwenang untuk mengakses sumber daya yang diminta. Jika sumber daya yang dimaksud memang benar tersedia, pengguna dikenal dan memiliki otoritas yang tepat, maka sumber daya yang dimaksud akan dikirim kembali ke pengguna melalui browser.

*Metadata* Terdiri dari *MIME* dan *Rule Table*. Adalah informasi yang berhubungan dengan setiap individu misalnya hak akses dan hak atas suatu file. *MIME* Merupakan suatu tabel penyesuaian dari file. Yang akan mendefinisikan jenis dari suatu file, apakah file tersebut adalah jenis text (\*.txt) ataupun jenis gambar (\*.jpg). Sedangkan *Rule Table* Adalah tabel penyesuaian antara setiap file

dengan pengguna (*User Table*) yang akan mengatur hak akses dari setiap file. Terdapat juga *User Table* yang akan menyimpan daftar pengguna.

*Static Resource Table* Merupakan tabel *index* yang akan mendefinisikan file-file statis yang diupload pada saat compile, biasanya berupa file logo dimana file tersebut tidak akan berubah-ubah pada aplikasi. Yang didefinikasi adalah lokasi dan ekstensi dari file tersebut *Dinamic Resource Table* Merupakan tabel *index* yang akan mendefinisikan file-file dimana pendefinisian dilakukan ketika program sudah berjalan. Seperti pada aplikasi *FTP* dimana aplikasi *FTP server* sedang berjalan, maka ketika pengguna meng-upload suatu file, file tersebut akan didefinisikan ketika file tersebut berhasil di upload. Yang didefinikasi adalah lokasi dan ekstensi dari file tersebut. *FS2* Salah satu jenis file system. Biasanya digunakan pada sistem unix. *FAT* Salah satu jenis file system. Biasanya digunakan pada sistem windows. *Program Flash* Tempat dimana source code dari program diletakkan atau di *embedd*. *RAM Random Access Memory Serial Flash* Media penyimpanan. Biasa dikenal dengan *Memory Card*.

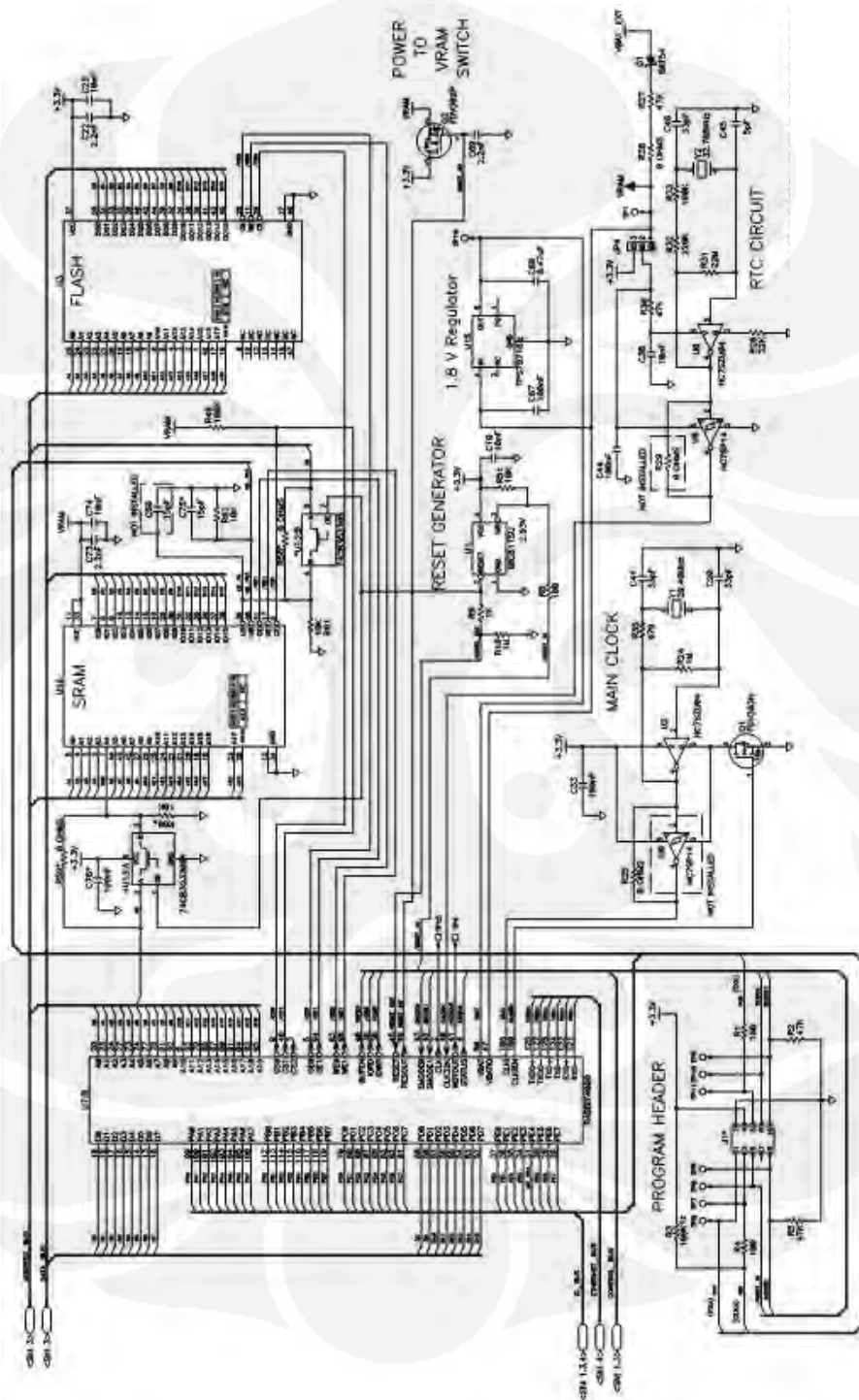
Semua informasi dari setiap tabel tersebut tersebut akan diatur oleh *Zserver*, *resource manager*, *Zserver* akan mengatur lalu lintas data, dimana data seharusnya diletakkan dan type file system yang digunakan. *Virtual File System* merupakan pendekatan yang digunakan dalam management file seperti yang diterapkan pada sistem unix, dimana *'/'* (*slash*) mengindikasikan sebagai *root*, */mnt* sebagai *mount point*.

Bagian terpenting lainnya adalah *TCP/IP* merupakan penghubung ke dunia luar dimana pengguna bisa melakukan akses ke dalam aplikasi melalui browser.

## 3.2. RANCANGAN PERANGKAT KERAS

### 3.2.1. Rangkaian Utama

Gambar 3.4. Berikut adalah rangkaian utama dari modul RCM4300 dimana semua aktifitas dari sistem sebagian besar akan ada disini.



Gambar 3.4. Rangkaian Utama [5]

Rangkaian utama di atas (**Gambar 3.4**) berfungsi sebagai pusat pengolahan sistem dan merupakan bagian dari Modul Rabbit RCM4300, bertanggung jawab untuk mengelola semua permintaan yang berasal dari dunia luar melalui antarmuka web, mengidentifikasi permintaan tersebut dan memberikan hasil keluaran yang sesuai. Rangkaian utama terdiri dari:

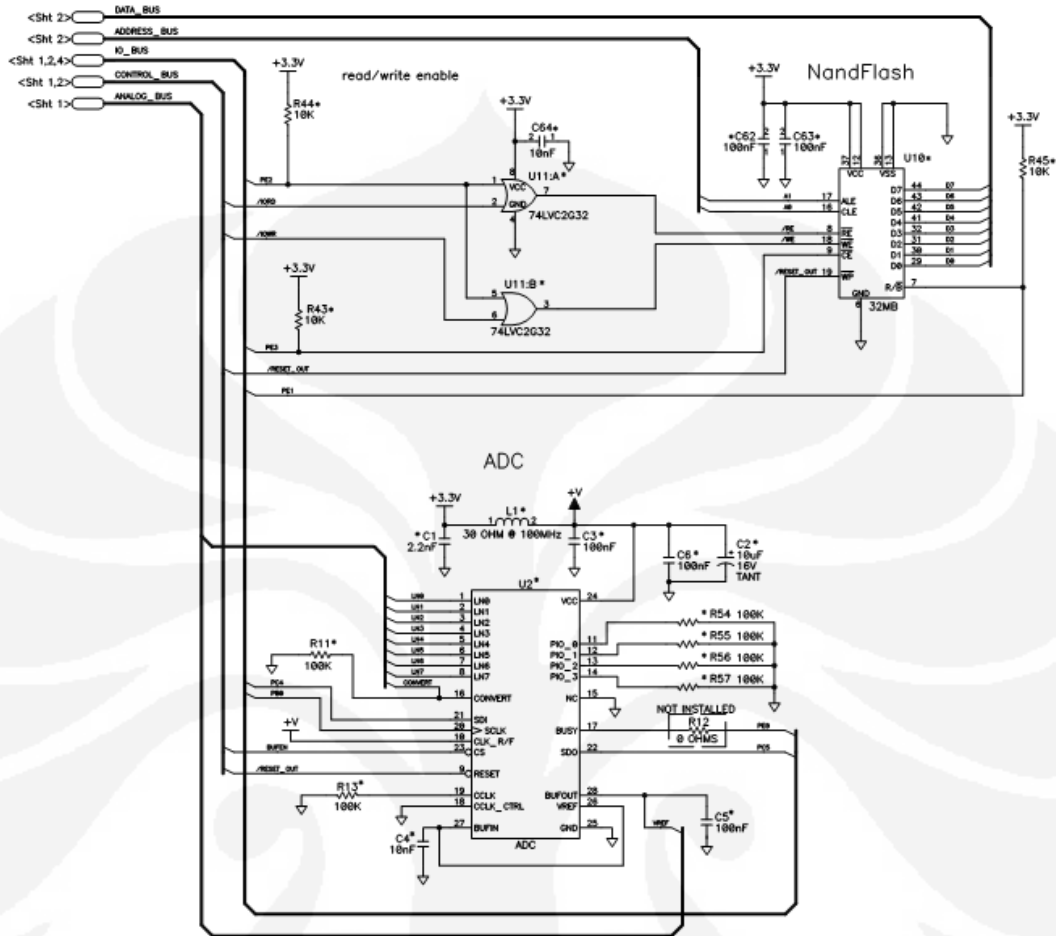
- a. *Microprocessor* Rabbit 4000
- b. *Static RAM*
- c. *Flash Memory*
- d. *Reset Generator*
- e. Sumber Detak
- f. Rangkaian *Real Time Clock* (RTC), dan
- g. Jalur Data *I/O Bus*

Pada saat pertama kali mesin dihidupkan, rangkaian reset generator yang U1 yang berupa IC MIC811SU sesaat setelah power supply mencapai tegangan optimum 3.3V akan memberikan sebuah pulsa reset kepada mikroprosessor Rabbit 4000. Pulsa reset yang dimaksud adalah sebuah pulsa yang bergerak dari logika 1 (high=3,3V) menuju logika 0 (Low=0V). Berawal dari reset inilah serangkaian proses pada mikroprosessor dimulai.

Setelah mendapatkan pulsa reset, pulsa reset tersebut bagi mikroprosessor dianggap sebagai sinyal agar mikroprosessor mengembalikan nilai program counter pada posisi reset 00H dan mengembalikan semua nilai register pada posisi default. Hal ini menyebabkan mikroprosessor membaca program yang terdefiniskan sebagai reset vektor yaitu pada alamat 00H yang ditempatkan pada *FLASH memory*, dan pada alamat ini berisi *source code* untuk eksekusi inisialisasi awal yang akan mode kerja dari setiap *I/O*, penggunaan fitur *ADC* dan *PWM*.

