

Sistem Kontrol Perangkat Elektronik Semi Digital

Tugas Akhir

Disampaikan Sebagai Bagian Dari Persyaratan Kelulusan Diploma 3
Program Studi Teknik Komputer

Oleh :

13314003 Aprilonita Simanjuntak
13314006 Welly Mikha Simanjuntak
13314012 Ucok Roles Situmorang



**Institut Teknologi Del
2016/2017**

Halaman ini sengaja dikosongkan



**Lembar Pengesahan Tugas Akhir
Institut Teknologi Del**

Sistem Kontrol Perangkat Elektronik Semi Digital

Oleh:

13314003 Aprilonita Simanjuntak
13314006 Welly Mikha Simanjuntak
13314012 Ucok Roles Situmorang

Sitoluama, 4 September 2017

Pembimbing



Marojahan Mula Timbul Sigirow, ST, M.Sc.
NIDN. 0108098301

**Dinyatakan memenuhi syarat dan karenanya disetujui dan disahkan sebagai
Laporan Tugas Akhir Diploma 3
Program Studi /Teknik Komputer/
Institut Teknologi Del**

Prakata

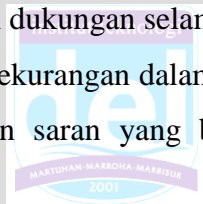
Puji dan syukur kepada Tuhan Yang Maha Esa atas rahmat yang diberikan kepada penulis sehingga Tugas Akhir ini dapat diselesaikan dengan baik.

Laporan Tugas Akhir ini bertujuan untuk memberikan informasi bagi pembaca mengenai implementasi *Sistem Kontrol Perangkat Elektronik Semi Digital*. Laporan Tugas Akhir ini merupakan bagian dari syarat kelulusan Diploma III Institut Teknologi Del.

Penulis menyampaikan terima kasih kepada Bapak Marojahan Mula Timbul Sigiroy sebagai dosen pembimbing yang telah memberikan saran, bimbingan dan arahan kepada penulis selama pengerjaan Tugas Akhir ini. Penulis juga mengucapkan terimakasih kepada Bapak Olmes Yosefa Hutajulu, Bapak Maisevli Harika dan Bapak Indra Hartarto sebagai dosen penguji Tugas Akhir ini, dan kepada koordinator Tugas Akhir atas saran yang diberikan dalam pengerjaan Tugas Akhir ini.

Penulis juga mengucapkan terimakasih kepada orangtua, teman sejawat dan semua pihak yang telah membantu dan memberikan dukungan selama pengerjaan Tugas Akhir ini.

Penulis menyadari bahwa masih ada kekurangan dalam laporan Tugas Akhir ini. Oleh karena itu, penulis mengharapkan kritik dan saran yang bersifat membangun untuk perbaikan laporan ini di masa mendatang.



Sitoluama, Mei 2017

Aprilonita Simanjuntak

Welly Mikha Simanjuntak

Ucok Roles Situmorang

Abstrak

Perangkat elektronik semi digital memiliki beberapa komponen yang memiliki umur pemakaian dan membutuhkan perawatan secara berkala. Perangkat elektronik seperti lampu listrik, LCD Proyektor, dan AC. Komponen tersebut akan berkurang performanya seiring dengan lama waktu pemakaian dari alat tersebut.

Orang-orang menggunakan teknologi pada umumnya untuk membantu dan memudahkan segala aktivitasnya. Namun dalam penggunaan peralatan elektronik tersebut, manusia kurang bisa mengontrol peralatan elektronik atau sering mengabaikan penggunaannya.

Sistem pengontrolan perangkat elektronik semi digital ini bertujuan untuk mempersingkat waktu pengecekan status dari perangkat tersebut. Sistem pengontrolan perangkat semi digital ini bekerja dengan bantuan jaringan LAN yang setiap datanya akan dikirim ke sebuah *server*. Pengguna akan dapat secara langsung melakukan pengecekan ke perangkat elektronik semi digital dengan mengakses *server*. Dalam hal ini, maksud dari pengecekan yang dilakukan adalah melihat status dari perangkat elektronik semi digital dalam keadaan *on* atau *off*.

Pada Tugas Akhir ini, penulis membangun “Sistem Kontrol Perangkat Elektronik Semi Digital” yang terdiri dari proyektor, lampu, **Arduino Uno**, *Ethernet Shield*, *RS232 Shield*, *server*, kabel LAN, kabel RS232, dan sebuah aplikasi web. Sistem ini diharapkan mampu mempermudah proses pengontrolan proyektor di lingkungan IT Del.

Kata Kunci : Mikrokontroler, sistem kontrol, Aplikasi Web, dan Server.

DAFTAR ISI

Prakata	3
Abstrak	4
Bab I Pendahuluan	9
1.1 Latar Belakang.....	9
1.2 Tujuan.....	10
1.3 Lingkup	10
1.4 Pendekatan.....	11
1.5 Sistematika Penyajian.....	11
Bab II Tinjauan Pustaka	12
2.1 Sistem Tertanam.....	12
2.2 Mikrokontroler	12
2.3 Perangkat Keras (<i>Hardware</i>).....	12
2.3.1 Arduino Uno	12
2.3.2 Serial <i>Port</i> RS232	16
2.3.3 Ethernet Shield.....	18
2.3.4 Relay	18
2.4 Kajian Pustaka	19
Bab III Analisis dan Perancangan.....	22
3.1 Analisis Sistem	22
3.1.1 Sistem <i>Overview</i>	22
3.1.2 Deskripsi Umum Sistem	23
3.2 Analisis Kebutuhan	23
3.2.1 Analisis Kebutuhan Pengguna	23
3.2.2 Analisis Kebutuhan Sistem	24
3.3.2.2 Analisis Kebutuhan Perangkat Lunak (<i>Software</i>)	24
3.3 Analisis Projector yang akan dikontrol.....	25
3.3.1 <i>Command</i> yang digunakan untuk mengontrol LCD Proyektor NEC.....	26
3.4 Perancangan Sistem.....	26
3.5 Perancangan Aplikasi Berbasis Web	28
3.6 Perancangan <i>Database</i>	29
3.7 <i>Flowchart</i>	30
Bab IV Implementasi dan Pengujian	31
4.1 Implementasi Rangkaian Mikrokontroler	31
4.1.1 Rangkaian Mikrokontroler dengan <i>Ethernet Shield</i>	31
4.1.2 Rangkaian Mikrokontroler dengan RS232 Shield	31
4.1.3 Rangkaian Mikrokontroler dengan LCD Proyektor	32
4.1.4 Rangkaian Mikrokontroler dengan <i>Relay</i>	32
4.1.5 Rangkaian <i>Relay</i> Untuk Kontrol Lampu	33
4.2 Implementasi Aplikasi Berbasis <i>Web</i> pada <i>Server</i>	33
4.2.1 Implementasi Aplikasi Pengontrolan menggunakan Framework Yii2.....	33
4.3 Instalasi	35
4.3.1 Instalasi CentOS 7	35
4.3.2 Instalasi Web Server	35
4.3.3 Instalasi Yii Framework	35
4.4 Konfigurasi	36
4.4.1 Konfigurasi DNS Server	36
4.4.2 Konfigurasi Httpd	37
4.5 Pengujian	37
4.5.1 Proses <i>on/off</i> Lampu 1.....	38
4.5.2 Proses <i>on/off</i> Lampu 2	39
4.5.3 Proses <i>on/off</i> semua lampu.....	40
4.5.3 Proses <i>on/off</i> LCD Proyektor via RS232.....	41
4.5.4 Proses <i>on/off</i> LCD Proyektor via LAN	42
4.5.5 Pengujian Kabel serial RS232 pada Switch	42
Bab V Kesimpulan dan Saran.....	44
5.1 Kesimpulan	44
5.2 Saran	44

Daftar Pustaka	45
Lampiran A Kode Program Pada Mikrokontroler	47
Pengujian Pengontrolan Proyektor	52



Daftar Tabel

Tabel 1 Indeks Board Arduino	13
Tabel 2 Fungsi Pin-pin Konektor RS232 yang akan digunakan	17
Tabel 3 Perangkat Keras yang dibutuhkan	24
Tabel 4 Pengaturan Komunikasi Perangkat	26
Tabel 5 Koneksi Pin Relay	32



Daftar Gambar

Gambar 1 Arduino Uno [3].....	13
Gambar 2 Serial Port RS232 [4].....	16
Gambar 3 RS232 Pinout(9 Pin) [5].....	17
Gambar 4 Ethernet Shield [6].....	18
Gambar 5 Relay 2 module [7]	19
Gambar 6 <i>High Level</i> Sistem Overview	22
Gambar 7 Sistem Overview Sistem	23
Gambar 8 Desain Skematik Mikrokontroler dengan Relay	27
Gambar 9 Desain Skematik Mikrokontroler dengan RS232 <i>Shield</i>	27
Gambar 11 Desain antarmuka aplikasi	28
Gambar 12 <i>Database</i>	29
Gambar 13 Flowchart	30
Gambar 14 Rangkaian Mikrokontroler dengan Ethernet Shield.....	31
Gambar 15 Rangkaian Mikrokontroler dengan RS232 <i>shield</i>	32
Gambar 16 Rangkaian Mikrokontroler dengan LCD Proyektor.....	32
Gambar 17 Rangkaian Mikrokontroler dengan Relay	32
Gambar 18 Rangkaian <i>relay</i> untuk kontrol lampu.....	33
Gambar 19 Modul Sistem.....	34
Gambar 20 Sistem Kontrol Device	34
Gambar 21 <i>Timer</i> pengontrolan <i>device</i>	34
Gambar 22 <i>Set</i> Domain pada <i>resolv.conf</i>	36
Gambar 23 Konfigurasi zone untuk domain	36
Gambar 24 Konfigurasi <i>file zone</i>	37
Gambar 25 Konfigurasi <i>httpd</i>	37
Gambar 26 Lampu Dalam Keadaan Mati	38
Gambar 27 Lampu Mati	38
Gambar 28 Lampu 1 dalam keadaan Hidup	39
Gambar 29 Lampu 1 Hidup	39
Gambar 30 Lampu 2 dalam keadaan Hidup	39
Gambar 31 Lampu 2 Hidup	40
Gambar 32 Semua Lampu dalam keadaan Hidup.....	40
Gambar 33 Lampu 1 dan 2 Hidup	41
Gambar 34 Pengontrolan LCD Proyektor	41
Gambar 35 Pengontrolan LCD proyektor <i>via</i> LAN.....	42
Gambar 36 Konfigurasi Switch <i>via</i> serial port RS232.....	43
Gambar 37 Control on/off Projector via Hype!Terminal.....	52
Gambar 38 Control on/off Projector via 232Analyzer.....	52

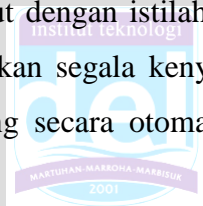
Bab I

Pendahuluan

Pada bab ini menjelaskan mengenai latar belakang, tujuan pemilihan topik, lingkup, pendekatan yang dilakukan, dan sistematika penyajiannya.

1.1 Latar Belakang

Dengan berkembangnya teknologi serta tingkat mobilitas manusia dituntut untuk dapat melakukan berbagai aktivitas dalam durasi waktu yang relatif singkat. Hal ini dapat dilihat dengan perkembangan jaringan Internet yang telah mencapai tingkat yang sangat membanggakan dan mengesankan karena hampir semua informasi dapat dilihat. Namun ada hal yang lebih membanggakan selain memanfaatkan teknologi Internet sebagai pusat informasi, dimana teknologi Internet juga bisa dimanfaatkan untuk mengendalikan peralatan elektronik dari jarak jauh yang disebut dengan istilah *smart home*, yang merupakan sebuah sistem komputer yang akan memberikan segala kenyamanan, keselamatan, keamanan, dan penghematan energi yang berlangsung secara otomatis dan terprogram melalui komputer pada sebuah gedung atau rumah.



Istilah *smart home* bisa juga disebut bagian dari *smart device*. *Smart device* merupakan sebuah perangkat yang bisa terhubung ke jaringan dan dapat dikontrol menggunakan jaringan Internet.

Smart device dapat digunakan untuk mengontrol hampir semua perlengkapan peralatan elektronik yang masih *semi digital*. Perangkat elektronik *semi digital* merupakan perangkat yang bisa dikontrol menggunakan komputer dengan serial port RS232 atau dengan port lainnya, dan belum bisa terhubung ke *network* secara langsung.

Penggambaran dari sistem yang akan diimplementasikan ini berupa, *device* yang akan dikontrol akan dipantau dari tempat tertentu dan setiap *device* dapat dikontrol dari tempat tersebut, pengendalian seperti ini memungkinkan setiap *device* dapat dinyalakan dan dimatikan secara terpusat sekaligus memantau penggunaan *device* tersebut, seperti status *on/off*.

Demikian juga halnya dengan pengontrolan lampu. Terdapat beberapa masalah yang timbul akibat penggunaan lampu listrik, salah satunya yaitu adalah pemborosan listrik ketika seseorang lupa mematikan lampu listrik ketika meninggalkan ruangan, yang berdampak pada pemborosan energi listrik dan biaya listrik yang harus dikeluarkan. Selain itu kelalaian dalam mengontrol peralatan listrik yang ada dapat berakibat pada terjadinya bencana seperti kebakaran. Oleh karena itu perangkat elektronik semi digital yang seolah-olah menjadi *smart device* yang dapat dikontrol dari jarak jauh dan menggunakan aplikasi *web*.

Dalam hal ini *smart device* menerapkan konsep IoT (*Internet of Things*), dimana sistemnya terintegrasi dan dapat diakses dengan Internet. Dengan semakin berkembangnya infrastruktur Internet, maka kita menuju babak berikutnya, dimana bukan hanya *smartphone* atau komputer saja yang dapat terkoneksi dengan Internet. Contohnya dapat berupa: mesin produksi, mobil, peralatan elektronik seperti LCD Proyektor, lampu, AC dan termasuk benda nyata apa saja yang semuanya tersambung ke jaringan lokal dan global menggunakan sensor atau aktuator yang tertanam.



1.2 Tujuan

Tujuan umum dari pengerjaan tugas akhir ini adalah dapat membuat sebuah sistem yang dapat mengontrol perangkat elektronik *semi digital* sehingga perangkat-perangkat tersebut memiliki perilaku seolah-olah sebuah *smart device*.

Dalam mencapai tujuan tersebut ada beberapa syarat yang harus dapat dipenuhi terlebih dahulu, antara lain :

1. Perancangan dan pembangunan sebuah perangkat *embedded* yang akan berperan sebagai antarmuka antara perangkat elektronik semi digital maupun analog dengan *server*.
2. Perancangan dan pembangunan aplikasi *server* yang berperan sebagai *controller* yang akan mengatur perilaku perangkat-perangkat yang terhubung ke sistem.

1.3 Lingkup

Lingkup kajian dari pengerjaan Tugas Akhir ini, yaitu:

1. Sistem yang dibangun pada Tugas Akhir ini merupakan sebuah prototipe yang hanya mengontrol sebuah *device* dan lampu secara terpusat.
2. Perilaku perangkat sebagai sebuah *smart device* masih pada sebatas *on/off* dan *timer*.
3. Perangkat analog yang akan dikontrol harus terlebih dahulu memiliki perangkat *interface* seperti LAN (*Local Area Network*).

1.4 Pendekatan

Pendekatan yang dilakukan pada Tugas Akhir ini terdiri dari:

1. Mempelajari literatur

Melakukan *study* literatur mengenai sistem kontrol perangkat elektronik semi digital dengan membaca dan memahami beberapa *paper* tentang sistem pengontrolan proyektor menggunakan mikrokontroler.

2. Membuat Dokumentasi

Menuliskan laporan dan dokumentasi selama pengerjaan Tugas Akhir.

3. Membuat gambaran *system overview*

Menggambarkan cara kerja *system overview pengontrolan perangkat elektronik semi digital*.

4. Merancang dan membangun perangkat dan aplikasi

Merancang dan membangun perangkat dan aplikasi berdasarkan gambaran *system overview* yang telah dirancang atau dibuat.

1.5 Sistematika Penyajian

Dokumen Laporan Tugas Akhir ini terdiri dari 5 bab, yaitu:

Bab 1 Pendahuluan

Bab ini berisi tentang latar belakang tujuan, lingkup kajian, pendekatan yang dilakukan, dan sistematika penyajian dalam laporan.

Bab 2 Tinjauan Pustaka

Pada bab ini dijelaskan tentang tinjauan pustaka yang mencakup penjelasan mengenai *Sistem kontrol Perangkat Elektronik Semi Digital*, serta kesimpulan dari tinjauan pustaka tersebut.

Bab 3 Perancangan dan Pembangunan

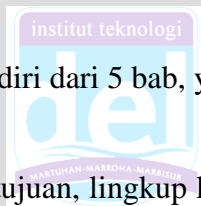
Pada bab ini dijelaskan tentang rancangan dan pembangunan sistem yang akan dibangun.

Bab 4 Implementasi dan Testing

Pada bab ini dijelaskan mengenai implementasi dan *testing* sistem yang telah dibangun.

Bab 5 Kesimpulan dan Saran

Pada bab ini dijelaskan tentang kesimpulan dan saran dari hasil yang dicapai selama pengerjaan Tugas Akhir.



Bab II

Tinjauan Pustaka

Pada bab ini dijelaskan tinjauan pustaka yang digunakan sebagai dasar teori untuk mengerjakan Tugas Akhir.

2.1 Sistem Tertanam

Sistem tertanam merupakan *computing device* yang didesain dengan tujuan tertentu secara spesifik untuk melakukan fungsi tertentu. Sistem tertanam terdiri dari perangkat keras dan perangkat lunak. Perangkat keras terdiri dari mikroprosesor atau mikrokontroler dengan penambahan memori eksternal, I/O dan komponen lainnya. Perangkat lunak sistem tertanam merupakan penggerak pada sistem. Perangkat lunak sistem tertanam biasanya disebut firmware karena perangkat lunak tipe ini dimuat ke ROM (*Read Only Memori*), EPROM (*Erasable Programmable Read Only Memori*) atau memory Flash. Sekali program dimasukkan kedalam perangkat keras maka tidak akan pernah berubah kecuali diprogram ulang.[1]



2.2 Mikrokontroler

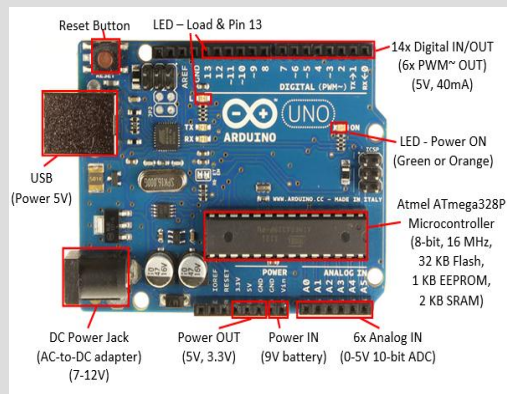
Mikrokontroler adalah suatu chip berupa IC (*Integrated Circuit*) yang dapat menerima sinyal input, mengolahnya dan memberikan sinyal output sesuai dengan program yang diisikan ke dalamnya. Sinyal input mikrokontroler berasal dari sensor yang merupakan informasi dari lingkungan sedangkan sinyal output ditujukan kepada actuator yang dapat memberikan efek ke lingkungan. Mikrokontroler dapat diibaratkan sebagai otak dari suatu perangkat atau produk yang mampu berinteraksi dengan lingkungan sekitarnya. [2]

2.3 Perangkat Keras (*Hardware*)

Pada bab ini akan dijelaskan mengenai tinjauan pustaka yang mendukung perancangan dan pembangunan sistem yang dibangun.

2.3.1 Arduino Uno

Arduino yang digunakan dalam tugas akhir ini yaitu **Arduino Uno**. Arduino yang digunakan dapat dilihat pada Gambar 1 di bawah.



Gambar 1 Arduino Uno [3]

Arduino Uno adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328 (*datasheet*).

2.3.1.1 Overview

Arduino Uno mempunyai 14 pin digital *input/output* (6 diantaranya dapat digunakan sebagai *output* PWM), 6 input analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah *power jack*, sebuah *ICSP header*, dan sebuah tombol *reset*. *Arduino Uno* memuat semua yang dibutuhkan untuk menunjang mikrokontroler, mudah menghubungkannya ke sebuah komputer dengan sebuah kabel USB (*Universal Serial Bus*) atau mensuplainya dengan sebuah *adaptor* AC ke DC atau menggunakan baterai untuk memulainya.

Tabel 1 Indeks Board Arduino

Bagian Arduino UNO	Spesifikasi
Microcontroller	Atmega328
Operating Voltage	5V
Input voltage	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog input Pins	6
DC Current per I/O Pin	40 mA

DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25

2.3.1.2 Daya (Power)

Arduino Uno dapat disuplai melalui koneksi USB (*Universal Serial Bus*) atau dengan sebuah *power suplai eksternal*. Sumber daya dipilih secara otomatis. *Board Arduino Uno* dapat beroperasi pada sebuah suplai eksternal 6 samapi 20 Volt. Kisaran yang direkomendasikan adalah 7 sampai 12 Volt. Pin-pin dayanya adalah sebagai berikut:

1. Vin

Tegangan input ke *Arduino board* ketika *board* sedang menggunakan sumber suplai eksternal (seperti 5 Volt dari koneksi USB atau sumber tenaga lainnya yang diatur) dengan cara menggunakan *power jack*.

2. 5V

Pin *output* ini merupakan tegangan 5 Volt yang diatur dari regulator pada *board*. *Board* dapat disuplai dengan salah satu suplai dari DC *power jack* (7-12V), USB *connector* (5V), atau pin Vin dari board (7-12). Penyuplaian tegangan melalui pin 5V atau 3,3V *membypass* regulator, dan dapat membahayakan *board*. Hal ini tidak dianjurkan.

3. 3V3

Sebuah suplai 3,3 Volt dihasilkan oleh regulator pada *board*. Arus maksimum yang dapat dilalui adalah 50 mA.

4. GND

Pin *Ground* berfungsi sebagai jalur *ground* pada **Arduino Uno**.

5. IOREF ()

Pin ini di papan Arduino memberikan tegangan referensi dengan yang mikrokontroler beroperasi. Sebuah perisai dikonfigurasi dengan baner dapat membaca pin tegangan IOREF dan pilih sumber daya yang tepat atau mengaktifkan penerjemah tegangan pada output untuk bekerja dengan 5V atau 3,3V.

2.3.1.3 Memori

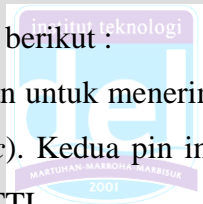
Atmega328 memiliki 32 KB (dengan 0,5 KB digunakan untuk *bootloader*), 2KB dari SRAM dan 1KB EEPROM

2.3.1.4 Input dan Output

Setiap 14 pin digital pada **Arduino Uno** dapat digunakan sebagai *input* dan *output*, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. I/O dioperasikan pada 5V.

Beberapa pin memiliki fungsi sebagai berikut :

Serial : 0 (RX) dan 1 (TX). Digunakan untuk menerima (RX) dan memancarkan (TX) serial data TTL (*Transistor-Transistor Logic*). Kedua pin ini dihubungkan ke pin-pin yang sesuai dari *chip* Serial Atmega8U2 USB-ke-TTL



1. **External Interrupts :** 2 dan 3. Pin-pin ini dapat dikonfigurasi untuk dipicu sebuah *interrupt* (gangguan) pada sebuah nilai rendah, suatu kenaikan atau penurunan yang besar, atau suatu perubahan nilai.
2. **PWM:** Pin 3, 5, 6, 9, 10, dan 11. Memberikan 8-bit PWM *output* dengan fungsi `analogWrite()`.
3. **SPI:** Pin 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Pin-pin ini mendukung komunikasi SPI menggunakan *SPI library*.
4. **LED :** Pin 13. Ada sebuah LED yang terpasang, terhubung ke pin digital 13. Ketika pin bernilai HIGH LED menyala, ketika pin bernilai LOW LED mati.

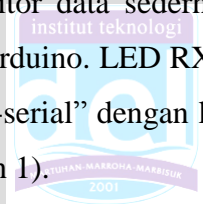
Arduino Uno mempunyai 6 *input* analog, diberi label A0 sampai A5, setiapnya memberikan 10 bit resolusi. Secara *default*, 6 input analog tersebut mengukur dari *ground* sampai tegangan 5 Volt, dengan itu mungkin untuk mengganti batas atas dari kisaran dengan menggunakan pin AREF dan fungsi `analogReference()`.

Disisi lain, beberapa pin mempunyai fungsi spesial :

1. **TWI:** pin A4 atau SDA dan pin A5 atau SCL. Mendukung komunikasi TWI dengan menggunakan Wire library.
2. **AREF.** Referensi tegangan untuk analog. Digunakan dengan `analogReference()`.
3. **Reset.** Membawa saluran ini LOW untuk me-reset mikrokontroler. Secara khusus digunakan untuk menambahkan sebuah tombol *reset* untuk melindungi yang mem-sesuatu pada *board*.

2.3.1.5 Komunikasi

Arduino Uno mempunyai sejumlah fasilitas untuk komunikasi dengan sebuah komputer, Arduino lainnya atau mikrokontroler lainnya. **ATMega328** menyediakan serial komunikasi UART TTL(5V), yang tersedia pada pin digital 0 (RX) dan 1 (TX). ATMega16U2 pada *board* menghubungkan komunikasi serial melalui USB dan sebagai *port virtual com* untuk perangkat lunak pada komputer. *Firmware*'8 U2 menggunakan *driver* USB standar COM. Namun, pada **Windows** diperlukan, sebuah *file* inf. Perangkat lunak Arduino terdapat monitor serial yang digunakan untuk memonitor data sederhana yang dikirim ke atau dari *board* Arduino. LED RX dan TX di *board* Arduino. LED RX dan TX di *board* berkedip ketika data sedang dikirim melalui *chip* "USB-to-serial" dengan koneksi USB ke komputer (tetapi tidak untuk komunikasi serial pada pin 0 dan 1).



2.3.2 Serial Port RS232

RS232 adalah standar komunikasi serial antar peripheral-peripheral dalam komputer yang dikembangkan oleh *Electronic Industries Association* (EIA). Penggunaan yang paling umum adalah antar komputer dengan modem atau komputer dengan komputer.



Gambar 2 Serial Port RS232 [4]

2.3.2.1 Konektor RS232

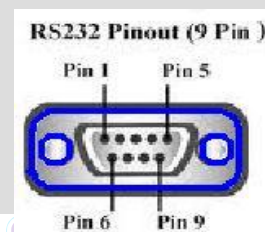
Standar ini menggunakan piranti dalam implementasinya. Piranti yang paling umum dipakai adalah *plug* DB-9 atau DB-25. RS232 dengan DB-9 biasanya dipakai untuk serial *port* komputer pribadi, dipakai untuk modem. Sedangkan konektor DB25 dipakai untuk *joystick*

game. Serial port RS232 dengan konektor DB9 memiliki 9 buah pin dan pada konektor DB25 memiliki pin 25 buah.

Fungsi masing-masing pin yang akan digunakan dalam pengerjaan tugas akhir ini pada konektor dapat dilihat pada tabel berikut :

Tabel 2 Fungsi Pin-pin Konektor RS232 yang akan digunakan

Pin DB25	PinDB9	Singkatan	Keterangan
Pin 2	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 5	Pin 8	CTS	Clear To Send



Gambar 3 RS232 Pinout(9 Pin) [5]

Dengan memanfaatkan pin konektor RS232 dapat digunakan untuk mengkoneksikan secara data serial antara komputer dan modem atau piranti lain, di industri koneksi ini digunakan untuk pengujian atau pengetesan.