Kemudian, eksekusi program dilanjutkan pada alamat memory selanjutnya yang terletak di Flash memory. Didalam Flash memory inilah *source code* aplikasi **Sistem Kontrol Berbasis *Embedded Web*** akan diletakkan. Sedangkan untuk semua variabel yang bersifat volatile akan disimpan pada SRAM dan nilainya akan hilang ketika tidak ada sumber tegangan.

### 3.2.2. Rangkaian ADC



**Gambar 3.5.** Rangkaian ADC ADS7870 & Memory [5]

Rangkaian ADC pada Gambar 3.5 di atas merupakan bagian dari modul RCM4300 yang bertanggung jawab melakukan konversi tegangan analog menjadi data digital dan memberikan data tersebut pada mikroprocessor. Bagian utama dari rangkaian ADC adalah:

- a. IC ADC ADS7870
- b. Analog Bus
- c. I/O Bus, dan
- d. Control Bus

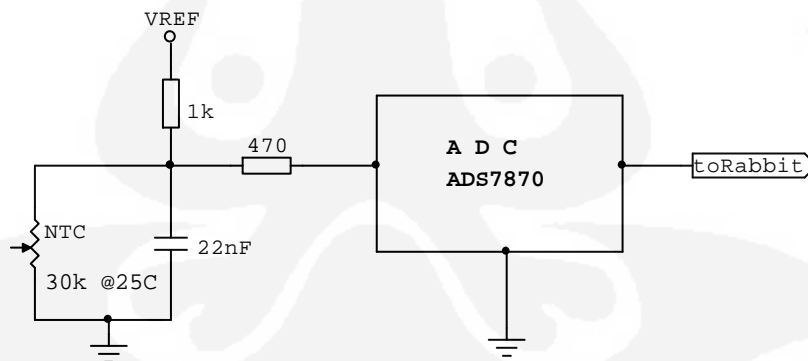
Analog Bus menyediakan sampai dengan 8 kanal sebagai jalur masukan bagi sinyal analog yang akan dikonversi. Untuk menentukan kanal mana yang ingin digunakan, disediakan Control Bus dimana mikroprocessor bisa mengatur kinerja dari setiap kanal dan status dari kesibukan dari ADC.



Hasil dari konversi *ADC* merupakan data digital selebar 12bit yang akan diumpankan ke mikroprocessor melalui jalur *I/O Bus* secara *serial* melalui pin SD0 dan SD1 (**Gambar 3.4**). Yang selanjutnya data digital tersebut diterima oleh mikroprocessor dan disimpan dalam memory untuk selanjutnya dilakukan proses penghitungan yang lebih lanjut. IC *ADC* yang digunakan adalah ADS7870 dimana jenis *ADC* ini mampu mencuplik data sebanyak 52.000 *sampling* setiap detiknya. Bisa bekerja pada tegangan supply dari mulai 2,7Volt sampai dengan maksimal 5,5Volt. Didalam IC *ADC* tersebut sudah dilengkapi dengan *PGA* yaitu *Programmable Gain Ampifier* yang memiliki impedansi input yang tinggi, rendah riak dan bisa diatur tingkat penguatannya sehingga hasil pembacaan menjadi cukup akurat dan tidak lagi memerlukan rangkaian *Signal Conditioning Eksternal* menjadikan penggunaan *ADC* lebih sederhana.

### 3.2.3. Rangkaian Sensor Suhu

Sensor suhu yang digunakan adalah *thermistor* jenis *NTC* dengan hambatan standar  $R_{tstd} = 3K\Omega$  pada suhu ruangan  $25^{\circ}C$  dan nilai hambatan ini akan menurun seiring dengan naiknya suhu yang ada disekitarnya. Rangkaian sensor tersebut seperti yang terlihat pada **Gambar 3.6** berikut ini.



**Gambar 3.6.** Ragkaian Sensor Suhu

Nilai hambatan dari thermistor pada rangkaian diatas (**Gambar 3.6**) akan sebanding dengan suhu ruangan, agar nilai dari *thermistor* ini bisa dibaca oleh *ADC* maka *thermistor* tersebut dirangkai dengan resistor senilai  $1K\Omega$  yang dirangkai secara seri dan diberikan sumber tegangan, dengan demikian nilai dari *thermistor* dapat dibaca dengan satuan tegangan. Dan nilai dari suhu ruangan adalah *equivalent* dengan nilai hambatan thermistor saat itu dan juga *equivalent*

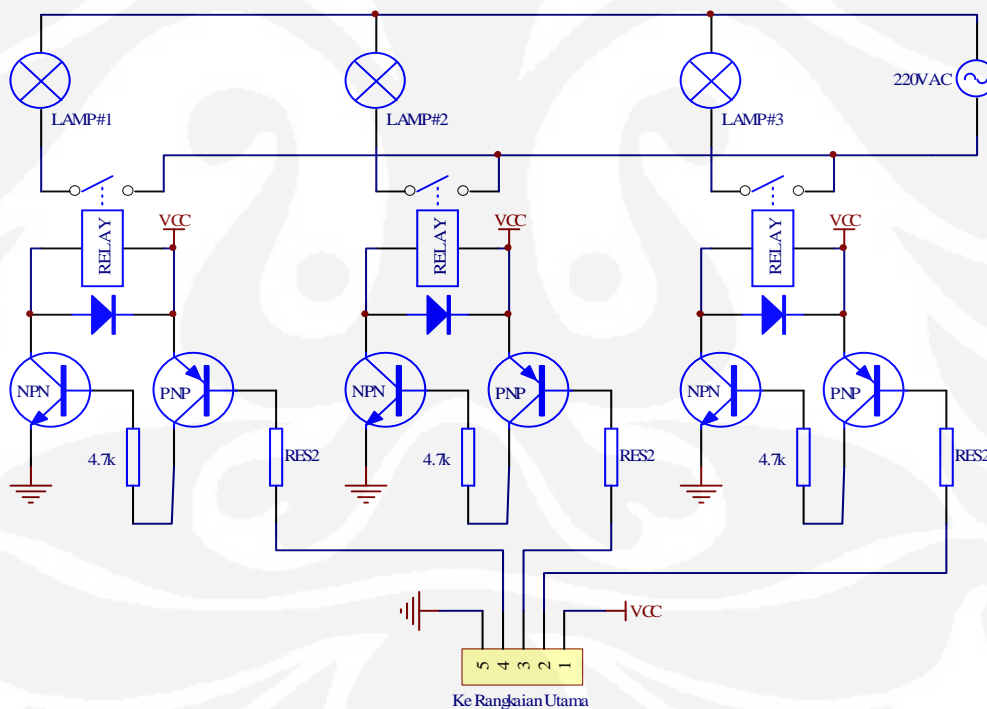
dengan nilai tegangan pada thermistor tersebut. Nilai tegangan dari thermistor bisa didapatkan dengan persamaan Hukum Kirchoff sebagai berikut:

$$V_{NTC} = \frac{R_{NTC}}{1K\Omega + R_{NTC}} \cdot V_{REF} \quad (3.1)$$

Kemudian nilai dari  $R_{NTC}$  tersebut akan diumpangkan ke ADC ADS7870 dimana didalam IC ADC tersebut sudah terdapat rangkaian pengkondisi sinyal berupa *PGA (Programmable Gain Amplifier)* baru kemudian dilakukan konversi menjadi data digital dan diberikan ke mikroprocessor melalui data bus serial.

### 3.2.4. Rangkaian Pengendali Lampu

Rangkaian dibawah ini (**Gambar 3.7**) berfungsi sebagai saklar bagi 3 buah lampu, yang akan mengatur sumber tagangan 220VAC bagi lampu sehingga digunakan relay sebagai aktuatornya, karena bisa mengisolasi tegangan tinggi 220VAC dengan tegangan kontrol 3,3VDC yang berasal dari mikroprocessor.



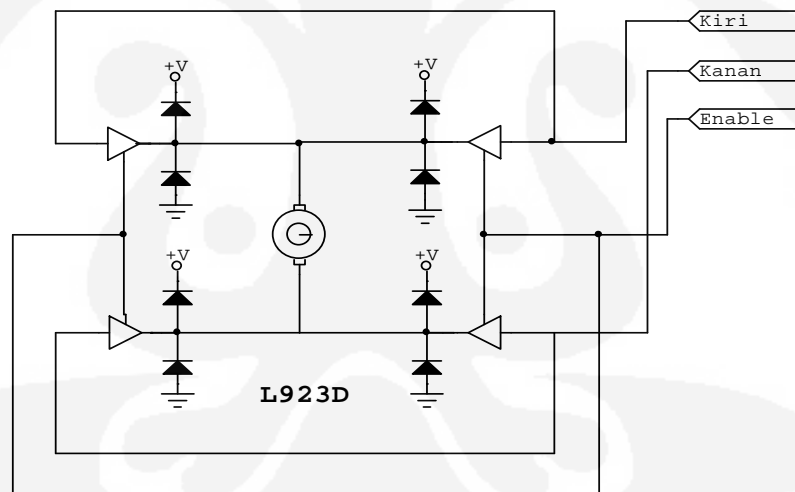
**Gambar 3.7.** Rangkaian Pengendali Lampu

Pada rangkaian diatas digunakan relay dengan tegangan kerja 5VDC dengan kemampuan aktuator sampai dengan 1Ampere pada tegangan kerja 120VAC, sehingga dalam penggunaannya bisa diperkirakan memiliki kemampuan melayani penggunaan beban maksimal 110Watt pada tegangan kerja 220VAC.

Relay tersebut dikendalikan dengan sepasang transistor yang terdiri dari sebuah transistor NPN sebagai saklar bagi relay dan transistor PNP sebagai sumber triger bagi transistor NPN. Transistor PNP akan dipicu dengan logika low oleh mikroprocessor melalui pin Basis sehingga sumber tegangan VCC akan mengalirkan arus melalui pin Emitor menuju *Collector* dan memberikan *triger* pada transistor NPN. Transistor NPN yang mendapatkan triger akan memberikan jalan bagi relay untuk meneruskan arus dari sumber tegangan VCC sehingga rangkaian menjadi *close-loop* dan seperti itulah relay akan bekerja.

### 3.2.5. Rangkaian Pengendali Motor

Rangkaian Pengendali Motor berfungsi untuk mengendalikan arah putaran motor yang lebih dikenal dengan rangkaian H-Bridge seperti yang ditunjukkan pada **Gambar 3.8** dibawah. Selain mengendalikan arah putaran, rangkaian ini juga bertanggung untuk memberikan sumber arus yang cukup agar motor dapat bekerja dengan baik. Rangkaian ini biasa dikenal dengan rangkaian H-Bridge.



**Gambar 3.8.** Rangkaian Pengendali Motor H-Bridge

Berdasarkan gambar rangkaian diatas, untuk mengendalikan putaran motor agar berputar ke arah kiri, maka Pin Kiri diberikan logika *High* dan Pin Kanan diberikan Logika *Low* dan Pin *Enable* selalu diberikan dengan logika *High* agar rangkaian selalu dalam kondisi bekerja. Berikut adalah tabel kebenaran untuk rangkaian pengendali Motor *H-Bridge*, dapat dilihat pada **Tabel 3.2** di bawah ini.



### 3.3. Komponen Yang Digunakan

Tabel 3.3. Daftar Komponen

No	Nama Komponen	Jml	Keterangan
<b>Rangkaian Utama</b>			
1.	Rabbit Processor RCM4300	1	Termasuk: uProcessor Rabbit4000 ADC ADS7870 Ethernet Controller
2.	DC Adaptor 12Volt	1	
<b>Rangkaian Power Supply</b>			
1.	Regulator LM2575	1	
2.	Regulator LM1117T	1	
3.	Dioda 1N4003	1	
4.	Kapasitor Elektrolit 47uF	1	
5.	Zener B140	1	
6.	Induktor 330uH	1	
7.	Kapasitor 330uF	1	
8.	Kapasitor Tantalum 10uF	2	
9.	Kapasitor 0.1 uF	2	
10.	Sikring 1A	1	
<b>Rangkaian Pengendali Motor DC</b>			
1.	Motor DC + Gear	1	
2.	IC H-Bridge L293D	1	
3.	Resistor 1k $\Omega$	3	
4.	Resistor 10k $\Omega$	3	
5.	Transistor NPN BC550	3	
6.	Transistor PNP BC640	3	
<b>Rangkaian Pengendali Lampu</b>			
1.	Lampu Pijar 10Watt	3	
2.	Relay 5VDC 1A@220V	3	
3.	Dioda 1N4001	3	
4.	Transistor NPN BC550	3	
5.	Transistor PNP BC640	3	
6.	Resistor 1k $\Omega$	6	
<b>Rangkaian Sensor Suhu</b>			
1.	Thermistor NTC	1	
2.	Resistor 1k $\Omega$	1	
3.	Resistor 470 $\Omega$	1	
4.	ADC ADS7870	1	

### 3.4. Rancangan Perangkat Lunak

#### 3.4.1 Rancangan Antar Muka (Web Interface)



Gambar 3.10. Tampilan antarmuka utama



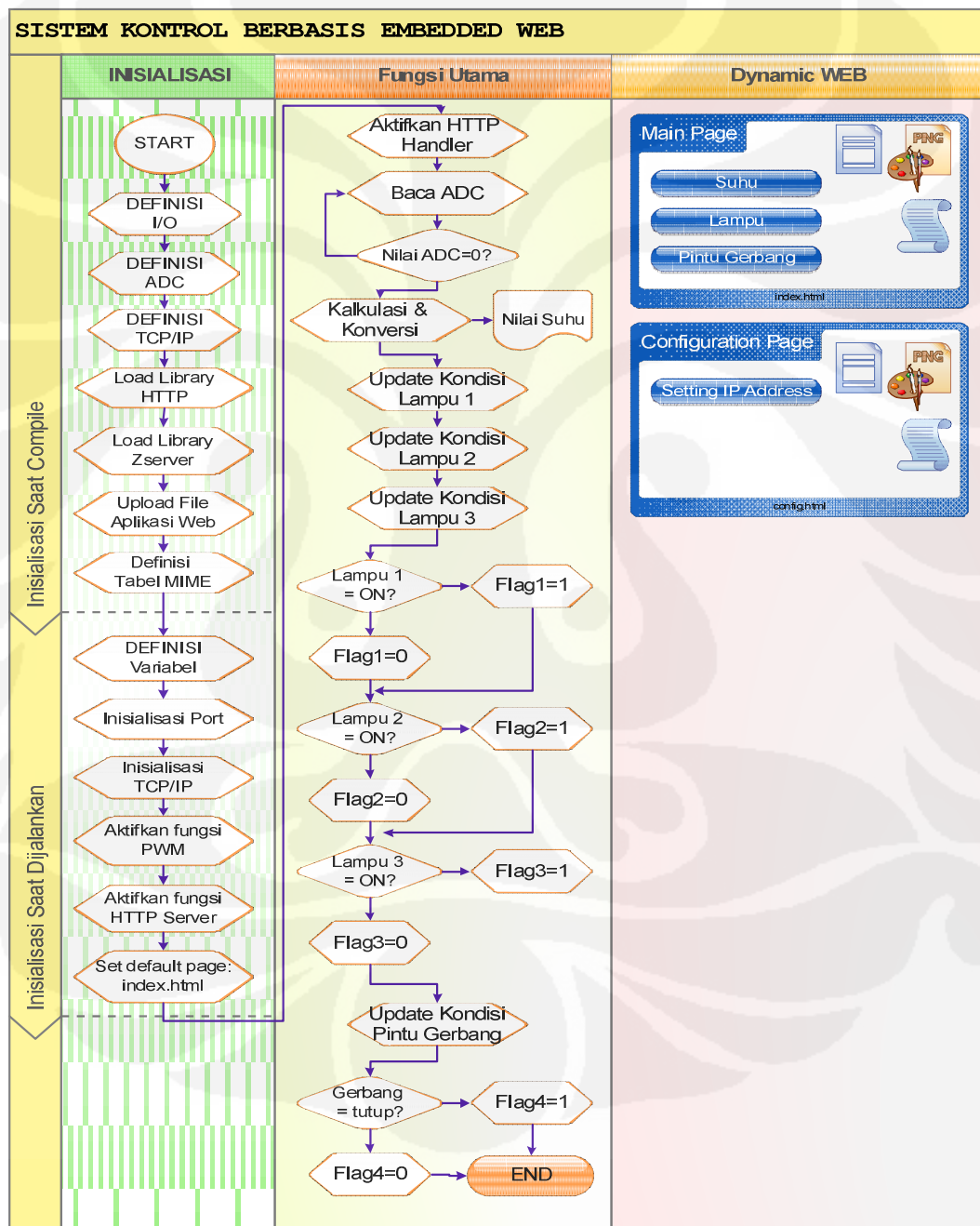
Gambar 3.11. Tampilan antarmuka konfigurasi



Gambar 3.12. Tampilan antarmuka bagian keterangan

### 3.4.2 Flow Chart

Pada Gambar 3.13 di bawah ini merupakan *Flow Chart* dari sistem yang akan dibuat yang terdiri dari 3 (tiga) bagian utama yaitu Inisialisasi, Fungsi Utama dan Tampilan Web sebagai antarmuka (Gambar 3.10). Untuk Inisialisasi terbagi menjadi 2 (dua) bagian yaitu inisialisasi saat kompilasi dan inisialisasi pada saat program dijalankan. Sedangkan pada Fungsi Utama memuat semua aktifitas dari keseluruhan fitur yang ada pada sistem. Berikut *flowchart* tersebut.



Gambar 3.13. Flow Chart

Berdasarkan flowchart diatas, pada saat source code program akan di upload ke dalam flash rom, compiler akan melakukan inisialisasi terhadap definisi I/O karena pada jenis mikroprocessor Rabbit 4000 memiliki 3 lapis fungsi untuk setiap pin I/O yaitu sebagai *Input*, *Output* dan *3-State* oleh karena itu definisi I/O harus dilakukan dan ini dilakukan pada saat compile. Aplikasi *Embedded Web* juga melibatkan penggunaan fitur *TCP/IP*, *HTTP*, *Zserver* sebagai resource manager dan tentunya melakukan *upload* beberapa file yang akan ditampilkan sebagai halaman web, dan fitur-fitur tersebut akan dinisialisasi pada saat dilakukannya kompilasi. tahap inilah yang disebut dengan ***Inisialisasi Saat Compile***.

Pada bagian program itu sendiri akan banyak melibatkan berbagai macam variabel seperti variabel untuk melakukan perhitungan nilai suhu ruangan, staus dari masing-masing lampu, status dari pintu gerbang dan beberapa variabel manipulasi lainnya. Variabel-variabel tersebut akan diinisialisasi ketika program sudah diupload dan dijalankan, inilah yang disebut dengan ***Inisialisasi Saat Dijalankan***.

Setelah melakukan kedua inisialisasi semua fitur dan variabel yang tercakup dalam dua tahap diatas, sekarang program masuk pada fungsi utama dimana fungsi utama ini akan dijalankan secara simultan terus menerus, oleh karenanya biasa disebut dengan ***mainloop***. Didalam *mainloop* tersebut akan dijalankan serangkaian proses, yaitu:

- a. HTTP dimana fungsi ini akan memberikan akses dari dunia luar (pengguna) melalui antar muka web, karena inilah mengapa aplikasi ini disebut dengan sistem kontrol berbasis ***Embedded Web***.
- b. Pembacaan Suhu melalui ADC. Fungsi ini akan memberikan informasi mengenai suhu ruangan pada saat itu dan nilai ini akan ditampilkan pada halaman web yang sudah disiapkan. Nilai suhu ruangan ini akan terus menerus diperbarui setiap 5 detik.
- c. Fungsi Kontrol lampu. Fungsi ini akan melakukan pemeriksaan terhadap semua lampu tujuannya adalah untuk merubah kondisi lampu agar sesuai dengan permintaan dari pengguna yang di-input melalui halaman web,



kemungkinan nilai input-nya hanya ada 2 (dua) yaitu ON atau OFF. Setelah itu, kondisi dari lampu akan diperiksa dan hasil dari pemeriksaan tersebut akan dimasukkan ke dalam flag yang bersesuaian agar kondisi terakhir dari masing-masing lampu tetap diingat oleh mikroprosesor.

- d. Fungsi Kontrol Pintu Gerbang. Fungsi ini akan menunggu permintaan dari pengguna untuk menutup pintu gerbang ataukah membuka pintu gerbang. Pada fungsi ini juga, kondisi dari fungsi gerbang akan selalu diingat dengan menyimpan kondisi terakhir dari pintu gerbang kedalam flag yang bersesuaian, sehingga ketika pengguna meminta untuk membuka pintu gerbang sedangkan kondisi pintu gerbang sendiri sudah terbuka, maka permintaan tersebut tidak akan dilakukan oleh mikroprosesor karena kondisi yang diminta memang sudah sesuai dengan kondisi yang sebenarnya. Kondisi dari pintu gerbang tersebut akan bisa dilihat dari halaman web, untuk memastikan bahwa halaman web memberikan informasi yang up-to-date karena telah ditinggalkan sekian lama, bisa dengan menekan tombol refresh yang disediakan oleh browser.

Untuk halaman web sendiri, disediakan 2 (dua) buah halaman, yang pertama adalah halaman utama dimana pada halaman ini memuat semua informasi mengenai objek yang dikontrol. Halaman kedua adalah halaman konfigurasi, halaman ini disediakan manakala ingin melakukan perubahan konfigurasi terhadap *IP Address, Default Gateway, DNS Server*.

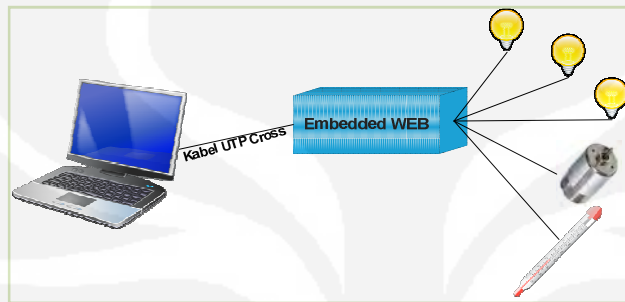
Dalam prosesnya, halaman web tersebut melakukan komunikasi dengan I/O Bus melalui variabel dengan awalan *#web*. Variabel inilah yang menjadi sarana komunikasi antara HTTP Server dengan sistem kontrol yang ada pada program mikroprocessor. Selain variabel *#web* juga disediakan fasilitas HTTP-X yang merupakan bahasa script untuk memanipulasi nilai variabel yang ada pada program utama sistem kontrol. Bahasa Script ini disebut dengan bahasa script ZHTML yang merupakan pengembangan dan penggabungan antara Bahasa C dengan bahasa HTML.