Fungsi pin berdasarkan tabel diatas dapat dijelaskan sebagai berikut :

1. **Transmit Data(TX)** berfungsi sebagai saluran keluarnya data dari UART atau sebagai pengirim data ke *device* secara serial.
2. **Receive Data(RX)** berfungsi sebagai saluran masuknya data ke UART atau sebagai penerima data dari *device* secara serial.
3. **Clear To Send(CTS)** berfungsi sebagai penerima jawaban atas pengiriman isyarat RTS bila modem/piranti telah menerima data.

2.3.2.2 Komunikasi Serial RS232

Dalam komunikasi data RS232 terdapat hubungan dua perangkat yaitu menghubungkan DTE (*Data Terminal Equipment*) ke DCE (*Data Communication Equipment*). Protokol standar yang digunakan untuk mengatur komunikasi data secara serial disebut RS232 yaitu berupa standar yang dikembangkan EIA (*Electronic Industries Association*). Prinsip komunikasinya

adalah komunikasi *asynchronous*, dan sinyal *clock* pada komunikasi ini tidak disertakan pada *frame data*. Untuk melakukan sinkronisasi maka setiap kali pengiriman data disertakan sebuah *start bit* dan sebuah *stop bit*. *Frame data* yang dikirimkan disusun dengan urutan *start bit*, diikuti *bit-bit data* dan diakhiri dengan *stop bit*.

Pada protokol RS232 perlu dilakukan *setting* agar *port* komunikasi dapat saling dihubungkan yaitu *Baud Rate*.

2.3.3 Ethernet Shield

Ethernet Shield menambah kemampuan *Arduino board* untuk dapat terhubung ke jaringan Internet secara luas, dengan adanya *Ethernet Shield* memungkinkan perangkat **Arduino Uno** untuk dapat mengirimkan atau menerima data dari mana saja dengan menggunakan koneksi Internet.

Ethernet Shield umumnya berbasiskan chip *Ethernet Wiznet W5100*. Pada *Ethernet Shield* terdapat sebuah *slot micro-SD*, yang dapat digunakan untuk menyimpan *file library* tambahan untuk *Arduino*. *Onboard micro-SD card reader* diakses dengan menggunakan *SD library*. *Arduino board* berkomunikasi dengan *W5100* dan *SD card* menggunakan bus *SPI (Serial Peripheral Interface)*, komunikasi ini diatur oleh *library SPI.h* dan *Ethernet.h*.

Bus *SPI* menggunakan pin digital 11, 12 dan 13 pada *Arduino*, pin digital 10 digunakan untuk memilih *W5100* dan pin digital 4 digunakan untuk memilih *SD card*. Pin-pin yang sudah disebutkan sebelumnya tidak dapat digunakan untuk *input/output* umum ketika menggunakan *Ethernet shield*. *Arduino Ethernet Shield* dapat pada gambar berikut ini.



Gambar 4 Ethernet Shield [6]

2.3.4 Relay

Peran dan fungsi *relay* adalah sebagai saklar otomatis untuk menghidupkan dan mematikan lampu, dimana *relay* bekerja sesuai dengan perintah yang *arduino* berikan. *Relay* akan bekerja apabila memperoleh input logika *High* dari *Arduino* begitu sebaliknya *relay* akan berhenti bekerja apabila memperoleh *input* logika *low* dari *Arduino*.



Gambar 5 Relay 2 module [7]

2.4 Kajian Pustaka

Pada subbab ini akan dibahas mengenai jurnal terkait dengan sistem yang akan dibangun. Berikut jurnal-jurnal yang terkait dengan sistem yang akan dibangun.

1. Sistem Pengontrolan alat elektronik dengan media komputer dan mikrokontroler AT89C5 melalui media komunikasi Serial Port RS232 [8]:

Sistem komunikasi komputer dan mikrokontroler dapat menghasilkan suatu pengontrolan sistem yang beraneka ragam, baik sistem yang sederhana maupun sistem yang lebih kompleks. Dalam hal ini komputer berfungsi sebagai pusat pengendalian dan bekerja dengan memberikan perintah kepada mikrokontroler untuk menerjemahkan perintah tersebut menjadi sebuah eksekusi dengan cara mengirimkan sebuah data.

Mikrokontroler sendiri berfungsi untuk mengolah segala perintah yang harus dilaksanakan, artinya mikrokontroler akan bekerja jika menerima sinyal dari komputer ataupun dari objek yang dikontrol.

Pesan dari sinyal yang sudah diolah secara sederhana ditampilkan dalam media LCD atau unit pengendali ini terhubung secara langsung dengan media PC akan semakin lengkap sistem pengaturannya.

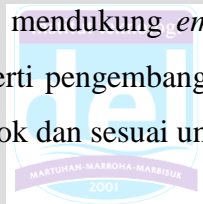
2. Simulasi Pengembangan Pengendali LCD Proyektor Berbasis Android menggunakan Mikrokontroler Arduino di Universitas Kanjuruhan Malang [9]:

LCD Proyektor di Universitas Kanjuruhan Malang khususnya dalam ruangan kelas termanfaatkan dengan baik akan tetapi bisa saja menimbulkan dampak buruk, misalnya LCD Proyektor bisa rusak dikarenakan dibiarkan dan tidak dimatikan terlebih dahulu ketika dosen ataupun mahasiswa meninggalkan kelas, keterbatasan *remote* yang telah ditentukan dari pihak gudang serta menggunakan tangan ataupun media lainnya untuk menyalakan maupun mengatur hal lain. Padahal ketika LCD Proyektor telah rusak, Kegiatan Belajar Mengajar (KBM) akan terbengkalai yang akibatnya dapat mengganggu kelas lain karena masih mencari kelas yang LCD Proyektornya hidup padahal di kelas tersebut nantinya ada jam kuliah, bahkan juga

akan mengeluarkan biaya lagi untuk membeli. Berdasarkan ulasan tersebut, bertujuan untuk membangun pengendali LCD Proyektor berbasis android yang menggunakan mikrokontroler Arduino yang dapat diterapkan dan untuk menggantikan penggunaan *remote* yang asli.

3. Pengembangan Sistem kendali dan Akuisisi Jarak jauh Perangkat Elektronik Berbasis Jaringan IP [10]:

Perkembangan teknologi *embedded system* telah membuat ukuran dari peralatan digital semakin kecil namun tetap handal, begitu pula dengan kepopuleran protokol TCP/IP yang mampu membuat proses komunikasi dan pertukaran informasi menjadi hal yang sangat mudah untuk dilakukan. Hingga kini teknologi tersebut telah membentuk sebuah jaringan Internet global yang mampu menghubungkan berbagai jenis peralatan dengan berbagai jenis teknologi dalam skala yang sangat luas. Penggabungan antara *embedded system* dengan jaringan berbasis TCP/IP akan membawa beberapa keuntungan, seperti kepraktisan dan konektivitas yang tinggi. Berbagai pengembangan yang mendukung *embedded ethernet* telah dilakukan baik dari segi perangkat lunak, seperti pengembangan kode TCP/IP stack, pengembangan perangkat keras yang lebih cocok dan sesuai untuk *embedded system*.



4. Perancangan Sistem Monitoring LCD Proyektor dan Komputer secara Terpusat [11]:

Perkembangan perangkat-perangkat seperti komputer dan proyektor sangat membantu manusia dalam menjalankan aktivitas keseharian seperti memberikan presentasi, materi perkuliahan, dan sebagainya. Perangkat-perangkat tersebut tidak berdiri sendiri melainkan saling terhubung untuk menghasilkan kegunaan yang optimal. Masalah teknis seperti komputer atau proyektor yang tidak dapat berfungsi dengan baik, lampu yang sudah meredup kerap terjadi. Masalah lain yang juga dapat terjadi yaitu perangkat-perangkat tersebut tetap menyala walaupun sudah tidak digunakan akibat kelalaian pengguna. Masalah-masalah tersebut berdampak pada efektivitas dan efisiensi perangkat bahkan dapat merusak perangkat.

Penelitian bertujuan untuk membuat suatu sistem yang dapat memonitor dan mengendalikan LCD proyektor serta komputer di setiap ruang secara *remote* dan terpusat, juga untuk mengetahui *life time* dari lampu LCD proyektor di setiap ruang. Metodologi yang digunakan dalam penelitian adalah *study* literatur dan *eksperimental*.

Metodologi *study* literatur untuk mencari sumber-sumber di buku, Internet serta jurnal terkait dengan tema penelitian. Metodologi *eksperimental* untuk melakukan implementasi terhadap rancangan yang dihasilkan.



Bab III

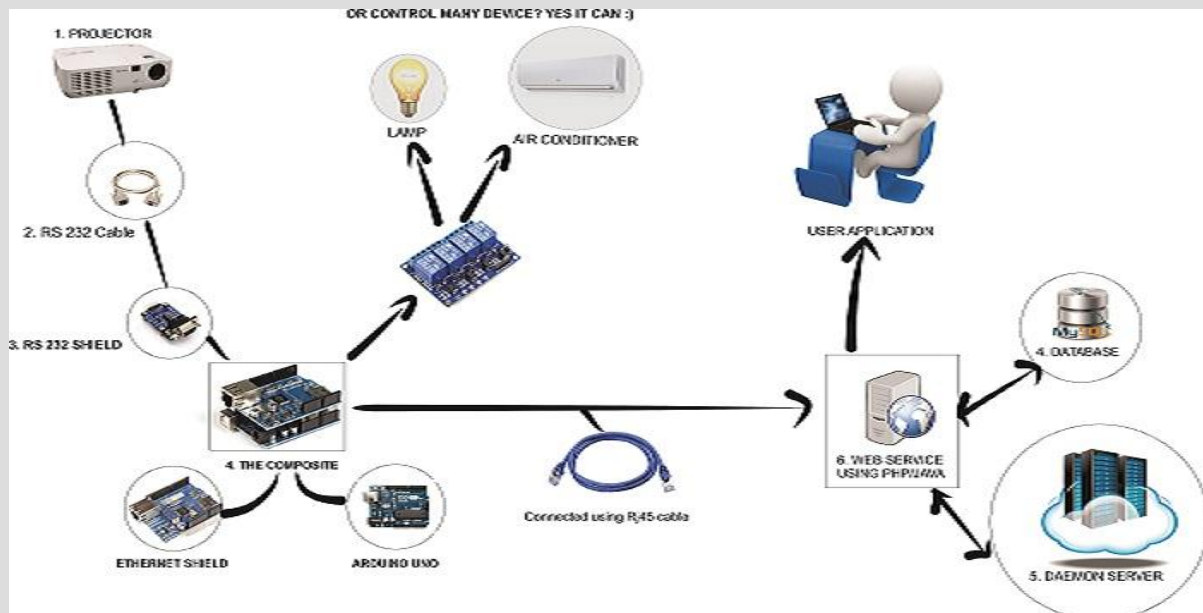
Analisis dan Perancangan

Pada bab ini dijelaskan mengenai analisis kebutuhan *software* dan analisis kebutuhan *hardware* dari sistem.

3.1 Analisis Sistem

Pada subbab ini akan dijelaskan proses penguraian masalah yang menjadi bagian yang digunakan untuk membangun sebuah sistem.

3.1.1 Sistem Overview



Gambar 6 High Level Sistem Overview

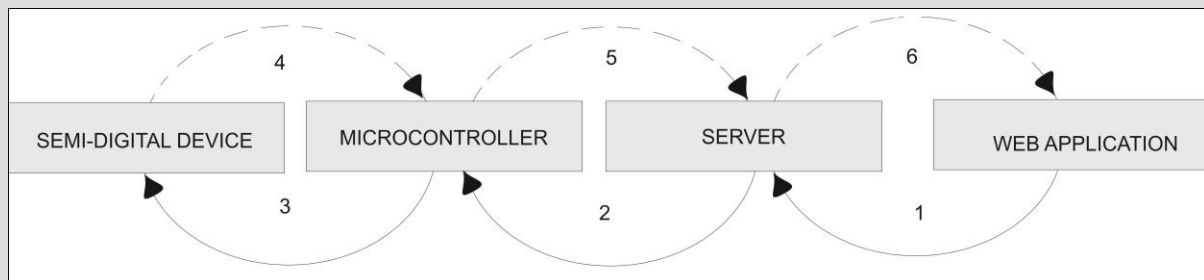
“Sistem Kontrol Perangkat Elektronik Semi Digital” secara umum dapat dilihat pada Gambar 7, sehingga dapat dijelaskan sebagai berikut.

1. Menghubungkan **Arduino Uno**, *Ethernet Shield*, dan *RS232 Shield* dengan cara menancapkan kaki *Ethernet Shield* ke port yang ada pada **Arduino Uno**. Setelah *Ethernet Shield* dan **Arduino Uno** terhubung, kemudian kita hubungkan *RS232 Shield* ke port yang ada pada *Ethernet Shield* sesuai dengan pin yang kita butuhkan.
2. Menghubungkan Proyektor dengan Mikrokontroller dengan menggunakan kabel RS232.
3. Menghubungkan Mikrokontroller dengan *Server* dengan menggunakan kabel RJ45.

4. Menghubungkan *User Application* dengan *server* dengan memanfaatkan *daemon* *Server* dan *database* yang ada pada *server*.

Pada Gambar 6 di atas, terdapat gambar lampu dan AC yang terhubung ke relay dan RS232, ini dapat dikembangkan jika pengontrolan proyektor selesai dikerjakan.

3.1.2 Deskripsi Umum Sistem



Gambar 7 Sistem Overview Sistem

“Sistem Kontrol Perangkat Elektronik Semi Digital” yaitu Pengontrolan perangkat elektronik adalah sistem yang digunakan untuk melakukan pengontrolan *on/off* proyektor. Sistem ini terdiri dari rangkaian mikrokontroler yang digunakan untuk melakukan pengontrolan perangkat elektronik dan hasil pengontrolan dikirimkan ke *server*. Setelah *command* dikirim oleh pengguna ke *server*, maka *server* mengirimkan *command* tersebut ke mikrokontroler dan mikrokontroler mengirimkan *command* tersebut ke perangkat elektronik semi digital. Sehingga perangkat elektronik semi digital bisa melaksanakan tugas yang diperintahkan oleh pengguna.

3.2 Analisis Kebutuhan

Pada subbab ini dijelaskan analisis kebutuhan pengguna dan kebutuhan sistem.

3.2.1 Analisis Kebutuhan Pengguna

Analisis kebutuhan pengguna berisi hal-hal yang diberikan oleh sistem. Hal-hal yang dibutuhkan oleh pengguna tersebut adalah sebuah sistem yang dapat melakukan sistem kontrol perangkat elektronik untuk *on/off* dari sebuah *device* dan lampu. Hasil pengontrolan yang dilakukan oleh sistem tersebut dikirimkan ke *server* dan dikelola oleh *server* sehingga hasil tersebut ditampilkan melalui *server* tersebut secara *real time*.

Sistem yang diharapkan dapat melakukan fungsi-fungsi berikut :

1. Memonitoring status proyektor dan lampu.

2. Menyimpan informasi tentang hasil pengontrolan proyektor dan lampu.
3. Menampilkan data hasil pengukuran secara *real time*.

3.2.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem meliputi analisis kebutuhan perangkat keras (*hardware*) dan analisis kebutuhan perangkat lunak (*software*).

3.2.2.1 Analisis Kebutuhan Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan dalam merancang dan membangun sistem dapat dilihat pada tabel berikut :

Tabel 3 Perangkat Keras yang dibutuhkan

No	Nama Perangkat	Keterangan
1	Arduino Uno	Arduino Uno digunakan sebagai mikrokontroler utama.
2	Serial Port RS232	Perangkat yang berfungsi untuk menghubungkan antara proyektor dengan mikrokontroler
3	Ethernet Shield	Perangkat yang berfungsi untuk menghubungkan arduino dengan jaringan Internet serta memberikan IP address pada perangkat
4	Kabel LAN (UTP)	Berfungsi sebagai media penghubung antara mikrokontroler dengan Ethernet Shield

3.3.2.2 Analisis Kebutuhan Perangkat Lunak (*Software*)

Pada pengontrolan Proyektor ini diperlukan beberapa perangkat lunak (*software*) sebagai berikut :

1. Sistem Operasi

Sistem operasi yang digunakan dalam perangkat ini adalah CentOS 7. CentOS adalah sebuah distro *Linux* berorientasi *server* yang didasarkan pada RHEL (Red Hat Enterprise Linux).

2. MySQL

Sebuah *service* yang ada pada *server* yang digunakan untuk menampung *database* yang digunakan oleh *server* tersebut. MySQL dapat digunakan untuk membuat dan mengelola *database* beserta isinya dan dapat memanfaatkan MySQL untuk menambahkan, mengubah, dan menghapus data yang berada dalam *database*.

3. Browser

Sistem pengontrolan terhadap perangkat elektronik dilakukan secara *online*. Maka dibutuhkan browser yang digunakan untuk mengakses IP dari perangkat elektronik untuk melakukan *monitoring*.

4. Arduino

Arduino *software* merupakan *software portable IDE* (*Integreted Development Environment*) yang bisa langsung didownload. Program ini ditulis dengan bahasa C yang sudah disederhanakan sehingga lebih mudah dalam mempelajari dan proses pembuatannya. Program yang sudah dibuat kemudian *dicompile* untuk memastikan program sudah tidak ada kesalahan (*error*). Program yang sudah *dicompile* tersebut kemudian *diupload* ke mikrokontroler Arduino menggunakan *port* USB yang menghubungkan antara PC dengan mikrokontroler Arduino. Program yang sudah di *upload* akan tersimpan pada memori Arduino yang selanjutnya akan mengeksekusi sistem agar dapat bekerja sesuai program yang dibuat.[12]

5. XAMPP

Sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri atas program Apache HTTP Server, MySQL database, dan penerjemahan bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP merupakan singkatan dari X (empaty sistem operasi apapun), Apache, MySQL, PHP dan Perl.[13]

3.3 Analisis Projector yang akan dikontrol

LCD Proyektor yang akan dikontrol pertama dihubungkan menggunakan serial port RS232. Untuk menghubungkan *Projector* dengan kabel RSS232 harus mengikuti langkah-langkah berikut :

1. Komputer dan proyektor harus dalam keadaan *off* sebelum kabel RS232 terhubung.
2. Kemudian hidupkan proyektor setelah komputer siap untuk digunakan.
3. Kemudian tentukan pengaturan komunikasi perangkat yang digunakan untuk mengirim dan menerima perintah, seperti yang ditunjukkan pada table di bawah.

Tabel 4 Pengaturan Komunikasi Perangkat

Item	Detail
Baud Rate	115200/38400/19200/9600/4800 bps
Data Length	8 bits
Parity bit	None
Stop bit	1 bit
Communication mode	Full Duplex

3.3.1 *Command* yang digunakan untuk mengontrol LCD Proyektor NEC

a. Perintah untuk menghidupkan LCD Proyektor:

Command : 02h 00h 00h 00h 00h 02h

Respon yang berikan Proyektor :

1. Ketika *command* sukses : 22h 00h <ID1> <ID2> 00h <CKS>
2. Ketika *command fails* : A2h 00h <ID1> <ID2> 02h <ERR1> <ERR2> <CKS>

b. Perintah untuk mematikan LCD Proyektor:

Command : 02h 01h 00h 00h 00h 03h

Respon yang diberikan Proyektor:

1. Ketika *command* sukses : 22h 01h <ID1> <ID2> 00h <CKS>
2. Ketika *command fails* : A2h 01h <ID1> <ID2> 02h <ERR1> <ERR2> <CKS>

3.4 Perancangan Sistem

“Sistem Kontrol Perangkat Elektronik Semi Digital” menggunakan mikrokontroler membutuhkan rangkaian mikrokontroler yang terdiri dari RS232 *Shield*, *Ethernet Shield*, **Arduino Uno**, dan PC sebagai *server*. Pusat operasi dari proses-proses yang dilakukan dalam sistem ini adalah PC. Pada mikrokontroler ditanamkan kode program yang berfungsi mengendalikan kinerja proyektor dan lampu.

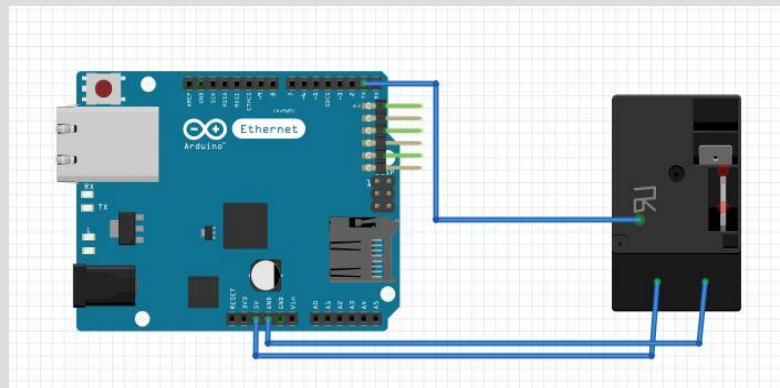
Cara kerja sistem ini, yaitu :

1. Mikrokontroler menerima tegangan sebesar 5 volt dan membaginya ke *Ethernet* dan RS232 *Shield* supaya bisa menerima setiap perintah atau program yang dimasukkan.
2. Perintah yang diterima oleh mikrokontroller akan diproses dan akan dilanjutkan ke perangkat elektronik yang akan dikontrol.

3. Setelah perangkat elektronik semi digital menerima perintah, maka akan langsung memberikan tanggapan ke mikrokontroller.
4. Untuk menampilkan hasil pengontrolan proyektor dan lampu oleh mikrokontroller digunakan aplikasi berbasis *web* yang dibangun menggunakan **Yii2 Framework** dalam bentuk pernyataan apakah perangkat semi digital dalam keadaan *on/off*.

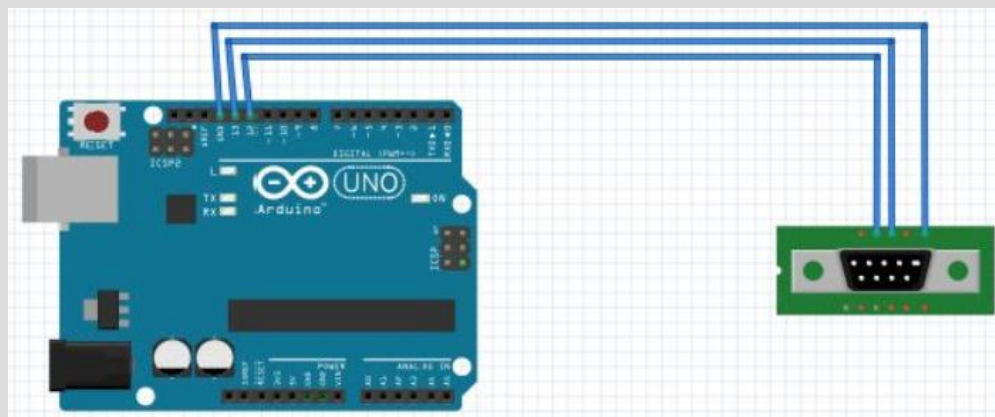
Berikut desain skematik perangkat yang digunakan dalam “Sistem Kontrol Perangkat Elektronik Semi Digital”.

Ethernet shield merupakan alat yang digunakan untuk membantu Arduino agar dapat berkomunikasi dengan laptop sehingga proses ini merupakan proses membuat rangkain dan menghubungkan pin-pin yang ada pada Ethernet *shield* dengan *relay* agar Arduino dapat memerintahkan *relay* saat Arduino menerima perintah. Berikut desain skematik mikrokontroler dengan *relay* dapat dilihat pada Gambar 8 di bawah.



Gambar 8 Desain Skematik Mikrokontroler dengan Relay

Berikut desain skematik mikrokontroler dengan RS232 shield dapat dilihat pada Gambar 9 di bawah.



Gambar 9 Desain Skematik Mikrokontroler dengan RS232 Shield

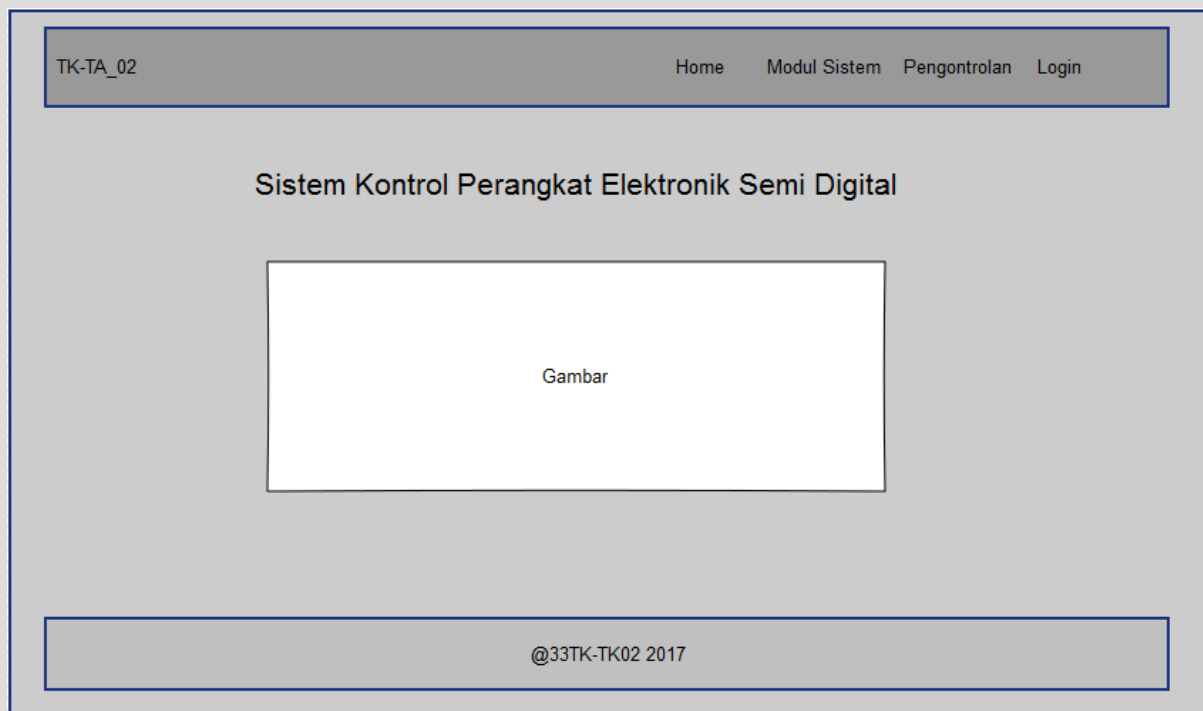
3.5 Perancangan Aplikasi Berbasis Web

Pada sub bab ini dijelaskan rancangan *Sistem Kontrol Perangkat Elektronik Semi Digital*.

Berikut ini adalah perancangan aplikasi berbasis *web*.

1. Aplikasi pengontrolan perangkat elektronik semi digital merupakan aplikasi *web* yang digunakan untuk menampilkan hasil pengontrolan proyektor dan lampu dengan *link* yang dapat diakses. Aplikasi pengontrolan proyektor ini nantinya akan dapat diakses oleh pengguna yang terhubung dengan *web server*.
2. Aplikasi tersebut dibangun dengan menggunakan PHP yang akan diisi dengan *action-action* yang akan melengkapi halaman *web servicenya*. Seperti *actionLogin*, *actionIndex*, dll.
3. Menampilkan hasil pengontrolan proyektor oleh *user*.