### 3.5. MODEL ANALISA

#### 3.5.1. Skema Jaringan Uji Coba

Tahapan akhir setelah perancangan dan implementasi adalah pengujian, untuk melakukan pengujian terhadap alat ini bisa menggunakan beberapa model topologi, 4 (empat) model topologi tersebut diantaranya adalah:

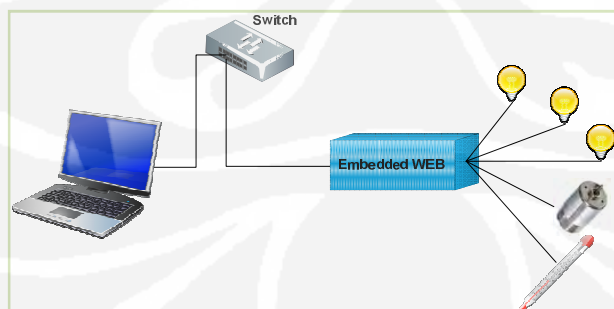
##### 3.5.1.1 Model *peer-to-peer*



**Gambar 3.14.** Skema jaringan uji coba model *peer-to-peer*

Pada model *peer-to-peer* seperti yang ditunjukkan pada **Gambar 3.14** diatas alat dihubungkan langsung dengan komputer atau laptop menggunakan kabel jaringan. Kabel yang digunakan adalah kabel UTP Cat5 dengan konfigurasi *Cross*. Penggunaan kabel *Straight* tidak bisa digunakan pada model *peer-to-peer* ini kecuali komputer atau laptop yang digunakan memiliki interface ethernet dengan fitur auto crossover detection yang biasanya dimiliki oleh *port ethernet* tipe *gigabit*.

##### 3.5.1.2. Model LAN



**Gambar 3.15.** Skema jaringan uji coba model LAN

Pada jaringan uji coba dengan model LAN seperti yang ditunjukkan melalui **Gambar 3.15** diatas, alat dihubungkan ke *switch/hub*, dan komputer

atau laptop dihubungkan ke switch/hub yang sama. Kabel yang digunakan pada model ini adalah kabel UTP Cat5 dengan konfigurasi *straight*.

### 3.5.1.3. Model WiFi

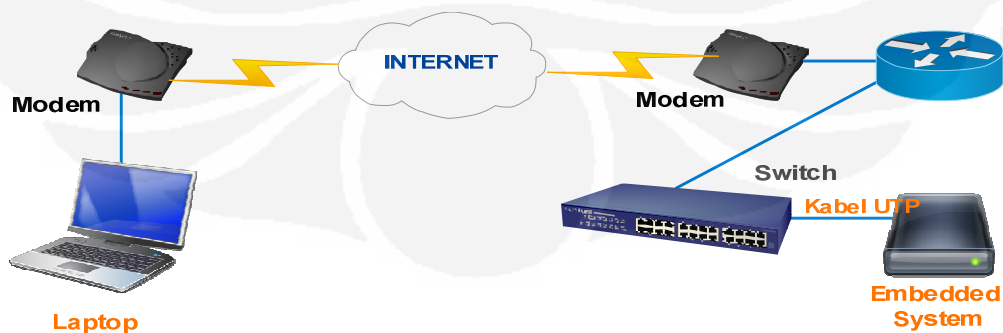
Pada jaringan *Wireless LAN*, laptop ataupun komputer pengguna akan dihubungkan dengan sistem *embedded web* melalui sambungan wireless seperti yang terlihat pada **Gambar 3.16** di bawah ini. Laptop pengguna akan dihubungkan dengan *wireless access point*, dan sistem *embedded* terhubung dengan *wireless access point* tetap menggunakan kabel *UTP*. *Embedded system* diakses dengan memasukkan alamat <http://172.16.172.23> melalui *browser*.



**Gambar 3.16.** Jaringan *Wireless LAN*

### 3.5.1.4. Model WAN

Pada jaringan model *WAN* (*Wide Area Network*) atau lebih mudah dikenal dengan sebutan internet seperti yang terlihat pada **Gambar 3.17** di bawah, koneksi antara laptop pengguna dengan *embedded system* bisa dilakukan tanpa ada batasan jarak. Jika pada jaringan uji coba sebelumnya hanya sebatas beberapa ratus meter karena keterbatasan jaringan akses menggunakan kabel dan wireless, dengan model jaringan *WAN* ini proses pengujian bisa dilakukan dari manapun selama antara kedua tempat terhubung dengan jaringan Internet.



**Gambar 3.17.** Jaringan Internet (*WAN*)

### 3.5.2. Parameter Evaluasi

Parameter evaluasi yang ditentukan disini merupakan parameter-parameter uji yang mungkin untuk dilakukan terhadap sistem yang dibuat. Secara garis besar proses evaluasi akan mengarah pada sensor suhu, aktuator lampu, motor dc dan kehandalan dari sistem itu sendiri. Mengingat interface dari sistem ini merupakan pokok bahasan utama, maka *interface* tersebut yang berupa konten halaman web juga akan dilakukan pengujian baik itu pengujian fungsional maupun pengujian terhadap kehandalan dari web itu sendiri.

Parameter evaluasi tersebut adalah:

- a. Pengujian Fungsi
- b. Pengujian Tingkat Akurasi
- c. Uji Kehandalan Sistem

Pengujian Fungsi akan meliputi pengujian terhadap fungsi-fungsi yang ada pada sistem secara menyeluruh dari mulai tahap input, proses hingga output yang dihasilkan. Pengujian tingkat akurasi dilakukan terhadap sensor suhu. Metode pengujian yang digunakan adalah dengan cara membandingkan hasil pembacaan antara sensor suhu dari sistem yang dibuat dengan hasil pembacaan dari sensor suhu standar. Dalam prosesnya akan digunakan Thermometer air raksa sebagai acuan suhu standar, selisih hasil pembacaan dari kedua sensor inilah yang akan dijadikan sebagai gambaran tingkat akurasi dari sensor suhu yang dibuat. Uji kehandalan sistem akan memberikan gambaran mengenai kekuatan dari sistem yang dibuat, selain itu juga akan memberikan gambaran mengenai kemampuan dari sistem jika sistem ini diaplikasikan pada lingkungan penggunaan yang sebenarnya yaitu *LAN*, *Wireless LAN* dan jaringan Internet (*WAN*)

# BAB 4

## IMPLEMENTASI DAN EVALUASI

Pada bab ini akan dibahas secara menyeluruh mengenai proses implementasi dan evaluasi kinerja dari ‘Sistem Kontrol Berbasis *Embedded Web*’. Pembahasan akan dimulai dari persiapan hardware, software yang digunakan, desain antarmuka web, proses kompilasi dan embedding, sampai dengan evaluasi kinerja. Kesimpulan dari hasil analisa ini akan disajikan pada bagian selanjutnya di Bab 5.

### 4.1. IMPLEMENTASI

Selanjutnya adalah proses implementasi ‘Sistem Kontrol Berbasis *Embedded Web*’ yang sebelumnya telah dirancang dan disajikan pada **Bab 3 Perancangan**. Proses implementasi ini akan melibatkan beberapa rangkaian dan software baik itu *software* hasil perancangan maupun *software* lain yang digunakan dalam proses implementasi dan desain antarmuka web.

#### 4.1.1. Hardware

Hardware yang akan digunakan adalah:

- a. Rangkaian Utama, yaitu Modul RCM4300
- b. Rangkaian Sensor Suhu
- c. Rangkaian Pengendali Lampu
- d. Rangkaian Pengendali Motor
- e. *USB Downloader*

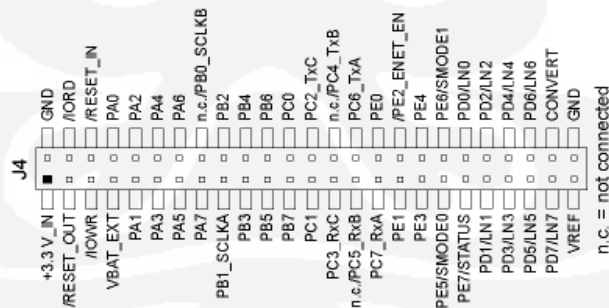
Rangkaian Utama merupakan modul siap pakai dengan model RCM4300 menggunakan mikrokontroler Rabbit 4000. Bentuk fisik dari modul tersebut bisa dilihat pada **Gambar 4.1**. Didalam modul tersebut telah dilengkapi dengan *I/O Port*, *ADC*, *PWM* dan *ethernet port* lengkap dengan kontrolernya. Rangkaian Sensor Suhu merupakan sambungan seri antara resistor  $1k\Omega$  dengan *thermistor NTC* sebagai *probe* sensornya, tegangan pada *NTC* akan dikonversi menggunakan *ADC ADS7870* yang sudah terintegrasi pada modul RCM4300 yang terletak pada pin LN7, *ADC* tersebut sekaligus sebagai *Signal Conditioning* karena didalam

ADS7870 telah terintegrasi fitur *PGA (Programmable Gain Amplifier)* sehingga penguatan sinyal input bisa ditentukan dari software.



**Gambar 4.1.** Fisik Modul Rabbit RCM4300. [7]  
(a) tampak atas (b) tampak bawah

Untuk Rangkaian Pengendali Lampu yang sudah dibahas sebelumnya pada Bab 3, digunakan serangkaian relay yang dikendalikan melalui *I/O Port* PB.1, PB.2 dan PB.3. Relay yang digunakan memiliki spesifikasi tegangan kumparan 5V sedangkan output dari *I/O* yang digunakan memiliki level tegangan 3,3V pada saat high dan 0,6V pada saat low, maka pada hasil perancangan (yang bisa dilihat pada **Gambar 3.7**) dapat dilihat dua buah transistor yang dirangkai cascade agar bisa berfungsi pada level tegangan 3,3 Volt sekaligus berfungsi sebagai saklar untuk mengendalikan relay. Selanjutnya rangkaian Pengendali Lampu Tersebut dihubungkan dengan modul RCM4300 melalui konektor *I/O* seperti yang ditunjukkan pada **Gambar 4.2** di bawah ini.