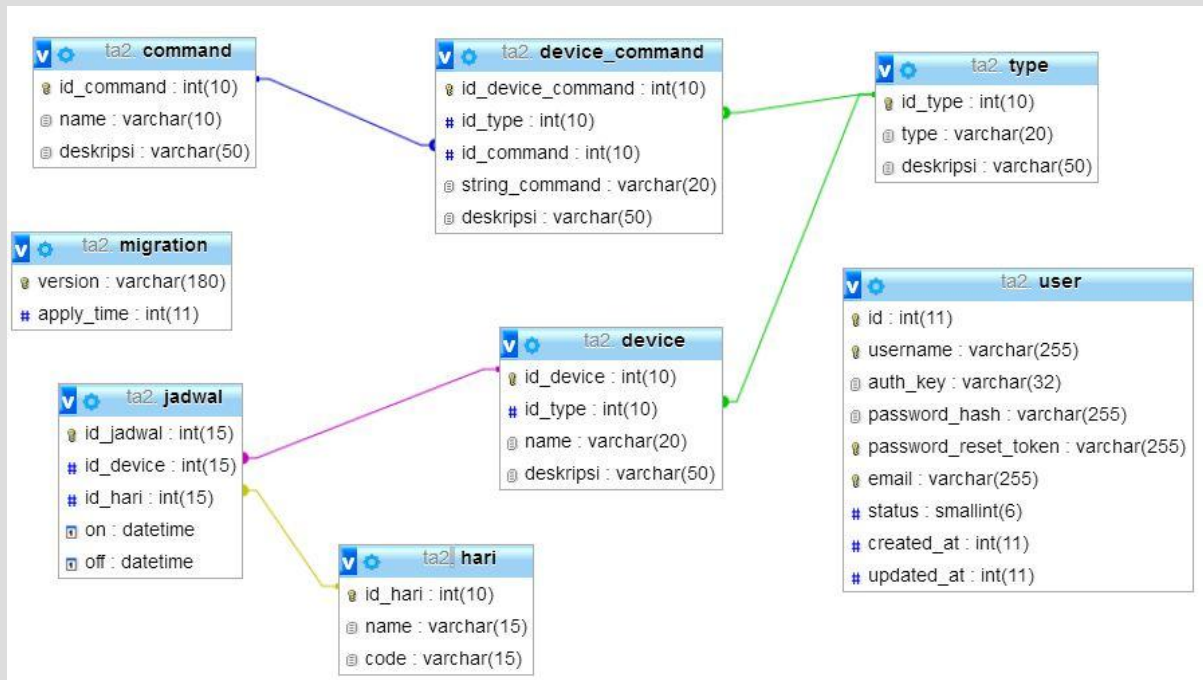
Berikut desain antarmuka perancangan aplikasi berbasis web.



Gambar 10 Desain antarmuka aplikasi

3.6 Perancangan Database

Pada subbab ini akan dijelaskan mengenai tabel *database* yang dibutuhkan untuk menyimpan data pada “Sistem kontrol Perangkat Elektronik Semi Digital” menggunakan mikrokontroler. Berikut merupakan Gambar 12 yang dibutuhkan dalam pengerjaan Tugas Akhir adalah :



Gambar 11 Database

Berikut keterangan dari setiap tabel di atas :

1. **Tabel type**, terdapat id_type, type, dan deskripsi. Pada table ini akan diisi *type device* yang akan digunakan misalnya LCD Proyektor dengan *type* NEC.
2. **Tabel device**, terdapat id_device, id_type, name, dan deskripsi. Pada tabel ini akan diisi nama *device* yang akan digunakan, misalnya LCD Proyektor-GD513, lampu Philips.
3. **Tabel command**, terdapat id_command, name, dan deskripsi. Pada tabel ini akan diisi command *on/off* pada setiap *device* yang akan dikontrol.
4. **Tabel device_command**, terdapat id_device_command, id_type, id_command, string_command, dan deskripsi. Pada tabel ini akan diisi *string command* yang akan dikirimkan pada saat pengontrolan dari setiap *device*.
5. **Tabel jadwal**, terdapat id_jadwal, id_device, id_hari, on, off. Pada tabel ini akan diisi jadwal pengontrolan *device* yang akan dikontrol.
6. **Tabel hari**, terdapat id_hari, name, dan code. Pada tabel ini akan diisi hari apa *device* yang akan dikontrol.

3.7 Flowchart

Flowchart pada sistem pengontrolan menggambarkan aliran proses pengaturan hidup dan matinya perangkat elektronik. Gambar 13 di bawah menampilkan *flowchart* dari sistem pengontrolan yang memiliki tiga peranan penting, diantaranya adalah kontrol hidup dan mati perangkat elektronik, sinkronisasi kondisi hidup dan mati perangkat dengan *database*.



Gambar 12 Flowchart

Bab IV

Implementasi dan Pengujian

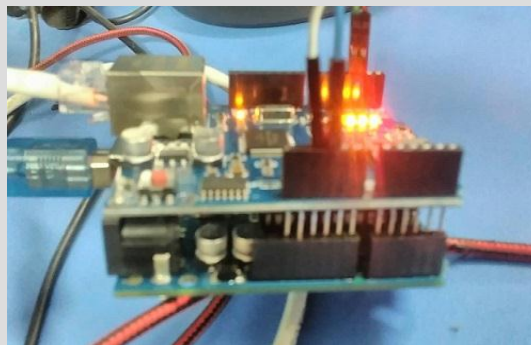
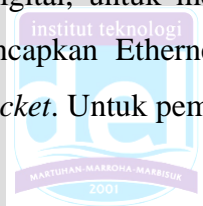
Pada bab ini dijelaskan mengenai implementasi terhadap sistem pengontrolan, pengujian implementasi yang telah dilakukan. Proses implementasi yang dilakukan berupa instalasi serta konfigurasi *web server*, pembuatan kode program. Menghubungkan *web* aplikasi dengan *web server* dan selanjutnya akan dilaksanakan pengujian terhadap implementasi yang telah dilakukan.

4.1 Implementasi Rangkaian Mikrokontroler

Pada subbab ini dijelaskan mengenai implementasi yang dilakukan untuk membangun Sistem Kontrol Perangkat Semi Digital menggunakan Mikrokontroler.

4.1.1 Rangkaian Mikrokontroler dengan Ethernet Shield

Rangkain mikrokontroler dengan Ethernet Shield, yaitu sistem kendali pada mikrokontroler memiliki 6 pin analog dan 14 pin digital, untuk menghubungkan Mikrokontroler dengan **Ethernet shield** hanya tinggal menancapkan Ethernet shield di atas Mikrokontroler saja karena keduanya sudah mempunyai *socket*. Untuk pemasangannya dapat dilihat pada Gambar 14 di bawah.

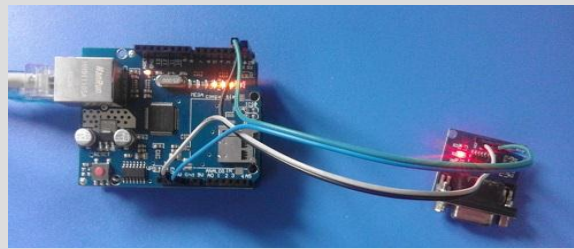


Gambar 13 Rangkaian Mikrokontroler dengan Ethernet Shield

4.1.2 Rangkaian Mikrokontroler dengan RS232 Shield

Rangkaian mikrokontroler dengan RS232 *shield*, yaitu pin 5V pada mikrokontroler dihubungkan dengan pin VCC pada RS232 shield. Pin RX pada mikrokontroler dihubungkan dengan pin RXD pada RS232 *shield* (*receiver*). Pin TX pada mikrokontroler dihubungkan dengan pin TXD pada RS232 *shield* (*transmitter*). Pin GND pada mikrokontoler

dihubungkan dengan pin GND pada RS232 *shield*. Rangkaian mikrokontroler dengan Ethernet shield dapat dilihat pada Gambar 14 di bawah.



Gambar 14 Rangkaian Mikrokontroler dengan RS232 *shield*

4.1.3 Rangkaian Mikrokontroler dengan LCD Proyektor

Rangkaian mikrokontroler dengan LCD Proyektor, yaitu mikrokontroler terhubung dengan RS232 *shield* dan RS232 *shield* dihubungkan ke LCD Proyektor melalui serial *port* RS232. Rangkaian mikrokontroler dengan LCD Proyektor dapat dilihat pada Gambar 16 di bawah.



Gambar 15 Rangkaian Mikrokontroler dengan LCD Proyektor

4.1.4 Rangkaian Mikrokontroler dengan *Relay*

Ethernet Shield dihubungkan dengan modul *relay* dengan menggunakan kabel *jumper male-female* 4 pin. Rangkaian mikrokontroler dengan *relay* dapat di lihat pada Gambar 17 di bawah.



Gambar 16 Rangkaian Mikrokontroler dengan Relay

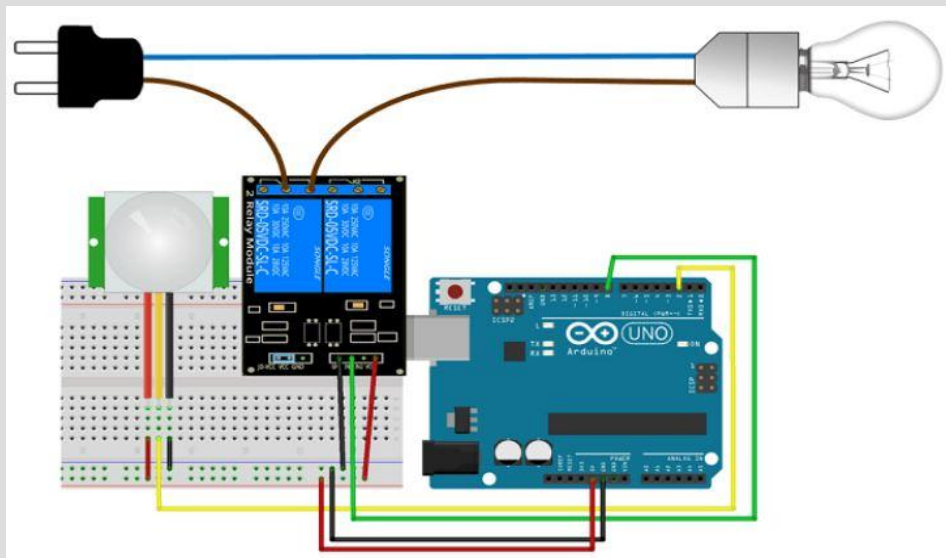
Koneksi dari masing-masing pin dapat dilihat pada table 5 di bawah:

Tabel 5 Koneksi Pin Relay

Ethernet Shield	Relay Module
Pin A0	IN1
Pin A1	IN2
5V	VCC
GND	GND

4.1.5 Rangkaian Relay Untuk Kontrol Lampu

Pengontrolan *on/off* lampu dapat dilakukan dari jarak jauh dengan mengakses *web browser* pada laptop dengan memasukkan URL yang telah ditentukan, untuk rangkaian *relay* akan dihubungkan ke **Arduino uno** pada pin yang sudah ditentukan. *Relay* akan bekerja apabila memperoleh *input* logika *High* dan *Low* dari Arduino. Untuk rangkaian *relay* pengontrol lampu dapat dilihat pada Gambar 18 berikut.



Gambar 17 Rangkaian *relay* untuk kontrol lampu

4.2 Implementasi Aplikasi Berbasis Web pada Server

Pada subbab ini dijelaskan implementasi aplikasi berbasis web dengan menggunakan **Yii2 Framework**.

4.2.1 Implementasi Aplikasi Pengontrolan menggunakan Framework Yii2

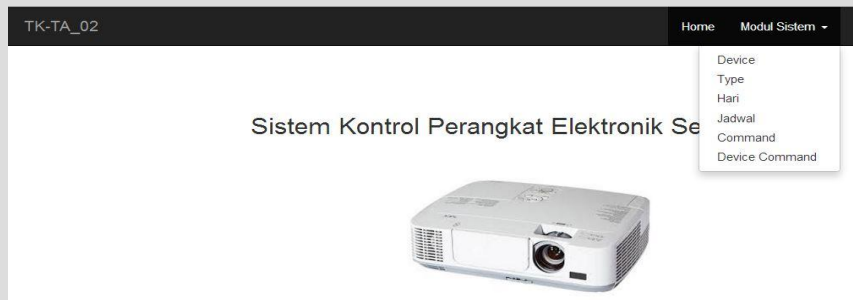
Aplikasi pengontrolan dibangun menggunakan **Yii2 Framework**. Data yang dikirimkan melalui serial komunikasi kemudian disimpan di *database*.

4.2.1.1 Beranda

Pada halaman ini ditampilkan modul *device* dan pengontrolan *device* sehingga setiap pengguna dapat memahami mengenai setiap *device* yang akan dikontrol nantinya.

4.2.1.1.1 Modul Device

Pada halaman ini terdapat modul *device*, yaitu terdiri dari *device*, *type*, *command*, *device_command*, hari, jadwal yang digunakan untuk mengisi data dari setiap *device* yang akan dikontrol dan dihubungkan ke *database*. Berikut tampilan dari modul *device* dapat dilihat pada Gambar 19.



Gambar 18 Modul Sistem

4.2.1.1.2 Pengontrolan

Pada halaman ini terdapat pengontrolan dari setiap *device* yang akan dikontrol untuk *on/off*, dan *timer* dari setiap *device*. Berikut tampilan dari pengontrolan *device*.

#	Nama Device	Type	Status	Kontrol Perangkat
1	Pro-521	NEC	off	On Off Timer
2	Lampu-513	Philips	on	On Off Timer

Gambar 19 Sistem Kontrol Device

Berikut tampilan dari *timer*, dimana *device* yang akan dikontrol dapat diset pada hari apa, mulai jam berapa dan sampai jam berapa.

#	Hari	Nama Device	On	Off
1	Senin	Pro-521	2017-03-04 08:00:00	2017-03-04 17:00:00
2	Senin	Lampu-513	2017-03-04 08:00:00	2017-03-04 17:00:00

Gambar 20 Timer pengontrolan device

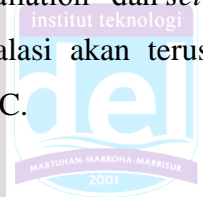
4.3 Instalasi

Pada subbab ini diuraikan tentang proses instalasi yang dilakukan dalam pengerjaan Sistem Kontrol Perangkat Elektronik Semi Digital.