**Gambar 4.2.** Pinout Modul RCM4300 [3]

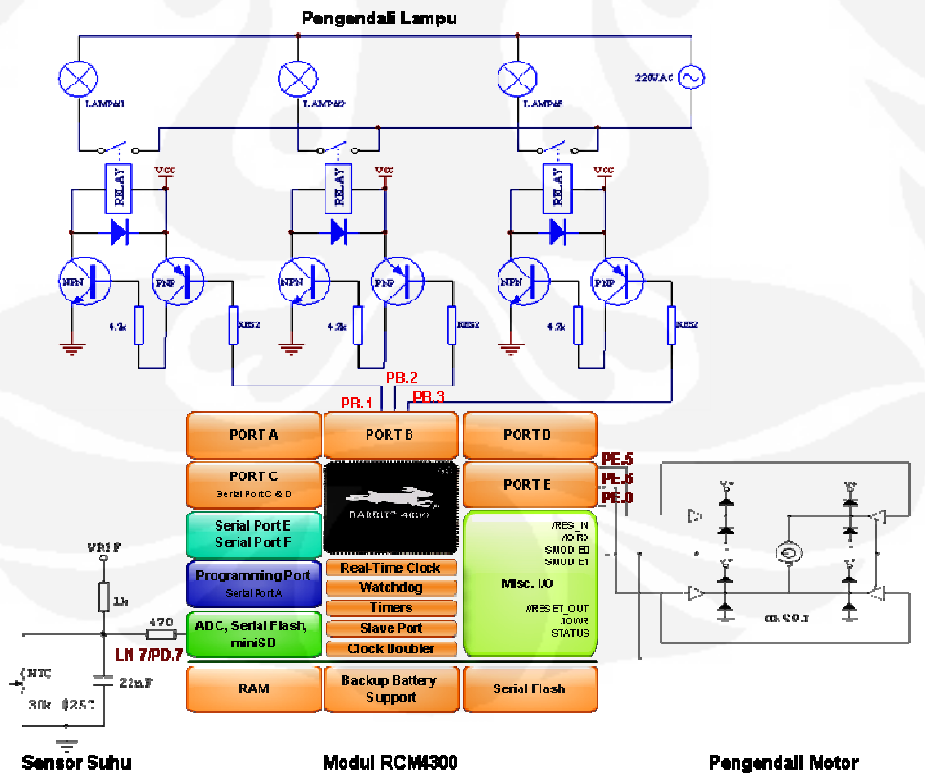
Rangkaian pengendali motor digunakan rangkaian *H-Bridge* yang sudah dibahas sebelumnya pada Bab 3 di **Gambar 3.8** sebagai sumber arus bagi motor dengan tegangan kerja 5V, sedangkan untuk mengendalikan kecepatan motor digunakan *PWM* yang sudah terintegrasi dengan modul RCM4300. *Output* dari

PWM ini berada pada PE.0 sedangkan untuk mengendalikan arah putaran digunakan port PE.5 dan PE.6.



**Gambar 4.3.** *USB Downloader* <sup>[6]</sup>

*USB Downloader* digunakan untuk memasukan kode program yang telah dibuat ke dalam *flash memory*. Bentuk fisik dari *USB Downloader* terlihat seperti **Gambar 4.3** di atas, pada satu ujungnya memiliki konektor *USB* untuk ditancapkan ke komputer dan ujung satunya berupa konektor *IDC* yang akan ditancapkan ke *programming port* pada modul RCM4300. Dengan bantuan software *Dynamic C*, kode program yang telah dibuat kemudian di *download* oleh sistem menggunakan kabel ini. Konfigurasi lengkap dari keseluruhan hardware akan tampak seperti **Gambar 4.4** di bawah ini.



**Gambar 4.4** Konfigurasi Hardware Keseluruhan

## 4.1.2. Software

Pada bagian ini akan menjelaskan mengenai implementasi *Embedded Web* dari sisi software yang akan meliputi fungsi kendali *I/O* melalui *Web* untuk mengendalikan lampu, pintu gerbang dan sensor suhu. Kompiler yang digunakan adalah Dynamic C yang merupakan kompiler bahasa C yang sudah dimodifikasi khusus untuk microprocessor jenis Rabbit. Melalui Dynamic C inilah Aplikasi Sistem Kontrol Berbasis *Embedded Web* dibangun. Sedangkan untuk bagian web digunakan bahasa HTML dan ZHTML.

### 4.1.2.1. Aplikasi Sistem Kontrol Berbasis *Embedded Web*

Secara garis besar aplikasi sistem ini terbagi menjadi 2 (dua) bagian, yaitu:

- Kode program aplikasi yang akan dibahas pada sub-bab 4.1.2.2, berisi kode-kode program untuk fasilitas *Web Server*, mengontrol *I/O* dan tentunya mendukung fasilitas *TCP/IP*.
- Konten dari web itu sendiri (sub-bab 4.1.2.3), berisi halaman web yang akan dijadikan sebagai antarmuka antara pengguna dan sistem.

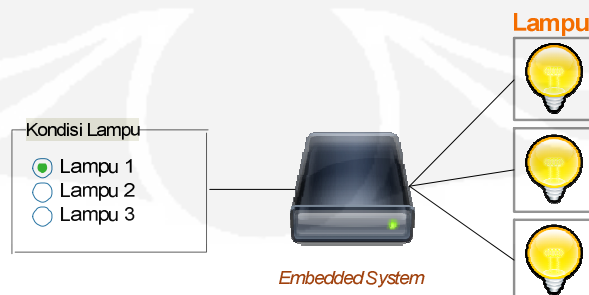
Berikut adalah cuplikan kode program dari aplikasi yang dimaksud.

### 4.1.2.2. Kode Program Aplikasi

Pada sub-bab ini akan menjelaskan bagian-bagian dari kode program aplikasi dalam mengontrol lampu, pintu gerbang dan sensor suhu.

#### a. Kontrol On/Off

Bagian ini merupakan *interfacing* antara tombol *radio button* pada halaman web dengan salah satu port *I/O* pada modul RCM4300 seperti yang di ilustrasikan pada **Gambar 4.5** di bawah, yang dalam hal ini adalah PB.1, PB.2 dan PB.3 dimana ketiga pin tersebut adalah pin yang digunakan untuk mengontrol lampu.



**Gambar 4.5.** Ilustrasi Kontrol Lampu



Untuk mengontrol sebuah lampu pada prinsipnya adalah mengontrol kondisi output dari sebuah pin I/O, dalam sistem ini kondisi Lampu 1 akan menyala jika kondisi pin PB.1 dalam keadaan LOW atau 0, dan Lampu 1 akan padam jika kondisi pin PB.1 dalam keadaan HIGH atau 1. Begitu pula dengan Lampu 2 dan Lampu 3 yang kondisinya akan bersesuaian dengan pin PB.2 dan PB.3. Dalam bahasa Dynamic C untuk mengkondisikan agar Lampu 1 menyala, perintahnya sebagai berikut:

```
Lamp1=0;
BitWrPortI(PBDR, &PBDRShadow, Lamp1, 1);
```

Atau mengkondisikan Lampu 1 menjadi padam dengan perintah:

```
Lamp1=1;
BitWrPortI(PBDR, &PBDRShadow, lamp1, 1);
```

Struktur perintah diatas adalah sebagai berikut:

```
void BitWrPortI(int io_port, char *PORTShadow, int value, int bitcode);
```

dimana:

- io\_port** adalah alamat dari port yang akan dikontrol
- \*PORTShadow** adalah alamat dari variabel shadow dari port diatas
- Value** adalah nilai yang akan diberikan. 1=High; 0=Low. dalam hal ini diwakili oleh variabel Lamp1
- Bitcode** adalah bit data (0-7) yang akan ditulis

Setelah itu menghubungkan perintah diatas agar bisa dieksekusi melalui *radio button* dari halaman web. **Tabel 4.1** di bawah ini adalah script HTML untuk dua buah radio button yang nantinya akan difungsikan sebagai tombol ON/OFF.

**Tabel 4.1.** Design Radio Button

<b>Design</b>	<input checked="" type="radio"/> ON <input type="radio"/> OFF
<b>Kode</b>	<pre>&lt;html&gt;   &lt;body&gt;     &lt;form id="Form1" name="Form1" method="post" action=""&gt;       &lt;p&gt;         &lt;input name="radiobutton" type="radio" value="radiobutton" /&gt;         ON         &lt;input name="radiobutton" type="radio" value="radiobutton" /&gt;         OFF       &lt;/p&gt;     &lt;/form&gt;   &lt;/body&gt; &lt;/html&gt;</pre>

Agar Radio Button diatas dapat mengendalikan I/O untuk melakukan kontrol terhadap kondisi lampu, maka didalam kode HTML tersebut ditambahkan kode ZHTML yang berfungsi untuk menyesuaikan kondisi pin PB.1 agar sesuai dengan kondisi *radio button* diatas. Berikut adalah kode setelah ditambahkan fungsi kontrol menggunakan bahasa ZHTML seperti yang ditunjukkan pada script di bawah ini.

```

<html>
<body>
<form id="form1" name="form1" method="post" action="/OnOff.zhtml">
<p>
<input name="lamp1" type="radio" value="0" onclick="submit()"/>
ON
<?z if ($lamp1==0) { ?>
    CHECKED
    <?z } ?>
<?z if ($lamp1flag==0) { ?>
    checked="checked"
    <?z } ?>

<input name="lamp1" type="radio" value="1" onclick="submit()"/>
OFF
<?z if ($lamp1==1) { ?>
    CHECKED
    <?z } ?>
    <?z if ($lamp1flag==1) { ?>
        checked="checked"
        <?z } ?>
</p>
</form>
</body>
</html>

```

Dari script diatas terlihat dua buah variabel terlibat, yaitu **lamp1** dan **lamp1flag**. Variabel tersebut sebelumnya telah ditentukan didalam program utama dengan mendklarasikan variabel:

```

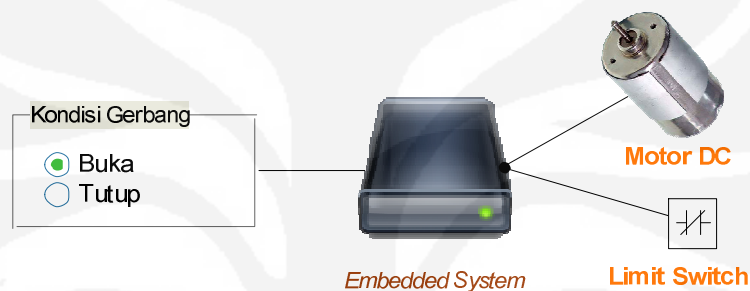
Int    lamp1;
#web  lamp1;

```

dimana variabel **lamp1** berfungsi untuk menampung nilai yang di input melalui radio button (perintah), sedangkan **lamp1flag** berisi nilai konfirmasi dari pin PB.1 yang akan mengindikasikan nilai PB.1 pada saat itu (kondisi terkini). Pseudocode dari program utama bisa dilihat pada bagian **e. Fungsi Utama**.

## b. Kontrol Buka/Tutup Pintu Gerbang

Tidak berbeda jauh dengan fungsi kontrol yang diterapkan pada lampu, untuk membuka dan menutup pintu gerbang juga digunakan *Radio Button* sebagai media *input* dari pengguna. Agar lebih jelasnya bisa dilihat pada **Gambar 4.6** di bawah ini, dua buah radio button yang akan mewakili sebuah variabel **gerbang**, *radio button* dengan nama **Buka** akan membuat isi dari variabel **gerbang** bernilai 0 dan *radio button* dengan nama **Tutup** akan membuat isi dari variabel **gerbang** bernilai sebaliknya yaitu 1.



**Gambar 4.6.** Ilustrasi Kontrol Pintu Gerbang

Berikut adalah script ZHTML untuk fungsi kontrol terhadap pintu gerbang.

```
<body>
<div align="center">
  <form id="form1" name="form1" method="post" action="">
    <label>
      <span class="style10">Buka</span>
      <input name="gerbang" type="radio" value="1" onclick="submit()"
      <?z if ($gerbang_flag==1) { ?>
        checked="checked"
      <?z } ?>
      <?z if ($gerbang==1) { ?>
        CHECKED
      <?z } ?>
    </label>
    <label>
      <input name="gerbang" type="radio" value="0" onclick="submit()"
      <?z if ($gerbang_flag==0) { ?>
        checked="checked"
      <?z } ?>
      <?z if ($gerbang==0) { ?>
        CHECKED
      <?z } ?>
    </label>
    <span class="style4"> Tutup</span></label>
  </form>
</div>
</body>
</html>
```

### c. Menampilkan Nilai Suhu

Informasi suhu didapatkan dari Thermistor NTC yang kemudian diumpankan ke *signal conditioning* dan ADC baru kemudian bisa dibaca oleh mikropresesor.