4.3.1 Instalasi CentOS 7

Berikut ini adalah langkah-langkah instalasi Centos 7.

1. Atur agar BIOS *Boot* dari USB *Flashdisk* pertama kali.
2. Masukkan USB *Flashdisk Installer CentOS* ke dalam *port* USB.
3. Jalankan *server* dan mulai instalasi. Tampilan awal cukup sederhana, kemudian pilih “Install CentOS 7”.
4. Selanjutnya, pilih bahasa yang akan digunakan untuk proses instalasi. Pilih “default” saja.
5. Kemudian pilih “Timezone Asia/Jakarta”. Setelah dipilih klik “Done” pada bagian atas.
6. Pada *software*, pilih “Minimal Install”.
7. Selanjutnya untuk partisi pilih otomatis supaya tidak susah mempartisi.
8. Untuk *network* bisa diset setelah instalasi selesai.
9. Kemudian pilih tombol “Begin Installation” dan *set password* untuk *user root*.
10. Setelah semua dikonfigurasi, instalasi akan terus berjalan sampai selesai, setelah itu *restart*, kemudian cabut USB dari PC.



4.3.2 Instalasi Web Server

Berikut ini adalah langkah-langkah instalasi *web server*.

1. *Install* named dengan perintah “yum -y install named”.
2. *Install* Apache dengan perintah “yum -y install httpd”.
3. *Install* php dengan perintah “yum -y install php”.
4. *Install* MariaDB dengan perintah “yum -y install mariadb”.
5. *Install* Compiler (*MinGW* dan *GCC*) dengan perintah “yum -y install gcc” dan “yum -y install mingw64”.
6. *Install* Bind dengan perintah “yum -y install bind”.

4.3.3 Instalasi Yii Framework

Berikut ini adalah langkah-langkah instalasi **Yii Framework**.

1. *Install* terlebih dahulu *web server*, dalam tugas akhir ini yang digunakan adalah XAMPP.
2. *Copy paste* file *Yii Framework* ke *folder* xampp di-*install*, yaitu **C:\xampp\htdocs**.
3. Buatlah *folder* baru pada *folder* **C:\xampp\htdocs** dengan nama **tugasakhir**.

4. Buka “cmd” sebagai administrator. Kemudian ketik **init** kemudian **yii migrate**.
5. Aktifkan Apache dari *XAMPP Control Panel* dan ketikkan **http://localhost/tugasakhir** pada *browser*.

4.4 Konfigurasi

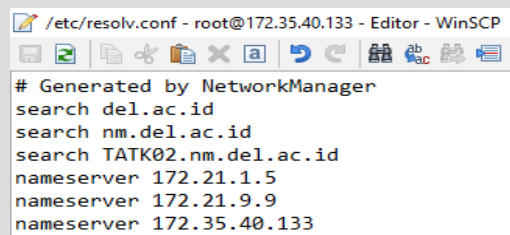
Pada subbab ini akan dijelaskan konfigurasi yang dilakukan terhadap instalasi yang dilakukan pada subbab sebelumnya. Konfigurasi dilakukan agar *software* yang sudah *diinstall* dapat berjalan sesuai fungsi dan kebutuhan. Berikut akan dijelaskan konfigurasi-konfigurasi yang dilakukan dalam pembangunan aplikasi yang dibangun.

4.4.1 Konfigurasi DNS Server

1. Pastikan *server* sudah ter-*install* dengan bind

```
root@localhost:~# rpm -q bind
```

2. Konfigurasi */etc/resolv.conf*



```
# Generated by NetworkManager
search del.ac.id
search nm.del.ac.id
search TATK02.nm.del.ac.id
nameserver 172.21.1.5
nameserver 172.21.9.9
nameserver 172.35.40.133
```

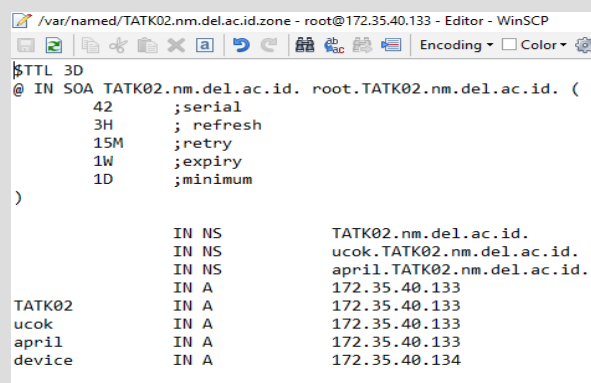
Gambar 21 Set Domain pada *resolv.conf*

3. Konfigurasi *Primary Server*

Konfigurasi */etc/named.conf*. Ada dua *zone* area yaitu :

2. Definisi file *Forward zone* yaitu untuk *mapping domain* ke *IP address*.
3. Definisi file *Reverse zone* yaitu untuk *mapping IP Address* ke *domain*.

Berikut Gambar 23 konfigurasi *zone* untuk *domain*.

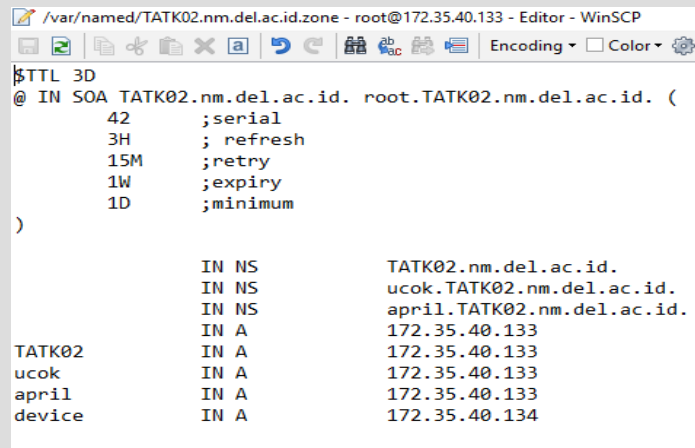


```
$TTL 3D
@ IN SOA TATK02.nm.del.ac.id. root.TATK02.nm.del.ac.id. (
    42      ; serial
    3H      ; refresh
    15M     ; retry
    1W      ; expiry
    1D      ; minimum
)

IN NS      TATK02.nm.del.ac.id.
IN NS      ucok.TATK02.nm.del.ac.id.
IN NS      april.TATK02.nm.del.ac.id.
TATK02     IN A      172.35.40.133
ucok       IN A      172.35.40.133
april      IN A      172.35.40.133
device     IN A      172.35.40.134
```

Gambar 22 Konfigurasi *zone* untuk *domain*

4. Konfigurasi *file zone* di **/var/named/TATK02.nm.del.ac.id.zone**. Berikut Gambar 24.



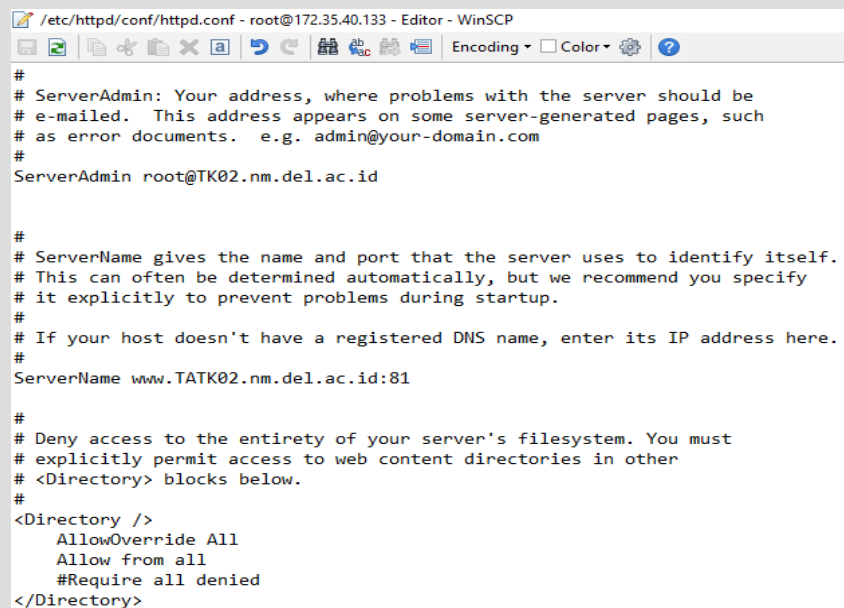
```
$TTL 3D
@ IN SOA TATK02.nm.del.ac.id. root.TATK02.nm.del.ac.id. (
    42      ;serial
    3H      ; refresh
    15M     ;retry
    1W      ;expiry
    1D      ;minimum
)

      IN NS      TATK02.nm.del.ac.id.
      IN NS      ucok.TATK02.nm.del.ac.id.
      IN NS      april.TATK02.nm.del.ac.id.
      IN A       172.35.40.133
TATK02      IN A       172.35.40.133
ucok        IN A       172.35.40.133
april       IN A       172.35.40.133
device      IN A       172.35.40.134
```

Gambar 23 Konfigurasi *file zone*

4.4.2 Konfigurasi Httpd

Berikut konfigurasi httpd yang dilakukan.



```
#
# ServerAdmin: Your address, where problems with the server should be
# e-mailed. This address appears on some server-generated pages, such
# as error documents. e.g. admin@your-domain.com
#
ServerAdmin root@TK02.nm.del.ac.id

#
# ServerName gives the name and port that the server uses to identify itself.
# This can often be determined automatically, but we recommend you specify
# it explicitly to prevent problems during startup.
#
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName www.TATK02.nm.del.ac.id:81

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
<Directory />
    AllowOverride All
    Allow from all
    #Require all denied
</Directory>
```

Gambar 24 Konfigurasi httpd

4.5 Pengujian

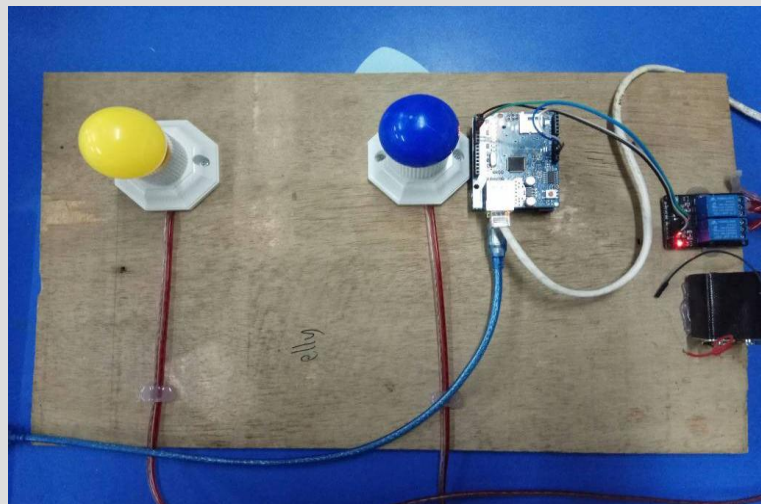
Pada subbab ini diuraikan tentang proses pengujian yang dilakukan dalam pengerjaan “Sistem Kontrol Perangkat Elektronik Semi Digital”. Pengujian adalah proses uji coba terhadap sistem yang dibangun mencakup rangkaian pengambilan data, transmisi data, dan penerimaan data.

4.5.1 Proses on/off Lampu 1

Pada proses ini sebelumnya lampu dalam keadaan mati. Jika ingin mematikan dan menhidupkan lampu *client* dapat mengakses **tatk02.nm.del.ac.id** pada *browser*. Dimana dalam proses pengontrolan terdapat Nama Device, Type Perangkat, Status perangkat, dan Kontrol perangkat. Berikut adalah tampilan pengontrolannya pada Gambar 27 di bawah.

TK-TA_02				
		Home	Modul Sistem	Pengontrolan Logout (admin)
Home / Daftar Perangkat				
Daftar Perangkat				
Showing 1-2 of 2 items.				
#	Nama Device	Type	Status	Kontrol Perangkat
1	Lampu Neon	Neon	off	<button>On</button> <button>Off</button> <button>Timer</button>
2	Lampu-513	Philips	off	<button>On</button> <button>Off</button> <button>Timer</button>

Gambar 25 Lampu Dalam Keadaan Mati

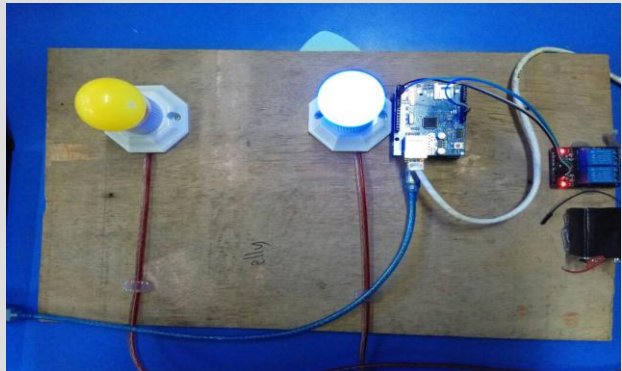


Gambar 26 Lampu Mati

Kemudian pada *browser* lampu 1 dihidupkan. Pada tampilan *web* berikut ada Nama Perangkat, Type Perangkat, Status Perangkat dan Kontrol Perangkat. Untuk menhidupkan lampu 1, pada pengontrolan perangkat kita tinggal mengklik *button on* yang ada di *web browser*, maka status perangkat akan berubah menjadi *on*. Jika ingin mematikan lampu 2 klik *button off* di *web browser*, maka statusnya akan berubah menjadi *Off*. Berikut tampilannya pada Gambar 28 di bawah.