Untuk membaca nilai ADC, dalam Dynamic C perintahnya adalah:

```
Auto float Draw;  
Draw = anaIn(7, SINGLE, GAIN_1);
```

Struktur dari perintah diatas adalah

```
int anaIn(int channel, Int Mode, int gaincode)
```

dimana:

**channel** adalah channel ADC yang digunakan  
**Mode** adalah mode yang digunakan, single atau differential.  
**Gaincode** adalah nilai Programmable Gain Amp. Yang digunakan

Dengan menggunakan perintah diatas, nilai suhu saat ini berada di variabel **Draw**. Kemudian dilakukan beberapa perhitungan aritmetika untuk mendapatkan nilai suhu dalam satuan Celcius yang di tempatkan di variabel **Tc**. Untuk menampilkan nilai suhu yang berada didalam variabel **Tc**, diperlukan tambahan script ZHTML yang diselipkan dalam *body* HTML seperti yang ditunjukkan pada Tabel 4.2 di bawah ini

Tabel 4.2. Design Tampilan Suhu

Design	25.4 °C
Kode	<pre>&lt;html&gt;   &lt;body&gt;     &lt;div align="left"&gt;       &lt;?z printf("%.1f", \$hum) ?&gt;     &lt;/div&gt;   &lt;/body&gt; &lt;/html&gt;</pre>

#### d. Fungsi Networking dan Web Server

Dengan menggunakan modul Rabbit RCM4300 (**Gambar 4.1**), kebutuhan hardware untuk Networking dan Web Server telah terintegrasi di dalam modul tersebut. Untuk menggunakannya harus dilakukan inisialisasi pada saat compile dan juga ada inisialisasi yang harus dilakukan pada saat program pertama kali dijalankan.

Inisialisasi yang dilakukan pada saat compile adalah:

```
#use RCM43xx.LIB
#use "dcrtcp.lib"
#use "http.lib"
#define TCPCONFIG 6
#define MY_IP_ADDRESS "172.16.172.23"
#define MY_NETMASK "255.255.255.0"
#define MY_GATEWAY "172.16.172.1"
#define MY_NAMESERVER "172.16.172.1"

#define USE_RABBITWEB 1
#define HTTP_HOMEDIR "/"
#define HTTP_DFLTFILE "index.html"

#define IFCONFIG_ETH0 \
    IFS_IPADDR,atona(MY_IP_ADDRESS), \
    IFS_NETMASK,atona(MY_NETMASK), \
    IFS_ROUTER_SET, atona(MY_GATEWAY), \
    IFS_NAMESERVER_SET, atona(MY_NAMESERVER), \
    IFS_UP

#ximport "skripsi/web/Design4/Design4.html" Design4_html
SSPEC_MIMETABLE_START
    SSPEC_MIME_FUNC(".zhtml", "text/html", zhtml_handler),
    SSPEC_MIME(".html", "text/html"),
    SSPEC_MIME(".png", "image/png"),
    SSPEC_MIME(".jpg", "image/jpg")
SSPEC_MIMETABLE_END
SSPEC_RESOURCETABLE_START
    SSPEC_RESOURCE_XMEMFILE("/index.html", Design4_html),
SSPEC_RESOURCETABLE_END
```

Inisialisasi pada saat program dijalankan:

```
sock_init();
http_init();
tcp_reserveport(80);
http_handler();
```

Dengan demikian fungsi untuk Networking dan Web Server telah siap digunakan. Selanjutnya setelah dilakukan inisialisasi Network diatas, kemudian dilanjutkan dengan fungsi utama yang akan di bahas melalui *pseudocode* pada sub-bab **e. Fungsi Utama** dibawah ini.

## e. Fungsi Utama

### NEVER ENDING LOOP while(1)

#### Process with Multitasking:

```
Task 1: IF Rubah_Conf = 2           //Cek apakah dalam proses merubah IP
        Abort
    ELSE
        Call Http_Handler()
Task 2: DO                           //Mendapatkan Nilai Suhu
    READ ADC_Data
    PUT ADC_Data into DataRaw
    WHILE DataRaw = 0
    CALCULATE Tc                       //Hitung Nilai Suhu dalam Celcius
Task 3: IF Lamp1 != 2 And Lamp1 != Lamp1_Flag //Periksa Status Lampu 1
        Set PB.1 ← Lamp1             //Nilai PB.1 (lampu1) disamakan dgn Lamp1
    IF PB.1 = 0
        Lamp1_Flag = 0
    ELSE
        Lamp1_Flag = 1
    END IF
    Set Lamp1 = 2                       //Nilai 2 berarti proses periksa selesai
    CALL beep()
    END IF
    IF Lamp2 != 2 And Lamp2 != Lamp2_Flag //Periksa Status Lampu 2
        Set PB.2 ← Lamp2             //Nilai PB.2 (lampu2) disamakan dgn Lamp2
    IF PB.2 = 0
        Lamp2_Flag = 0
    ELSE
        Lamp2_Flag = 1
    END IF
    Set Lamp2 = 2                       //Nilai 2 berarti proses periksa selesai
    CALL beep()
    END IF
    IF Lamp3 != 2 And Lamp3 != Lamp3_Flag //Periksa Status Lampu 3
        Set PB.3 ← Lamp3             //Nilai PB.3 (lampu3) disamakan dgn Lamp3
    IF PB.3 = 0
        Lamp3_Flag = 0
    ELSE
        Lamp3_Flag = 1
    END IF
    Set Lamp3 = 2                       //Nilai 2 berarti proses periksa selesai
    CALL beep()
    END IF
Task 4: IF Bunyi_Beep = 0
        Abort
    ELSE
        SET Buzzer
        Wait For Delay: 100ms
        RESET Buzzer
        Wait For Delay: 1s
    END IF
```

```

Task 5: IF Gerbang = 0 //Apakah ada perintah utk tutup gerbang?
    IF Gerbang_Status = sudah_tutup
        Abort
    END IF
    Set Bunyi_Beep
    Set Direction of Motor = Right/Close
    Set Speed of Motor = High
    Reset Gerbang_Flag
    ACTIVATE MOTOR
    READ Limit_Switch
    IF Limit_Switch != 0
        Set Speed of Motor = Low
        Activate New Speed
    END IF
    Wait For Delay: 4.5 Seconds
    Set gerbang_status = sudah_tutup
    Set Gerbang = 2 //Nilai 2 berarti proses telah selesai
ELSE //Apakah ada perintah utk buka gerbang?
    IF Gerbang = 1
        IF Gerbang_Status = sudah_buka
            Abort
        END IF
        Set Bunyi Beep
        Set Direction of Motor = Left/Open
        Set Speed of Motor = High
        Set Gerbang_Flag
        ACTIVATE MOTOR
        READ Limit_Switch
        IF Limit_Switch != 1
            SET Speed of Motor = Low
            ACTIVATE New Speed
        END IF
        Wait For Delay: 1.8 seconds
        SET Gerbang_Status = sudah_buka
        SET Gerbang = 2 //Indikasi bahwa proses selesai
    END IF
    SET Direction of Motor = Don't Move
    SET Speed of Motor = 0
    ACTIVATE New Speed
    RESET Bunyi_Beep
Task 6: IF Rubah_Conf = 1 And New_Network != Current_Network
    SET Rubah_conf = 2
    BRING Network Down
    COPY New_Network → Current_Network
    BRING Network UP
    RESET Rubah_Conf = 0
ELSE
    RESET Rubah_Conf = 0
    Abort
END IF

```

**END OF LOOP**

### 4.1.2.3. Konten Antarmuka Web

#### a. Halaman Utama

Design grafis dari halaman utama bisa dilihat pada **Gambar 4.7**. Halaman ini terdiri dari 5 (lima) *frame* utama, yaitu *frame* A, B, C, D, dan E. didalam *frame* utama tersebut terdapat *i-frame* E1, E2, E3, C1, C2, C3, C4, dan C5.



**Gambar 4.7.** Komponen Halaman Utama

Masih berdasarkan **Gambar 4.7** di atas, *Frame* A adalah *Header* yang terletak paling atas, berisi logo Universitas Indonesia pada sebelah kiri dan *link* menuju halaman lain terletak di sebelah kanan. *Frame* B adalah *frame* tipis yang difungsikan sebagai *bar* pembatas estetika antara bagian atas dengan bagian bawahnya. *Frame* C terletak di kanan tengah untuk menempatkan informasi *monitoring* suhu, kondisi lampu dan posisi pintu gerbang. *Frame* D adalah bagian *footer*. *Frame* E merupakan *frame* yang memiliki area paling luas, ditempatkan dalam *frame* ini sebuah gambar rumah dengan 3 buah lampu dan sebuah pintu gerbang dengan maksud menggambarkan kondisi rumah yang sebenarnya.

Didalam *Frame* C dan *Frame* E terdapat *i-frame* (*inline frame*). *I-frame* C1 merupakan informasi suhu, *i-frame* C2 memberikan informasi mengenai posisi pintu gerbang, *i-frame* C3 sampai dengan C5 memberikan informasi mengenai kondisi lampu1, lampu2, dan lampu3 secara berurutan. Selain itu, didalam *frame* E juga terdapat *i-frame* E1, E2, dan E3 yang menggambarkan kondisi lampu



secara *visual*, dimana jika lampu dalam keadaan hidup digambarkan dengan gambar lampu yang berpijar cerah, sebaliknya jika lampu dalam keadaan mati maka digambarkan dengan gambar lampu yang berwarna gelap.

Setiap *frame* ataupun *i-frame* memiliki file html yang terpisah, keuntungan dari hal ini adalah ketika kita ingin memperbarui/*refresh* informasi tertentu cukup hanya dengan melakukan *refresh* terhadap frame yang bersesuaian saja, sehingga proses *refresh* tidak melibatkan halaman statik lain yang tidak memerlukan *refresh*, dalam sistem *embedded* trik ini cukup bermanfaat mengingat *resource* dari *processor* yang terbatas, trik ini menjadikan kinerja sistem menjadi lebih cepat.

## b. Halaman Konfigurasi

Halaman konfigurasi seperti yang diperlihatkan pada **Gambar 4.8** di bawah ini merupakan halaman web yang berisi *form* untuk melakukan perubahan konfigurasi pada sistem. Konfigurasi yang bisa dilakukan pada halaman ini adalah konfigurasi jaringan *IP Address*, *Netmask*, *Gateway* dan *DNS Server*. Komposisi dari halaman konfigurasi tidak jauh berbeda dengan halaman utama selain pada bagian tengah yaitu *Frame C* dan *Frame E*, dimana pada *frame E* tersebut kontennya diganti menjadi halaman lain dan pada *Frame C* dibiarkan kosong.



**Gambar 4.8.** Komponen Halaman Konfigurasi

### c. Halaman Informasi

Pada halaman Informasi yang ditunjukkan pada **Gambar 4.9** di bawah ini, hanya berisi konten informasi yang mencantumkan judul dari sistem yang dibuat, nama pembimbing dan nama penulis sendiri.

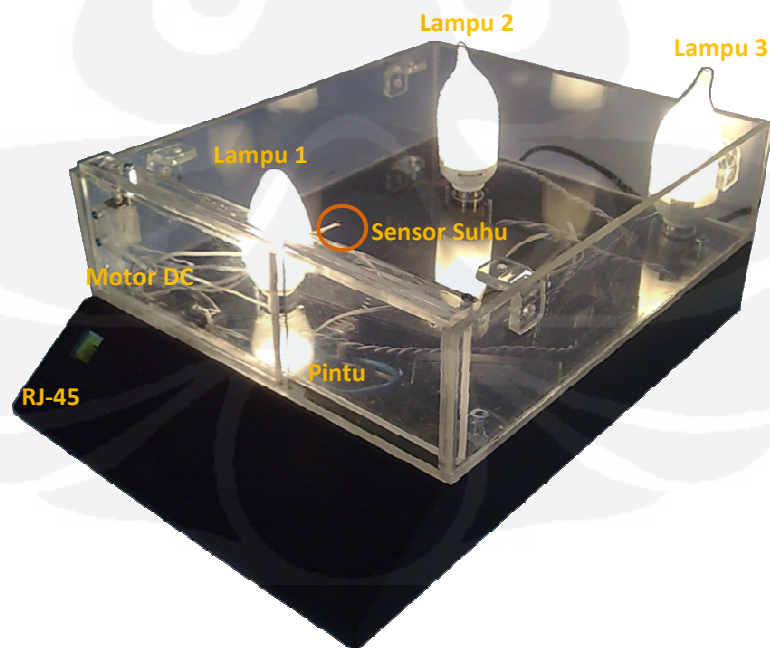


**Gambar 4.9.** Komponen Halaman Informasi

Seperti halnya dengan halaman konfigurasi, konten pada halaman informasi hanya terdiri dari sebuah halaman about.html dengan sebuah gambar bola sebagai estetika.

#### 4.1.2.4 Miniatur

Produk akhir dari Implementasi Sistem Kontrol Berbasis Embedded Web berupa miniatur seperti **Gambar 4.10** di bawah ini.



**Gambar 4.10.** Implementasi Alat Berbentuk Miniatur Rumah

## 4.2. EVALUASI KINERJA

Pada bagian ini akan memberikan gambaran mengenai proses evaluasi dari kinerja sistem yang dibuat. Kegiatan evaluasi ini akan membahas mengenai peralatan yang digunakan selama melakukan evaluasi, menentukan parameter-parameter apa saja yang akan dijadikan sebagai bahan evaluasi dan pada bagian akhir akan diberikan ringkasan dari hasil evaluasi tersebut. Karena sistem yang dibuat memiliki fasilitas *TCP/IP* maka dalam proses evaluasi akan ditentukan topologi jaringan yang digunakan dalam melakukan evaluasi. Berikut lebih detail mengenai evaluasi.

### 4.2.1. Peralatan yang Digunakan

Beberapa alat bantu digunakan dalam melakukan evaluasi, alat tersebut merupakan alat ukur dan beberapa diantaranya adalah alat bantu. Berikut adalah daftar peralatan tersebut.

- a. Sistem Kontrol Berbasis *Embedded Web*
- b. Miniatur Rumah
- c. Laptop
- d. Kabel Jaringan UTP Cat.5 *Straight*
- e. *AVO Meter*
- f. Pemanas
- g. *Stopwatch*
- h. Thermometer Air Raksa
- i. *Switch*
- j. *Modem*
- k. *Router*
- l. *Wireless Access Point*
- m. Layanan Akses Internet

Sistem Kontrol Berbasis *Embedded Web* sebagai *target* evaluasi yang dibantu dengan miniatur rumah sebagai ilustrasi penggambaran manfaat dari sistem tersebut yang dilengkapi dengan miniatur pintu gerbang. Laptop disini digunakan untuk mengakses halaman web yang telah tertanam di dalam sistem, dengan

menggunakan Kabel Jaringan *UTP* yang disambung lurus/*straight* laptop dapat berkomunikasi dengan sistem melalui kabel tersebut.

Sedangkan *AVO Meter*, *Stopwatch* dan *Thermometer* air raksa digunakan sebagai alat ukur standar yang akan dijadikan sebagai tolok ukur dalam melakukan evaluasi kinerja dari sistem. *AVO Meter* digunakan dalam pengukuran arus dan tegangan untuk selanjutnya dilakukan perhitungan konsumsi daya dari sistem. *Stopwatch* dimanfaatkan untuk menghitung waktu atau kecepatan respon dan *Thermometer* akan digunakan sebagai pembanding pembacaan sensor suhu dari sistem, nilai selisih dari hasil pembacaan tersebutlah yang akan dijadikan sebagai tingkat akurasi dari sensor. Digunakan juga pemanas sebagai alat bantu dalam menciptakan suhu lingkungan agar sesuai dengan parameter yang diuji yaitu pada suhu kamar 25°C, 30°C, 40°C dan seterusnya.

*Switch*, *Modem*, *Router*, *Wireless Access Point* dan Layanan akses internet digunakan untuk menciptakan lingkungan uji coba bagi sistem. Dimana proses pengujian akan dilakukan tidak hanya koneksi langsung antara laptop dan sistem yang diuji (*peer-to-peer*) tetapi juga akan diuji dalam lingkungan jaringan *LAN*, *WiFi*, dan akses jarak jauh melalui internet (*WAN*), maka itulah dibutuhkan peralatan jaringan tersebut termasuk layanan akses untuk internet.

## **4.2.2. Parameter Evaluasi**

Parameter evaluasi yang ditentukan disini merupakan parameter-parameter uji yang mungkin untuk dilakukan terhadap sistem yang dibuat. Secara garis besar proses evaluasi akan mengarah pada sensor suhu, aktuator lampu, motor dc dan kehandalan dari sistem itu sendiri. Mengingat interface dari sistem ini merupakan pokok bahasan utama, maka interface tersebut yang berupa konten halaman web juga akan dilakukan pengujian baik itu pengujian fungsional maupun pengujian terhadap kehandalan dari web itu sendiri. Berikut lebih detil mengenai parameter evaluasi tersebut.

### **4.2.2.1. Uji Fungsi**

Uji fungsi ini meliputi pengujian terhadap fungsi-fungsi yang ada pada sistem secara menyeluruh dari mulai tahap input, proses hingga output yang dihasilkan. Fungsi-fungsi yang akan diuji adalah sebagai berikut.

#### **a. Lampu**

Pengujian ini dilakukan untuk memberikan gambaran bahwa fungsi dari pengendali lampu sudah berfungsi dengan baik sesuai dengan hasil perancangan. Pengujian ini dilakukan dengan cara mengamati kondisi dari setiap lampu, apakah dalam keadaan hidup lampu dapat berpijar dengan baik atau tidak. Perubahan kondisi dari setiap lampu tersebut juga diamati ketika terjadi perubahan posisi pada tombol, apakah kondisi lampu terpengaruh dengan kondisi tombol pada antarmuka web dan apakah kondisi lampu-lampu tersebut sesuai dengan kondisi yang ditampilkan pada antarmuka web.

#### **b. Sensor Suhu**

Pengujian fungsi pada sensor suhu dilakukan dengan cara memberikan pergeseran suhu pada lingkungan sekitar sensor tersebut untuk mengetahui apakah sensor memberikan respon terhadap perubahan suhu tersebut. Jika sensor memberikan respon terhadap perubahan suhu yang diberikan maka dianggap sensor tersebut berfungsi, jika tidak maka kesimpulannya akan sebaliknya. Pengujian berikutnya pada sensor suhu adalah menguji tingkat akurasi yang akan dijelaskan pada sub bab berikutnya.

#### **c. Motor DC**

Pengujian fungsi pada motor DC dilakukan dengan cara melakukan kontrol terhadap motor tersebut berupa perintah arah putaran kanan, kiri dan berhenti untuk mengetahui apakah motor tersebut mampu merespon setiap perintah yang diberikan. Jika motor tersebut memberikan respon yang sesuai maka fungsi dari motor tersebut dianggap baik.

#### **4.2.2.2. Tingkat Akurasi**

Pengujian tingkat akurasi ini dilakukan terhadap sensor suhu. Metode pengujian yang digunakan adalah dengan cara membandingkan hasil pembacaan antara sensor suhu dari sistem yang dibuat dengan hasil pembacaan dari sensor suhu standar. Dalam prosesnya akan digunakan Thermometer air raksa sebagai acuan suhu standar, selisih hasil pembacaan dari kedua sensor inilah yang akan

dijadikan sebagai gambaran tingkat akurasi dari sensor suhu yang dibuat. Hasil pengukuran dari tingkat akurasi ini bisa dilihat pada sub bab **2.4. Hasil Evaluasi**.

#### **4.2.2.3. Kehandalan**

Setelah sebelumnya dilakukan uji fungsi dan uji tingkat akurasi, tahap selanjutnya adalah melakukan evaluasi terhadap kehandalan dari sistem. Evaluasi kehandalan ini akan memberikan gambaran mengenai kekuatan dari sistem yang dibuat, selain itu juga akan memberikan gambaran mengenai kemampuan dari sistem jika sistem ini diaplikasikan pada lingkungan penggunaan yang sebenarnya yaitu *LAN*, *Wireless LAN* dan jaringan Internet (*WAN*). Evaluasi kehandalan tersebut akan meliputi:

a. Waktu akses web

Adalah waktu yang dibutuhkan oleh pengguna (laptop) untuk membuka halaman antar muka web yang sudah tertanam dalam sistem.

b. Respon aktuator lampu

Menghitung waktu yang dibutuhkan oleh sistem untuk menjalankan perintah *on/off* dari lampu.

c. Respon pintu gerbang

Waktu yang dibutuhkan untuk merespon perintah terkait pintu gerbang.

d. Respon update nilai suhu

Waktu yang dibutuhkan ketika sistem meng-update nilai suhu.

e. Konsumsi daya dari sistem

f. Menguji kehandalan sistem selama 2x24 jam nonstop

Setiap parameter pengujian tersebut akan dilakukan pada 4 (empat) lingkungan jaringan, yaitu:

a. *Peer-to-peer* (Gambar 3.14)

b. *LAN* (Gambar 3.15)

c. *Wireless LAN* (Gambar 3.16)

d. *WAN (Internet)* (Gambar 3.17)

Dengan metode tersebut, akan dihasilkan gambaran menyeluruh terhadap kehandalan dari sistem yang dibuat.

## 4.3. Hasil Evaluasi

### 4.3.1. Evaluasi Uji Fungsi

Dari hasil evaluasi uji fungsi menunjukkan kinerja yang baik. Tabel berikut memberikan gambaran lebih detil mengenai hasil evaluasi uji fungsi.