TK-TA_02		Home	Modul Sistem	Pengontrolan	Logout (admin)
Home / Daftar Perangkat					
Daftar Perangkat					
Showing 1-2 of 2 items.					
#	Nama Device	Type	Status	Kontrol Perangkat	
1	Lampu Neon	Neon	off	On	Off
2	Lampu-513	Philips	on	On	Off

Gambar 27 Lampu 1 dalam keadaan Hidup



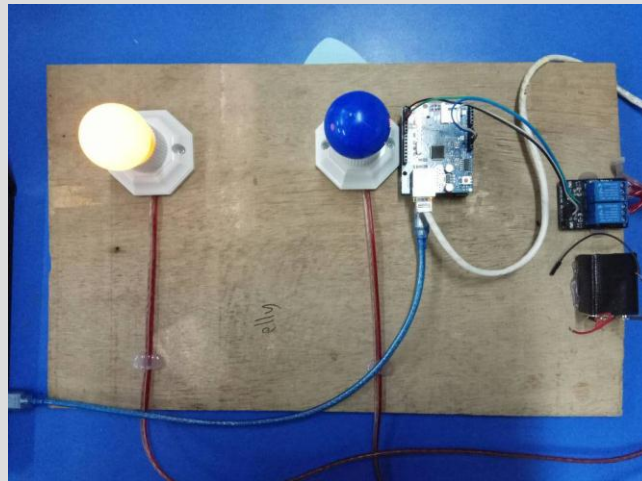
Gambar 28 Lampu 1 Hidup

4.5.2 Proses on/off Lampu 2

Untuk mengakses “Sistem Kontrol Perangkat Elektronik Semi Digital”, IP **tatk02.nm.del.ac.id** diakses pada *browser*. Kemudian pada *browser* lampu 1 dihidupkan. Pada tampilan berikut ada Nama Perangkat, Type Perangkat, Status Perangkat, dan Kontrol Perangkat. Untuk menghidupkan lampu 1, pada *web browser* kita dapat mengklik *button On*, maka pada status perangkat akan berubah menjadi *on*, dan jika ingin mematikan lampu 1 klik kita tinggal mengklik *button off* maka status perangkat akan berubah menjadi *off*. Berikut tampilannya pada Gambar 30 di bawah.

TK-TA_02		Home	Modul Sistem	Pengontrolan	Logout (admin)
Home / Daftar Perangkat					
Daftar Perangkat					
Showing 1-2 of 2 items.					
#	Nama Device	Type	Status	Kontrol Perangkat	
1	Lampu Neon	Neon	on	On	Off
2	Lampu-513	Philips	off	On	Off

Gambar 29 Lampu 2 dalam keadaan Hidup



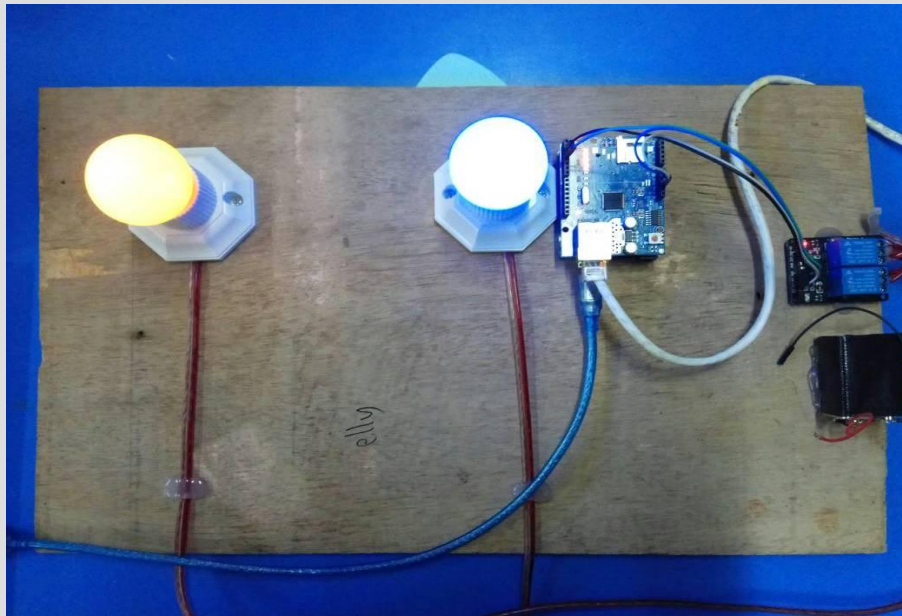
Gambar 30 Lampu 2 Hidup

4.5.3 Proses *on/off* semua lampu

Pada proses ini semua lampu dalam keadaan hidup. Kemudian pada *browser* semua lampu dihidupkan. Pada tampilan berikut ada Nama Perangkat, Type Perangkat, Status Perangkat, dan Kontrol Perangkat, untuk menghidupkan kedua lampu, kita tinggal mengklik button *On* di bagian control perangkat, maka status dari kedua lampu akan berubah menjadi *On*. Jika ingin mematikan lampu klik “Off”. Berikut tampilannya pada Gambar 32 di bawah.

TK-TA_02		Home	Modul Sistem ▾	Pengontrolan	Logout (admin)
Home / Daftar Perangkat					
Daftar Perangkat					
Showing 1-2 of 2 items.					
#	Nama Device	Type	Status	Kontrol Perangkat	
1	Lampu Neon	Neon	on	On	Off
2	Lampu-513	Philips	on	On	Off

Gambar 31 Semua Lampu dalam keadaan Hidup



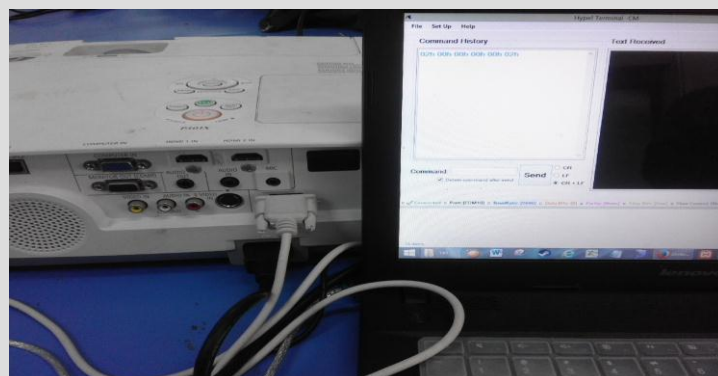
Gambar 32 Lampu 1 dan 2 Hidup

4.5.3 Proses *on/off* LCD Proyektor *via* RS232

Proses pengontrolan ini dilakukan dengan cara membuka terminal seperti *putty* untuk mengirimkan *command on/off* pada LCD Proyektor. *Command* dapat dikirimkan setelah semua kabel terhubung ke LCD Proyektor.



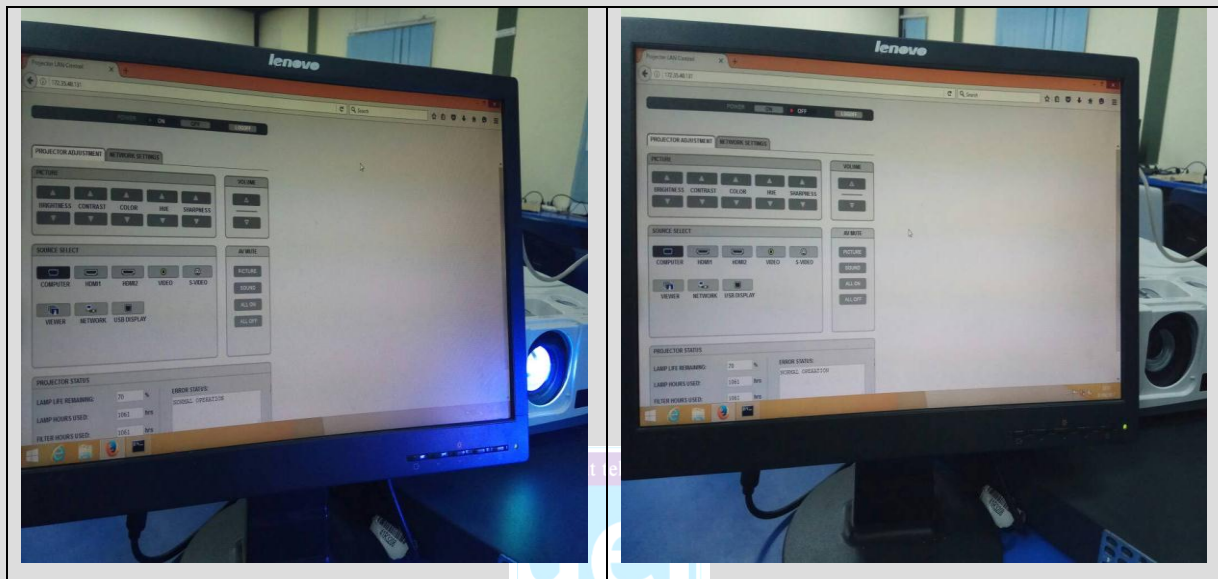
Berikut tampilannya pada Gambar 34 di bawah.



Gambar 33 Pengontrolan LCD Proyektor

4.5.4 Proses on/off LCD Proyektor via LAN

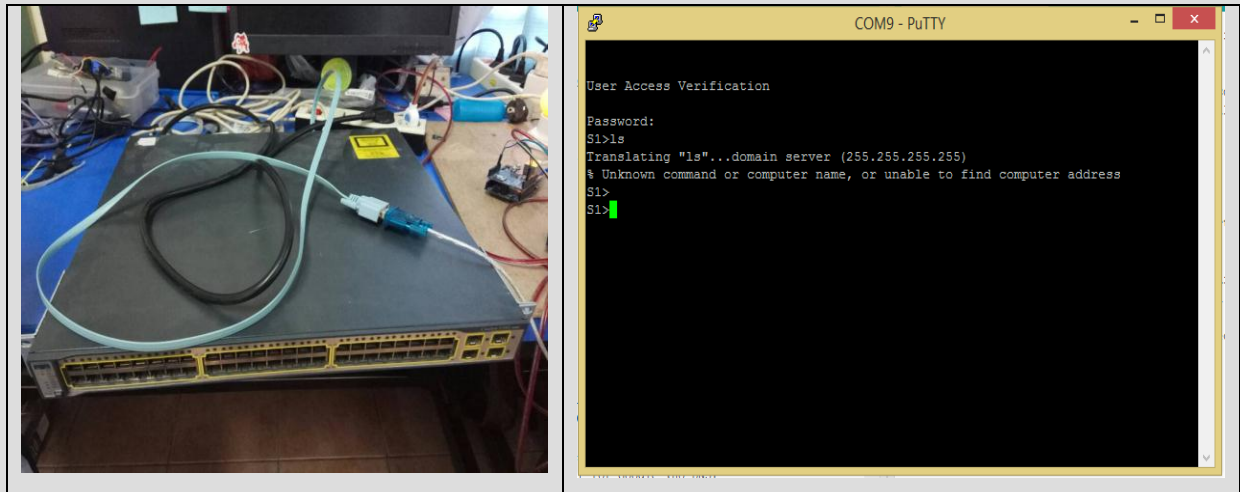
Proses pengontrolan ini dilakukan dengan cara mengakses IP yang telah diset pada LCD proyektor, kemudian LCD proyektor memberikan *web* seperti pada gambar dibawah, dan pada *web* tersebut terdapat *button on* dan *off* untuk mengontrol LCD Proyektor. Pada awalnya proyektor dalam keadaan mati, untuk menghidupkannya klik button on, dan untuk mematikan klik tombol *off*. LCD proyektor yang dikontrol pada tugas akhir ini adalah NEC. Hasil dari pengontrolan dapat dilihat pada Gambar 35 di bawah.



Gambar 34 Pengontrolan LCD proyektor via LAN

4.5.5 Pengujian Kabel serial RS232 pada Switch

Pada pengujian ini, dapat dibuktikan bahwa kabel *serial* RS232 dapat digunakan pada *switch* berikut. *Switch* dapat dikonfigurasi melalui kabel RS232 tersebut, dan tidak ada kesalahan pada kabel RS232. Hasil pengujian dapat dilihat pada Gambar 36 di bawah.



Gambar 35 Konfigurasi Switch via serial port RS232



Bab V

Kesimpulan dan Saran

Pada bab ini berisi kesimpulan dan saran mengenai Tugas Akhir yang sudah dilakukan, dan saran-saran yang akan disampaikan untuk masa yang akan datang bagi pengembang dan juga untuk pembaca.

5.1 Kesimpulan

Kesimpulan yang didapat dari hasil pengujian yang dilakukan terhadap “ Sistem Kontrol Perangkat Elektronik Semi Digital “ adalah sebagai berikut.

1. Peralatan atau *system* ini saat ini dapat digunakan untuk menyalakan, mematikan, *device* yang digunakan.
2. LCD Proyektor dapat dikontrol melalui LAN, dan menggunakan *button* melalui web yang diberikan langsung oleh LCD proyektor.
3. Kabel Serial RS232 dapat digunakan, dan diimplementasikan pada *switch* dengan baik.
4. Kabel Serial RS232 pada LCD proyektor tidak dapat diimplementasikan, LCD proyektor tidak memberi respon pada *serial port* RS232 tersebut.
5. Lampu dapat dikontrol melalui *relay* dan menggunakan *web server*.
6. LCD proyektor belum bisa dikontrol melalui RS232. Karena proyektor yang kami kontrol melalui terminal apapun tidak memberi respon.

5.2 Saran

Saran yang dapat diberikan sebagai berikut.

1. Memastikan bahwa perangkat tersebut dapat dikontrol menggunakan RS232.
2. Sebaiknya pada saat mengerjakan *web* aplikasinya, pastikan bahwa itu dapat terintegrasi dengan program Arduino.
3. Sebaiknya pada saat mengerjakan tugas akhir ini, pastikan bahwa semua perangkatnya telah terhubung dengan baik.