**Tabel 4.3.** Hasil Evaluasi Uji Fungsi

No	Parameter Uji		Hasil Uji	Ket
<b>Uji Fungsi Lampu</b>				
1.	Kondisi Lampu	Lampu 1	Baik	-
		Lampu 2	Baik	-
		Lampu 3	Baik	-
2.	Fungsi <i>on/off</i>	Lampu 1	Baik	-
		Lampu 2	Baik	-
		Lampu 3	Baik	-
<b>Uji Fungsi Sensor Suhu</b>				
1.	Hasil indera sensor terbaca pada web		Baik	-
2.	Sensor merespon perubahan suhu		Baik	-
<b>Uji Fungsi Motor DC</b>				
1.	Kondisi Motor		Baik	-
2.	Fungsi arah putaran		Baik	-
3.	Fungsi Kecepatan		Baik	-
<b>Uji Fungsi Sistem</b>				
1.	Fungsi Jaringan		Baik	-
2.	Fungsi Web Server		Baik	-
3.	Fungsi Konfigurasi		Baik	-
4.	Fungsi Indikator Lampu		Baik	-
5.	Fungsi Pembacaan Suhu		Baik	-
6.	Informasi IP Address		Baik	-

### 4.3.2. Evaluasi Tingkat Akurasi

Dilakukan pengambilan sampel sebanyak 20 kali terhadap rentang suhu antara 32,2°C dan 42,0 °C. dari hasil pengukuran tersebut didapatkan perbedaan antara hasil pengukuran menggunakan *thermometer* standar dengan sensor suhu yang dibuat, didapatkan selisih maksimal 1,3°C dengan selisih rata-rata sebesar 0,3°C. angka tersebut menunjukkan tingkat akurasi yang cukup baik dengan akurasi rata-rata mencapai 99,1%. Tabel berikut menggambarkan lebih detail.

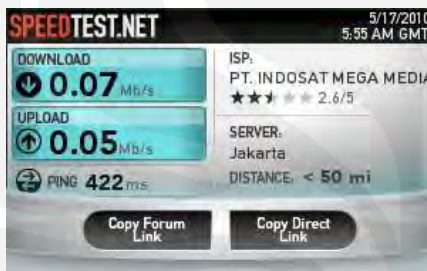
**Tabel 4.4.** Hasil Evaluasi Tingkat Akurasi Sensor Suhu

Set Point Percobaan	Sensor	Termometer Acuan	Selisih	Akurasi
1	34,8	35,4	0,6	98,3%
2	35,6	35,7	0,1	99,7%
3	33,0	34,3	1,3	96,2%
4	35,2	35,3	0,1	99,7%
5	35,2	35,3	0,1	99,7%
6	35,6	35,4	0,2	99,4%
7	35,5	35,6	0,1	99,7%
8	35,6	35,7	0,1	99,7%
9	35,3	35,8	0,5	98,6%
10	35,0	35,0	0	100,0%
11	35,2	35,4	0,2	99,4%
12	34,0	33,6	0,4	98,8%
13	34,2	33,7	0,5	98,5%
14	32,4	32,7	0,3	99,1%
15	31,9	32,2	0,3	99,1%
16	41,2	40,9	0,3	99,3%
17	41,5	42,0	0,5	99,8%
18	35,1	35,3	0,2	99,4%
19	34,7	34,8	0,1	99,7%
20	34,5	34,8	0,3	99,1%
<b>Rata-rata</b>			<b>0,3</b>	<b>99,1%</b>



### 4.3.3. Evaluasi Kehandalan

Evaluasi kehandalan berikut merupakan hasil evaluasi sistem pada 4 (empat) model jaringan yaitu *PTP (Peer-to-peer)*, *LAN*, *WiFi*, dan *WAN*. Evaluasi pada model jaringan *WAN* melibatkan kualitas dari layanan internet di kedua tempat dimana pada tempat pertama pengguna menggunakan layanan internet dari *IndosatM2 (Gambar 4.11.a)* dengan menggunakan *modem* bergerak, sedangkan sisi lain dimana web server berada digunakan internet dari *FastNet (Gambar 4.11.b)*.



(a)



(b)

**Gambar 4.11.** (a) Hasil tes kecepatan layanan internet *broadband Indosat M2*

(b) Hasil tes kecepatan layanan internet *Broadband FastNet* <sup>[9]</sup>

Dari hasil pengukuran didapatkan kecepatan akses terbaik menggunakan model jaringan *Peer-to-peer* dan konsumsi daya sangat hemat dengan hanya mencapai kurang dari 2 watt. **Tabel 4.5** berikut menggambarkan lebih detail.

**Tabel 4.5.** Hasil Evaluasi Kehandalan

Parameter	Model Jaringan Uji				Rata-rata
	PTP	LAN	WiFi	WAN	
Waktu akses Halaman Utama	2,5 det	2,7 det	3 det	8,4 det	4,15 det
Waktu akses Halaman Konfig	700 mdet	800 mdet	750 mdet	3,6 det	1,46 det
Waktu akses Halaman Informasi	500 mdet	600 mdet	1 det	3 det	1,27 det
Respon Lampu 1	500 mdet	500 mdet	500 mdet	1 det	0,6 det
Respon Lampu 2	500 mdet	500 mdet	500 mdet	1 det	0,6 det
Respon Lampu 3	500 mdet	500 mdet	500 mdet	1 det	0,6 det
Respon Pintu Gerbang	500 mdet	500 mdet	500 mdet	1 det	0,6 det
Respon Update Data Suhu	250 mdet	250 mdet	250 mdet	3 det	0,9 det
Konsumsi Daya saat booting	3,36 watt (12V x 280mA)				
Konsumsi Daya Normal	1,92 watt (12V x 160mA)				

## **BAB 5**

### **KESIMPULAN**

Dari hasil pembelajaran teori, perancangan, implementasi, dan evaluasi kinerja dari **Sistem Kontrol Berbasis *Embedded Web***, bisa disimpulkan beberapa hal sebagai berikut.

1. Sistem *embedded* menggunakan *microprocessor* Rabbit 4000 memiliki performa yang cukup baik untuk diaplikasikan sebagai *embedded web server*.
2. Hasil evaluasi menunjukkan kontrol on/off lampu, sensor suhu dan kontrol terhadap pintu gerbang dapat berfungsi dengan baik.
3. Dari sebanyak 20 (dua puluh) percobaan, tingkat akurasi dari sensor suhu menunjukkan angka yang baik, yaitu 99,1%
4. Dari hasil pengujian menunjukkan kehandalan yang baik dengan konsumsi daya sebesar 1,92watt dan untuk membuka antarmuka web hanya membutuhkan waktu 2,7 detik pada jaringan *LAN* dan 8,4 detik dari jaringan internet.
5. Halaman web yang dirancang saat ini menggunakan teknik *embedd* kedalam *flash memori* pada saat kompilasi, oleh karena itu untuk mengganti *design* dari halaman web akan sangat merepotkan karena melibatkan proses kompilasi dari awal. Untuk itu perlu adanya mekanisme yang mempermudah *upload* file halaman web sehingga akan mempermudah dalam memodifikasi halaman web sesuai dengan keinginan. Hal ini memungkinkan jika fasilitas FTP server bisa ditambahkan kedalam sistem ini, namun fitur tersebut tidak termasuk dalam pokok bahasan kali ini mengingat keterbatasan yang dimiliki.
6. Semoga kajian ini memberikan manfaat bagi pengembangan yang lebih lanjut dan mampu melengkapi aplikasi kajian sistem *embedded* lainnya.

## DAFTAR REFERENSI

- [1] Rabbit Semiconductor, *Datasheet Poster*. April 4, 2010.  
[http://www.rabbit.com/documentation/docs/rab40\\_ref\\_poster.pdf](http://www.rabbit.com/documentation/docs/rab40_ref_poster.pdf)
- [2] Rabbit Semiconductor, *Rabbit 4000 Pin Definition*. May 13, 2010.  
<http://rabbit.com/documentation/docs/manuals/Rabbit4000/UsersManual/29pins.htm#1011545>
- [3] Rabbit Semiconductor, *Rabbit 4000 Pinout*. May 13, 2010.  
<http://rabbit.com/documentation/docs/manuals/Rabbit4000/UsersManual/29pins.htm#992650>
- [4] Rabbit Semiconductor, *Rabbit Core RCM4300 User Manual*. April 4, 2010.  
<http://rabbit.com/documentation/docs/manuals/RCM4300/RC4300UM.pdf>
- [5] Rabbit Semiconductor, *Rabbit Core RCM4300 Schematic*. April 4, 2010.  
<HTTP://WWW.RABBIT.COM/DOCUMENTATION/SCHEMAT/090-0229.PDF>
- [6] Rabbit Semiconductor, *Rabbit USB Programming Cable*. April 4, 2010.  
<http://www.rabbit.com/products/usb-prg-cable/largeView.shtml>
- [7] Rabbit Semiconductor, *RCM4300 Rabbit Core Image*. April 4, 2010.  
<http://www.rabbit.com/products/rcm4300/largeView.shtml>
- [8] Rabbit Semiconductor, *Rabbit 4000 Microprocessor*. Juni 30, 2010.  
<http://www.rabbit.com/products/rab4000/largeView.shtml>
- [9] The Global Broadband Speed Test, Speed Test. June 13, 2010. <http://speedtest.net/>
- [10] Topbits Tech Community. *Relay*. June 20, 2010. <http://www.topbits.com/relay.html>
- [11] Wikipedia Free Encyclopedia. *Thermistor*. June 20, 2010.  
<http://en.wikipedia.org/wiki/Thermistor>

## DAFTAR PUSTAKA

Andrew S. Tanenbaum. 1997. **“Jaringan Komputer”**, Edisi Bahasa Indonesia, Jilid 2, Terj. Gurnita Priatna. Indonesia: Percetakan Andi Offset Printer.

Computer Desktop Encyclopedia, **“Internet Protocol”**  
<http://www.yourdictionary.com/computer/internet-protocol>

Edhy Sutanta. 2005. **“Komunikasi Data & Jaringan Komputer”**. Yogyakarta: Graha Ilmu

International Organization for Standardization, **“Open Systems Interconnections”**  
[http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269\\_ISO\\_IEC\\_7498-1\\_1994\(E\).zip](http://standards.iso.org/ittf/PubliclyAvailableStandards/s020269_ISO_IEC_7498-1_1994(E).zip)

Internet Assigned Numbers Authority, **“PORT NUMBERS”**  
<http://www.iana.org/assignments/port-numbers>

Niall Mansfield. 2004. **“Practical TCP/IP”**, Jilid 1, Terj. Dwi Prabantini. Yogyakarta: Andi

Niall Mansfield. 2004. **“Practical TCP/IP”**, Jilid 2, Terj. Dwi Prabantini. Yogyakarta: Andi

Rabbit Semiconductor, **“Datasheet Poster”**  
[http://www.rabbit.com/documentation/docs/rab40\\_ref\\_poster.pdf](http://www.rabbit.com/documentation/docs/rab40_ref_poster.pdf)

Rabbit Semiconductor, **“Rabbit 4000 Pin Definition”**  
<http://rabbit.com/documentation/docs/manuals/Rabbit4000/UsersManual/29pins.htm#1011545>

Rabbit Semiconductor, **“Rabbit 4000 Pinout”**  
<http://rabbit.com/documentation/docs/manuals/Rabbit4000/UsersManual/29pins.htm#992650>

Rabbit Semiconductor, **“Rabbit Core RCM4300 User Manual”**  
<http://rabbit.com/documentation/docs/manuals/RCM4300/RC4300UM.pdf>.

Wikipedia Indonesia, **“Hypertext markup language”**  
<http://id.wikipedia.org/wiki/HTML>

Wikipedia Indonesia, **“Model DARPA”**  
[http://id.wikipedia.org/wiki/Model\\_DARPA](http://id.wikipedia.org/wiki/Model_DARPA)

Wikipedia Indonesia, **“Model OSI”**  
[http://id.wikipedia.org/wiki/Model\\_OSI](http://id.wikipedia.org/wiki/Model_OSI)