Daftar Pustaka

- [1] <https://fitripanp.wordpress.com/2011/09/27/pengertian-sistem-embedded/>, diakses pada 05 Mei 2017
- [2] <http://eprints.polsri.ac.id/2035/3/BAB%20II.pdf>, diakses pada 05 Mei 2017
- [3] <http://www.ntu.edu.sg/home/ehchua/programming/arduino/Arduino.html>, diakses pada 05 Mei 2017
- [4] <http://www.ebay.com/itm/Mini-MAX3232-RS232-Shield-Module-Arduino-Compatible-/270941700769>, diakses pada 05 Mei 2017
- [5] <http://www.vedcmalang.com/pppptkboemlg/index.php/menuutama/listrik-electro/1054-ds1>, diakses pada 05 Mei 2017
- [6] <http://dayatarduino.blogspot.co.id/2015/01/belajar-arduino-ethernet-shield-w5100.html>, diakses pada 05 Mei 2017
- [7] <http://www.indo-ware.com/produk-2788-modul-relay-2-indoware.html>, diakses pada 05 Mei 2017
- [8] Sukendar, Tateng. *Sistem Pengontrolan Alat Elektronik dengan media komputer dan Mikrokontroler AT89C51 melalui media Komunikasi Serial Port RS232*. Universitas Suryadarma.
- [9] Putri, dkk. *Simulasi pengembangan pengendali LCD Proyektor Berbasis Android menggunakan mikrokontroler Arduino di Universitas Kanjuruhan Malang*. Teknik Informatika, Universitas Kanjuruhan Malang.
- [10] Pratomo, Bina. 2011. *Pengembangan Sistem Kendali dan Akuisisi Jarak jauh Perangkat Elektronik Berbasis Jaringan IP*. Institut Pertanian Bogor.
- [11] Robby, dkk. *Perancangan Sistem Monitoring LCD Proyektor dan Komputer secara Terpusat*. Universitas Bina Nusantara.
- [12] Okky, Junaedy. 2015. *Sistem kontrol rumah pintar menggunakan Arduino Uno berbasis Android*. Politeknik Negeri Manado.
- [13] Bhakti, dkk. 2014. *Pembuatan SMS Gateway dengan menggunakan Arduino Uno dan GPRS Shield*. Bandung, Telkom University.
- [14] Wibowo, Satriyo. *Perancangan Sistem Kontrol Jarak Jauh Berbasis Web untuk memudahkan pengguna dalam pengendalian perangkat listrik rumah tangga*. Teknik Informatika, Sekolah Tinggi Informatika dan Komputer Indonesia.

- [15] Alamsyah, dkk.2015. *Perancangan dan Penerapan Sistem Kontrol Peralatan Elektronik Jarak Jauh Berbasis Web*. Universitas Tadulako, Palu.
- [16] https://www.google.co.id/search?q=belajar+serial+port+rs232&oq=belajar+serial+port+rs232_2&sourceid=chrome&ie=UTF-8#q=belajar+serial+port+rs232+pdf, diakses pada 10 November 2016.
- [17] <http://www.lapkomputer.blogspot.co.id/2015/02/pengertian-kabel-lan-jenis-jenis-kabel.html>, diakses pada 10 November 2016.
- [18] <https://id.m.wikipedia.org/wiki/XAMPP>, diakses pada 9 Januari 2017.



Lampiran A

Kode Program Pada Mikrokontroler

Berikut kode program pada Mikrokontroler untuk mengontrol lampu melalui relay.

```
#include <SPI.h>
#include <Ethernet.h>

int relay1 = 2;
int relay2 = 3;

boolean statusR1, statusR2 = false;

String readString;
char c;
byte mac[] = {0xDA, 0xAD, 0xBE, 0xEB, 0xFE, 0xFB};
IPAddress ip(172, 35, 40, 133);
EthernetServer server(80);

void setup() {
  pinMode(relay1, OUTPUT);
  pinMode(relay2, OUTPUT);

  digitalWrite(relay1, HIGH);
  digitalWrite(relay2, HIGH);

  Serial.begin(9600);
  while (!Serial) { ; }
  Ethernet.begin(mac, ip);
  server.begin();
}

void loop() {
  EthernetClient client = server.available();
  if (client) {
    while (client.connected()) {
      if (client.available()) {
        c = client.read();
        if (readString.length() < 100) {
          readString += c;
        }
      }
    }
  }
}
```




```

    }
    if (c == '\n') {
        client.println("HTTP/1.1 200 OK");
        client.println("Content-Type: text/html");
        client.println();
        client.println("<HTML>");
        client.println("<HEAD>");
        client.println("<meta      http-equiv=\"Refresh\"      content=\"2;
url=/TATK02\" />");
        client.println("<meta      http-equiv=\"Content-Type\"
content=\"application/vnd.wap.xhtml+xml; charset=utf-8\" />");
        client.println("<link      type=\"text/css\"      rel=\"stylesheet\"
href=\"http://haidao.mw.lt/js_css/css_pages.css\" />");
        client.println("<TITLE>Ruang Control</TITLE>");
        client.println("</HEAD>");
        client.println("<BODY align=\"center\">");
        client.println("<h1>Sistem Kontrol Perangkat Semi Digital </h1>");
        client.println("<hr />");
        client.println("<br />");
        client.println("<br />");
        client.println("<table align=\"center\"><tr>");
        client.println("<th width=\"10%\">Nama Perangkat</th>");
        client.println("<th width=\"10%\">Status Perangkat</th>");
        client.println("<th width=\"10%\">Saklar ON/OFF</th></tr>");
        client.println("<tr class=\"trh1\"><td align=\"left\">");
        client.println(" ");
        client.println(" ");
        client.println("Relay 1");
        client.println("</td><td align=\"center\">");
        if (statusR1 == false) {
            client.println("<b>MATI</b>");
        } else {
            client.println("<b>HIDUP</b>");
        }
        client.println("</td><td align=\"center\">");
        if (statusR1 == false) {
            client.println("<a href=\"/lon\" />HIDUP</a>");
        } else {
            client.println("<a href=\"/loff\" />MATI</a>");
        }
        client.println("</td></tr>");
    }

```

```

client.println();
client.println("<tr class=\"trh2\"><td align=\"left\">");
client.println(" ");
client.println(" ");
client.println("Relay 2");
client.println("</td><td align=\"center\">");
if (statusR2 == false) {
    client.println("<b>MATI</b>");
} else {
    client.println("<b>HIDUP</b>");
}
    client.println("</td><td align=\"center\">");
if (statusR2 == false) {
    client.println("<a href=\"/2on\" />HIDUP</a>");
} else {
    client.println("<a href=\"/2off\" />MATI</a>");
}
client.println("</td></tr>");
client.println();
client.println("<tr class=\"trh1\"><td align=\"left\">");
client.println(" ");
client.println(" ");
client.println("</BODY>");
delay(1);
client.stop();

if (readString.indexOf("1on") >0){
    digitalWrite(relay1, LOW);
    statusR1 = true;
}
if (readString.indexOf("1off") >0){
    digitalWrite(relay1, HIGH);
    statusR1 = false;
}
if (readString.indexOf("2on") >0){
    digitalWrite(relay2, LOW);
    statusR2 = true;
}
if (readString.indexOf("2off") >0){
    digitalWrite(relay2, HIGH);
    statusR2 = false;
}

```



```
}

readString=""; } } } }
```

Kode Program *Web Client*

```
#include <SPI.h>
#include <Ethernet.h>
// Enter a MAC address for your controller below.
// Newer Ethernet shields have a MAC address printed on a sticker on the
shield
byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
// if you don't want to use DNS (and reduce your sketch size)
// use the numeric IP instead of the name for the server:
IPAddress server(172,35,40,133); // numeric IP for Google (no DNS)
char server[] = "http://TATK02.nm.del.ac.id"; // name address for Google
(using DNS)
// Set the static IP address to use if the DHCP fails to assign
IPAddress ip(172, 35, 40, 13);
// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;
void setup() {
// Open serial communications and wait for port to open:
Serial.begin(9600);
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}
// start the Ethernet connection:
if (Ethernet.begin(mac) == 0) {
Serial.println("Failed to configure Ethernet using DHCP");
// try to configure using IP address instead of DHCP:
Ethernet.begin(mac, ip);
}
// give the Ethernet shield a second to initialize:
delay(1000);
Serial.println("connecting...");
// if you get a connection, report back via serial:
if (client.connect(server, 80)) {
```

```
Serial.println("connected");
// Make a HTTP request:
client.println("GET /search?q=arduino HTTP/1.1");
client.println("Host: www.google.com");
client.println("Connection: close");
client.println();
} else {
// if you didn't get a connection to the server:
Serial.println("connection failed");
}
}

void loop() {
// if there are incoming bytes available
// from the server, read them and print them:
if (client.available()) {
char c = client.read();
Serial.print(c);
}
// if the server's disconnected, stop the client:
if (!client.connected()) {
Serial.println();
Serial.println("disconnecting.");
client.stop();
// do nothing forevermore:
while (true);
}
}
```

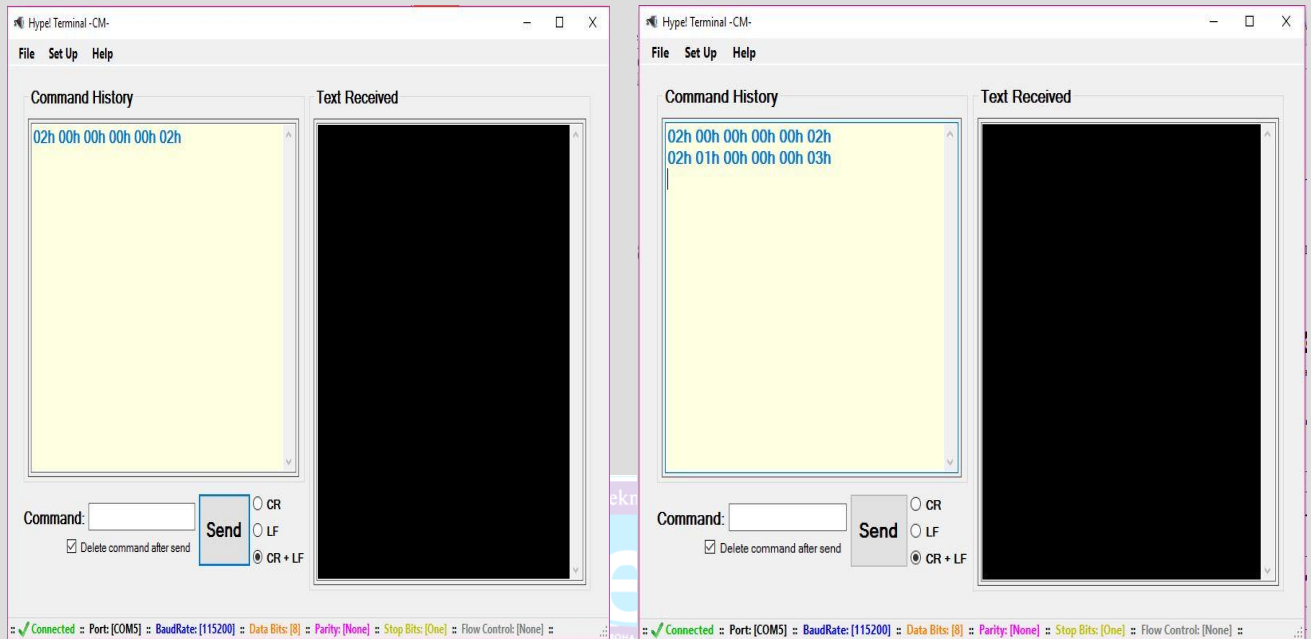
Lampiran B

Pengujian Pengontrolan Proyektor

Pada subbab ini dijelaskan mengenai pengujian yang dilakukan pada pengontrolan proyektor.

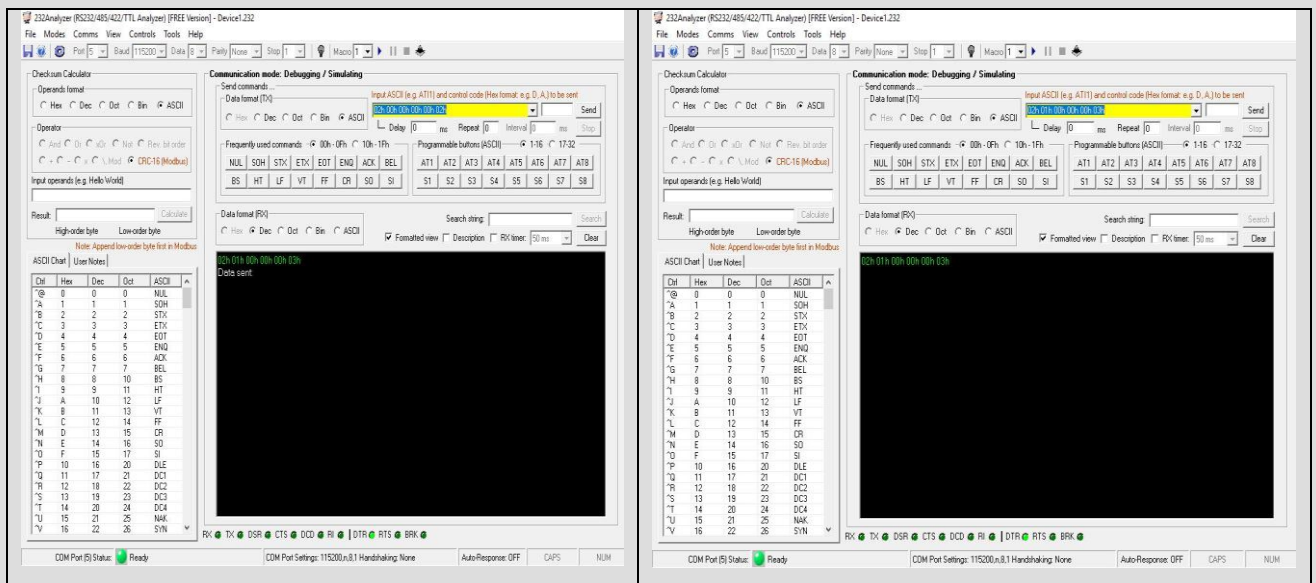
Berikut hasil pengujian yang telah dilakukan melalui terminal yang berbeda-beda.

1. Pengontrolan Proyektor melalui *Hype!Terminal*



Gambar 36 Control on/off Projector via Hype!Terminal

2. Pengontrolan Proyektor melalui 232Analyzer



Gambar 37 Control on/off Projector via 232Analyzer